開發人員指南

AWS IoT FleetWise

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT FleetWise: 開發人員指南

Copyright © 2025 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 IoT AWS IoT FleetWise 的新手嗎？ 3
存取 AWS IoT FleetWise 3
AWS IoT FleetWise 定價 3
相關服務 3
重要概念 4
重要概念 4
AWS IoT FleetWise 的功能 8
支援 AWS 的區域 8
設定 AWS IoT FleetWise 10
設定您的 AWS 帳戶 10
註冊 AWS 帳戶 10
建立具有管理存取權的使用者 10
在主控台中開始使用 12
設定您的設定 12
進行設定 (主控台) 12
設定 設定 (AWS CLI) 13
搭配 AWS IoT FleetWise 使用 IPv6 14
控制平面端點的 IPv6 先決條件 14
AWS PrivateLink 端點的 IPv6 支援 14
測試 IPv6 地址相容性 15
在 IAM 原則中使用 IPv6 地址 15
使用雙堆疊端點 16
開始使用 18
簡介 18
先決條件 18
步驟 1：設定 Edge Agent 軟體 for AWS IoT FleetWise 19
步驟 2：建立車輛模型 20
步驟 3：建立解碼器資訊清單 22
步驟 4：設定解碼器資訊清單 23
步驟 5：建立 車輛 23
步驟 6：建立行銷活動 25

步驟 7：清除 26
後續步驟 26
擷取資料 27
模型車輛 30
訊號目錄 32
設定訊號 35
建立訊號目錄 40
匯入訊號目錄 45
更新訊號目錄 55
刪除訊號目錄 58
取得訊號目錄資訊 59
車輛模型 60
建立車輛模型 60
更新車輛模型 67
刪除車輛模型 68
取得車輛模型資訊 70
解碼器資訊清單 71
設定界面和訊號 73
建立解碼器資訊清單 76
更新解碼器資訊清單 85
刪除解碼器資訊清單 88
取得解碼器資訊清單資訊 90
管理 車輛 91
佈建 車輛 92
驗證 車輛 92
授權 車輛 94
預留主題 95
建立 車輛 99
建立車輛 （主控台） 99
建立 車輛 (AWS CLI) 101
建立多部車輛 104
更新車輛 105
更新多部車輛 108
刪除車輛 109
刪除車輛 （主控台） 109
刪除車輛 (AWS CLI) 110

取得車輛資訊 111
管理機群 113
建立叢集 114
將車輛與機群建立關聯 115
取消車輛與機群的關聯 116
更新機群 117
刪除機群 117
驗證機群刪除 118
取得機群資訊 119
使用行銷活動管理資料 122
建立行銷活動 127
建立行銷活動 （主控台） 128
建立行銷活動 (AWS CLI) 135
AWS IoT FleetWise 行銷活動的邏輯表達式 140
更新行銷活動 141
刪除行銷活動 142
刪除行銷活動 （主控台） 142
刪除行銷活動 (AWS CLI) 143
驗證行銷活動刪除 143
取得行銷活動資訊 143
儲存和轉送 144
建立資料分割區 145
上傳行銷活動資料 148
使用 AWS IoT 任務上傳資料 149
收集診斷問題碼資料 150
診斷問題碼關鍵字 151
為診斷問題碼建立資料收集行銷活動 153
診斷故障碼使用案例 155
視覺化車輛資料 159
處理傳送至 MQTT 主題的車輛資料 159
在 Timestream 中處理車輛資料 160
視覺化儲存在 Timestream 中的車輛資料 161
在 Amazon S3 中處理車輛資料 161
Amazon S3 物件格式 162
分析存放在 Amazon S3 的車輛資料 163
命令 165

命令概念 166
命令關鍵概念 166
命令執行狀態 168
車輛和命令 172
工作流程概觀 172
車輛工作流程 173
命令工作流程 176
（選用） 命令通知 177
建立和管理命令 178
建立命令資源 179
擷取命令的相關資訊 181
列出您帳戶中的命令 181
更新或取代命令資源 182
刪除命令資源 184
啟動和監控命令執行 184
更新命令執行結果 188
取得命令執行 189
列出您帳戶中的命令執行 191
刪除命令執行 192
範例：使用 命令 193
車輛轉向模式範例概觀 193
先決條件 194
使用遠端命令的 IAM 政策 194
執行 AWS IoT 命令 (AWS CLI) 196
清除 200
命令使用案例 202
建立不含參數的命令 203
建立具有參數預設值的命令 204
使用參數值建立命令 205
搭配狀態範本使用命令 206
上次已知狀態 208
建立狀態範本 209
將 an AWS IoT FleetWise 狀態範本與 車輛建立關聯 210
更新狀態範本 211
刪除狀態範本 213
取得狀態範本資訊 213

狀態範本操作 214
啟用和停用狀態資料收集 215
擷取車輛狀態快照 219
使用 MQTT 訊息處理上次已知狀態車輛資料 222
設定網路無關的資料收集 226
簡介 226
環境設定 226
資料模型 226
訊號目錄更新 227
車輛模型和解碼器 228
傳送命令 231
AWS CLI 和 SDKs 233
故障診斷 234
解碼器資訊清單問題 234
Edge 代理程式問題 237
問題：Edge Agent 軟體不會啟動。 237
問題：【ERROR】【IoTFleetWiseEngine：：connect】：【無法初始化持續性程式庫】. 238
問題：Edge Agent 軟體不會收集內建診斷 (OBD) II PIDs和診斷故障碼 DTCs)。 239
問題：Edge Agent for AWS IoT FleetWise 軟體不會從網路收集資料，或無法套用資料檢查規則。 239
問題：【ERROR】【AwsIotConnectivityModule：：connect】：【連線失敗並發生錯誤】或 【WARN】【AwsIotChannel：：send】：【無作用中 MQTT 連線。】 240
存放和轉送問題 240
問題：接收`AccessDeniedException`具有所有必要 IAM 許可的 240
問題：上傳至 AWS IoT Jobs 的資料會忽略 `endTime` 240
問題：上傳至 AWS IoT 任務的資料具有`REJECTED`執行狀態。 240
安全 241
資料保護 241
靜態加密 in AWS IoT FleetWise 242
傳輸中加密 243
AWS IoT FleetWise 中的資料加密 243
控制存取 254
授予 AWS IoT FleetWise 許可，以傳送和接收 MQTT 主題的資料 254
授予對 Amazon S3 目的地的 AWS IoT FleetWise 存取權 257
授予對 Amazon Timestream 目的地的 AWS IoT FleetWise 存取權 260
授予使用 產生命令承載的 AWS IoT Device Management 許可 AWS IoT FleetWise 263

身分和存取權管理 267
目標對象 267
使用身分驗證 268
使用政策管理存取權 271
AWS IoT FleetWise 如何與 IAM 搭配使用 273
身分型政策範例 280
故障診斷 283
API 許可參考 284
法規遵循驗證 295
恢復能力 296
基礎架構安全 296
透過介面 VPC 端點連線至 AWS IoT FleetWise 297
組態與漏洞分析 300
安全最佳實務 300
盡可能授予最低的許可 300
請勿記錄敏感資訊 300
使用 AWS CloudTrail 來檢視 API 呼叫歷史記錄 301
讓裝置的時鐘保持同步 301
監控 AWS IoT FleetWise 302
使用 CloudWatch 進行監控 302
使用 CloudWatch Logs 監控 306
在 CloudWatch 主控台中檢視 AWS IoT FleetWise 日誌 307
設定 記錄 313
CloudTrail 日誌 315
CloudTrail 中的AWS IoT FleetWise 資訊 316
了解 日誌檔案項目 316
文件歷史紀錄 318
..... cccxx

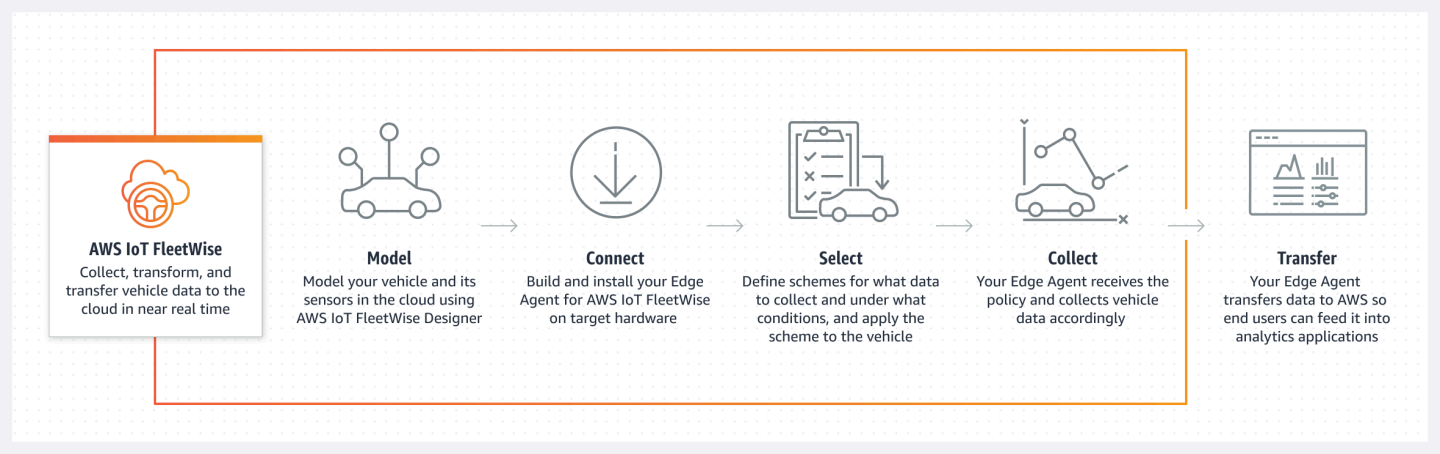
什麼是 AWS IoT FleetWise？

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

AWS IoT FleetWise 是一項受管服務，可用來收集車輛資料，並在雲端中進行組織。您可以使用收集的資料來改善車輛品質、效能和自主性。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組織資料。 AWS IoT FleetWise 有助於將低階訊息轉換為人類可讀的值，並將雲端中的資料格式標準化以進行資料分析。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，以及何時將這些資料傳輸至雲端。

當車輛資料位於雲端時，您可以將其用於分析車輛機群運作狀態的應用程式。此資料可協助您識別潛在的維護問題、讓車用資訊娛樂系統更智慧，以及透過分析和機器學習 (ML) 改善自動駕駛和駕駛協助系統等進階技術。

下圖顯示 AWS IoT FleetWise 的基本架構。



主題

- 優勢
- 使用案例
- 您是 IoT AWS IoT FleetWise 的新手嗎？
- 存取 AWS IoT FleetWise
- AWS IoT FleetWise 定價

- 相關服務
- AWS IoT FleetWise 的重要概念和功能
- AWSAWS IoT FleetWise 中的區域和功能可用性

優勢

AWS IoT FleetWise 的主要優點如下：

更智慧地收集車輛資料

透過僅將所需資料傳送至雲端進行分析的智慧型資料收集來改善資料相關性。

輕鬆分析標準化的整個機群資料

分析來自車輛機群的標準化資料，而不需要開發自訂資料收集或記錄系統。

雲端中的自動資料同步

取得從標準感應器（遙測資料）和視覺系統（攝影機、雷達和光圈的資料）收集的資料的統一檢視，並在雲端中自動同步。 AWS IoT FleetWise 可自動同步雲端中的結構化和非結構化視覺系統資料、中繼資料和標準感應器資料。這可簡化彙整事件全貌檢視的程序，並取得洞見。

將資料存放在 Edge，並在最佳條件下轉送

暫時將資料存放在車輛上，以降低傳輸成本。在指定的最佳條件下，您可以將選取的資料轉送至雲端，例如當車輛連線到 Wi-Fi 時。

> **Note**
>
> 視覺系統資料處於預覽版本，可能會有所變更。

使用案例

您可以使用 AWS IoT FleetWise 的案例包括下列項目：

訓練 AI/ML 模型

透過從生產車輛收集資料，持續改進用於自動和進階駕駛輔助系統的機器學習模型。

增強數位客戶體驗

使用資訊娛樂系統的資料，讓車載影音內容和應用程式內洞見更具相關性。

維護機群運作狀態

使用機群資料的洞見來監控 EV 電池運作狀態和充電狀態、管理維護排程、分析油耗等。

建立和管理命令

使用命令從雲端對車輛執行命令。您可以遠端將命令傳送至車輛，並在幾秒鐘內，車輛將執行命令。例如，您可以設定命令來鎖定車輛的門或設定溫度。

建立和管理狀態範本

狀態範本為車主提供追蹤其車輛狀態的機制。在車輛上執行的 AWS IoT FleetWise Edge 代理程式會收集並傳送訊號更新至雲端。

您是 IoT AWS IoT FleetWise 的新手嗎？

如果您是 IoT AWS IoT FleetWise 的新手，建議您先閱讀以下章節：

- AWS IoT FleetWise 的重要概念和功能
- 設定 AWS IoT FleetWise
- 教學課程： AWS IoT FleetWise 入門
- Ingest AWS IoT FleetWise 資料至雲端

存取 AWS IoT FleetWise

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

AWS IoT FleetWise 定價

車輛會透過 MQTT 訊息將資料傳送至雲端。您在每個月月底為您在 AWS IoT FleetWise 中建立的車輛付費。您還需要為從車輛收集的訊息付費。如需定價的最新資訊，請參閱 AWS IoT FleetWise 定價頁面。若要進一步了解 MQTT 訊息通訊協定，請參閱《 AWS IoT Core 開發人員指南》中的 MQTT。

相關服務

AWS IoT FleetWise 與下列 AWS 服務整合，以改善雲端解決方案的可用性和可擴展性。

- AWS IoT Core – 註冊和控制將車輛資料上傳至 AWS IoT FleetWise AWS IoT 的裝置，並遠端將命令傳送至車輛。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 Timestream 開發人員指南》中的什麼是 Amazon Timestream。
- Amazon S3 – 使用物件儲存服務來存放和管理 車輛資料。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的什麼是 Amazon S3。

AWS IoT FleetWise 的重要概念和功能

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列各節提供 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 一節以了解如何設定 AWS IoT FleetWise。

主題

- 重要概念
- AWS IoT FleetWise 的功能

重要概念

AWS IoT FleetWise 提供車輛建模架構，可讓您在雲端中建立車輛及其感應器和傳動器的模型。為了啟用 車輛與雲端之間的安全通訊， AWS IoT FleetWise 也提供參考實作，協助您開發可在車輛中安裝的 Edge Agent 軟體。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機制來控制要收集哪些資料，以及何時將其傳輸至雲端。

以下是 AWS IoT FleetWise 的核心概念。

Signal

訊號是您定義為包含車輛資料及其中繼資料的基本結構。訊號可以是屬性、分支、感應器或致動器。例如，您可以建立感應器來接收車載溫度值，以及存放其中繼資料，包括感應器名稱、資料類型和單位。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄。

屬性

屬性代表通常不會變更的靜態資訊，例如製造商和製造日期。

分支

分支代表巢狀結構中的訊號。分支示範訊號階層。例如，`Vehicle`分支具有子分支 `Powertrain`。`Powertrain` 分支具有子分支 `combustionEngine`。若要尋找`combustionEngine`分支，請使用 `Vehicle.Powertrain.combustionEngine`運算式。

感測器

感應器資料會報告車輛的目前狀態，並隨著車輛狀態的變化而變更，例如油位、溫度、振動或電壓。

傳動器

傳動器資料會報告車輛裝置的狀態，例如馬達、加熱器和門鎖。變更車輛裝置的狀態可以更新傳動器資料。例如，您可以定義一個作動器來代表加熱器。當您開啟或關閉加熱器時，傳動器會收到新資料。

自訂結構

自訂結構（也稱為結構）代表複雜或更高順序的資料結構。它有助於邏輯繫結或分組源自相同來源的資料。在原子操作中讀取或寫入資料時，會使用結構，例如 代表複雜的資料類型或更高順序的形狀。

結構類型的訊號是使用結構資料類型的參考而非基本資料類型在訊號目錄中定義。結構可用於所有類型的訊號，包括感應器、屬性、傳動器和視覺系統資料類型。如果傳送或接收結構類型的訊號， 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，如果結構包含項目 Vehicle.Camera.Image.height、 Vehicle.Camera.Image.width 和 Vehicle.Camera.Image.data – 預期傳送的訊號包含所有這些項目的值。

> **Note**
>
> 視覺系統資料處於預覽版本，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基本或另一個結構。

使用結構和自訂屬性來表示高階形狀時，預期的高階形狀一律會定義為樹狀結構，並視為視覺效果。自訂屬性用於定義所有分葉節點，而結構用於定義所有非分葉節點。

訊號目錄

訊號目錄包含一組訊號。訊號目錄中的訊號可用來建立使用不同通訊協定和資料格式的車輛模型。例如，不同汽車製造商製造的兩輛汽車：一輛使用控制區域網路 (CAN 匯流排） 通訊協定，另一輛則使用車載診斷 (OBD) 通訊協定。您可以在訊號目錄中定義感應器，以接收車載溫度值。此感應器可用來代表兩輛車中的熱電偶。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄。

車輛模型 （模型資訊清單）

車輛模型是宣告式結構，可用來標準化車輛的格式，並定義車輛訊號之間的關係。車輛模型在相同類型的多個車輛之間強制執行一致的資訊。您可以新增訊號來建立車輛模型。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛模型。

解碼器資訊清單

解碼器資訊清單包含車輛模型中每個訊號的解碼資訊。車輛中的感應器和傳動器會傳輸低階訊息（二進位資料）。使用解碼器資訊清單， AWS IoT FleetWise 能夠將二進位資料轉換為人類可讀取的值。每個解碼器資訊清單都與車輛模型相關聯。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

網路介面

包含車輛網路使用的通訊協定相關資訊。 AWS IoT FleetWise 支援下列通訊協定。

控制器區域網路 (CAN 匯流排）

定義資料在電子控制單元 (ECUs) 之間通訊方式的通訊協定。ECUs可以是引擎控制單元、安全氣囊或音訊系統。

內建診斷 (OBD) II

進一步開發的通訊協定，定義自我診斷資料如何在 ECUs 之間通訊。它提供多種標準診斷問題代碼 (DTCs)，可協助識別您車輛的問題。

車輛中介軟體

車輛中介軟體定義為一種網路界面類型。車輛中介軟體的範例包括機器人作業系統 (ROS 2) 和可擴展的服務導向 MiddlewarE over IP (SOME/IP)。

> **Note**
>
> AWS IoT FleetWise 支援 ROS 2 中介軟體來處理視覺系統資料。

自訂界面

您也可以使用自己的界面在 Edge 解碼訊號。這可以節省您的時間，因為您不需要在雲端中建立解碼規則。

訊號解碼器

提供特定訊號的詳細解碼資訊。車輛模型中指定的每個訊號都必須與訊號解碼器配對。如果解碼器資訊清單包含 CAN 網路介面，則必須包含 CAN 解碼器訊號。如果解碼器資訊清單包含 OBD 網路介面，則必須包含 OBD 訊號解碼器。

如果解碼器資訊清單也包含車輛中介軟體界面，則必須包含訊息訊號解碼器。或者，如果解碼器資訊清單包含自訂解碼介面，它也必須包含自訂解碼訊號。

車輛

實體車輛的虛擬呈現，例如汽車或卡車。車輛是 車輛模型的執行個體。從相同車輛模型建立的車輛會繼承相同的訊號群組。每部車輛對應至 實 AWS IoT 物。

機群

機群代表一組車輛。您必須先將個別車輛與機群建立關聯，才能輕鬆管理機群。

Campaign

包含資料收集方案。您可以在雲端中定義行銷活動，並將其部署到車輛或機群。行銷活動會提供 Edge Agent 軟體指示，說明如何選取、收集資料，以及將資料傳輸至雲端。

資料分割區

在行銷活動中設定分割資料，以暫時存放訊號資料。您可以設定將資料轉送至雲端的時機和方式。

資料收集方案

資料收集方案提供 Edge Agent 軟體如何收集資料的指示。目前， AWS IoT FleetWise 支援條件型收集方案和時間型收集方案。

條件型集合結構描述

使用邏輯表達式來識別要收集的資料。Edge Agent 軟體會在滿足條件時收集資料。例如，如果表達式為 `$variable.myVehicle.InVehicleTemperature >35.0`，Edge Agent 軟體會收集大於 35.0 的溫度值。

以時間為基礎的收集方案

指定以毫秒為單位的時段，以定義收集資料的頻率。例如，如果時段為 10，000 毫秒，Edge Agent 軟體會每 10 秒收集資料一次。

命令

命令會從雲端對車輛執行命令。您可以遠端將命令傳送至車輛，並在幾秒鐘內，車輛將執行命令。例如，您可以設定命令來鎖定車輛的門或設定溫度。

命令是由 管理的資源 AWS IoT Device Management。它包含可在傳送命令執行至車輛時套用的可重複使用組態。如需詳細資訊，請參閱《 AWS IoT Core 開發人員指南》中的AWS IoT 命令。

狀態範本

狀態範本為車主提供追蹤其車輛狀態的機制。在車輛上執行的 Edge Agent 軟體代理程式會收集並傳送訊號更新至雲端。每個狀態範本都包含從中收集資料的訊號清單。

AWS IoT FleetWise 的功能

以下是 AWS IoT FleetWise 的主要功能。

車輛建模

建置您車輛的虛擬呈現，並套用通用格式來組織車輛訊號。 AWS IoT FleetWise 支援 車輛訊號規格 (VSS)，您可以用來標準化車輛訊號。

結構描述型資料收集

定義僅將高價值車輛資料傳輸至雲端的方案。您可以定義條件型方案來控制要收集的資料，例如大於 40 度的資料車載溫度值。您也可以定義以時間為基礎的方案，以控制收集資料的頻率。

Edge Agent for AWS IoT FleetWise 軟體

在 車輛中執行的 Edge Agent 軟體可促進車輛與雲端之間的通訊。當車輛連接到雲端時，Edge Agent 軟體會持續接收資料收集方案，並相應地收集資料。

AWSAWS IoT FleetWise 中的區域和功能可用性

如需支援 AWS IoT FleetWise AWS 的區域清單，請參閱 AWS IoT FleetWise 端點和配額。 AWS IoT FleetWise 功能在其區域支援方面有所不同。

> **Note**
>
> 亞太區域 （孟買） 區域和部分 AWS IoT FleetWise 功能的存取目前受到鎖定。若要請求存取此 AWS 區域和所有門控功能，請聯絡您的客戶經理或 AWS 支援中心。

下表顯示依區域列出的功能支援：

功能/區域	美國東部 (維吉尼亞北部)	歐洲 (法蘭克福)	亞太區域（孟買）注意：僅限 Gated 存取
訊號目錄	是	是	門控
車輛模型	是	是	門控
解碼器資訊清單	是	是	門控
車輛	是	是	門控
機群	是	是	門控
行銷活動	是	是	門控
視覺系統資料（預覽版本中）	是	是	門控
MQTT 主題做為行銷活動資料目的地	門控	門控	門控
存放和轉送	門控	門控	門控
命令	門控	門控	門控
上次已知狀態	門控	門控	門控
使用自訂解碼界面的網路無關資料收集	門控	門控	門控
診斷故障碼 (DTC) 擷取*	門控	門控	門控

*DTC 擷取提供超出基本 DTC 資料擷取範圍的功能。此功能包含自訂功能，可讓您在邊緣定義函數，並在條件式行銷活動表達式中依名稱叫用函數。此外，它支援收集無限制字串，提供靈活的字串資料類型處理。Edge Agent 可以定期擷取資料，或由特定條件觸發，從而增強資料收集程序的適應性和效率。如需詳細資訊，請參閱 Edge Agent 開發人員指南中的自訂函數指南和 DTC 資料收集參考實作。

設定 AWS IoT FleetWise

首次使用 AWS IoT FleetWise 之前，請完成下列各節中的步驟。

主題

- 設定您的 AWS 帳戶
- 在主控台中開始使用
- 設定您的 AWS IoT FleetWise 設定
- 使用 IPv6 向 AWS IoT FleetWise 提出請求

設定您的 AWS 帳戶

完成下列任務以註冊 AWS 並建立管理使用者。

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

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

   註冊程序的一部分包括接聽電話或文字訊息，並在電話鍵盤上輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 在註冊程序完成後， 會傳送確認電子郵件給您。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。

建立具有管理存取權的使用者

註冊 後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center和建立 管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。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)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者（主控台）啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

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 Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 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 Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》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 Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

> **Note**
>
> 您可以使用服務連結角色搭配 AWS IoT FleetWise。服務連結角色由 AWS IoT FleetWise 預先定義，並包含 AWS IoT FleetWise 將指標傳送至 Amazon CloudWatch 所需的許可。如需詳細資訊，請參閱使用 AWS IoT FleetWise 的服務連結角色。

在主控台中開始使用

如果您尚未登入 AWS 帳戶，請登入，然後開啟 AWS IoT FleetWise 主控台。若要開始使用 AWS IoT FleetWise，請建立 車輛模型。車輛模型會標準化您車輛的格式。

1. 開啟 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詳細資訊，請參閱 建立 An AWS IoT FleetWise 車輛模型。

設定您的 AWS IoT FleetWise 設定

您可以使用 AWS IoT FleetWise 主控台或 API 來設定 Amazon CloudWatch Logs 指標、Amazon CloudWatch Logs 的設定，並使用 加密資料 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的服務限制。如需 CloudWatch 指標的詳細資訊，請參閱使用 Amazon CloudWatch 監控 AWS IoT FleetWise Amazon CloudWatch。

透過 CloudWatch Logs， AWS IoT FleetWise 會將日誌資料傳送至 CloudWatch 日誌群組，您可以在其中使用它來識別和緩解任何問題。如需 CloudWatch Logs 的詳細資訊，請參閱Configure AWS IoT FleetWise 記錄。

使用資料加密時， AWS IoT FleetWise 會使用 AWS 受管金鑰 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 AWS IoT FleetWise 中的資料加密。

進行設定 (主控台)

如果您尚未登入 AWS 帳戶，請登入，然後開啟 AWS IoT FleetWise 主控台。

1. 開啟 AWS IoT FleetWise 主控台。

2. 在左側窗格中，選擇設定。
3. 在指標中，選擇啟用。 AWS IoT FleetWise 會自動將 CloudWatch 受管政策連接至服務連結角色，並啟用 CloudWatch 指標。
4. 在記錄中，選擇編輯。
   a. 在 CloudWatch 記錄區段中，輸入日誌群組。
   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AWS IoT FleetWise 中的金鑰管理。
      i. 使用 AWS 金鑰 – AWS IoT FleetWise 擁有和管理金鑰。
      ii. 選擇不同的 AWS Key Management Service 金鑰 – 您可以管理您帳戶中 AWS KMS keys 的 。
   b. 若要儲存變更，請選擇提交。

設定 設定 (AWS CLI)

在 中 AWS CLI，註冊帳戶以設定 設定。

1. 若要設定設定，請執行下列命令。

```
aws iotfleetwise register-account
```

2. 若要驗證您的設定，請執行下列命令來擷取註冊狀態。

> **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，請參閱使用 AWS IoT FleetWise 的服務連結角色。

```
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/AWSIoTFleetwiseServiceRole"
    },
    "lastModificationTime": "2022-07-28T11:31:22.854000-07:00",
    }
}
```

註冊狀態可以是下列其中一項：

- `REGISTRATION_SUCCESS` – AWS 資源已成功註冊。
- `REGISTRATION_PENDING` – AWS IoT FleetWise 正在處理註冊請求。此程序大約需要五分鐘才能完成。
- `REGISTRATION_FAILURE` – AWS IoT FleetWise 無法註冊 AWS 資源。請稍後再試。

使用 IPv6 向 AWS IoT FleetWise 提出請求

您可以透過網際網路通訊協定第 6 版 (IPv6) 和 IPv4 與 AWS IoT FleetWise 通訊，以管理您的 資源。雙堆疊端點支援透過 IPv6 和 IPv4 向 AWS IoT FleetWise APIs提出請求。透過 IPv6 進行通訊無需額外費用。

IPv6 通訊協定是具有其他安全功能的新一代 IP 標準。它提供 128 位元長地址空間，而 IPv4 有 32 位元長地址。IPv4 可以產生 4.29 x 10^9 地址，而 IPv6 可以有 3.4 x 10^38 地址。

控制平面端點的 IPv6 先決條件

IPv6 通訊協定支援會自動為控制平面端點啟用。將端點用於控制平面用戶端時，您必須提供伺服器名稱指示 (SNI) 延伸。用戶端可以使用 SNI 延伸來指出正在聯絡的伺服器名稱，以及使用一般端點或雙堆疊端點。請參閱 使用雙堆疊端點。

AWS PrivateLink 端點的 IPv6 支援

AWS IoT FleetWise 支援使用 與介面 VPC 端點進行 IPv6 通訊 AWS PrivateLink。

測試 IPv6 地址相容性

如果您使用 Linux/Unix 或 Mac OS X，您可以使用 curl 命令測試是否可以透過 IPv6 存取雙堆疊端點，如下列範例所示：

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

您會收到類似下例的資訊。如果您透過 IPv6 連接，連接的 IP 地址將是 IPv6 地址。

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

如果您使用的是 Microsoft Windows 7 或 Windows 10，您可以使用 ping 命令測試是否可以透過 IPv6 或 IPv4 存取雙堆疊端點，如下列範例所示。

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

在 IAM 原則中使用 IPv6 地址

將 IPv6 用於資源之前，您必須確保用於 IP 地址篩選的任何 IAM 政策都包含 IPv6 地址範圍。如需 IAM 管理存取許可的詳細資訊，請參閱 Identity and Access Management for AWS IoT FleetWise。

篩選 IP 地址的 IAM 原則使用IP 地址條件運算子。下列政策會使用 IP 地址條件運算子來識別允許的 IPv4 地址`54.240.143.*`範圍。由於所有 IPv6 地址都超出允許的範圍，因此此政策會防止使用 IPv6 地址進行通訊。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "iotfleetwise:*",
      "Resource": "arn:aws:iotfleetwise:*",
```

```
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

若要包含 IPv6 地址，您可以修改政策的條件元素，以允許 IPv4 (54.240.143.0/24) 和 IPv6 (2001：DB8：1234：5678：：/64) 地址範圍，如下列範例所示。

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

使用雙堆疊端點

AWS IoT FleetWise 雙堆疊端點支援透過 IPv6 和 IPv4 向 AWS IoT FleetWise APIs提出請求。當您向雙堆疊端點提出請求時，它會自動解析為 IPv4 或 IPv6 地址。在雙堆疊模式中，接受 IPv4 和 IPv6 用戶端連線。

如果您使用 REST API，則可以使用端點名稱 (URI) 直接存取 AWS IoT FleetWise 端點。 AWS IoT FleetWise 僅支援區域雙堆疊端點名稱，這表示您必須指定 AWS 區域 做為名稱的一部分。

下表顯示使用 IPv4 和雙堆疊模式時 AWS IoT FleetWise 的控制平面端點格式。如需這些端點的詳細資訊，請參閱 AWS IoT FleetWise 端點。

端點	IPv4 地址	雙堆疊模式
控制平台	iotfleetwise.*<region>*.amazonaws.com	iotfleetwise.*<region>*.api.aws

使用 AWS CLI AWS SDKs時，您可以使用`AWS_USE_DUALSTACK_ENDPOINT`環境變數或`use_dualstack_endpoint` 參數，這是共用組態檔案設定，以變更為雙堆疊端點。您也可以直接將

雙堆疊端點指定為組態檔案中 AWS IoT FleetWise 端點的覆寫。如需詳細資訊，請參閱雙堆疊和 FIPS 端點。

使用 時 AWS CLI，您可以將組態值設定為 Config `use_dualstack_endpoint` `true` 檔案中的設定檔中的 AWS 。這會將命令提出的所有 AWS IoT FleetWise 請求導向至指定區域的雙堆疊端點。您可以在設定檔或命令中使用 `--region` 選項指定區域。

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

不是針對所有命令使用雙堆疊端點，而是針對特定命令使用這些端點：

- 您可以設定這些命令的 `--endpoint-url` 參數，以針對特定命令使用雙堆疊端點。例如，在下列命令中，您可以將 *<endpoint-url>* 取代為 `iotfleetwise.`*<region>*`.api.aws`。

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

- 您可以在 Config AWS 檔案中設定個別的設定檔。例如，建立一個`use_dualstack_endpoint`設定為 true 的設定檔，以及一個未設定 的設定檔`use_dualstack_endpoint`。當您執行命令時，必須依據是否要使用雙堆疊端點來指定所要使用的設定檔。

教學課程： AWS IoT FleetWise 入門

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來開始使用 AWS IoT FleetWise。

請參閱下列主題，進一步了解 AWS IoT FleetWise：

- Ingest AWS IoT FleetWise 資料至雲端
- Model AWS IoT FleetWise 車輛
- Manage AWS IoT FleetWise 車輛
- 在 AWS IoT FleetWise 中管理機群
- 使用行銷活動收集 AWS IoT FleetWise 資料

簡介

使用 AWS IoT FleetWise 以近乎即時的方式收集、轉換唯一資料格式，並從自動車輛傳輸到雲端。您可以存取整個機群的洞見。這可協助您有效地偵測和緩解車輛運作狀態的問題、傳輸高價值資料訊號，以及遠端診斷問題，同時降低成本。

本教學課程說明如何開始使用 AWS IoT FleetWise。您將了解如何建立 車輛模型 （模型資訊清單）、解碼器資訊清單、 車輛和行銷活動。

如需 AWS IoT FleetWise 關鍵元件和概念的詳細資訊，請參閱 AWS IoT FleetWise 的重要概念和功能。

預估時間：約 45 分鐘。

> **Important**
>
> 您將需要支付此示範建立和使用的 AWS IoT FleetWise 資源的費用。如需詳細資訊，請參閱 AWS IoT FleetWise AWS 定價頁面中的 IoT FleetWise。

先決條件

若要完成此入門教學課程，您首先需要下列項目：

- AWS 帳戶：如果您沒有 AWS 帳戶，請參閱 AWS 帳戶管理 參考指南中的建立 AWS 帳戶。

- 存取支援 AWS IoT FleetWise AWS 區域 的 。目前，美國東部（維吉尼亞北部）和歐洲（法蘭克福）支援 AWS IoT FleetWise。您可以使用 中的區域選擇器 AWS Management Console 來切換到其中一個區域。如需詳細資訊，請參閱 AWS IoT FleetWise 端點和配額。
- Amazon Timestream 資源：
  - Amazon Timestream 資料庫。如需詳細資訊，請參閱《Amazon Timestream 開發人員指南》中的建立資料庫。
  - 在 Amazon Timestream 中建立的 Amazon Timestream 資料表，可存放您的資料。如需詳細資訊，請參閱《Amazon Timestream 開發人員指南》中的建立資料表。
- Edge Agent 軟體示範。（設定示範的說明位於下一個步驟。）
  - 您可以使用 Explore Edge Agent 快速入門示範來探索 AWS IoT FleetWise，並了解如何開發 Edge Agent 軟體 for AWS IoT FleetWise。此示範使用 AWS CloudFormation 範本。它會引導您檢閱 Edge Agent 參考實作、開發 Edge Agent，然後在 Amazon EC2 Graviton 上部署 Edge Agent 軟體，以及產生範例車輛資料。示範也提供指令碼，可用來建立訊號目錄、車輛模型、解碼器資訊清單、車輛、機群和行銷活動，全都在雲端。
  - 若要下載示範，請導覽至 AWS IoT FleetWise 主控台。在服務首頁的 AWS IoT FleetWise 入門區段中，選擇探索邊緣代理程式。

步驟 1：設定 Edge Agent 軟體 for AWS IoT FleetWise

> **Note**
>
> 此步驟中的 CloudFormation 堆疊使用遙測資料。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 Vision System Data Developer Guide。視覺系統資料處於預覽版本，可能會有所變更。

您的 Edge Agent 軟體 for AWS IoT FleetWise 可促進車輛與雲端之間的通訊。它會從資料收集方案收到如何從雲端連線車輛收集資料的指示。

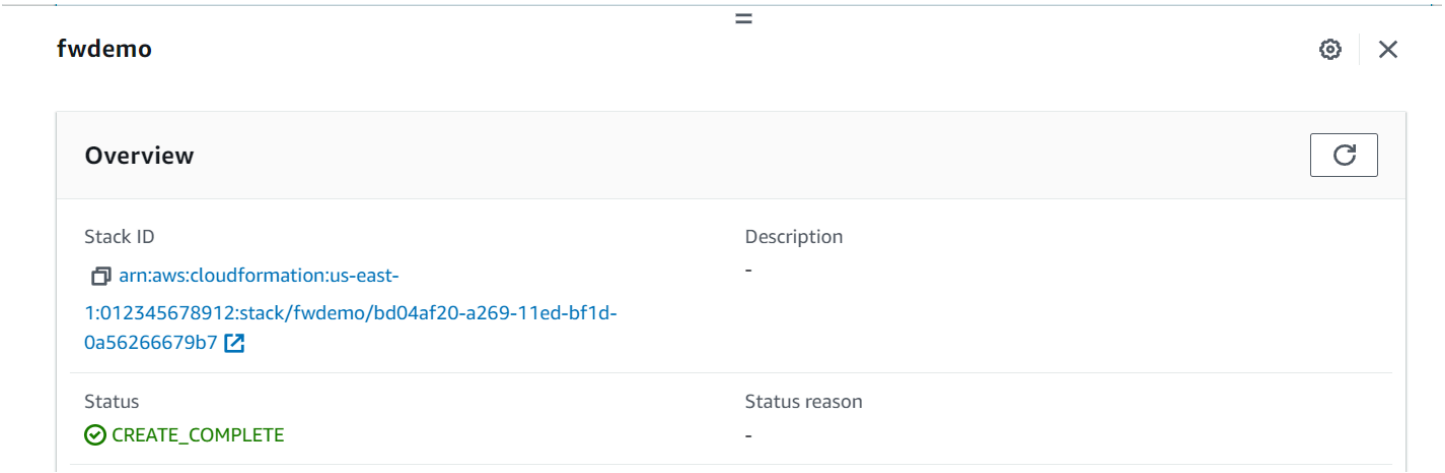
若要設定 Edge Agent 軟體，請在一般資訊中執行下列動作：

1. 開啟啟動 CloudFormation 範本。
2. 在快速建立堆疊頁面上，針對堆疊名稱輸入 AWS IoT FleetWise 資源堆疊的名稱。堆疊是一種易記的名稱，以字首形式出現在此 AWS CloudFormation 範本建立的資源名稱上。
3. 在參數下，輸入與堆疊相關的參數自訂值。

a. Fleetsize - 您可以更新 Fleetsize 參數來增加機群中的車輛數量。

b. IoTCoreRegion - 您可以更新 IoTCoreRegion 參數來指定 AWS IoT 物件的建立區域。您必須使用與用來建立 AWS IoT FleetWise 車輛相同的區域。如需詳細資訊 AWS 區域，請參閱區域和區域 - Amazon Elastic Compute Cloud。

4. 在功能區段中，選取方塊以確認 AWS CloudFormation 會建立 IAM 資源。
5. 選擇建立堆疊，然後等待約 15 分鐘讓堆疊的狀態顯示 CREATE_COMPLETE。
6. 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，並尋找 CREATE_COMPLETE。



> **Important**
> 您將需要支付此示範建立和使用的 AWS IoT FleetWise 資源的費用。如需詳細資訊，請參閱 AWS IoT FleetWise AWS 定價頁面中的 IoT FleetWise。

步驟 2：建立車輛模型

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

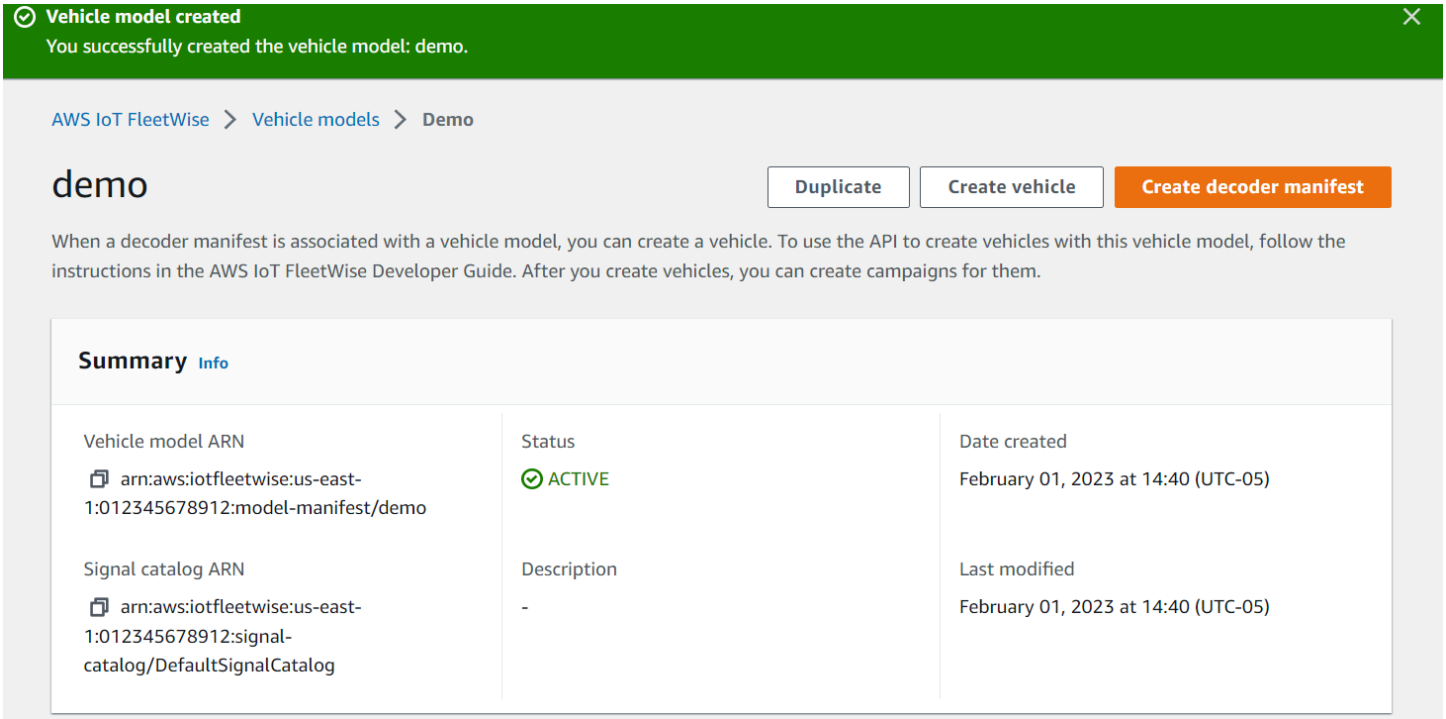
您可以使用 車輛模型來標準化車輛的格式，並協助定義您所建立車輛中訊號之間的關係。當您建立 車輛模型時，也會建立訊號目錄。訊號目錄是一組標準化訊號，可以重複使用來建立車輛模型。訊號是您定義為包含車輛資料及其中繼資料的基本結構。目前， AWS IoT FleetWise 服務 AWS 區域 每個帳戶僅支援一個訊號目錄。這有助於驗證從機群處理的資料是否一致。

建立車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 在車輛模型頁面上，選擇建立車輛模型。
4. 在一般資訊區段中，輸入 車輛模型的名稱，例如 Vehicle1 和選用的描述。然後選擇下一步。
5. 從訊號目錄中選擇一或多個訊號。您可以在搜尋目錄中依名稱篩選訊號，或從清單中選擇它們。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 Next (下一步)。
6. 選擇您的 .dbc 檔案，然後從本機裝置上傳。選擇 Next (下一步)。

   > **Note**
   >
   > 在本教學課程中，您可以下載範例 .dbc 檔案以上傳此步驟。

7. 將屬性新增至您的車輛模型，然後選擇下一步。
   a. 名稱 - 輸入車輛屬性的名稱，例如製造商名稱或製造日期。
   b. 資料類型 - 在資料類型功能表中，選擇資料類型。
   c. 單位 - （選用） 輸入單位值，例如公里或攝氏。
   d. 路徑 - （選用） 輸入訊號路徑的名稱，例如點 `Vehicle.Engine.Light.` (.) 表示它是子訊號。
   e. 預設值 - （選用） 輸入預設值。
   f. 描述 - （選用） 輸入屬性的描述。
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，告知您的車輛模型已成功建立。



步驟 3：建立解碼器資訊清單

解碼器資訊清單與您建立的車輛模型相關聯。其中包含的資訊可協助 AWS IoT FleetWise 解碼車輛資料，並將車輛資料從二進位格式轉換為可供分析的人類可讀值。網路介面和解碼器訊號是有助於設定解碼器資訊清單的元件。網路介面包含您車輛網路使用的 CAN 或 OBD 通訊協定相關資訊。解碼器訊號提供特定訊號的解碼資訊。

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 車輛模型。
3. 在 車輛模型區段中，選擇您要用來建立解碼器資訊清單的車輛模型。
4. 選擇建立解碼器資訊清單。

步驟 4：設定解碼器資訊清單

設定解碼器資訊清單

> **Important**
> 您不能使用 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。反之，請使用 AWS CLI。如需詳細資訊，請參閱建立解碼器資訊清單 (AWS CLI)。

1. 為了協助您識別解碼器資訊清單，請輸入其名稱和選用描述。然後選擇下一步。
2. 若要新增一或多個網路介面，請選擇 CAN_INTERFACE 或 OBD_INTERFACE 類型。
   - 車載診斷 (OBD) 界面 - 如果您想要定義電子控制元件 (ECUs) 之間如何通訊自我診斷資料的通訊協定，請選擇此界面類型。此通訊協定提供許多標準診斷問題代碼 (DTCs)，可協助您對車輛的問題進行疑難排解。
   - Controller Area Network (CAN 匯流排） 界面 - 如果您想要定義 ECUs 之間資料通訊方式的通訊協定，請選擇此界面類型。ECUs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. 針對您在上一個步驟中新增的訊號，選擇解碼器訊號。若要提供解碼資訊，請上傳 .dbc 檔案。車輛模型中的每個訊號都必須與解碼器訊號配對，您可以從清單中選擇。
6. 若要新增另一個網路介面，請選擇新增網路介面。新增網路介面完成後，請選擇下一步。
7. 檢閱您的組態，然後選擇建立。系統會顯示通知，指出您的解碼器資訊清單已成功建立。

步驟 5：建立 車輛

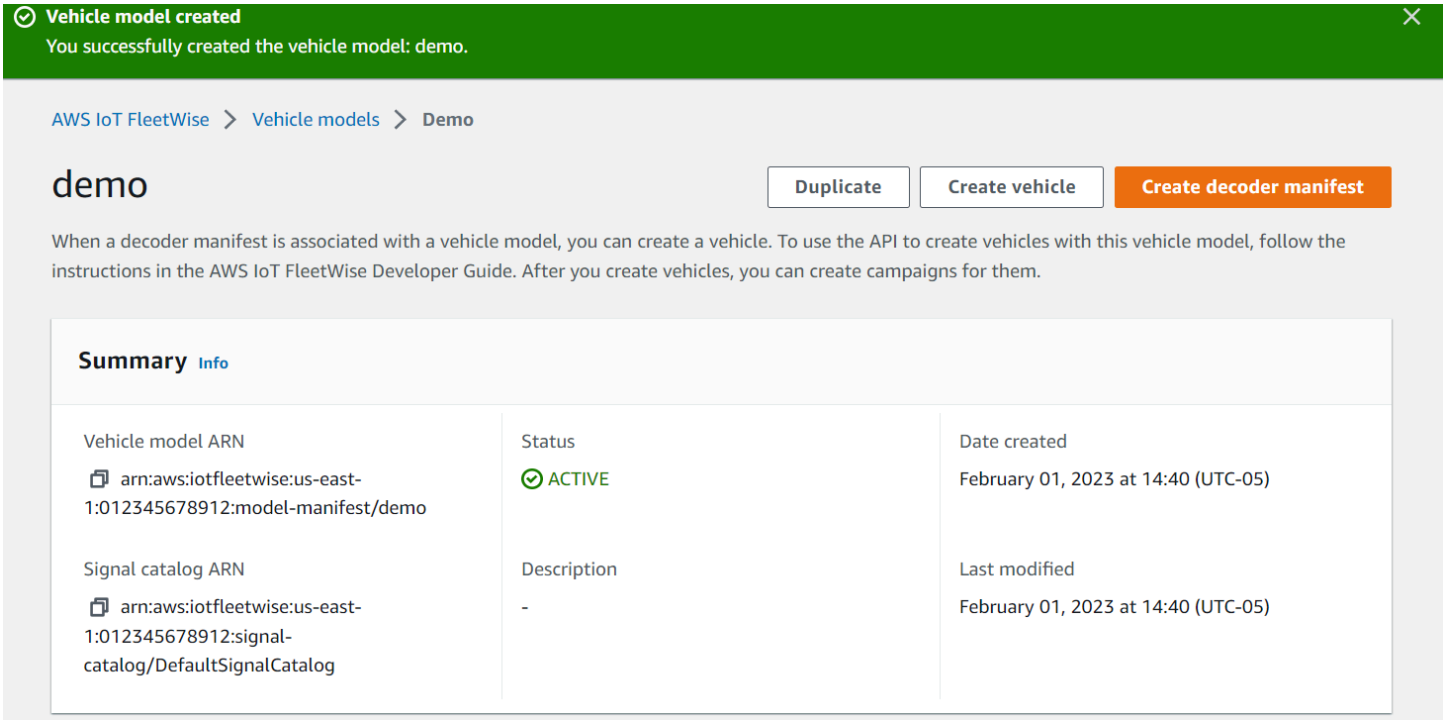
在 AWS IoT FleetWise 中， 車輛是真實物理車輛的虛擬表示法。從相同車輛模型建立的所有車輛都會繼承相同的訊號群組，而您建立的每個車輛都會對應至新建立的 IoT 物件。您必須將所有車輛與解碼器資訊清單建立關聯。

先決條件

1. 確認您已建立車輛模型和解碼器資訊清單。此外，請確認車輛模型的狀態為 ACTIVE。
   a. 若要確認車輛模型的狀態為 ACTIVE，請開啟 AWS IoT FleetWise 主控台。

b. 在導覽窗格中，選擇 車輛模型。

c. 在摘要區段的狀態下，檢查車輛的狀態。



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. 若要定義車輛屬性，請輸入車輛名稱，然後選擇模型資訊清單（車輛模型）和解碼器資訊清單。
5. （選用）若要定義車輛屬性，請輸入鍵值對，然後選擇新增屬性。
6. （選用）若要標記您的 AWS 資源，請新增標籤，然後選擇新增標籤。
7. 選擇 Next (下一步)。
8. 若要設定車輛憑證，您可以上傳自己的憑證，或選擇自動產生新憑證。 我們建議您自動產生憑證，以便更快速地設定。如果您已經有憑證，您可以選擇改用它。
9. 下載公有和私有金鑰檔案，然後選擇下一步。
10. 若要將政策連接至車輛憑證，您可以輸入現有的政策名稱或建立新的政策。若要建立新的政策，請選擇建立政策，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：建立行銷活動

在 AWS IoT FleetWise 中，行銷活動用於協助選擇、收集資料，以及將資料從車輛傳輸到雲端。行銷活動包含資料收集方案，提供 Edge Agent 軟體如何使用條件型收集方案或時間型收集方案收集資料的指示。

建立行銷活動

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇行銷活動。
3. 選擇 Create campaign (建立活動)。
4. 輸入您的行銷活動名稱和選用描述。
5. 若要設定行銷活動的資料收集方案，您可以手動定義資料收集方案，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機制。

   a. 若要手動定義資料收集方案，請選擇定義資料收集方案，然後選擇您想要用於行銷活動的資料收集方案類型。您可以選擇條件型收集方案或時間型收集方案。
   b. 如果您選擇以時間為基礎的收集機制，您必須指定行銷活動收集車輛資料的持續時間。
   c. 如果您選擇條件型收集機制，則必須指定表達式來識別要收集的資料。請務必將訊號的名稱指定為變數、比較運算子和比較值。
   d. （選用）選擇表達式的語言版本，或將其保留為預設值 1。
   e. （選用）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。根據預設，Edge Agent for AWS IoT FleetWise 軟體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，在第一次觸發時收集資料。
   g. （選用）您可以選擇更進階的方案選項。
6. 若要指定資料收集方案將從中收集資料的訊號，請從選單搜尋訊號的名稱。
7. （選用）您可以選擇最大取樣計數或最小取樣間隔。您也可以新增更多訊號。
8. 選擇 Next (下一步)。
9. 定義您希望行銷活動傳輸資料的儲存目的地。您可以在 Amazon S3 或 Amazon Timestream 中存放資料。

   a. Amazon S3 – 選擇 AWS IoT FleetWise 具有 許可的 S3 儲存貯體。
   b. Amazon Timestream – 選擇 Timestream 資料庫和資料表名稱。輸入允許 將資料傳送至 Timestream AWS IoT FleetWise 的 IAM 角色。

10. 選擇 Next (下一步)。
11. 從搜尋方塊中選擇車輛屬性或車輛名稱。
12. 輸入與您選擇的車輛屬性或名稱相關的值。
13. 選擇您的行銷活動將從中收集資料的車輛。然後選擇下一步。
14. 檢閱行銷活動的組態，然後選擇建立行銷活動。您或您的團隊必須將行銷活動部署到 車輛。

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 Stacks 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. 您可以處理和視覺化行銷活動收集的車輛資料。如需詳細資訊，請參閱Visualize AWS IoT FleetWise 車輛資料。
2. 您可以對 AWS IoT FleetWise 進行故障診斷並解決問題。如需詳細資訊，請參閱故障診斷 AWS IoT FleetWise。

Ingest AWS IoT FleetWise 資料至雲端

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

Edge Agent for AWS IoT FleetWise 軟體在車輛中安裝和執行時，旨在促進車輛與雲端之間的安全通訊。

> **Note**
>
> - AWS IoT FleetWise 不適用於或與任何危險環境或關鍵系統的操作相關，這些環境或關鍵系統可能會導致嚴重的身體傷害或死亡，或造成環境或財產損壞。透過您使用 AWS IoT FleetWise 收集的車輛資料僅供參考，您可能無法使用 AWS IoT FleetWise 來控制或操作車輛功能。
> - 應針對您的使用案例評估透過您使用 AWS IoT FleetWise 收集的車輛資料的準確性，包括為了滿足您根據適用車輛安全法規（例如安全監控和報告義務）可能擁有的任何合規義務。此類評估應包括透過其他產業標準方法和來源（例如車輛駕駛的報告）收集和檢閱資訊。

若要將資料擷取至雲端，請執行下列動作：

1. 在您的車輛中開發並安裝 Edge Agent for AWS IoT FleetWise 軟體。如需如何使用 Edge Agent 軟體的詳細資訊，請執行下列動作來下載 Edge Agent for AWS IoT 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 AWS IoT FleetWise 入門區段中，選擇探索邊緣代理程式。
2. 建立或匯入訊號目錄，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 AWS IoT FleetWise 訊號目錄 和 匯入訊號目錄 (AWS CLI)。

   > **Note**
   >
   > - 如果您使用 AWS IoT FleetWise 主控台來建立第一個車輛模型，則不需要手動建立訊號目錄。當您建立第一個車輛模型時， AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。

- AWS IoT FleetWise 目前支援每個 AWS 帳戶的訊號目錄 AWS 區域。

3. 使用訊號目錄中的訊號來建立車輛模型。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。

> **Note**
>
> - 如果您使用 AWS IoT FleetWise 主控台建立車輛模型，則可以上傳 .dbc 檔案以匯入訊號。.dbc 是 Controller Area Network (CAN 匯流排） 資料庫支援的檔案格式。建立車輛模型後，新的訊號會自動新增至訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。
> - 如果您使用 `CreateModelManifest` API 操作來建立車輛模型，則必須使用 `UpdateModelManifest` API 操作來啟用車輛模型。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 車輛模型。
> - 如果您使用 AWS IoT FleetWise 主控台建立 車輛模型， AWS IoT FleetWise 會自動為您啟用 車輛模型。

4. 建立解碼器資訊清單。解碼器資訊清單包含您在上一個步驟中建立的車輛模型中指定之每個訊號的解碼資訊。解碼器資訊清單會與您建立的車輛模型相關聯。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

> **Note**
>
> - 如果您使用 `CreateDecoderManifest` API 操作來建立解碼器資訊清單，則必須使用 `UpdateDecoderManifest` API 操作來啟用解碼器資訊清單。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 解碼器資訊清單。
> - 如果您使用 AWS IoT FleetWise 主控台建立解碼器資訊清單， AWS IoT FleetWise 會自動為您啟用解碼器資訊清單。

5. 從 車輛模型建立 車輛。從相同車輛模型建立的車輛會繼承相同的訊號群組。您必須使用 AWS IoT Core 佈建車輛，才能將資料擷取至雲端。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛。
6. （選用） 建立機群以代表一組車輛，然後將個別車輛與機群建立關聯。這可協助您同時管理多部車輛。如需詳細資訊，請參閱在 AWS IoT FleetWise 中管理機群。

7. （選用）建立行銷活動。行銷活動會部署到車輛或機群。行銷活動會提供 Edge Agent 軟體指示，說明如何選取、收集資料，以及將資料傳輸至雲端。如需詳細資訊，請參閱使用行銷活動收集 AWS IoT FleetWise 資料。您可以建立行銷活動、狀態範本（下方）或兩者來收集資料。

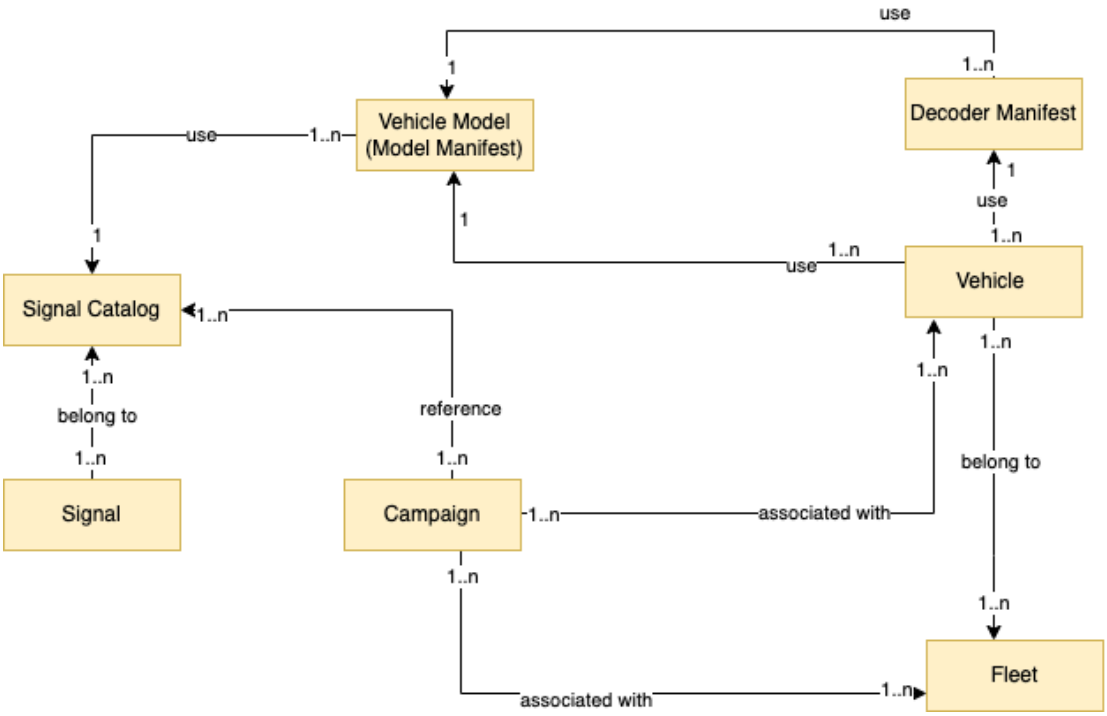
> **Note**
>
> 您必須使用 `UpdateCampaign` API 操作來核准行銷活動， AWS IoT FleetWise 才能將其部署到車輛或機群。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 行銷活動。

8. （選用）建立狀態範本。狀態範本會部署到 車輛。狀態範本為車主提供追蹤其車輛狀態的機制。如需詳細資訊，請參閱監控您車輛的最後已知狀態。

Edge Agent 軟體 AWS IoT Core 會使用您選擇的 MQTT 主題，將車輛資料傳輸至 。若要將資料傳送至行銷活動的 AWS IoT FleetWise，會使用預留主題 `$aws/iotfleetwise/vehicles/vehicleName/signals`。針對上次已知狀態，Edge Agent 會使用預留主題 `$aws/iotfleetwise/vehicles/vehicleName/last_known_states/data`。如需如何處理擷取資料的詳細資訊，請參閱Visualize AWS IoT FleetWise 車輛資料。

Model AWS IoT FleetWise 車輛

AWS IoT FleetWise 提供車輛建模架構，可讓您在雲端中建置車輛的虛擬呈現。訊號、訊號目錄、車輛模型和解碼器資訊清單是您用來建立車輛模型的核心元件。



Signal

訊號是您定義包含車輛資料及其中繼資料的基本結構。訊號可以是屬性、分支、感應器或傳動器。例如，您可以建立感應器來接收車載溫度值，以及存放其中繼資料，包括感應器名稱、資料類型和單位。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄。

訊號目錄

訊號目錄包含一組訊號。訊號目錄中的訊號可用來建立使用不同通訊協定和資料格式的車輛模型。例如，不同汽車製造商製造的兩部車：一部使用控制區域網路 (CAN 匯流排） 通訊協定；另一部使用車載診斷 (OBD) 通訊協定。您可以在訊號目錄中定義感應器，以接收車載溫度值。此感應器可用來代表兩部車中的熱電偶。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄。

車輛模型 （模型資訊清單）

車輛模型是宣告式結構，可用來標準化車輛的格式，並定義車輛訊號之間的關係。車輛模型會在相同類型的多部車輛之間強制執行一致的資訊。您可以新增訊號來建立車輛模型。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛模型。

解碼器資訊清單

解碼器資訊清單包含車輛模型中每個訊號的解碼資訊。車輛中的感應器和傳動器會傳輸低階訊息（二進位資料）。使用解碼器資訊清單， AWS IoT FleetWise 能夠將二進位資料轉換為人類可讀取的值。每個解碼器資訊清單都與車輛模型相關聯。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

您可以使用 AWS IoT FleetWise 主控台或 API，以下列方式建立車輛模型。

1. 建立或匯入訊號目錄，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 AWS IoT FleetWise 訊號目錄及匯入訊號目錄 (AWS CLI)。

   > **Note**
   >
   > - 如果您使用 AWS IoT FleetWise 主控台來建立第一個車輛模型，則不需要手動建立訊號目錄。當您建立第一個車輛模型時， AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。
   > - AWS IoT FleetWise 目前支援每個 AWS 帳戶的訊號目錄 AWS 區域。

2. 在訊號目錄中使用訊號來建立車輛模型。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。

   > **Note**
   >
   > - 如果您使用 AWS IoT FleetWise 主控台建立車輛模型，則可以上傳 .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，新的訊號會自動新增至訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。
   > - 如果您使用 `CreateModelManifest` API 操作來建立車輛模型，則必須使用 `UpdateModelManifest` API 操作來啟用車輛模型。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 車輛模型。
   > - 如果您使用 AWS IoT FleetWise 主控台建立車輛模型， AWS IoT FleetWise 會自動為您啟用車輛模型。

3. 建立解碼器資訊清單。解碼器資訊清單包含您在上一個步驟中建立之車輛模型中指定之每個訊號的解碼資訊。解碼器資訊清單與您建立的車輛模型相關聯。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

> **Note**
>
> - 如果您使用 `CreateDecoderManifest` API 操作來建立解碼器資訊清單，則必須使用 `UpdateDecoderManifest` API 操作來啟用解碼器資訊清單。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 解碼器資訊清單。
> - 如果您使用 AWS IoT FleetWise 主控台建立解碼器資訊清單， AWS IoT FleetWise 會自動為您啟用解碼器資訊清單。

CAN 匯流排資料庫支援 .dbc 檔案格式。您可以上傳 .dbc 檔案來匯入訊號和訊號解碼器。若要取得範例 .dbc 檔案，請執行下列動作。

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`。

主題

- Manage AWS IoT FleetWise 訊號目錄
- Manage AWS IoT FleetWise 車輛模型
- Manage AWS IoT FleetWise 解碼器資訊清單

Manage AWS IoT FleetWise 訊號目錄

> **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。

訊號目錄是標準化訊號的集合，可以重複使用這些訊號來建立車輛模型。 AWS IoT FleetWise 支援 車輛訊號規格 (VSS)，您可以遵循這些訊號來定義訊號。訊號可以是下列任何類型。

屬性

屬性代表通常不會變更的靜態資訊，例如製造商和製造日期。

分支

分支代表巢狀結構中的訊號。分支示範訊號階層。例如，`Vehicle`分支具有子分支 `Powertrain`。`Powertrain` 分支具有子分支 `combustionEngine`。若要尋找`combustionEngine`分支，請使用 `Vehicle.Powertrain.combustionEngine`運算式。

感測器

感應器資料會報告車輛的目前狀態，並隨著時間而變更，因為車輛狀態會變更，例如油位、溫度、振動或電壓。

傳動器

傳動器資料會報告車輛裝置的狀態，例如馬達、暖器和門鎖。變更車輛裝置的狀態可以更新傳動器資料。例如，您可以定義一個傳動器來代表暖器。當您開啟或關閉暖機時，傳動器會收到新資料。

自訂結構

自訂結構（也稱為結構）代表複雜或更高順序的資料結構。它有助於邏輯繫結或分組來自相同來源的資料。以原子操作讀取或寫入資料時，會使用結構，例如代表複雜資料類型或更高順序的形狀。

結構類型的訊號是使用結構資料類型的參考，而非基本資料類型在訊號目錄中定義。結構可用於所有類型的訊號，包括感應器、屬性、傳動器和視覺系統資料類型。如果傳送或接收結構類型的訊號， 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，如果結構包含項目 Vehicle.Camera.Image.height、Carve.Camera.Image.width 和 Vehicle.Camera.Image.data – 預期傳送的訊號包含所有這些項目的值。

> **Note**
>
> 視覺系統資料處於預覽版本，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基本或另一個結構。

使用結構和自訂屬性來表示高階形狀時，預期的高階形狀一律會定義為樹狀結構，並視作樹狀結構。自訂屬性用於定義所有分葉節點，而結構用於定義所有非分葉節點。

> **Note**
>
> - 如果您使用 AWS IoT FleetWise 主控台來建立第一個車輛模型，則不需要手動建立訊號目錄。當您建立第一個車輛模型時， AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。
> - 如果您使用 AWS IoT FleetWise 主控台建立車輛模型，則可以上傳 .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，新的訊號會自動新增至訊號目錄。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。
> - 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 操作，您可以用來建立和管理訊號目錄。

- CreateSignalCatalog – 建立新的訊號目錄。
- ImportSignalCatalog – 透過上傳 .json 檔案匯入訊號以建立訊號目錄。訊號必須依照 VSS 定義，並以 JSON 格式儲存。
- UpdateSignalCatalog – 透過更新、移除或新增訊號來更新現有的訊號目錄。
- DeleteSignalCatalog – 刪除現有的訊號目錄。
- 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） 的分頁清單。
- GetSignalCatalog – 擷取訊號目錄的相關資訊。

教學課程

- Configure AWS IoT FleetWise 訊號
- 建立 AWS IoT FleetWise 訊號目錄
- 匯入 an AWS IoT FleetWise 訊號目錄
- 更新 an AWS IoT FleetWise 訊號目錄
- 刪除 AWS IoT FleetWise 訊號目錄
- Get AWS IoT FleetWise 訊號目錄資訊

Configure AWS IoT FleetWise 訊號

本節說明如何設定分支、屬性、感應器和傳動器。

主題

- 設定分支
- 設定屬性
- 設定感應器或傳動器
- 設定複雜的資料類型

設定分支

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 dot(.) 來參考子分支。例如， `Vehicle.Chassis.SteeringWheel`是`SteeringWheel`分支的完整名稱。`Vehicle.Chassis.`是此分支的路徑。

  完整名稱最多可有 150 個字元。有效字元：a–z、A–Z、0–9、冒號 (：) 和底線 (_)。
- （選用） `Description` – 分支的描述。

  描述最多可有 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棄用訊息。

  deprecationMessage 最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號）、_（底線） 和 -（連字號）。
- （選用） `comment` – 描述以外的註解。註解可用來提供有關分支的其他資訊，例如分支的原理或相關分支的參考。

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

設定屬性

若要設定屬性，請指定下列資訊。

- `dataType` – 屬性的資料類型必須是下列其中一項：INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64,
  UINT64、BOOLEAN、FLOAT、DOUBLE、STRRING、UNIX_TIMESTAMP、INT8_ARRAY、UINT8_ARR
  自訂分支中定義的資料類型。
- `fullyQualifiedName` – 屬性的完整名稱是屬性加上屬性名稱的路徑。使用 dot(.) 來參考子訊號。例如， `Diameter` `Vehicle.Chassis.SteeringWheel.Diameter`是 屬性的完整名稱。`Vehicle.Chassis.SteeringWheel.`是此屬性的路徑。

  完整名稱最多可有 150 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號） 和 _ （底線）。
- （選用） `Description` – 屬性的描述。

  描述最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號）、_ （底線） 和 - （連字號）。
- （選用） `unit` – 屬性的科學單位，例如 km 或 Celsius。
- （選用） `min` – 屬性的最小值。
- （選用） `max` – 屬性的最大值。
- （選用） `defaultValue` – 屬性的預設值。
- （選用） `assignedValue` – 指派給屬性的值。
- （選用） `allowedValues` – 屬性接受的值清單。
- （選用） `deprecationMessage` – 正在移動或刪除的節點或分支的棄用訊息。

  deprecationMessage 最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號）、_ （底線） 和 - （連字號）。
- （選用） `comment` – 描述以外的註解。註解可用來提供屬性的其他資訊，例如屬性的原理或相關屬性的參考。

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

設定感應器或傳動器

若要設定感應器或傳動器，請指定下列資訊。

- `dataType` – 訊號的資料類型必須是下列其中一項：INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64,

UINT64、BOOLEAN、FLOAT、DOUBLE、STRRING、UNIX_TIMESTAMP、INT8_ARRAY、UTINT8_AR
或自訂類型的分支中定義的資料類型。

- `fullyQualifiedName` – 訊號的完整名稱是訊號的路徑加上訊號的名稱。使用 dot(.) 來參考子訊號。例如，`Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`是`HandsOffSteeringS`動器的完整名稱。 `Vehicle.Chassis.SteeringWheel.HandsOff.`是此傳動器的路徑。

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

- （選用） `Description` – 訊號的描述。

  描述最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 : （冒號）、_（底線） 和 -（連字號）。

- （選用） `unit` – 訊號的科學單位，例如 km 或 Celsius。
- （選用） `min` – 訊號的最小值。
- （選用） `max` – 訊號的最大值。
- （選用） `assignedValue` – 指派給訊號的值。
- （選用） `allowedValues` – 訊號接受的值清單。
- （選用） `deprecationMessage` – 正在移動或刪除的節點或分支的棄用訊息。

  deprecationMessage 最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 : （冒號）、_（底線） 和 -（連字號）。

- （選用） `comment` – 描述以外的註解。註解可用來提供有關感應器或傳動器的其他資訊，例如其原理或相關感應器或傳動器的參考。

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

設定複雜的資料類型

建立視覺系統的模型時，會使用複雜的資料類型。除了分支之外，這些資料類型是由結構（也稱為結構） 和屬性組成。結構是由多個值描述的訊號，例如影像。屬性代表結構的成員，例如基本資料類型（例如 UINT8) 或其他結構（例如時間戳記）。例如，Carve.Cameras.Front 代表分支、Carve.Cameras.Front.Image 代表結構，而 Vehicle.Cameras.Timestamp 代表屬性。

下列複雜資料類型範例示範 訊號和資料類型如何匯出至單一 .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。
> 視覺系統資料處於預覽版本，可能會有所變更。

設定結構

若要設定自訂結構（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`。

  完整名稱最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）。
- （選用）`Description` – 訊號的描述。

  描述最多可有 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` – 正在移動或刪除的節點或分支的棄用訊息。

  deprecationMessage 最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、：（冒號）、_（底線）和 -（連字號）。
- （選用）`comment` – 描述以外的註解。註解可用來提供有關感應器或傳動器的其他資訊，例如其原理或相關感應器或傳動器的參考。

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

設定 屬性

若要設定自訂屬性，請指定下列資訊。

- `dataType` – 訊號的資料類型必須是下列其中一項：INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64,
  UINT64、BOOLEAN、FLOAT、DOUBLE、STRRING、UNIX_TIMESTAMP、INT8_ARRAY、UINT8_ARR
  或 UNKOWNOWN。
- `fullyQualifiedName` – 自訂屬性的完整名稱。例如，自訂屬性的完整名稱可能是 `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`。

  完整名稱最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）
- （選用）`Description` – 訊號的描述。

  描述最多可有 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` – 正在移動或刪除的節點或分支的棄用訊息。

  deprecationMessage 最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號）、_（底線） 和 -（連字號）。
- （選用） `comment` – 描述以外的註解。註解可用來提供有關感應器或傳動器的其他資訊，例如其原理或相關感應器或傳動器的參考。

  註解最多可有 2048 個字元。有效字元：a–z、A–Z、0–9、 ： （冒號）、_（底線） 和 -（連字號）。
- （選用） `dataEncoding` – 指出 屬性是否為二進位資料。自訂屬性的資料編碼必須是下列其中一項：BINARY 或 TYPED。
- （選用） `structFullyQualifiedName` – 如果自訂屬性的資料類型為 Struct 或 StructArray，則為自訂屬性的結構（結構） 節點的完整名稱。

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

建立 AWS IoT FleetWise 訊號目錄

您可以使用 CreateSignalCatalog API 操作來建立訊號目錄。下列範例使用 AWS CLI。

若要建立訊號目錄，請執行下列命令。

將 *signal-catalog-configuration* 取代為包含組態的 .json 檔案名稱。

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

- 將 *signal-catalog-name* 取代為您建立的訊號目錄名稱。
- （選用） 將*##*取代為描述，以協助您識別訊號目錄。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 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。
> 視覺系統資料處於預覽版本，可能會有所變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateSignalCatalog` API 操作。

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

匯入 an AWS IoT FleetWise 訊號目錄

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

主題

- 匯入訊號目錄（主控台）
- 匯入訊號目錄 (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**
>
> 您最多可以有一個訊號目錄。如果您已有訊號目錄，則不會在主控台中看到匯入訊號目錄的選項。

匯入訊號目錄

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇訊號目錄。
3. 在訊號目錄摘要頁面上，選擇匯入訊號目錄。

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. 對於儲存貯體，輸入儲存貯體名稱或物件，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，輸入 Amazon Simple Storage Service 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. 驗證訊號目錄，然後選擇匯入檔案。

匯入訊號目錄 (AWS CLI)

您可以使用 ImportSignalCatalog API 操作上傳 JSON 檔案，以協助建立訊號目錄。您必須遵循 車輛訊號規格 (VSS)，才能將訊號儲存在 JSON 檔案中。下列範例使用 AWS CLI。

若要匯入訊號目錄，請執行下列命令。

- 將 *signal-catalog-name* 取代為您建立的訊號目錄名稱。
- （選用）將描述取代為*##*，以協助您識別訊號目錄。
- 將 *signal-catalog-configuration-vss* 取代為 JSON 字串檔案名稱，其中包含 VSS 中定義的訊號。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 訊號。

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

JSON 必須經過字串化，並透過 `vssJson` 欄位傳遞。以下是 VSS 中定義的訊號範例。

```
{
 "Vehicle": {
  "type": "branch",
  "children": {
   "Chassis": {
    "type": "branch",
    "description": "All data concerning steering, suspension, wheels, and brakes.",
    "children": {
     "SteeringWheel": {
      "type": "branch",
      "description": "Steering wheel signals",
      "children": {
       "Diameter": {
        "type": "attribute",
        "description": "The diameter of the steering wheel",
        "datatype": "float",
        "unit": "cm",
        "min": 1,
        "max": 50
       },
       "HandsOff": {
        "type": "branch",
        "children": {
         "HandsOffSteeringState": {
          "type": "actuator",
          "description": "HndsOffStrWhlDtSt. Hands Off Steering State",
          "datatype": "boolean"
         },
         "HandsOffSteeringMode": {
          "type": "actuator",
          "description": "HndsOffStrWhlDtMd. Hands Off Steering Mode",
          "datatype": "int8",
          "min": 0,
          "max": 2
         }
        }
       }
      }
     },
     "Accelerator": {
      "type": "branch",
      "description": "",
      "children": {
```

```
      "AcceleratorPedalPosition": {
       "type": "sensor",
       "description": "Throttle__Position. Accelerator pedal position as percent. 0 =
Not depressed. 100 = Fully depressed.",
       "datatype": "uint8",
       "unit": "%",
       "min": 0,
       "max": 100.000035
      }
     }
    }
   }
  },
  "Powertrain": {
   "type": "branch",
   "description": "Powertrain data for battery management, etc.",
   "children": {
    "Transmission": {
     "type": "branch",
     "description": "Transmission-specific data, stopping at the drive shafts.",
     "children": {
      "VehicleOdometer": {
       "type": "sensor",
       "description": "Vehicle_Odometer",
       "datatype": "float",
       "unit": "km",
       "min": 0,
       "max": 67108863.984375
      }
     }
    },
    "CombustionEngine": {
     "type": "branch",
     "description": "Engine-specific data, stopping at the bell housing.",
     "children": {
      "Engine": {
       "type": "branch",
       "description": "Engine description",
       "children": {
        "timing": {
         "type": "branch",
         "description": "timing description",
         "children": {
          "run_time": {
```

```
            "type": "sensor",
            "description": "Engine run time",
            "datatype": "int16",
            "unit": "ms",
            "min": 0,
            "max": 10000
           },
           "idle_time": {
            "type": "sensor",
            "description": "Engine idle time",
            "datatype": "int16",
            "min": 0,
            "unit": "ms",
            "max": 10000
           }
          }
         }
        }
       }
      }
     }
    }
   },
   "Axle": {
    "type": "branch",
    "description": "Axle signals",
    "children": {
     "TireRRPrs": {
      "type": "sensor",
      "description": "TireRRPrs. Right rear Tire pressure in kilo-Pascal",
      "datatype": "float",
      "unit": "kPaG",
      "min": 0,
      "max": 1020
     }
    }
   }
  }
 },
 "Cameras": {
  "type": "branch",
  "description": "Branch to aggregate all cameras in the vehicle",
  "children": {
   "FrontViewCamera": {
```

```
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Front view camera"
   },
   "RearViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Rear view camera"
   },
   "LeftSideViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Left side view camera"
   },
   "RightSideViewCamera": {
    "type": "sensor",
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Right side view camera"
   }
  }
 },
 "ComplexDataTypes": {
  "VehicleDataTypes": {
   "type": "branch",
   "description": "Branch to aggregate all camera related higher order data types",
   "children": {
    "SVMCamera": {
     "type": "struct",
     "description": "This data type represents Surround View Monitor (SVM) camera
system in a vehicle",
     "comment": "Test comment",
     "deprecation": "Test deprecation message",
     "children": {
      "Make": {
       "type": "property",
       "description": "Make of the SVM camera",
       "datatype": "string",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Description": {
       "type": "property",
       "description": "Description of the SVM camera",
       "datatype": "string",
```

```
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "FPS": {
        "type": "property",
        "description": "FPS of the SVM camera",
        "datatype": "double",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "Orientation": {
        "type": "property",
        "description": "Orientation of the SVM camera",
        "datatype": "VehicleDataTypes.Orientation",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "Range": {
        "type": "property",
        "description": "Range of the SVM camera",
        "datatype": "VehicleDataTypes.Range",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "RawData": {
        "type": "property",
        "description": "Represents binary data of the SVM camera",
        "datatype": "uint8[]",
        "dataencoding": "binary",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      },
      "CapturedFrames": {
        "type": "property",
        "description": "Represents selected frames captured by the SVM camera",
        "datatype": "VehicleDataTypes.Frame[]",
        "dataencoding": "typed",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
      }
    }
  },
  "Range": {
    "type": "struct",
```

```
    "description": "Range of a camera in centimeters",
    "comment": "Test comment",
    "deprecation": "Test deprecation message",
    "children": {
     "Min": {
      "type": "property",
      "description": "Minimum range of a camera in centimeters",
      "datatype": "uint32",
      "comment": "Test comment",
      "deprecation": "Test deprecation message"
     },
     "Max": {
      "type": "property",
      "description": "Maximum range of a camera in centimeters",
      "datatype": "uint32",
      "comment": "Test comment",
      "deprecation": "Test deprecation message"
     }
    }
   },
   "Orientation": {
    "type": "struct",
    "description": "Orientation of a camera",
    "comment": "Test comment",
    "deprecation": "Test deprecation message",
    "children": {
     "Front": {
      "type": "property",
      "description": "Indicates whether the camera is oriented to the front of the
vehicle",
      "datatype": "boolean",
      "comment": "Test comment",
      "deprecation": "Test deprecation message"
     },
     "Rear": {
      "type": "property",
      "description": "Indicates whether the camera is oriented to the rear of the
vehicle",
      "datatype": "boolean",
      "comment": "Test comment",
      "deprecation": "Test deprecation message"
     },
     "Side": {
      "type": "property",
```

```
        "description": "Indicates whether the camera is oriented to the side of the
 vehicle",
        "datatype": "boolean",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
       }
      }
     },
     "Frame": {
      "type": "struct",
      "description": "Represents a camera frame",
      "comment": "Test comment",
      "deprecation": "Test deprecation message",
      "children": {
       "Data": {
        "type": "property",
        "datatype": "string",
        "dataencoding": "binary",
        "comment": "Test comment",
        "deprecation": "Test deprecation message"
       }
      }
     }
    }
   }
  }

}
```

下列範例顯示 JSON 字串中在 VSS 中定義的相同訊號。

```
{
   "vssJson": "{\"Vehicle\":{\"type\":\"branch\",\"children\":{\"Chassis\":{\"type
\":\"branch\",\"description\":\"All data concerning steering, suspension, wheels,
 and brakes.\",\"children\":{\"SteeringWheel\":{\"type\":\"branch\",\"description
\":\"Steering wheel signals\",\"children\":{\"Diameter\":{\"type\":\"attribute\",
\"description\":\"The diameter of the steering wheel\",\"datatype\":\"float\",\"unit
\":\"cm\",\"min\":1,\"max\":50},\"HandsOff\":{\"type\":\"branch\",\"children\":
{\"HandsOffSteeringState\":{\"type\":\"actuator\",\"description\":\"HndsOffStrWhlDtSt.
 Hands Off Steering State\",\"datatype\":\"boolean\"},\"HandsOffSteeringMode\":
{\"type\":\"actuator\",\"description\":\"HndsOffStrWhlDtMd. Hands Off Steering Mode
\",\"datatype\":\"int8\",\"min\":0,\"max\":2}}}}},\"Accelerator\":{\"type\":\"branch
\",\"description\":\"\",\"children\":{\"AcceleratorPedalPosition\":{\"type\":\"sensor
```

```
\",\"description\":\"Throttle__Position. Accelerator pedal position as percent. 0
 = Not depressed. 100 = Fully depressed.\",\"datatype\":\"uint8\",\"unit\":\"%\",
\"min\":0,\"max\":100.000035}}}}},\"Powertrain\":{\"type\":\"branch\",\"description
\":\"Powertrain data for battery management, etc.\",\"children\":{\"Transmission\":
{\"type\":\"branch\",\"description\":\"Transmission-specific data, stopping at the
 drive shafts.\",\"children\":{\"VehicleOdometer\":{\"type\":\"sensor\",\"description
\":\"Vehicle_Odometer\",\"datatype\":\"float\",\"unit\":\"km\",\"min\":0,\"max
\":67108863.984375}}},\"CombustionEngine\":{\"type\":\"branch\",\"description\":
\"Engine-specific data, stopping at the bell housing.\",\"children\":{\"Engine\":
{\"type\":\"branch\",\"description\":\"Engine description\",\"children\":{\"timing\":
{\"type\":\"branch\",\"description\":\"timing description\",\"children\":{\"run_time\":
{\"type\":\"sensor\",\"description\":\"Engine run time\",\"datatype\":\"int16\",\"unit
\":\"ms\",\"min\":0,\"max\":10000},\"idle_time\":{\"type\":\"sensor\",\"description
\":\"Engine idle time\",\"datatype\":\"int16\",\"min\":0,\"unit\":\"ms\",\"max
\":10000}}}}}}}}},\"Axle\":{\"type\":\"branch\",\"description\":\"Axle signals\",
\"children\":{\"TireRRPrs\":{\"type\":\"sensor\",\"description\":\"TireRRPrs. Right
 rear Tire pressure in kilo-Pascal\",\"datatype\":\"float\",\"unit\":\"kPaG\",\"min
\":0,\"max\":1020}}}}}}"
}
```

> **Note**
>
> 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。
> 視覺系統資料處於預覽版本，可能會有所變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ImportSignalCatalog` API 操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
{
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
```

```
    },
  ]
}
```

更新 an AWS IoT FleetWise 訊號目錄

您可以使用 UpdateSignalCatalog API 操作來更新現有的訊號目錄。下列範例使用 AWS CLI。

若要更新現有的訊號目錄，請執行下列命令。

將 *signal-catalog-configuration* 取代為包含組態的 .json 檔案名稱。

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

將 *signal-catalog-name* 取代為您更新之訊號目錄的名稱。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 訊號。

> **Important**
>
> 自訂結構是不可變的。如果您需要將屬性重新排序或插入現有自訂結構（結構），請刪除結構，並建立具有所需屬性順序的新結構。
> 若要刪除自訂結構，請在 中新增結構的完整名稱nodesToRemove。如果結構是由任何訊號所參考，則無法刪除。任何參考結構的訊號（其資料類型定義為目標結構）都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
      }
     }]
    }
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `UpdateSignalCatalog` API 操作。

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

```
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

驗證訊號目錄更新

您可以使用 ListSignalCatalogNodes API 操作來驗證訊號目錄是否已更新。下列範例使用 AWS CLI。

若要擷取指定訊號目錄中所有訊號（節點）的分頁清單，請執行下列命令。

將 *signal-catalog-name* 取代為您檢查的訊號目錄名稱。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListSignalCatalogNodes` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

刪除 AWS IoT FleetWise 訊號目錄

您可以使用 DeleteSignalCatalog API 操作來刪除訊號目錄。下列範例使用 AWS CLI。

> **Important**
>
> 在刪除訊號目錄之前，請確定它沒有相關聯的車輛模型、解碼器資訊清單、車輛、機群或行銷活動。如需詳細說明，請參閱下列主題：
>
> - 刪除 an AWS IoT FleetWise 車輛模型
> - 刪除 an AWS IoT FleetWise 解碼器資訊清單
> - 刪除 AWS IoT FleetWise 車輛
> - 刪除 AWS IoT FleetWise 機群
> - 刪除 an AWS IoT FleetWise 行銷活動

若要刪除現有的訊號目錄，請執行下列命令。將 *signal-catalog-name* 取代為您要刪除的訊號目錄名稱。

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

驗證訊號目錄刪除

您可以使用 ListSignalCatalogs API 操作來驗證訊號目錄是否已刪除。下列範例使用 AWS CLI。

若要擷取所有訊號目錄的摘要分頁清單，請執行下列命令。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListSignalCatalogs` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
```

```
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Get AWS IoT FleetWise 訊號目錄資訊

您可以使用 GetSignalCatalog API 操作來擷取訊號目錄資訊。下列範例使用 AWS CLI。

若要擷取訊號目錄的相關資訊，請執行下列命令。

將 *signal-catalog-name* 取代為您要擷取的訊號目錄名稱。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetSignalCatalog` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

> **Note**
>
> 此操作為最終一致。換句話說，可能不會立即反映對訊號目錄的變更。

Manage AWS IoT FleetWise 車輛模型

您可以使用訊號來建立車輛模型，以協助標準化車輛的格式。車輛模型會在相同類型的多部車輛之間強制執行一致的資訊，以便您處理來自車輛機群的資料。從相同車輛模型建立的車輛會繼承相同的訊號群組。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛。

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

- `ACTIVE` – 車輛模型處於作用中狀態。
- `DRAFT` – 已儲存車輛模型的組態。

> **Important**
>
> - 您必須有訊號目錄，才能使用 `CreateModelManifest` API 操作建立車輛模型。如需詳細資訊，請參閱建立 AWS IoT FleetWise 訊號目錄。
> - 如果您使用 AWS IoT FleetWise 主控台建立車輛模型， AWS IoT FleetWise 會自動為您啟用車輛模型。
> - 如果您使用 `CreateModelManifest` API 操作來建立車輛模型，則車輛模型會保持 `DRAFT` 狀態。
> - 您無法從處於 `DRAFT` 狀態的車輛模型建立車輛。使用 `UpdateModelManifest` API 操作將車輛模型變更為 `ACTIVE` 狀態。
> - 您無法編輯處於 `ACTIVE` 狀態的車輛模型。

主題

- 建立 An AWS IoT FleetWise 車輛模型
- 更新 an AWS IoT FleetWise 車輛模型
- 刪除 an AWS IoT FleetWise 車輛模型
- Get AWS IoT FleetWise 車輛模型資訊

建立 An AWS IoT FleetWise 車輛模型

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

主題

- 建立車輛模型（主控台）
- 建立車輛模型 (AWS CLI)

建立車輛模型（主控台）

在 AWS IoT FleetWise 主控台中，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 提供內建診斷 (OBD) II、J1979 範本，可自動為您建立訊號目錄、車輛模型和解碼器資訊清單。範本也會將 OBD 網路介面新增至解碼器資訊清單。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

使用範本建立車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 在車輛模型頁面上，選擇新增提供的範本。
4. 選擇內建診斷 (OBD) II。
5. 輸入 AWS IoT FleetWise 所建立之 OBD 網路界面的名稱。
6. 選擇新增。

手動建立車輛模型

您可以上傳一或多個 .dbc 檔案，從訊號目錄新增訊號或匯入訊號。.dbc 檔案是控制器區域網路 (CAN 匯流排）資料庫支援的檔案格式。

> **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。反 AWS CLI 之，請使用 來建立車輛模型。

視覺系統資料處於預覽版本，可能會有所變更。

手動建立車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 在車輛模型頁面上，選擇建立車輛模型，然後執行下列動作。

主題

- 步驟 1：設定車輛模型
- 步驟 2：新增訊號
- 步驟 3：匯入訊號
- （選用） 步驟 4：新增屬性
- 步驟 4：檢閱和建立

步驟 1：設定車輛模型

在一般資訊中，執行下列動作。

1. 輸入車輛模型的名稱。
2. (選用) 輸入描述。
3. 選擇下一步。

步驟 2：新增訊號

> **Note**
>
> - 如果這是您第一次使用 AWS IoT FleetWise，在您擁有訊號目錄之前，此步驟將無法使用。建立第一個車輛模型時， AWS IoT FleetWise 會自動建立訊號目錄，並將訊號新增至第一個車輛模型。
> - 如果您使用 AWS IoT FleetWise，您可以從訊號目錄中選取訊號或上傳 .dbc 檔案以匯入訊號，將訊號新增至您的車輛模型。
> - 您必須至少有一個訊號才能建立車輛模型。

新增訊號

1. 從您要新增至車輛模型的訊號目錄中，選擇一或多個訊號。您可以在右側窗格中檢閱選取的訊號。

> **Note**
>
> 只有選取的訊號才會新增至車輛模型。

2. 選擇下一步。

步驟 3：匯入訊號

> **Note**
>
> - 如果這是您第一次使用 AWS IoT FleetWise，您必須至少上傳一個 .dbc 檔案來匯入訊號。
> - 如果您使用 AWS IoT FleetWise，您可以從訊號目錄中選取訊號或上傳 .dbc 檔案以匯入訊號，將訊號新增至您的車輛模型。
> - 您必須至少有一個訊號才能建立車輛模型。

匯入訊號

1. 選擇選擇檔案。
2. 在對話方塊中，選擇包含訊號的 .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。

   在訊號區段中，為每個訊號指定下列中繼資料。

   - 名稱 – 訊號的名稱。

     訊號名稱必須是唯一的。訊號名稱加上路徑最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）。
   - 資料類型 – 訊號的資料類型必須是下列其中一項：INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64,
     UINT64、BOOLEAN、FLOAT、DOUBLE、STRRING、UNIX_TIMESTAMP、INT8_ARRAY、UINT8_
     或 UNKNOWNOWN。
   - 訊號類型 – 訊號的類型，可以是感應器或傳動器。

- （選用）單位 – 訊號的科學單位，例如 km 或 Celsius。
- （選用）路徑 – 訊號的路徑。與 JSONPath 類似，請使用 dot(.) 來參考子訊號。例如 **`Vehicle.Engine.Light`**。

  訊號名稱加上路徑最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）。
- （選用）最小值 – 訊號的最小值。
- （選用）最大值 – 訊號的最大值。
- （選用）描述 – 訊號的描述。

  描述最多可有 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。

（選用）步驟 4：新增屬性

您最多可以新增 100 個屬性，包括訊號目錄中的現有屬性。

新增屬性

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。

     訊號名稱必須是唯一的。訊號名稱和路徑最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）
   - 資料類型 – 屬性的資料類型必須是下列其中一項：INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64,
     UINT64、BOOLEAN、FLOAT、DOUBLE、STRRING、UNIX_TIMESTAMP、INT8_ARRAY、UINT8_
     或 UNKNOWNOWN
   - （選用）單位 – 屬性的科學單位，例如 km 或 Celsius。
   - （選用）路徑 – 訊號的路徑。與 JSONPath 類似，請使用 dot(.) 來參考子訊號。例如 **`Vehicle.Engine.Light`**。

     訊號名稱加上路徑最多可有 150 個字元。有效字元：a–z、A–Z、0–9、：（冒號）和 _（底線）
   - （選用）最小值 – 屬性的最小值。

- （選用）最大值 – 屬性的最大值。
- （選用）描述 – 屬性的描述。

  描述最多可有 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

驗證車輛模型的組態，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組態，以建立新的模型。所選車輛模型中指定的訊號會複製到新車輛模型。

複製車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 從車輛模型清單中選擇模型，然後選擇複製模型。

若要設定車輛模型，請遵循手動建立車輛模型教學課程。

AWS IoT FleetWise 可能需要幾分鐘的時間來處理建立車輛模型的請求。成功建立車輛模型後，在車輛模型頁面上，狀態欄會顯示 ACTIVE。當車輛模型變為作用中時，您無法編輯它。

建立車輛模型 (AWS CLI)

您可以使用 CreateModelManifest API 操作來建立車輛模型（模型資訊清單）。下列為使用 AWS CLI 的範例。

> **Important**
>
> 您必須有訊號目錄，才能使用 `CreateModelManifest` API 操作建立車輛模型。如需如何建立訊號目錄的詳細資訊，請參閱 建立 AWS IoT FleetWise 訊號目錄。

若要建立車輛模型，請執行下列命令。

將 *vehicle-model-configuration* 取代為包含組態的 .json 檔案名稱。

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

- 將 *vehicle-model-name* 取代為您建立之車輛模型的名稱。
- 將 *signal-catalog-ARN* 取代為訊號目錄的 Amazon Resource Name (ARN)。
- （選用）將##取代為描述，以協助您識別車輛模型。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 訊號。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateModelManifest` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

更新 an AWS IoT FleetWise 車輛模型

您可以使用 UpdateModelManifest API 操作來更新現有的車輛模型（模型資訊清單）。下列為使用 AWS CLI的範例。

若要更新現有的車輛模型，請執行下列命令。

將 *update-vehicle-model-configuration* 取代為包含組態的 .json 檔案名稱。

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

- 將 *vehicle-model-name* 取代為您更新的車輛模型名稱。
- （選用）若要啟用車輛模型，請以 取代 *vehicle-model-status*ACTIVE。

> **Important**
>
> 車輛模型啟用後，您就無法變更車輛模型。

- （選用）將##取代為更新的描述，以協助您識別車輛模型。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 UpdateModelManifest API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
```

```
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

驗證車輛模型更新

您可以使用 ListModelManifestNodes API 操作來驗證車輛模型是否已更新。下列範例使用 AWS CLI。

若要擷取指定車輛模型中所有訊號（節點）的分頁清單，請執行下列命令。

將 *vehicle-model-name* 取代為您檢查的車輛模型名稱。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListModelManifestNodes` API 操作。

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

刪除 an AWS IoT FleetWise 車輛模型

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

> **Important**
>
> 與車輛模型相關聯的車輛和解碼器資訊清單必須先刪除。如需詳細資訊，請參閱刪除 AWS IoT FleetWise 車輛及刪除 an AWS IoT FleetWise 解碼器資訊清單。

刪除車輛模型（主控台）

若要刪除車輛模型，請使用 AWS IoT FleetWise 主控台。

刪除車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 在車輛模型頁面上，選擇目標車輛模型。
4. 選擇 刪除。
5. 在刪除 **vehicle-model-name**？ 中，輸入要刪除的車輛模型名稱，然後選擇確認。

刪除車輛模型 (AWS CLI)

您可以使用 DeleteModelManifest API 操作來刪除現有的車輛模型（模型資訊清單）。下列為使用 AWS CLI的範例。

若要刪除車輛模型，請執行下列命令。

將 *model-manifest-name* 取代為您要刪除之車輛模型的名稱。

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

驗證刪除車輛模型

您可以使用 ListModelManifests API 操作來驗證是否已刪除車輛模型。下列範例使用 AWS CLI。

若要擷取所有車輛模型的摘要分頁清單，請執行下列命令。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListModelManifests` API 操作。

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

Get AWS IoT FleetWise 車輛模型資訊

您可以使用 GetModelManifest API 操作來擷取車輛模型的相關資訊。下列範例使用 AWS CLI。

若要擷取車輛模型的相關資訊，請執行下列命令。

將 ####取代為您要擷取的車輛模型名稱。

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

> **Note**
>
> 此操作為最終一致。換句話說，可能不會立即反映對車輛模型的變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetModelManifest` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
```

```
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Manage AWS IoT FleetWise 解碼器資訊清單

> **Important**
>
> 存取特定 AWS IoT FleetWise 功能目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

解碼器資訊清單包含 AWS IoT FleetWise 用來將車輛資料（二進位資料）轉換為人類可讀值的解碼資訊，並準備您的資料以供資料分析。網路界面和訊號解碼器是您用來設定解碼器資訊清單的核心元件。

網路介面

包含車輛網路使用的通訊協定相關資訊。 AWS IoT FleetWise 支援下列通訊協定。

控制器區域網路 (CAN 匯流排）

定義如何在電子控制單元 (ECUs之間通訊資料的通訊協定。ECUs可以是引擎控制單元、安全氣囊或音訊系統。

內建診斷 (OBD) II

進一步開發的通訊協定，定義自我診斷資料如何在 ECUs 之間通訊。它提供多種標準診斷問題代碼 (DTCs)，可協助識別您車輛的問題。

車輛中介軟體

車輛中介軟體定義為一種網路界面類型。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) 的可擴展服務導向 MiddlewarE。

> **Note**
>
> AWS IoT FleetWise 支援 ROS 2 中介軟體來處理視覺系統資料。

自訂界面

您也可以使用自己的界面在 Edge 解碼訊號。這可以節省您的時間，因為您不需要在雲端中建立解碼規則。

訊號解碼器

提供特定訊號的詳細解碼資訊。車輛模型中指定的每個訊號都必須與訊號解碼器配對。如果解碼器資訊清單包含 CAN 網路介面，則必須包含 CAN 解碼器訊號。如果解碼器資訊清單包含 OBD 網路介面，則必須包含 OBD 訊號解碼器。

如果解碼器資訊清單也包含車輛中介軟體界面，則必須包含訊息訊號解碼器。或者，如果解碼器資訊清單包含自訂解碼介面，它也必須包含自訂解碼訊號。

每個解碼器資訊清單都必須與車輛模型相關聯。 AWS IoT FleetWise 會使用相關聯的解碼器資訊清單，從根據車輛模型建立的車輛解碼資料。

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` – 解碼器資訊清單處於作用中狀態。
- `DRAFT` – 不會儲存解碼器資訊清單的組態。
- `VALIDATING` – 解碼器資訊清單正在驗證其資格。這僅適用於至少包含一個視覺系統資料訊號的解碼器資訊清單。
- `INVALID` – 解碼器資訊清單驗證失敗，尚無法啟用。這僅適用於至少包含一個視覺系統資料訊號的解碼器資訊清單。您可以使用 ListDecoderManifests 和 GetDecoderManifest APIs 來檢查驗證失敗的原因。

Important

- 如果您使用 AWS IoT FleetWise 主控台建立解碼器資訊清單， AWS IoT FleetWise 會自動為您啟用解碼器資訊清單。
- 如果您使用 `CreateDecoderManifest` API 操作來建立解碼器資訊清單，解碼器資訊清單會保持 `DRAFT` 狀態。
- 您無法從與`DRAFT`解碼器資訊清單相關聯的車輛模型建立車輛。使用 `UpdateDecoderManifest` API 操作將解碼器資訊清單變更為 `ACTIVE` 狀態。

- 您無法編輯處於 ACTIVE 狀態的解碼器資訊清單。

主題

- Configure AWS IoT FleetWise 網路介面和解碼器訊號
- 建立 an AWS IoT FleetWise 解碼器資訊清單
- 更新 an AWS IoT FleetWise 解碼器資訊清單
- 刪除 an AWS IoT FleetWise 解碼器資訊清單
- Get AWS IoT FleetWise 解碼器資訊清單資訊

Configure AWS IoT FleetWise 網路介面和解碼器訊號

> **Important**
>
> 目前已封鎖對特定 AWS IoT FleetWise 功能的存取。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

每個解碼器資訊清單至少都有一個網路界面和訊號解碼器，搭配相關聯車輛模型中指定的訊號。

如果解碼器資訊清單包含 CAN 網路介面，則必須包含 CAN 訊號解碼器。如果解碼器資訊清單包含 OBD 網路介面，則必須包含 OBD 訊號解碼器。

主題

- 設定網路介面
- 設定訊號解碼器

設定網路介面

若要設定 CAN 網路介面，請指定下列資訊。

- name – CAN 界面的名稱。

  界面名稱必須是唯一的，並且可以有 1-100 個字元。
- （選用）protocolName – 通訊協定的名稱。

有效值： `CAN-FD`和 `CAN`

- （選用） `protocolVersion` – AWS IoT FleetWise 目前支援 CAN-FD 和 CAN 2.0b。

  有效值： `1.0`和 `2.0b`

若要設定 OBD 網路界面，請指定下列資訊。

- `name` – OBD 界面的名稱。

  界面名稱必須是唯一的，並且可以有 1-100 個字元。
- `requestMessageId` – 請求資料之訊息的 ID。
- （選用） `dtcRequestIntervalSeconds` – 在幾秒鐘內向車輛請求診斷問題代碼 (DTCs) 的頻率。例如，如果指定的值為 120，Edge Agent 軟體會每 2 分鐘收集一次儲存DTCs。
- （選用） `hasTransmissionEcu` – 車輛是否有傳輸控制模組 (TCM)。

  有效值： `true`和 `false`
- （選用） `obdStandard` – AWS IoT FleetWise 支援的 OBD 標準。 AWS IoT FleetWise 目前支援全球協調內建診斷 (WWH-OBD) ISO15765-4 標準。
- （選用） `pidRequestIntervalSeconds` – 從車輛請求 OBD II PIDs的頻率。例如，如果指定的值為 120，Edge Agent 軟體會每 2 分鐘收集一次 OBD II PIDs。
- （選用） `useExtendedIds` – 是否在訊息中使用延伸 IDs。

  有效值： `true`和 `false`

若要設定車輛中介軟體網路界面，請指定下列資訊。

- `name` – 車輛中介軟體界面的名稱。

  界面名稱必須是唯一的，並且可以有 1-100 個字元。
- `protocolName` – 通訊協定的名稱。

  有效值：`ROS_2`

若要設定自訂解碼界面，請指定下列資訊。

- `name` – 您用來在 Edge 解碼訊號的解碼器名稱。

解碼器界面名稱可以有 1-100 個字元。

設定訊號解碼器

若要設定 CAN 訊號解碼器，請指定下列資訊。

- factor – 用來解碼訊息的乘數。
- isBigEndian – 訊息的位元組排序是否為大端。如果是大端值，序列中最重要的值會先儲存在最低的儲存地址。
- isSigned – 是否簽署訊息。如果已簽署，訊息可以同時代表正數和負數。
- length – 訊息的長度，以位元組為單位。
- messageId – 訊息的 ID。
- offset – 用來計算訊號值的位移。結合因素，計算為 `value = raw_value * factor + offset`。
- startBit – 指出訊息第一個位元的位置。
- （選用）name – 訊號的名稱。
- （選用）signalValueType – 訊號的值類型。整數是預設值類型。

若要設定 OBD 訊號解碼器，請指定下列資訊。

- byteLength – 訊息的長度，以位元組為單位。
- offset – 用來計算訊號值的位移。結合擴展，計算為 `value = raw_value * scaling + offset`。
- pid – 用來向車輛請求此訊號訊息的診斷代碼。
- pidResponseLength – 請求訊息的長度。
- scaling – 用來解碼訊息的乘數。
- serviceMode – 訊息中的操作模式（診斷服務）。
- startByte – 表示訊息的開頭。
- （選用）bitMaskLength – 訊息中遮罩的位元數。
- （選用）bitRightShift – 向右移動的位置數。
- （選用）isSigned – 是否簽署訊息。如果已簽署，訊息可以同時代表正數和負數。預設不會簽署訊息 (false)。

- （選用） `signalValueType` – 訊號的值類型。整數是預設值類型。

若要設定訊息訊號解碼器，請指定下列資訊。

- `topicName` – 訊息訊號的主題名稱。它對應至 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 primitiveMessageDefinition、structuredMessageListDefinition 或 structuredMessageDefinition 遞迴定義。

若要設定自訂解碼訊號，請指定下列資訊。

- （選用） `id` – 您使用解碼器界面自行解碼的訊號 ID。訊號 ID 可以有 1-150 個字元。如果未指定，則 `id`預設為訊號`fullyQualifiedName`的 。

建立 an AWS IoT FleetWise 解碼器資訊清單

> **Important**
>
> 存取特定 AWS IoT FleetWise 功能目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 AWS IoT FleetWise 主控台或 API 為您的車輛模型建立解碼器資訊清單。

主題

- 建立解碼器資訊清單（主控台）
- 建立解碼器資訊清單 (AWS CLI)

建立解碼器資訊清單（主控台）

您可以使用 AWS IoT FleetWise 主控台來建立與您的車輛模型相關聯的解碼器資訊清單。

> **Important**
>
> 您不能使用 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。請改用 AWS CLI。視覺系統資料處於預覽版本，可能會有所變更。

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 車輛模型。
3. 選擇目標車輛模型。
4. 在車輛模型摘要頁面上，選擇建立解碼器資訊清單，然後執行下列動作。

主題

- 步驟 1：設定解碼器資訊清單
- 步驟 2：映射 CAN 介面
- 步驟 3：檢閱和建立

步驟 1：設定解碼器資訊清單

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。

新增網路介面

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

新增網路界面

1. 上傳網路介面檔案。您可以上傳 CAN 通訊協定的 .dbc 檔案，或是 ROS 2 或自訂界面的 .json 檔案。
2. 輸入網路界面的名稱。如果您上傳了自訂界面，則已提供名稱。

映射遺失訊號

如果車輛模型中有訊號在上傳的網路介面中遺失配對訊號解碼器，您可以建立預設自訂解碼器來映射遺失的訊號。這是選用的，因為您可以在下一個步驟中手動映射訊號。

建立預設自訂解碼器

1. 選取為遺失訊號建立預設自訂解碼器。
2. 選擇下一步。

步驟 2：映射 CAN 介面

您可以使用 CAN 訊號解碼器對應 CAN 訊號。如果您選取為遺失訊號建立預設自訂解碼器核取方塊，則遺失解碼器訊號的任何訊號都會自動對應至預設自訂訊號解碼器。

映射 CAN 訊號

1. 在 CAN 訊號映射中，選取訊號解碼器。
2. 選擇下一步。

> **Note**
> 如果您已新增 ROS 2 或自訂界面，您可以在建立解碼器資訊清單之前驗證映射。

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。

建立解碼器資訊清單 (AWS CLI)

您可以使用 CreateDecoderManifest API 操作來建立解碼器資訊清單。下列為使用 AWS CLI的範例。

> **Important**
> 您必須先擁有 車輛模型，才能建立解碼器資訊清單。每個解碼器資訊清單都必須與車輛模型相關聯。如需詳細資訊，請參閱建立 An AWS IoT FleetWise 車輛模型。

若要建立解碼器資訊清單，請執行下列命令。

將 *decoder-manifest-configuration* 取代為包含組態的 .json 檔案名稱。

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

- 將 *decoder-manifest-name* 取代為您建立的解碼器資訊清單名稱。
- 將 *vehicle-model-ARN* 取代為 車輛模型的 Amazon Resource Name (ARN)。
- （選用）將##取代為描述，以協助您識別解碼器資訊清單。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 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": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_INTERFACE"
        }
    ],
    "signalDecoders": [
        {
            "canSignal": {
                "name": "Engine_Idle_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
                "offset": 0,
                "startBit": 16
            },
            "fullyQualifiedName": "Vehicle.EngineIdleTime",
```

```
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        },
        {
            "canSignal": {
                "name": "Engine_Run_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
                "offset": 0,
                "startBit": 40
            },
            "fullyQualifiedName": "Vehicle.EngineRunTime",
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        }
    ]
}
```

- 將 *decoder-manifest-name* 取代為您建立的解碼器資訊清單名稱。
- 將 *vehicle-model-ARN* 取代為 車輛模型的 Amazon Resource Name (ARN)。
- （選用）將*##*取代為描述，以協助您識別解碼器資訊清單。

結構（結構）中的屬性節點順序必須保持一致，如訊號目錄和車輛模型（模型資訊清單）中所定義。如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 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": "Qq1acaenByOB3sSM39SYm",
  "type": "CAN_INTERFACE"
```

```
 }, {
  "type": "VEHICLE_MIDDLEWARE",
  "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
  "vehicleMiddleware": {
   "name": "ROS2_test",
   "protocolName": "ROS_2"
  }
 }],
 "signalDecoders": [{
   "canSignal": {
    "name": "Engine_Idle_Time",
    "factor": 1,
    "isBigEndian": true,
    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 16
   },
   "fullyQualifiedName": "Vehicle.EngineIdleTime",
   "interfaceId": "Qq1acaenByOB3sSM39SYm",
   "type": "CAN_SIGNAL"
  },
  {
   "canSignal": {
    "name": "Engine_Run_Time",
    "factor": 1,
    "isBigEndian": true,
    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 40
   },
   "fullyQualifiedName": "Vehicle.EngineRunTime",
   "interfaceId": "Qq1acaenByOB3sSM39SYm",
   "type": "CAN_SIGNAL"
  },
  {
   "fullyQualifiedName": "Vehicle.CompressedImageTopic",
   "type": "MESSAGE_SIGNAL",
   "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
   "messageSignal": {
    "topicName": "CompressedImageTopic:sensor_msgs/msg/CompressedImage",
```

```
    "structuredMessage": {
     "structuredMessageDefinition": [{
       "fieldName": "header",
       "dataType": {
        "structuredMessageDefinition": [{
          "fieldName": "stamp",
          "dataType": {
           "structuredMessageDefinition": [{
             "fieldName": "sec",
             "dataType": {
              "primitiveMessageDefinition": {
               "ros2PrimitiveMessageDefinition": {
                "primitiveType": "INT32"
               }
              }
             }
            },
            {
             "fieldName": "nanosec",
             "dataType": {
              "primitiveMessageDefinition": {
               "ros2PrimitiveMessageDefinition": {
                "primitiveType": "UINT32"
               }
              }
             }
            }
           ]
          }
         },
         {
          "fieldName": "frame_id",
          "dataType": {
           "primitiveMessageDefinition": {
            "ros2PrimitiveMessageDefinition": {
             "primitiveType": "STRING"
            }
           }
          }
         }
        ]
       }
      },
      {
```

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

- 將 *decoder-manifest-name* 取代為您建立的解碼器資訊清單名稱。
- 將 *vehicle-model-ARN* 取代為 車輛模型的 Amazon Resource Name (ARN)。
- （選用）將##取代為描述，以協助您識別解碼器資訊清單。

如需如何設定分支、屬性、感應器和傳動器的詳細資訊，請參閱 Configure AWS IoT FleetWise 網路介面和解碼器訊號。

```
{
```

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

Note

您可以下載示範指令碼，以建立具有視覺系統訊號的解碼器資訊清單。如需詳細資訊，請參閱視覺系統資料開發人員指南。

視覺系統資料處於預覽版本，可能會有所變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateDecoderManifest` API 操作。

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

更新 an AWS IoT FleetWise 解碼器資訊清單

> **Important**
>
> 存取特定 AWS IoT FleetWise 功能目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 UpdateDecoderManifest API 操作來更新解碼器資訊清單。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。

將 *decoder-manifest-name* 取代為您更新之解碼器資訊清單的名稱。

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

如果訊號沒有指定的解碼規則，您可以建立預設解碼規則。訊號會新增至自訂解碼界面，並將 `CustomDecodingSignal$id`設定為訊號的完整名稱。若要使用預設解碼規則更新解碼器資訊清單，請執行下列命令。

將 *decoder-manifest-name* 取代為您更新之解碼器資訊清單的名稱。

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

> **Important**
>
> 啟用解碼器資訊清單後，您就無法編輯。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `UpdateDecoderManifest` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

驗證解碼器資訊清單更新

您可以使用 ListDecoderManifestSignals API 操作來驗證解碼器資訊清單中的解碼器訊號是否已更新。下列範例使用 AWS CLI。

若要擷取指定解碼器資訊清單中所有解碼器訊號（節點）的分頁清單，請執行下列命令。

將 *decoder-manifest-name* 取代為您檢查的解碼器資訊清單名稱。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListDecoderManifestSignals` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

您可以使用 ListDecoderManifestNetworkInterfaces API 操作來驗證解碼器資訊清單中的網路介面是否已更新。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的分頁清單，請執行下列命令。

將 *decoder-manifest-name* 取代為您檢查的解碼器資訊清單名稱。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListDecoderManifestNetworkInterfaces` API 操作。

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

    {
```

```
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

刪除 an AWS IoT FleetWise 解碼器資訊清單

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

> **Important**
>
> 與解碼器資訊清單相關聯的車輛必須先刪除。如需詳細資訊，請參閱刪除 AWS IoT FleetWise 車輛。

主題

- 刪除解碼器資訊清單（主控台）
- 刪除解碼器資訊清單 (AWS CLI)

刪除解碼器資訊清單（主控台）

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

刪除解碼器資訊清單

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛模型。
3. 選擇目標車輛模型。
4. 在車輛模型摘要頁面上，選擇解碼器資訊清單索引標籤。
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**？ 中，輸入要刪除的解碼器資訊清單名稱，然後選擇確認。

刪除解碼器資訊清單 (AWS CLI)

您可以使用 DeleteDecoderManifest API 操作來刪除解碼器資訊清單。下列範例使用 AWS CLI。

> **Important**
>
> 刪除解碼器資訊清單之前，請先刪除相關聯的車輛。如需詳細資訊，請參閱刪除 AWS IoT FleetWise 車輛。

若要刪除解碼器資訊清單，請執行下列命令。

將 *decoder-manifest-name* 取代為您要刪除的解碼器資訊清單名稱。

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

驗證解碼器資訊清單刪除

您可以使用 ListDecoderManifests API 操作來驗證是否已刪除解碼器資訊清單。下列範例使用 AWS CLI。

若要擷取所有解碼器資訊清單的摘要分頁清單，請執行下列命令。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListDecoderManifests` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
```

```
  ]
}
```

Get AWS IoT FleetWise 解碼器資訊清單資訊

您可以使用 GetDecoderManifest API 操作來驗證解碼器資訊清單中的網路介面和訊號解碼器是否已更新。下列範例使用 AWS CLI。

若要擷取解碼器資訊清單的相關資訊，請執行下列命令。

將 *decoder-manifest* 取代為您要擷取的解碼器資訊清單名稱。

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

> **Note**
>
> 此操作為最終一致。換言之，可能不會立即反映解碼器資訊清單的變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetDecoderManifest` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Manage AWS IoT FleetWise 車輛

車輛是 車輛模型的執行個體。車輛必須從車輛模型建立，並與解碼器資訊清單相關聯。車輛會將一或多個資料串流上傳至雲端。例如，車輛可以將加熱資料的里程、引擎溫度和狀態傳送至雲端。每部車輛都包含下列資訊：

`vehicleName`

識別車輛的 ID。

請勿在您的車輛名稱中新增個人身分識別資訊 (PII) 或其他機密或敏感資訊。車輛名稱可供 AWS 其他服務存取，包括 Amazon CloudWatch。車輛名稱不適用於私有或敏感資料。

`modelManifestARN`

車輛模型（模型資訊清單）的 Amazon Resource Name (ARN)。每部車輛都是從車輛模型建立的。從相同車輛模型建立的車輛由繼承自車輛模型的相同訊號群組組成。這些訊號是在訊號目錄中定義和標準化。

`decoderManifestArn`

解碼器資訊清單的 ARN。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）轉換為人類可讀取的值。解碼器資訊清單必須與車輛模型相關聯。 AWS IoT FleetWise 使用相同的解碼器資訊清單，從根據相同車輛模型建立的車輛解碼原始資料。

`attributes`

屬性是包含靜態資訊的鍵值對。車輛可以包含繼承自車輛模型的屬性。您可以新增其他屬性，以區分個別車輛與從相同車輛模型建立的其他車輛。例如，如果您有黑車，您可以為 屬性指定下列值：`{"color": "black"}`。

> **Important**
>
> 屬性必須在相關聯的車輛模型中定義，才能將其新增至個別車輛。

如需車輛模型、解碼器資訊清單和屬性的詳細資訊，請參閱 Model AWS IoT FleetWise 車輛。

AWS IoT FleetWise 提供下列 API 操作，您可以用來建立和管理 車輛。

- CreateVehicle – 建立新車輛。
- BatchCreateVehicle – 建立一或多個新車輛。

- UpdateVehicle – 更新現有車輛。
- BatchUpdateVehicle – 更新一或多個現有車輛。
- DeleteVehicle – 刪除現有車輛。
- ListVehicles – 擷取所有車輛的分頁摘要清單。
- GetVehicle – 擷取車輛的相關資訊。

教學課程

- Provision AWS IoT FleetWise 車輛
- AWS IoT FleetWise 中的預留主題
- 建立 AWS IoT FleetWise 車輛
- 建立多個 AWS IoT FleetWise 車輛
- 更新 an AWS IoT FleetWise 車輛
- 更新多個 AWS IoT FleetWise 車輛
- 刪除 AWS IoT FleetWise 車輛
- Get AWS IoT FleetWise 車輛資訊

Provision AWS IoT FleetWise 車輛

車輛中執行的 Edge Agent for AWS IoT FleetWise 軟體會收集資料並將其傳輸至雲端。 AWS IoT FleetWise 與 整合 AWS IoT Core ，以支援 Edge Agent 軟體與透過 MQTT 的雲端之間的安全通訊。每部車輛對應至 實 AWS IoT 物。您可以使用現有的 AWS IoT 實物來建立車輛，或設定 AWS IoT FleetWise 來自動為您的車輛建立實 AWS IoT 物。如需詳細資訊，請參閱建立 AWS IoT FleetWise 車輛。

AWS IoT Core 支援身分驗證和授權，可協助安全地控制對 AWS IoT FleetWise 資源的存取。車輛可以使用 X.509 憑證進行身分驗證 （登入），以使用 AWS IoT FleetWise 和 AWS IoT Core 政策來取得授權 （具有許可） 來執行指定的動作。

驗證 車輛

您可以建立 AWS IoT Core 政策來驗證您的 車輛。

驗證您的 車輛

- 若要建立 AWS IoT Core 政策，請執行下列命令。

- 將 *policy-name* 取代為您要建立的政策名稱。
- 將 *file-name* 取代為包含 AWS IoT Core 政策的 JSON 檔案名稱。

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

使用範例政策之前，請執行下列動作：

- 將##取代為您建立 AWS IoT FleetWise 資源 AWS 的區域。
- 將 *awsAccount* 取代為 AWS 您的帳戶 ID。

此範例包含 AWS IoT FleetWise 保留的主題。您必須將主題新增至政策。如需詳細資訊，請參閱AWS IoT FleetWise 中的預留主題。

```
{
    "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**
>
> 建議您為每個車輛建立新的憑證。

1. 若要建立 RSA 金鑰對並發行 X.509 憑證，請執行下列命令。

   - 將 *cert* 取代為儲存 certificatePem 命令輸出內容的檔案名稱。

- 將 *public-key* 取代為儲存 keyPair.PublicKey.
- 將 *private-key* 取代為儲存 keyPair.PrivateKey.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile cert.pem \
    --public-key-outfile public-key.key" \
    --private-key-outfile  private-key.key"
```

2. 從輸出複製憑證的 Amazon Resource Name (ARN)。
3. 若要將政策連接至憑證，請執行下列命令。
   - 將 *policy-name* 取代為您建立 AWS IoT Core 的政策名稱。
   - 將 *certificate-arn* 取代為您複製之憑證的 ARN。

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

4. 若要將憑證連接至物件，請執行下列命令。
   - 以您的*####### ID ## thing-name*。 AWS IoT
   - 將 *certificate-arn* 取代為您複製之憑證的 ARN。

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

AWS IoT FleetWise 中的預留主題

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

AWS IoT FleetWise 保留下列主題的使用。如果預留主題允許，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。如果您使用不支援的發佈或訂閱操作搭配預留主題，可能會導致連線結束。

主題	允許用戶端操作	描述
`$aws/iotfleetwise/vehicles/vehicleName/checkins`	發布	Edge Agent 軟體會將車輛狀態資訊發佈至此主題。 車輛狀態資訊會以通訊協定緩衝區 (Protobuf) 格式交換。如需詳細資訊，請參閱 Edge Agent for AWS IoT FleetWise 軟體開發人員指南。
`$aws/iotfleetwise/vehicles/vehicleName/signals`	發布	Edge Agent 軟體會將訊號發佈至此主題。 訊號資訊會以通訊協定緩衝區 (Protobuf) 格式交換。如需詳細資訊，請參閱 Edge Agent for AWS IoT FleetWise 軟體開發人員指南。
`$aws/iotfleetwise/vehicles/vehicleName/collection_schemes`	訂閱	AWS IoT FleetWise 會將資料收集方案發佈至此主題。車輛會使用這些資料收集方案。
`$aws/iotfleetwise/`	訂閱	AWS IoT FleetWise 會將解碼器資訊清單

主題	允許用戶端操作	描述	
vehicles/ *vehicleName* /decoder_manifests		發佈至此主題。車輛會使用這些解碼器資訊清單。	
$aws/iotfleetwise/vehicles/ *vehicleName* /command/request	訂閱	AWS IoT FleetWise 會將執行命令的請求發佈至此主題。車輛接著會使用這些命令請求。	
$aws/iotfleetwise/vehicles/ *vehicleName* /command/response	發布	Edge Agent 軟體會將來自車輛的命令回應發佈至此主題。 命令回應會以通訊協定緩衝區 (Protobuf) 格式交換。如需詳細資訊，請參閱 Edge Agent for AWS IoT FleetWise 軟體開發人員指南。	
$aws/iotfleetwise/vehicles/ *vehicleName* /command/notification	訂閱	AWS IoT FleetWise 會將命令狀態更新發佈至此主題。通知會以 JSON 格式傳送。	

主題	允許用戶端操作	描述	
$aws/iotfleetwise/vehicles/*$vehicle_name* /last_known_states/config	訂閱	AWS IoT FleetWise 會將狀態範本組態發佈至此主題。車輛會使用這些狀態範本組態。	
$aws/iotfleetwise/vehicles/*$vehicle_name* /last_known_states/data	發布	Edge Agent 軟體會將從訊號收集的資料發佈至此主題。	
$aws/iotfleetwise/vehicles/*$vehicle_name* /last_known_state/*$state_template_name* /data	訂閱	AWS IoT FleetWise 會從針對*$state_template_name* 本主題指定的 中設定的訊號發佈收集的資料。更新可以是部分更新。例如，如果狀態範本關聯包含具有變更中更新策略的多個訊號，則指定的訊息中只會包含已變更的訊號。 訊號資訊會以通訊協定緩衝區 (Protobuf) 格式交換。如需詳細資訊，請參閱 Edge Agent for AWS IoT FleetWise 軟體開發人員指南。	

建立 AWS IoT FleetWise 車輛

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

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

> **Important**
>
> 開始之前，請檢查下列項目：
>
> - 您必須擁有 車輛模型，且車輛模型的狀態必須為 `ACTIVE`。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛模型。
> - 您的 車輛模型必須與解碼器資訊清單相關聯，且解碼器資訊清單的狀態必須為 `ACTIVE`。如需詳細資訊，請參閱Manage AWS IoT FleetWise 解碼器資訊清單。

主題

- 建立車輛（主控台）
- 建立 車輛 (AWS CLI)

建立車輛（主控台）

您可以使用 AWS IoT FleetWise 主控台來建立車輛。

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：將政策連接至憑證
- 步驟 4：檢閱和建立

步驟 1：定義車輛屬性

在此步驟中，您會命名車輛，並將其與模型資訊清單和解碼器資訊清單建立關聯。

1. 輸入車輛的唯一名稱。

> **Important**
>
> 車輛對應至 AWS IoT 物件。如果已存在使用該名稱的物件，請選擇將車輛與 IoT 物件建立關聯，以更新該物件與車輛。或者，選擇不同的車輛名稱， AWS IoT FleetWise 會自動為車輛建立新的物件。

2. 從清單中選擇車輛模型（模型資訊清單）。
3. 從清單中選擇解碼器資訊清單。解碼器資訊清單與車輛模型相關聯。
4. （選用）若要關聯車輛屬性，請選擇新增屬性。如果您略過此步驟，您必須在建立車輛之後新增屬性，才能將其部署到行銷活動。
5. （選用）若要將標籤與車輛建立關聯，請選擇新增標籤。您也可以在建立車輛之後新增標籤。
6. 選擇下一步。

步驟 2：設定車輛憑證

若要使用 車輛做為 AWS IoT 物件，您必須設定已連接政策的車輛憑證。如果您略過此步驟，則必須在建立車輛之後設定憑證，才能將其部署到行銷活動。

1. 選擇自動產生新憑證（建議）。
2. 選擇下一步。

步驟 3：將政策連接至憑證

將政策連接至您在上一個步驟中設定的憑證。

1. 針對政策，輸入現有的政策名稱。若要建立新政策，請選擇建立政策。

2. 選擇下一步。

步驟 4：檢閱和建立

驗證車輛的組態，然後選擇建立車輛。

> **Important**
>
> 建立車輛後，您必須下載憑證和金鑰。您將使用憑證和私有金鑰，在 Edge Agent for AWS IoT FleetWise 軟體中連接車輛。

建立 車輛 (AWS CLI)

建立車輛時，您必須使用與解碼器資訊清單相關聯的車輛模型。您可以使用 CreateVehicle API 操作來建立車輛。下列為使用 AWS CLI的範例。

若要建立車輛，請執行下列命令。

將 *file-name* 取代為包含車輛組態的 .json 檔案的名稱。

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

Example – 車輛組態

- （選用）`associationBehavior`值可以是下列其中一項：
  - `CreateIotThing` – 建立車輛時， AWS IoT FleetWise 會自動為車輛建立具有車輛 ID 名稱的 AWS IoT 物件。
  - `ValidateIotThingExists` – 使用現有 AWS IoT 物件來建立車輛。

    若要建立 AWS IoT 物件，請執行下列命令。將 *thing-name* 取代為您要建立的物件名稱。

    ```
    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 AWS IoT 物件。

> **Important**
>
> 請確定在建立車輛後佈建 AWS IoT 物件。如需詳細資訊，請參閱Provision AWS IoT FleetWise 車輛。

- 以下列其中一項取代 *vehicle-name*。
  - 如果 `associationBehavior`設定為 ，您的 AWS IoT 物件名稱`ValidateIotThingExists`。
  - 如果 `associationBehavior` 設定為 ，則要建立的車輛 IDCreateIotThing。

    車輛 ID 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 *model-manifest-ARN* 取代為您車輛模型的 ARN （模型資訊清單）。
- 將 *decoder-manifest-ARN* 取代為與指定車輛模型相關聯之解碼器資訊清單的 ARN。
- （選用） 您可以新增其他屬性，以區分此車輛與從相同車輛模型建立的其他車輛。例如，如果您有電動汽車，您可以為 屬性指定下列值：`{"fuelType": "electric"}`。

> **Important**
>
> 屬性必須在相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"
    }
}
```

Example – 將狀態範本與車輛建立關聯

您可以使用 `stateTemplates` 欄位，將狀態範本與車輛建立關聯，以允許從雲端中的車輛收集狀態更新。

在此範例中， *stateTemplateUpdateStrategy* 可以是下列其中一項：

- `periodic`：可讓您指定 Edge Agent 軟體將訊號更新傳送到雲端的固定速率（即使訊號值未在更新之間變更，Edge Agent 軟體也會傳送更新）。
- `onChange`：Edge Agent 軟體會在訊號變更時傳送訊號更新。

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

`create-vehicle.json` 檔案包含的位置（例如）：

```
{
    "associationBehavior": "associationBehavior",
    "vehicleName": "vehicle-name",
    "modelManifestArn": "model-manifest-ARN",
    "decoderManifestArn": "decoder-manifest-ARN",
    "attributes": {
        "key": "value"
    },
    "stateTemplates": [
        {
            "identifier": "state-template-name",
            "stateTemplateUpdateStrategy": {
                "periodic": {
                    "stateTemplateUpdateRate": {
                        "unit": "SECOND",
                        "value": 10
                    }
                }
            }
        }
    ]
}
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateVehicle` API 操作。

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

```
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

建立多個 AWS IoT FleetWise 車輛

您可以使用 BatchCreateVehicle API 操作一次建立多部車輛。下列為使用 AWS CLI的範例。

若要建立多部車輛，請執行下列命令。

將 *file-name* 取代為包含多部車輛組態的 .json 檔案名稱。

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

Example – 車輛組態

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

```
}
```

您可以為每個批次操作建立最多 10 部車輛。如需車輛組態的詳細資訊，請參閱 建立 AWS IoT FleetWise 車輛。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `BatchCreateVehicle` API 操作。

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

更新 an AWS IoT FleetWise 車輛

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

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，請執行下列命令。

將 *file-name* 取代為包含 車輛組態的 .json 檔案的名稱。

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

Example – 車輛組態

- 以您要更新的車輛 ID 取代 *vehicle-name*。
- （選用）將 *model-manifest-ARN* 取代為您用來取代使用中車輛模型之車輛模型（模型資訊清單）的 ARN。
- （選用）將 *decoder-manifest-ARN* 取代為您指定的新車輛模型相關聯的解碼器資訊清單 ARN。
- （選用）將 *attribute-update-mode* 取代為 車輛屬性。
  - `Merge` – 使用新值更新現有屬性，並在新屬性不存在時新增屬性，以將新屬性合併至現有屬性。

    例如，如果車輛具有下列屬性：`{"color": "black", "fuelType": "electric"}`，而您使用下列屬性更新車輛：`{"color": "", "fuelType": "gasoline", "model": "x"}`，則更新的車輛具有下列屬性：`{"fuelType": "gasoline", "model": "x"}`。
  - `Overwrite` – 將現有屬性取代為新屬性。

    例如，如果車輛具有下列屬性：`{"color": "black", "fuelType": "electric"}`，而您使用 `{"model": "x"}` 屬性更新車輛，則更新的 車輛具有 `{"model": "x"}` 屬性。

  如果輸入中存在屬性，則此為必要項目。
- （選用）若要新增屬性或使用新值更新現有屬性，請設定 `attributes`。例如，如果您有電動汽車，您可以為 屬性指定下列值：`{"fuelType": "electric"}`。

  若要刪除屬性，請將 `attributeUpdateMode`設定為 `Merge`。

> **Important**
>
> 屬性必須在相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"
        }
}
```

Example – 新增或移除與車輛相關聯的狀態範本

您可以使用下列欄位建立其他狀態範本的關聯，或從車輛中移除現有的關聯：

- `stateTemplatesToAdd`
- `stateTemplatesToRemove`

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

update-vehicle.json 檔案包含的位置（例如）：

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `UpdateVehicle` API 操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
```

```
      ]
    },
  ]
}
```

更新多個 AWS IoT FleetWise 車輛

您可以使用 BatchUpdateVehicle API 操作一次更新多個現有車輛。下列為使用 AWS CLI的範例。

若要更新多部車輛，請執行下列命令。

將 *file-name* 取代為包含多部車輛組態的 .json 檔案名稱。

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

Example – 車輛組態

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

每個批次操作最多可以更新 10 部車輛。如需各車輛組態的詳細資訊，請參閱 更新 an AWS IoT FleetWise 車輛。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `BatchUpdateVehicle` API 操作。

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

刪除 AWS IoT FleetWise 車輛

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

> **Important**
>
> 刪除車輛後， AWS IoT FleetWise 會自動從相關聯的機群和行銷活動中移除車輛。如需詳細資訊，請參閱在 AWS IoT FleetWise 中管理機群及使用行銷活動收集 AWS IoT FleetWise 資料。不過，車輛仍以物件形式存在，或仍與物件相關聯 AWS IoT Core。如需刪除物件的指示，請參閱《 開發人員指南》中的刪除物件。 AWS IoT Core

刪除車輛 （主控台）

您可以使用 AWS IoT FleetWise 主控台來刪除車輛。

刪除車輛

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛。

3. 在 車輛頁面上，選取您要刪除之車輛旁的按鈕。
4. 選擇 刪除。
5. 在刪除 **vehicle-name**中，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 操作來刪除車輛。下列範例使用 AWS CLI。

若要刪除車輛，請執行下列命令。

將 *vehicle-name* 取代為您要刪除之車輛的 ID。

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

驗證車輛刪除

您可以使用 ListVehicles API 操作來驗證是否已刪除車輛。下列為使用 AWS CLI的範例。

若要擷取所有車輛的分頁清單，請執行下列命令。

```
aws iotfleetwise list-vehicles
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListVehicles` API 操作。

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

Get AWS IoT FleetWise 車輛資訊

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 GetVehicle API 操作來擷取車輛資訊。下列為使用 AWS CLI的範例。

若要擷取車輛的中繼資料，請執行下列命令。

將 *vehicle-name* 取代為您要擷取之車輛的 ID。

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

> **Note**
>
> 此操作為最終一致。換句話說，可能不會立即反映對車輛的變更。

您可以使用 GetVehicleStatus API 操作來擷取與車輛相關聯的資源狀態。下列為使用 AWS CLI的範例。

若要擷取與車輛相關聯的資源狀態，請執行下列命令。

- 將 *vehicle-name* 取代為與資源相關聯的 車輛 ID。
- 將*##*取代為您要擷取其狀態的資源類型。`type` 的有效值為 `CAMPAIGN`、`STATE_TEMPLATE` 和 `DECODER`。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetVehicle`或 `GetVehicleStatus` API 操作。

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

```
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

在 AWS IoT FleetWise 中管理機群

機群代表一組車輛。沒有相關聯車輛的機群是空的實體。您必須先將車輛與機群建立關聯，才能使用機群來同時管理多部車輛。車輛可以屬於多個機群。您可以藉由部署行銷活動，控制要從車輛機群收集哪些資料，以及何時收集資料。如需詳細資訊，請參閱使用行銷活動收集 AWS IoT FleetWise 資料。

機群包含下列資訊。

`fleetId`

機群的 ID。

(選用) `description`

協助您尋找機群的描述。

`signalCatalogArn`

訊號目錄的 Amazon Resource Name (ARN)。

AWS IoT FleetWise 提供下列 API 操作，您可以用來建立和管理機群。

- CreateFleet – 建立包含相同訊號群組的車輛群組。
- AssociateVehicleFleet – 將車輛與機群建立關聯。
- DisassociateVehicleFleet – 取消車輛與機群的關聯。
- UpdateFleet – 更新現有機群的描述。
- DeleteFleet – 刪除現有的機群。
- ListFleets – 擷取所有機群的分頁清單。
- ListFleetsForVehicle – 擷取車輛所屬所有機群的分頁 IDs 清單。
- ListVehiclesInFleet – 擷取機群中所有車輛的分頁清單。
- GetFleet – 擷取機群的相關資訊。

主題

- 建立 AWS IoT FleetWise 機群
- 將 AWS IoT FleetWise 車輛與機群建立關聯
- 取消 AWS IoT FleetWise 車輛與機群的關聯
- 更新 AWS IoT FleetWise 機群

- 刪除 AWS IoT FleetWise 機群
- Get AWS IoT FleetWise 機群資訊

建立 AWS IoT FleetWise 機群

您可以使用 CreateFleet API 操作來建立機群。下列範例使用 AWS CLI。

> **Important**
>
> 您必須先擁有訊號目錄，才能建立機群。如需詳細資訊，請參閱建立 AWS IoT FleetWise 訊號目錄。

若要建立機群，請執行下列命令。

- 將 *fleet-id* 取代為您建立之機群的 ID。

  機群 ID 必須是唯一的，且具有 1-100 個字元。有效字元：字母 (A-Z 和 a-z)、數字 (0-9)、冒號 (：)、破折號 (-) 和底線 (_)。
- （選用）將##取代為描述。

  描述可以有 1-2048 個字元。
- 將 *signal-catalog-arn* 取代為訊號目錄的 ARN。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateFleet` API 操作。

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

```
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

將 AWS IoT FleetWise 車輛與機群建立關聯

您可以使用 AssociateVehicleFleet API 操作，將車輛與機群建立關聯。下列範例使用 AWS CLI。

> **Important**
>
> - 您必須先擁有 車輛和機群，才能將車輛與機群建立關聯。如需詳細資訊，請參閱Manage AWS IoT FleetWise 車輛。
> - 如果您將車輛與行銷活動目標機群建立關聯， AWS IoT FleetWise 會自動將行銷活動部署到車輛。

若要將車輛與機群建立關聯，請執行下列命令。

- 將 *fleet-id* 取代為機群的 ID。
- 將 *vehicle-name* 取代為車輛的 ID。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `AssociateVehicleFleet` API 操作。

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

```
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

取消 AWS IoT FleetWise 車輛與機群的關聯

您可以使用 DisassociateVehicleFleet API 操作來取消車輛與機群的關聯。下列範例使用 AWS CLI。

若要取消車輛與機群的關聯，請執行下列命令。

- 將 *fleet-id* 取代為機群的 ID。
- 將 *vehicle-name* 取代為車輛的 ID。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `DisassociateVehicleFleet` API 操作。

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

```
}
```

更新 AWS IoT FleetWise 機群

您可以使用 UpdateFleet API 操作來更新機群的描述。以下範例使用 AWS CLI。

若要更新機群，請執行下列命令。

- 將 *fleet-id* 取代為您更新之機群的 ID。
- 將*##*取代為新的描述。

  描述可以有 1-2048 個字元。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `UpdateFleet` API 操作。

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

刪除 AWS IoT FleetWise 機群

您可以使用 DeleteFleet API 操作來刪除機群。下列範例使用 AWS CLI。

> **Important**
>
> 刪除機群之前，請確定它沒有相關聯的車輛。如需如何取消車輛與機群的關聯的指示，請參閱取消 AWS IoT FleetWise 車輛與機群的關聯。

若要刪除機群，請執行下列命令。

將 *fleet-id* 取代為您要刪除之機群的 ID。

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

驗證機群刪除

您可以使用 ListFleets API 操作來驗證機群是否已刪除。下列為使用 AWS CLI的範例。

若要擷取所有機群的分頁清單，請執行下列命令。

```
aws iotfleetwise list-fleets
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListFleets` API 操作。

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

Get AWS IoT FleetWise 機群資訊

您可以使用 ListFleetsForVehicle API 操作來擷取車輛所屬所有機群的分頁 IDs 清單。下列為使用 AWS CLI的範例。

若要擷取車輛所屬之所有機群的分頁 IDs 清單，請執行下列命令。

以*### ID ## vehicle-name*。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListFleetsForVehicle` API 操作。

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

您可以使用 ListVehiclesInFleet API 操作來擷取機群中所有車輛的分頁清單。下列為使用 AWS CLI的範例。

若要擷取機群中所有車輛的分頁清單，請執行下列命令。

將 *fleet-id* 取代為機群的 ID。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `ListVehiclesInFleet` API 操作。

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

您可以使用 GetFleet API 操作來擷取機群資訊。下列為使用 AWS CLI的範例。

若要擷取機群的中繼資料，請執行下列命令。

將 *fleet-id* 取代為機群的 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的變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetFleet` API 操作。

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

```
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

使用行銷活動收集 AWS IoT FleetWise 資料

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

行銷活動是資料收集規則的協調。行銷活動會提供 Edge Agent for AWS IoT FleetWise 軟體指示，說明如何選取、收集資料並將資料傳輸至雲端。

您可以在雲端中建立活動。在您或您的團隊核准行銷活動後， AWS IoT FleetWise 會將行銷活動設定為準備好部署，並在下次車輛進廠時部署。您可以選擇將行銷活動部署至車輛或機群。Edge Agent 軟體不會開始收集資料，直到正在執行的行銷活動部署到車輛為止。

> **Important**
>
> 除非您有以下項目，否則行銷活動將無法運作。
>
> - Edge Agent 軟體正在您的 車輛中執行。如需如何開發、安裝和使用 Edge Agent 軟體的詳細資訊，請執行下列動作。
> 1. 開啟 AWS IoT FleetWise 主控台。
> 2. 在服務首頁的開始使用 AWS IoT FleetWise 區段中，選擇探索邊緣代理程式。
> - 您已設定 AWS IoT Core 來佈建您的 車輛。如需詳細資訊，請參閱Provision AWS IoT FleetWise 車輛。

> **Note**
>
> 您也可以 監控您車輛的最後已知狀態（而非機群） 近乎即時地使用狀態範本，可讓您使用「變更中」或「定期」更新策略串流遙測資料。此功能也提供「隨需」功能，以啟用或停用先前部署的範本，或一次性請求目前的車輛狀態 （擷取）。
> 目前已封鎖對最後一個已知狀態的存取。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

每個行銷活動都包含下列資訊。

`signalCatalogArn`

與行銷活動相關聯之訊號目錄的 Amazon Resource Name (ARN)。

(選用) `tags`

標籤是可用於管理行銷活動的中繼資料。您可以將相同的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`

有助於識別行銷活動的唯一名稱。

`collectionScheme`

資料收集方案為 Edge Agent 軟體提供有關要收集哪些資料或何時收集資料的說明。 AWS IoT FleetWise 目前支援以條件為基礎的收集方案和以時間為基礎的收集方案。

- `conditionBasedCollectionScheme` – 條件型收集機制使用邏輯表達式來識別要收集的資料。Edge Agent 軟體會在滿足條件時收集資料。
  - `expression` – 用來辨識要收集哪些資料的邏輯表達式。例如，如果指定`$variable.`myVehicle.InVehicleTemperature` > 50.0`表達式，Edge Agent 軟體會收集大於 50.0 的溫度值。如需如何撰寫表達式的說明，請參閱 AWS IoT FleetWise 行銷活動的邏輯表達式。
  - （選用）`conditionLanguageVersion` – 條件式表達式語言的版本。
  - （選用）`minimumTriggerIntervalMs` – 兩個資料收集事件之間的最短持續時間，以毫秒為單位。如果訊號經常變更，您可能會以較慢的速度收集資料。
  - （選用）`triggerMode` – 可以是下列其中一個值：
    - `RISING_EDGE` – Edge Agent 軟體只會在第一次符合條件時收集資料。例如 `$variable.`myVehicle.AirBagDeployed` == true`。
    - `ALWAYS` – Edge Agent 軟體會在符合條件時收集資料。
- `timeBasedCollectionScheme` – 當您定義以時間為基礎的收集方案時，請以毫秒為單位指定時段。Edge Agent 軟體會使用期間來決定收集資料的頻率。例如，如果時段為 120，000 毫秒，Edge Agent 軟體會每兩分鐘收集資料一次。
  - `periodMs` – 決定收集資料頻率的期間（以毫秒為單位）。

(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。

根據預設 (`OFF`)，Edge Agent 軟體不會壓縮資料。

`dataDestinationConfigs`

選擇行銷活動將傳輸車輛資料的單一目的地。您可以將資料傳送至 MQTT 主題，或將其存放在 Amazon S3 或 Amazon Timestream 中。

MQTT （訊息佇列遙測傳輸） 是一種輕量且廣泛採用的訊息通訊協定。您可以將資料發佈至 MQTT 主題，以使用 AWS IoT rule. AWS IoT support for MQTT 以 MQTT v3.1.1 規格和 MQTT v5.0 規格為基礎，但有一些差異。如需詳細資訊，請參閱 MQTT 差異。

S3 可以是經濟實惠的資料儲存機制，可提供持久的資料管理功能和下游資料服務。您可以使用 S3 處理與駕駛行為或分析長期維護相關的資料。

Timestream 是一種資料持久性機制，可協助您近乎即時地識別趨勢和模式。您可以使用 Timestream 處理時間序列資料，例如分析車輛速度或煞車的歷史趨勢。

> **Note**
>
> Amazon Timestream 不適用於亞太區域 （孟買） 區域。

(選用) `dataExtraDimensions`

您可以新增一或多個屬性，以提供訊號的其他資訊。

(選用) `dataPartitions`

建立資料分割區以暫時將訊號資料存放在車輛上。您可以設定將資料轉送至雲端的時間和方式。

- 透過定義最大儲存大小、最短存留時間和儲存位置，指定 AWS IoT FleetWise 如何在車輛或機群上儲存資料。
- 行銷活動`spoolingMode`必須為 `TO_DISK`。
- 上傳組態包括定義條件語言的版本和邏輯表達式。

(選用) `description`

新增描述以協助識別行銷活動的目的。

(選用) `diagnosticsMode`

當診斷模式設定為 時`SEND_ACTIVE_DTCS`，行銷活動會傳送已儲存的標準診斷故障碼 (DTCs)，以協助識別您車輛的問題。例如，P0097 表示引擎控制模組 (ECM) 已判斷吸入空氣溫度感應器 2 (IAT2) 輸入低於正常感應器範圍。

根據預設 (`OFF`)，Edge Agent 軟體不會傳送診斷代碼。

(選用) `expiryTime`

定義行銷活動的過期日期。當行銷活動到期時，Edge Agent 軟體會停止收集此行銷活動中指定的資料。如果車輛部署了多個行銷活動，Edge Agent 軟體會使用其他行銷活動來收集資料。

預設值：`253402243200`(9999 年 12 月 31 日 00：00：00 UTC)

(選用) `postTriggerCollectionDuration`

您可以定義觸發後收集持續時間，以便 Edge Agent 軟體在調用方案後繼續收集資料一段指定的期間。例如，如果調用具有下列表達式的條件型集合結構描述：`$variable.`myVehicle.Engine.RPM` > 7000.0`，則 Edge Agent 軟體會繼續為引擎收集每分鐘轉數 (RPM) 值。即使 RPM 僅高於 7000 次，也可能表示存在機械問題。在這種情況下，您可能希望 Edge Agent 軟體繼續收集資料，以協助監控條件。

預設值：`0`

(選用) `priority`

指定整數以指出行銷活動的優先順序層級。數量較小的行銷活動具有較高的優先順序。如果您將多個行銷活動部署到車輛，則會先啟動優先順序較高的行銷活動。

預設值：`0`

(選用) `signalsToCollect`

叫用資料收集方案時從中收集資料的訊號清單。

- `name` – 叫用資料收集結構描述時從中收集資料的訊號名稱。
- `dataPartitionId` – 要在訊號中使用的資料分割區 ID。ID 必須符合 中提供的其中一個 IDs`dataPartitions`。如果您上傳訊號做為資料分割區中的條件，則這些相同的訊號必須包含在 中`signalsToCollect`。
- （選用） `maxSampleCount` – Edge Agent 軟體在叫用資料收集方案時收集並傳輸到雲端的資料樣本數量上限。

- （選用）`minimumSamplingIntervalMs` – 兩個資料範例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度收集資料。

  有效範圍：0-4294967295

(選用) `spoolingMode`

如果 `spoolingMode` 設定為 `TO_DISK`，則 Edge Agent 軟體會在車輛未連線至雲端時暫時將資料存放在本機。重新建立連線後，本機儲存的資料會自動傳輸至雲端。

預設值：`OFF`

(選用) `startTime`

核准的行銷活動會在開始時間啟動。

預設值：`0`

行銷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建立行銷活動。
- `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 操作。行銷活動獲得核准後， AWS IoT FleetWise 會自動將行銷活動部署到目標車輛或機群。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 行銷活動。
- `RUNNING` – 行銷活動為作用中。
- `SUSPENDED` – 行銷活動已暫停。若要繼續行銷活動，請使用 `UpdateCampaign` API 操作。

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

- CreateCampaign – 建立新的行銷活動。
- UpdateCampaign – 更新現有的行銷活動。建立行銷活動後，您必須使用此 API 操作來核准行銷活動。
- DeleteCampaign – 刪除現有的行銷活動。
- ListCampaigns – 擷取所有行銷活動的分頁摘要清單。
- GetCampaign – 擷取行銷活動的相關資訊。

教學課程

- 建立 an AWS IoT FleetWise 行銷活動
- 更新 an AWS IoT FleetWise 行銷活動
- 刪除 an AWS IoT FleetWise 行銷活動
- Get AWS IoT FleetWise 行銷活動資訊
- 儲存和轉送行銷活動資料
- 使用 AWS IoT FleetWise 收集診斷問題碼資料
- Visualize AWS IoT FleetWise 車輛資料

建立 an AWS IoT FleetWise 行銷活動

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 AWS IoT FleetWise 主控台或 API 建立行銷活動來收集車輛資料。

> **Important**
>
> 若要讓您的行銷活動運作，您必須具備下列項目：
>
> - Edge Agent 軟體正在您的 車輛中執行。如需如何開發、安裝和使用 Edge Agent 軟體的詳細資訊，請執行下列動作：
> 1. 開啟 AWS IoT FleetWise 主控台。
> 2. 在服務首頁的開始使用 AWS IoT FleetWise 區段中，選擇探索邊緣代理程式。
> - 您已設定 AWS IoT Core 來佈建您的 車輛。如需詳細資訊，請參閱Provision AWS IoT FleetWise 車輛。

主題

- 建立行銷活動（主控台）
- 建立行銷活動 (AWS CLI)
- AWS IoT FleetWise 行銷活動的邏輯表達式

建立行銷活動（主控台）

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至雲端。

建立行銷活動

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇行銷活動。
3. 在行銷活動頁面上，選擇建立行銷活動，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：指定儲存和上傳條件
- 步驟 3：設定資料目的地
- 步驟 4：新增 車輛
- 步驟 4：檢閱和建立
- 步驟 6：部署行銷活動

> **Important**
>
> - 建立行銷活動之前，您必須擁有訊號目錄和車輛。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄及Manage AWS IoT FleetWise 車輛。
> - 建立行銷活動後，您必須核准行銷活動。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 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. 輸入行銷活動的名稱。
2. (選用) 輸入描述。

設定行銷活動的資料收集方案。資料收集方案為 Edge Agent 軟體提供有關要收集哪些資料或何時收集資料的指令。在 AWS IoT FleetWise 主控台中，您可以透過下列方式設定資料收集方案：

- 手動定義資料收集結構描述。
- 上傳檔案以自動定義資料收集方案。

在組態選項中，選擇下列其中一項：

- 若要手動指定資料收集結構描述的類型並定義自訂結構描述的選項，請選擇定義資料收集結構描述。

  手動指定資料收集結構描述的類型，並定義自訂結構描述的選項。

  1. 在資料收集方案詳細資訊區段中，選擇您希望此行銷活動使用的資料收集方案類型。若要使用邏輯表達式來辨識要收集的車輛資料，請選擇條件型。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. 定義行銷活動收集資料的持續時間。

     > **Note**
     > 根據預設，核准的行銷活動會立即啟用，而且沒有設定的結束時間。若要避免額外費用，您必須指定時間範圍。

  3. 如果您指定條件式資料收集方案，則必須定義邏輯表達式來識別要收集哪些資料。 AWS IoT FleetWise 使用邏輯表達式來識別條件式方案要收集哪些資料。表達式必須將訊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`表達式，AWS IoT FleetWise 會收集大於 50.0 的溫度值。如需如何撰寫表達式的說明，請參閱 AWS IoT FleetWise 行銷活動的邏輯表達式。

     輸入用來辨識要收集哪些資料的邏輯表達式。
  4. （選用）指定條件式表達式的語言版本。預設值為 1。
  5. （選用）指定最短觸發間隔，這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度收集資料。
  6. 指定 Edge Agent 軟體收集資料的觸發模式條件。根據預設，Edge Agent for AWS IoT FleetWise 軟體一律會在符合條件時收集資料。或者，它只能在第一次符合條件時收集資料，即第一次觸發時。
  7. 如果您指定以時間為基礎的資料收集方案，則必須指定從 10，000 - 60，000 毫秒的時段，以毫秒為單位。Edge Agent 軟體會使用期間來決定收集資料的頻率。
  8. （選用）編輯方案的進階方案選項。

a. 若要透過壓縮資料來節省無線頻寬並減少網路流量，請選擇 Snappy。

b. （選用）若要定義資料收集事件之後繼續收集資料的時間長度，您可以指定觸發後收集持續時間。

c. （選用）若要指出行銷活動的優先順序，請指定行銷活動優先順序。優先順序較小的行銷活動會先部署，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 軟體可以在本機暫時存放資料。重新建立連線後，本機儲存的資料會自動傳輸至雲端。指定您是否希望 Edge 代理程式在連線中斷期間將資料儲存在本機。

e. （選用）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方案，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。

  上傳包含資料收集方案詳細資訊的 .json 檔案。

  1. 若要匯入資料收集方案的相關資訊，請選擇選擇檔案。如需所需檔案格式的詳細資訊，請參閱 CreateCampaign API 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 FleetWise 為您選取的選項。如有需要，您可以更新選項。

步驟 2：指定儲存和上傳條件

若要選擇 Edge Agent 軟體是否在車輛未連線至雲端時暫時將資料存放在本機，請指定多工緩衝處理模式。

- 在資料多工緩衝處理模式中，選擇下列其中一項：
  - 未儲存 – Edge Agent 軟體會收集資料，但不會在車輛離線時暫時將資料存放在本機。當車輛重新連線時，Edge Agent 軟體會將資料傳輸到雲端。
  - 存放至磁碟 – Edge Agent 軟體會在車輛離線時收集並暫時將資料存放於本機。收集的資料會暫時存放在 Edge Agent 組態檔案「持久性」區段所定義的位置。當車輛重新連線時，Edge Agent 會將資料傳輸到雲端。

- 存放到具有分割區的磁碟 – 車輛一律會將資料暫時存放在您指定資料分割區的 Edge 上。您可以選擇何時將儲存的資料轉送至雲端。
  1. （選用）輸入分割區 ID 以指定特定資料集。
  2. 輸入資料夾名稱做為將存放資料的位置。儲存位置的絕對路徑為 `{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`。
  3. 輸入存放在分割區中資料的儲存大小上限。當分割區達到大小上限時，較新的資料會覆寫較舊的資料。
  4. 輸入此分割區中的資料將保留在磁碟上的最短時間。
  5. （選用）輸入分割區的上傳條件。

指定訊號

您可以指定要在行銷活動期間從 收集資料的訊號。

指定要從 收集資料的訊號

1. 選取訊號名稱。
2. （選用）針對最大範例計數，輸入 Edge Agent 軟體在行銷活動期間收集並傳輸至雲端的資料範例數目上限。
3. （選用）對於最短取樣間隔，輸入兩個資料範例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度收集資料。
4. 若要新增另一個訊號，請選擇新增更多訊號。您最多可以新增 999 個訊號。
5. 選擇下一步。

步驟 3：設定資料目的地

> **Note**
>
> 如果行銷活動包含視覺系統資料訊號，您只能將車輛資料存放在 Amazon S3 中。您無法將其存放在 Timestream 中，或將其傳送至 MQTT 主題。
> 視覺系統資料處於預覽版本中，可能會有所變更。
> Amazon Timestream 不適用於亞太區域（孟買）區域。

選擇您要傳送或存放行銷活動所收集資料的目的地。您可以將車輛資料傳送至 MQTT 主題，或將其存放在 Amazon S3 或 Amazon Timestream 中。

在目的地設定中，執行下列動作：

- 從下拉式清單中選擇 Amazon S3、Amazon Timestream 或 MQTT 主題。

Amazon S3

> **Important**
>
> 如果 AWS IoT FleetWise 具有寫入 S3 儲存貯體的許可，您只能將資料傳輸至 S3。如需授予存取權的詳細資訊，請參閱使用 AWS IoT FleetWise 控制存取權。

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，可將資料儲存為儲存貯體中的物件。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的建立、設定和使用 Amazon S3 儲存貯體。

S3 可最佳化資料儲存的成本，並提供其他機制來使用車輛資料，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 來存放資料以進行批次處理和分析。例如，您可以為機器學習 (ML) 模型建立硬中斷事件的報告。傳入的車輛資料會在交付前緩衝 10 分鐘。

在 S3 目的地設定中，執行下列動作：

1. 針對 S3 儲存貯體，選擇具有 許可的儲存 AWS IoT FleetWise 貯體。
2. （選用）輸入自訂字首，可用來組織存放在 S3 儲存貯體中的資料。
3. 選擇輸出格式，這是儲存為 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。我們建議您壓縮資料，因為它可將儲存成本降至最低。
5. 您在 S3 目的地設定中選取的選項會變更範例 S3 物件 URI。這是在 S3 中儲存為 的檔案的範例。

Amazon Timestream

> **Important**
>
> 只有在 AWS IoT FleetWise 具有將資料寫入 Timestream 的許可時，您才能將資料傳輸至資料表。如需授予存取權的詳細資訊，請參閱使用 AWS IoT FleetWise 控制存取權。

Amazon Timestream 不適用於亞太區域（孟買）區域。

若要將車輛資料儲存在 Timestream 資料表中，請選擇 Amazon Timestream。您可以使用 Timestream 查詢車輛資料，以便識別趨勢和模式。例如，您可以使用 Timestream 為車輛油位建立警示。傳入的車輛資料會以近乎即時的方式傳輸至 Timestream。如需詳細資訊，請參閱《Amazon Timestream 開發人員指南》中的什麼是 Amazon Timestream？。

在 Timestream 資料表設定中，執行下列動作：

1. 對於 Timestream 資料庫名稱，從下拉式清單中選擇 Timestream 資料庫的名稱。
2. 對於 Timestream 資料表名稱，從下拉式清單中選擇 Timestream 資料表的名稱。

在 Timestream 的服務存取中，執行下列動作：

- 從下拉式清單中選擇 IAM 角色。

MQTT 主題

> **Important**
>
> 如果 AWS IoT FleetWise 具有主題的許可，您只能將資料路由到 MQTT AWS IoT 主題。如需授予存取權的詳細資訊，請參閱使用 AWS IoT FleetWise 控制存取權。

若要將車輛資料傳送至 MQTT 主題，請選擇 MQTT 主題。

MQTT 訊息傳送的車輛資料會以近乎即時的方式交付，並可讓您使用規則來採取動作，或將資料路由至其他目的地。如需使用 MQTT 的詳細資訊，請參閱《 AWS IoT Core 開發人員指南》中的 裝置通訊協定和 的規則 AWS IoT。

1. 在 MQTT 主題下，輸入主題名稱。
2. 在 MQTT 服務存取主題下，選擇您是否要讓 AWS IoT FleetWise Create 為您使用新的服務角色。如果您想要使用現有的服務角色，請在選取角色下的下拉式清單中選取角色。

- 選擇下一步。

步驟 4：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。透過搜尋您在建立車輛時新增的屬性及其值，或依車輛名稱來篩選車輛。

在篩選車輛中，執行下列動作：

1. 在搜尋方塊中，尋找屬性或車輛名稱，然後從清單中選擇。

   > **Note**
   > 每個屬性只能使用一次。

2. 輸入您要部署行銷活動的屬性或車輛名稱的值。例如，如果屬性的完整名稱是 `fuelType`，請輸入 `gasoline`做為其值。
3. 若要搜尋其他車輛屬性，請重複上述步驟。您可以搜尋最多五個車輛屬性和無限數量的車輛名稱。
4. 符合您搜尋的車輛會列在車輛名稱下。選擇您要部署行銷活動的車輛。

   > **Note**
   > 搜尋結果中最多可顯示 100 部車輛。選擇全選，將所有車輛新增至行銷活動。

5. 選擇下一步。

步驟 4：檢閱和建立

驗證行銷活動的組態，然後選擇建立行銷活動。

> **Note**
> 建立行銷活動後，您或您的團隊必須將行銷活動部署到 車輛。

步驟 6：部署行銷活動

建立行銷活動後，您或您的團隊必須將行銷活動部署到 車輛。

部署行銷活動

1. 在行銷活動摘要頁面上，選擇部署。
2. 檢閱並確認您想要開始部署，並開始從連線至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線至行銷活動的車輛收集資料，請在行銷活動摘要頁面上選擇暫停。若要繼續從連線至行銷活動的車輛收集資料，請選擇繼續。

建立行銷活動 (AWS CLI)

您可以使用 CreateCampaign API 操作來建立行銷活動。下列為使用 AWS CLI的範例。

當您建立行銷活動時，從車輛收集的資料可以傳送到 MQTT 主題，或存放在 Amazon S3 (S3) 或 Amazon Timestream 中。針對快速、可擴展且無伺服器的時間序列資料庫選擇 Timestream，例如存放需要近乎即時處理的資料。針對具有業界領先可擴展性、資料可用性、安全性和效能的物件儲存選擇 S3。選擇 MQTT 以近乎即時的方式交付資料，並使用 的規則 AWS IoT來執行您定義或路由資料至其他目的地的動作。

> **Important**
>
> 如果 AWS IoT FleetWise 具有代表您傳送 MQTT 訊息的許可，或將資料寫入 SAmazon S3 或 Timestream，您只能將車輛資料傳輸至 MQTT 主題、Amazon S3 或 Amazon Timestream。S3 如需授予存取權的詳細資訊，請參閱使用 AWS IoT FleetWise 控制存取權。
> Amazon Timestream 不適用於亞太區域（孟買）區域。

建立行銷活動

> **Important**
>
> - 建立行銷活動之前，您必須擁有訊號目錄和車輛或機群。如需詳細資訊，請參閱Manage AWS IoT FleetWise 訊號目錄、Manage AWS IoT FleetWise 車輛及在 AWS IoT FleetWise 中管理機群。
> - 建立行銷活動後，您必須使用 `UpdateCampaign` API 操作來核准行銷活動。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 行銷活動

若要建立行銷活動，請執行下列命令。

將 *file-name* 取代為包含行銷活動組態的 .json 檔案名稱。

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

- 將 *campaign-name* 取代為您建立的行銷活動名稱。
- 將 *signal-catalog-arn* 取代為訊號目錄的 Amazon Resource Name (ARN)。
- 將 *target-arn* 取代為您建立的機群或車輛的 ARN。
- 將 *bucket-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"
      }
    }
  ],
     "dataPartitions": [
      { ...  }
  ]
}
```

> **Note**
>
> Amazon Timestream 不適用於亞太區域（孟買）區域。

- 將 *campaign-name* 取代為您建立的行銷活動名稱。
- 將 *signal-catalog-arn* 取代為訊號目錄的 ARN。
- 將 *target-arn* 取代為您建立的機群或車輛的 ARN。
- 將 *role-arn* 取代為任務執行角色的 ARN，該角色授予 AWS IoT FleetWise 將資料交付至 Timestream 資料表的許可。
- 將 *table-arn* 取代為 Timestream 資料表的 ARN。

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

```
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
    {
      "timestreamConfig": {
        "executionRoleArn": "role-arn",
        "timestreamTableArn": "table-arn"
      }
    }
  ],
   "dataPartitions": [
      { ...  }
  ]
}
```

- 將 *campaign-name* 取代為您建立的行銷活動名稱。
- 將 *signal-catalog-arn* 取代為訊號目錄的 Amazon Resource Name (ARN)。
- 將 *target-arn* 取代為您建立的機群或車輛的 ARN。
- 將 *topic-arn* 取代為您指定為包含車輛資料之訊息目的地的 MQTT 主題 ARN。
- 以授予 AWS IoT FleetWise 許可的任務執行角色 ARN 取代 *role-arn*，以傳送、接收您指定的 MQTT 主題的訊息並對其採取動作。

```
{
  "name": "campaign-name",
  "targetArn": "target-arn",
```

```
  "signalCatalogArn": "signal-catalog-arn",
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
      {
          "mqttTopicConfig": {
              "mqttTopicArn": "topic-arn",
              "executionRoleArn": "role-arn"
          }
      }
  ]
}
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `CreateCampaign` API 操作。

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

```
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

AWS IoT FleetWise 行銷活動的邏輯表達式

AWS IoT FleetWise 使用邏輯表達式來識別要收集哪些資料作為行銷活動的一部分。如需表達式的詳細資訊，請參閱《 AWS IoT Events 開發人員指南》中的表達式。

運算式變數的建構應符合所收集資料類型的規則。對於遙測系統資料，表達式變數應該是訊號的完整名稱。對於視覺系統資料，表達式結合了訊號的完整名稱，以及從訊號資料類型到其中一個屬性的路徑。

例如，如果訊號目錄包含下列節點：

```
{
    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 事件表達式變數：

```
{
...
    $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 an AWS IoT FleetWise 行銷活動

您可以使用 UpdateCampaign API 操作來更新現有的行銷活動。使用下列命令 AWS CLI。

- 將 *campaign-name* 取代為您更新之行銷活動的名稱。
- 將##取代為下列其中一項：
  - APPROVE – 核准行銷活動，以允許 AWS IoT FleetWise 將其部署到車輛或機群。
  - SUSPEND – 暫停行銷活動。行銷活動會從車輛中刪除，暫停行銷活動中的所有車輛都會停止傳送資料。
  - RESUME – 重新啟用SUSPEND行銷活動。行銷活動已設定為在下次入住時重新部署至所有車輛，且車輛會繼續傳送資料。
  - UPDATE – 透過定義屬性並將其與行銷活動建立關聯來更新行銷活動。
- 將##取代為新的描述。

描述最多可有 2，048 個字元。

- 將 *data-extra-dimensions* 取代為指定的車輛屬性，以豐富行銷活動期間收集的資料。例如，您可以將車輛品牌和模型新增至行銷活動， AWS IoT FleetWise 會將資料與這些屬性建立關聯，做為 Amazon Timestream 中的維度。然後，您可以針對車輛品牌和模型來查詢資料。

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

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `UpdateCampaign` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

刪除 an AWS IoT FleetWise 行銷活動

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

刪除行銷活動（主控台）

若要刪除行銷活動，請使用 AWS IoT FleetWise 主控台。

刪除行銷活動

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇行銷活動。
3. 在行銷活動頁面上，選擇目標行銷活動。
4. 選擇 刪除。
5. 在刪除 **campaign-name**？ 中，輸入要刪除的行銷活動名稱，然後選擇確認。

刪除行銷活動 (AWS CLI)

您可以使用 DeleteCampaign API 操作來刪除行銷活動。下列範例使用 AWS CLI。

若要刪除行銷活動，請執行下列命令。

將 *campaign-name* 取代為您要刪除之車輛的名稱。

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

> **刪除的資料分割區無法復原**
>
> 刪除行銷活動會移除裝置中的所有資料，而且分割區中的資料不會上傳至雲端。

驗證行銷活動刪除

您可以使用 ListCampaigns API 操作來驗證行銷活動是否已刪除。下列為使用 AWS CLI的範例。

若要擷取所有行銷活動的分頁摘要清單，請執行下列命令。

```
aws iotfleetwise list-campaigns
```

Get AWS IoT FleetWise 行銷活動資訊

您可以使用 GetCampaign API 操作來擷取車輛資訊。下列為使用 AWS CLI的範例。

若要擷取行銷活動的中繼資料，請執行下列命令。

將 *campaign-name* 取代為您要擷取的行銷活動名稱。

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

> **Note**
>
> 此操作為最終一致。換言之，可能不會立即反映行銷活動的變更。

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetCampaign` API 操作。

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

儲存和轉送行銷活動資料

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

使用行銷活動中的資料分割區，暫時將訊號資料存放在車輛和機群的 Edge 上。透過設定資料分割區的上傳和儲存選項，您可以將資料轉送到指定資料目的地（例如 Amazon S3 儲存貯體）的理想條件最佳化。例如，您可以設定資料分割區，將資料存放在車輛上，直到其連線至 Wi-Fi 為止。然後，一旦車

輛連線，行銷活動會觸發要傳送至雲端的特定分割區中的資料。或者，您可以使用 AWS IoT 任務收集資料。

主題

- 建立資料分割區
- 上傳行銷活動資料
- 使用 AWS IoT 任務上傳資料

建立資料分割區

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

行銷活動中的資料分割區會暫時存放訊號資料。您可以設定將資料轉送至雲端的時間和方式。

資料分割區的運作方式是先使用行銷活動`dataPartitionId`的 來指定特定資料集。然後，您可以進一步定義分割區儲存選項，例如大小上限、保持資料分割區即時的最短時間（在磁碟上），以及將資料存放在 Edge 上的位置。您可以使用 來判斷車輛上的儲存位置`storageLocation`。儲存位置會決定行銷活動儲存資料夾下資料分割區的資料夾名稱。行銷活動儲存資料夾位於 Edge 組態檔案中定義的持久性路徑下，名為 車輛名稱的資料夾下。這是儲存位置的絕對路徑：`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`。

多工緩衝處理模式設定為`TO_DISK`指定分割的資料應儲存至車輛的磁碟。資料分割區的資料儲存是以 FIFO（先進先出）為基礎運作。如果您刪除行銷活動，也可以刪除相關聯資料分割區中的資料。如果您未指定連線開/關使用案例的資料分割區， AWS IoT FleetWise 仍會在沒有連線時將資料存放在車輛的環形緩衝區中。當連線恢復時， AWS IoT FleetWise 會將資料上傳至雲端。此行為可在 Edge Agent for AWS IoT FleetWise 軟體中設定。

> **Important**
>
> 如果您的資料分割區超過設定的最大儲存限制，則較新的資料會在分割區達到大小上限時覆寫較舊的資料。Edge 上遺失的資料無法復原。儲存體大小取決於您的 Edge 儲存體限制。
> 當資料上傳到雲端時，可以在最短的存留時間之後移除。將最短時間設定為適當運作，以避免意外刪除。

上傳選項會決定變數表達式和條件語言。如果指定了上傳選項，您還必須指定儲存選項。您也可以請求將資料分割區中的訊號上傳到雲端。如需詳細資訊，請參閱上傳行銷活動資料。

定義資料分割區條件後， `signalsToCollect`有助於指定要在資料分割區中考慮的訊號。您可以為資料分割區指定 IDs，或`dataPartitionIddefault`將 設定為使用建立的預設資料分割區。沒有指定的訊號`dataPartitionId`將與預設 相關聯`dataPartition`。

建立資料分割區

使用下列範例，建立具有資料分割區儲存條件的行銷活動。此範例行銷活動設定為將車輛資料存放在 Amazon Timestream 中。

1. 將 *campaign-name* 取代為您建立的行銷活動名稱。
2. （選用）提供描述。
3. 將 *role-arn* 取代為任務執行角色的 Amazon Resource Name (ARN)，授予 AWS IoT FleetWise 將資料交付至 Timestream 資料表的許可。
4. 將 *table-arn* 取代為 Timestream 資料表的 ARN。
5. 將 *signal-catalog-arn* 取代為訊號目錄的 ARN。
6. 將 `dataPartitions` ID 和 的 *data-partition-id* 都取代為要與 建立關聯的 IDsignalsToCollect。首先，取代要在訊號中使用的資料分割區 ID。對於 `signalsToCollect`，ID 必須與 中提供的其中一個 IDs 相符`dataPartitions`。

   > **Note**
   >
   > 使用 `default`做為 ID，為行銷活動建立預設資料分割區。

7. 將 *target-arn* 取代為您建立的機群或車輛的 ARN。

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

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

符合所有指定條件後，分割的資料會轉送至雲端，以便收集和儲存新的分割訊號。

接下來，您將呼叫 `UpdateCampaign` API，將其部署到 Edge Agent for AWS IoT FleetWise 軟體。如需詳細資訊，請參閱上傳行銷活動資料。

上傳行銷活動資料

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

有兩種方式可在 Edge 上上傳行銷活動資料：

- 符合您上傳條件的行銷活動會在核准後自動將資料上傳至雲端。若要核准行銷活動，請使用 `updateCampaign` API 操作。
- 透過 AWS IoT 任務，即使未滿足指定的條件，您也可以強制資料上傳。如需詳細資訊，請參閱使用 AWS IoT 任務上傳資料。

使用 **`UpdateCampaign`** API 操作上傳行銷活動資料

建立行銷活動後，行銷活動狀態會顯示為 ，`WAITING_FOR_APPROVAL`直到您`action`將 變更為 為止`APPROVED`。

- 使用下列範例`action`，透過呼叫 UpdateCampaign API 操作來更新行銷活動。

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

使用 AWS IoT 任務上傳資料

> **Important**
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

使用 AWS IoT 任務，您可以設定行銷活動，在需要時將存放的車輛資料上傳至雲端。

為您的行銷活動建立任務文件

- 使用下列範例來建立行銷活動的任務文件。任務文件是 .json 檔案，其中包含執行任務所需的車輛或機群資訊。如需建立任務文件的詳細資訊，請參閱《 AWS IoT 開發人員指南》中的使用 建立和管理任務 AWS CLI。

  若要請求只有一輛車上傳資料，請將任務目標設定為與車輛相關聯的 AWS IoT 物件。若要請求多個車輛（在相同的行銷活動中）上傳資料，請建立與車輛對應的所有物件物件群組，然後將任務目標設定為物件群組。

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

  a. `CampaignArn` 將 取代為相同區域和帳戶中行銷活動的 Amazon Resource Name (ARN)。行銷活動 ARN 為必要項目。

  b. （選用）`endTime`以 ISO 8601 UTC 格式（無毫秒）的車輛收集資料的時間戳記取代 。例如：`2024-03-05T23:00:00Z`。時間戳記是排他性的，並決定要上傳的最後一個資料點。如果您省略 `endTime`，則 Edge Agent 軟體會繼續上傳，直到上傳行銷活動的所有儲存資料為止。上傳所有資料後，它會將任務執行狀態更新為 `SUCCEEDED`。任務的狀態更新為 `COMPLETED`。

使用受管任務範本建立任務

1. 從受管範本清單中選擇 IoT-IoTFleetWise-CollectCampaignData。如需詳細資訊，請參閱《 AWS IoT 開發人員指南》中的從 AWS 受管範本建立任務。
2. 受管範本具有 `CampaignArn`和 `endTime` 參數。

   a. `CampaignArn` 將 取代為相同區域和帳戶中行銷活動的 Amazon Resource Name (ARN)。行銷活動 ARN 為必要項目。

   b. （選用） `endTime`以 ISO 8601 UTC 格式 （無毫秒） 的車輛收集資料的時間戳記取代 。例如：`2024-03-05T23:00:00Z`。時間戳記是排他性的，並決定要上傳的最後一個資料點。如果您省略 `endTime`，則 Edge Agent 軟體會繼續上傳，直到上傳行銷活動的所有儲存資料為止。上傳所有資料後，它會將任務執行狀態更新為 `SUCCEEDED`。任務的狀態會更新為 `COMPLETED`。

如需相關故障診斷主題，請參閱 存放和轉送問題。

如需 AWS IoT 任務的詳細資訊，請參閱《 AWS IoT 開發人員指南》中的任務。

使用 AWS IoT FleetWise 收集診斷問題碼資料

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

當車輛偵測到錯誤時，會產生診斷故障碼 (DTC)，並記錄受影響感應器或致動器的快照。DTC 可協助您近乎即時地了解錯誤、了解造成這些錯誤的原因，並採取修正動作。 AWS IoT FleetWise 支援 DTCs 的集合，包括對應的 DTC 快照，以及透過資料收集活動延伸資料。本主題介紹有助於 DTC 資料收集的概念、工作流程和關鍵字，如範例所示。

以下顯示使用 DTC 的關鍵概念。

自訂定義的函數

自訂定義的函數是叫用和執行您自己在 Edge Agent 上預先定義的函數的能力，延伸自訂解碼概念。這些函數會與 AWS IoT FleetWise Agent 協調使用。Edge Agent for AWS IoT FleetWise 軟體提供內建函數，可用來計算訊號統計資料，例如最小值、最大值和平均值。自訂定義的函數可讓您

針對特定使用案例建立自訂邏輯，藉此延伸此功能。對於診斷故障碼 (DTC) 資料收集，開發人員可以利用自訂函數來實作進階資料擷取機制，例如透過統一診斷服務 (UDS) 或替代診斷介面直接從車輛的 Edge 擷取 DTC 碼、快照和延伸資料。

如需詳細資訊，請參閱 Edge Agent 開發人員指南中的自訂函數指南和 DTC 資料收集參考實作。

訊號擷取

在資料收集活動中，訊號通常會持續從裝置收集，並在 Edge Agent 軟體上緩衝。然後，訊號會定期上傳或存放在以時間為基礎的行銷活動中，或由條件為基礎的行銷活動中的特定條件觸發。不過，由於擔心裝置流量擁塞，無法從裝置收集 DTC 訊號並持續緩衝。為了解決這個問題， AWS IoT FleetWise 提供訊號擷取，可確保目標訊號從裝置不連續地擷取。

訊號擷取支援定期和條件驅動的動作。您可以使用自訂定義的函數，針對不應持續從裝置收集的每個訊號，定義擷取驅動的方法、條件和確切動作。對於由訊號擷取機制管理的訊號，本機儲存或雲端上傳的觸發類型和條件仍受 和 `timeBasedCollectionScheme` `conditionBasedCollectionScheme` 支援`CollectionScheme`，這與一般訊號相同。

下列主題說明如何建立和使用 DTCs。

主題

- 診斷問題碼關鍵字
- 為診斷問題碼建立資料收集行銷活動
- 診斷故障碼使用案例

診斷問題碼關鍵字

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

`signalsToFetch` 建立行銷活動的 參數

使用 signalsToFetch 語法來設定如何在 Edge 上擷取訊號資訊。標準訊號擷取是透過建模控制，這是在解碼器資訊清單中明確定義的規則，或透過 Edge First Modeling 定義的自訂規則。透過要擷取的訊號，您可以定義行銷活動期間擷取資料的時間和方式。

要擷取的訊號允許收集 DTC 資訊。例如，您可以建立名為 的字串類型訊號`DTC_Info`，其中包含每個引擎控制單元 (ECU) 的 DTC 資訊。或者，您可以篩選特定 ECU。

- `SignalFetchInformation` 結構和參數定義。

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

  - `fullyQualifiedName`：您要使用自訂擷取之訊號的完整名稱 (FQDN)。
  - `signalFetchConfig`：定義如何擷取上述定義訊號的規則。它支援時間型和條件型擷取。
  - `conditionLanguageVersion`：用於剖析組態中表達式的條件式語言版本。
  - `actions`：在 Edge 上評估的所有動作表達式清單。Edge 將取得已定義訊號的值。

> **Important**
>
> 動作只能使用 `custom_function`。

行銷活動表達式關鍵字

下列表達式會取得車輛支援之訊號的完整名稱，如果訊號在 Edge 的訊號緩衝區中沒有任何資料，則傳回 true。另一方面，它會傳回 false。

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

Example 使用方式

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

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

此表達式採用下列輸入：

functionName：String

Edge 支援的自訂函數名稱

參數：varargs*Expression*

的參數`functionName`。這可以是任何表達式清單。

參數支援常值類型：字串、整數 、布林值或雙值。

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

Example 使用方式

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

為診斷問題碼建立資料收集行銷活動

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

本主題說明如何建立診斷問題碼 (DTC) 的資料收集行銷活動。

1. 在 Edge 上定義自訂訊號。您需要將 Edge 上 DTC 訊號的解碼規則定義為自訂解碼訊號。如需詳細資訊，請參閱教學課程：使用自訂解碼界面設定網路無關的資料收集。
2. 在 Edge 上定義自訂函數。您需要定義自訂函數，以在編譯時間收集 Edge 上的 DTC 訊號。

   如需詳細資訊，請參閱 Edge Agent 開發人員指南中的自訂函數指南和 DTC 資料收集參考實作。

> **Note**
>
> 範例自訂定義的函數DTC_QUERY如示範指令碼所示。

3. 建立將 DTC 訊號建模為字串類型的訊號目錄。

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

4. 新增 DTC 訊號，以建立和啟用車輛模型。
5. 建立和啟用已新增 DTC 訊號的解碼器資訊清單。DTC 訊號應為具有CUSTOM_DECODING_INTERFACE網路介面類型的CUSTOM_DECODING_SIGNAL訊號解碼器類型。

   Example 訊號解碼器

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

```
      "id": "Vehicle.ECU1.DTC_INFO"
    }
  }
 ]
```

Example 網路介面

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

> **Note**
>
> 控制器區域網路 (CAN) 訊號不支援字串資料類型。

6. 佈建和建立 車輛。車輛必須使用在先前步驟中啟用的車輛模型 （ 模型資訊清單 ） 和解碼器資訊清單。
7. 建立和核准行銷活動。您需要透過定義 DTC 訊號 （ 選用使用遙測訊號 ） 並部署到車輛來建立行銷活動。
8. 存取已定義目的地中的資料。DTC 資料在行銷活動中定義的資料目的地中，包含 `DTCCode`、`DTCSnapshot`和 `DTCExtendedDatastrings`做為原始字串。

診斷故障碼使用案例

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列使用案例假設 `DTC_QUERY`函數已在示範指令碼中定義。

定期擷取

以設定的間隔擷取 DTC 集合。

下列範例是具有針對所有 ECUs `Vehicle.DTC_INFO` 的狀態遮罩之所有 DTCs 定期擷取訊號的行銷活動。為 收集的資料有條件`Vehicle.DTC_INFO`。

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "timeBased": {
        // The FleetWise Edge Agent will query the UDS module for all DTCs every five
 seconds.
          "executionFrequencyMs": 5000
        }
      },
      "actions": [
      // Every five seconds, this action is called and its output is stored in the
      // signal history buffer of Vehicle.DTC_INFO
        "custom_function(\"DTC_QUERY\", -1, 2, -1)"
      ]
    }
  ],
  "signalsToCollect": [
    {
      "name": "Vehicle.ECU1.DTC_INFO"
    }
  ],
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      // Whenever a new DTC is filled into the signal, the data is ingested.
      "expression": "!isNull($variable.`Vehicle.ECU1.DTC_INFO`)",
      "minimumTriggerIntervalMs": 1000,
      // Make sure that data is ingested only when there are new DTCs.
      "triggerMode": "RISING_EDGE"
    }
  },
  "dataDestinationConfigs": [
```

```
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

條件驅動型擷取

符合條件時擷取 DTC 集合。例如，當 CAN 訊號為 時`Vehicle.Ignition == 1`，請擷取和上傳 DTC 資料。

下列範例行銷活動具有 的條件驅動訊號擷取`Vehicle.ECU1.DTC_INFO`，以檢查 DTC ("AAA123") 是否正在等待recordNumber 1 用於 ECU-1。此行銷活動具有以時間為基礎的資料收集和上傳。

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "conditionBased": {
        // The action will only run when the ignition is on.
          "conditionExpression": "$variable.`Vehicle.Ignition` == 1",
          "triggerMode": "ALWAYS"
        }
      },
      // The UDS module is only requested for the specific ECU address and the specific
 DTC Number/Status.
      "actions": ["custom_function(\"DTC_QUERY\", 1, 2, 8, \"0xAAA123\")"]
    }
  ],
  "signalsToCollect": [
    {
      "name": "Vehicle.ECU1.DTC_INFO"
    },
    {
```

```
      "name": "Vehicle.Ignition"
    }
  ],
  "collectionScheme": {
    "timeBasedCollectionScheme": {
      "periodMs": 10000
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

隨需擷取

擷取機群的特定 DTC。

對於隨需使用案例，您可以使用與定期擷取中定義的相同行銷活動。使用 AWS IoT FleetWise 主控台或執行下列 CLI 命令，在行銷活動部署後立即暫停行銷活動，即可實現隨需效果。

- 以命令名稱取代 *command-name*。

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

然後，在 DTC 資料送達後暫停行銷活動。

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

您可以再次繼續行銷活動以進行 DTC 資料擷取。

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

Visualize AWS IoT FleetWise 車輛資料

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

Edge Agent for AWS IoT FleetWise 軟體會將選取的車輛資料傳送至 MQTT 主題，或將其傳輸至 Amazon Timestream 或 Amazon Simple Storage Service (Amazon S3)。資料送達資料目的地後，您可以使用其他 AWS 服務來處理、重新路由、視覺化和共用資料。

> **Note**
>
> Amazon Timestream 不適用於亞太區域（孟買）區域。

處理傳送至 MQTT 主題的車輛資料

MQTT 訊息傳送的車輛資料會以近乎即時的方式交付，並可讓您使用規則來採取行動，或將資料路由至其他目的地。如需使用 MQTT 的詳細資訊，請參閱《 AWS IoT Core 開發人員指南》中的 裝置通訊協定和 的規則 AWS IoT。

在 MQTT 訊息中傳送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 ID。
`vehicleName`	varchar	收集資料的車輛 ID。
`name`	varchar	Edge Agent 軟體用來收集資料的行銷活動名稱。

欄位名稱	資料類型	描述
`time`	timestamp	資料點的時間戳記。
`measure_name`	varchar	訊號的名稱。
`measure_value::bigint`	bigint	Integer 類型的訊號值。
`measure_value::double`	double	Double 類型的訊號值。
`measure_value::boolean`	boolean	布林值類型的訊號值。
`measure_value::varchar`	varchar	varchar 類型的訊號值。

在 Timestream 中處理車輛資料

Timestream 是全受管的時間序列資料庫，可每天存放和分析數兆個時間序列資料點。您的資料會存放在客戶受管的 Timestream 資料表中。您可以使用 Timestream 查詢車輛資料，以便深入了解您的車輛。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 ID。
`vehicleName`	varchar	收集資料的車輛 ID。
`name`	varchar	Edge Agent 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。

欄位名稱	資料類型	描述
`measure_value::bigint`	bigint	Integer 類型的訊號值。
`measure_value::double`	double	Double 類型的訊號值。
`measure_value::boolean`	boolean	布林值類型的訊號值。
`measure_value::varchar`	varchar	varchar 類型的訊號值。

視覺化儲存在 Timestream 中的車輛資料

將車輛資料傳輸至 Timestream 之後，您可以使用下列 AWS 服務來視覺化、監控、分析和共用您的資料。

- 使用 Grafana 或 Amazon Managed Grafana 視覺化和監控儀表板中的資料。您可以使用單一 Grafana 儀表板視覺化來自多個 AWS 來源（例如 Amazon CloudWatch 和 Timestream) 和其他資料來源的資料。
- 使用 QuickSight 分析和視覺化儀表板中的資料。

在 Amazon 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。您可以使用 S3 處理各種使用案例，例如資料湖、備份和還原、封存、企業應用程式、 AWS IoT 裝置和大數據分析。您的資料會存放在 S3 中，做為儲存貯體中的物件。如需詳細資訊，請參閱什麼是 Amazon S3？

傳輸至 Amazon S3 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 ID。
`vehicleName`	varchar	收集資料的車輛 ID。

欄位名稱	資料類型	描述
name	varchar	Edge Agent 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	Integer 類型的訊號值。
measure_value_DOUBLE	double	Double 類型的訊號值。
measure_value_BOOLEAN	boolean	布林值類型的訊號值。
measure_value_STRUCT	struct	結構類型的訊號值。
measure_value_VARCHAR	varchar	varchar 類型的訊號值。

Amazon S3 物件格式

AWS IoT FleetWise 會將車輛資料傳輸到儲存為物件的 S3。您可以使用可唯一識別資料的物件 URI 來尋找行銷活動中的資料。S3 物件 URI 格式取決於收集的資料是非結構化還是已處理的資料。

非結構化資料

非結構化資料會以非預先定義的方式存放在 S3 中。它可以採用各種格式，例如影像或影片。

使用來自 Amazon Ion 檔案的訊號資料傳遞至 AWS IoT FleetWise 的車輛訊息會解碼，並以物件形式傳輸到 S3。S3 物件代表每個訊號，並以二進位編碼。

非結構化資料 S3 物件 URI 使用以下格式：

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

已處理的資料

已處理的資料會存放在 S3 中，並進行驗證、擴充和轉換訊息的處理步驟。物件清單和速度是已處理資料的範例。

傳輸至 S3 的資料會儲存為物件，代表緩衝約 10 分鐘的記錄。根據預設， AWS IoT FleetWise 會在將物件寫入 S3 `year=YYYY/month=MM/date=DD/hour=HH`之前，以 格式新增 UTC 時間字首。此字首會在儲存貯體中建立邏輯階層，其中每個正斜線 (/) 都會在階層中建立層級。處理的資料也包含非結構化資料的 S3 物件 URI。

處理的資料 S3 物件 URI 使用以下格式：

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

原始資料

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疑難排解任何問題或根本原因錯誤。

原始資料 S3 物件 URI 使用以下格式：

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

分析存放在 Amazon S3 的車輛資料

將車輛資料傳輸至 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 SageMaker AI 擷取和分析資料，以進行下游標記和機器學習 (ML) 工作流程。

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

- 程序資料
- 訓練機器學習模型
- 標籤影像

使用 為資料編製目錄， AWS Glue 編目程式 並在 Amazon Athena 中進行分析。根據預設，寫入 S3 的物件具有 Apache Hive 樣式時間分割區，其中包含以等號連接的鍵值對的資料路徑。

如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的下列主題：

- 在 Athena 中分割資料
- 使用 AWS Glue 連線到 Amazon S3 中的資料來源
- 搭配 使用 Athena 時的最佳實務 AWS Glue

透過直接讀取 Athena 資料表或 S3 儲存貯體，使用 QuickSight 視覺化資料。

> **Tip**
>
> 如果您直接從 S3 讀取，請確認您的車輛資料為 JSON 格式，因為 QuickSight 不支援 Apache Parquet 格式。

如需詳細資訊，請參閱《Amazon QuickSight 使用者指南》中的下列主題：

- 支援的資料來源
- 建立資料來源

命令

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。
> 本文件說明如何使用適用於 AWS IoT FleetWise 的命令功能。如需在 中使用命令功能的詳細資訊 AWS IoT Device Management，請參閱 命令。
> 您全權負責以安全且符合相關法律的方式部署命令。如需 責任的詳細資訊，請參閱 AWSAWS IoT 服務條款。

使用命令功能，從雲端對車輛執行命令。命令一次以一個裝置為目標，可用於低延遲、高輸送量的應用程式，例如擷取裝置端日誌，或啟動裝置狀態變更。

命令是由 管理的資源 AWS IoT Device Management。它包含可在傳送命令執行至車輛時套用的可重複使用組態。您可以針對特定使用案例預先定義一組命令，或使用它們為經常性使用案例建立可重複使用的組態。例如，您可以設定 命令，供 應用程式用來鎖定車輛的門或遠端變更溫度。

使用 AWS IoT 命令功能，您可以：

- 建立命令資源並重複使用組態，將多個命令傳送至您的目標裝置，然後在裝置上執行這些命令。
- 控制要在裝置上執行每個命令的精細程度。例如，您可以將車輛佈建為 AWS IoT 物件，然後傳送命令來鎖定或解鎖車輛的門。
- 在目標裝置上同時執行多個命令，無需等待上一個命令完成。
- 選擇 以啟用命令事件的通知，並在執行命令時以及完成後，從裝置擷取狀態和結果資訊。

下列主題說明如何建立、傳送、接收和管理命令。

主題

- 命令概念
- 車輛和命令
- 建立和管理命令
- 啟動和監控命令執行
- 範例：使用命令控制車輛轉向模式 (AWS CLI)

- 命令使用案例

命令概念

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

命令是從雲端傳送至目標裝置的指示。目標裝置可以是 車輛，且必須在AWS IoT 物件登錄檔中註冊為物件。命令可以包含參數，定義車輛傳動器需要執行的動作。車輛接著會剖析命令及其參數，並處理它們以採取對應的動作。然後，它會以命令執行的狀態回應雲端應用程式。

如需詳細的工作流程，請參閱 車輛和命令。

主題

- 命令關鍵概念
- 命令執行狀態

命令關鍵概念

以下顯示一些使用 命令功能的重要概念，以及其如何使用上次已知狀態 (LKS) 狀態範本。

命令

命令是您可以用來將指示傳送至實體車輛的實體，讓它執行動作，例如開啟引擎或變更視窗的位置。您可以針對特定使用案例預先定義一組命令，或使用它們為經常性使用案例建立可重複使用的組態。例如，您可以設定 命令，供 應用程式用來鎖定車輛的門或遠端變更溫度。

命名空間

使用命令功能時，您必須指定命令的命名空間。當您在 AWS IoT FleetWise 中建立命令時，您必須選擇 `AWS-IoT-FleetWise`做為命名空間。當您使用此命名空間時，必須提供將用於在車輛上執行 命令的參數。如果您想要 AWS IoT Device Management 改為在 中建立命令，則必須改用 `AWS-IoT` 命名空間。如需詳細資訊，請參閱 AWS IoT Device Management 開發人員指南中的 命令。

命令狀態

您建立的命令將處於可用狀態，這表示它可用於在車輛上啟動命令執行。如果命令過期，您可以棄用命令。對於處於已棄用狀態的命令，現有的命令執行將執行到完成。您無法更新命令或執行任何新的執行。若要傳送新的執行，您必須還原命令，使其變成可用。

如果不再需要命令，您也可以將其刪除。當您將命令標記為刪除時，如果命令已棄用超過最大逾時 24 小時的持續時間，則會立即刪除命令。如果命令未棄用，或已棄用超過最大逾時的持續時間，則命令將處於待定刪除狀態。命令會在 24 小時後自動從您的帳戶中移除。

參數

建立命令時，您可以選擇指定您希望目標車輛在執行命令時執行的參數。您建立的 命令是可重複使用的組態，可用來將多個命令執行傳送至您的 車輛，並同時執行它們。或者，您也可以只在執行時間指定參數，並選擇執行建立命令並將其傳送至車輛的一次性操作。

目標車輛

當您想要執行 命令時，您必須指定將接收命令並執行特定動作的目標車輛。目標車輛必須已註冊為物件 AWS IoT。將命令傳送至車輛後，它會根據您指定的參數和值，開始執行命令的執行個體。

傳動器

當您想要執行 命令時，您必須在車輛上指定將接收命令的傳動器，以及決定要執行動作的值。您可以選擇性地設定傳動器的預設值，以避免傳送不正確的命令。例如，您可以將預設值 `LockDoor`用於門鎖致動器，讓命令不會意外解除鎖定門。如需傳動器的一般資訊，請參閱 重要概念。

資料類型支援

用於命令功能的傳動器支援以下資料類型。

> **Note**
>
> 遠端資訊處理資料、命令或最後已知狀態 (LKS) 不支援陣列。您只能將陣列資料類型用於視覺系統資料。

- 浮點數類型。支援下列類型。
  - 浮點數 (32 位元)
  - 雙 (64 位元)
- 整數 （已簽章和未簽章）。支援下列整數類型。
  - int8 和 uint8

  - int16 和 uint16
  - int32 和 uint32
- 長。支援下列長型。
  - 長 (int64)
  - 未簽署的長 (uint64)
- 字串
- Boolean

命令執行

命令執行是在目標裝置上執行之命令的執行個體。車輛會使用您在建立命令或啟動命令執行時指定的參數來執行命令。車輛接著會執行指定的操作，並傳回執行的狀態。

> **Note**
>
> 對於指定的車輛，您可以同時執行多個命令。如需您可以為每個車輛執行並行執行數目上限的資訊，請參閱AWS IoT Device Management 命令配額。

上次已知狀態 (LKS) 狀態範本

狀態範本為車主提供追蹤其車輛狀態的機制。若要近乎即時地監控您車輛的最後已知狀態 (LKS)，您可以建立狀態範本並將其與您的車輛建立關聯。

使用 命令功能，您可以執行可用於狀態資料收集和處理的「隨需」操作。例如，您可以請求目前車輛狀態一次性（擷取），或啟用或停用先前部署的 LKS 狀態範本，以開始或停止報告車輛資料。如需示範如何使用 命令搭配狀態範本的範例，請參閱 命令使用案例。

命令執行狀態

啟動命令執行後，您的車輛可以發佈執行狀態，並提供狀態原因做為執行的其他資訊。下列各節說明各種命令執行狀態和狀態碼。

主題

- 命令執行狀態原因代碼和描述
- 命令執行狀態和狀態碼
- 命令執行逾時狀態

命令執行狀態原因代碼和描述

若要報告命令執行狀態的更新，您的車輛可以使用 `UpdateCommandExecution` API，使用AWS IoT Core 開發人員指南中所述的命令保留主題，將更新後的狀態資訊發佈至雲端。報告狀態資訊時，您的裝置可以使用 `StatusReason` 物件提供每個命令執行狀態的其他內容，以及包含在 物件中的欄位 `reasonCode` 和 `reasonDescription` 。

命令執行狀態和狀態碼

下表顯示各種命令執行狀態碼，以及命令執行可以轉換到的允許狀態。它也會顯示命令執行是否為「終端」（即不會進一步進行狀態更新）、變更是由車輛或雲端啟動，以及不同的預先定義狀態碼，以及它們如何對應到雲端報告的狀態。

- 如需有關 AWS IoT FleetWise 如何使用預先定義狀態碼和 `statusReason` 物件的資訊，請參閱 Edge Agent for AWS IoT FleetWise 軟體文件中的命令狀態。
- 如需有關終端機和非終端機執行以及狀態之間轉換的其他資訊，請參閱AWS IoT Core 開發人員指南中的命令執行狀態。

命令執行狀態和來源

命令執行狀態	描述	由裝置/雲端啟動？	終端機執行？	允許的狀態轉換	預先定義的狀態碼
`CREATED`	當開始執行命令的 API 請求 (`StartCommandExecution` API) 成功時，命令執行狀態會變更為 `CREATED`。	雲端	否	• `IN_PROGRESS` • `SUCCEEDED` • `FAILED` • `REJECTED` • `TIMED_OUT`	無
`IN_PROGRESS`	當車輛開始執行命令時，可以發佈訊息至回應主題，將狀態更新為	裝置	否	• `IN_PROGRESS` • `SUCCEEDED` (成功) • 失敗	`COMMAND_STATUS_COMMAND_IN_PROGRESS`

命令執行狀態	描述	由裝置/雲端啟動？	終端機執行？	允許的狀態轉換	預先定義的狀態碼
	IN_PROGRESS 。			• REJECTED • TIMED_OUT	
SUCCEEDED	當車輛成功處理命令並完成執行時，可以將訊息發佈至回應主題，將狀態更新為 SUCCEEDED 。	裝置	是	不適用	COMMAND_STATUS_SUCCEEDED
FAILED	當車輛無法執行命令時，可以將訊息發佈至回應主題，以將狀態更新為 FAILED。	裝置	是	不適用	COMMAND_STATUS_EXECUTION_FAILED
REJECTED	如果車輛無法接受命令，可以將訊息發佈至回應主題，將狀態更新為 REJECTED。	裝置	是	不適用	無

命令執行狀態	描述	由裝置/雲端啟動？	終端機執行？	允許的狀態轉換	預先定義的狀態碼
TIMED_OUT	由於下列任何原因TIMED_OUT，命令執行狀態可能會變更為。 • 未收到命令執行的結果，雲端會自動報告TIMED_OUT狀態。 • 車輛回報嘗試執行 命令時發生逾時。在此情況下，命令執行會變成終端機。 如需此狀態的詳細資訊，請參閱 命令執行逾時狀態。	裝置和雲端	否	• SUCCEEDED (成功) • 失敗 • REJECTED • TIMED_OUT	COMMAND_STATUS_EXECUTION_TIMEOUT

命令執行逾時狀態

雲端和裝置都可以報告命令執行逾時。命令傳送至裝置後，計時器便會啟動。如果在指定的持續時間內沒有收到來自裝置的回應，則雲端會報告TIMED_OUT狀態。在此情況下，TIMED_OUT狀態為 的命令執行為非終端。

裝置可以將此狀態覆寫為終端機狀態，例如 SUCCEEDED、 FAILED或 REJECTED。它也可以報告執行命令時發生逾時。在此情況下，命令執行狀態會保持在 ，TIMED_OUT但StatusReason物件的欄位會根據裝置報告的資訊更新。狀態中的命令執行TIMED_OUT現在會變成終端機。

如需詳細資訊，請參閱AWS IoT Core 《 開發人員指南》中的命令執行逾時考量事項。

車輛和命令

> **Important**
>
> 存取特定 AWS IoT FleetWise 功能目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。
> 您全權負責以安全且符合適用法律的方式部署命令。

若要使用命令功能：

1. 首先，建立命令資源。或者，指定包含執行命令所需資訊的參數。
2. 指定將接收命令的目標車輛，並執行指定的動作。
3. 現在，您可以在目標裝置上執行 命令，並檢查命令執行詳細資訊以擷取狀態，並使用 CloudWatch 日誌進一步疑難排解任何問題。

以下各節顯示 車輛和 命令之間的工作流程。

主題

- 工作流程概觀
- 車輛工作流程
- 命令工作流程
- （選用）命令通知

工作流程概觀

下列步驟提供 車輛和 命令之間的命令工作流程概觀。當您使用任何 命令 HTTP API 操作時，會使用 Sigv4 登入資料來簽署請求。

> **Note**
>
> 除了 `StartCommandExecution` API 操作之外，透過 HTTP 通訊協定執行的所有操作都會使用控制平面端點。

1. 建立 MQTT 連線並訂閱命令主題

   若要準備命令工作流程，裝置必須與`iot:Data-ATS`端點建立 MQTT 連線，並訂閱上述命令請求主題。或者，您的裝置也可以訂閱接受和拒絕的命令回應主題。

2. 建立車輛模型和命令資源

   您現在可以使用 和 `CreateCommand`控制平面 API 操作來建立 車輛`CreateVehicle`和命令資源。命令資源包含要在車輛上執行命令時套用的組態。

3. 在目標裝置上啟動命令執行

   使用`StartCommandExecution`資料平面 API 搭配您的帳戶特定`iot:Jobs`端點，在車輛上啟動命令執行。API 會將 protobuf 編碼的承載訊息發佈至命令請求主題。

4. 更新命令執行的結果

   車輛會處理 命令和收到的承載，然後使用 `UpdateCommandExecution` API 將命令執行的結果發佈至回應主題。如果您的車輛訂閱了接受和拒絕的命令回應主題，則會收到訊息，指出雲端服務是否接受或拒絕回應。

5. （選用）擷取命令執行結果

   若要擷取命令執行的結果，您可以使用`GetCommandExecution`控制平面 API 操作。在您的車輛將命令執行結果發佈至回應主題後，此 API 會傳回更新的資訊。

6. （選用）訂閱和管理命令事件

   若要接收命令執行狀態更新的通知，您可以訂閱命令事件主題。然後，您可以使用`CreateTopicRule`控制平面 API 將命令事件資料路由至其他應用程式，例如 AWS Lambda 函數或 Amazon SQS，並在其上建置應用程式。

車輛工作流程

下列步驟詳細說明使用 命令功能時的車輛工作流程。

> **Note**
> 本節所述的操作使用 MQTT 通訊協定。

1. 建立 MQTT 連線

   若要讓您的車輛準備好使用命令功能，必須先連線到 AWS IoT Core 訊息代理程式。必須允許車輛執行 `iot:Connect`動作，以連線至 AWS IoT Core 並與訊息代理程式建立 MQTT 連線。若要尋找 的資料平面端點 AWS 帳戶，請使用 `DescribeEndpoint` API 或 CLI `describe-endpoint` 命令，如下所示。

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

   執行此命令會傳回帳戶特定的資料平面端點，如下所示。

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

2. 暫停命令請求主題

   建立連線之後，您的裝置就可以訂閱 AWS IoT 命令 MQTT 請求主題。當您在目標裝置上建立命令並啟動命令執行時，訊息代理程式會將 protobuf 編碼的承載訊息發佈到請求主題。然後，您的裝置可以接收承載訊息並處理 命令。在此範例中，將 取代`<DeviceID>`為目標車輛的唯一識別符。此 ID 可以是您車輛的唯一識別符或物件名稱

   > **Note**
   > 傳送至裝置的承載訊息必須使用 protobuf 格式。

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

3. （選用）訂閱命令回應主題

   或者，您可以訂閱這些命令回應主題，以接收訊息，指出雲端服務是否接受或拒絕來自裝置的回應。

> **Note**
>
> 您的 車輛可選擇訂閱 /accepted和 /rejected回應主題。即使未明確訂閱這些主題，您的車輛也會自動收到這些回應訊息。

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

4. 更新命令執行的結果

目標車輛接著會處理 命令。然後，它會使用 UpdateCommandExecution API 將執行結果發佈至下列 MQTT 回應主題。

> **Note**
>
> 對於指定的車輛和命令執行，*<DeviceID>* 必須符合裝置訂閱之請求主題中的對應欄位。

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

UpdateCommandExecution API 是 MQTT 上的資料平面 API 操作，已透過 TLS 驗證。

- 如果雲端服務成功處理命令執行結果，則會將訊息發佈至 MQTT 接受的主題。接受的主題使用下列格式。

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

- 如果雲端服務無法處理命令執行結果，則會將回應發佈至 MQTT 拒絕主題。被拒絕的主題使用下列格式。

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

如需此 API 和範例的詳細資訊，請參閱 更新命令執行結果。

命令工作流程

下列步驟會詳細說明命令工作流程。

> **Note**
>
> 本節所述的操作使用 HTTP 通訊協定。

1. 註冊您的 車輛

   現在，您已經準備好 車輛使用命令功能，您可以註冊 車輛，然後建立將傳送至車輛的命令，以準備應用程式。若要註冊車輛，請使用`CreateVehicle`控制平面 API 操作建立車輛模型（模型資訊清單）的執行個體。如需詳細資訊和範例，請參閱建立車輛。

2. 建立命令

   使用 `CreateCommand` HTTP 控制平面 API 操作來建立適用於您目標車輛的命令模型。指定執行命令時要使用的任何參數和預設值，並確認它使用 `AWS-IoT-FleetWise` 命名空間。如需使用此 API 的詳細資訊和範例，請參閱 建立命令資源。

3. 啟動命令執行

   您現在可以使用`StartCommandExecution`資料平面 API 操作來執行您在車輛上建立的命令。AWS IoT Device Management 會擷取命令和命令參數，並驗證傳入的請求。然後，它會呼叫具有必要參數的 AWS IoT FleetWise API，以產生車輛特定的承載。承載會透過 AWS IoT Device Management MQTT 傳送至裝置訂閱的命令請求主題。如需使用此 API 的詳細資訊和範例，請參閱 傳送命令 (AWS CLI)。

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

   > **Note**
   >
   > 如果從雲端傳送命令時裝置離線，且正在使用 MQTT 持久性工作階段，則命令會在訊息代理程式中等待。如果裝置在逾時持續時間之前恢復線上狀態，且已訂閱命令請求主題，則裝置可以處理命令並將結果發佈至回應主題。如果裝置未在逾時持續時間之前恢復上線，則命令執行將會逾時，承載訊息將會過期。

4. 擷取命令執行

   在裝置上執行 命令之後，請使用`GetCommandExecution`控制平面 API 操作來擷取和監控命令執行的結果。您也可以使用 API 取得執行資料的其他資訊，例如上次更新的時間、執行完成的時間，以及指定的參數。

   > **Note**
   > 若要擷取最新狀態資訊，您的裝置必須將命令執行結果發佈至回應主題。

   如需使用此 API 的詳細資訊和範例，請參閱 取得命令執行。

（選用） 命令通知

您可以訂閱命令事件，以便在命令執行狀態變更時接收通知。下列步驟說明如何訂閱命令事件，然後處理它們。

1. 建立主題規則

   您可以訂閱命令事件主題，並在命令執行狀態變更時收到通知。您也可以建立主題規則，將車輛處理的資料路由到其他應用程式，例如 AWS Lambda 函數。您可以使用 AWS IoT 主控台或`CreateTopicRule` AWS IoT Core 控制平面 API 操作來建立主題規則。如需詳細資訊，請參閱建立 和 AWS IoT 規則。

   在此範例中，*<CommandID>*將 取代為您想要接收通知之命令的識別符，並將*<CommandExecutionStatus>*取代為命令執行的狀態。

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

   > **Note**
   > 若要接收所有命令和命令執行狀態的通知，您可以使用萬用字元並訂閱下列主題。

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

2. 接收和處理命令事件

   如果您在上一個步驟中建立了主題規則來訂閱命令事件，則可以管理您收到的命令推播通知。您也可以選擇性地在應用程式上建置應用程式，例如使用您建立的主題規則搭配 AWS Lambda、Amazon SQS、Amazon SNS 或 AWS Step Functions。

下列程式碼顯示您將收到的命令事件通知的範例承載。

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

建立和管理命令

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以設定可重複使用的遠端動作，或傳送一次性的立即指示至您的裝置。使用此功能時，您可以指定裝置可以近乎即時執行的指示。命令可讓您為目標車輛設定可重複使用的遠端動作。建立命令之後，您可以啟動以特定車輛為目標的命令執行。

本主題說明如何使用 AWS IoT Core API 或 建立和管理命令資源 AWS CLI。它說明如何在命令資源上執行下列動作。

主題

- 建立命令資源
- 擷取命令的相關資訊
- 列出您帳戶中的命令
- 更新或取代命令資源
- 刪除命令資源

建立命令資源

您可以使用`CreateCommand` AWS IoT Core 控制平面 API 操作或 AWS IoT FleetWise 主控台來建立命令。

建立命令（主控台）

您可以使用 AWS IoT FleetWise 主控台來建立命令。

建立命令

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇命令。
3. 選擇建立命令。
4. 指定唯一的命令 ID，協助您識別要在車輛上執行的命令。
5. （選用）指定選用的顯示名稱和描述。
6. （選用）選取致動器和預設參數值。參數指定目標車輛在收到命令時可執行的動作。如果您未新增參數，則需要在執行 命令時提供參數。
7. 選擇授予許可以產生命令承載的 IAM 角色。請參閱控制存取。
8. 選擇建立命令。

建立命令 (AWS CLI)

下列範例示範如何使用 參數建立命令。

建立命令時的考量事項

當您在 中建立命令時 AWS IoT FleetWise：

- 您必須指定 `roleArn`，以授予在您的車輛上建立和執行命令的許可。如需範例政策的詳細資訊，包括啟用 KMS 金鑰的時間，請參閱 授予使用 產生命令承載的 AWS IoT Device Management 許可 AWS IoT FleetWise。
- 您必須指定 `AWS-IoT-FleetWise`做為命名空間。
- 您可以略過 `mandatory-parameters` 欄位，並在執行時間指定它們。或者，您可以使用參數建立命令，並選擇性地為其指定預設值。如果您指定了預設值，則可以在執行時間使用這些值，或指定自己的值來覆寫這些值。如需這些其他範例，請參閱 命令使用案例。
- 您可以為 `mandatory-parameters` 欄位指定最多三個名稱/值對。不過，在車輛上執行 命令時，只接受一個名稱/值對，而 `name` 欄位必須使用完整名稱與`$actuatorPath.`字首。

- 將 *`command-id`* 取代為命令的唯一識別符。您可以使用 UUID、英數字元、"-" 和 "_"。
- 將 *`role-arn`* 取代為授予您建立和執行命令許可的 IAM 角色，例如 `"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"`。
- （選用）以易於使用的命令名稱取代 *`display-name`*，並以有意義的命令描述取代*`##`*。
- 將`mandatory-parameters`物件*`###`*和*`#`*取代為所建立命令的必要資訊。`name` 欄位是訊號目錄中定義為`$actuatorPath.`字首的完整名稱。例如， `name`可以是 *`$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode`*，valu 可以是指示轉向模式狀態的布林值，例如 *`{"B"# false}`*。

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

`CreateCommand` API 操作會傳回回應，其中包含 命令的 ID 和 ARN (Amazon Resource Name)。

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

```
}
```

擷取命令的相關資訊

您可以使用GetCommand AWS IoT Core 控制平面 API 操作來擷取命令資源的相關資訊。

若要取得命令資源的相關資訊，請執行下列命令。將 *command-id* 取代為建立命令時使用的識別符。

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

GetCommand API 操作會傳回包含下列資訊的回應。

- 命令的 ID 和 ARN (Amazon Resource Name)。
- 命令建立和上次更新的日期和時間。
- 命令狀態，指出它是否可以在車輛上執行。
- 您在建立 命令時指定的任何參數。

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

列出您帳戶中的命令

您可以使用ListCommands AWS IoT Core 控制平面 API 操作來列出您建立之帳戶中的所有命令。

若要列出您帳戶中的命令，請執行下列命令。根據預設，API 會傳回為兩個命名空間建立的命令。若要篩選清單以僅顯示為 建立的命令 AWS IoT FleetWise，請執行下列命令。

> **Note**
> 您也可以依遞增或遞減順序排序清單，或篩選清單，以僅顯示具有特定命令參數名稱的命令。

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

`ListCommands` API 操作會傳回包含下列資訊的回應。

- 命令的 ID 和 ARN (Amazon Resource Name)。
- 命令建立和上次更新的日期和時間。
- 命令狀態，指出命令是否可以在車輛上執行。

更新或取代命令資源

您可以使用`UpdateCommand` AWS IoT Core 控制平面 API 操作或 AWS IoT FleetWise 主控台來更新命令資源。您可以更新命令的顯示名稱和描述。如果命令目前未使用，您也可以將其棄用。

> **Note**
> 您無法修改執行 命令時要使用的命名空間資訊或參數。

更新命令（主控台）

更新命令

若要從主控台更新命令，請前往 AWS IoT FleetWise 主控台的命令頁面，並執行下列步驟。

1. 選擇您要更新的命令，然後選擇編輯。
2. 編輯命令詳細資訊，然後選擇儲存變更。

棄用命令

若要從主控台棄用命令，請前往 AWS IoT FleetWise 主控台的命令頁面，並執行下列步驟。

1. 選擇您要棄用的命令，然後選擇棄用。

2. 確認棄用，然後選擇棄用。

更新命令 (AWS CLI)

更新命令

若要更新命令資源，請執行下列命令。將 *command-id* 取代為您要更新的命令識別符，並提供更新的 *display-name* 和*##*。

```
aws iot update-command \
    --command-id command-id \
    --display-name display-name \
    --description description
```

`UpdateCommand` API 操作會傳回下列回應。

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

棄用命令

當您打算不再為裝置繼續使用命令或命令已過期時，會棄用命令。下列範例顯示如何棄用命令。

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

`UpdateCommand` API 操作會傳回回應，其中包含 命令的 ID 和 ARN (Amazon Resource Name)。

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

一旦命令已棄用，現有的命令執行將繼續在車輛上執行，直到變成終端機為止。若要執行任何新的命令執行，您必須使用 `UpdateCommand` API 還原命令，使其變成可用。如需有關棄用和還原命令及其考量事項的其他資訊，請參閱《 AWS IoT Core 開發人員指南》中的棄用命令資源。

刪除命令資源

您可以使用`DeleteCommand` AWS IoT Core 控制平面 API 操作或 AWS IoT FleetWise 主控台來刪除命令資源。

> **Note**
> 刪除動作為永久性動作，且無法還原。命令將從您的帳戶中永久移除。

刪除命令（主控台）

若要從主控台刪除命令，請前往 AWS IoT FleetWise 主控台的命令頁面，並執行下列步驟。

1. 選擇您要刪除的命令，然後選擇刪除。
2. 確認您想要刪除命令，然後選擇刪除。

刪除命令 (AWS CLI)

若要刪除命令資源，請執行下列命令。以您要刪除之命令的識別符取代 *command-id*。下列範例顯示如何刪除命令資源。

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

如果刪除請求成功：

- 如果命令已棄用超過最大逾時 24 小時的持續時間，則會立即刪除命令，而且您會看到 HTTP `statusCode` 為 204。
- 如果命令未棄用，或已棄用超過最大逾時的持續時間，則命令將處於 `pending deletion` 狀態，您會看到 HTTP `statusCode` 為 202。最長逾時 24 小時後，系統會自動從您的帳戶移除命令。

啟動和監控命令執行

> **Important**
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

建立命令資源後，您可以在目標車輛上啟動命令執行。一旦車輛開始執行命令，就可以開始更新命令執行的結果，並將狀態更新和結果資訊發佈至 MQTT 預留主題。然後，您可以擷取命令執行的狀態，並監控您帳戶中的執行狀態。

本主題說明如何使用 AWS CLI or AWS IoT FleetWise 主控台將命令傳送至您的 車輛。它還說明如何監控和更新命令執行的狀態。

主題

- 更新命令執行結果
- 取得命令執行
- 列出您帳戶中的命令執行
- 刪除命令執行

傳送命令（主控台）

若要從主控台傳送命令，請前往 AWS IoT FleetWise 主控台的車輛頁面，並執行下列步驟。

1. 選擇您要傳送命令的車輛。
2. 選擇執行命令。
3. 選取命令 ID。
4. 指定命令執行逾時，然後選擇執行命令。

傳送命令 (AWS CLI)

您可以使用`StartCommandExecution` AWS IoT 資料平面 API 操作，將命令傳送至車輛。接著，車輛會將命令轉送至汽車中介軟體服務（例如 SOME/IP（透過 IP 的可擴展服務導向中介軟體）），或將其發佈至車輛網路（例如控制器區域網路 (CAN) 裝置界面）。下列為使用 AWS CLI的範例。

主題

- 傳送命令時的考量事項
- 取得帳戶特定的資料平面端點
- 傳送命令範例

傳送命令時的考量事項

當您在 中啟動命令執行時 AWS IoT FleetWise：

- 您必須為車輛佈建 AWS IoT 物件。如需詳細資訊，請參閱Provision AWS IoT FleetWise 車輛。
- 您必須已使用 建立命令`AWS-IoT-FleetWise`做為命名空間`role-Arn`，並提供 授予您在 AWS IoT FleetWise 中建立和執行命令的許可。如需詳細資訊，請參閱建立命令資源。
- 如果您選擇在建立命令時使用為參數指定的任何預設值，則可以略過 `parameters` 欄位。如果`mandatory-parameters`未在建立時指定 ，或者如果您想要透過為參數指定自己的值來覆寫任何預設值，則必須指定 `parameters` 欄位。如需這些其他範例，請參閱 命令使用案例。
- 您可以為 `mandatory-parameters` 欄位指定最多三個名稱/值對。不過，在車輛上執行 命令時，只接受一個名稱/值對，而 `name` 欄位必須使用完整名稱與`$actuatorPath.`字首。

取得帳戶特定的資料平面端點

執行 API 命令之前，您必須取得端點的帳戶特定`iot:Jobs`端點 URL。例如，如果您執行此命令：

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

它會傳回帳戶特定的端點 URL，如以下範例回應所示。

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

傳送命令範例

若要將命令傳送至車輛，請執行下列命令。

- 將 *`command-arn`* 取代為您要執行之命令的 ARN。您可以從 CLI `create-command` 命令的回應取得此資訊。
- 將 *`target-arn`* 取代為您要執行命令的目標裝置或 AWS IoT 物件的 ARN。

> **Note**
>
> 您可以指定 AWS IoT 物件的目標 ARN (AWS IoT FleetWise 車輛）。目前不支援物件群組和機群。

- 以您在 中取得的帳戶特定端點取代 *`endpoint-url`*取得帳戶特定的資料平面端點，字首為 `https://`，例如 `https://`*`123456789012abcd`*`.jobs.iot.`*`ap-south-1`*`.amazonaws.com`。
- 將*`##`*和*`#`*取代為您使用 CLI `create-command` 建立命令時指定的`mandatory-parameters`欄位。

name 欄位是訊號目錄中定義的完整名稱，字首$actuatorPath.為 。例如， name可以是 *$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode*，valu 可以是指示轉向模式狀態的布林值，例如 *{"B"# false}*。

- （選用）您也可以指定其他參數 executionTimeoutSeconds。此選用欄位指定裝置必須以執行結果回應的時間，以秒為單位。您可以將逾時設定為最大值 24 小時。

  建立命令執行後，計時器會啟動。在計時器過期之前，如果命令執行狀態未變更為使其終止的狀態，例如 SUCCEEDED或 FAILED，則狀態會自動變更為 TIMED_OUT。

> **Note**
>
> 裝置也可以報告TIMED_OUT狀態，或將此狀態覆寫為 狀態，例如 SUCCEEDED、 FAILED或 REJECTED，而且命令執行會變成終端機。如需詳細資訊，請參閱命令執行逾時狀態。

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

StartCommandExecution API 操作會傳回命令執行 ID。您可以使用此 ID 查詢命令執行狀態、詳細資訊和命令執行歷史記錄。

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

執行 命令後，您的裝置會收到包含下列資訊的通知。issued_timestamp_ms 欄位對應至叫用 StartCommandExecution API 的時間。timeout_ms 對應至叫用 StartCommandExecution API 時使用 executionTimeoutSeconds 參數設定的逾時值。

```
timeout_ms: 9000000
issued_timestamp_ms: 1723847831317
```

更新命令執行結果

若要更新命令執行的狀態，您的裝置必須已建立 MQTT 連線並訂閱下列命令請求主題。

在此範例中，將 取代*<device-id>*為目標裝置的唯一識別符，可以是 `VehicleId`或物件名稱，並將*<execution-id>*取代為命令執行的識別符。

> **Note**
>
> - 承載必須使用 protobuf 格式。
> - 您的裝置可選擇是否訂閱 `/accepted`和 `/rejected`回應主題。即使您的裝置尚未明確訂閱，也會收到這些回應訊息。

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

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

您的裝置可以將訊息發佈至命令回應主題。處理命令之後，它會傳送 protobuf 編碼的回應至此主題。*<DeviceID>* 欄位必須符合請求主題中的對應欄位。

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

裝置發佈對本主題的回應後，您可以使用 `GetCommandExecution` API 擷取更新的狀態資訊。命令執行的狀態可以是此處列出的任何項目。

- `IN_PROGRESS`
- `SUCCEEDED`
- `FAILED`
- `REJECTED`

- TIMED_OUT

請注意，狀態為 SUCCEEDED、 FAILED和 的命令執行REJECTED為終端機，且狀態是由裝置回報。當命令執行為終端機時，這表示不會進一步更新其狀態或相關欄位。裝置或雲端可能會報告TIMED_OUT狀態。如果雲端回報，裝置稍後可能會更新狀態原因欄位。

例如，以下顯示裝置發佈的範例 MQTT 訊息。

> **Note**
>
> 針對命令執行狀態，如果您的裝置使用 statusReason 物件來發佈狀態資訊，您必須確定：
>
> - reasonCode 使用 模式 [A-Z0-9_-]+，長度不超過 64 個字元。
> - 長度reasonDescription不超過 1，024 個字元。它可以使用控制字元以外的任何字元，例如新行。

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

如需示範如何使用 AWS IoT Core MQTT 測試用戶端訂閱主題並查看命令執行訊息的範例，請參閱《AWS IoT Core 開發人員指南》中的使用 MQTT 測試用戶端檢視命令更新。

取得命令執行

您可以使用GetCommandExecution AWS IoT 控制平面 API 操作來擷取命令執行的相關資訊。您必須已使用 StartCommandExecution API 操作執行此命令。

若要擷取已執行命令的中繼資料，請執行下列命令。

- 將 *execution-id* 取代為 命令的 ID。您可以從 CLI start-command-execution 命令的回應取得此資訊。

- 將 *target-arn* 取代為您要執行命令的目標車輛或 AWS IoT 物件的 ARN。

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

GetCommandExecution API 操作會傳回回應，其中包含命令執行的 ARN、執行狀態，以及命令開始執行的時間和完成時間的相關資訊。下列程式碼顯示來自 API 請求的範例回應。

為了提供有關每個命令執行狀態的其他內容，命令功能會提供 statusReason 物件。物件包含兩個欄位 reasonCode和 reasonDescription。使用這些欄位，您的裝置可以提供有關命令執行狀態的其他資訊。此資訊會覆寫從雲端報告的任何預設 reasonCode和 reasonDescription 。

若要報告此資訊，您的裝置可以將更新的狀態資訊發佈至雲端。然後，當您使用 GetCommandExecution API 擷取命令執行狀態時，您會看到最新的狀態碼。

> **Note**
>
> 執行回應中的 completedAt 欄位對應至裝置向雲端回報終端機狀態的時間。在TIMED_OUT狀態的情況下，只有在裝置報告 A 逾時時，才會設定此欄位。狀態由雲端設定TIMED_OUT時，TIMED_OUT狀態不會更新。如需逾時行為的詳細資訊，請參閱 命令執行逾時狀態。

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

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

```
}
```

列出您帳戶中的命令執行

使用ListCommandExecutions AWS IoT Core 控制平面 HTTP API 操作列出您帳戶中的所有命令執行。此範例使用 AWS CLI。

主題

- 列出命令執行時的考量事項
- 列出命令執行範例

列出命令執行時的考量事項

以下是使用 ListCommandExecutions API 時的一些考量。

- 您必須commandArn根據要列出特定命令targetArn或目標車輛的執行，至少指定 或 。API 請求不能是空的，也不能包含相同請求中的兩個欄位。
- 您必須僅提供 startedTimeFilter或 completedTimeFilter資訊。API 請求不能是空的，也不能包含相同請求中的兩個欄位。您可以使用 物件的 before和 after 欄位，列出在特定時間範圍內建立或完成的命令執行。
- before 和 after 欄位不得大於目前時間。根據預設，如果您未指定任何值， before 欄位是目前時間， after 欄位是目前時間 - 6 個月。也就是說，根據您使用的篩選條件，API 會列出過去六個月內建立或完成的所有執行。
- 您可以使用 sort-order 參數來指定是否要以遞增順序列出執行。根據預設，如果您未指定此欄位，則執行會以遞減順序列出。
- 當列出命令 ARN 的命令執行時，您無法根據命令執行的狀態來篩選命令執行。

列出命令執行範例

下列範例示範如何在 中列出命令執行 AWS 帳戶。

執行 命令時，您必須指定是否要篩選清單，以僅顯示使用 為特定裝置建立的命令執行targetArn，還是使用 為特定命令指定的執行commandArn。

在此範例中，取代：

- *<target-arn>* 您以執行為目標之裝置的 Amazon Resource Number (ARN)，例如 `arn:aws:iot:us-east-1:123456789012:thing/b8e4157c98f332cffb37627f`。
- *<target-arn>* 您以執行為目標之裝置的 Amazon Resource Number (ARN)，例如 `arn:aws:iot:us-east-1:123456789012:thing/b8e4157c98f332cffb37627f`。
- *<after>* 以及您想要列出建立之執行的時間，例如 `2024-11-01T03:00`。

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

執行此命令會產生回應，其中包含您建立的命令執行清單、執行開始執行的時間，以及執行完成的時間。它也提供狀態資訊，以及包含狀態額外資訊的`statusReason`物件。

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

刪除命令執行

如果您不想再使用命令執行，可以從您的帳戶永久移除它。

> **Note**
>
> 命令執行只有在進入終端狀態時才能刪除，例如 SUCCEEDED、 FAILED或 REJECTED。

下列範例示範如何使用 命令刪除`delete-command-execution` AWS CLI 命令執行。*<execution-id>* 將 取代為您要刪除之命令執行的識別符。

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

如果 API 請求成功，則命令執行會產生 200 狀態碼。您可以使用 `GetCommandExecution` API 來驗證 帳戶中不再存在命令執行。

範例：使用命令控制車輛轉向模式 (AWS CLI)

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列範例示範如何使用 命令功能 AWS CLI。此範例使用 AWS IoT FleetWise 車輛做為目標裝置，示範如何傳送命令以遠端控制轉向模式。

主題

- 車輛轉向模式範例概觀
- 先決條件
- 使用遠端命令的 IAM 政策
- 執行 AWS IoT 命令 (AWS CLI)
- 清除

車輛轉向模式範例概觀

在此範例中，您將：

1. 使用 為 操作建立命令資源，`create-command` AWS CLI 以變更車輛的轉向模式。

2. 擷取命令的相關資訊，例如使用 建立或上次更新命令的時間`get-command` AWS CLI。
3. 使用 將 命令傳送至車輛，`start-command-execution` AWS CLI 並以 轉向模式做為強制性參數，然後會在裝置上執行該參數。
4. 使用 取得命令執行的結果`get-command-execution` AWS CLI。您可以檢查執行何時完成，並擷取執行結果等其他詳細資訊，以及完成執行命令所需的時間。
5. 移除您不想再使用的任何命令和命令執行，以執行清除活動。

先決條件

在您執行此範例之前：

- 將 AWS IoT FleetWise 車輛佈建為 AWS IoT 登錄檔中的 AWS IoT 物件。您還必須將憑證新增至物件並啟用，然後將政策連接至物件。然後，您的裝置可以連線到雲端並執行命令。如需詳細資訊，請參閱佈建 車輛。
- 建立 IAM 使用者和 IAM 政策，授予您使用 命令執行 API 操作的許可，如 所示使用遠端命令的 IAM 政策。

使用遠端命令的 IAM 政策

下表顯示範例 IAM 政策，授予對命令功能之所有控制平面和資料平面 API 操作的存取權。應用程式的使用者將具有執行所有遠端命令 API 操作的許可，如下表所示。

API 操作

API 動作	控制/資料平面	通訊協定	描述	資源
`CreateCommand`	控制平台	HTTP	建立命令資源	• command
`GetCommand`	控制平台	HTTP	擷取命令的相關資訊	• command
`UpdateCommand`	控制平台	HTTP	更新命令或 的相關資訊以取代命令	• command
`ListCommands`	控制平台	HTTP	列出您帳戶中的命令	• command
`DeleteCommand`	控制平台	HTTP	刪除命令	• command

API 動作	控制/資料平面	通訊協定	描述	資源
StartCommandExecution	資料平面	HTTP	開始執行命令	• command • 物件
UpdateCommandExecution	資料平面	MQTT	更新命令執行	• command • 物件
GetCommandExecution	控制平台	HTTP	擷取命令執行的相關資訊	• command • 物件
ListCommandExecutions	控制平台	HTTP	列出您帳戶中的命令執行	• command • 物件
DeleteCommandExecution	控制平台	HTTP	刪除命令執行	• command • 物件

在此範例中，取代：

- *region* 您的 AWS 區域，例如 ap-south-1。
- *account-id* 您的 AWS 帳戶 號碼，例如 57EXAMPLE833。
- *command-id*、 *command-id1*和 *command-id2*，以及您的唯一命令識別符，例如 LockDoor或 TurnOffAC。
- *thing-name* 您的 AWS IoT 物件名稱，例如 my_car。

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
        "Action": [
            "iot:CreateCommand",
            "iot:GetCommand",
            "iot:ListCommands",
            "iot:UpdateCommand",
            "iot:DeleteCommand"
```

```
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id1",
            "arn:aws:iot:<region>:<account-id>:command/command-id2",
        ]
    },
    {
        "Action": [
            "iot:GetCommandExecution",
            "iot:ListCommandExecutions",
            "iot:DeleteCommandExecution"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
        ]
    },
    {
        "Action": "iot:StartCommandExecution",
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
        ]
    }
  ]
}
```

執行 AWS IoT 命令 (AWS CLI)

以下說明如何使用 AWS CLI 執行命令操作和變更車輛轉向模式。

1. 建立轉向模式操作的命令資源

   使用 `create-command`CLI 建立您要傳送到裝置的命令。在此範例中，指定：

   - `command-id` 作為 *`TurnOffSteeringMode`*
   - `role-arn` 因為`role-arn`必須提供 `"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"` ，因為它是 IAM 角色，

授予在您的車輛上建立和執行命令的許可。如需詳細資訊，請參閱授予使用 產生命令承載的 AWS IoT Device Management 許可 AWS IoT FleetWise。

- `display-name` 為 "*Turn off steering mode*"
- `namespace` 必須 `AWS-IoT-FleetWise`
- `mandatory-parameters` 作為名稱值對，其中 "*$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode*" name和 defaultValue 作為 *{ "S": "true" }*

> **Note**
>
> 您也可以建立命令，而無需指定任何必要參數。然後，您必須指定使用 CLI `start-command-execution` 執行命令時要使用的參數。如需範例，請參閱「命令使用案例」。

> **Important**
>
> 使用 `AWS-IoT-FleetWise` 命名空間時，您必須確保指定為 一部分Name的欄位mandatory-parameters使用 `$actuatorPath.`字首，而 `Value` 欄位必須使用 字串資料類型。

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

下列輸出顯示來自 CLI 的範例回應，其中 `ap-south-1`和 `123456789012`是 AWS 區域 和 AWS 帳戶 ID 的範例。

```
{
```

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

如需使用此命令的其他範例，請參閱 建立命令資源。

2. 擷取命令的相關資訊

執行下列命令來擷取命令的相關資訊，其中 `command-id`是從`create-command`上方操作輸出中的命令 ID。

> **Note**
>
> 如果您建立多個命令，您可以使用 `ListCommands` API 列出帳戶中的所有命令，然後使用 `GetCommand` API 取得特定命令的其他資訊。如需詳細資訊，請參閱列出您帳戶中的命令。

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

執行此命令會產生下列回應。您將看到建立命令的時間、上次更新的時間、您指定的任何參數，以及命令是否可在裝置上執行。

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

如需使用此命令的其他範例，請參閱 擷取命令的相關資訊。

3. 啟動命令執行

   執行下列命令以開始執行命令，其中 `command-arn`是從`get-command`上方操作輸出中的命令 ARN。`target-arn` 是您執行命令的目標裝置的 ARN，例如 *`myVehicle`*。

   在此範例中，由於您在建立命令時提供參數的預設值，因此 `start-command-execution` CLI 可以在執行命令時使用這些值。您也可以選擇在使用 CLI 時，透過為參數指定不同的值來覆寫預設值。

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

   執行此命令會傳回命令執行 ID。您可以使用此 ID 來查詢命令執行狀態、詳細資訊和命令執行歷史記錄。

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

   如需使用 CLI 的其他範例，請參閱 傳送命令 (AWS CLI)。

4. 擷取命令執行的相關資訊

   執行下列命令，以擷取您在目標裝置上執行之命令的相關資訊。指定`execution-id`您從上方取得做為`start-command-execution`操作輸出的 ，以及 `target-arn`，這是您要鎖定之裝置的 ARN。

> **Note**
>
> - 若要取得最新狀態資訊，您的裝置必須使用 MQTT API 將更新的狀態資訊發佈至命令的 `UpdateCommandExecution` MQTT 預留回應主題。如需詳細資訊，請參閱更新命令執行結果。
> - 如果您啟動多個命令執行，您可以使用 `ListCommandExecutions` API 列出帳戶中的所有命令執行，然後使用 `GetCommandExecution` API 取得特定執行的其他資訊。如需詳細資訊，請參閱列出您帳戶中的命令執行。

```
aws iot get-command-execution \
    --execution-id <"07e4b780-7eca-4ffd-b772-b76358da5542"> \
    --target-arn arn:aws:iot:<region>:<account>:thing/myVehicle
```

執行此命令會傳回命令執行、執行狀態、開始執行的時間，以及完成時間的相關資訊。例如，以下回應顯示命令在目標裝置上成功執行，且轉向模式已關閉。

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myVehicle",
    "result": "SUCCEEDED",
     "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "result": {
        "KeyName": {
            "S": "",
            "B": true,
            "BIN": null
        }
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode":
         { "S": "true" }
    }'
}
```

清除

現在您已建立命令並在裝置上執行，如果您不想再使用此命令，可以將其刪除。任何進行中的待定命令執行都會繼續執行，而不會受到刪除請求的影響。

> **Note**
> 或者，如果命令已過期，而且稍後可能需要用它在目標裝置上執行，您也可以棄用命令。

1. （選用）棄用命令資源

   執行下列命令以取代命令，其中 `command-id`是從`get-command`上方操作輸出中的命令 ID。

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

   執行此命令會傳回輸出，顯示命令已棄用。您也可以使用 CLI 還原命令。

> **Note**
> 您也可以使用 `update-command` CLI 來更新命令的顯示名稱和描述。如需其他資訊，請參閱 更新或取代命令資源。

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

2. 刪除命令

   執行下列命令來刪除由 指定的命令`command-id`。

> **Note**
> 刪除動作是永久的，無法復原。

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

如果刪除請求成功，您會看到 HTTP `statusCode`為 202 或 204，這取決於您是否將命令標記為棄用，以及何時棄用。如需詳細資訊和範例，請參閱 刪除命令資源。

您可以使用 `get-command` CLI 來驗證命令是否已從您的帳戶中移除。

3. （選用）刪除命令執行

   根據預設，所有命令執行都會在您建立後的六個月內刪除。您可以使用 `GetCommandExecution` API 中的 `timeToLive` 參數來檢視此資訊。

   或者，如果您的命令執行變成結束，例如當您的執行狀態為 `SUCCEEDED`、 `FAILED`或 之一時`REJECTED`，您可以刪除命令執行。執行下列命令來刪除執行，其中 `execution-id`是上述`get-command-execution`操作輸出中的執行 ID。

```
aws iot delete-command-execution \
            --execution-id "07e4b780-7eca-4ffd-b772-b76358da5542"
```

   您可以使用 `get-command-execution` CLI 來驗證命令執行是否已從您的帳戶中移除。

命令使用案例

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

使用命令功能時，您可以在下列案例中建立和執行命令：

- 您可以在建立期間省略參數，並僅指定命令 ID。在此情況下，您需要指定在目標裝置上執行 命令時要使用的參數。
- 您可以指定一或多個參數，並在建立命令時為其設定預設值。提供預設值有助於防止您傳送不正確的命令。
- 您可以指定一或多個參數，並在建立命令時為其設定值。可以提供多個參數，但只會執行其中一個參數，而且此參數`Name`的欄位必須使用 `$actuatorPath`字首。

本節提供 `CreateCommand`和 `StartCommandExecution` API 以及使用 參數的一些使用案例。它還顯示一些使用 命令搭配狀態範本的範例。

主題

- 建立不含參數的命令
- 建立具有參數預設值的命令
- 使用參數值建立命令
- 搭配狀態範本使用命令

建立不含參數的命令

下列使用案例顯示如何使用 `CreateCommand` API 或 `create-command` CLI 來建立不含參數的命令。當您建立命令時，只需要提供命令 ID 和角色 ARN。

此使用案例在經常性使用案例中特別有用，例如當您想要多次將相同的命令傳送到車輛時。在此情況下， 命令不會繫結至特定作動器，並可讓您彈性地在任何作動器上執行命令。使用 `StartCommandExecution` API 或 `start-command-execution`CLI 執行命令時，您必須在執行時間指定參數，其中包含致動器和實體訊號值。

建立不含**`mandatory-parameters`**輸入的命令

此使用案例示範如何在沒有任何必要參數輸入的情況下建立命令。

```
aws iot create-command \
    --command-id "UserJourney1" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --description "UserJourney1 - No mandatory parameters" \
    --namespace "AWS-IoT-FleetWise"
```

執行在沒有**`mandatory-parameters`**輸入的情況下建立的命令

在此第一個範例中，上述建立的 命令可讓您在任何 傳動器上執行命令，而不受限制。若要`actuator1`將 設定為 10 的值，請執行：

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
```

```
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

同樣地，您可以執行 命令，將 actuator3設定為 的值true。

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator3": {"S": "true"}
    }'
```

建立具有參數預設值的命令

此命令只允許您在指定的作動器上執行命令。提供預設值有助於防止您傳送不正確的命令。例如，可以使用預設值來設定鎖定和解鎖門的LockDoor命令，以避免命令意外解鎖門。

當您想要多次傳送相同的命令，並在相同的作動器上執行不同的動作，例如鎖定和解鎖車輛的門時，此使用案例特別有用。如果您想要將致動器設定為預設值，則不需要將 qny 傳遞parameters給 start-command-execution CLI。如果您在 CLI start-command-execution parameters中為指定不同的值，它會覆寫預設值。

使用 的預設值建立命令 **mandatory-parameters**

下列命令顯示如何為 致動器 1 提供預設值。

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1",
            "defaultValue": {"S": "0"}
        }
    ]'
```

執行使用預設值為 建立的命令 **mandatory-parameters**

命令UserJourney2可讓您執行命令，而不需要在執行時間傳遞輸入值。在此情況下，執行時間的執行將使用建立期間指定的預設值。

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle
```

您也可以在執行時間期間，為相同的作動器 - 作動器 1 傳遞不同的值，這會覆寫預設值。

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "139"}
    }'
```

使用參數值建立命令

此命令只允許您在指定的作動器上執行命令。它也會強制您在執行時間期間設定致動器的值。

當您希望最終使用者在車輛上執行某些特定的動作時，此使用案例特別有用。

> **Note**
>
> 您可以有超過 個`mandatory-parameters`輸入的名稱值對，其中有部分或全部的預設值。然後，在執行時間，您可以判斷要在啟動器上執行時使用的參數，前提是啟動器名稱使用完整名稱與`$actuatorPath.`字首。

建立不含預設值的 命令 `mandatory-parameters`

此命令只允許您在指定的作動器上執行命令。它也會強制您在執行時間期間設定致動器的值。

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1"
        }
    ]'
```

執行在沒有預設值的情況下為 建立的命令 **mandatory-parameters**

執行 命令時，在此情況下，您必須指定 致動器 1 的值。以下顯示的命令執行會成功將 的值設定為 `actuator1` `10`。

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney2 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

搭配狀態範本使用命令

您也可以使用命令 API 操作進行狀態資料收集和處理。例如，您可以擷取一次性狀態快照，或啟用或停用狀態範本，以開始或停止收集車輛狀態資料。下列範例示範如何搭配狀態範本使用命令功能。如需詳細資訊，請參閱資料收集和處理的狀態範本操作

> **Note**
>
> 指定為`mandatory-parameters`輸入一部分的名稱欄位必須使用 `$stateTemplate`字首。

範例 1：使用預設值為狀態範本建立命令

此範例示範如何使用 `create-command` CLI 來啟用狀態範本。

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name"
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      }
    ]'
```

同樣地，以下命令顯示如何將 `start-command-execution` CLI 用於狀態範本的範例。

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
       "$stateTemplate.name": {"S": "ST345"}
    }'
```

範例 2：為沒有預設值的狀態範本建立命令

下列命令會建立多個狀態範本，而沒有任何參數的預設值。它會強制您使用這些參數及其值來執行 命令。

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name",
          "defaultValue": {"S": "ST123"}
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      },
      {
          "name": "$stateTemplate.deactivateAfterSeconds",
          "defaultValue": {"L": "120"}
      }
    ]'
```

下列命令顯示如何針對上述範例使用 `start-command-execution` CLI。

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
        "$stateTemplate.name": {"S": "ST345"},
        "$stateTemplate.operation": {"S": "activate"},
        "$stateTemplate.deactivateAfterSeconds" : {"L": "120"}
```

監控您車輛的最後已知狀態

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以建立狀態範本並將其與 車輛建立關聯，以近乎即時的方式監控車輛的最後已知狀態。與狀態範本相關聯的車輛會使用 `onChange`或`periodic`更新策略串流遙測資料。使用變更時更新策略，相關車輛會在發生變更時串流遙測資料。在定期更新策略期間，相關聯的車輛會在指定的期間內串流遙測資料。

透過隨需操作，您可以一次請求目前的車輛狀態（擷取）。您也可以啟用或停用先前部署的狀態範本，以開始或停止報告車輛狀態資料。使用 AWS IoT 命令 APIs執行上次已知狀態操作。

每個狀態範本都包含下列資訊。

`name`

狀態範本的唯一別名。

`signalCatalogArn`

與狀態範本相關聯之訊號目錄的 Amazon Resource Name (ARN)。

`stateTemplateProperties`

收集資料的訊號清單。狀態範本屬性會決定車輛傳送至雲端的特定訊號更新。

`dataExtraDimensions`

要包含在通訊協定緩衝區 (Protobuf) 編碼處理資料中的車輛屬性清單。

`metadataExtraDimensions`

以 MQTT 5 使用者屬性形式使用已處理資料發佈的車輛屬性清單。

`id`

唯一、服務產生的識別符。

如需使用 Edge Agent for AWS IoT FleetWise 軟體收集車輛傳送資料的方法，請參閱 使用 MQTT 訊息處理上次已知狀態車輛資料。如需如何將狀態範本與車輛建立關聯的詳細資訊，請參閱 建立 AWS IoT FleetWise 車輛。

主題

- 建立 AWS IoT FleetWise 狀態範本
- 更新 an AWS IoT FleetWise 狀態範本
- 刪除 AWS IoT FleetWise 狀態範本
- Get AWS IoT FleetWise 狀態範本資訊
- 資料收集和處理的狀態範本操作

建立 AWS IoT FleetWise 狀態範本

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 AWS IoT FleetWise API 或主控台來建立狀態範本。狀態範本提供追蹤 車輛狀態的機制。在車輛上執行的 Edge Agent for AWS IoT FleetWise 軟體會收集並傳送訊號更新至雲端。

建立狀態範本 （主控台）

您可以使用 AWS IoT FleetWise 主控台來建立狀態範本。

建立狀態範本

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇狀態範本。
3. 在狀態範本頁面上，選擇建立狀態範本。
4. 在狀態範本詳細資訊中，輸入狀態範本的名稱，並選擇性地輸入描述。
5. 在選擇訊號中，新增您要從中擷取車輛狀態資訊的訊號。
6. 選擇建立狀態範本。

成功建立狀態範本後，您會在狀態範本頁面上看到它。您現在可以將其與車輛建立關聯。

建立狀態範本 (AWS CLI)

您可以使用 CreateStateTemplate API 操作來建立狀態範本。下列為使用 AWS CLI的範例。

若要建立狀態範本，請執行下列命令。

將 *create-state-template* 取代為包含狀態範本組態的 .json 檔案名稱。

```
aws iotfleetwise create-state-template \
    --cli-input-json file://create-state-template.json
```

Example 狀態範本組態

`stateTemplateProperties` 應該包含訊號的完整名稱。

`dataExtraDimensions` 和 `metadataExtraDimensions`應包含車輛屬性的完整名稱。指定的維度會取代狀態範本中的任何現有維度值。

```
{
    "name": "state-template-name",
    "signalCatalogArn": "arn:aws:iotfleetwise:region:account:signal-catalog/catalog-name",
    "stateTemplateProperties": [
        "Vehicle.Signal.One",
        "Vehicle.Signal.Two"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

將 an AWS IoT FleetWise 狀態範本與 車輛建立關聯

將狀態範本與車輛建立關聯（主控台）

您可以使用 AWS IoT FleetWise 主控台，將相關聯的狀態範本新增至車輛。

建立狀態範本的關聯

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛。
3. 從清單中選擇車輛以開啟其詳細資訊頁面。
4. 在狀態範本索引標籤上，選擇管理狀態範本。
5. 選擇新增狀態範本。
6. 選取狀態範本，然後選擇其報告方法。
   a. 變更時 – 狀態範本會報告車輛狀態的變更。
   b. 定期 – 狀態範本會報告指定時間間隔的更新。
7. 選擇儲存變更。

將 an AWS IoT FleetWise 狀態範本與 車輛建立關聯 (AWS CLI)

將建立的狀態範本與車輛建立關聯，以允許從車輛到雲端的狀態更新集合。若要這樣做，請使用：

- 建立車輛時，請使用 `create-vehicle`命令的 `stateTemplates` 欄位。如需詳細資訊，請參閱建立 AWS IoT FleetWise 車輛。
- 更新車輛時，請使用 `update-vehicle`命令的 `stateTemplatesToAdd`或 `stateTemplatesToRemove` 欄位。如需詳細資訊，請參閱更新 an AWS IoT FleetWise 車輛。

更新 an AWS IoT FleetWise 狀態範本

> **Important**
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 UpdateStateTemplate API 操作或 AWS IoT FleetWise 主控台來更新現有的狀態範本。

更新狀態範本 （主控台）

若要從主控台更新狀態範本，請前往 AWS IoT FleetWise 主控台的狀態範本頁面，並執行下列步驟。

1. 選擇您要更新的狀態範本，然後選擇編輯。
2. 編輯狀態範本詳細資訊，然後選擇儲存變更。

更新狀態範本 (AWS CLI)

若要更新狀態範本，請執行下列命令。

以包含狀態範本組態的 .json 檔案名稱取代 *update-state-template*。

```
aws iotfleetwise update-state-template \
    --cli-input-json file://update-state-template.json
```

Example 狀態範本組態

`stateTemplateProperties` 應該包含訊號的完整名稱。

`dataExtraDimensions` 和 `metadataExtraDimensions`應包含車輛屬性的完整名稱。

```
{
    "identifier": "state-template-name",
    "stateTemplatePropertiesToAdd": [
        "Vehicle.Signal.Three"
    ],
    "stateTemplatePropertiesToRemove": [
        "Vehicle.Signal.One"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

刪除 AWS IoT FleetWise 狀態範本

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 DeleteStateTemplate API 操作或 AWS IoT FleetWise 主控台來刪除狀態範本。

刪除狀態範本（主控台）

若要從主控台刪除狀態範本，請前往 AWS IoT FleetWise 主控台的狀態範本頁面，並執行下列步驟。

1. 選擇您要刪除的狀態範本，然後選擇刪除。
2. 確認您想要刪除狀態範本，然後選擇刪除。

刪除狀態範本 (AWS CLI)

若要刪除狀態範本，請執行下列命令。

將###取代為狀態範本的名稱或 ID。

```
aws iotfleetwise delete-state-template \
    --identifier idenitfier
```

Get AWS IoT FleetWise 狀態範本資訊

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用 GetStateTemplate API 操作來擷取狀態範本的相關資訊。下列為使用 AWS CLI的範例。

將###取代為狀態範本的名稱。

```
aws iotfleetwise get-state-template \
    --identifier idenitfier
```

您可以使用 ListStateTemplates API 操作來擷取您建立的狀態範本清單。下列為使用 AWS CLI的範例。

```
aws iotfleetwise list-state-templates
```

如果您使用客戶受管 AWS KMS 金鑰啟用加密，請包含下列政策陳述式，讓您的角色可以叫用 `GetStateTemplate`或 `ListStateTemplates` API 操作。

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

資料收集和處理的狀態範本操作

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列各節說明如何使用狀態範本來啟用和停用資料收集、執行擷取操作，以及處理您車輛的狀態資料。

主題

- 使用狀態範本啟用和停用狀態資料收集
- 使用狀態範本擷取車輛狀態快照
- 使用 MQTT 訊息處理上次已知狀態車輛資料

使用狀態範本啟用和停用狀態資料收集

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列各節說明如何使用 啟用和停用狀態範本的資料擷取 AWS CLI。

> **Important**
>
> 開始之前，請確定您已建立狀態範本，並將其及其更新策略與車輛建立關聯。

您必須啟用狀態範本， Edge Agent 才能將訊號更新傳送至雲端。

若要使用狀態範本執行這些操作，請先建立命令資源，然後在車輛上啟動命令執行。下一節說明如何使用此 API，以及如何啟用和停用資料擷取。

主題

- 使用 CreateCommand API
- 範例：啟用狀態範本
- 範例：停用狀態範本

使用 `CreateCommand` API

在「`AWS-IoTFleetwise`」命名空間中建立命令資源，並在為狀態範本建立或傳送命令資源時使用下列參數：

- `$stateTemplate.name` – 要在其中執行操作的狀態範本名稱。狀態範本必須套用至車輛，您才能執行 操作。如需詳細資訊，請參閱將 an AWS IoT FleetWise 狀態範本與 車輛建立關聯。

- `$stateTemplate.operation` – 要在狀態範本上執行的操作。針對此參數使用下列其中一個值：
  - `activate` – 當您將狀態範本套用至車輛時，Edge Agent 會根據`stateTemplateUpdateStrategy`您指定的（變更中或定期）開始傳送訊號更新至雲端。如需詳細資訊，請參閱將 an AWS IoT FleetWise 狀態範本與 車輛建立關聯。

    此外，您可以定義自動狀態範本停用時間，在指定的期間之後停止更新。如果未提供自動停用時間，狀態範本會持續傳送更新，直到發出停用呼叫為止。

    一旦收到`activate`命令，裝置應根據更新策略傳送狀態範本中指定的訊號。 AWS IoT FleetWise 建議當裝置收到啟用命令時，其傳送的第一個訊息應包含狀態範本中所有訊號的快照。後續的訊息應根據更新策略傳送。
  - `deactivate` – Edge Agent 會停止傳送訊號更新至雲端。
  - `fetchSnapshot` – Edge Agent 會傳送狀態範本中定義的訊號的一次性快照`stateTemplateUpdateStrategy`，無論您在將狀態範本套用至車輛時指定的 為何。
- （選用）`$stateTemplate.deactivateAfterSeconds` – 狀態範本會在指定的時間之後自動停用。只有在 參數的值為「啟用」時，才能使用此`$stateTemplate.operation`參數。如果未指定此參數，或如果此參數的值為 0，則 Edge Agent 會持續將訊號更新傳送至雲端，直到狀態範本收到「停用」操作為止。狀態範本永遠不會自動停用。

  最小值：0，最大值：4294967295。

> **Note**
>
> - API 會傳回成功，以回應已處於作用中狀態之範本的啟用請求。
> - API 會傳回成功，以回應已處於停用狀態之範本的停用請求。
> - 您在狀態範本上提出的最新請求是生效的請求。例如，如果您在一小時內提出停用狀態範本的請求，然後在四小時內提出第二個停用相同範本的請求，四小時停用就會生效，因為它是最新的請求。

> **Important**
>
> 驗證例外狀況可能發生在下列任一情況下：
>
> - 提供的狀態範本並非`ASSOCIATED`搭配 車輛使用。
> - 提出啟用狀態範本的請求，但尚未`DEPLOYED`在車輛上。

- 系統會向狀態範本提出請求，但正在DELETED車輛上。

範例：啟用狀態範本

若要啟用狀態範本，請先建立命令資源。然後，您可以將下列命令傳送至您要啟用狀態範本的車輛。此範例示範如何在建立命令時指定參數的預設值。啟動命令執行時，會使用這些參數及其值來啟用狀態範本。

1. 建立命令資源

   您必須先建立命令資源，才能將命令傳送至車輛。當您將命令傳送至車輛時，可以指定強制參數的替代值。如需詳細資訊，請參閱建立命令資源。

   > **Important**
   >
   > `$stateTemplate.name` 和 `$stateTemplate.operation` 參數必須以字串資料類型提供。如果提供任何其他資料類型，或者如果缺少這兩個參數中的任何一個，則命令執行會失敗，並出現驗證例外狀況。`$stateTemplate.deactivateAfterSeconds` 參數必須以Long資料類型提供。

```
aws iot create-command \
    --description "This command activates a state template on a vehicle"
    --command-id ActivateStateTemplate \
    --display-name "Activate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "activate"}
    },
    {
        "name": "$stateTemplate.deactivateAfterSeconds",
        "defaultValue": {"L": "120"}
    }
```

```
]'
```

2. 在車輛上啟動命令執行

   建立命令後，將命令傳送至車輛。如果您在建立命令資源時未指定必要參數的值，您現在必須指定這些參數。如需詳細資訊，請參閱傳送命令 (AWS CLI)。

   > **Important**
   > 請確定您使用帳戶特定的 AWS IoT 任務資料平面 API 端點進行 API 操作。

```
aws iot-jobs-data start-command-execution \
    --endpoint-url <endpoint-url> \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/ActivateStateTemplate \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

3. 擷取狀態範本操作的狀態

   啟動命令執行後，您可以使用 `GetCommandExecution` API 來擷取狀態範本。

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

範例：停用狀態範本

若要停用狀態範本，請先建立命令資源。然後，您可以將下列命令傳送至您要停用狀態範本的車輛。此範例示範如何在建立命令時指定參數的預設值。啟動命令執行時，會使用這些參數及其值來停用狀態範本。

1. 建立命令資源

   您必須先建立命令資源，才能將命令傳送至車輛。當您將命令傳送至車輛時，可以指定強制參數的替代值。如需詳細資訊，請參閱建立命令資源。

```
aws iot create-command \
    --description "This command deactivates a state template on a vehicle"
    --command-id DeactivateStateTemplate \
    --display-name "Deactivate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
```

```
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "deactivate"}
    }
]'
```

2. 在車輛上啟動命令執行

   建立命令後，將命令傳送至車輛。如果您在建立命令資源時未指定必要參數的值，您現在必須指定這些參數。如需詳細資訊，請參閱傳送命令 (AWS CLI)。

```
aws iot-jobs-data start-command-execution \
    --endpoint-url <endpoint-url> \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/DeactivateStateTemplate
 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

3. 擷取狀態範本操作的狀態

   啟動命令執行後，您可以使用 `GetCommandExecution` API 來擷取狀態範本。

```
aws iot get-command-execution  --execution-id <EXECUTION_ID>
```

使用狀態範本擷取車輛狀態快照

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

您可以使用`CreateCommand` AWS IoT Core 控制平面 API 操作或 AWS IoT FleetWise 主控台擷取車輛的上次已知狀態。

> **Important**
>
> 驗證例外狀況可能發生在下列任一情況下：
>
> - 提供的狀態範本並非ASSOCIATED搭配 車輛使用。
> - 提出啟用狀態範本的請求，但尚未DEPLOYED在車輛上。
> - 系統會向狀態範本提出請求，但正在DELETED車輛上。

擷取車輛狀態快照（主控台）

您可以使用 AWS IoT FleetWise 主控台來擷取車輛的上次已知狀態。 AWS IoT FleetWise 會建立命令，供您擷取資料。

擷取車輛狀態

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 車輛。
3. 從清單中選擇車輛以開啟其詳細資訊頁面。
4. 在狀態範本索引標籤上，選擇擷取資料。
5. 選取授予 AWS IoT FleetWise 許可的 IAM 角色，以傳送命令並擷取資料。請參閱控制存取。
6. 選擇擷取狀態。

擷取車輛狀態快照 (AWS CLI)

若要擷取狀態快照，請先建立命令資源。然後，您可以將下列命令傳送至您要為其擷取狀態快照的車輛。如需使用 `CreateCommand` API 及其參數的詳細資訊，請參閱 使用 `CreateCommand` API。

1. 建立命令資源

   下列範例示範如何建立命令資源來執行擷取操作。當您將命令傳送至車輛時，可以指定強制參數的替代值。如需詳細資訊，請參閱建立命令資源。

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "FetchSnapshot State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
```

```
      {
          "name": "$stateTemplate.name",
          "defaultValue": {"S": "ST123"}
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "fetchSnapshot"}
      }
    ]'
```

回應：

```
{
    "commandId": "<COMMAND_ID>",
    "commandArn": "arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID>"
}
```

2. 開始命令執行以擷取狀態快照

   建立命令後，將命令傳送至車輛。如果您在建立命令資源時未指定必要參數的值，您現在必須指定這些參數。如需詳細資訊，請參閱傳送命令 (AWS CLI)。

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

   回應：

```
{
    "executionId": "<UNIQUE_UUID>"
}
```

3. 擷取狀態範本操作的狀態

   啟動命令執行後，您可以使用 `GetCommandExecution` API 來擷取狀態範本。

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

使用 MQTT 訊息處理上次已知狀態車輛資料

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

若要從您的車輛接收更新並處理其資料，請訂閱下列 MQTT 主題。如需詳細資訊，請參閱《 開發人員指南》中的 MQTT 主題。 AWS IoT Core

```
$aws/iotfleetwise/vehicles/$vehicle_name/last_known_state/$state_template_name/data
```

因為 MQTT 不保證排序，所以最後的已知狀態訊號更新訊息可能會未依順序接收。任何使用 MQTT 接收和處理車輛資料的用戶端都必須處理此問題。最後一個已知狀態訊號更新訊息遵循 MQTT 5 訊息通訊協定。

每個 MQTT 訊息的訊息標頭具有下列使用者屬性：

- vehicleName – 車輛的唯一識別符。
- stateTemplateName – 上次已知狀態範本的唯一識別符。

此外，您可以在更新或建立狀態範本時指定`metadataExtraDimensions`請求參數，以指定要包含在 MQTT 訊息標頭中的車輛屬性。（請參閱狀態範本。）

MQTT 訊息標頭中的使用者屬性有助於將訊息路由到不同的目的地，而無需檢查承載。

MQTT 訊息承載包含從車輛收集的資料。您可以在建立或更新狀態範本時指定`extraDimensions`請求參數，以指定要包含在 MQTT 訊息承載中的車輛屬性（請參閱 建立 AWS IoT FleetWise 狀態範本)。額外維度會透過將額外維度與車輛建立關聯，來豐富從車輛收集的資料。

MQTT 訊息承載是通訊協定緩衝區 (Protobuf) 編碼，MQTT 訊息標頭包含定義為 application/octet-stream 的內容類型指標。Protobuf 編碼結構描述如下所示：

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

syntax = "proto3";

option java_package = "com.amazonaws.iot.autobahn.schemas.lastknownstate";
```

```
package Aws.IoTFleetWise.Schemas.CustomerMessage;

message LastKnownState {

  /*
   * The absolute timestamp in milliseconds since Unix Epoch of when the event was
 triggered in vehicle.
   */
  uint64 time_ms = 1;

  /*
   * This field is deprecated, use signals instead
   */
  repeated Signal signal = 2 [ deprecated = true ];

  repeated Signal signals = 3;

  repeated ExtraDimension extra_dimensions = 4;
}

message Signal {

  /*
   * The Fully Qualified Name of the signal is the path to the signal plus the signal's
 name.
   * For example, Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  /*
   * The FWE reported signal value can be one of the following data types.
   */
  oneof SignalValue {
    double double_value = 2;

    bool boolean_value = 3;

    sint32 int8_value = 4;

    uint32 uint8_value = 5;

    sint32 int16_value = 6;
```

```
    uint32 uint16_value = 7;

    sint32 int32_value = 8;

    uint32 uint32_value = 9;

    sint64 int64_value = 10;

    uint64 uint64_value = 11;

    float float_value = 12;
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 13;
  }
}

message ExtraDimension {
  /*
   * The Fully Qualified Name of the attribute is the path to the attribute plus the
 attribute's name.
   * For example, Vehicle.Model.Color
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  oneof ExtraDimensionValue {
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 2;
  }
}
```

其中：

- `time_ms`:

  在車輛中觸發事件時的絕對時間戳記（自 Unix Epoch 起以毫秒為單位）。Edge Agent 軟體會針對此時間戳記，在車輛的時鐘上使用。

- signal:

  Signal包含訊號資訊的 陣列： name（字串） signalValue 和 支援下列資料類型 - double、bool、int8、uint8、int16uint16、、int32uint32int64、、uint64、float、、str

- extra_dimensions:

  ExtraDimensions 包含車輛屬性資訊的 陣列： name（字串）extraDimensionValue，目前僅支援 string資料類型。

教學課程：使用自訂解碼界面設定網路無關的資料收集

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

簡介

本教學課程概述如何設定 AWS IoT FleetWise，以使用利用自訂解碼界面的網路無關資料收集來收集資料和執行命令。透過網路無關的資料收集，您可以使用自己的方法來解碼訊號，然後再將訊號傳送到指定的資料目的地。這可節省時間，因為您不需要特別為 AWS IoT FleetWise 建立訊號解碼器。您可以使用自己的實作來解碼訊號子集，也可以在建立或更新解碼器資訊清單`defaultForUnmappedSignals`時使用。這也可讓您靈活地收集車輛中各種來源的訊號和觸發條件。

本教學課程適用於不在標準控制器區域網路 (CAN 匯流排） 界面上的車輛訊號。例如，以自訂車輛格式或配置編碼的資料。

環境設定

本教學課程假設您已完成設定環境以存取 AWS IoT FleetWise 雲端，以及 Edge 實作 APIs和程式碼庫的步驟。

資料模型

下一節說明如何使用自訂解碼界面建立車輛屬性的模型。這適用於資料收集以及命令使用案例。它也適用於車輛中使用的任何基礎資料來源建模，例如 IDLs。

在此範例中，有兩種車輛屬性：要收集的車輛感應器（目前車輛位置）和要遠端控制的車輛傳動器（冷氣器）。這兩個都在此結構描述中定義：

```
// Vehicle WGS84 Coordinates
double Latitude;
double Longitude;

// Vehicle AC
```

```
Boolean ActivateAC;
```

下一個步驟是使用自訂解碼介面 APIs 將這些定義匯入 AWS IoT FleetWise。

訊號目錄更新

在訊號目錄中匯入這些定義。如果您已有 AWS IoT FleetWise 中的訊號目錄，請直接使用更新 API。如果您沒有訊號目錄，請先建立訊號目錄，然後呼叫更新 API。

首先，您必須建立這些車輛訊號的 VSS 表示法。VSS 做為分類來代表 AWS IoT FleetWise 中的車輛資料。使用下列內容建立名為 'vehicle-signals.json' 的 json 檔案：

```
// vehicle-signals.json
// Verify that branches and nodes are unique in terms of fully qualified name
// in the signal catalog.
[
 {
    "branch": {
      "fullyQualifiedName": "Vehicle",
      "description": "Vehicle Branch"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.CurrentLocation",
      "description": "CurrentLocation"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "description": "Latitude"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "description": "Longitude"
    }
  },
  {
```

```
    "actuator": {
      "fullyQualifiedName": "Vehicle.ActivateAC",
      "description": "AC Controller",
      "dataType": "BOOLEAN"
    }
  }
]
```

如果您沒有訊號目錄，則需要叫用 `create-signal-catalog`：

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise create-signal-catalog \
        --name my-signal-catalog \
        --nodes "${VEHICLE_NODES}"
```

如果您已經有訊號目錄，您可以使用 `update-signal-catalog` API 新增這些訊號：

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise update-signal-catalog \
        --name my-signal-catalog \
        --nodes-to-add "${VEHICLE_NODES}"
```

車輛模型和解碼器

在訊號目錄中插入訊號後，下一個步驟是建立車輛模型並執行個體化這些訊號。為此，您可以使用 `create-model-manifest`和 `create-decoder-manifest` APIs。

首先，格式化您要插入車輛模型的訊號名稱：

```
# Prepare the signals for insertion into the vehicle model.
VEHICLE_NODES=`cat vehicle-signals.json`
VEHICLE_NODES=`echo ${VEHICLE_NODES} | jq -r ".[] | .actuator,.sensor
 | .fullyQualifiedName" | grep Vehicle\\.`
VEHICLE_NODES=`echo "${VEHICLE_NODES}" | jq -Rn [inputs]`
# This is how the vehicle model input looks.
echo $VEHICLE_NODES
# [ "Vehicle.CurrentLocation.Latitude",
#   "Vehicle.CurrentLocation.Longitude",
#   "Vehicle.ActivateAC" ]
# Create the vehicle model with those signals.
aws iotfleetwise create-model-manifest \
    --name my-model-manifest \
```

```
    --signal-catalog-arn arn:xxxx:signal-catalog/my-signal-catalog \
    --nodes "${VEHICLE_NODES}"

# Activate the vehicle model.
 aws iotfleetwise update-model-manifest \
    --name my-model-manifest --status ACTIVE
```

現在，使用自訂解碼界面來建立解碼器資訊清單。

> **Note**
>
> 只有在您想要指定自訂 IDs 時才需要建立網路介面和訊號，這不是此範例的一部分。
> 如需在完整名稱 (FQN) 與自訂解碼訊號 ID 不同時映射解碼資訊的資訊，請參閱 Edge Agent 開發人員指南。

```
// Create a network interface that is of type : CUSTOM_DECODING_INTERFACE
// custom-interface.json
[
  {
    "interfaceId": "NAMED_SIGNAL",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  },
  {
    "interfaceId": "AC_ACTUATORS",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
// custom-decoders.json
// Refer to the fully qualified names of the signals, make them of
// type CUSTOM_DECODING_SIGNAL, and specify them as part of the same interface ID
// that was defined above.
[
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "interfaceId": "NAMED_SIGNAL",
```

```
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Longitude"
      }
    },
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Latitude"
      }
    },
    {
        "fullyQualifiedName": "Vehicle.ActivateAC",
        "interfaceId": "AC_ACTUATORS",
        "type": "CUSTOM_DECODING_SIGNAL",
        "customDecodingSignal": {
          "id": "Vehicle.ActivateAC"
        }
    }
]
# Create the decoder manifest.
 CUSTOM_INTERFACE=`cat custom-interface.json`
 CUSTOM_DECODERS=`cat custom-decoders.json`

aws iotfleetwise create-decoder-manifest \
   --name my-decoder-manifest \
   --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
   --network-interfaces "${CUSTOM_INTERFACE}" \
   --signal-decoders "${CUSTOM_DECODERS}"

 # Activate the decoder manifest.
 aws iotfleetwise update-decoder-manifest \
    --name my-decoder-manifest \
    --status ACTIVE
```

此時，您已在 AWS IoT FleetWise 中完全建立這些訊號的模型。接著建立車輛，並將其與您建立的模型建立關聯。您可以針對以下項目使用 `create-vehicle` API：

```
aws iotfleetwise create-vehicle \
    --decoder-manifest-arn arn:xxx:decoder-manifest/my-decoder-manifest \
    --association-behavior ValidateIotThingExists \
```

```
    --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
    --vehicle-name "my-vehicle"
```

下一個步驟是專注於 AWS IoT FleetWise Edge 程式碼庫，並撰寫必要的程式碼延伸。

> **Note**
>
> 如需 Edge 實作的相關資訊，請參閱 Edge Agent 開發人員指南。

傳送命令

現在，編譯軟體（確定將標頭和 C++ 檔案新增至 CMake 檔案），然後返回雲端 APIs 來測試此啟動器的命令：

```
// Create a command targeting your vehicle.
aws iot create-command --command-id activateAC \
    --namespace "AWS-IoT-Fleetwise" \
    --endpoint-url endpoint-url \
    --role-arn ${SERVICE_ROLE_ARN} \
    --mandatory-parameters '[ { "name": "$actuatorPath.Vehicle.ActivateAC",
 "defaultValue": {"B": "false"} } ]' \
// You will receive the command ARN.

{
    "commandId": "activateAC",
    "commandArn": "arn:aws:iot:xxx:command/activateAC"
}

// You can send the command to activate the AC targeting your vehicle.

JOBS_ENDPOINT_URL=`aws iot describe-endpoint --endpoint-type iot:Jobs | jq -
j .endpointAddress`
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:xxx:command/activateAC \
    --target-arn arn:xxx:vehicle/my-vehicle \
    --parameters '{ "$actuatorPath.Vehicle.ActivateAC" : {"B": "true"}}' \
    --endpoint-url https://${JOBS_ENDPOINT_URL}
// You will receive the corresponding execution ID.
{
    "executionId": "01HSK4ZH6ME7D43RB2BV8JC51D"
}
```

```
// If you have the AWS IoT FleetWise Edge Agent running, you can see the logs.
[AcCommandDispatcher.cpp:26] [setActuatorValue()]:
[Actuator Vehicle.ActivateAC executed successfully for command ID
 01HSK4ZH6ME7D43RB2BV8JC51D]
```

搭配 AWS IoT FleetWise 使用 AWS CLI 和 SDK AWS SDKs

本節提供有關提出 AWS IoT FleetWise API 請求的資訊。如需 AWS IoT FleetWise 操作和資料類型的詳細資訊，請參閱 AWS IoT FleetWise API 參考。

若要將 AWS IoT FleetWise 與各種程式設計語言搭配使用，請使用 SDKAWS SDKs，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

對於命令列存取，請使用 AWS IoT FleetWise 搭配 AWS CLI。您可以從命令列控制 AWS IoT FleetWise 和其他 服務，並透過指令碼自動化。

故障診斷 AWS IoT FleetWise

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

以下資訊可協助您對 AWS IoT FleetWise 的常見問題進行故障診斷。

主題

- AWS IoT FleetWise 解碼器資訊清單問題
- Edge Agent for AWS IoT FleetWise 軟體問題
- 存放和轉送問題

AWS IoT FleetWise 解碼器資訊清單問題

故障診斷解碼器資訊清單問題。

診斷解碼器資訊清單 API 呼叫

錯誤	準則疑難排解
`UpdateOperationFailure.ConflictingDecoderUpdate`	相同的解碼器資訊清單具有多個更新請求。請稍候，然後再試一次。
`UpdateOperationFailure.InternalFailure`	InternalFailure 會啟動為封裝的例外狀況。問題本身取決於封裝的例外狀況。
`UpdateOperationFailure.ActiveDecoderUpdate`	解碼器資訊清單處於 `Active` 狀態，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 正嘗試驗證其他人正在修改的車輛模型（模型資訊清單）。請稍候，然後再試一次。
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND`	車輛模型沒有任何與其相關聯的訊號。將訊號新增至車輛模型，並確認訊號可在相關聯的訊號目錄中找到。

錯誤	準則疑難排解
UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE	更新車輛模型使其處於 ACTIVE 狀態，然後再試一次。
UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_FOUND	AWS IoT FleetWise 找不到與解碼器資訊清單相關聯的車輛模型。驗證車輛模型的 Amazon Resource Name (ARN)，然後再試一次。
UpdateOperationFailure.ModelManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE	車輛模型的驗證失敗，因為在訊號目錄中找不到來自車輛模型的訊號名稱。確認車輛模型中的訊號都包含在相關聯的訊號目錄中。
UpdateOperationFailure.ValidationFailure	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。確認例外狀況傳回的所有訊號和網路介面都存在、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訊號的介面。
UpdateOperationFailure.KmsKeyAccessDenied	用於 操作的 AWS Key Management Service (AWS KMS) 金鑰發生許可問題。請確認您使用的角色可存取金鑰，然後再試一次。
UpdateOperationFailure.DecoderDoesNotExist	解碼器資訊清單不存在。驗證解碼器資訊清單名稱，然後再試一次。

具有SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG原因的視覺系統資料錯誤訊息會在回應中包含提示，提供請求失敗原因的相關資訊。您可以使用提示來判斷要遵循哪些疑難排解準則。

> **Note**
>
> 視覺系統資料處於預覽版本，可能會有所變更。

診斷解碼器資訊清單視覺系統資料驗證

錯誤	準則疑難排解
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)`	AWS IoT FleetWise 找不到使用訊號目錄的訊號解碼器中使用的根訊號結構。確認已在訊號目錄中正確定義結構的根訊號。
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	未在解碼器資訊清單更新請求中使用相同資料類型定義訊號目錄中的基本訊息。確認請求中定義的基本訊息符合其對應的訊號目錄定義。
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)`	訊號目錄中結構中定義的屬性數目，與您嘗試在解碼器資訊清單中解碼的屬性數目不相符。將訊號與訊號目錄中定義的訊號進行比較，確認您有正確數量的訊號要解碼。
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	AWS IoT FleetWise 在訊號目錄中找到定義為 STRUCT 的訊號，而沒有在解碼器資訊清單請求中定義的 structuredMessageDefinition。確定每個結構在解碼器資訊清單更新請求中定義為 structuredMessageDefinition。
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	解碼器資訊清單中使用的結構根訊號未正確定義為訊號目錄中的結構。解碼器資訊清單中使用的根訊號結構必須定義其欄位 structFullyQualifiedName。它還需要具有該 fullyQualifiedName 的 STRUCT 節點。
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	解碼器資訊清單請求中使用的其中一個分葉訊息未定義為基本訊息。確認請求中的所有分葉物件都定義為基本訊息。
`InvalidSignalDecoder.withReason(SignalDecoderFailureRe`	訊號目錄中的陣列物件並未在解碼器資訊清單更新請求中定義為 structuredMessageListDefini

錯誤	準則疑難排解
ason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)	tion。確認解碼器資訊清單更新請求中的所有陣列屬性都定義為 structuredMessageListDefinition。

Edge Agent for AWS IoT FleetWise 軟體問題

故障診斷 Edge Agent 軟體問題。

問題

- 問題：Edge Agent 軟體不會啟動。
- 問題：【ERROR】 【IoTFleetWiseEngine：：connect】 ： 【無法初始化持續性程式庫】
- 問題：Edge Agent 軟體不會收集內建診斷 (OBD) II PIDs和診斷故障碼 DTCs)。
- 問題：Edge Agent for AWS IoT FleetWise 軟體不會從網路收集資料，或無法套用資料檢查規則。
- 問題：【ERROR】 【AwsIotConnectivityModule：：connect】 ： 【連線失敗並發生錯誤】 或 【WARN】 【AwsIotChannel：：send】 ： 【無作用中 MQTT 連線。】

問題：Edge Agent 軟體不會啟動。

當 Edge Agent 軟體未啟動時，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 Edge Agent for AWS IoT FleetWise 軟體組態檔案使用有效的 JSON 格式。例如，確保已正確使用逗號。如需組態檔案的詳細資訊，請執行下列動作來下載 Edge Agent for AWS IoT FleetWise 軟體開發人員指南。

  1. 開啟 AWS IoT FleetWise 主控台。
  2. 在服務首頁的開始使用 AWS IoT FleetWise 區段中，選擇探索邊緣代理程式。
- ```
  [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 軟體無法與組態檔案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界面是否可用，請執行下列命令。

```
ip link show
```

若要讓網路介面上線，請執行下列命令。將 *network-interface-id* 取代為網路界面的 ID。

```
sudo ip link set network-interface-id up
```

-
```
[ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error]
[WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]
# or
[WARN] [AwsIotChannel::send]: [aws-c-common: AWS_ERROR_FILE_INVALID_PATH]
```

解決方案：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。檢查下列項目是否已正確設定，並重新啟動 Edge Agent 軟體。

- `mqttConnection::endpointUrl` – AWS 帳戶的 IoT 裝置端點。
- `mqttConnection::clientID` – Edge Agent 軟體執行所在的車輛 ID。
- `mqttConnection::certificateFilename` – 車輛憑證檔案的路徑。
- `mqttConnection::privateKeyFilename` – 車輛私有金鑰檔案的路徑。
- 您已使用 AWS IoT Core 來佈建 車輛。如需詳細資訊，請參閱Provision AWS IoT FleetWise 車輛。

如需更多故障診斷資訊，請參閱適用於 C++ 的 AWS IoT Device SDK 常見問答集。

問題：【ERROR】【IoTFleetWiseEngine：：connect】：【無法初始化持續性程式庫】

解決方案：當 Edge Agent 軟體找不到持久性儲存時，您可能會看到此錯誤。檢查下列項目是否已正確設定，並重新啟動 Edge Agent 軟體。

`persistency:persistencyPath` – 用來保留收集方案、解碼器資訊清單和資料快照的本機路徑。

問題：Edge Agent 軟體不會收集內建診斷 (OBD) II PIDs和診斷故障碼 DTCs)。

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 軟體是在自動傳輸車輛中執行，請確定 `obdInterface:hasTransmissionEcu` 已設定為 `true`。

如果您的車輛支援擴充控制器區域網路 (CAN 匯流排） 仲裁 IDs，請確定 `obdInterface:useExtendedIds` 已設定為 `true`。

問題：Edge Agent for AWS IoT FleetWise 軟體不會從網路收集資料，或無法套用資料檢查規則。

解決方案：當違反預設配額時，您可能會看到此錯誤。

資源	配額	可調整	注意
訊號 ID 的值	訊號 ID 必須小於或等於 50，000	是	Edge Agent 軟體不會從 ID 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的訊號數量。
每部車輛作用中的資料收集方案數量	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行銷活動數量，以及每個行銷活動包含的方案數量。
訊號歷史記錄緩衝區的大小	20 MB	是	如果超出配額，Edge Agent 軟體會停止收集新資料。

問題：【ERROR】 【AwsIotConnectivityModule：：connect】： 【連線失敗並發生錯誤】 或 【WARN】 【AwsIotChannel：：send】： 【無作用中 MQTT 連線。】

解決方案：當 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 AWS IoT Core 每分鐘會傳送 ping 請求至 ，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。

存放和轉送問題

> **Important**
>
> 目前已封鎖存取特定 AWS IoT FleetWise 功能。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

問題：接收`AccessDeniedException`具有所有必要 IAM 許可的

解決方案：行銷活動中資料分割的儲存和轉送功能的早期存取版本需要允許清單。請聯絡 服務團隊，以確保您的資源透過允許清單擁有足夠的許可。

問題：上傳至 AWS IoT Jobs 的資料會忽略 `endTime`

解決方案：您已`endtime`在任務文件中指定無效的 。例如， `endtime`不遵循 ISO 8601 UTC 格式）。在 AWS IoT FleetWise 客服人員日誌中，可能有警告層級陳述式指出 `Malformed IoT Job endTime: customer configured endTime. Not setting endTime`。

問題：上傳至 AWS IoT 任務的資料具有`REJECTED`執行狀態。

解決方案：您已`campaignArn`在任務文件中指定無效的 。例如，如果您為未在車輛上執行的行銷活動指定 ARN，則 AWS IoT FleetWise 代理程式日誌`CampaignArn value in the received job document does not match the ARN of a Store and Forward campaign`中可能會顯示錯誤層級陳述式。

安全 in AWS IoT FleetWise

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為符合最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。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 Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 AWS IoT FleetWise 的合規計劃，請參閱合規計劃的 AWS 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規

本文件可協助您了解如何在使用 AWS IoT FleetWise 時套用共同責任模型。它說明如何設定 AWS IoT FleetWise 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS IoT FleetWise 資源。

目錄

- AWS IoT FleetWise 中的資料保護
- 使用 控制存取 AWS IoT FleetWise
- Identity and Access Management for AWS IoT FleetWise
- AWS IoT FleetWise 的合規驗證
- AWS IoT FleetWise 中的彈性
- AWS IoT FleetWise 中的基礎設施安全
- AWS IoT FleetWise 中的組態和漏洞分析
- AWS IoT FleetWise 的安全最佳實務

AWS IoT FleetWise 中的資料保護

AWS 共同責任模型適用於 AWS IoT FleetWise 中的資料保護。如此模型所述， AWS 負責保護執行所有 的 全球基礎設施 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，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 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。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 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 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，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) 140-3。

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 AWS IoT FleetWise 或使用 AWS 服務 主控台、API AWS CLI或其他 AWS SDKs 時。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 旨在與您開發並在支援的車輛硬體上安裝的 Edge 代理程式搭配使用，以將車輛資料傳輸至 AWS 雲端。從車輛擷取資料可能受到某些司法管轄區的資料隱私權法規的約束。在使用 AWS IoT FleetWise 和安裝 Edge Agent 之前，強烈建議您根據適用的法律評估您的合規義務。這包括任何適用的法律要求，以提供法律上適當的隱私權通知，並取得擷取車輛資料所需的任何同意。

靜態加密 in AWS IoT FleetWise

從車輛收集的資料會透過 MQTT AWS IoT Core 訊息通訊協定的訊息傳輸到雲端。 AWS IoT FleetWise 會將資料交付到您的 Amazon Timestream 資料庫。在 Timestream 中，您的資料會加密。根據預設，所有靜態 AWS 服務 加密資料。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的使用加密保護資料和 Timestream for LiveAnalytics 中的資料保護。

靜態加密與 AWS Key Management Service (AWS KMS) 整合，以管理用來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 FleetWise 收集的資料。您可以透過 建立、管理和檢視加密金鑰 AWS KMS。如需詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的什麼是 AWS Key Management Service？。

傳輸中加密

AWS IoT 與服務交換的所有資料都會使用 Transport Layer Security (TLS) 在傳輸中加密。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， AWS IoT Core 支援身分驗證和授權，以協助安全地控制對 AWS IoT FleetWise 資源的存取。車輛可以使用 X.509 憑證進行身分驗證 （ 登入 ），以使用 AWS IoT FleetWise 並使用 AWS IoT Core 政策來取得授權 （ 具有許可 ） 來執行指定的動作。如需詳細資訊，請參閱the section called “佈建 車輛”。

AWS IoT FleetWise 中的資料加密

資料加密是指在傳輸中 （ 往返 AWS IoT FleetWise 以及閘道和伺服器之間 ） 和靜態 （ 存放在本機裝置或 中 ） 時保護資料 AWS 服務。您可以使用用戶端加密來保護靜態資料。

> **Note**
>
> AWS IoT FleetWise 邊緣處理會公開在 AWS IoT FleetWise 閘道內託管APIs，並且可透過本機網路存取。這些 APIs 透過由 AWS IoT FleetWise Edge 連接器擁有的伺服器憑證支援的 TLS 連線公開。對於用戶端身分驗證，這些 APIs 使用存取控制密碼。伺服器憑證私有金鑰和存取控制密碼都存放在磁碟上。 AWS IoT FleetWise 節點處理依賴檔案系統加密，以確保這些靜態憑證的安全性。

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 in AWS IoT FleetWise
- AWS IoT FleetWise 中的金鑰管理

靜態加密 in AWS IoT FleetWise

AWS IoT FleetWise 會將您的資料存放在 AWS 雲端和閘道。

AWS 雲端中的靜態資料

AWS IoT FleetWise 會將資料存放在預設 AWS 服務 加密靜態資料的其他 中。靜態加密與 AWS Key Management Service (AWS KMS) 整合，用於管理加密金鑰，用於加密 AWS IoT FleetWise 中的資產

屬性值和彙總值。您可以選擇使用客戶受管金鑰來加密資產屬性值，並在 AWS IoT FleetWise 中彙總值。您可以透過 建立、管理和檢視加密金鑰 AWS KMS。

您可以選擇 AWS 擁有的金鑰 或客戶受管金鑰來加密您的資料。

運作方式

靜態加密與 整合 AWS KMS ，用於管理用來加密資料的加密金鑰。

- AWS 擁有的金鑰 – 預設加密金鑰。 AWS IoT FleetWise 擁有此金鑰。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 日誌中查看金鑰的操作。您可以使用此金鑰，無需額外費用。
- 客戶受管金鑰 – 金鑰存放在您的帳戶中，由您建立、擁有和管理。您可以完全控制 KMS 金鑰。 AWS KMS 需支付額外費用。

AWS 擁有的金鑰

AWS 擁有的金鑰 不會存放在您的帳戶中。它們是 KMS 金鑰集合的一部分，這些金鑰 AWS 擁有和管理以在 multiple 中使用 AWS 帳戶。 AWS 服務 可以使用 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方式。不過，您不需要採取任何動作或變更任何程式來保護加密資料的金鑰。

如果您使用 ，則不會向您收取費用 AWS 擁有的金鑰，而且這些費用不會計入您帳戶的 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，如下所示：

- 建立和維護其金鑰政策、IAM 政策和授權
- 啟用和停用它們
- 輪換其密碼編譯資料
- 新增標籤
- 建立參考它們的別名
- 排程刪除它們

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

如果您使用客戶受管金鑰，則必須授予 AWS IoT FleetWise 存取存放在您帳戶中的 KMS 金鑰的權限。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根金鑰。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。

下列範例政策授予 AWS IoT FleetWise 許可，以使用您的 AWS KMS 金鑰。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

Important

當您將新區段新增至 KMS 金鑰政策時，請勿變更政策中的任何現有區段。如果啟用了 for AWS IoT FleetWise 加密，且符合下列任一條件， AWS IoT FleetWise 就無法執行資料的操作：

- KMS 金鑰已停用或刪除。
- 服務未正確設定 KMS 金鑰政策。

將視覺系統資料與靜態加密搭配使用

> **Note**
>
> 視覺系統資料處於預覽版本，可能會有所變更。

如果您的 AWS IoT FleetWise 帳戶已啟用 AWS KMS 金鑰的客戶受管加密，而且您想要使用視覺系統資料，請將加密設定重設為與複雜的資料類型相容。這可讓 AWS IoT FleetWise 建立視覺系統資料所需的額外許可。

> **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 操作來檢查是否已啟用 AWS KMS 加密。如果加密類型為 ，則不需要進一步的動作`FLEETWISE_DEFAULT_ENCRYPTION`。
2. 如果加密類型為 `KMS_BASED_ENCRYPTION`，請使用 PutEncryptionConfiguration API 操作將加密類型重設為 `FLEETWISE_DEFAULT_ENCRYPTION`。

```
aws iotfleetwise put-encryption-configuration \
      --encryption-type FLEETWISE_DEFAULT_ENCRYPTION
```

3. 使用 PutEncryptionConfiguration API 操作將加密類型重新啟用至 `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 AWS IoT FleetWise 中的金鑰管理。

AWS IoT FleetWise 中的金鑰管理

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

AWS IoT FleetWise 雲端金鑰管理

根據預設， AWS IoT FleetWise 會使用 AWS 受管金鑰 來保護 中的資料 AWS 雲端。您可以更新您的設定，以使用客戶受管金鑰來加密 AWS IoT FleetWise 中的資料。您可以透過 AWS Key Management Service () 建立、管理和檢視加密金鑰AWS KMS。

AWS IoT FleetWise 支援使用存放在 中的客戶受管金鑰進行伺服器端加密， AWS KMS 以加密下列資源的資料。

AWS IoT FleetWise 資源	資料類型	使用客戶受管金鑰進行靜態加密的欄位
訊號目錄		description
	屬性	description， allowedValues， defaultValue， min， max
	傳動器	description， allowedValues， min， max
	感測器	description， allowedValues， min， max
車輛模型（模型資訊清單）		description
解碼器資訊清單		description
	CanInterface	protocolName、protocolVersion
	ObdInterface	requestMessageId、dtcRequestIntervalSeconds、 hasTransmissionEcu、obdStandard、pidRequestIntervalSeconds、useExtendedIds
	CanSignal	factor、isBigEndian、isSigned、length、messageId、 offset、startBit

AWS IoT FleetWise 資源	資料類型	使用客戶受管金鑰進行靜態加密的欄位
	ObdSignal	byteLength， offset， pid， pidResponseLength， scaling， serviceMode， startByte， bitMaskLength， bitRightShift
車輛		屬性
Campaign		description
	conditionBasedCollectionScheme	expression， conditionLanguageVersion， minimumTriggerIntervalMs， triggerMode
	TimeBasedCollectionScheme	periodMs
狀態範本		description

> **Note**
> 其他資料和資源會使用預設加密搭配由 AWS IoT FleetWise 管理的金鑰進行加密。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《 AWS Key Management Service 開發人員指南》中的什麼是 AWS Key Management Service？。

使用 KMS 金鑰啟用加密（主控台）

若要搭配 AWS IoT FleetWise 使用客戶受管金鑰，您必須更新您的 AWS IoT FleetWise 設定。

使用 KMS 金鑰啟用加密（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，選擇選擇不同的 AWS KMS 金鑰。這可使用存放於 的客戶受管金鑰進行加密 AWS KMS。

> **Note**
>
> 您只能使用 AWS IoT FleetWise 資源的客戶受管金鑰加密。這包括訊號目錄、車輛模型（模型資訊清單）、解碼器資訊清單、車輛、機群和行銷活動。

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 – 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 AWS KMS 金鑰。

   > **Note**
   >
   > 這會開啟 AWS KMS 主控台。如需建立 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。

使用 KMS 金鑰啟用加密 (AWS CLI)

您可以使用 PutEncryptionConfiguration API 操作來啟用 AWS IoT FleetWise 帳戶的加密。下列範例使用 AWS CLI。

若要啟用加密，請執行下列命令。

- 以 KMS 金鑰的 ID 取代 *kms_key_id*。

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

Example response

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

```
}
```

KMS 金鑰政策

建立 KMS 金鑰後，您至少必須將下列陳述式新增至您的 KMS 金鑰政策，以便其使用 AWS IoT FleetWise。KMS 金鑰政策陳述`iotfleetwise.amazonaws.com`式中的 AWS IoT FleetWise 服務主體允許 AWS IoT FleetWise 存取 KMS 金鑰。

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

做為安全最佳實務，請將 `aws:SourceArn`和 `aws:SourceAccount`條件金鑰新增至 KMS 金鑰政策。IAM 全域條件金鑰`aws:SourceArn`有助於確保 AWS IoT FleetWise 僅針對服務特定的資源 Amazon Resource Name (ARNs) 使用 KMS 金鑰。

如果您設定 的值`aws:SourceArn`，則一律必須為 `arn:aws:iotfleetwise:us-east-1:account_id:*`。這可讓 KMS 金鑰為此存取所有 AWS IoT FleetWise 資源 AWS 帳戶。AWS IoT FleetWise 支援每個帳戶中所有資源的一個 KMS 金鑰 AWS 區域。針對 使用任何其他值`SourceArn`，或未針對 ARN 資源欄位使用萬用字元 (*)，可防止 AWS IoT FleetWise 存取 KMS 金鑰。

的值`aws:SourceAccount`是您的帳戶 ID，用於進一步限制 KMS 金鑰，使其只能用於您的特定帳戶。如果您將 `aws:SourceAccount`和 `aws:SourceArn`條件金鑰新增至 KMS 金鑰，請確定任何其他服務或帳戶不會使用該金鑰。這有助於避免失敗。

下列政策包含服務委託人（服務的識別符），以及根據 `aws:SourceAccount` 和您的帳戶 ID `aws:SourceArn`設定 AWS 區域 和 以供使用。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:SourceAccount": "AWS-account-ID"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iotfleetwise:region:AWS-account-ID:*"
    }
  }
}
```

如需有關編輯 KMS 金鑰政策以搭配 AWS IoT FleetWise 使用的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的變更金鑰政策。

> **Important**
>
> 當您將新區段新增至 KMS 金鑰政策時，請勿變更政策中的任何現有區段。如果啟用了 for AWS IoT FleetWise 加密，且符合下列任何條件， AWS IoT FleetWise 就無法執行資料的操作：
>
> - KMS 金鑰已停用或刪除。
> - 服務未正確設定 KMS 金鑰政策。

AWS KMS 加密許可

如果您啟用 AWS KMS 加密，則必須在角色政策中指定許可，以便您可以呼叫 AWS IoT FleetWise APIs。下列政策允許存取所有 AWS IoT FleetWise 動作，以及 AWS KMS 特定許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:*",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

您的角色需要下列政策陳述式，才能叫用加密 APIs。此政策陳述式允許來自 AWS IoT FleetWise 的 `PutEncryptionConfiguration`和 `GetEncryptionConfiguration`動作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:GetEncryptionConfiguration",
        "iotfleetwise:PutEncryptionConfiguration",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
```

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

刪除 AWS KMS 金鑰後的復原

如果您在使用 AWS IoT FleetWise 啟用加密後刪除 AWS KMS 金鑰，您必須先刪除所有資料，再再次使用 AWS IoT FleetWise 來重設您的帳戶。您可以使用清單和刪除 API 操作來清除帳戶中的資源。

清除帳戶中的資源

1. 使用將 `listResponseScope` 參數設為 APIs`METADATA_ONLY`。這提供資源清單，包括資源名稱和其他中繼資料，例如 ARNs 和時間戳記。
2. 使用刪除 APIs移除個別資源。

您必須依照下列順序清除資源。

1. 行銷活動
   a. 列出 `listResponseScope` 參數設定為 的所有行銷活動`METADATA_ONLY`。
   b. 刪除行銷活動。
2. 機群和車輛
   a. 列出 `listResponseScope` 參數設定為 的所有機群`METADATA_ONLY`。
   b. 列出每個機群的所有車輛，並將 `listResponseScope` 參數設為 `METADATA_ONLY`。
   c. 取消所有車輛與每個機群的關聯。
   d. 刪除機群。
   e. 刪除車輛。
3. 解碼器資訊清單
   a. 列出 `listResponseScope` 參數設定為 的所有解碼器資訊清單`METADATA_ONLY`。
   b. 刪除所有解碼器資訊清單。
4. 車輛模型（模型資訊清單）
   a. 列出 `listResponseScope` 參數設定為 的所有車輛模型`METADATA_ONLY`。
   b. 刪除所有車輛模型。

5. 狀態範本
   a. 列出 `listResponseScope` 參數設定為 的所有狀態範本`METADATA_ONLY`。
   b. 刪除所有狀態範本。
6. 訊號目錄
   a. 列出所有訊號目錄。
   b. 刪除所有訊號目錄。

使用 控制存取 AWS IoT FleetWise

> **Important**
>
> 對特定 AWS IoT FleetWise 功能的存取目前受到鎖定。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

下列各節說明如何控制對 AWS IoT FleetWise 資源的存取。他們涵蓋的資訊包括如何授予應用程式存取權，以便 AWS IoT FleetWise 可以在行銷活動期間傳輸車輛資料。它們也說明如何授予 Amazon S3 (S3) 儲存貯體或 Amazon Timestream 資料庫和資料表的 AWS IoT FleetWise 存取權，以存放資料，或授予用於從車輛傳送資料的 MQTT 訊息。

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

- 授予 AWS IoT FleetWise 許可，以傳送和接收 MQTT 主題的資料
- 授予對 Amazon S3 目的地的 AWS IoT FleetWise 存取權
- 授予對 Amazon Timestream 目的地的 AWS IoT FleetWise 存取權
- 授予使用 產生命令承載的 AWS IoT Device Management 許可 AWS IoT FleetWise

授予 AWS IoT FleetWise 許可，以傳送和接收 MQTT 主題的資料

當您使用 MQTT 主題時，您的車輛會使用 AWS IoT MQTT 訊息代理程式傳送資料。您必須授予 AWS IoT FleetWise 許可，才能訂閱您指定的 MQTT 主題。如果您也使用 AWS IoT 規則來採取行動，或將

資料路由到其他目的地，則必須將政策連接到 IAM 角色，以允許 AWS IoT FleetWise 將資料轉送到 IoT 規則。

此外，您的其他應用程式或裝置可以訂閱您指定的主題，以近乎即時的方式接收車輛資料，而且這些應用程式或裝置必須視需要獲得許可和存取權。

如需使用 MQTT 以及所需角色和許可的詳細資訊，請參閱：

- 裝置通訊協定
- 的規則 AWS IoT
- 授予 AWS IoT 規則所需的存取權
- 傳遞角色許可

開始之前，請檢查下列項目：

> **Important**
>
> - 建立 AWS IoT FleetWise 車輛行銷活動資源時，您必須使用相同的 AWS 區域。如果您切換 AWS 區域，您可能會遇到存取資源的問題。
> - AWS IoT FleetWise 在美國東部（維吉尼亞北部）和歐洲（法蘭克福）提供。

您可以使用 AWS CLI 建立具有 MQTT 訊息信任政策的 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

- 將 *IotTopicExecutionRole* 取代為您建立的角色名稱。
- 將 *trust-policy* 取代為包含信任政策的 JSON 檔案。

```
aws iam create-role --role-name IotTopicExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

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

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

建立許可政策以授予 AWS IoT FleetWise 發佈訊息至您指定之 MQTT 主題的許可。若要建立許可政策，請執行下列命令。

建立許可政策

- 將 *AWSIoTFleetwiseAccessIotTopicPermissionsPolicy* 取代為您建立的政策名稱。
- 將 *permissions-policy* 取代為包含許可政策的 JSON 檔案名稱。

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessIotTopicPermissionsPolicy --
policy-document file://permissions-policy.json
```

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

```
        "Resource": [
          "topic-arn"
        ]
      }
    ]
}
```

將許可政策連接至您的 IAM 角色

1. 從輸出中，複製許可政策的 Amazon Resource Name (ARN)。
2. 若要將 IAM 許可政策連接至您的 IAM 角色，請執行下列命令。
   - 將 *permissions-policy-arn* 取代為您在上一個步驟中複製的 ARN。
   - 將 *IotTopicExecutionRole* 取代為您建立的 IAM 角色名稱。

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

如需詳細資訊，請參閱《IAM 使用者指南》中的存取 AWS 資源的管理。

授予對 Amazon S3 目的地的 AWS IoT FleetWise 存取權

當您使用 Amazon S3 目的地時， 會將車輛資料 AWS IoT FleetWise 交付至 S3 儲存貯體，並可選擇使用您擁有的 AWS KMS 金鑰進行資料加密。如果啟用錯誤記錄， AWS IoT FleetWise 也會將資料交付錯誤傳送至 CloudWatch 日誌群組和串流。建立交付串流時，您需要有 IAM 角色。

AWS IoT FleetWise 使用儲存貯體政策搭配 S3 目的地的服務主體。如需新增儲存貯體政策的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon S3 主控台新增儲存貯體政策。

使用以下存取政策， AWS IoT FleetWise 讓 存取您的 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。這會授予儲存貯體擁有者對 交付物件的完整存取權 AWS IoT FleetWise。如需如何安全存取儲存貯體中物件的詳細資訊，請參閱《Amazon Simple Storage Service 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 AWS 區域中帳戶中的所有行銷活動。

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

如果您將 KMS 金鑰連接到 S3 儲存貯體，則金鑰將需要下列政策。如需金鑰管理的資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用伺服器端加密與 AWS Key Management Service 金鑰 (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**
>
> 當您建立儲存貯體時，S3 會建立預設存取控制清單 (ACL)，授予資源擁有者對資源的完整控制權。如果 AWS IoT FleetWise 無法將資料交付至 S3，請務必停用 S3 儲存貯體上的 ACL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的停用所有新儲存貯體ACLs 並強制執行物件擁有權。

授予對 Amazon Timestream 目的地的 AWS IoT FleetWise 存取權

當您使用 Timestream 目的地時， 會將車輛資料 AWS IoT FleetWise 傳送到 Timestream 資料表。您必須將政策連接至 IAM 角色，以允許 AWS IoT FleetWise 將資料傳送至 Timestream。

如果您使用主控台建立行銷活動， AWS IoT FleetWise 會自動將所需的政策連接到角色。

> **Note**
>
> Amazon Timestream 不適用於亞太區域（孟買）區域。

開始之前，請檢查下列項目：

> **Important**
>
> - 當您為 AWS IoT FleetWise 建立 Timestream 資源時，必須使用相同的 AWS 區域。如果您切換 AWS 區域，存取 Timestream 資源時可能會遇到問題。
> - AWS IoT FleetWise 在美國東部（維吉尼亞北部）、歐洲（法蘭克福）和亞太區域（孟買）提供。
> - 如需支援的 區域清單，請參閱《》中的 Timestream 端點和配額AWS 一般參考。

- 您必須擁有 Timestream 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 Timestream 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》中的建立資料表。

您可以使用 AWS CLI 建立具有 Timestream 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

- 將 *TimestreamExecutionRole* 取代為您建立的角色名稱。
- 將 *trust-policy* 取代為包含信任政策的 .json 檔案。

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

建立許可政策以授予 AWS IoT FleetWise 將資料寫入 Timestream 的許可。若要建立許可政策，請執行下列命令。

建立許可政策

- 將 *AWSIoTFleetwiseAccessTimestreamPermissionsPolicy* 取代為您建立的政策名稱。
- 將 *permissions-policy* 取代為包含許可政策的 JSON 檔案名稱。

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessTimestreamPermissionsPolicy --
policy-document file://permissions-policy.json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "timestreamIngestion",
      "Effect": "Allow",
      "Action": [
        "timestream:WriteRecords",
        "timestream:Select",
        "timestream:DescribeTable"
      ],
      "Resource": "table-arn"
    },
    {
      "Sid": "timestreamDescribeEndpoint",
      "Effect": "Allow",
      "Action": [
        "timestream:DescribeEndpoints"
      ],
      "Resource": "*"
    }
  ]
}
```

將許可政策連接至您的 IAM 角色

1. 從輸出中，複製許可政策的 Amazon Resource Name (ARN)。
2. 若要將 IAM 許可政策連接至您的 IAM 角色，請執行下列命令。

- 將 *permissions-policy-arn* 取代為您在上一個步驟中複製的 ARN。
- 以您建立的 IAM 角色名稱取代 *TimestreamExecutionRole*。

```
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》中的存取 AWS 資源管理。

授予使用 產生命令承載的 AWS IoT Device Management 許可 AWS IoT FleetWise

當您使用命令功能啟動命令執行時， AWS IoT Device Management 會從傳入的請求中擷取命令和命令參數。然後，它需要存取 AWS IoT FleetWise 資源的許可，以驗證請求並產生承載。接著會透過 AWS IoT Device Management MQTT 將承載傳送至 車輛已訂閱的命令請求主題。

您必須先建立授予產生承載 AWS IoT Device Management 所需許可的 IAM 角色。然後，使用 `roleArn` 欄位將此角色的 ARN 提供給 `CreateCommand` API。以下顯示一些政策範例。

> **Important**
>
> 對於 IAM 角色，您必須使用 AWS 區域 與建立車輛和命令資源相同的角色。如果您切換 AWS 區域，存取資源時可能會遇到問題。

IAM 角色需要具有下列信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RemoteCommandsTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

授予許可給所有車輛 (IoT 物件)

下列範例顯示如何授予許可，以產生所有註冊為 AWS IoT 實物之車輛的承載。

> **Note**
>
> - 此政策可能過於寬鬆。使用最低權限原則，以確保您只授予必要的許可。
> - 若要改為拒絕許可，請在 IAM 政策`"Effect": "Deny"`中`"Effect": "Allow"`將 變更為 。

在此範例中，取代：

- *<AWS_REGION>* 搭配 AWS 區域 您使用 AWS IoT FleetWise 資源的 。
- *<ACCOUNT_ID>* AWS 帳戶 與您的 號碼。

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

將許可授予特定車輛 (IoT 物件)

下列範例顯示如何授予許可，為註冊為 AWS IoT 物件的特定車輛產生承載。

在此範例中，取代：

- *<AWS_REGION>* 搭配 AWS 區域 您使用 AWS IoT FleetWise 資源的 。
- *<ACCOUNT_ID>* AWS 帳戶 與您的 號碼。

- *<VEHICLE_NAME>* 搭配您車輛的 IoT 物件名稱 。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>"
        }
    ]
}
```

將許可授予特定車輛和訊號

下列範例顯示如何授予許可，以產生特定車輛的傳動器承載。

在此範例中，取代：

- *<AWS_REGION>* 搭配 AWS 區域 您使用 AWS IoT FleetWise 資源的 。
- *<ACCOUNT_ID>* AWS 帳戶 與您的 號碼。
- *<VEHICLE_NAME>* 搭配您車輛的 IoT 物件名稱。
- 訊號名稱為 *<SIGNAL_FQN>*，例如 *<Vehicle.actuator2>*。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iotfleetwise:Signals": ["<SIGNAL_FQN>"]
                }
            }
        }
    ]
```

```
}
```

將許可授予特定車輛和狀態範本

下列範例顯示如何授予許可，以產生特定車輛和狀態範本的承載。

在此範例中，取代：

- *<AWS_REGION>* 是您 AWS 區域 使用 AWS IoT FleetWise 資源的 。
- *<ACCOUNT_ID>* 是您的 AWS 帳戶 號碼。
- *<VEHICLE_NAME>* 是您車輛的 IoT 物件名稱。
- *<STATE_TEMPLATE_ID>*，其中包含狀態範本的識別符。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": [
                "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
                "arn:aws:iotfleetwise:<AWS_REGION>:<ACCOUNT_ID>:state-
template/<STATE_TEMPLATE_ID>"]
        }
    ]
}
```

授予使用客戶受管 KMS 金鑰的許可

如果您已為 啟用客戶受管 KMS 金鑰 AWS IoT FleetWise，則下列範例顯示如何授予產生承載的許可。

在此範例中，取代：

- *<AWS_REGION>* 搭配 AWS 區域 您使用 AWS IoT FleetWise 資源的 。
- *<ACCOUNT_ID>* 搭配您的 AWS 帳戶 號碼。
- *<KMS_KEY_ID>* 搭配 KMS 金鑰的 ID。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:<AWS_REGION>:<ACCOUNT_ID>:key/<KMS_KEY_ID>"
        }
    ]
}
```

Identity and Access Management for AWS IoT FleetWise

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （登入） 和授權 （具有許可），以使用 AWS IoT FleetWise 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- AWS IoT FleetWise 如何與 IAM 搭配使用
- AWS IoT FleetWise 的身分型政策範例
- 故障診斷 AWS IoT FleetWise 身分和存取
- AWS IoT FleetWise API 動作和資源參考

目標對象

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，取決於您在 AWS IoT FleetWise 中執行的工作。

服務使用者 – 如果您使用 AWS IoT FleetWise 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS IoT FleetWise 功能來執行工作時，您可能需要額外的許可。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 FleetWise 中的功能，請參閱 故障診斷 AWS IoT FleetWise 身分和存取。

服務管理員 – 如果您在公司負責 AWS IoT FleetWise 資源，您可能擁有 AWS IoT FleetWise 的完整存取權。您的任務是判斷服務使用者應存取的 AWS IoT FleetWise 功能和資源。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使用 IAM 搭配 AWS IoT FleetWise，請參閱 AWS IoT FleetWise 如何與 IAM 搭配使用。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解如何撰寫政策以管理 AWS IoT FleetWise 存取的詳細資訊。若要檢視您可以在 IAM 中使用的範例 AWS IoT FleetWise 身分型政策，請參閱 AWS IoT FleetWise 的身分型政策範例。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您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。當您使用聯合 AWS 身分存取 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。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的登入資料以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。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》中的適用於 API 請求的AWS Signature 第 4 版。

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 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。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 IAM 中的AWS 多重要素驗證。

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，該身分可完整存取帳戶中的所有 AWS 服務 和 資源。此身分稱為 AWS 帳戶 Theroot 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

最佳實務是要求人類使用者，包括需要管理員存取權的使用者，使用身分提供者的聯合身分來 AWS 服務 使用臨時憑證來存取 。

聯合身分是來自您的企業使用者目錄、Web 身分提供者、 AWS Directory Service、 Identity Center 目錄或任何使用透過身分來源提供的登入資料 AWS 服務 存取的使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

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，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，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 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，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，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色 （主控台）。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需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 (聯合) 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 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以直接將政策連接到資源（而不是使用角色做為代理）。如需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 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。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。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。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 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 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。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
  - 服務連結角色 – 服務連結角色是一種連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。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 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。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 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。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臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源相關聯時， 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，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。具有該政策的使用者可以從 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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

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。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，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。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 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來自 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。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 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政策類型。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。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 之多個 的服務。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 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括支援 RCPs AWS 服務 的 清單，請參閱AWS Organizations 《 使用者指南》中的資源控制政策 (RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。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。如需詳細資訊，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。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS IoT FleetWise 如何與 IAM 搭配使用

在您使用 IAM 管理對 AWS IoT FleetWise 的存取之前，請先了解哪些 IAM 功能可與 AWS IoT FleetWise 搭配使用。

您可以搭配 AWS IoT FleetWise 使用的 IAM 功能

IAM 功能	AWS IoT FleetWise 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC(政策中的標籤)	部分
臨時憑證	是
主體許可	是
服務角色	否
服務連結角色	否

若要全面了解 AWS IoT FleetWise 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《IAM 使用者指南》中的AWS 與 IAM 搭配使用的 服務。

AWS IoT FleetWise 的身分型政策

支援身分型政策：是

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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

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。

AWS IoT FleetWise 的身分型政策範例

若要檢視 AWS IoT FleetWise 身分型政策的範例，請參閱 AWS IoT FleetWise 的身分型政策範例。

AWS IoT FleetWise 內的資源型政策

支援資源型政策：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。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做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位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予主體實體（使用者或角色）存取資源的許可。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 中的快帳戶資源存取。

AWS IoT FleetWise 的政策動作

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內容。也就是說，哪個主體在什麼条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。政策動作通常具有與相關聯 AWS API 操作相同的名稱。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 動作的清單，請參閱《服務授權參考》中的AWS IoT FleetWise 定義的動作。

AWS IoT FleetWise 中的政策動作在動作之前使用以下字首：

```
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 身分型政策的範例，請參閱 AWS IoT FleetWise 的身分型政策範例。

AWS IoT FleetWise 的政策資源

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 ARNs，請參閱《服務授權參考》中的AWS IoT FleetWise 定義的資源。若要了解您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 身分型政策的範例，請參閱 AWS IoT FleetWise 的身分型政策範例。

AWS IoT FleetWise 的政策條件索引鍵

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， 會使用邏輯`OR`操作 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的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS IoT FleetWise 條件金鑰的清單，請參閱《服務授權參考》中的AWS IoT FleetWise 的條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱AWS IoT FleetWise 定義的動作。

若要檢視 AWS IoT FleetWise 身分型政策的範例，請參閱 AWS IoT FleetWise 的身分型政策範例。

AWS IoT FleetWise 中的存取控制清單 (ACLs)

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。

使用 AWS IoT FleetWise 的屬性型存取控制 (ABAC)

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。您可以將標籤連接到 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。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，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》中的使用 ABAC 授權定義許可。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 僅支援 `CreateCampaign` API 操作`iam:PassRole`所需的 。

搭配 AWS IoT FleetWise 使用暫時登入資料

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法運作。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的使用 IAM 的 。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。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》中的從使用者切換至 IAM 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，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS IoT FleetWise 的跨服務主體許可

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，您會被視為委託人。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 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的

請求時，才會提出 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 FleetWise 的服務角色

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。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

> **Warning**
> 變更服務角色的許可可能會中斷 AWS IoT FleetWise 功能。只有在 AWS IoT FleetWise 提供指引時，才能編輯服務角色。

AWS IoT FleetWise 的服務連結角色

支援服務連結角色：否

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。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。

使用 AWS IoT FleetWise 的服務連結角色

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 FleetWise 預先定義，並包含 AWS IoT FleetWise 將指標傳送至 Amazon CloudWatch 所需的許可。如需詳細資訊，請參閱使用 Amazon CloudWatch 監控 AWS IoT FleetWise Amazon CloudWatch。

服務連結角色可讓您更快地設定 AWS IoT FleetWise，因為您不必手動新增必要的許可。AWS IoT FleetWise 會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 才能擔任其角色。已定義的許可包括信任政策和許可政策。此許可政策無法連接到任何其他 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。這可保護您的 AWS IoT FleetWise 資源，因為您不會不小心移除存取資源的許可。

如需有關支援服務連結角色的其他 服務的資訊，請參閱AWS 使用 IAM 的服務，並在服務連結角色欄中尋找具有是的服務。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 FleetWise 的服務連結角色許可

AWS IoT FleetWise 使用名為 AWSServiceRoleForIoTFleetWise 的服務連結角色 – 一種 AWS 受管政策，用於 AWS IoT FleetWise 的所有out-of-the-box許可。

AWSServiceRoleForIoTFleetWise 服務連結角色信任下列服務擔任該角色：

- `IoTFleetWise`

名為 AWSIoTFleetwiseServiceRolePolicy 的角色許可政策允許 AWS IoT FleetWise 對指定的資源完成下列動作：

- 動作：在資源`cloudwatch:PutMetricData`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 FleetWise 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。如需詳細資訊，請參閱設定您的 AWS IoT FleetWise 設定。

在 AWS IoT FleetWise 中建立服務連結角色（主控台）

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、CLI 或 AWS API AWS 中註冊帳戶時，AWS IoT FleetWise 會為您建立服務連結角色。

編輯 AWS IoT FleetWise 的服務連結角色

您無法在 AWS IoT FleetWise 中編輯 AWSServiceRoleForIoTFleetWise FleetWise 服務連結角色。AWS IoT 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」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。若要了解如何透過主控台、 AWS CLI 或 AWS API service-linked-role，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，然後需要再次建立該角色，則可以向 AWS IoT FleetWise 註冊 帳戶。然後AWS IoT FleetWise 會再次為您建立服務連結角色。

AWS IoT FleetWise 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AWS IoT FleetWise 資源的許可。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。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。

如需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 (主控台)。

如需 AWS IoT FleetWise 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的適用於 AWS IoT FleetWise 的動作、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
- 存取 Amazon Timestream 中的資源

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 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 受管政策並轉向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。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。

- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。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。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，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，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 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許可。這些許可必須允許您列出和檢視 中 AWS IoT FleetWise 資源的詳細資訊 AWS 帳戶。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。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。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。

為了確保使用者和角色仍然可以使用 AWS IoT FleetWise 主控台，也請將 AWS IoT FleetWise `ConsoleAccess`或 `ReadOnly` AWS 受管政策連接到實體。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。此政策包含在 主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

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

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

存取 Amazon Timestream 中的資源

使用 AWS IoT FleetWise 之前，您必須註冊 AWS 您的帳戶、IAM 和 Amazon Timestream 資源，以授予 AWS IoT FleetWise AWS 雲端 代表您將車輛資料傳送至 的許可。若要註冊，您需要：

- Amazon Timestream 資料庫。
- 在指定的 Amazon Timestream 資料庫中建立的資料表。
- 允許 AWS IoT FleetWise 將資料傳送至 Amazon Timestream 的 IAM 角色。

如需詳細資訊，包括程序和範例政策，請參閱 設定您的 AWS IoT FleetWise 設定。

故障診斷 AWS IoT FleetWise 身分和存取

使用以下資訊來協助您診斷和修正使用 AWS IoT FleetWise 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 AWS IoT FleetWise 中執行動作
- 我未獲得執行 iam:PassRole 的授權
- 我想要允許 以外的人員 AWS 帳戶 存取 my AWS IoT FleetWise 資源

我無權在 AWS IoT FleetWise 中執行動作

如果 AWS Management Console 告訴您無權執行 動作，則必須聯絡您的管理員尋求協助。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檢視虛構*`myVehicle`*資源的詳細資訊，但沒有 `iotfleetwise:`*`GetVehicleStatus`*許可時，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。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 `iam:PassRole`動作，您的政策必須更新，以允許您將角色傳遞至 AWS IoT FleetWise。

有些 AWS 服務 可讓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 `marymajor` 使用者嘗試使用主控台在 AWS IoT FleetWise 中執行動作時，會發生下列範例錯誤。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 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取 my 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：

- 若要了解 AWS IoT FleetWise 是否支援這些功能，請參閱 AWS IoT FleetWise 如何與 IAM 搭配使用。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需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)。
- 如需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 中的跨帳戶資源存取。

AWS IoT FleetWise API 動作和資源參考

當您使用 IAM 身分的政策管理存取權時，您可以使用下表做為參考。下表列出每個 AWS IoT FleetWise API、您可以授予執行動作許可的對應動作，以及您可以授予許可 AWS 的資源。

在政策的 `Action` 欄位中指定動作，然後在政策的 `Resource` 欄位中指定資源值。若要指定動作，請使用 `iotfleetwise:`字首，後面接著動作名稱。例如 `iotfleetwise:CreateSignalCatalog`。

目前， AWS IoT FleetWise 支援下列動作、資源和條件索引鍵。

使用捲軸查看資料表的其餘部分。

AWS IoT FleetWise 動作	所需的許可	資源
AssociateVehicleFleet	iotfleetwise：AssociateVehicleFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*

AWS IoT FleetWise 動作	所需的許可	資源
		arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id*
CreateCampaign	iotfleetwise：CreateCampaign	arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name* arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*
CreateDecoderManifest	iotfleetwise：CreateDecoderManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name*
CreateFleet	iotfleetwise：CreateFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id* arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*

AWS IoT FleetWise 動作	所需的許可	資源
CreateModelManifest	iotfleetwise：CreateModelManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
CreateSignalCatalog	iotfleetwise：CreateSignalCatalog	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
CreateStateTemplate	iotfleetwise：CreateStateTemplate	arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id* arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
CreateVehicle	iotfleetwise：CreateVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*

AWS IoT FleetWise 動作	所需的許可	資源
BatchCreateVehicle	iotfleetwise：CreateVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
DeleteCampaign	iotfleetwise：DeleteCampaign	arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name*
DeleteDecoderManifest	iotfleetwise：DeleteDecoderManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
DeleteFleet	iotfleetwise：DeleteFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*
DeleteModelManifest	iotfleetwise：DeleteModelManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
DeleteSignalCatalog	iotfleetwise：DeleteSignalCatalog	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
DeleteStateTemplate	iotfleetwise：DeleteStateTemplate	arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*

AWS IoT FleetWise 動作	所需的許可	資源
DeleteVehicle	iotfleetwise：DeleteVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id*
DisassociateVehicleFleet	iotfleetwise：DisassociateVehicleFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*
GetCampaign	iotfleetwise：GetCampaign	arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name*
GetDecoderManifest	iotfleetwise：GetDecoderManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
GetEncryptionConfiguration	iotfleetwise：GetEncryptionConfiguration	
GetFleet	iotfleetwise：GetFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*
GetLoggingOptions	iotfleetwise：GetLoggingOptions	
GetModelManifest	iotfleetwise：GetModelManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name*
GetRegisterAccountStatus	iotfleetwise：GetRegisterAccountStatus	

AWS IoT FleetWise 動作	所需的許可	資源
GetSignalCatalog	iotfleetwise：GetSignalCatalog	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
GetStateTemplate	iotfleetwise：GetStateTemplate	arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*
GetVehicle	iotfleetwise：GetVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id*
GetVehicleStatus	iotfleetwise：GetVehicleStatus	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id*
ImportDecoderManifest	iotfleetwise：ImportDecoderManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
ImportSignalCatalog	iotfleetwise：ImportSignalCatalog	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
ListCampaigns	iotfleetwise：ListCampaigns	
ListDecoderManifestNetworkInterfaces	iotfleetwise：ListDecoderManifestNetworkInterfaces	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
ListDecoderManifests	iotfleetwise：ListDecoderManifests	

AWS IoT FleetWise 動作	所需的許可	資源
ListDecoderManifestSignals	iotfleetwise：ListDecoderManifestSignals	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*
ListFleets	iotfleetwise：ListFleets	
ListFleetsForVehicle	iotfleetwise：ListFleetsForVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id*
ListModelManifestNodes	iotfleetwise：ListModelManifestNodes	arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name*
ListModelManifests	iotfleetwise：ListModelManifests	
ListSignalCatalogNodes	iotfleetwise：ListSignalCatalogNodes	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
ListSignalCatalogs	iotfleetwise：ListSignalCatalogs	
ListStateTemplates	iotfleetwise：ListStateTemplates	
ListVehicles	iotfleetwise：ListVehicles	
ListVehiclesInFleet	iotfleetwise：ListVehiclesInFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*

AWS IoT FleetWise 動作	所需的許可	資源
ListTagsForResource	iotfleetwise：ListTagsForResource	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id* arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name* arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*
PutEncryptionConfiguration	iotfleetwise：PutEncryptionConfiguration	
PutLoggingOptions	iotfleetwise：PutLoggingOptions	
RegisterAccount	iotfleetwise：RegisterAccount	

AWS IoT FleetWise 動作	所需的許可	資源
TagResource	iotfleetwise：TagResource	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id* arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name* arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*

AWS IoT FleetWise 動作	所需的許可	資源
UntagResource	iotfleetwise：Untag Resource	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id* arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name* arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*
UpdateCampaign	iotfleetwise：UpdateCampaign	arn：*partition*：iotfleetwise：*region*：*account_id*：campaign/*campaign-name*
UpdateDecoderManifest	iotfleetwise：UpdateDecoderManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*

AWS IoT FleetWise 動作	所需的許可	資源
UpdateFleet	iotfleetwise：UpdateFleet	arn：*partition*：iotfleetwise：*region*：*account_id*：fleet/*fleet-id*
UpdateModelManifest	iotfleetwise：UpdateModelManifest	arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name*
UpdateSignalCatalog	iotfleetwise：UpdateSignalCatalog	arn：*partition*：iotfleetwise：*region*：*account_id*：signal-catalog/*name*
UpdateStateTemplate	iotfleetwise：UpdateStateTemplate	arn：*partition*：iotfleetwise：*region*：*account_id*：state-template/*state-template-id*
UpdateVehicle	iotfleetwise：UpdateVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*

AWS IoT FleetWise 動作	所需的許可	資源
BatchUpdateVehicle	iotfleetwise：UpdateVehicle	arn：*partition*：iotfleetwise：*region*：*account_id*：vehicle/*vehicle-id* arn：*partition*：iotfleetwise：*region*：*account_id*：model-manifest/*name* arn：*partition*：iotfleetwise：*region*：*account_id*：decoder-manifest/*name*

AWS IoT FleetWise 的合規驗證

> **Note**
>
> AWS IoT FleetWise 不在任何 AWS 合規計劃的範圍內。

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱 AWS Compliance Programs。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告下載 AWS Artifact。

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- 安全合規與治理 - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- HIPAA 合格服務參考 – 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- AWS 合規資源 – 此工作手冊和指南集合可能適用於您的產業和位置。
- AWS 客戶合規指南 – 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構（包括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)) 保護 AWS 服務和映射指南至安全控制的最佳實務。

- 《 AWS Config 開發人員指南》中的使用 規則評估資源 – AWS Config 服務會評估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 可讓您全面檢視其中的安全狀態 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 控制參考」。
- Amazon GuardDuty – 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- AWS Audit Manager – 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和業界標準的方式。

AWS IoT FleetWise 中的彈性

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。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 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

> **Note**
>
> AWS IoT FleetWise 處理的資料會存放在 Amazon Timestream 資料庫中。Timestream 支援備份到其他 AWS 可用區域或區域。不過，您可以使用 Timestream SDK 撰寫自己的應用程式來查詢資料，並將其儲存至您選擇的目的地。
> 如需 Amazon Timestream 的詳細資訊，請參閱《Amazon Timestream 開發人員指南》中的
> 。
> Amazon Timestream 不適用於亞太區域（孟買）區域。

AWS IoT FleetWise 中的基礎設施安全

作為受管服務， AWS IoT FleetWise 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及 如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 AWS IoT FleetWise。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 操作，但 AWS IoT FleetWise 支援以資源為基礎的存取政策，其中可能包含根據來源 IP 地址的限制。您也可以使用 AWS IoT FleetWise 政策來控制來自特定 Amazon Virtual Private Cloud (Amazon VPC) 端點或特定 VPCs存取。實際上，這只會隔離網路中特定 VPC 對特定 AWS IoT FleetWise 資源 AWS 的網路存取。

主題

- 透過介面 VPC 端點連線至 AWS IoT FleetWise

透過介面 VPC 端點連線至 AWS IoT FleetWise

您可以使用虛擬私有雲端 (VPC) 中的介面 VPC 端點 (AWS PrivateLink) 直接連線至 AWS IoT FleetWise，而不是透過網際網路連線。當您使用界面 VPC 端點時，VPC 和 AWS IoT FleetWise 之間的通訊會完全在 AWS 網路中執行。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。

介面 VPC 端點會將您的 VPC 直接連接到 AWS IoT FleetWise，無需網際網路閘道、NAT 裝置、VPN 連接或 AWS Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，即可與 AWS IoT FleetWise API 通訊。

若要透過 VPC 使用 AWS IoT FleetWise，您必須從 VPC 內的執行個體連線，或使用 AWS Virtual Private Network (VPN) 或 將私有網路連線至 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。如需 的詳細資訊 AWS Direct Connect，請參閱AWS Direct Connect 《 使用者指南》中的建立連線。

您可以使用 主控台或 AWS Command Line Interface (AWS CLI) 命令，建立介面 VPC 端點以連線至 AWS IoT FleetWise AWS 。如需詳細資訊，請參閱建立介面端點。

建立介面 VPC 端點之後，如果您為端點啟用私有 DNS 主機名稱，預設 AWS IoT FleetWise 端點會解析為您的 VPC 端點。IoT AWS IoT FleetWise 的預設服務名稱端點格式如下。

```
iotfleetwise.Region.amazonaws.com
```

如果您未啟用私有 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》中的界面 VPC 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。

> **Note**
>
> AWS IoT FleetWise 支援具有雙堆疊模式的所有 VPC 端點。如需服務端點的相關資訊，請參閱 AWS IoT FleetWise 端點和配額。

為 AWS IoT FleetWise 建立 VPC 端點政策

您可以為 Amazon VPC 端點 for AWS IoT FleetWise 建立政策，以指定下列項目：

- 可執行或不可執行動作的主體
- 可執行或無法執行的動作

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 – 拒絕來自指定 AWS 帳戶的所有存取的 VPC 端點政策

下列 VPC 端點政策拒絕 AWS 帳戶 *123456789012* 使用端點的所有 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 存取的 VPC 端點政策

下列 VPC 端點政策僅允許完整存取 AWS 帳戶 *123456789012* 中的使用者 *lijuan*。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 端點政策

以下是 AWS IoT FleetWise 端點政策的範例。連接到端點時，此政策會授予 *123456789012* 中 IAM 使用者 fleetWise 的 listed AWS IoT FleetWise 動作存取權 AWS 帳戶 。 FleetWise *fleetWise*

```
{
    "Statement": [
        {
             "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/fleetWise"
```

```
                },
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "iotfleetwise:ListFleets",
                "iotfleetwise:ListCampaigns",
                "iotfleetwise:CreateVehicle",
            ]
           }
    ]
}
```

AWS IoT FleetWise 中的組態和漏洞分析

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，由於裝置通常受限於運算能力、記憶體和儲存功能，因此在裝置上使用加密和其他形式的安全性會受到限制。裝置通常會使用具有已知漏洞的軟體。這些因素使得 IoT 裝置包括收集 AWS IoT FleetWise 資料的車輛成為駭客的有吸引力目標，並且難以持續保護裝置。

組態和 IT 控制是客戶 AWS 與您之間的共同責任。如需詳細資訊，請參閱 AWS 共同的責任模型。

AWS IoT FleetWise 的安全最佳實務

AWS IoT FleetWise 提供許多安全功能，供您在開發和實作自己的安全政策時考慮。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。

若要了解 中的安全， AWS IoT 請參閱《 AWS IoT 開發人員指南》中的 中的安全最佳實務 AWS IoT Core

盡可能授予最低的許可

在 IAM 角色中使用最低權限集，以遵循最低權限原則。限制對 IAM 政策中的 `Action`和 `Resource` 屬性使用*萬用字元。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)。我們建議您實作下列保護措施：

- 請勿在裝置名稱中使用敏感資訊。
- 請勿在 AWS IoT FleetWise 資源的名稱和 IDs 中使用敏感資訊，例如行銷活動名稱、解碼器資訊清單、車輛模型和訊號目錄，或車輛和機群的 IDs。

使用 AWS CloudTrail 來檢視 API 呼叫歷史記錄

您可以檢視在您的帳戶上進行的 AWS IoT FleetWise API 呼叫的歷史記錄，用於安全性分析和操作故障診斷。若要接收對您的帳戶發出的 AWS IoT FleetWise API 呼叫歷史記錄，只需在 中開啟 CloudTrail 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《 開發人員指南》中的保持裝置的時鐘同步最佳實務。 AWS IoT Core

Monitor AWS IoT FleetWise

監控是維護 AWS IoT FleetWise 和其他 AWS 解決方案可靠性、可用性和效能的重要部分。 AWS 提供下列監控工具來監看 AWS IoT FleetWise、報告錯誤，並適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。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，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。
- Amazon CloudWatch Logs 可用來監控、存放和存取來自 Amazon EC2 執行個體、CloudTrail 和其他來源的日誌檔案。CloudWatch Logs 可監控日誌檔案中的資訊，並在達到特定閾值時通知您。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 Logs 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，它會將日誌檔案交付到您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。

使用 Amazon CloudWatch 監控 AWS IoT FleetWise Amazon CloudWatch

> **Important**
>
> 某些 AWS IoT FleetWise 功能的存取目前已封鎖。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

Amazon CloudWatch 指標是監控 AWS 資源及其執行方式的一種方式。 AWS IoT FleetWise 會將指標傳送至 CloudWatch。您可以使用 AWS Management Console、 AWS CLI或 API 來列出 AWS IoT FleetWise 傳送至 CloudWatch 的指標。如需更多資訊，請參閱 Amazon CloudWatch 使用者指南。

> **Important**
>
> 您必須設定 設定，IoT AWS IoT FleetWise 才能將指標傳送至 CloudWatch。如需詳細資訊，請參閱設定您的 AWS IoT FleetWise 設定。

`AWS/IoTFleetWise` 命名空間包含下列指標。

訊號指標

指標	描述
IllegalMessageFromEdge	從車輛傳送並由 AWS IoT FleetWise 接收的訊息不符合所需的格式。 單位：計數 維度：無 有效的統計資訊：總和
MessageThrottled	從車輛傳送至 AWS IoT FleetWise 的訊息已調節。這是因為您在目前區域中超過此帳戶的服務限制。 單位：計數 維度：無 有效的統計資訊：總和
ModelingError	從車輛傳送並由 AWS IoT FleetWise 接收的訊息包含無法驗證車輛模型的訊號。 單位：計數 維度：ModelName、StateTemplateName（選用）、 SignalCatalogName（選用）
DecodingError	從車輛傳送並由 AWS IoT FleetWise 接收的訊息包含無法針對車輛的解碼器資訊清單解碼的訊號。 單位：計數 維度：DecoderName 有效的統計資訊：總和

指標	描述
MessageSizeLimitExceeded	從車輛傳送至 AWS IoT FleetWise 的訊息已捨棄。這是因為您在目前區域中超過此帳戶的訊息服務限制大小上限。 單位：計數 維度：無 有效的統計資訊：總和

車輛指標

指標	描述
VehicleNotFound	AWS IoT FleetWise 收到的訊息，其中車輛未知。 單位：計數 維度：無 有效的統計資訊：總和

行銷活動指標

指標	描述
CampaignInvalid	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中行銷活動無效。 單位：計數 維度：CampaignName 有效的統計資訊：總和
CampaignNotFound	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中行銷活動不明。

指標	描述
	單位：計數 維度：CampaignName 有效的統計資訊：總和

狀態範本指標

指標	描述
NoStateTemplatesAssociated	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中沒有與車輛相關聯的狀態範本。 單位：計數 有效的統計資訊：總和

行銷活動資料目的地指標

指標	描述
TimestreamWriteError	AWS IoT FleetWise 無法將訊息從車輛寫入 Amazon Timestream 資料表。 單位：計數 維度： DatabaseName、TableName 有效的統計資訊：總和
S3WriteError	AWS IoT FleetWise 無法將訊息從車輛寫入 Amazon Simple Storage Service (Amazon S3) 儲存貯體。 單位：計數 維度：BucketName

指標	描述
	有效的統計資訊：總和
S3ReadError	AWS IoT FleetWise 無法讀取 Amazon Simple Storage Service (Amazon S3) 儲存貯體中車輛的物件金鑰。 單位：計數 維度：BucketName 有效的統計資訊：總和

客戶受管 AWS KMS 金鑰指標

指標	描述
KMSKeyAccessDenied	AWS IoT FleetWise 無法將訊息從車輛寫入 Timestream 資料表或 Amazon S3 儲存貯體，因為 AWS KMS 金鑰存取遭拒錯誤。 單位：計數 維度：KMSKeyId 有效的統計資訊：總和

使用 Amazon CloudWatch Logs 監控 AWS IoT FleetWise Amazon CloudWatch

> **Important**
>
> 目前已封鎖對特定 AWS IoT FleetWise 功能的存取。如需詳細資訊，請參閱AWSAWS IoT FleetWise 中的區域和功能可用性。

Amazon CloudWatch Logs 會監控資源中發生的事件，並在發生任何問題時提醒您。如果您收到提醒，您可以存取日誌檔案，以取得特定事件的相關資訊。如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南。

在 CloudWatch 主控台中檢視 AWS IoT FleetWise 日誌

> **Important**
>
> 在 CloudWatch 主控台中看到 AWS IoT FleetWise 日誌群組之前，請確定下列項目為真：
>
> - 您已在 AWS IoT FleetWise 中啟用記錄。如需記錄的詳細資訊，請參閱 Configure AWS IoT FleetWise 記錄。
> - 操作已寫入日誌項目 AWS IoT 。

在 CloudWatch 主控台中檢視您的 AWS IoT FleetWise 日誌

1. 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇日誌、日誌群組。
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產生的日誌事件完整清單。
5. 選擇展開圖示以查看個別串流，並尋找日誌層級為 的所有日誌ERROR。

   您也可以在篩選事件搜尋方塊中輸入查詢。例如，您可以嘗試下列查詢：

   ```
   { $.logLevel = "ERROR" }
   ```

如需建立篩選條件表達式的詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的篩選條件和模式語法。

Example 日誌項目

```
{
  "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
MODELING_ERROR	從車輛傳送並由 AWS IoT FleetWise 接收的訊息包含無法驗證車輛模型的訊號。 屬性： vehicleName、campaignName（選用）、signalCatalogName、signalId（選用）、signalValue（選用）、signalValueRangeMin（選用）、signalValueRangeMax（選用）、modelManifestName（選用）、signalIds、stateTemplateName
ILLEGAL_MESSAGE_FROM_EDGE	從車輛傳送並由 AWS IoT FleetWise 接收的訊息不符合所需的格式。 屬性： vehicleName、campaignName、signalCatalogName
DECODING_ERROR	從車輛傳送並由 AWS IoT FleetWise 接收的訊息包含無法針對車輛的解碼器資訊清單解碼的訊號。 屬性：campaignName、 signalCatalogName、decoderManifestName、（選用） signalName、（選用） s3URI
MESSAGE_THROTTLED	從車輛傳送至 AWS IoT FleetWise 的訊息已調節。這是因為您超過目前區域中此帳戶的服務限制。 屬性：accountId、 vehicleName、訊息、eventType、logLevel、時間戳記

事件類型	描述
MESSAGE_SIZE_LIMIT_EXCEEDED	從車輛傳送並由 AWS IoT FleetWise 接收的訊息超過訊息服務限制的大小上限。 屬性：accountId、 vehicleName

車輛事件類型

事件類型	描述
VEHICLE_NOT_FOUND	AWS IoT FleetWise 收到的訊息，其中車輛不明。 屬性： vehicleName、campaignName（選用）、stateTemplateName （選用）

行銷活動事件類型

事件類型	描述
CAMPAIGN_NOT_FOUND	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中行銷活動不明。 屬性： vehicleName （選用）、campaignName
CAMPAIGN_INVALID	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中行銷活動無效。 屬性： vehicleName （選用）、campaignName

行銷活動資料目的地事件類型

事件類型	描述
TIMESTREAM_WRITE_ERROR	AWS IoT FleetWise 無法將訊息從車輛寫入 Amazon Timestream 資料表。 屬性： vehicleName、campaignName、timestreamDatabaseName、timestreamTableName
S3_WRITE_ERROR	AWS IoT FleetWise 無法將訊息從車輛寫入 Amazon Simple Storage Service (Amazon S3) 儲存貯體。 屬性：campaignName、destinationName
S3_READ_ERROR	AWS IoT FleetWise 無法從 Amazon Simple Storage Service (Amazon S3) 儲存貯體中的車輛讀取物件金鑰。 屬性：campaignName、destinationName

狀態範本事件類型

事件類型	描述
STATE_TEMPLATE_NOT_FOUND	從車輛傳送並由 AWS IoT FleetWise 接收的訊息，其中狀態範本不明。 屬性： vehicleName （選用）stateTemplateName

客戶受管 AWS KMS 金鑰事件類型

事件類型	描述
KMS_KEY_ACCESS_DENIED	AWS IoT FleetWise 無法將訊息從車輛寫入 Timestream 資料表或 Amazon S3 儲存貯體，因為 AWS KMS 金鑰存取遭拒錯誤。 屬性：kmsKeyId （選用）、resourceArn （選用）

Attributes

所有 CloudWatch Logs 項目都包含這些屬性：

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

正在使用的日誌層級。如需詳細資訊，請參閱《 AWS IoT Core 開發人員指南》中的日誌層級。

message

包含日誌的特定詳細資訊。

timestamp

當 AWS IoT FleetWise 處理日誌時的 epoch 毫秒時間戳記。

選擇性屬性

CloudWatch Logs 項目可選擇包含這些屬性，視 而定`eventType`：

decoderManifestName

包含訊號的解碼器資訊清單名稱。

destinationName

車輛資料的目的地名稱。例如，Amazon S3 儲存貯體名稱。

campaignName

行銷活動的名稱。

signalCatalogName

包含訊號的訊號目錄名稱。

signalId

錯誤訊號的 ID。

signalIds

錯誤訊號 IDs的清單。

signalName

訊號的名稱。

signalTimestampEpochMs

錯誤訊號的時間戳記。

signalValue

錯誤訊號的值。

signalValueRangeMax

錯誤訊號的最大範圍。

signalValueRangeMin

錯誤訊號的最小範圍。

s3URI

車輛訊息中 Amazon Ion 檔案的 Amazon S3 唯一識別符。

timestreamDatabaseName

Timestream 資料庫的名稱。

timestreamTableName

Timestream 資料表的名稱。

vehicleName

車輛名稱。

Configure AWS IoT FleetWise 記錄

您可以將 AWS IoT FleetWise 日誌資料傳送至 CloudWatch 日誌群組。如果 AWS IoT FleetWise 無法處理來自車輛的訊息，CloudWatch Logs 會提供可見性。例如，這可能是因為組態錯誤或其他用戶端錯誤。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 CloudWatch 日誌群組，才能將日誌傳送至 CloudWatch。使用與 AWS IoT FleetWise 搭配使用的相同帳戶和相同區域中設定日誌群組。當您在 AWS IoT FleetWise 中啟用記錄時，請提供日誌群組名稱。啟用記錄後， AWS IoT FleetWise 會將日誌交付至日誌串流中的 CloudWatch 日誌群組。

您可以在 CloudWatch 主控台中檢視從 AWS IoT FleetWise 傳送的日誌資料。如需設定 CloudWatch 日誌群組和檢視日誌資料的詳細資訊，請參閱使用日誌群組。

將日誌發佈至 CloudWatch 的許可

設定 CloudWatch 日誌群組的記錄需要本節所述的許可設定。如需有關管理許可的資訊，請參閱《IAM 使用者指南》中的存取 AWS 資源的管理。

使用這些許可，您可以變更記錄組態、設定 CloudWatch 的日誌交付，以及擷取日誌群組的相關資訊。

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

允許對所有 AWS 資源執行動作時，政策中會顯示動作，其`"Resource"`設定為 `"*"`。這表示每個動作支援的所有 AWS 資源都允許這些動作。

在 AWS IoT FleetWise 中設定記錄（主控台）

本節說明如何使用 AWS IoT FleetWise 主控台來設定記錄。

使用 AWS IoT FleetWise 主控台設定記錄

1. 開啟 AWS IoT FleetWise 主控台。
2. 在左側窗格中，選擇 Settings (設定)。
3. 在設定頁面的記錄區段中，選擇編輯。
4. 在 CloudWatch 記錄區段中，輸入日誌群組。
5. 若要儲存變更，請選擇提交。

啟用記錄後，您可以在 CloudWatch 主控台中檢視日誌資料。

在 AWS IoT FleetWise (CLI) 中設定預設記錄

本節說明如何使用 CLI 設定 for AWS IoT FleetWise 的記錄。

您也可以使用 API 中對應至此處所示 CLI 命令的方法，以 AWS API 執行此程序。您可以使用 GetLoggingOptions API 操作來擷取目前的組態，並使用 PutLoggingOptions API 操作來修改組態。

使用 CLI 設定 AWS IoT FleetWise 的記錄

1. 若要取得帳戶的記錄選項，請使用 get-logging-options命令。

```
aws iotfleetwise get-logging-options
```

2. 若要啟用記錄，請使用 put-logging-options命令。

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
 logType=ERROR,logGroupName=MyLogGroup
```

其中：

logType

將資料傳送至 CloudWatch Logs 的日誌類型。若要停用記錄，請將 值變更為 OFF。

logGroupName

操作傳送資料的目標 CloudWatch Logs 群組。在啟用 for AWS IoT FleetWise 記錄之前，請務必建立日誌群組名稱。

啟用記錄之後，請參閱使用 CLI AWS 搜尋日誌項目。

Log AWS IoT FleetWise API 呼叫使用 AWS CloudTrail

AWS IoT FleetWise 已與 整合 AWS CloudTrail，此服務提供使用者、角色或 AWS IoT FleetWise AWS 服務所採取動作的記錄。CloudTrail 會將 AWS IoT FleetWise 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 AWS IoT FleetWise 主控台呼叫，以及對 AWS IoT FleetWise API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 AWS IoT FleetWise 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷對 AWS IoT FleetWise 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的AWS IoT FleetWise 資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 AWS IoT FleetWise 中發生時，該活動會記錄於 CloudTrail 事件，以及事件歷史記錄中的其他服務 AWS 事件。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 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶中的事件，包括 AWS IoT FleetWise 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。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。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 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

CloudTrail 會記錄所有 AWS IoT FleetWise 動作，並記錄在 AWS IoT FleetWise API 參考中。例如，對 `CreateCampaign`、`AssociateVehicleFleet` 以及 `GetModelManifest` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案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。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AWS IoT FleetWise 日誌檔案項目

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 日誌檔案包含一或多個日誌專案。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 呼叫的堆疊追蹤排序，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 for AWS IoT FleetWise 的文件版本。

變更	描述	日期
區域擴展	AWS IoT FleetWise 現已在亞太區域（孟買）區域提供（僅限門控存取）。	2024 年 11 月 21 日
新功能的門控一般可用性	AWS IoT FleetWise 現在支援行銷活動的門控存取，以存放和轉送資料、將 MQTT 主題設定為資料目的地，以及收集診斷問題碼資料。它現在也支援使用自訂解碼介面、設定命令和監控車輛最後已知狀態的網路無關資料收集的門控存取。	2024 年 11 月 21 日
將行銷活動資料傳送至 MQTT 主題	AWS IoT FleetWise 現在支援將行銷活動期間收集的資料傳送到您指定的 MQTT 主題，以及將資料存放在 Amazon S3 或 Amazon Timestream 的功能。	2024 年 5 月 1 日
視覺系統資料預覽	您可以使用來自 AWS IoT FleetWise 的視覺系統資料的預覽，從車輛視覺系統收集和組織資料，包括攝影機、雷達和光圈。它可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、行銷活動、車輛）和標準感應器（遙測資料）。	2023 年 11 月 26 日
AWS KMS 客戶受管金鑰	AWS IoT FleetWise 現在支援 AWS KMS 客戶受管金鑰。	2023 年 10 月 16 日

	您可以使用 KMS 金鑰來加密與存放在其中的 AWS IoT FleetWise 資源（訊號目錄、車輛模型、解碼器資訊清單、車輛和資料收集行銷活動組態）相關的伺服器端資料 AWS 雲端。	
Amazon S3 中的物件儲存	AWS IoT FleetWise 現在支援使用 Amazon Simple Storage Service (Amazon S3) 儲存資料。除了 Amazon Timestream 之外，您還可以在 Amazon S3 中儲存行銷活動期間收集的資料。	2023 年 6 月 1 日
一般可用性	這是 AWS IoT FleetWise 的公開版本。	2022 年 9 月 27 日
初始版本	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。