개발자 가이드

AWS IoT FleetWise

AWS IoT FleetWise: 개발자 가이드

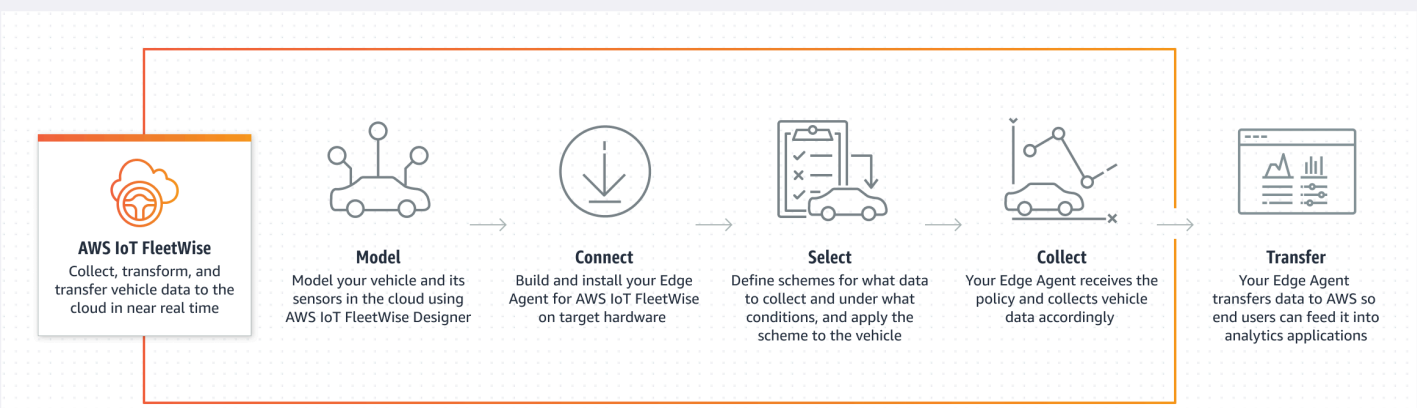
Table of Contents

AWS IoT FleetWise란 무엇입니까?

AWS IoT FleetWise 는 차량 데이터를 수집하고 클라우드에서 구성하는 데 사용할 수 있는 관리형 서비스입니다. 수집된 데이터를 사용하여 차량 품질, 성능 및 자율성을 개선할 수 있습니다. AWS IoT FleetWise를 사용하면 다양한 프로토콜과 데이터 형식을 사용하는 차량에서 데이터를 수집하고 구성할 수 있습니다. AWS IoT FleetWise 는 하위 수준 메시지를 사람이 읽을 수 있는 값으로 변환하고 데이터 분석을 위해 클라우드의 데이터 형식을 표준화하는 데 도움이 됩니다. 또한 데이터 수집 캠페인을 정의하여 수집할 차량 데이터와 해당 데이터를 클라우드로 전송할 시기를 제어할 수 있습니다.

차량 데이터가 클라우드에 있으면 플릿 상태를 분석하는 애플리케이션에 사용할 수 있습니다. 이 데이터는 잠재적인 유지보수 문제를 식별하고, 차량 내 인포테인먼트 시스템을 더 스마트하게 만들고, 분석 및 기계 학습(ML)을 통해 자율 주행 및 운전자 지원 시스템과 같은 고급 기술을 개선할 수 있습니다.

다음 다이어그램은 AWS IoT FleetWise 의 기본 아키텍처를 보여줍니다.



주제

이점

AWS IoT FleetWise 의 주요 이점은 다음과 같습니다.

보다 지능적으로 차량 데이터 수집

분석을 위해 필요한 데이터만 클라우드로 전송하는 지능형 데이터 수집을 통해 데이터 관련성을 개선할 수 있습니다.

표준화된 플릿 전체 데이터를 쉽게 분석

사용자 지정 데이터 수집 또는 로깅 시스템을 개발할 필요 없이 차량 플릿의 표준화된 데이터를 분석할 수 있습니다.

클라우드에서 자동으로 데이터 동기화

표준 센서(텔레메트리 데이터)와 비전 시스템(카메라, 레이더 및 라이더의 데이터) 모두에서 수집된 데이터를 통합적으로 확인하고 클라우드에서 자동으로 동기화합니다. AWS IoT FleetWise 는 구조화 및 비구조화 비전 시스템 데이터, 메타데이터 및 표준 센서 데이터를 클라우드에서 자동으로 동기화합니다. 이를 통해 이벤트를 전체적으로 파악하고 인사이트를 얻는 프로세스가 간소화됩니다.

> **Note**
>
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

사용 사례

AWS IoT FleetWise 를 사용할 수 있는 시나리오에는 다음이 포함됩니다.

AI/ML 모델 훈련

프로덕션 차량에서 데이터를 수집하여 자율 주행 및 첨단 운전자 지원 시스템에 사용되는 기계 학습 모델을 지속적으로 개선합니다.

디지털 고객 경험 향상

인포테인먼트 시스템의 데이터를 사용하여 차량 내 시청각 콘텐츠와 인앱 인사이트를 보다 관련성 있게 만듭니다.

차량 플릿 상태 유지

플릿 데이터에서 얻은 인사이트를 사용하여 EV 배터리 상태 및 충전 수준을 모니터링하고, 유지 보수 일정을 관리하고, 연료 소비를 분석하는 등의 작업을 수행합니다.

AWS IoT FleetWise를 처음 사용하시나요?

AWS IoT FleetWise 처음 사용하는 경우 다음 섹션을 읽는 것으로 시작하는 것이 좋습니다.

- AWS IoT FleetWise의 주요 개념 및 기능
- AWS IoT FleetWise 설정
- 자습서: AWS IoT FleetWise 시작하기
- 클라우드에 AWS IoT FleetWise 데이터 삽입

AWS IoT FleetWise 액세스

AWS IoT FleetWise 콘솔 또는 를 사용하여 AWS IoT FleetWiseAPI에 액세스할 수 있습니다.

AWS IoT FleetWise 요금

차량은 MQTT 메시지를 통해 클라우드로 데이터를 전송합니다. AWS IoT FleetWise 에서 생성한 차량에 대해 매월 말에 요금을 지불합니다. 또한 차량에서 수집한 메시지에 대한 비용도 지불합니다. 요금에 대한 최신 정보는 AWS IoT FleetWise 요금 페이지를 참조하세요. MQTT 메시징 프로토콜에 대한 자세한 내용은 개발자 안내서MQTT의 섹션을 참조하세요. AWS IoT Core

AWS IoT FleetWise의 주요 개념 및 기능

다음 섹션에서는 AWS IoT FleetWise 서비스 구성 요소와 이러한 구성 요소가 상호 작용하는 방식에 대한 개요를 제공합니다.

이 소개를 읽은 후 AWS IoT FleetWise 설정 섹션을 참조하여 AWS IoT FleetWise를 설정하는 방법을 알아봅니다.

주제

- 주요 개념
- AWS IoT FleetWise의 기능

주요 개념

AWS IoT FleetWise 는 클라우드에서 차량과 해당 센서 및 액추에이터를 모델링할 수 있는 차량 모델링 프레임워크를 제공합니다. 차량과 클라우드 간의 보안 통신을 활성화하기 위해 AWS IoT FleetWise 는 차량에 설치할 수 있는 Edge Agent 소프트웨어를 개발하는 데 도움이 되는 참조 구현도 제공합니다. 클라우드에서 데이터 수집 체계를 정의하고 이를 차량에 배포할 수 있습니다. 차량에서 실행 중인 Edge Agent 소프트웨어는 데이터 수집 체계를 사용하여 수집할 데이터와 클라우드로 전송할 시기를 제어합니다.

다음은 AWS IoT FleetWise 의 핵심 개념입니다.

신호

신호는 차량 데이터와 해당 메타데이터를 포함하도록 정의하는 기본 구조입니다. 신호는 속성, 분기, 센서 또는 액추에이터일 수 있습니다. 예를 들어, 차량 내 온도 값을 수신하고 센서 이름, 데이터 유형 및 단위를 포함한 메타데이터를 저장하는 센서를 생성할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리 단원을 참조하십시오.

속성

속성은 일반적으로 변경되지 않는 정적 정보(예: 제조업체 및 제조일)를 나타냅니다.

브랜치

브랜치는 신호를 중첩된 구조로 나타냅니다. 브랜치는 신호 계층 구조를 보여줍니다. 예를 들어, `Vehicle` 브랜치에는 하위 브랜치, `Powertrain`이 있습니다. `Powertrain` 브랜치에는 하위 브랜치, `combustionEngine`이 있습니다. `combustionEngine` 브랜치를 찾으려면 `Vehicle.Powertrain.combustionEngine` 표현식을 사용하세요.

센서

센서 데이터는 차량의 현재 상태와 시간에 따른 변화(예: 유체 수준, 온도, 진동, 전압 등)를 보고합니다.

액추에이터

액추에이터 데이터는 모터, 히터, 도어록과 같은 차량 장치의 상태를 보고합니다. 차량 장치의 상태를 변경하면 액추에이터 데이터를 업데이트할 수 있습니다. 예를 들어, 히터를 나타내는 액추에이터를 정의할 수 있습니다. 히터를 켜거나 끌 때 액추에이터가 새 데이터를 수신합니다.

사용자 지정 구조

사용자 지정 구조(구조체라고도 함)는 복합 또는 고차 데이터 구조를 나타냅니다. 이를 통해 동일한 소스에서 생성된 데이터를 쉽게 논리적으로 바인딩하거나 그룹화할 수 있습니다. 구조체는 복합 데

이터 유형 또는 고차 모양을 나타내는 것과 같이 원자성 연산으로 데이터를 읽거나 쓸 때 사용됩니다.

구조체 유형의 신호는 프리미티브 데이터 유형 대신 구조체 데이터 유형에 대한 참조를 사용하여 신호 카탈로그에 정의됩니다. 구조체는 센서, 속성, 액추에이터, 비전 시스템 데이터 유형을 비롯한 모든 유형의 신호에 사용할 수 있습니다. 구조 유형의 신호가 전송되거나 수신되는 경우 AWS IoT FleetWise 는 포함된 모든 항목에 유효한 값이 있을 것으로 예상하므로 모든 항목은 필수입니다. 예를 들어 구조체에 Vehicle.Camera.Image.height, Vehicle.Camera.Image.width 및 Vehicle.Camera.Image.data 항목이 포함된 경우 송신된 신호에 이러한 모든 항목의 값이 포함될 것으로 예상됩니다.

> **Note**
>
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

사용자 지정 속성

사용자 지정 속성은 복합 데이터 구조의 멤버를 나타냅니다. 속성의 데이터 유형은 프리미티브이거나 다른 구조체일 수 있습니다.

구조체와 사용자 지정 속성을 사용하여 고차 모양을 표현할 때는 의도한 고차 모양이 항상 트리 구조로 정의되고 표시됩니다. 사용자 지정 속성은 모든 리프 노드를 정의하는 데 사용되고 구조체는 리프가 아닌 모든 노드를 정의하는 데 사용됩니다.

신호 카탈로그

신호 카탈로그에는 신호 컬렉션이 포함되어 있습니다. 신호 카탈로그에 있는 신호는 다양한 프로토콜과 데이터 형식을 사용하는 차량을 모델링할 때 사용할 수 있습니다. 예를 들어, 서로 다른 자동차 제조업체에서 만든 자동차는 두 대가 있습니다. 하나는 제어 영역 네트워크(CAN 버스) 프로토콜을 사용하고 다른 하나는 온보드 진단(OBD) 프로토콜을 사용합니다. 신호 카탈로그에서 센서를 정의하여 차량 내 온도 값을 수신할 수 있습니다. 이 센서는 두 차량의 열전대를 나타내는 데 사용할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리 단원을 참조하십시오.

차량 모델(모델 매니페스트)

차량 모델은 차량 형식을 표준화하고 차량 내 신호 간의 관계 정의에 사용할 수 있는 선언적 구조입니다. 차량 모델은 동일한 유형의 다중 차량에 일관된 정보를 적용합니다. 신호를 추가하여 차량 모델을 생성합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 관리 단원을 참조하십시오.

디코더 매니페스트

디코더 매니페스트에는 차량 모델의 각 신호에 대한 디코딩 정보가 포함되어 있습니다. 차량의 센서와 액추에이터는 저수준 메시지(바이너리 데이터)를 전송합니다. 디코더 매니페스트를 사용하면 AWS IoT FleetWise 가 바이너리 데이터를 사람이 읽을 수 있는 값으로 변환할 수 있습니다. 모든 디코더 매니페스트는 차량 모델과 연결됩니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

네트워크 인터페이스

차량 내 네트워크가 사용하는 프로토콜에 대한 정보를 포함합니다. AWS IoT FleetWise 는 다음 프로토콜을 지원합니다.

컨트롤러 영역 네트워크(CAN 버스)

전자 제어 장치() 간에 데이터가 통신되는 방식을 정의하는 프로토콜입니다ECUs. ECUs 는 엔진 컨트롤 유닛, 에어백 또는 오디오 시스템일 수 있습니다.

온보드 진단(OBD) II

자체 진단 데이터가 간에 전달되는 방법을 정의하는 추가 개발 프로토콜입니다ECUs. 차량에 무엇이 잘못되었는지 식별하는 데 도움이 되는 여러 표준 진단 문제 코드(DTCs)를 제공합니다.

차량 미들웨어

네트워크 인터페이스 유형으로 정의되는 차량 미들웨어입니다. 차량 미들웨어의 예로는 로봇 운영 체제(ROS 2)와 확장 가능한 서비스 지향 MiddlewarE over IP(SOME/IP)가 있습니다.

> **Note**
>
> AWS IoT FleetWise 는 비전 시스템 데이터에 대한 미들웨어 ROS 2개를 지원합니다.

디코더 신호

특정 신호에 대한 자세한 디코딩 정보를 제공합니다. 차량 모델에 지정된 모든 신호는 디코더 신호와 페어링되어야 합니다. 디코더 매니페스트에 CAN 네트워크 인터페이스가 포함된 경우 CAN 디코더 신호가 포함되어야 합니다. 디코더 매니페스트에 OBD 네트워크 인터페이스가 포함된 경우 OBD 디코더 신호가 포함되어야 합니다.

디코더 매니페스트에 차량 미들웨어 인터페이스도 포함되어 있는 경우 메시지 디코더 신호가 포함되어야 합니다.

차량

실제 차량(예: 자동차 또는 트럭)을 가상으로 표현한 것입니다. 차량은 차량 모델의 인스턴스입니다. 동일한 차량 모델에서 생성된 차량은 동일한 신호 그룹을 상속받습니다. 각 차량은 AWS IoT 사물에 해당합니다.

플릿

플릿은 차량 그룹을 나타냅니다. 여러 차량을 쉽게 관리하려면 먼저 개별 차량을 플릿에 연결해야 합니다.

Campaign

데이터 수집 체계를 포함합니다. 클라우드에서 캠페인을 정의하고 이를 차량 또는 플릿에 배포합니다. 캠페인은 Edge Agent 소프트웨어 지침에 따라 데이터를 선택하고 수집하고 클라우드로 전송하는 방법을 제공합니다.

데이터 수집 체계

데이터 수집 체계는 Edge Agent 소프트웨어에 데이터 수집 방법에 대한 지침을 제공합니다. 현재 AWS IoT FleetWise 는 조건 기반 수집 체계와 시간 기반 수집 체계를 지원합니다.

조건 기반 수집 체계

논리적 표현식을 사용하여 수집할 데이터를 인식합니다. Edge Agent 소프트웨어는 조건이 충족되는 경우 데이터를 수집합니다. 예를 들어, 표현식이 `$variable.myVehicle.InVehicleTemperature >35.0`인 경우 Edge Agent 소프트웨어는 35.0보다 큰 온도 값을 수집합니다.

시간 기반 수집 체계

밀리초 단위로 기간을 지정하여 데이터 수집 주기를 정의합니다. 예를 들어, 기간이 10,000밀리초인 경우 Edge Agent 소프트웨어는 10초마다 한 번씩 데이터를 수집합니다.

AWS IoT FleetWise의 기능

다음은 AWS IoT FleetWise 의 주요 기능입니다.

차량 모델링

차량의 가상 표현을 구축하고 공통 형식을 적용하여 차량 신호를 구성합니다. AWS IoT FleetWise 는 차량 신호를 표준화하는 데 사용할 수 있는 차량 신호 사양(VSS)을 지원합니다.

체계 기반 데이터 수집

가치가 높은 차량 데이터만 클라우드로 전송하는 방식을 정의하합니다. 40도를 초과하는 차량 내 온도 값 데이터와 같이 수집할 데이터를 제어하는 조건 기반 체계를 정의할 수 있습니다. 데이터 수집 빈도를 제어하는 시간 기반 체계를 정의할 수도 있습니다.

Edge Agent for AWS IoT FleetWise 소프트웨어

차량에서 실행되는 Edge Agent 소프트웨어는 차량과 클라우드 간의 통신을 용이하게 합니다. 차량이 클라우드에 연결되어 있는 동안 Edge Agent 소프트웨어는 지속적으로 데이터 수집 체계를 수신하고 그에 따라 데이터를 수집합니다.

관련 서비스

AWS IoT FleetWise 는 다음 AWS 서비스와 통합되어 클라우드 솔루션의 가용성과 확장성을 개선합니다.

- AWS IoT Core – 차량 데이터를 AWS IoT FleetWise에 업로드하는 AWS IoT 디바이스를 등록하고 제어합니다. 자세한 내용은 AWS IoT 개발자 가이드의 AWS IoT이란 무엇입니까?를 참조하세요.
- Amazon Timestream — 시계열 데이터베이스를 사용하여 차량 데이터를 저장하고 분석합니다. 자세한 내용을 알아보려면 Amazon Timestream 개발자 안내서의 Timestream은 무엇인가를 참조하세요.
- Amazon S3 — 객체 스토리지 서비스를 사용하여 차량 데이터를 저장하고 관리합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3는 무엇인가를 참조하세요.

AWS IoT FleetWise 설정

AWS IoT FleetWise 를 처음 사용하기 전에 다음 섹션의 단계를 완료하세요.

주제

설정 AWS 계정

다음 작업을 완료하여 에 가입 AWS 하고 관리 사용자를 생성합니다.

에 가입 AWS 계정

가 없는 경우 다음 단계를 AWS 계정완료하여 를 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/가입 을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정AWS 계정 루트 사용자가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/로 이동하여 내 계정 을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 AWS 계정 루트 사용자를 AWS 계정보호하고, 를 활성화하고 AWS IAM Identity Center, 관리 사용자를 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로 에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 User Guide의 루트 사용자로 로그인을 참조하십시오.
2. 루트 사용자에 대해 다중 인증(MFA)을 켭니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송URL된 로그인을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 권한을 적용하는 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

> **Note**
>
> AWS IoT FleetWise 에서 서비스 연결 역할을 사용할 수 있습니다. 서비스 연결 역할은 AWS IoT FleetWise 에서 사전 정의하며 AWS IoT FleetWise 가 Amazon 에 지표를 보내는 데 필요한 권한을 포함합니다 CloudWatch. 자세한 내용은 AWS IoT FleetWise에 서비스 연결 역할 사용 단원을 참조하십시오.

콘솔에서 시작하기

아직 에 로그인하지 않은 경우 AWS 계정로그인한 다음 AWS IoT FleetWise 콘솔 을 엽니다. AWS IoT FleetWise를 시작하려면 차량 모델을 생성합니다. 차량 모델은 차량 형식을 표준화합니다.

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 시작하기 AWS IoT FleetWise에서 시작하기를 선택합니다.

차량 모델 생성에 대한 자세한 내용은 차량 모델 생성(콘솔)을 참조하세요.

AWS IoT FleetWise 설정 구성

AWS IoT FleetWise 콘솔 또는 API 를 사용하여 Amazon CloudWatch Logs 지표, Amazon CloudWatch Logs 및 를 사용한 데이터 암호화에 대한 설정을 구성할 수 있습니다 AWS 관리형 키.

CloudWatch 지표를 사용하면 AWS IoT FleetWise 및 기타 AWS 리소스를 모니터링할 수 있습니다. 지표를 사용하여 CloudWatch 예를 들어 서비스 제한이 초과되었는지 확인하는 등의 지표를 수집하고 추적할 수 있습니다. CloudWatch 지표에 대한 자세한 내용은 섹션을 참조하세요Amazon을 통한 Monitor AWS IoT FleetWise CloudWatch.

CloudWatch Logs를 사용하면 AWS IoT FleetWise 는 로그 데이터를 CloudWatch 로그 그룹으로 전송하여 이를 사용하여 문제를 식별하고 완화할 수 있습니다. CloudWatch 로그에 대한 자세한 내용은 섹션을 참조하세요AWS IoT FleetWise 로깅 구성.

AWS IoT FleetWise 는 데이터 암호화를 사용하여 데이터를 암호화 AWS 관리형 키 합니다. 를 사용하여 키를 생성하고 관리하도록 선택할 수도 있습니다 AWS KMS. 암호화에 대한 자세한 내용은 AWS IoT FleetWise의 데이터 암호화 섹션을 참조하세요.

설정 구성(콘솔)

아직 에 로그인하지 않은 경우 AWS 계정로그인한 다음 AWS IoT FleetWise 콘솔 을 엽니다.

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 왼쪽 창에서 설정을 선택합니다.
3. 지표 에서 활성화를 선택합니다. AWS IoT FleetWise 는 CloudWatch 관리형 정책을 서비스 연결 역할에 자동으로 연결하고 CloudWatch 지표를 활성화합니다.
4. 로깅에서 편집을 선택합니다.
   a. CloudWatch 로깅 섹션에서 로그 그룹 을 입력합니다.
   b. 변경 사항을 저장하려면, 제출을 선택합니다.
5. 암호화 섹션에서 편집을 선택합니다.
   a. 사용하려는 키 타입을 선택합니다. 자세한 내용은 AWS IoT FleetWise의 키 관리 단원을 참조하십시오.
      i. AWS 키 사용 - AWS IoT FleetWise 가 키를 소유하고 관리합니다.
      ii. 다른 AWS Key Management Service 키 선택 - 계정에 AWS KMS keys 있는 를 관리합니다.
   b. 변경 사항을 저장하려면, 제출을 선택합니다.

설정 구성(AWS CLI)

에서 계정을 AWS CLI등록하여 설정을 구성합니다.

1. 설정을 구성하려면, 다음 명령을 실행합니다.

```
aws iotfleetwise register-account
```

2. 설정을 확인하려면 다음 명령을 실행하여 등록 상태를 검색합니다.

> **Note**
> 서비스 연결 역할은 에 AWS IoT FleetWise 지표를 게시하는 데만 사용됩니다 CloudWatch. 자세한 내용은 AWS IoT FleetWise에 서비스 연결 역할 사용 단원을 참조하십시오.

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

Example 응답

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

등록 상태는 다음 중 하나일 수 있습니다.

- REGISTRATION_SUCCESS – AWS 리소스가 성공적으로 등록되었습니다.
- REGISTRATION_PENDING – AWS IoT FleetWise 가 등록 요청을 처리하고 있습니다. 이 프로세스를 완료하는 데 5분가량 소요됩니다.
- REGISTRATION_FAILURE – AWS IoT FleetWise 는 AWS 리소스를 등록할 수 없습니다. 나중에 다시 시도해 주십시오.

자습서: AWS IoT FleetWise 시작하기

AWS IoT FleetWise를 사용하면 차량 데이터를 수집, 변환 및 전송할 수 있습니다. 이 섹션의 자습서를 사용하여 AWS IoT FleetWise를 시작합니다.

AWS IoT FleetWise 에 대해 자세히 알아보려면 다음 주제를 참조하세요.

- 클라우드에 AWS IoT FleetWise 데이터 삽입
- Model AWS IoT FleetWise 차량
- AWS IoT FleetWise 차량 관리
- AWS IoT FleetWise에서 플릿 관리
- 캠페인으로 AWS IoT FleetWise 데이터 수집

소개

AWS IoT FleetWise 를 사용하여 자동화된 차량에서 클라우드로 거의 실시간으로 고유한 데이터 형식을 수집, 변환 및 전송할 수 있습니다. 플릿 전반에 걸친 인사이트에 액세스할 수 있습니다. 이를 통해 차량 상태 문제를 효율적으로 감지하고 해결하며, 고부가가치 데이터 신호를 전송하고 원격으로 문제를 진단할 수 있으며, 동시에 이 모든 과정에서 비용을 절감할 수 있습니다.

이 자습서에서는 AWS IoT FleetWise 시작하는 방법을 보여줍니다. 차량 모델(모델 매니페스트), 디코더 매니페스트, 차량 및 캠페인을 만드는 방법을 배우게 됩니다.

AWS IoT FleetWise의 주요 구성 요소 및 개념에 대한 자세한 내용은 섹션을 참조하세요AWS IoT FleetWise의 주요 개념 및 기능.

예상 소요 시간: 45분.

> **Important**
>
> 이 데모에서 생성하고 사용하는 AWS IoT FleetWise 리소스에 대한 요금이 부과됩니다. 자세한 내용은 AWS IoT FleetWise 요금 페이지의AWS IoT FleetWise 를 참조하세요.

사전 조건

이 시작하기 튜토리얼을 완료하려면 다음이 필요합니다.

- AWS 계정. 가 없는 경우 AWS Account Management 참조 가이드의 생성 AWS 계정 섹션을 AWS 계정참조하세요.
- IoT를 AWS 리전 지원하는 에 대한 액세스. AWS IoT FleetWise 현재 AWS IoT FleetWise 는 미국 동부(버지니아 북부)와 유럽(프랑크푸르트)에서 지원됩니다. 의 리전 선택기 AWS Management Console 를 사용하여 이러한 리전 중 하나로 전환할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 엔드포인트 및 할당량 섹션을 참조하세요.
- Amazon Timestream 리소스:
  - Amazon Timestream 데이터베이스. 자세한 내용은 Amazon Timestream 개발자 안내서의 데이터베이스 생성을 참조하세요.
  - Amazon Timestream에서 생성한 Amazon Timestream 테이블로, 데이터를 보관합니다. 자세한 내용은 Amazon Timestream 개발자 가이드의 테이블 생성을 참조하세요.
- Edge Agent 소프트웨어 데모. (데모 설정 지침은 다음 단계에 있습니다.)
  - Explore Edge Agent 빠른 시작 데모를 사용하여 AWS IoT FleetWise 를 탐색하고 IoT용 Edge Agent 소프트웨어를 개발하는 방법을 배울 수 AWS IoT FleetWise. 이 데모에서는 AWS CloudFormation 템플릿을 사용합니다. Edge Agent 참조 구현을 검토하고, Edge Agent를 개발한 다음 Amazon EC2 Graviton에 Edge Agent 소프트웨어를 배포하고, 샘플 차량 데이터를 생성하는 방법을 안내합니다. 또한 데모는 클라우드에서 신호 카탈로그, 차량 모델, 디코더 매니페스트, 차량, 플릿 및 캠페인 생성에 사용할 수 있는 스크립트를 제공합니다.
  - 데모를 다운로드하려면 AWS IoT FleetWise 콘솔 로 이동합니다. 서비스 홈 페이지의 AWS IoT FleetWise 시작하기 섹션에서 엣지 에이전트 탐색을 선택합니다.

1단계: AWS IoT FleetWise용 Edge Agent 소프트웨어 설정

> **Note**
>
> 이 단계의 CloudFormation 스택은 원격 측정 데이터를 사용합니다. 비전 시스템 데이터를 사용하여 CloudFormation 스택을 생성할 수도 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요.
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

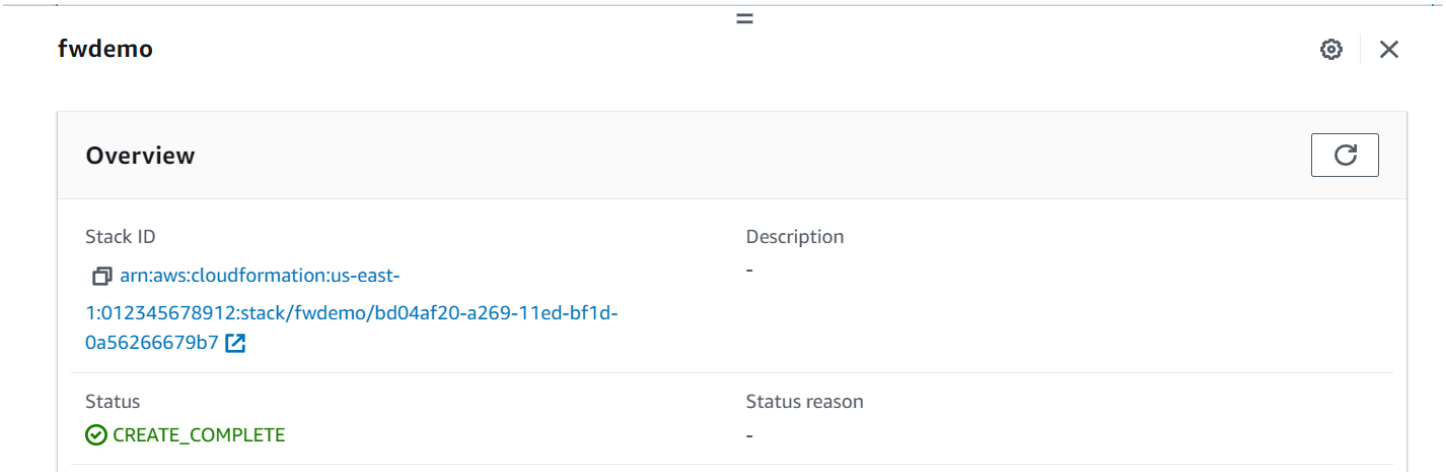
AWS IoT FleetWise 용 Edge Agent 소프트웨어는 차량과 클라우드 간의 통신을 용이하게 합니다. 데이터 수집 체계로부터 클라우드 연결 차량에서 데이터를 수집하는 방법에 대한 지침을 받습니다.

Edge Agent 소프트웨어를 설정하려면 일반 정보에서 다음을 수행하세요.

1. 시작 CloudFormation 템플릿 을 엽니다.
2. 스택 빠른 생성 페이지의 스택 이름 에 AWS IoT FleetWise 리소스 스택의 이름을 입력합니다. 스택은 이 AWS CloudFormation 템플릿이 생성하는 리소스의 이름에 접두사로 표시되는 친숙한 이름입니다.
3. 매개 변수에서 스택과 관련된 매개 변수의 사용자 지정 값을 입력합니다.

   a. Fleetsize - Fleetsize 파라미터를 업데이트하여 플릿의 차량 수를 늘릴 수 있습니다.

   b. I oTCore리전 - I 리전 파라미터를 업데이트하여 AWS IoT 사물이 생성되는 oTCore리전을 지정할 수 있습니다. AWS IoT FleetWise 차량을 생성하는 데 사용한 것과 동일한 리전을 사용해야 합니다. 에 대한 자세한 내용은 리전 및 영역 - Amazon Elastic Compute Cloud 를 AWS 리전참조하세요. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#using-regions-availability-zones-setup
4. 기능 섹션에서 가 IAM 리소스를 AWS CloudFormation 생성함을 확인하는 상자를 선택합니다.
5. 스택 생성을 선택한 다음 스택 상태가 CREATE_로 표시될 때까지 약 15분 정도 기다립니다COMPLETE.
6. 스택이 생성되었는지 확인하려면 스택 정보 탭을 선택하고 보기를 새로 고친 다음 CREATE_를 찾습니다COMPLETE.



Important

이 데모에서 생성하고 사용하는 AWS IoT FleetWise 리소스에 대한 요금이 부과됩니다. 자세한 내용은 AWS IoT FleetWise 요금 페이지의AWS IoT FleetWise 를 참조하세요.

2단계: 차량 모델 생성

> **Important**
>
> AWS IoT FleetWise 콘솔에서 비전 시스템 데이터 신호로 차량 모델을 생성할 수 없습니다. 그 대신 AWS CLI을 사용합니다.

차량 모델을 사용하여 차량의 형식을 표준화할 수 있으며 생성되는 차량 신호 간의 관계를 정의할 수 있습니다. 신호 카탈로그는 차량 모델을 생성할 때도 생성됩니다. 차량 모델 생성에 재사용할 수 있는 표준화된 신호 컬렉션을 생성합니다. 신호는 차량 데이터와 해당 메타데이터를 포함하도록 정의하는 기본 구조입니다. 현재 AWS IoT FleetWise 서비스는 계정 AWS 리전 당 하나의 신호 카탈로그만 지원합니다. 이를 통해 차량 플릿에서 처리된 데이터를 일관되게 유지할 수 있습니다.

차량를 생성하는 방법

1. AWS IoT FleetWise 콘솔을 엽니다.
2. 기본 탐색 창에서 차량 모델을 선택합니다.
3. 모델 페이지에서 모델 생성을 선택합니다.
4. 일반 정보 섹션에서 차량 모델 이름(예: Vehicle1)과 선택적 설명을 입력합니다. 다음을 선택합니다.
5. 신호 카탈로그에서 하나 이상의 신호를 선택합니다. 검색 카탈로그에서 이름을 기준으로 신호를 필터링하거나 목록에서 신호를 선택할 수 있습니다. 예를 들어, 타이어 압력과 브레이크 압력에 대한 신호를 선택하여 이러한 신호와 관련된 데이터를 수집할 수 있습니다. 다음을 선택합니다.
6. .dbc 파일을 선택하고 로컬 장치에서 업로드하세요. 다음을 선택합니다.

   > **Note**
   >
   > 이 자습서에서는 이 단계에서 업로드할 샘플.dbc 파일을 다운로드할 수 있습니다.

7. 차량 모델에 속성을 추가한 후 다음을 선택합니다.

   a. 이름 - 제조업체 이름 또는 제조 날짜와 같은 차량 속성의 이름을 입력합니다.

   b. 데이터 유형 - 데이터 유형 메뉴에서 데이터 유형을 선택합니다.

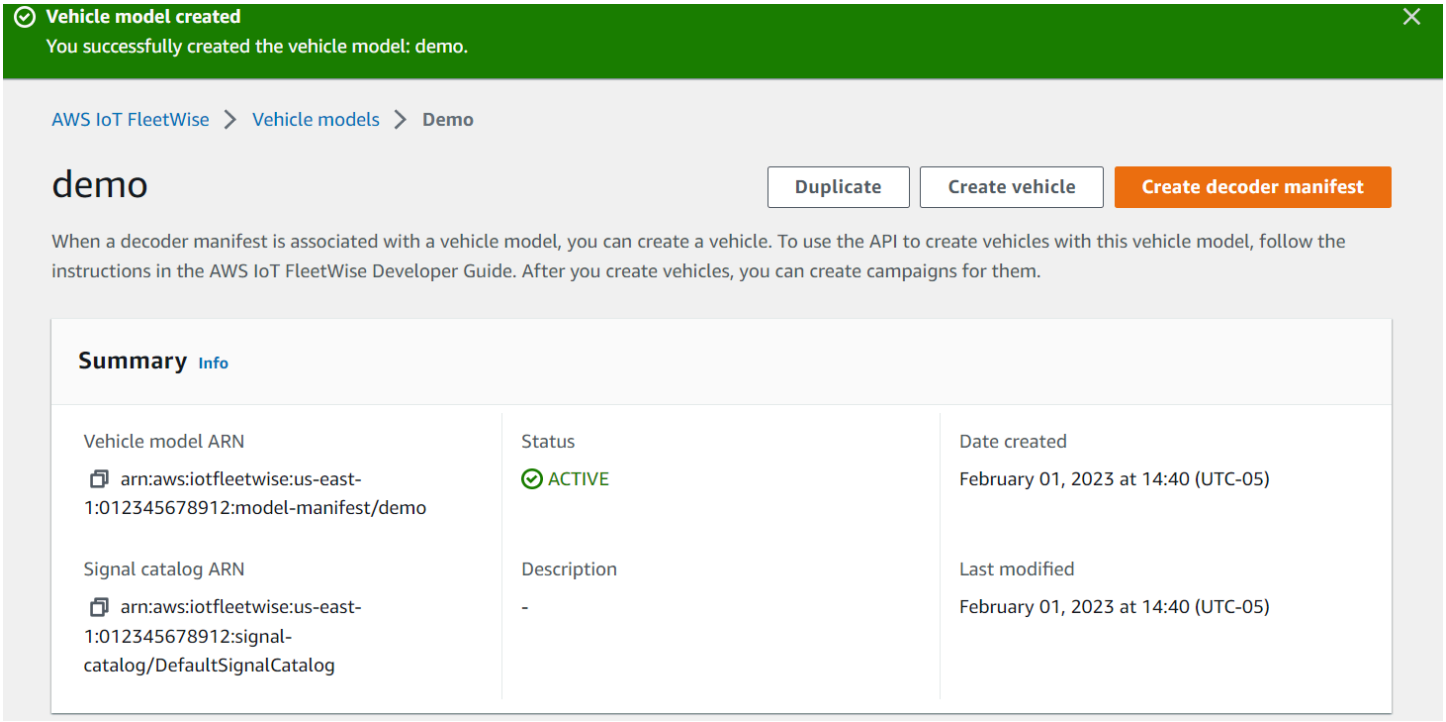
   c. 단위 - (선택 사항) 킬로미터 또는 섭씨와 같은 단위 값을 입력합니다.

d. 경로 - (선택 사항) 신호 경로 이름(예: `Vehicle.Engine.Light.`)을 입력합니다. 점(.)은 신호가 하위 신호임을 나타냅니다.

e. 기본값 - (선택 사항) 기본값을 입력합니다.

f. 설명 - (선택 사항) 속성에 대한 설명을 입력합니다.

8. 구성을 검토합니다. 준비가 되었으면 생성을 선택합니다. 차량 모델이 성공적으로 생성되었다는 알림이 나타납니다.



3단계: 디코더 매니페스트 생성

디코더 매니페스트는 생성한 차량 모델과 연결됩니다. 여기에는 AWS IoT FleetWise 가 바이너리 형식에서 차량 데이터를 디코딩하고 분석할 수 있는 사람이 읽을 수 있는 값으로 변환하는 데 도움이 되는 정보가 포함되어 있습니다. 네트워크 인터페이스와 디코더 신호는 디코더 매니페스트를 구성하는 데 도움이 되는 구성 요소입니다. 네트워크 인터페이스에는 차량 네트워크에서 사용하는 CAN 또는 OBD 프로토콜에 대한 정보가 포함되어 있습니다. 디코더 신호는 특정 신호에 대한 디코딩 정보를 제공합니다.

디코더 매니페스트를 생성하려는 경우

1. AWS IoT FleetWise 콘솔을 엽니다.
2. 기본 탐색 창에서 차량 모델을 선택합니다.

3. 차량 모델 섹션에서 디코더 매니페스트 생성에 사용할 차량 모델을 선택합니다.
4. 디코더 매니페스트 생성을 선택합니다.

4단계: 디코더 매니페스트 구성

디코더 매니페스트를 구성하려는 경우

> **Important**
>
> AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트에서 비전 시스템 데이터 신호를 구성할 수 없습니다. 그 대신 AWS CLI을 사용합니다. 자세한 내용은 디코더 매니페스트 만들기 (AWS CLI) 단원을 참조하십시오.

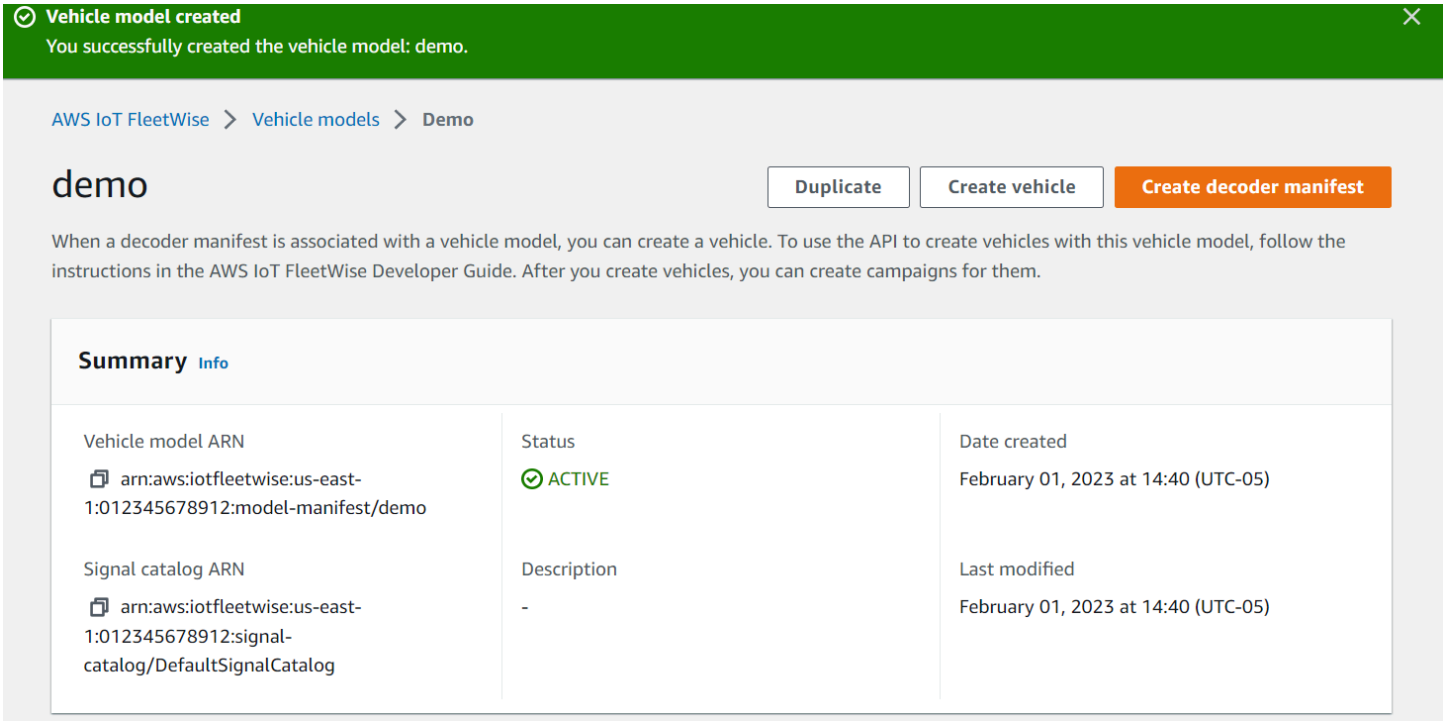
1. 디코더 매니페스트를 식별하는 데 도움이 되도록 이름과 설명(선택 사항)을 입력합니다. 그리고 다음을 선택합니다.
2. 네트워크 인터페이스를 하나 이상 추가하려면 CAN_INTERFACE 또는 OBD_INTERFACE 유형을 선택합니다.
   - 온보드 진단(OBD) 인터페이스 - 자체 진단 데이터가 전자 제어 장치() 간에 통신되는 방식을 정의하는 프로토콜을 원하는 경우 이 인터페이스 유형을 선택합니다ECUs. 이 프로토콜은 차량 문제를 해결하는 데 도움이 되는 여러 표준 진단 문제 코드(DTCs)를 제공합니다.
   - 컨트롤러 영역 네트워크(CAN 버스) 인터페이스 - 간에 데이터가 통신되는 방식을 정의하는 프로토콜을 원하는 경우 이 인터페이스 유형을 선택합니다ECUs. ECUs 는 엔진 컨트롤 유닛, 에어백 또는 오디오 시스템일 수 있습니다.
3. 네트워크 인터페이스 이름을 입력합니다.
4. 네트워크 인터페이스에 신호를 추가하려면 목록에서 하나 이상의 신호를 선택합니다.
5. 이전 단계에서 추가한 신호에 사용할 디코더 신호를 선택합니다. 디코딩 정보를 제공하려면.dbc 파일을 업로드하세요. 차량 모델의 각 신호는 목록에서 선택할 수 있는 디코더 신호와 페어링되어야 합니다.
6. 보조 네트워크 인터페이스를 추가하려면 네트워크 인터페이스 추가를 선택합니다. 네트워크 인터페이스를 모두 추가했으면 다음을 선택합니다.
7. 구성을 살펴본 후 생성을 선택합니다. 디코더 매니페스트가 성공적으로 생성되었다는 알림이 나타납니다.

5단계: 차량 생성

AWS IoT FleetWise에서 차량은 실제 물리적 차량의 가상 표현입니다. 동일한 차량 모델에서 생성된 모든 차량은 동일한 신호 그룹을 상속하며, 생성한 각 차량은 새로 생성된 IoT 사물에 해당합니다. 모든 차량을 디코더 매니페스트에 연결해야 합니다.

사전 조건

1. 차량 모델 및 디코더 매니페스트를 이미 생성했는지 확인하세요. 또한 차량 모델의 상태가 인지 확인합니다ACTIVE.
   a. 차량 모델의 상태가 인지 확인하려면 AWS IoT FleetWise 콘솔을 ACTIVE엽니다.
   b. 기본 탐색 창에서 차량 모델을 선택합니다.
   c. 요약 섹션의 상태에서 차량 상태를 확인합니다.



차량을 생성하려는 경우

1. AWS FleetWise 콘솔을 엽니다.
2. 탐색 창에서 차량을 선택합니다.
3. 차량 생성을 선택합니다.

4. 차량 속성을 정의하려면 차량 이름을 입력한 다음 모델 매니페스트(차량 모델)와 디코더 매니페스트를 선택합니다.
5. (선택 사항) 차량 속성을 정의하려면 키-값 쌍을 입력한 다음 속성 추가를 선택합니다.
6. (선택 사항) AWS 리소스에 레이블을 지정하려면 태그를 추가한 다음 새 태그 추가를 선택합니다.
7. Next(다음)를 선택합니다.
8. 차량 인증서를 구성하려면 자체 인증서를 업로드하거나 신규 인증서 자동 생성을 선택할 수 있습니다. 더 빠르게 설정하려면 인증서를 자동 생성하는 것이 좋습니다. 이미 인증서가 있는 경우 해당 인증서를 대신 사용하도록 선택할 수 있습니다.
9. 공개 및 개인 키 파일을 다운로드한 후 다음을 선택합니다.
10. 정책을 차량 인증서에 첨부하려면 기존 정책 이름을 입력하거나 새 정책을 생성할 수 있습니다. 새 정책을 생성하려면 정책 생성을 선택한 후 다음을 선택합니다.
11. 구성을 검토합니다. 완료했으면 차량 생성을 선택합니다.

6단계: 캠페인 생성

AWS IoT FleetWise 에서 캠페인은 차량에서 클라우드로 데이터를 선택, 수집 및 전송하는 데 사용됩니다. 캠페인에는 데이터 수집 체계가 포함되어 있으며 이는 Edge Agent 소프트웨어에 조건 기반 수집 체계 또는 시간 기반 수집 체계로 데이터를 수집하는 방법에 대한 지침을 제공합니다.

캠페인을 생성하려는 경우

1. AWS IoT FleetWise 콘솔을 엽니다.
2. 탐색 창에서 캠페인을 선택합니다.
3. 캠페인 생성을 선택합니다.
4. 캠페인에 대한 이름과 선택 사항 설명을 입력합니다.
5. 캠페인의 데이터 수집 체계를 구성하려면 데이터 수집 체계를 수동으로 정의하거나 로컬 디바이스에서 .json 파일을 업로드합니다. .json 파일을 업로드하면 데이터 수집 체계가 자동으로 정의됩니다.

    a. 데이터 수집 체계를 수동으로 정의하려면 데이터 수집 체계 정의를 선택하고 캠페인에 사용할 데이터 수집 체계 유형을 선택합니다. 조건 기반 수집 체계 또는 시간 기반 수집 체계를 선택할 수 있습니다.

    b. 시간 기반 수집 체계를 선택하는 경우 캠페인에서 차량 데이터를 수집하는 기간을 지정해야 합니다.

c. 조건 기반 수집 체계를 선택하는 경우 수집할 데이터를 인식하기 위한 표현식을 지정해야 합니다. 신호 이름을 변수, 비교 연산자 및 비교 값으로 지정해야 합니다.

d. (선택 사항) 표현식의 언어 버전을 선택하거나 기본값인 1로 유지합니다.

e. (선택 사항) 두 데이터 수집 이벤트 간의 트리거 간격을 지정합니다.

f. 데이터를 수집하려면 Edge Agent 소프트웨어의 트리거 모드 조건을 선택합니다. 기본적으로 Edge Agent for AWS IoT FleetWise 소프트웨어는 조건이 충족될 때마다 항상 데이터를 수집합니다. 또는 조건이 처음으로 충족되는 경우, 즉 첫 번째 트리거 시에만 데이터를 수집할 수 있습니다.

g. (선택 사항) 고급 체계 옵션을 더 선택할 수 있습니다.

6. 데이터 수집 체계에서 데이터를 수집할 신호를 지정하려면 메뉴에서 신호 이름을 검색합니다.
7. (선택 사항) 최대 샘플 수 또는 최소 샘플링 간격을 선택할 수 있습니다. 또한 더 많은 신호를 추가할 수 있습니다.
8. Next(다음)를 선택합니다.
9. 캠페인에서 데이터를 전송할 저장 대상을 정의합니다. 데이터를 Amazon S3 또는 Amazon Timestream에 저장할 수 있습니다.

   a. Amazon S3 - AWS IoT FleetWise 권한이 있는 S3 버킷을 선택합니다.

   b. Amazon Timestream — Timestream 데이터베이스 및 테이블 이름을 선택합니다. 가 Timestream AWS IoT FleetWise 으로 데이터를 전송할 수 있도록 허용하는 IAM 역할을 입력합니다.

10. Next(다음)를 선택합니다.
11. 검색 상자에서 차량 속성 또는 차량 이름을 선택합니다.
12. 차량에 대해 선택한 속성 또는 이름과 관련된 값을 입력합니다.
13. 캠페인에서 데이터를 수집할 차량을 선택합니다. 그리고 다음을 선택합니다.
14. 캠페인 구성을 검토한 다음 캠페인 생성을 선택합니다. 사용자 또는 팀이 차량에 캠페인을 배포해야 합니다.

7단계: 정리

이 자습서에서 사용한 리소스에 대한 추가 요금을 피하려면 AWS CloudFormation 스택과 모든 스택 리소스를 삭제하세요.

AWS CloudFormation 스택을 삭제하려면

1. AWS CloudFormation 콘솔을 엽니다.
2. 스택 목록에서 1단계에서 생성한 스택을 선택합니다.
3. Delete(삭제)를 선택합니다.
4. 삭제를 확인하려면 삭제를 선택합니다. 스택을 삭제하는 데 약 15분이 걸립니다.

다음 단계

1. 캠페인에서 수집하는 차량 데이터를 처리하고 시각화할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 차량 데이터 시각화 단원을 참조하십시오.
2. AWS IoT FleetWise 관련 문제를 해결하고 해결할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 문제 해결 단원을 참조하십시오.

클라우드에 AWS IoT FleetWise 데이터 삽입

Edge Agent for AWS IoT FleetWise 소프트웨어는 차량에 설치 및 실행할 때 차량과 클라우드 간의 안전한 통신을 용이하게 하도록 설계되었습니다.

> **Note**
>
> - AWS IoT FleetWise 는 심각한 신체 부상 또는 사망을 초래하거나 환경 또는 재산 피해를 초래할 수 있는 위험한 환경 또는 중요한 시스템의 작동에 사용하거나 이와 관련하여 사용해서는 안 됩니다. AWS IoT FleetWise 를 사용하여 수집된 차량 데이터는 정보 제공 목적으로만 사용되며, AWS IoT FleetWise 를 사용하여 차량 기능을 제어하거나 운영할 수 없습니다.
> - AWS IoT FleetWise 사용을 통해 수집된 차량 데이터는 해당 차량 안전 규정(예: 안전 모니터링 및 보고 의무)에 따라 가질 수 있는 규정 준수 의무를 충족하기 위한 목적을 포함하여 사용 사례에 적합한 정확도를 평가해야 합니다. 이러한 평가에는 다른 업계 표준 수단 및 출처(예: 차량 운전자 보고서)를 통한 정보 수집 및 검토가 포함되어야 합니다.

클라우드로 데이터를 인제스트하려면 다음을 수행합니다.

1. 차량에 Edge Agent for AWS IoT FleetWise 소프트웨어를 개발하고 설치합니다. Edge Agent 소프트웨어 작업 방법에 대한 자세한 내용은 다음을 수행하여 Edge Agent for AWS IoT FleetWise 소프트웨어 개발자 안내서를 다운로드하세요.
   1. AWS IoT FleetWise 콘솔 로 이동합니다.
   2. 서비스 홈 페이지의 AWS IoT FleetWise 시작하기 섹션에서 엣지 에이전트 탐색을 선택합니다.
2. 차량 모델을 만드는 데 사용할 신호가 포함된 신호 카탈로그를 생성하거나 가져옵니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 생성 및 신호 카탈로그 가져오기(AWS CLI) 단원을 참조하세요.

   > **Note**
   >
   > - AWS IoT FleetWise 콘솔을 사용하여 첫 번째 차량 모델을 생성하는 경우 신호 카탈로그를 수동으로 생성할 필요가 없습니다. 첫 번째 차량 모델을 생성하면 AWS IoT FleetWise 가 자동으로 신호 카탈로그를 생성합니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.

- AWS IoT FleetWise 는 현재 에 따라 각 AWS 계정에 대한 신호 카탈로그를 지원합니다 AWS 리전.

3. 신호 카탈로그의 신호를 사용하여 차량 모델을 생성하세요. 자세한 내용은 AWS IoT FleetWise 차량 모델 생성 단원을 참조하십시오.

> **Note**
>
> - AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 .dbc 파일을 업로드하여 신호를 가져올 수 있습니다. .dbc는 컨트롤러 영역 네트워크(CAN 버스) 데이터베이스가 지원하는 파일 형식입니다. 차량 모델이 생성되면 새 신호가 신호 카탈로그에 자동으로 추가됩니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.
> - `CreateModelManifest` API 작업을 사용하여 차량 모델을 생성하는 경우 `UpdateModelManifest` API 작업을 사용하여 차량 모델을 활성화해야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 업데이트 단원을 참조하십시오.
> - AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 AWS IoT FleetWise는 자동으로 차량 모델을 활성화합니다.

4. 디코더 매니페스트를 생성합니다. 디코더 매니페스트에는 차량 모델(이전 단계에서 생성)에 명시된 모든 신호의 디코딩 정보가 포함되어 있습니다. 디코더 매니페스트는 생성한 차량 모델과 연결됩니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

> **Note**
>
> - `CreateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 생성하는 경우 `UpdateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 활성화해야 합니다. 자세한 내용은 AWS IoT FleetWise 디코더 매니페스트 업데이트 단원을 참조하십시오.
> - AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트를 생성하는 경우 AWS IoT FleetWise 는 자동으로 디코더 매니페스트를 활성화합니다.

5. 차량 모델에서 차량을 만듭니다. 동일한 차량 모델에서 생성된 차량은 동일한 신호 그룹을 상속받습니다. 클라우드 AWS IoT Core 에 데이터를 수집하려면 먼저 를 사용하여 차량을 프로비저닝해야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 관리 단원을 참조하십시오.

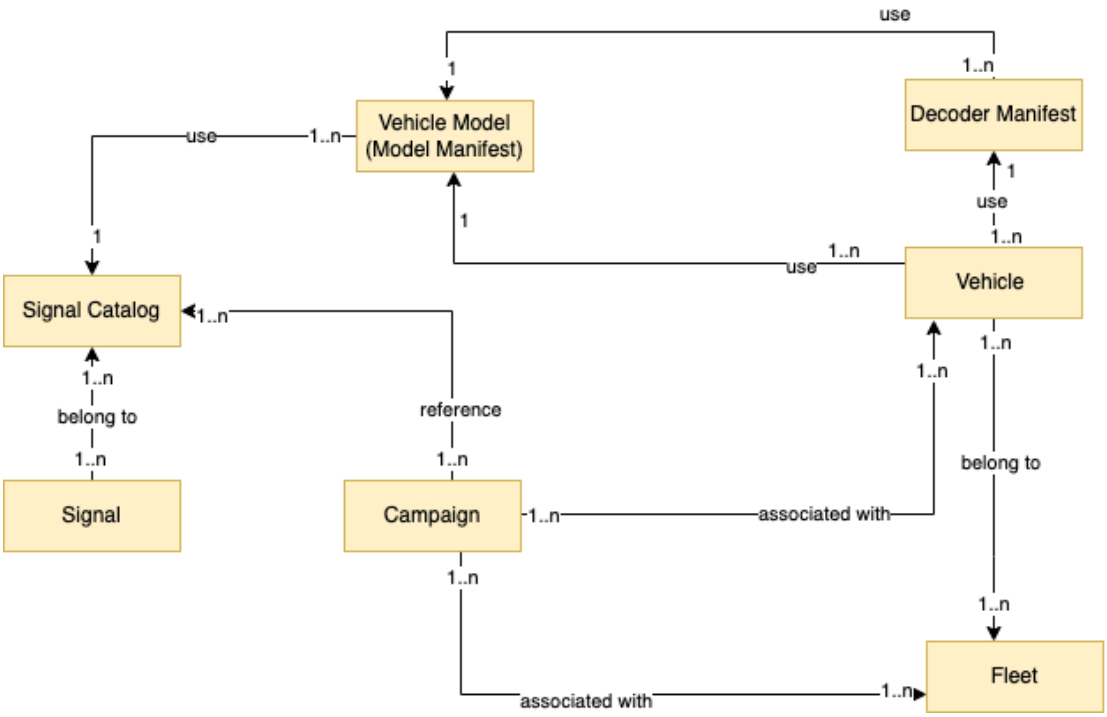
6. (선택 사항) 차량 그룹을 나타내는 플릿을 생성한 다음 개별 차량을 플릿과 연결합니다. 이를 통해 동시에 여러 차량을 관리할 수 있습니다. 자세한 내용은 AWS IoT FleetWise에서 플릿 관리 단원을 참조하십시오.
7. 캠페인을 생성합니다. 캠페인은 차량 또는 여러 차량의 플릿에 배포됩니다. 캠페인은 Edge Agent 소프트웨어에서 데이터를 선택하고 수집하고 클라우드로 전송하는 방법에 대한 지침을 제공합니다. 자세한 내용은 캠페인으로 AWS IoT FleetWise 데이터 수집 단원을 참조하십시오.

> **Note**
>
> AWS IoT FleetWise 가 캠페인을 차량 또는 플릿에 배포하려면 먼저 `UpdateCampaign` API 작업을 사용하여 캠페인을 승인해야 합니다. 자세한 내용은 AWS IoT FleetWise 캠페인 업데이트 단원을 참조하십시오.

Edge Agent 소프트웨어는 예약된 주제 를 AWS IoT Core 사용하여 차량 데이터를 로 전송`$aws/iotfleetwise/vehicles/`*`vehicleName`*`/signals`하여 AWS IoT FleetWise. AWS IoT FleetWise 로 데이터를 전송한 다음 Timestream 테이블 또는 Amazon S3 버킷으로 데이터를 전송합니다. Timestream을 사용하여 데이터를 쿼리하고 Amazon QuickSight 또는 Grafana를 사용하여 데이터를 시각화할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 차량 데이터 시각화 단원을 참조하십시오.

Model AWS IoT FleetWise 차량

AWS IoT FleetWise 는 클라우드에서 차량의 가상 표현을 구축하는 데 사용할 수 있는 차량 모델링 프레임워크를 제공합니다. 신호, 신호 카탈로그, 차량 모델 및 디코더 매니페스트는 차량 모델링에 사용하는 핵심 구성 요소입니다.



신호

신호는 차량 데이터와 해당 메타데이터를 포함하도록 정의하는 기본 구조입니다. 신호는 속성, 분기, 센서 또는 액추에이터일 수 있습니다. 예를 들어, 차량 내 온도 값을 수신하고 센서 이름, 데이터 유형 및 단위를 포함한 메타데이터를 저장하는 센서를 생성할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리 단원을 참조하십시오.

신호 카탈로그

신호 카탈로그에는 신호 컬렉션이 포함되어 있습니다. 신호 카탈로그에 있는 신호는 다양한 프로토콜과 데이터 형식을 사용하는 차량을 모델링할 때 사용할 수 있습니다. 예를 들어 자동차 제조업체마다 두 개의 자동차가 만들어집니다. 하나는 제어 영역 네트워크(CAN 버스) 프로토콜을 사용하고 다른 하나는 온보드 진단(OBD) 프로토콜을 사용합니다. 신호 카탈로그에서 센서를 정의하여 차량 내 온도 값을 수신할 수 있습니다. 이 센서는 두 차량의 열전대를 나타내는 데 사용할 수 있습니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리 단원을 참조하십시오.

차량 모델(모델 매니페스트)

차량 모델은 차량 형식을 표준화하고 차량 내 신호 간의 관계 정의에 사용할 수 있는 선언적 구조입니다. 차량 모델은 동일한 유형의 다중 차량에 일관된 정보를 적용합니다. 신호를 추가하여 차량 모델을 생성합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 관리 단원을 참조하십시오.

디코더 매니페스트

디코더 매니페스트에는 차량 모델의 각 신호에 대한 디코딩 정보가 포함되어 있습니다. 차량의 센서와 액추에이터는 저수준 메시지(바이너리 데이터)를 전송합니다. 디코더 매니페스트를 사용하면 AWS IoT FleetWise 가 바이너리 데이터를 사람이 읽을 수 있는 값으로 변환할 수 있습니다. 모든 디코더 매니페스트는 차량 모델과 연결됩니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

AWS IoT FleetWise 콘솔 또는 를 사용하여 다음과 같은 방법으로 차량을 API 모델링할 수 있습니다.

1. 차량 모델을 만드는 데 사용할 신호가 포함된 신호 카탈로그를 생성하거나 가져옵니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 생성 및 신호 카탈로그 가져오기(AWS CLI) 단원을 참조하세요.

   > **Note**
   > - AWS IoT FleetWise 콘솔을 사용하여 첫 번째 차량 모델을 생성하는 경우 신호 카탈로그를 수동으로 생성할 필요가 없습니다. 첫 번째 차량 모델을 생성하면 AWS IoT FleetWise 가 자동으로 신호 카탈로그를 생성합니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.
   > - AWS IoT FleetWise 는 현재 에 따라 각 AWS 계정의 신호 카탈로그를 지원합니다 AWS 리전.

2. 신호 카탈로그의 신호를 사용하여 차량 모델을 생성하세요. 자세한 내용은 AWS IoT FleetWise 차량 모델 생성 단원을 참조하십시오.

   > **Note**
   > - AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 .dbc 파일을 업로드하여 신호를 가져올 수 있습니다. .dbc는 컨트롤러 영역 네트워크(CAN 버스) 데이터베이스가 지원하는 파일 형식입니다. 차량 모델이 생성되면 새 신호가 신호 카탈로그에 자동으로 추가됩니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.

- `CreateModelManifest` API 작업을 사용하여 차량 모델을 생성하는 경우 `UpdateModelManifest` API 작업을 사용하여 차량 모델을 활성화해야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 업데이트 단원을 참조하십시오.
- AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 AWS IoT FleetWise는 자동으로 차량 모델을 활성화합니다.

3. 디코더 매니페스트를 생성합니다. 디코더 매니페스트에는 차량 모델(이전 단계에서 생성)에 명시된 모든 신호의 디코딩 정보가 포함되어 있습니다. 디코더 매니페스트는 생성한 차량 모델과 연결됩니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

> **Note**
>
> - `CreateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 생성하는 경우 `UpdateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 활성화해야 합니다. 자세한 내용은 AWS IoT FleetWise 디코더 매니페스트 업데이트 단원을 참조하십시오.
> - AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트를 생성하는 경우 AWS IoT FleetWise 는 자동으로 디코더 매니페스트를 활성화합니다.

CAN 버스 데이터베이스는 .dbc 파일 형식을 지원합니다. .dbc 파일을 업로드하여 신호 및 디코더 신호를 가져올 수 있습니다. .dbc 파일 예제를 가져오려면 다음을 수행합니다.

.dbc 파일을 가져오려는 경우

1. EngineSignals.zip 을 다운로드합니다.
2. `EngineSignals.zip` 파일을 다운로드한 디렉터리로 이동합니다.
3. 콘텐츠의 압축을 풀고 `EngineSignals.dbc`로 로컬로 저장합니다.

주제

- AWS IoT FleetWise 신호 카탈로그 관리
- AWS IoT FleetWise 차량 모델 관리
- Manage AWS IoT FleetWise 디코더 매니페스트

AWS IoT FleetWise 신호 카탈로그 관리

> **Note**
>
> 데모 스크립트를 다운로드하여 ROS 2개의 메시지를 신호 카탈로그와 호환되는 VSS JSON 파일로 변환할 수 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요.

신호 카탈로그는 차량 모델을 생성하는 데 재사용할 수 있는 표준화된 신호 모음입니다. AWS IoT FleetWise 는 신호를 정의하는 데 따를 수 있는 차량 신호 사양(VSS)을 지원합니다. 신호는 다음 유형 중 하나일 수 있습니다.

속성

속성은 일반적으로 변경되지 않는 정적 정보(예: 제조업체 및 제조일)를 나타냅니다.

브랜치

브랜치는 신호를 중첩된 구조로 나타냅니다. 브랜치는 신호 계층 구조를 보여줍니다. 예를 들어, `Vehicle` 브랜치에는 하위 브랜치, `Powertrain`이 있습니다. `Powertrain` 브랜치에는 하위 브랜치, `combustionEngine`이 있습니다. `combustionEngine` 브랜치를 찾으려면 `Vehicle.Powertrain.combustionEngine` 표현식을 사용하세요.

센서

센서 데이터는 차량의 현재 상태와 시간에 따른 변화(예: 유체 수준, 온도, 진동, 전압 등)를 보고합니다.

액추에이터

액추에이터 데이터는 모터, 히터, 도어록과 같은 차량 장치의 상태를 보고합니다. 차량 장치의 상태를 변경하면 액추에이터 데이터를 업데이트할 수 있습니다. 예를 들어, 히터를 나타내는 액추에이터를 정의할 수 있습니다. 히터를 켜거나 끌 때 액추에이터가 새 데이터를 수신합니다.

사용자 지정 구조

사용자 지정 구조(구조체라고도 함)는 복합 또는 고차 데이터 구조를 나타냅니다. 이를 통해 동일한 소스에서 생성된 데이터를 쉽게 논리적으로 바인딩하거나 그룹화할 수 있습니다. 구조체는 복합 데이터 유형 또는 고차 모양을 나타내는 것과 같이 원자성 연산으로 데이터를 읽거나 쓸 때 사용됩니다.

구조체 유형의 신호는 프리미티브 데이터 유형 대신 구조체 데이터 유형에 대한 참조를 사용하여 신호 카탈로그에 정의됩니다. 구조체는 센서, 속성, 액추에이터, 비전 시스템 데이터 유형

을 비롯한 모든 유형의 신호에 사용할 수 있습니다. 구조 유형의 신호가 전송되거나 수신되면 AWS IoT FleetWise 는 포함된 모든 항목에 유효한 값이 있을 것으로 예상하므로 모든 항목은 필수입니다. 예를 들어 구조체에 Vehicle.Camera.Image.height, Vehicle.Camera.Image.width 및 Vehicle.Camera.Image.data 항목이 포함된 경우 송신된 신호에 이러한 모든 항목의 값이 포함될 것으로 예상됩니다.

> **Note**
>
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

사용자 지정 속성

사용자 지정 속성은 복합 데이터 구조의 멤버를 나타냅니다. 속성의 데이터 유형은 프리미티브이거나 다른 구조체일 수 있습니다.

구조체와 사용자 지정 속성을 사용하여 고차 모양을 표현할 때는 의도한 고차 모양이 항상 트리 구조로 정의되고 표시됩니다. 사용자 지정 속성은 모든 리프 노드를 정의하는 데 사용되고 구조체는 리프가 아닌 모든 노드를 정의하는 데 사용됩니다.

> **Note**
>
> - AWS IoT FleetWise 콘솔을 사용하여 첫 번째 차량 모델을 생성하는 경우 신호 카탈로그를 수동으로 생성할 필요가 없습니다. 첫 번째 차량 모델을 생성하면 AWS IoT FleetWise 가 자동으로 신호 카탈로그를 생성합니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.
> - AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 .dbc 파일을 업로드하여 신호를 가져올 수 있습니다. .dbc는 컨트롤러 영역 네트워크(CAN 버스) 데이터베이스가 지원하는 파일 형식입니다. 차량 모델이 생성되면 새 신호가 신호 카탈로그에 자동으로 추가됩니다. 자세한 내용은 차량 모델 생성(콘솔) 단원을 참조하십시오.
> - AWS IoT FleetWise 는 현재 리전 AWS 계정 당 각 에 대한 신호 카탈로그를 지원합니다.

AWS IoT FleetWise 는 신호 카탈로그를 생성하고 관리하는 데 사용할 수 있는 다음 API 작업을 제공합니다.

- CreateSignalCatalog - 새 신호 카탈로그를 생성합니다.

- ImportSignalCatalog - JSON 파일을 업로드하여 신호를 가져와 신호 카탈로그를 생성합니다. 신호는 다음과 같이 정의VSS되고 JSON 형식으로 저장되어야 합니다.
- UpdateSignalCatalog - 신호를 업데이트, 제거 또는 추가하여 기존 신호 카탈로그를 업데이트합니다.
- DeleteSignalCatalog - 기존 신호 카탈로그를 삭제합니다.
- ListSignalCatalogs - 모든 신호 카탈로그의 페이지가 매겨진 요약 목록을 검색합니다.
- ListSignalCatalogNodes - 지정된 신호 카탈로그의 모든 신호(노드) 요약에 대한 페이지 매김 목록을 검색합니다.
- GetSignalCatalog - 신호 카탈로그에 대한 정보를 검색합니다.

자습서

- AWS IoT FleetWise 신호 구성
- AWS IoT FleetWise 신호 카탈로그 생성
- AWS IoT FleetWise 신호 카탈로그 가져오기
- AWS IoT FleetWise 신호 카탈로그 업데이트
- AWS IoT FleetWise 신호 카탈로그 삭제
- Get AWS IoT FleetWise 신호 카탈로그 정보

AWS IoT FleetWise 신호 구성

이 섹션에서는 분기, 속성, 센서 및 액추에이터를 구성하는 방법을 보여줍니다.

주제

- 브랜치 구성
- 구성 속성
- 센서 또는 액추에이터를 구성합니다
- 복합 데이터 유형 구성

브랜치 구성

새 연결을 추가하려면 다음 정보를 지정합니다.

- `fullyQualifiedName` – 브랜치의 완전히 정규화된 이름은 브랜치 경로에 브랜치 이름을 더한 것입니다. 자식 브랜치를 가리키려면 점(.)을 사용합니다. 예를 들어

`Vehicle.Chassis.SteeringWheel`은 `SteeringWheel` 브랜치의 완전히 정규화된 이름입니다. `Vehicle.Chassis.`가 브랜치의 경로입니다.

정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, 콜론(:) 및 밑줄(_)

- (선택 사항) `Description` – 브랜치에 대한 설명입니다.

  설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)
- (선택 사항) `deprecationMessage` - 이동 또는 삭제 중인 노드 또는 분기에 대한 지원 중단 메시지입니다.

  는 최대 2,048자를 포함할 deprecationMessage 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)
- (선택 사항) `comment` - 설명 외에 코멘트를 추가합니다. 코멘트는 브랜치에 대한 이론적 근거나 관련 브랜치에 대한 참조 등 브랜치에 대한 추가 정보를 제공하는 데 사용될 수 있습니다.

  코멘트는 최대 2048자까지 쓸 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

구성 속성

속성을 구성하려면 다음 정보를 지정하세요.

- `dataType` - 속성의 데이터 유형은 INT8, , UINT8, INT16, UINT16, INT32UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, _, INT8_ARRAY, INT16_UINT8ARRAY, ARRAYUINT16_ARRAY, INT32_ARRAY, _ARRAY, UINT32_, INT64ARRAYUINT64ARRAYBOOLEAN_ARRAYFLOATARRAY, DOUBLE_, STRING_ARRAY, _ARRAY, _, UNIX_TIMESTAMPARRAY_, , UNKNOWN fullyQualifiedName, 또는 데이터 유형 브랜치에 정의된 사용자 지정 구조 중 하나여야 합니다.
- `fullyQualifiedName` – 속성의 완전히 정규화된 이름은 속성 경로에 속성 이름을 더한 값입니다. 자식 신호를 나타내려면 점(.)을 사용합니다. 예를 들어 `Vehicle.Chassis.SteeringWheel.Diameter`는 `Diameter` 속성의 완전히 정규화된 이름입니다. `Vehicle.Chassis.SteeringWheel.`은 속성의 경로입니다.

  완전히 정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).
- (선택 사항) `Description` — 속성에 대한 설명입니다.

  설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) `unit` — 속성의 과학 단위 (예: km 또는 섭씨).
- (선택 사항) `min` — 속성의 최소값입니다.
- (선택 사항) `max` — 속성의 최대값입니다.
- (선택 사항) `defaultValue` — 속성의 기본값입니다.
- (선택 사항) `assignedValue` — 속성에 할당된 값입니다.
- (선택 사항) `allowedValues` — 속성이 허용하는 값 목록입니다.
- (선택 사항) `deprecationMessage` — 이동 또는 삭제되는 노드 또는 분기에 대한 지원 중단 메시지입니다.

  는 최대 2,048자를 포함할 deprecationMessage 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)
- (선택 사항) `comment` - 설명 외에 코멘트를 추가합니다. 코멘트를 사용하여 속성에 대한 이론적 근거나 관련 속성에 대한 참조 등 속성에 대한 추가 정보를 제공할 수 있습니다.

  코멘트는 최대 2048자까지 쓸 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

센서 또는 액추에이터를 구성합니다

센서 또는 액추에이터를 구성하려면 다음 정보를 지정합니다.

- `dataType` - 신호의 데이터 형식은 INT8, , UINT8, , UINT16, INT16, INT32UINT32INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, _, INT8_ARRAY, INT16_UINT8ARRAY, ARRAYUINT16_ARRAY, INT32_ARRAY, _ARRAY, UINT32_, INT64ARRAYUINT64ARRAYBOOLEAN_ARRAY, ARRAYFLOAT_, STRING_DOUBLEARRAY, _ARRAY, _, UNIX_TIMESTAMPARRAY_ fullyQualifiedName, UNKNOWN, 또는 데이터 형식 브랜치에 정의된 사용자 지정 구조 중 하나여야 합니다.
- `fullyQualifiedName`— 신호의 완전히 정규화된 이름은 신호 경로에 신호 이름을 더한 것입니다. 하위 신호를 나타내려면 점(.)을 사용합니다. 예를 들어 `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`는 `HandsOffSteeringState` 액추에이터의 완전히 정규화된 이름입니다. `Vehicle.Chassis.SteeringWheel.HandsOff.`가 액추에이터의 경로입니다.

  완전히 정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).
- (선택 사항) `Description` — 신호에 대한 설명입니다.

설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) `unit` — 신호의 과학 단위 (예: km 또는 섭씨).
- (선택 사항) `min` — 신호의 최소값입니다.
- (선택 사항) `max` — 신호의 최대값입니다.
- (선택 사항) `assignedValue` — 신호에 할당된 값입니다.
- (선택 사항) `allowedValues` — 신호가 받아들이는 값 목록입니다.
- (선택 사항) `deprecationMessage` — 이동 또는 삭제되는 노드 또는 분기에 대한 지원 중단 메시지입니다.

  는 최대 2,048자를 포함할 deprecationMessage 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)
- (선택 사항) `comment` — 설명 외에 코멘트를 추가합니다. 코멘트를 사용하여 센서 또는 액추에이터에 대한 추가 정보 (예: 이론적 근거 또는 관련 센서 또는 액추에이터에 대한 참조) 를 제공할 수 있습니다.

  코멘트는 최대 2048자까지 쓸 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

복합 데이터 유형 구성

복합 데이터 유형은 비전 시스템을 모델링할 때 사용됩니다. 분기 외에도 이러한 데이터 유형은 구조(구조체라고도 함)와 속성으로 구성됩니다. 구조체는 이미지와 같이 여러 값으로 설명되는 신호입니다. 속성은 기본 데이터 유형(예: UINT8) 또는 다른 구조(예: 타임스탬프)와 같은 구조의 멤버를 나타냅니다. 예를 들어 Vehicle.Cameras.Front는 분기를 나타내고 Vehicle.Cameras.Front.Image는 구조체를 나타내며 Vehicle.Cameras.Timestamp는 속성을 나타냅니다.

다음 복잡한 데이터 유형 예제는 신호 및 데이터 유형을 단일 JSON 파일로 내보내는 방법을 보여줍니다.

Example 복합 데이터 유형

```
{
  "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**
>
> 데모 스크립트를 다운로드하여 ROS 2개의 메시지를 신호 카탈로그와 호환되는 VSS JSON 파일로 변환할 수 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요. 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

구조체 구성

사용자 지정 구조(또는 구조체)를 구성하려면 다음 정보를 지정합니다.

- `fullyQualifiedName` - 사용자 지정 구조의 정규화된 이름입니다. 예를 들어, 사용자 지정 구조의 정규화된 이름은 `ComplexDataTypes.VehicleDataTypes.SVMCamera`일 수 있습니다.

  완전히 정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).

- (선택 사항) Description — 신호에 대한 설명입니다.

  설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) deprecationMessage — 이동 또는 삭제되는 노드 또는 분기에 대한 지원 중단 메시지입니다.

  는 최대 2,048자를 포함할 deprecationMessage 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) comment — 설명 외에 코멘트를 추가합니다. 코멘트를 사용하여 센서 또는 액추에이터에 대한 추가 정보 (예: 이론적 근거 또는 관련 센서 또는 액추에이터에 대한 참조) 를 제공할 수 있습니다.

  코멘트는 최대 2048자까지 쓸 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

속성 구성

사용자 지정 속성을 구성하려면 다음 정보를 지정합니다.

- dataType – 신호의 데이터 유형은 INT8, , UINT8, INT16, , UINT16, INT32UINT32, INT64, UINT64BOOLEAN, FLOAT, STRING, DOUBLE, , UNIX_TIMESTAMP, _, INT8_ARRAY, INT16_UINT8ARRAY, ARRAYUINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64ARRAYBOOLEAN_ARRAY, FLOATARRAYDOUBLE_ARRAY, STRING_ARRAY, _, UNIX_ARRAY_, TIMESTAMP__, STRUCT, STRUCT_, _ARRAY, _ 중 하나여야 합니다UNKNOWN.
- fullyQualifiedName - 사용자 지정 속성의 정규화된 이름입니다. 예를 들어, 사용자 지정 속성의 정규화된 이름은 ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS일 수 있습니다.

  완전히 정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄)

- (선택 사항) Description — 신호에 대한 설명입니다.

  설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) deprecationMessage — 이동 또는 삭제되는 노드 또는 분기에 대한 지원 중단 메시지입니다.

  는 최대 2,048자를 포함할 deprecationMessage 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

- (선택 사항) comment — 설명 외에 코멘트를 추가합니다. 코멘트를 사용하여 센서 또는 액추에이터에 대한 추가 정보 (예: 이론적 근거 또는 관련 센서 또는 액추에이터에 대한 참조) 를 제공할 수 있습니다.

  코멘트는 최대 2048자까지 쓸 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)
- (선택 사항) dataEncoding - 속성이 바이너리 데이터인지 여부를 나타냅니다. 사용자 지정 속성의 데이터 인코딩은 BINARY 또는 중 하나여야 합니다TYPED.
- (선택 사항) structFullyQualifiedName - 사용자 지정 속성의 데이터 유형이 Struct 또는 인 경우 사용자 지정 속성에 대한 구조(구조) 노드의 정규화된 이름입니다 StructArray.

  완전히 정규화된 이름은 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).

AWS IoT FleetWise 신호 카탈로그 생성

CreateSignalCatalog API 작업을 사용하여 신호 카탈로그를 생성할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

신호 카탈로그를 만들려면 다음 명령을 실행합니다.

Replace *signal-catalog-configuration* 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

- Replace *signal-catalog-name* 생성하려는 신호 카탈로그의 이름이 표시됩니다.
- (선택 사항) 바꾸기 *description* 신호 카탈로그를 식별하는 데 도움이 되는 설명이 포함되어 있습니다.

분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 신호 구성 섹션을 참조하세요.

```
{
    "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**
>
> 데모 스크립트를 다운로드하여 ROS 2개의 메시지를 신호 카탈로그와 호환되는 VSS JSON 파일로 변환할 수 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요. 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

AWS IoT FleetWise 신호 카탈로그 가져오기

AWS IoT FleetWise 콘솔 또는 를 사용하여 신호 카탈로그를 API 가져올 수 있습니다.

주제

신호 카탈로그 가져오기(콘솔)

AWS IoT FleetWise 콘솔을 사용하여 신호 카탈로그를 가져올 수 있습니다.

> **Important**
>
> 최대 1개의 신호 카탈로그를 가질 수 있습니다. 신호 카탈로그가 이미 있는 경우 콘솔에 신호 카탈로그를 가져오는 옵션이 표시되지 않습니다.

신호 카탈로그를 가져오려는 경우

1. AWS IoT FleetWise 콘솔 을 엽니다.
2. 탐색 창에서 신호 카탈로그(Signal catalog)를 선택합니다.
3. 신호 카탈로그 요약 페이지에서 신호 카탈로그 가져오기를 선택합니다.
4. 신호가 포함된 파일을 가져옵니다.
   - S3 버킷으로부터 파일을 업로드하려면
     a. S3에서 가져오기(Import from S3)를 선택합니다.
     b. S3 찾아보기(Browse S3)를 선택합니다.
     c. 버킷의 경우 버킷 이름 또는 객체를 입력하고 목록에서 선택한 다음 목록에서 파일을 선택합니다. 파일 선택 버튼을 선택합니다.

     또는 S3 URI에 Amazon Simple Storage Service 를 입력합니다URI. 자세한 내용은 Amazon S3 사용 설명서의 버킷에 액세스하는 방법을 참조하세요.
   - 컴퓨터에서 파일을 업로드하려는 경우
     a. Import file(파일 가져오기)을 선택합니다.

b. 차량 신호 사양(VSS) 형식으로 .json 파일을 업로드합니다.

5. 신호 카탈로그를 확인한 다음 파일 가져오기를 선택합니다.

신호 카탈로그 가져오기(AWS CLI)

ImportSignalCatalog API 작업을 사용하여 신호 카탈로그를 생성하는 데 도움이 되는 JSON 파일을 업로드할 수 있습니다. Vehicle Signal Specification(VSS)을 따라 JSON 파일에 신호를 저장해야 합니다. 다음 예제에서는 를 사용합니다 AWS CLI.

신호 카탈로그를 가져오려면 다음 명령을 실행합니다.

- Replace *signal-catalog-name* 생성하려는 신호 카탈로그의 이름이 표시됩니다.
- (선택 사항) 설명을 로 바꾸기 *description* 신호 카탈로그를 식별하는 데 도움이 됩니다.
- Replace *signal-catalog-configuration-vss* 에 정의된 신호가 포함된 JSON 문자열 파일의 이름VSS.

분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 신호 구성 섹션을 참조하세요.

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

는 문자열화되고 vssJson 필드를 통과해야 JSON 합니다. 다음은 에 정의된 신호의 예입니다VSS.

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

}
```

다음 예제는 JSON 문자열VSS에 정의된 것과 동일한 신호를 보여줍니다.

```
{
   "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**
>
> 데모 스크립트를 다운로드하여 ROS 2개의 메시지를 신호 카탈로그와 호환되는 VSS JSON 파일로 변환할 수 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요. 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

AWS IoT FleetWise 신호 카탈로그 업데이트

UpdateSignalCatalog API 작업을 사용하여 기존 신호 카탈로그를 업데이트할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

기존 신호 카탈로그를 업데이트하려면 다음 명령을 실행합니다.

Replace *signal-catalog-configuration* 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

Replace *signal-catalog-name* 업데이트하려는 신호 카탈로그의 이름이 표시됩니다.

분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 신호 구성 섹션을 참조하세요.

> **Important**
>
> 사용자 지정 구조는 변경할 수 없습니다. 기존 사용자 지정 구조(구조)에 속성을 재정렬하거나 삽입해야 하는 경우 구조를 삭제하고 원하는 속성 순서로 새 구조를 생성합니다.
> 사용자 지정 구조를 삭제하려면 nodesToRemove에서 해당 구조의 정규화된 이름을 추가합니다. 신호에서 참조되는 구조는 삭제할 수 없습니다. 구조를 참조하는 모든 신호(해당 데이터 유형이 대상 구조로 정의됨)는 신호 카탈로그 업데이트를 요청하기 전에 업데이트하거나 삭제해야 합니다.

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

AWS IoT FleetWise 신호 카탈로그 삭제

DeleteSignalCatalog API 작업을 사용하여 신호 카탈로그를 삭제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

> **Important**
>
> 신호 카탈로그를 삭제하기 전에 관련 차량 모델, 디코더 매니페스트, 차량, 플릿 또는 캠페인이 없는지 확인하세요. 지침은 다음을 참조하세요.
>
> - AWS IoT FleetWise 차량 모델 삭제
> - AWS IoT FleetWise 디코더 매니페스트 삭제
> - AWS IoT FleetWise 차량 삭제
> - AWS IoT FleetWise 플릿 삭제
> - AWS IoT FleetWise 캠페인 삭제

기존 신호 카탈로그를 삭제하려면 다음 명령을 실행합니다. Replace *signal-catalog-name* 삭제하려는 신호 카탈로그의 이름이 표시됩니다.

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

> **Note**
>
> 이 명령은 출력을 생성하지 않습니다.

Get AWS IoT FleetWise 신호 카탈로그 정보

ListSignalCatalogs API 작업을 사용하여 신호 카탈로그가 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

모든 신호 카탈로그의 요약 목록을 페이지별로 검색하려면 다음 명령을 실행합니다.

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

ListSignalCatalogNodes API 작업을 사용하여 신호 카탈로그가 업데이트되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

지정된 신호 카탈로그에 있는 모든 신호(노드)의 요약 목록을 페이지별로 구분하여 검색하려면 다음 명령을 실행합니다.

Replace *signal-catalog-name* 확인 중인 신호 카탈로그의 이름이 표시됩니다.

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

GetSignalCatalog API 작업을 사용하여 신호 카탈로그 정보를 검색할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

신호 카탈로그에 대한 정보를 검색하려면 다음 명령을 실행합니다.

Replace *signal-catalog-name* 검색하려는 신호 카탈로그의 이름이 표시됩니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 즉, 신호 카탈로그의 변경 사항이 즉시 반영되지 않을 수도 있습니다.

AWS IoT FleetWise 차량 모델 관리

신호를 사용하여 차량 형식 표준화에 도움이 되는 차량 모델을 생성합니다. 차량 모델은 동일한 유형의 여러 차량에 일관된 정보를 적용하므로 여러 차량의 데이터를 처리할 수 있습니다. 동일한 차량 모델에서 생성된 차량은 동일한 신호 그룹을 상속받습니다. 자세한 내용은 AWS IoT FleetWise 차량 관리 단원을 참조하십시오.

각 차량 모델에는 차량 모델의 상태가 포함된 상태 필드가 있습니다. 상태는 다음 값 중 하나일 수 있습니다.

- `ACTIVE`— 차량 모델이 활성 상태입니다.
- `DRAFT`— 차량 모델의 구성이 저장됩니다.

Important

- `CreateModelManifest` API 작업을 사용하여 차량 모델을 생성하려면 먼저 신호 카탈로그가 있어야 합니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 생성 단원을 참조하십시오.
- AWS IoT FleetWise 콘솔을 사용하여 차량 모델을 생성하는 경우 AWS IoT FleetWise는 자동으로 차량 모델을 활성화합니다.
- `CreateModelManifest` API 작업을 사용하여 차량 모델을 생성하는 경우 차량 모델은 `DRAFT` 상태를 유지합니다.
- `DRAFT` 상태에 있는 차량 모델로는 차량을 생성할 수 없습니다. `UpdateModelManifest` API 작업을 사용하여 차량 모델을 `ACTIVE` 상태로 변경합니다.
- `ACTIVE` 상태에 있는 차량 모델은 편집할 수 없습니다.

주제

AWS IoT FleetWise 차량 모델 생성

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량 모델을 API 생성할 수 있습니다.

주제

차량 모델 생성(콘솔)

AWS IoT FleetWise 콘솔에서 다음과 같은 방법으로 차량 모델을 생성할 수 있습니다.

- 차량 모델 복제

에서 제공하는 템플릿 사용 AWS

AWS IoT FleetWise 는 신호 카탈로그, 차량 모델 및 디코더 매니페스트를 자동으로 생성하는 온보드 진단(OBD) II, J1979 템플릿을 제공합니다. 또한 템플릿은 디코더 매니페스트에 OBD 네트워크 인터페이스를 추가합니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

템플릿을 사용하여 차량 모델을 만들려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 차량 모델을 선택합니다.
3. 차량 모델 페이지에서 제공된 템플릿 추가를 선택합니다.
4. 온보드 진단(OBD) II 를 선택합니다.
5. AWS IoT FleetWise 가 생성하는 OBD 네트워크 인터페이스의 이름을 입력합니다.
6. 추가를 선택합니다.

수동으로 차량 모델 생성

하나 이상의.dbc 파일을 업로드하여 신호 카탈로그의 신호를 추가하거나 신호를 가져올 수 있습니다. .dbc 파일은 Controller Area Network(CAN 버스) 데이터베이스가 지원하는 파일 형식입니다.

> **Important**
>
> AWS IoT FleetWise 콘솔을 사용하여 비전 시스템 데이터 신호로 차량 모델을 생성할 수 없습니다. 대신 AWS CLI 를 사용하여 차량 모델을 생성합니다.
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

차량 모델을 수동으로 생성하려면

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 기본 탐색 창에서 차량 모델을 선택합니다.
3. 차량 모델 페이지에서 차량 모델 생성을 선택하고 다음을 수행합니다.

주제

1단계: 차량 모델 구성

일반 정보에서 다음을 수행합니다.

1. 차량 모델 이름을 입력합니다.
2. (선택 사항) 설명을 입력합니다.
3. 다음을 선택합니다.

2단계: 신호 추가

> **Note**
>
> - AWS IoT FleetWise 처음 사용하는 경우 신호 카탈로그가 있을 때까지 이 단계를 사용할 수 없습니다. 첫 번째 차량 모델이 생성되면 AWS IoT FleetWise 는 첫 번째 차량 모델에 추가된 신호가 포함된 신호 카탈로그를 자동으로 생성합니다.
> - AWS IoT FleetWise 를 사용한 경험이 있는 경우 신호 카탈로그에서 신호를 선택하거나 .dbc 파일을 업로드하여 신호를 가져와 차량 모델에 신호를 추가할 수 있습니다.
> - 차량 모델을 생성하려면 신호가 하나 이상 있어야 합니다.

신호를 추가하려는 경우

1. 차량 모델에 추가할 신호 카탈로그에서 하나 이상의 신호를 선택합니다. 오른쪽 창에서 선택한 신호를 리뷰할 수 있습니다.

   > **Note**
   >
   > 선택한 신호만 차량 모델에 추가됩니다.

2. 다음을 선택합니다.

3단계: 신호 가져오기

> **Note**
>
> - AWS IoT FleetWise를 처음 사용한 경우 신호를 가져오려면 .dbc 파일을 하나 이상 업로드해야 합니다.
> - AWS IoT FleetWise 를 사용한 경험이 있는 경우 신호 카탈로그에서 신호를 선택하거나 .dbc 파일을 업로드하여 신호를 가져와 차량 모델에 신호를 추가할 수 있습니다.
> - 차량 모델을 생성하려면 신호가 하나 이상 있어야 합니다.

신호를 가져오려는 경우

1. 파일 선택을 선택합니다.
2. 대화 상자에서 신호가 포함된.dbc 파일을 선택합니다. 여러 개의 .dbc 파일을 업로드할 수 있습니다.
3. AWS IoT FleetWise 검색하기 위해 .dbc 파일을 구문 분석합니다.

   신호 섹션에서 각 신호에 대해 다음 메타데이터를 지정합니다.

   - 이름 - 신호 이름.

     신호는 고유해야 합니다. 신호 이름과 경로는 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).
   - 데이터 유형 – 신호의 데이터 유형은 INT8, , UINT8, INT16, , UINT16, INT32 UINT32INT64, UINT64, BOOLEAN, , FLOAT, DOUBLE, UNIX_STRING, _TIMESTAMP, INT8_ARRAY, UINT8ARRAYINT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64ARRAY BOOLEANARRAYFLOAT_ARRAY, DOUBLE_, _ARRAY, STRING_, _, _ARRAY, _, _, UNIX_TIMESTAMP_ARRAY, 또는 중 하나여야 합니다 UNKNOWN.
   - 신호 유형 — 신호 유형으로, 센서 또는 액추에이터일 수 있습니다.
   - (선택 사항) 단위 — 신호의 과학 단위(예: km 또는 섭씨).
   - (선택 사항) 경로 — 신호 경로. 와 마찬가지로 점(.)을 JSONPath사용하여 하위 신호를 참조합니다. 예: **`Vehicle.Engine.Light`**.

     신호 이름과 경로는 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄).

- (선택 사항) 최소 — 신호의 최소값입니다.
- (선택 사항) 최대 — 신호의 최대값입니다.
- (선택 사항) 설명 – 신호에 대한 설명입니다.

  설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

4. Next(다음)를 선택합니다.

(선택 사항) 4단계: 속성 추가

신호 카탈로그의 기존 속성을 포함하여 최대 100개의 속성을 추가할 수 있습니다.

속성을 추가하려는 경우

1. 속성 추가에서 각 속성에 대해 다음 메타데이터를 지정합니다.

   - 이름 – 속성의 이름.

     신호 이름은 고유해야 합니다. 신호 이름과 경로는 최대 150자까지 가능합니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄)
   - 데이터 유형 - 속성의 데이터 유형은 INT8, , UINT8, INT16, UINT16, INT32, UINT32INT64, UINT64, BOOLEAN, , FLOAT, DOUBLE, , UNIX_STRING, _TIMESTAMP, INT8_ARRAY, UINT8ARRAYINT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64ARRAYUINT64_ARRAY, BOOLEANARRAYFLOAT_ARRAY, DOUBLE_ARRAY, _, STRING_, _, _ARRAY, _, _ARRAY_, __, UNIX_TIMESTAMP_ 중 하나여야 합니다. UNKNOWN
   - (선택 사항) 단위 — 속성의 과학 단위 (예: km 또는 섭씨).
   - (선택 사항) 경로 — 신호 경로. 와 마찬가지로 점(.)을 JSONPath사용하여 하위 신호를 참조합니다. 예: **`Vehicle.Engine.Light`**.

     신호 이름과 경로는 최대 150자까지 입력할 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, : (콜론), 및 _ (밑줄)
   - (선택 사항) 최소 — 속성의 최솟값입니다.
   - (선택 사항) 최대 — 속성의 최대값입니다.
   - (선택 사항) 설명 — 속성에 대한 설명입니다.

     설명은 최대 2048자입니다. 유효한 문자: a–z, A–Z, 0–9, :(콜론), _(밑줄), 및 -(하이픈)

2. Next(다음)를 선택합니다.

5단계: 검토 및 생성

차량 모델의 구성을 확인한 다음 생성을 선택합니다.

차량 모델 복제

AWS IoT FleetWise 는 기존 차량 모델의 구성을 복사하여 새 모델을 생성할 수 있습니다. 선택한 차량 모델에 지정된 신호가 새 차량 모델에 복사됩니다.

차량 모델을 복제하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 차량 모델을 선택합니다.
3. 차량 모델 목록에서 모델을 선택한 다음 모델 복제를 선택합니다.

차량 모델을 구성하려면 수동으로 차량 모델 생성 튜토리얼을 따르세요.

AWS IoT FleetWise 가 차량 모델 생성 요청을 처리하는 데 몇 분 정도 걸릴 수 있습니다. 차량 모델이 성공적으로 생성되면 차량 모델 페이지에 상태 열에 이 표시됩니다ACTIVE. 차량 모델이 활성화되면 편집할 수 없습니다.

차량 모델 생성(AWS CLI)

CreateModelManifest API 작업을 사용하여 차량 모델(모델 매니페스트)을 생성할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

> **Important**
>
> `CreateModelManifest` API 작업을 사용하여 차량 모델을 생성하려면 먼저 신호 카탈로그가 있어야 합니다. 신호 카탈로그 생성 방법에 대한 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 생성 섹션을 참조하세요.

차량를 생성하려면 다음 명령을 실행합니다.

Replace *`vehicle-model-configuration`* 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

- Replace *vehicle-model-name* 생성하려는 차량 모델의 이름이 표시됩니다.
- Replace *signal-catalog-ARN* 신호 카탈로그의 Amazon 리소스 이름(ARN)을 사용합니다.
- (선택 사항) 바꾸기 *description* 차량 모델을 식별하는 데 도움이 되는 설명이 포함되어 있습니다.

분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 신호 구성 섹션을 참조하세요.

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

AWS IoT FleetWise 차량 모델 업데이트

UpdateModelManifest API 작업을 사용하여 기존 차량 모델(모델 매니페스트)을 업데이트할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

기존 차량 모델을 업데이트하려면 다음 명령을 실행합니다.

Replace *update-vehicle-model-configuration* 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

- Replace *vehicle-model-name* 업데이트하려는 차량 모델의 이름이 표시됩니다.
- (선택 사항) 차량 모델을 활성화하려면 바꾸기 *vehicle-model-status* 를 사용합니다`ACTIVE`.

> **Important**
>
> 차량 모델을 활성화시킨 후에는 차량 모델을 변경할 수 없습니다.

- (선택 사항) 바꾸기 *description* 차량 모델을 식별하는 데 도움이 되는 업데이트된 설명을 제공합니다.

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

AWS IoT FleetWise 차량 모델 삭제

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량 모델을 API 삭제할 수 있습니다.

> **Important**
>
> 차량 모델과 관련된 차량 및 디코더 매니페스트를 먼저 삭제해야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 삭제 및 AWS IoT FleetWise 디코더 매니페스트 삭제 단원을 참조하세요.

차량 모델 삭제(콘솔)

차량 모델을 삭제하려면 AWS IoT FleetWise 콘솔을 사용합니다.

차량 모델 삭제하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 차량 모델을 선택합니다.
3. 차량 모델 페이지에서 대상 차량 모델을 선택합니다.
4. Delete(삭제)를 선택합니다.
5. 삭제하시겠습니까 **vehicle-model-name**?에서 삭제할 차량 모델 이름을 입력한 다음 확인을 선택합니다.

차량 모델 삭제(AWS CLI)

DeleteModelManifest API 작업을 사용하여 기존 차량 모델(모델 매니페스트)을 삭제할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량 모델을 삭제하려면, 다음 명령을 실행합니다.

Replace *model-manifest-name* 삭제하려는 차량 모델의 이름을 사용합니다.

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

> **Note**
>
> 이 명령은 출력을 생성하지 않습니다.

Get AWS IoT FleetWise 차량 모델 정보

ListModelManifests API 작업을 사용하여 차량 모델이 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

모든 차량 모델의 페이지별 요약 목록을 검색하려면 다음 명령을 실행합니다.

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

ListModelManifestNodes API 작업을 사용하여 차량 모델이 업데이트되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

지정된 차량 모델에 있는 모든 신호(노드)의 요약 목록을 페이지별로 구분하여 검색하려면 다음 명령을 실행합니다.

Replace *vehicle-model-name* 확인 중인 차량 모델의 이름이 표시됩니다.

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

차량 모델에 대한 정보를 검색하려면 다음 명령을 실행합니다.

Replace *vehicle-model* 검색하려는 차량 모델의 이름을 사용합니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 다시 말해서 차량 모델을 변경하더라도 바로 반영되지 않을 수도 있습니다.

Manage AWS IoT FleetWise 디코더 매니페스트

디코더 매니페스트에는 AWS IoT FleetWise 가 차량 데이터(바이너리 데이터)를 사람이 읽을 수 있는 값으로 변환하고 데이터 분석을 위해 데이터를 준비하는 데 사용하는 디코딩 정보가 포함되어 있습니다. 네트워크 인터페이스 및 디코더 신호는 디코더 매니페스트 구성에 사용하는 핵심 구성 요소입니다.

네트워크 인터페이스

차량 내 네트워크가 사용하는 프로토콜에 대한 정보를 포함합니다. AWS IoT FleetWise 는 다음 프로토콜을 지원합니다.

컨트롤러 영역 네트워크(CAN 버스)

전자 제어 장치() 간에 데이터가 통신되는 방식을 정의하는 프로토콜입니다ECUs. ECUs 는 엔진 컨트롤 유닛, 에어백 또는 오디오 시스템일 수 있습니다.

온보드 진단(OBD) II

자체 진단 데이터가 간에 전달되는 방식을 정의하는 추가 개발 프로토콜입니다ECUs. 차량에 무엇이 잘못되었는지 식별하는 데 도움이 되는 여러 표준 진단 문제 코드(DTCs)를 제공합니다.

차량 미들웨어

네트워크 인터페이스 유형으로 정의되는 차량 미들웨어입니다. 차량 미들웨어의 예로는 로봇 운영 체제(ROS 2)와 IP(SOME/IP)를 통한 확장 가능한 서비스 지향 MiddlewarE가 있습니다.

> **Note**
>
> AWS IoT FleetWise 는 비전 시스템 데이터에 대한 미들웨어 ROS 2개를 지원합니다.

디코더 신호

특정 신호에 대한 자세한 디코딩 정보를 제공합니다. 차량 모델에 지정된 모든 신호는 디코더 신호와 페어링되어야 합니다. 디코더 매니페스트에 CAN 네트워크 인터페이스가 포함된 경우 CAN 디코더 신호가 포함되어야 합니다. 디코더 매니페스트에 OBD 네트워크 인터페이스가 포함된 경우 OBD 디코더 신호가 포함되어야 합니다.

디코더 매니페스트에 차량 미들웨어 인터페이스도 포함되어 있는 경우 메시지 디코더 신호가 포함되어야 합니다.

각 디코더 매니페스트는 차량 모델과 연결되어야 합니다. AWS IoT FleetWise 는 연결된 디코더 매니페스트를 사용하여 차량 모델을 기반으로 생성된 차량에서 데이터를 디코딩합니다.

각 디코더 매니페스트에는 디코더 매니페스트의 상태가 포함된 상태 필드가 있습니다. 상태는 다음 값 중 하나일 수 있습니다.

- `ACTIVE` – 디코더 매니페스트가 활성 상태입니다.
- `DRAFT` – 디코더 매니페스트의 컨피그레이션이 저장되지 않습니다.
- `VALIDATING` - 디코더 매니페스트가 적격성을 검증하고 있습니다. 이는 하나 이상의 비전 시스템 데이터 신호가 포함된 디코더 매니페스트에만 적용됩니다.
- `INVALID` - 디코더 매니페스트가 검증에 실패하여 아직 활성화할 수 없습니다. 이는 하나 이상의 비전 시스템 데이터 신호가 포함된 디코더 매니페스트에만 적용됩니다. ListDecoderManifests 및 GetDecoderManifest APIs 를 사용하여 검증 실패 이유를 확인할 수 있습니다.

> **Important**
>
> - AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트를 생성하는 경우 AWS IoT FleetWise는 자동으로 디코더 매니페스트를 활성화합니다.
> - `CreateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 생성하는 경우 디코더 매니페스트는 `DRAFT` 상태를 유지합니다.
> - `DRAFT` 디코더 매니페스트와 연결된 차량 모델로는 차량을 만들 수 없습니다. `UpdateDecoderManifest` API 작업을 사용하여 디코더 매니페스트를 `ACTIVE` 상태로 변경합니다.
> - `ACTIVE` 상태에 있는 디코더 매니페스트는 편집할 수 없습니다.

주제

AWS IoT FleetWise 네트워크 인터페이스 및 디코더 신호 구성

모든 디코더 매니페스트에는 적어도 하나의 네트워크 인터페이스와 디코더 신호가 있으며 이는 관련된 차량 모델에 지정된 신호와 짝을 이룹니다.

디코더 매니페스트에 CAN 네트워크 인터페이스가 포함된 경우 CAN 디코더 신호가 포함되어야 합니다. 디코더 매니페스트에 OBD 네트워크 인터페이스가 포함된 경우 OBD 디코더 신호가 포함되어야 합니다.

주제

네트워크 인터페이스 구성

CAN 네트워크 인터페이스를 구성하려면 다음 정보를 지정합니다.

- `name` - CAN 인터페이스의 이름입니다.

  인터페이스 이름은 고유해야 하며 1~100자를 포함할 수 있습니다.
- (선택 사항) `protocolName` — 프로토콜 이름.

  유효한 값: `CAN-FD` 및 `CAN`
- (선택 사항) `protocolVersion` - AWS IoT FleetWise 는 현재 CAN-FD 및 CAN 2.0b를 지원합니다.

  유효한 값: `1.0` 및 `2.0b`

OBD 네트워크 인터페이스를 구성하려면 다음 정보를 지정합니다.

- `name` - OBD 인터페이스의 이름입니다.

  인터페이스 이름은 고유해야 하며 1~100자를 포함할 수 있습니다.
- `requestMessageId` – 차량 데이터를 요청하는 메시지의 ID입니다.
- (선택 사항) `dtcRequestIntervalSeconds` - 몇 초 만에 차량에서 진단 문제 코드(DTCs)를 요청하는 빈도입니다. 예를 들어 지정된 값이 120인 경우 Edge Agent 소프트웨어는 2분마다 한 DTCs 번씩 저장된 를 수집합니다.
- (선택 사항) `hasTransmissionEcu` - 차량에 변속기 제어 모듈이 있는지 여부(TCM).

유효한 값: `true` 및 `false`

- (선택 사항) `obdStandard` - AWS IoT FleetWise 가 지원하는 OBD 표준입니다. AWS IoT FleetWise 는 현재 World Wide Harmonization On-Board Diagnostics(WWH-OBD) ISO15765-4 표준을 지원합니다.
- (선택 사항) `pidRequestIntervalSeconds` - 차량PIDs에서 OBD II를 요청하는 빈도입니다. 예를 들어 지정된 값이 120인 경우 Edge Agent 소프트웨어는 2분마다 한 PIDs 번씩 OBD II를 수집합니다.
- (선택 사항) `useExtendedIds` - 메시지IDs에서 확장 를 사용할지 여부입니다.

  유효한 값: `true` 및 `false`

차량 미들웨어 네트워크 인터페이스를 구성하려면 다음 정보를 지정합니다.

- `name` - 차량 미들웨어 인터페이스의 이름입니다.

  인터페이스 이름은 고유해야 하며 1~100자를 포함할 수 있습니다.
- `protocolName` - 프로토콜 이름입니다.

  유효값: `ROS_2`

디코더 신호 구성

CAN 디코더 신호를 구성하려면 다음 정보를 지정합니다.

- `factor` – 메시지를 디코딩하는 데 사용되는 승수입니다.
- `isBigEndian` – CAN 메시지의 바이트 순서가 빅엔디언인지 여부입니다. 빅엔디안인 경우 시퀀스에서 가장 중요한 값이 가장 낮은 저장소 주소에 먼저 저장됩니다.
- `isSigned`— 메시지 서명 여부. 서명된 메시지는 양수와 음수를 모두 표시할 수 있습니다.
- `length` – 메시지의 바이트 길이.
- `messageId` – 메시지의 ID입니다.
- `offset`— 신호 값을 계산하는 데 사용되는 오프셋입니다. 팩터와 함께 계산하면 계산은 `value = raw_value * factor + offset`와 같습니다.
- `startBit`— 메시지의 첫 번째 비트 위치를 나타냅니다.
- (선택 사항) `name` — 신호의 이름입니다.

OBD 디코더 신호를 구성하려면 다음 정보를 지정합니다.

- `byteLength` – 바이트 단위의 메시지 길이입니다.
- `offset`— 신호 값을 계산하는 데 사용되는 오프셋입니다. 스케일링과 함께 계산하면 `value = raw_value * scaling + offset`와 같습니다.
- `pid` – 이 신호에 대해 차량에서 데이터 요청에 사용되는 진단 코드입니다.
- `pidResponseLength`— 요청된 메시지의 길이입니다.
- `scaling` – 메시지를 디코딩하는 데 사용되는 승수입니다.
- `serviceMode` – 메시지의 작업 모드(진단 서비스)입니다.
- `startByte` – 메시지의 시작을 나타냅니다.
- (선택 사항) `bitMaskLength` — 메시지에서 마스킹되는 비트 수입니다.
- (선택 사항) `bitRightShift` — 오른쪽으로 이동한 위치 수입니다.

메시지 디코더 신호를 구성하려면 다음 정보를 지정합니다.

- `topicName` - 메시지 신호의 주제 이름입니다. 이는 ROS 2의 주제에 해당합니다. 구조화된 메시지 객체에 대한 자세한 내용은 섹션을 참조하세요StructuredMessage.
- `structuredMessage` - 메시지 신호에 대한 구조화된 메시지입니다. primitiveMessageDefinition, structuredMessageList정의 또는 structuredMessageDefinition 반복으로 정의할 수 있습니다.

AWS IoT FleetWise 디코더 매니페스트 생성

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량 모델에 대한 디코더 매니페스트를 API 생성할 수 있습니다.

> **Important**
>
> 디코더 매니페스트를 생성하려면 우선 차량 모델을 보유해야 합니다. 모든 디코더 매니페스트는 차량 모델과 연결되어야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 생성 단원을 참조하십시오.

주제

- 디코더 매니페스트 생성(콘솔)
- 디코더 매니페스트 만들기(AWS CLI)

디코더 매니페스트 생성(콘솔)

AWS IoT FleetWise 콘솔을 사용하여 차량 모델과 연결된 디코더 매니페스트를 생성할 수 있습니다.

> **Important**
>
> AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트에서 비전 시스템 데이터 신호를 구성할 수 없습니다. 그 대신 AWS CLI을 사용합니다. 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

디코더 매니페스트를 생성하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 기본 탐색 창에서 차량 모델을 선택합니다.
3. 대상 차량 모델을 선택합니다.
4. 차량 모델 요약 페이지에서 디코더 매니페스트 생성을 선택하고 다음을 수행합니다.

주제

- 1단계: 디코더 매니페스트 구성
- 2단계: 네트워크 인터페이스 추가
- 3단계: 검토 및 생성

1단계: 디코더 매니페스트 구성

일반 정보에서 다음을 수행합니다.

1. 디코더 매니페스트에 고유한 이름을 입력합니다.
2. (선택 사항) 설명을 입력합니다.
3. 다음을 선택합니다.

2단계: 네트워크 인터페이스 추가

각 디코더 매니페스트에는 네트워크 인터페이스가 하나 이상 있어야 합니다. 여러 개의 네트워크 인터페이스를 디코더 매니페스트에 추가할 수 있습니다.

네트워크 인터페이스 생성

- 네트워크 인터페이스에서 다음을 수행합니다.

  a. 네트워크 인터페이스 유형 에서 CAN_INTERFACE 또는 OBD_INTERFACE를 선택합니다.
  b. 네트워크 인터페이스의 고유한 이름을 입력합니다.
  c. 고유한 네트워크 인터페이스 ID를 입력합니다. AWS IoT FleetWise 에서 생성된 ID를 사용할 수 있습니다.
  d. 차량 모델에 지정된 하나 이상의 신호를 선택하여 디코더 신호와 페어링합니다.
  e. 디코딩 정보를 제공하려면.dbc 파일을 업로드하세요. AWS IoT FleetWise 는 .dbc 파일을 구문 분석하여 디코더 신호를 검색합니다.
  f. 페어링 신호 섹션에서 모든 신호가 디코더 신호와 페어링되었는지 확인합니다.
  g. 다음을 선택합니다.

> **Note**
>
> - 각 네트워크 인터페이스마다 한개의 .dbc 파일만 업로드할 수 있습니다.
> - 차량 모델에 지정된 모든 신호가 디코더 신호와 페어링되었는지 확인합니다.
> - 다른 네트워크 인터페이스를 추가하기로 선택한 후에는 편집 중인 인터페이스를 편집할 수 없습니다. 기존 네트워크 인터페이스를 모두 삭제할 수 있습니다.

3단계: 검토 및 생성

디코더 매니페스트의 구성을 확인한 다음 생성을 선택합니다.

디코더 매니페스트 만들기(AWS CLI)

CreateDecoderManifest API 작업을 사용하여 디코더 매니페스트를 생성할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

디코더 매니페스트를 만들려면 다음 명령을 실행합니다.

Replace *decoder-manifest-configuration* 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

- Replace *decoder-manifest-name* 생성하려는 디코더 매니페스트의 이름을 사용합니다.
- Replace *vehicle-model-ARN* 차량 모델의 Amazon 리소스 이름(ARN)을 사용합니다.
- (선택 사항) 바꾸기 *description* 디코더 매니페스트를 식별하는 데 도움이 되는 설명이 포함되어 있습니다.

분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 네트워크 인터페이스 및 디코더 신호 구성 섹션을 참조하세요.

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

- Replace *decoder-manifest-name* 생성하려는 디코더 매니페스트의 이름을 사용합니다.
- Replace *vehicle-model-ARN* 차량 모델의 Amazon 리소스 이름(ARN)을 사용합니다.
- (선택 사항) 바꾸기 *description* 디코더 매니페스트를 식별하는 데 도움이 되는 설명이 포함되어 있습니다.

구조(구조체) 내 속성 노드의 순서는 신호 카탈로그 및 차량 모델(모델 매니페스트)에 정의된 것과 동일하게 유지되어야 합니다. 분기, 속성, 센서 및 액추에이터를 구성하는 방법에 대한 자세한 내용은 AWS IoT FleetWise 네트워크 인터페이스 및 디코더 신호 구성 섹션을 참조하세요.

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

> **Note**
>
> 데모 스크립트를 다운로드하여 비전 시스템 신호가 포함된 디코더 매니페스트를 생성할 수 있습니다. 자세한 내용은 Vision System Data Developer Guide를 참조하세요.
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

AWS IoT FleetWise 디코더 매니페스트 업데이트

UpdateDecoderManifest API 작업을 사용하여 디코더 매니페스트를 업데이트할 수 있습니다. 네트워크 인터페이스와 신호 디코더를 추가, 제거 및 업데이트할 수 있습니다. 디코더 매니페스트의 상태를 변경할 수도 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

디코더 매니페스트를 업데이트하려면 다음 명령을 실행합니다.

Replace *decoder-manifest-name* 업데이트하려는 디코더 매니페스트의 이름을 사용합니다.

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

> **Important**
> 디코더 매니페스트를 활성화한 후에는 편집할 수 없습니다.

AWS IoT FleetWise 디코더 매니페스트 삭제

AWS IoT FleetWise 콘솔 또는 를 사용하여 디코더 매니페스트를 API 삭제할 수 있습니다.

> **Important**
> 디코더 매니페스트와 연결된 차량을 먼저 삭제해야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 삭제 단원을 참조하십시오.

주제

- 디코더 매니페스트 삭제(콘솔)
- 디코더 매니페스트 삭제(AWS CLI)

디코더 매니페스트 삭제(콘솔)

AWS IoT FleetWise 콘솔을 사용하여 디코더 매니페스트를 삭제할 수 있습니다.

디코더 매니페스트를 삭제하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 기본 탐색 창에서 차량 모델을 선택합니다.
3. 대상 차량 모델을 선택합니다.
4. 차량 모델 요약 페이지에서 디코더 매니페스트 탭을 선택합니다.

5. 대상 디코더 매니페스트를 선택한 다음 삭제를 선택합니다.
6. 삭제하시겠습니까 **decoder-manifest-name**?에서 삭제할 디코더 매니페스트의 이름을 입력한 다음 확인을 선택합니다.

디코더 매니페스트 삭제(AWS CLI)

DeleteDecoderManifest API 작업을 사용하여 디코더 매니페스트를 삭제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

> **Important**
>
> 디코더 매니페스트를 삭제하기 전에 먼저 관련 차량을 삭제합니다. 자세한 내용은 AWS IoT FleetWise 차량 삭제 단원을 참조하십시오.

디코더 매니페스트를 삭제하려면 다음 명령을 실행합니다.

Replace *decoder-manifest-name* 삭제하려는 디코더 매니페스트의 이름을 사용합니다.

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

Get AWS IoT FleetWise 디코더 매니페스트 정보

ListDecoderManifests API 작업을 사용하여 디코더 매니페스트가 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

모든 디코더 매니페스트의 요약 목록을 페이지 단위로 검색하려면 다음 명령을 실행합니다.

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

ListDecoderManifestSignals API 작업을 사용하여 디코더 매니페스트의 디코더 신호가 업데이트되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

지정된 디코더 매니페스트에 있는 모든 디코더 신호(노드)의 요약 목록을 페이지별로 구분하여 검색하려면 다음 명령어를 실행합니다.

Replace *decoder-manifest-name* 확인 중인 디코더 매니페스트의 이름을 사용합니다.

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

```
                --name decoder-manifest-name
```

ListDecoderManifestNetworkInterfaces API 작업을 사용하여 디코더 매니페스트의 네트워크 인터페이스가 업데이트되었는지 확인할 수 있습니다. 다음 예에는 AWS CLI가 사용됩니다.

지정된 디코더 매니페스트에서 모든 네트워크 인터페이스의 요약 목록을 페이지별로 분류하여 검색하려면 다음 명령을 실행합니다.

Replace *decoder-manifest-name* 확인 중인 디코더 매니페스트의 이름을 사용합니다.

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

GetDecoderManifest API 작업을 사용하여 디코더 매니페스트의 네트워크 인터페이스와 디코더 신호가 업데이트되었는지 확인할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

디코더 매니페스트에 대한 정보를 검색하려면 다음 명령을 실행합니다.

Replace *decoder-manifest* 검색하려는 디코더 매니페스트의 이름을 사용합니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 다시 말해서 디코더 매니페스트를 변경하더라도 바로 반영되지 않을 수도 있습니다.

AWS IoT FleetWise 차량 관리

차량은 차량 모델의 인스턴스입니다. 차량은 차량 모델에서 생성되고 디코더 매니페스트와 연결되어야 합니다. 차량은 하나 이상의 데이터 스트림을 클라우드에 업로드합니다. 예를 들어 차량은 주행 거리, 엔진 온도, 히터 상태 데이터를 클라우드로 전송할 수 있습니다. 모든 차량에는 다음 정보가 포함되어 있습니다.

`vehicleName`

차량을 식별하는 ID입니다.

차량 이름에 개인 식별 정보(PII) 또는 기타 기밀 또는 민감한 정보를 추가하지 마세요. 차량 이름은 Amazon 을 포함한 다른 AWS 서비스에서 액세스할 수 있습니다 CloudWatch. 차량 이름은 개인 데이터나 민감한 데이터에 사용하기 위한 것이 아닙니다.

`modelManifestARN`

차량 모델(모델 매니페스트ARN)의 Amazon 리소스 이름()입니다. 모든 차량은 차량 모델을 기반으로 생성됩니다. 동일한 차량 모델에서 생성된 차량은 차량 모델에서 상속된 동일한 신호 그룹으로 구성됩니다. 이러한 신호는 신호 카탈로그에서 정의되고 표준화됩니다.

`decoderManifestArn`

디코더 매니페스트ARN의 . 디코더 매니페스트는 AWS IoT FleetWise 가 원시 신호 데이터(이진 데이터)를 사람이 읽을 수 있는 값으로 변환하는 데 사용할 수 있는 디코딩 정보를 제공합니다. 디코더 매니페스트는 차량 모델과 연결되어야 합니다. AWS IoT FleetWise 는 동일한 디코더 매니페스트를 사용하여 동일한 차량 모델을 기반으로 생성된 차량에서 원시 데이터를 디코딩합니다.

`attributes`

속성은 정적 정보가 포함된 키-값 페어입니다. 차량에는 차량 모델에서 상속된 속성이 포함될 수 있습니다. 속성을 추가하여 개별 차량을 동일한 차량 모델에서 생성된 다른 차량과 구별할 수 있습니다. 예를 들어 검은색 자동차가 있는 경우 속성에 다음 값을 지정할 수 있습니다: `{"color": "black"}`.

> **Important**
>
> 속성을 개별 차량에 추가하려면 먼저 관련 차량 모델에서 속성을 정의해야 합니다.

차량 모델, 디코더 매니페스트 및 속성에 대한 자세한 내용은 Model AWS IoT FleetWise 차량을(를) 참조하세요.

AWS IoT FleetWise 는 차량을 생성하고 관리하는 데 사용할 수 있는 다음과 같은 API 작업을 제공합니다.

- CreateVehicle - 새 차량을 생성합니다.
- BatchCreateVehicle - 하나 이상의 새 차량을 생성합니다.
- UpdateVehicle - 기존 차량을 업데이트합니다.
- BatchUpdateVehicle - 하나 이상의 기존 차량을 업데이트합니다.
- DeleteVehicle - 기존 차량을 삭제합니다.
- ListVehicles - 모든 차량의 페이지가 매겨진 요약 목록을 검색합니다.
- GetVehicle - 차량에 대한 정보를 검색합니다.

자습서

- Provision AWS IoT FleetWise 차량
- AWS IoT FleetWise의 예약된 주제
- AWS IoT FleetWise 차량 생성
- 여러 AWS IoT FleetWise 차량 생성
- AWS IoT FleetWise 차량 업데이트
- 여러 AWS IoT FleetWise 차량 업데이트
- AWS IoT FleetWise 차량 삭제
- Get AWS IoT FleetWise 차량 정보

Provision AWS IoT FleetWise 차량

차량에서 실행되는 Edge Agent for AWS IoT FleetWise 소프트웨어는 데이터를 수집하고 클라우드로 전송합니다. AWS IoT FleetWise 는 와 통합되어 AWS IoT Core 를 통해 Edge Agent 소프트웨어와 클라우드 간의 보안 통신을 지원합니다MQTT. 각 차량은 AWS IoT 사물에 해당합니다. 기존 AWS IoT 사물을 사용하여 차량을 생성하거나 AWS IoT FleetWise 를 설정하여 차량에 대한 AWS IoT 사물을 자동으로 생성할 수 있습니다. 자세한 내용은 차량 생성(AWS CLI) 단원을 참조하십시오.

AWS IoT Core 는 AWS IoT FleetWise 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 인증 및 권한 부여를 지원합니다. 차량은 X.509 인증서를 사용하여 인증(로그인)을 받고 AWS IoT

FleetWise 및 AWS IoT Core 정책을 사용하여 지정된 작업을 수행할 수 있는 권한(권한 보유)을 부여받을 수 있습니다.

차량 인증

차량을 인증하는 AWS IoT Core 정책을 생성할 수 있습니다.

차량을 인증하려는 경우

- AWS IoT Core 정책을 생성하려면 다음 명령을 실행합니다.
  - Replace *policy-name* 생성하려는 정책의 이름이 표시됩니다.
  - Replace *file-name* AWS IoT Core 정책이 포함된 JSON 파일의 이름을 사용합니다.

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

예제 정책을 사용하기 전에 다음을 수행합니다.

- Replace *region* AWS IoT FleetWise 리소스를 생성한 AWS 리전을 사용합니다.
- Replace *awsAccount* AWS 계정 ID를 사용합니다.

이 예제에는 AWS IoT FleetWise 에서 예약한 주제가 포함되어 있습니다. 정책에 주제를 추가해야 합니다. 자세한 내용은 AWS IoT FleetWise의 예약된 주제 단원을 참조하십시오.

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

차량 권한 부여

X.509 인증서를 생성하여 차량을 인증할 수 있습니다.

차량 인증하기

> **Important**
>
> 각 차량에 대한 새 인증서를 생성하는 것이 좋습니다.

1. RSA 키 페어를 생성하고 X.509 인증서를 발급하려면 다음 명령을 실행합니다.

   - Replace *cert* 의 명령 출력 내용을 저장하는 파일의 이름을 사용합니다certificatePem.
   - Replace *public-key* 의 명령 출력 내용을 저장하는 파일의 이름을 사용합니다 keyPairPublicKey.
   - Replace *private-key* 의 명령 출력 내용을 저장하는 파일의 이름을 사용합니다 keyPairPrivateKey.

   ```
   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. 출력에서 인증서의 Amazon 리소스 이름(ARN)을 복사합니다.
3. 인증서에 정책을 첨부하려면 다음 명령을 실행합니다.

   - Replace *policy-name* 생성한 AWS IoT Core 정책의 이름이 표시됩니다.
   - Replace *certificate-arn* 복사한 인증서ARN의 를 사용합니다.

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

4. 인증서를 사물에 연결하려면 다음 명령을 실행합니다.

   - Replace *thing-name* AWS IoT 사물의 이름 또는 차량의 ID를 사용합니다.
   - Replace *certificate-arn* 복사한 인증서ARN의 를 사용합니다.

   ```
   aws iot attach-thing-principal \
   ```

```
    --thing-name thing-name \
    --principal "certificate-arn"
```

AWS IoT FleetWise의 예약된 주제

AWS IoT FleetWise 는 다음 주제를 사용합니다. 예약된 주제가 허용하는 경우 해당 주제를 구독하거나 게시할 수 있습니다. 그러나 달러 기호로 시작하는 새 주제를 생성할 수는 없습니다. 예약된 주제와 함께 지원되지 않는 게시 또는 구독 작업을 사용하면 연결이 종료될 수 있습니다.

주제	허용된 클라이언트 작업	설명	
`$aws/iotfleetwise/vehicles/vehicleName/checkins`	게시	Edge Agent 소프트웨어는 차량 상태 정보를 이 주제에 게시합니다. 차량 상태 정보는 프로토콜 버퍼(protobuf) 형식으로 교환됩니다. 자세한 내용은 Edge Agent for AWS IoT FleetWise 소프트웨어 개발자 안내서를 참조하세요.	
`$aws/iotfleetwise/vehicles/vehicleName/signals`	Publish	엣지 에이전트 소프트웨어는 이 주제에 신호를 게시합니다. 신호 정보는 프로토콜 버퍼(protobuf) 형식으로 교환됩니다. 자세한 내용은 Edge Agent for AWS IoT FleetWise 소프트웨어 개발자 안내서를 참조하세요.	

주제	허용된 클라이언트 작업	설명	
`$aws/iotfleetwise/vehicles/vehicleName/collection_schemes`	Subscribe	AWS IoT FleetWise는 이 주제에 데이터 수집 체계를 게시합니다. 차량은 이러한 데이터 수집 체계를 사용합니다.	
`$aws/iotfleetwise/vehicles/vehicleName/decoder_manifests`	Subscribe	AWS IoT FleetWise는 디코더 매니페스트를 이 주제에 게시합니다. 차량은 이러한 디코더 매니페스트를 사용합니다.	

AWS IoT FleetWise 차량 생성

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량을 API 생성할 수 있습니다.

> **Important**
>
> 시작하기 전에 다음 사항에 유의하세요.
>
> - 차량 모델이 있어야 하고 차량 모델이 `ACTIVE` 상태여야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 모델 관리 단원을 참조하십시오.
> - 차량 모델은 디코더 매니페스트와 연결되어야 하고 디코더 매니페스트는 `ACTIVE` 상태여야 합니다. 자세한 내용은 Manage AWS IoT FleetWise 디코더 매니페스트 단원을 참조하십시오.

주제

- 차량 생성 (콘솔)
- 차량 생성(AWS CLI)

차량 생성 (콘솔)

AWS IoT FleetWise 콘솔을 사용하여 차량을 생성할 수 있습니다.

차량을 생성하려는 경우

1. AWS IoT FleetWise 콘솔 을 엽니다.
2. 탐색 창에서 차량을 선택합니다.
3. 차량 요약 페이지에서 차량 생성을 선택하고 다음 단계를 수행합니다.

주제

1단계: 차량 속성 정의

이 단계에서는 차량 이름을 지정하고 모델 매니페스트 및 디코더 매니페스트와 연결합니다.

1. 차량의 고유한 이름을 입력합니다.

   > **Important**
   >
   > 차량은 AWS IoT 사물에 해당합니다. 해당 이름의 사물이 이미 있는 경우 차량을 IoT 사물에 연결을 선택하여 헤딩 차량으로 사물을 업데이트합니다. 또는 다른 차량 이름을 선택하면 AWS IoT FleetWise 가 차량에 대한 새 항목을 자동으로 생성합니다.

2. 목록에서 차량 모델(모델 매니페스트)을 선택합니다.
3. 목록에서 디코더 매니페스트를 선택합니다. 디코더 매니페스트는 차량 모델과 연결됩니다.
4. (선택 사항) 차량 속성을 연결하려면 속성 추가를 선택합니다. 이 단계를 건너뛰면 차량이 생성된 후 속성을 추가해야 캠페인에 배포할 수 있습니다.
5. (선택 사항) 태그를 차량에 연결하려면 새 태그 추가를 선택합니다. 차량을 만든 후에도 태그를 추가할 수 있습니다.
6. 다음을 선택합니다.

2단계: 차량 인증서 구성

차량을 AWS IoT 사물으로 사용하려면 정책이 연결된 차량 인증서를 구성해야 합니다. 이 단계를 건너뛰면 차량이 생성된 후 인증서를 구성해야 캠페인에 배포할 수 있습니다.

1. 신규 인증서 자동 생성(권장)을 선택합니다.
2. 다음을 선택합니다.

3단계: 인증서에 정책 연결

이전 단계에서 구성한 인증서에 정책을 연결합니다.

1. 정책의 경우 기존 정책 이름을 입력합니다. 새 정책을 생성하려면, 정책 생성을 선택합니다.
2. 다음을 선택합니다.

4단계: 검토 및 생성

차량 구성을 확인한 다음 차량 생성을 선택합니다.

> **Important**
>
> 차량을 생성한 후에는 인증서와 키를 다운로드해야 합니다. 인증서와 프라이빗 키를 사용하여 Edge Agent for AWS IoT FleetWise 소프트웨어에서 차량을 연결합니다.

차량 생성(AWS CLI)

차량을 만들 때는 디코더 매니페스트와 연결된 차량 모델을 사용해야 합니다. CreateVehicle API 작업을 사용하여 차량을 생성할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량을 생성하려면 다음 명령을 실행합니다.

Replace *file-name* 차량 구성이 포함된 JSON 파일의 이름을 포함합니다.

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

Example 차량 구성

- (선택 사항) `associationBehavior` 값은 다음 중 하나일 수 있습니다.

- `CreateIotThing` - 차량이 생성되면 AWS IoT FleetWise 는 차량의 차량 ID 이름으로 AWS IoT 사물을 자동으로 생성합니다.
- `ValidateIotThingExists`— 기존 사물을 사용하여 AWS IoT 차량을 만들 수 있습니다.

  AWS IoT 사물을 생성하려면 다음 명령을 실행합니다. Replace *`thing-name`* 생성하려는 항목의 이름이 표시됩니다.

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

지정되지 않은 경우 AWS IoT FleetWise 는 차량에 대한 AWS IoT 사물을 자동으로 생성합니다.

> **Important**
>
> 차량이 생성된 후 AWS IoT 사물이 프로비저닝되었는지 확인합니다. 자세한 내용은 Provision AWS IoT FleetWise 차량 단원을 참조하십시오.

- Replace *`vehicle-name`* 다음 중 하나를 사용합니다.
  - 가 로 구성된 경우 AWS IoT 사물의 이름`associationBehavior`입니다`ValidateIotThingExists`.
  - `associationBehavior`가 `CreateIotThing`으로 구성된 경우 생성할 차량 ID입니다.

  차량 ID는 1~100자일 수 있습니다. 유효한 문자: a–z, A–Z, 0–9, 대쉬 (-), 밑줄 (_), 및 콜론 (:).
- Replace *`model-manifest-ARN`* ARN 차량 모델의 를 사용합니다(모델 매니페스트).
- Replace *`decoder-manifest-ARN`* 지정된 차량 모델과 연결된 디코더 매니페스트ARN의 를 사용합니다.
- (선택 사항) 속성을 추가하여 이 차량을 동일한 차량 모델에서 만든 다른 차량과 구별할 수 있습니다. 예를 들어 전기 자동차를 사용하는 경우 속성에 다음 값을 지정할 수 있습니다: `{"fuelType": "electric"}`.

> **Important**
>
> 속성을 개별 차량에 추가하려면 먼저 관련 차량 모델에서 속성을 정의해야 합니다.

```
{
    "associationBehavior": "associationBehavior",
    "vehicleName": "vehicle-name",
```

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

여러 AWS IoT FleetWise 차량 생성

BatchCreateVehicle API 작업을 사용하여 한 번에 여러 차량을 생성할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량을 여러 대 생성하려면 다음 명령을 실행합니다.

Replace *file-name* 여러 차량의 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

Example 차량 구성

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

```
}
```

각 배치 작업에 대해 최대 10대의 차량을 생성할 수 있습니다. 구성 파일에 대한 자세한 내용은 차량 생성(AWS CLI) 섹션을 참조하세요.

AWS IoT FleetWise 차량 업데이트

UpdateVehicle API 작업을 사용하여 기존 차량을 업데이트할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량을 업데이트하려면 다음 명령을 실행합니다.

Replace *file-name* 차량의 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

Example 차량 구성

- Replace *vehicle-name* 업데이트하려는 차량의 ID를 사용합니다.
- (선택 사항) 바꾸기 *model-manifest-ARN* 사용 중인 ARN 차량 모델을 교체하는 데 사용하는 차량 모델(모델 매니페스트)의 를 사용합니다.
- (선택 사항) 바꾸기 *decoder-manifest-ARN* 지정한 새 차량 모델과 연결된 디코더 매니페스트 ARN의 를 사용합니다.
- (선택 사항) 바꾸기 *attribute-update-mode* 차량 속성 포함.
  - `Merge`— 기존 속성을 새 값으로 업데이트하고 새 속성이 없으면 추가하여 새 속성을 기존 속성에 병합합니다.

    예를 들어, 차량에 `{"color": "black", "fuelType": "electric"}`와(과) 같은 속성이 있고 차량을 `{"color": "", "fuelType": "gasoline", "model": "x"}` 속성으로 업데이트하면 업데이트된 차량의 속성은 다음과 같습니다: `{"fuelType": "gasoline", "model": "x"}`.
  - `Overwrite`— 기존 속성을 새 속성으로 대체합니다.

    예를 들어, 차량에 `{"color": "black", "fuelType": "electric"}`와(과) 같은 속성이 있는 경우 차량을 해당 `{"model": "x"}` 속성으로 업데이트하면 업데이트된 차량에도 해당 `{"model": "x"}` 속성이 있습니다.

  입력에 속성이 있는 경우 이는 필수입니다.

- (선택 사항) 새 속성을 추가하거나 기존 속성을 새 값으로 업데이트하려면 attributes를 구성하세요. 예를 들어 전기 자동차를 사용하는 경우 속성에 다음 값을 지정할 수 있습니다: {"fuelType": "electric"}.

  속성을 삭제하려면 attributeUpdateMode를 Merge로 구성하세요.

  > **Important**
  >
  > 속성을 개별 차량에 추가하려면 먼저 관련 차량 모델에서 속성을 정의해야 합니다.

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

여러 AWS IoT FleetWise 차량 업데이트

BatchUpdateVehicle API 작업을 사용하여 여러 기존 차량을 한 번에 업데이트할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

여러 차량을 업데이트하려면 다음 명령을 실행합니다.

Replace *file-name* 여러 차량의 구성이 포함된 JSON 파일의 이름을 사용합니다.

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

Example 차량 구성

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

각 일괄 작업에 대해 최대 10대의 차량을 업데이트할 수 있습니다. 이런 종류의 구성에 대한 자세한 정보는 AWS IoT FleetWise 차량 업데이트 섹션을 참조하세요.

AWS IoT FleetWise 차량 삭제

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량을 API 삭제할 수 있습니다.

> **Important**
>
> 차량이 삭제되면 AWS IoT FleetWise 는 연결된 플릿 및 캠페인에서 차량을 자동으로 제거합니다. 자세한 내용은 AWS IoT FleetWise에서 플릿 관리 및 캠페인으로 AWS IoT FleetWise 데이터 수집 단원을 참조하세요. 하지만 차량은 여전히 사물로 존재하거나 의 사물과 연결되어 있습니다 AWS IoT Core. 사물 삭제에 대한 지침은 AWS IoT Core 개발자 가이드의 사물 삭제를 참조하세요.

차량 삭제(콘솔)

AWS IoT FleetWise 콘솔을 사용하여 차량을 삭제할 수 있습니다.

차량을 삭제하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 차량을 선택합니다.
3. 차량 페이지에서 삭제하려는 차량 옆에 있는 버튼을 선택합니다.

4. Delete(삭제)를 선택합니다.
5. 삭제**vehicle-name**에서, 차량 이름을 입력한 다음 삭제를 선택합니다.

차량 삭제(AWS CLI)

DeleteVehicle API 작업을 사용하여 차량을 삭제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

차량을 삭제하려면 다음 명령을 실행합니다.

Replace *vehicle-name* 삭제하려는 차량의 ID를 사용합니다.

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

Get AWS IoT FleetWise 차량 정보

ListVehicles API 작업을 사용하여 차량이 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

모든 차량에 대한 페이지 매겨진 요약 목록을 검색하려면 다음 명령을 실행합니다.

```
aws iotfleetwise list-vehicles
```

GetVehicle API 작업을 사용하여 차량 정보를 검색할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량의 메타데이터를 검색하려면 다음 명령을 실행합니다.

Replace *vehicle-name* 검색하려는 차량의 ID를 사용합니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 다시 말해서 차량을 변경하더라도 바로 반영되지 않을 수도 있습니다.

AWS IoT FleetWise에서 플릿 관리

플릿은 차량 그룹을 나타냅니다. 관련 차량이 없는 플릿은 공허한 존재입니다. 플릿을 사용하여 여러 차량을 동시에 관리하려면 먼저 차량을 플릿과 연결해야 합니다. 한 대의 차량이 다중 플릿에 속할 수 있습니다. 캠페인을 배포하여 여러 차량의 플릿에서 어떤 데이터를 수집할지, 언제 데이터를 수집할지 제어할 수 있습니다. 자세한 내용은 캠페인으로 AWS IoT FleetWise 데이터 수집 단원을 참조하십시오.

플릿에는 다음 정보가 포함됩니다.

`fleetId`

플릿의 ID입니다.

(선택 사항) `description`

플릿을 찾는 데 도움이 되는 설명.

`signalCatalogArn`

신호 카탈로그의 Amazon 리소스 이름(ARN)입니다.

AWS IoT FleetWise 는 플릿을 생성하고 관리하는 데 사용할 수 있는 다음과 같은 API 작업을 제공합니다.

- CreateFleet - 동일한 신호 그룹을 포함하는 차량 그룹을 생성합니다.
- AssociateVehicleFleet – 차량을 플릿에 연결합니다.
- DisassociateVehicleFleet - 플릿에서 차량의 연결을 해제합니다.
- UpdateFleet - 기존 플릿에 대한 설명을 업데이트합니다.
- DeleteFleet - 기존 플릿을 삭제합니다.
- ListFleets - 모든 플릿의 페이지가 매겨진 요약 목록을 검색합니다.
- ListFleetsForVehicle - 차량이 속한 모든 플릿IDs의 페이지가 매겨진 목록을 검색합니다.
- ListVehiclesInFleet - 플릿의 모든 차량에 대한 페이지가 매겨진 요약 목록을 검색합니다.
- GetFleet - 플릿에 대한 정보를 검색합니다.

주제

- AWS IoT FleetWise 플릿 생성

- AWS IoT FleetWise 차량을 플릿과 연결
- 플릿에서 AWS IoT FleetWise 차량 연결 해제
- AWS IoT FleetWise 플릿 업데이트
- AWS IoT FleetWise 플릿 삭제
- Get AWS IoT FleetWise 플릿 정보

AWS IoT FleetWise 플릿 생성

CreateFleet API 작업을 사용하여 차량 플릿을 생성할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

> **Important**
>
> 플릿을 생성하려면 먼저 신호 카탈로그를 생성해야 합니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 생성 단원을 참조하십시오.

플릿을 생성하려면 다음 명령을 실행합니다.

- Replace *fleet-id* 생성하려는 플릿의 ID를 사용합니다.

  플릿 ID는 고유해야 하며 1~100자여야 합니다. 유효한 문자: 문자 (A-Z 및 a~z), 숫자 (0-9), 콜론 (:), 대시 (-), 및 밑줄 (_).
- (선택 사항) 바꾸기 *description* 설명이 있습니다.

  설명은 1~2048자까지 입력할 수 있습니다.
- Replace *signal-catalog-arn* 신호 카탈로그ARN의 를 사용합니다.

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

AWS IoT FleetWise 차량을 플릿과 연결

AssociateVehicleFleet API 작업을 사용하여 차량을 플릿과 연결할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

> **Important**
>
> - 차량을 플릿과 연결하려면 먼저 차량과 플릿이 있어야 합니다. 자세한 내용은 AWS IoT FleetWise 차량 관리 단원을 참조하십시오.
> - 차량을 캠페인의 대상이 되는 플릿과 연결하면 AWS IoT FleetWise 는 캠페인을 차량에 자동으로 배포합니다.

차량을 플릿과 연결하려면 다음 명령을 실행합니다.

- Replace *fleet-id* 플릿의 ID를 사용합니다.
- Replace *vehicle-name* 차량 ID를 사용합니다.

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

플릿에서 AWS IoT FleetWise 차량 연결 해제

DisassociateVehicleFleet API 작업을 사용하여 차량과 플릿의 연결을 해제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

차량과 플릿의 연결을 해제하려면 다음 명령을 실행합니다.

- Replace *fleet-id* 플릿의 ID를 사용합니다.
- Replace *vehicle-name* 차량 ID를 사용합니다.

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

AWS IoT FleetWise 플릿 업데이트

UpdateFleet API 작업을 사용하여 플릿에 대한 설명을 업데이트할 수 있습니다. 다음 예에는 AWS CLI가 사용됩니다.

플릿을 업데이트하려면 다음 명령을 실행합니다.

- Replace *fleet-id* 업데이트하려는 플릿의 ID를 사용합니다.
- Replace *description* 새 설명과 함께.

  설명은 1~2048자까지 입력할 수 있습니다.

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

AWS IoT FleetWise 플릿 삭제

DeleteFleet API 작업을 사용하여 플릿을 삭제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

> **Important**
>
> 플릿을 삭제하기 전에 연결된 차량이 없는지 확인하세요. 차량과 플릿의 연결을 해제하는 방법에 대한 자세한 내용은 플릿에서 AWS IoT FleetWise 차량 연결 해제을 참조하세요.

플릿을 삭제하려면 다음 명령을 실행합니다.

Replace *fleet-id* 삭제하려는 플릿의 ID를 사용합니다.

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

Get AWS IoT FleetWise 플릿 정보

ListFleets API 작업을 사용하여 플릿이 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

모든 플릿의 요약이 페이지별로 구분된 목록을 검색하려면 다음 명령을 실행합니다.

```
aws iotfleetwise list-fleets
```

ListFleetsForVehicle API 작업을 사용하여 차량이 속한 모든 플릿IDs의 페이지가 매겨진 목록을 검색할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

차량이 속한 모든 플릿IDs의 페이지 매김 목록을 검색하려면 다음 명령을 실행합니다.

Replace *vehicle-name* 차량 ID를 사용합니다.

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

ListVehiclesInFleet API 작업을 사용하여 플릿의 모든 차량에 대한 페이지가 매겨진 요약 목록을 검색할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

플릿의 모든 차량에 대한 요약 목록을 페이지 단위로 검색하려면 다음 명령을 실행합니다.

Replace *fleet-id* 플릿의 ID를 사용합니다.

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

GetFleet API 작업을 사용하여 플릿 정보를 검색할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

플릿의 메타데이터를 검색하려면 다음 명령을 실행합니다.

Replace *fleet-id* 플릿의 ID를 사용합니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 다시 말해서 플릿을 변경하더라도 바로 반영되지 않을 수도 있습니다.

캠페인으로 AWS IoT FleetWise 데이터 수집

캠페인은 데이터 수집 규칙의 오케스트레이션입니다. 캠페인은 Edge Agent for AWS IoT FleetWise 소프트웨어에 데이터를 선택, 수집 및 클라우드로 전송하는 방법에 대한 지침을 제공합니다.

클라우드에 캠페인을 생성합니다. 사용자 또는 팀이 캠페인을 승인하면 AWS IoT FleetWise 는 캠페인을 차량에 자동으로 배포합니다. 캠페인을 차량 또는 여러 차량의 플릿에 배포하도록 선택할 수 있습니다. Edge Agent 소프트웨어는 실행 중인 캠페인이 차량에 배포될 때까지 데이터 수집을 시작하지 않습니다.

> **Note**
>
> 캠페인은 다음과 같은 상황이 발생하기 전까지는 작동하지 않습니다.
>
> - Edge Agent 소프트웨어가 차량에서 실행되고 있습니다. Edge Agent 소프트웨어를 개발, 설치 및 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.
>
> 1. AWS IoT FleetWise 콘솔 로 이동합니다.
> 2. 서비스 홈 페이지의 AWS IoT FleetWise 시작하기 섹션에서 엣지 에이전트 탐색을 선택합니다.
>
> - 차량을 프로비저닝 AWS IoT Core 하도록 설정했습니다. 자세한 내용은 Provision AWS IoT FleetWise 차량 단원을 참조하십시오.

각 캠페인에는 다음 정보가 포함되어 있습니다.

`signalCatalogArn`

캠페인과 연결된 신호 카탈로그의 Amazon 리소스 이름(ARN)입니다.

(선택 사항) `tags`

태그는 캠페인을 관리하는 데 사용할 수 있는 메타데이터입니다. 서로 다른 서비스의 리소스에 동일한 태그를 지정하여 리소스가 서로 연관되어 있음을 나타낼 수 있습니다.

`TargetArn`

캠페인ARN이 배포되는 차량 또는 플릿의 입니다.

`name`

캠페인을 식별하는 데 도움이 되는 고유한 이름.

`collectionScheme`

데이터 수집 체계는 Edge Agent 소프트웨어에 수집할 데이터 또는 수집 시기에 대한 지침을 제공합니다. AWS IoT FleetWise 는 현재 조건 기반 수집 체계 및 시간 기반 수집 체계를 지원합니다.

`conditionBasedCollectionScheme`

조건 기반 수집 체계는 수집할 데이터를 인식하기 위한 논리적 표현식을 사용합니다. Edge Agent 소프트웨어는 조건이 충족되는 경우 데이터를 수집합니다.

`expression`

수집할 데이터를 인식하는 데 사용되는 논리적 표현식입니다. 예를 들어 `` $variable.`myVehicle.InVehicleTemperature` > 50.0 `` 표현식이 지정된 경우 Edge Agent 소프트웨어는 50.0보다 큰 온도 값을 수집합니다. 표현식을 작성하는 방법에 대한 지침은 AWS IoT FleetWise 캠페인에 대한 논리적 표현식 섹션을 참조하세요.

(선택 사항) `triggerMode`는 다음 값 중 하나일 수 있습니다.

- `RISING_EDGE`— Edge Agent 소프트웨어는 조건이 처음으로 충족되는 경우에만 데이터를 수집합니다. 예: `` $variable.`myVehicle.AirBagDeployed` == true ``.
- `ALWAYS`— Edge Agent 소프트웨어는 조건이 충족될 때마다 데이터를 수집합니다.

(선택 사항) `minimumTriggerIntervalMs`

두 데이터 수집 이벤트 사이의 최소 기간(밀리초)입니다. 신호가 자주 바뀌는 경우 더 느린 속도로 데이터를 수집할 수 있습니다.

(선택 사항) `conditionLanguageVersion`

조건부 표현식 언어의 버전.

`timeBasedCollectionScheme`

시간 기반 수집 체계를 정의할 때는 기간을 밀리초 단위로 지정합니다. Edge Agent 소프트웨어는 기간을 사용하여 데이터 수집 빈도를 결정합니다. 예를 들어, 기간이 120,000밀리초인 경우, Edge Agent 소프트웨어는 2분에 한 번씩 데이터를 수집합니다.

(선택 사항) `compression`

무선 대역폭을 절약하고 네트워크 트래픽을 줄이기 위해 차량에서 데이터를 압축SNAPPY하도록를 지정할 수 있습니다.

기본적으로(`OFF`), Edge Agent 소프트웨어는 데이터를 압축하지 않습니다.

`dataDestinationConfigs`

캠페인에서 차량 데이터를 전송할 목적지를 선택합니다. 데이터를 Amazon S3 또는 Amazon Timestream에 저장하도록 선택할 수 있습니다.

S3는 내구성이 뛰어난 데이터 관리 기능과 다운스트림 데이터 서비스를 제공하는 비용 효율적인 데이터 스토리지 메커니즘입니다. S3를 운전 습관과 관련된 데이터나 장기 유지 관리 분석에 사용할 수 있습니다.

Timestream은 추세와 패턴을 거의 실시간으로 식별하는 데 도움이 되는 데이터 지속성 메커니즘입니다. Timestream을 사용하여 차량 속도 또는 제동의 과거 추세를 분석하는 것과 같은 시계열 데이터에 사용할 수 있습니다.

(선택 사항) `dataExtraDimensions`

신호에 대한 추가 정보를 제공하기 위한 속성을 하나 이상 추가할 수 있습니다.

(선택 사항) `description`

캠페인을 식별할 수 있는 설명을 추가할 수 있습니다.

(선택 사항) `diagnosticsMode`

진단 모드가 로 구성된 경우 `SEND_ACTIVE_DTCS`캠페인은 차량에 문제가 있는 것을 식별하는 데 도움이 되는 저장된 표준 진단 문제 코드(DTCs)를 보냅니다. 예를 들어 P0097은 엔진 제어 모듈(ECM)이 유입 공기 온도 센서 2(IAT2) 입력이 정상 센서 범위보다 낮다고 판단했음을 나타냅니다.

기본적으로(`OFF`), Edge Agent 소프트웨어는 진단 코드를 전송하지 않습니다.

(선택 사항) `expiryTime`

캠페인의 만료일을 정의할 수 있습니다. 캠페인이 만료되면 Edge Agent 소프트웨어는 이 캠페인에 지정된 대로 데이터 수집을 중지합니다. 차량에 여러 캠페인을 배포한 경우 Edge Agent 소프트웨어는 다른 캠페인을 사용하여 데이터를 수집합니다.

기본값: `253402243200` (12월 31일, 9999, 00:00:00UTC)

(선택 사항) `postTriggerCollectionDuration`

사후 트리거 수집 기간을 정의하여 스키마가 간접적으로 호출된 후 Edge Agent 소프트웨어가 지정된 기간 동안 데이터를 계속 수집하도록 할 수 있습니다. 예를 들어 다음 표현식의 조건 기반 수집 체계가 호출되는 경우 `$variable.`myVehicle.Engine.RPM` > 7000.0`Edge Agent 소프트웨어는 엔진에 대한 분당 회전수(RPM) 값을 계속 수집합니다. RPM 만 7000보다 한 번 높아지더라도 기계적 문제가 있음을 나타낼 수 있습니다. 이 경우 Edge Agent 소프트웨어에서 상태를 모니터링하는 데 도움이 되는 데이터를 계속 수집하는 것이 좋습니다.

기본 값: `0`

(선택 사항) `priority`

캠페인의 우선순위를 나타내는 정수를 지정할 수 있습니다. 수가 적은 캠페인일수록 우선 순위가 높습니다. 차량에 여러 캠페인을 배포하는 경우 우선 순위가 높은 캠페인이 먼저 시작됩니다.

기본 값: `0`

(선택 사항) `signalsToCollect`

데이터 수집 체계를 실행할 때 데이터가 수집되는 신호 목록입니다.

`name`

데이터 수집 체계를 호출할 때 데이터가 수집되는 신호의 이름입니다.

(선택 사항) `maxSampleCount`

데이터 수집 체계를 호출할 때 Edge Agent 소프트웨어가 수집하여 클라우드로 전송하는 최대 데이터 샘플 수입니다.

(선택 사항) `minimumSamplingIntervalMs`

두 데이터 샘플 수집 이벤트 사이의 최소 기간(밀리초)입니다. 신호가 자주 바뀌는 경우 이 파라미터를 사용하여 더 느린 속도로 데이터를 수집할 수 있습니다.

유효 범위: 0-4294967295

(선택 사항) `spoolingMode`

`spoolingMode`가 `TO_DISK`로 구성된 경우 Edge Agent 소프트웨어는 차량이 클라우드에 연결되어 있지 않을 때 데이터를 로컬에 임시로 저장합니다. 연결이 다시 설정되면 로컬에 저장된 데이터가 클라우드로 자동 전송됩니다.

기본 값: `OFF`

(선택 사항) `startTime`

승인된 캠페인은 시작 시 활성화됩니다.

기본 값: `0`

캠페인의 상태는 다음 값 중 하나일 수 있습니다.

- `CREATING` – AWS IoT FleetWise 가 캠페인 생성 요청을 처리하고 있습니다.

- `WAITING_FOR_APPROVAL`— 캠페인이 생성되면, `WAITING_FOR_APPROVAL` 상태가 됩니다. 캠페인을 승인하려면 `UpdateCampaign` API 작업을 사용합니다. 캠페인이 승인되면 AWS IoT FleetWise 는 대상 차량 또는 플릿에 캠페인을 자동으로 배포합니다. 자세한 내용은 AWS IoT FleetWise 캠페인 업데이트 단원을 참조하십시오.
- `RUNNING` — 캠페인이 활성화되었습니다.
- `SUSPENDED`— 캠페인이 일시 중지되었습니다. 캠페인을 재개하려면 `UpdateCampaign` API 작업을 사용합니다.

AWS IoT FleetWise 는 캠페인을 생성하고 관리하는 데 사용할 수 있는 다음 API 작업을 제공합니다.

- CreateCampaign - 새 캠페인을 생성합니다.
- UpdateCampaign - 기존 캠페인을 업데이트합니다. 캠페인이 생성된 후에는 이 API 작업을 사용하여 캠페인을 승인해야 합니다.
- DeleteCampaign - 기존 캠페인을 삭제합니다.
- ListCampaigns - 모든 캠페인에 대한 페이지가 매겨진 요약 목록을 검색합니다.
- GetCampaign - 캠페인에 대한 정보를 검색합니다.

자습서

- AWS IoT FleetWise 캠페인 생성
- AWS IoT FleetWise 캠페인 업데이트
- AWS IoT FleetWise 캠페인 삭제
- Get AWS IoT FleetWise 캠페인 정보

AWS IoT FleetWise 캠페인 생성

AWS IoT FleetWise 콘솔 또는 를 사용하여 차량 데이터를 수집하는 캠페인을 API 생성할 수 있습니다.

> **Important**
>
> 캠페인을 제대로 수행하려면 다음이 필요합니다.
>
> - Edge Agent 소프트웨어가 차량에서 실행되고 있습니다. Edge Agent 소프트웨어를 개발, 설치 및 사용하는 방법에 대한 자세한 내용은 다음을 참조하세요.
>
> 1. AWS IoT FleetWise 콘솔 로 이동합니다.

2. 서비스 홈 페이지의 AWS IoT FleetWise 시작하기 섹션에서 엣지 에이전트 탐색을 선택합니다.

- 차량을 프로비저닝 AWS IoT Core 하도록 설정했습니다. 자세한 내용은 Provision AWS IoT FleetWise 차량 단원을 참조하십시오.

주제

- 캠페인 생성하기(콘솔)
- 캠페인 생성(AWS CLI)
- AWS IoT FleetWise 캠페인에 대한 논리적 표현식

캠페인 생성하기(콘솔)

AWS IoT FleetWise 콘솔을 사용하여 캠페인을 생성하여 차량 데이터를 선택, 수집 및 클라우드로 전송할 수 있습니다.

캠페인을 생성하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 캠페인을 선택합니다.
3. 캠페인 페이지에서 캠페인 생성을 선택하고 다음 항목의 단계를 완료하세요.

주제

- 1단계: 캠페인을 구성합니다
- 2단계: 스토리지 대상 정의
- 3단계: 차량 추가
- 4단계: 검토 및 생성
- 5단계: 캠페인을 배포합니다

Important

- 캠페인을 생성하려면 먼저 신호 카탈로그와 차량이 있어야 합니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리 및 AWS IoT FleetWise 차량 관리 단원을 참조하세요.

- 캠페인을 생성한 후에는 캠페인을 승인해야 합니다. 자세한 내용은 5단계: 캠페인을 배포합니다 단원을 참조하십시오.

1단계: 캠페인을 구성합니다

일반 정보 섹션에서 다음을 수행합니다.

1. 캠페인 이름을 입력합니다.
2. (선택 사항) 설명을 입력합니다.

캠페인의 데이터 수집 체계를 구성합니다. 데이터 수집 체계는 수집할 데이터나 수집 시기에 대한 Edge Agent 소프트웨어 지침을 제공합니다. AWS IoT FleetWise 콘솔에서 다음과 같은 방법으로 데이터 수집 체계를 구성할 수 있습니다.

- 데이터 수집 체계를 수동으로 정의합니다.
- 파일을 업로드하여 데이터 수집 체계를 자동으로 정의합니다.

구성 옵션에서 다음 옵션 중 하나를 선택합니다.

- 데이터 수집 체계 유형을 수동으로 지정하고 구성표를 사용자 지정하는 옵션을 정의하려면 데이터 수집 체계 정의를 선택합니다.

  데이터 수집 체계의 유형을 수동으로 지정하고 체계를 사용자 지정하는 옵션을 정의합니다.

  1. 데이터 수집 체계 세부 정보 섹션에서 이 캠페인에서 사용할 데이터 수집 체계 유형을 선택합니다. 논리적 표현식을 사용하여 수집할 차량 데이터를 인식하려면 상태 기반을 선택합니다. 특정 기간을 사용하여 차량 데이터 수집 빈도를 결정하려면 시간 기반을 선택합니다.
  2. 캠페인에서 데이터를 수집하는 기간을 정의합니다.

     > **Note**
     >
     > 기본적으로 승인된 캠페인은 즉시 활성화되며 종료 시간이 설정되지 않습니다. 추가 요금을 피하려면 시간 범위를 지정해야 합니다.

  3. 조건 기반 데이터 수집 체계를 지정한 경우 수집할 데이터를 인식하기 위해 논리적 표현식을 정의해야 합니다. AWS IoT FleetWise 는 논리적 표현식을 사용하여 조건 기반 체계에 대해 수집

할 데이터를 인식합니다. 표현식에는 신호의 완전히 정규화된 이름을 변수, 비교 연산자 및 비교 값으로 지정해야 합니다.

예를 들어 표현식`$variable.`myVehicle.InVehicleTemperature` > 50.0`을 지정하면 AWS IoT FleetWise 는 50.0보다 큰 온도 값을 수집합니다. 표현식을 작성하는 방법에 대한 지침은 AWS IoT FleetWise 캠페인에 대한 논리적 표현식 섹션을 참조하세요.

수집할 데이터 인식에 사용되는 논리 표현식을 입력합니다.

4. (선택 사항) 조건부 표현식 언어의 버전을 지정합니다. 기본값은 1입니다.
5. (선택 사항) 두 데이터 수집 이벤트 사이의 최소 기간인 최소 트리거 간격을 지정할 수 있습니다. 예를 들어, 신호가 자주 바뀌는 경우 더 느린 속도로 데이터를 수집할 수 있습니다.
6. Edge Agent 소프트웨어가 데이터를 수집할 수 있도록 트리거 모드 조건을 지정합니다. 기본적으로 Edge Agent for AWS IoT FleetWise 소프트웨어는 항상 조건이 충족될 때마다 데이터를 수집합니다. 또는 조건이 처음으로 충족되는 경우, 즉 첫 번째 트리거 시에만 데이터를 수집할 수 있습니다.
7. 시간 기반 데이터 수집 체계를 지정한 경우 10,000~60,000밀리초 범위의 기간을 밀리초 단위로 지정해야 합니다. Edge Agent 소프트웨어는 기간을 사용하여 데이터 수집 빈도를 결정합니다.
8. (선택 사항) 구성표의 고급 구성표 옵션을 편집할 수 있습니다.

   a. 데이터를 압축하여 무선 대역폭을 절약하고 네트워크 트래픽을 줄이려면 Snappy를 선택합니다.

   b. (선택 사항) 데이터 수집 이벤트 후 데이터 수집을 계속하는 시간 (밀리초) 을 정의하려면 사후 트리거 수집 기간을 지정할 수 있습니다.

   c. (선택 사항) 캠페인의 우선 순위 수준을 나타내려면 캠페인 우선 순위를 지정할 수 있습니다. 우선 순위가 적은 캠페인이 먼저 배포되며 우선 순위가 높은 것으로 간주됩니다.

   d. Edge Agent 소프트웨어는 차량이 클라우드에 연결되어 있지 않을 때 데이터를 로컬에 임시로 저장할 수 있습니다. 연결이 다시 설정되면 로컬에 저장된 데이터가 클라우드로 자동 전송됩니다. 연결이 끊긴 경우 Edge Agent가 로컬에서 데이터 저장 여부를 지정합니다.

   e. (선택 사항) 신호에 대한 추가 정보를 제공하려면 최대 5개의 속성을 추가 데이터 차원으로 추가합니다.

- 파일을 업로드하여 데이터 수집 체계를 정의하려면 로컬 디바이스에서 .json 파일 업로드를 선택합니다. AWS IoT FleetWise 는 파일에서 정의할 수 있는 옵션을 자동으로 정의합니다. 선택한 옵션을 검토하고 업데이트할 수 있습니다.

  데이터 수집 체계에 대한 세부 정보가 포함된.json 파일을 업로드합니다.

1. 데이터 수집 체계에 대한 정보를 가져오려면 파일 선택을 선택합니다. 필요한 파일 형식에 대한 자세한 내용은 CreateCampaign API 설명서를 참조하세요.

   > **Note**
   >
   > AWS IoT FleetWise 는 현재 .json 파일 형식 확장을 지원합니다.

2. AWS IoT FleetWise 는 파일의 정보를 기반으로 데이터 수집 체계를 자동으로 정의합니다. AWS IoT FleetWise 가 선택한 옵션을 검토합니다. 필요한 경우 옵션을 업데이트할 수 있습니다.

신호 지정

데이터 수집 체계가 호출될 때 데이터를 수집할 신호를 지정할 수 있습니다.

데이터를 수집할 신호를 지정하는 경우

1. 신호의 완전히 정규화된 이름을 검색합니다.

   > **Note**
   >
   > 신호의 완전히 정규화된 이름은 신호 경로에 신호 이름을 더한 것입니다. 하위 신호를 나타내려면 점(.)을 사용합니다.
   > 예를 들어 `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`는 `HandsOffSteeringState` 액추에이터의 완전한 자격 이름입니다. `Vehicle.Chassis.SteeringWheel.HandsOff.`가 액추에이터의 경로입니다.

2. (선택 사항) 최대 샘플 수에는 데이터 수집 체계가 간접적으로 호출될 때 Edge Agent 소프트웨어가 수집하여 클라우드로 전송하는 최대 데이터 샘플 수를 입력합니다.
3. (선택 사항) 최소 샘플링 간격에 두 데이터 샘플 수집 이벤트 사이의 최소 시간(밀리초)을 입력합니다. 신호가 자주 바뀌는 경우 이 파라미터를 사용하여 더 느린 속도로 데이터를 수집할 수 있습니다.
4. 다른 신호를 추가하려면 신호 추가를 선택합니다. 최대 999개의 신호를 추가할 수 있습니다.
5. 다음을 선택합니다.

2단계: 스토리지 대상 정의

> **Note**
>
> 캠페인에 비전 시스템 데이터 신호가 포함된 경우에만 차량 데이터를 Amazon S3로 전송할 수 있습니다.
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

캠페인에서 수집한 데이터를 저장할 대상을 선택합니다. Amazon S3 또는 Amazon Timestream으로 차량 데이터를 전송할 수 있습니다.

목적지 설정에서 다음을 수행합니다.

- 드롭다운 목록에서 S3 또는 Timestream을 선택합니다.

차량 데이터를 S3 버킷에 저장하려면 Amazon S3를 선택합니다. S3는 데이터를 버킷 내의 객체로 저장하는 객체 스토리지 서비스입니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 버킷 만들기, 구성 및 작업을 참조하세요.

S3는 데이터 스토리지 비용을 최적화하고 데이터 레이크, 중앙 집중식 데이터 스토리지, 데이터 처리 파이프라인, 분석과 같은 차량 데이터를 사용하기 위한 추가 메커니즘을 제공합니다. S3를 사용하여 일괄 처리 및 분석을 위한 데이터를 저장할 수 있습니다. 예를 들어, 기계 학습(ML) 모델을 위한 하드 브레이킹 이벤트 보고서를 생성할 수 있습니다. 수신 차량 데이터는 배송 전 10분 동안 버퍼링됩니다.

Amazon S3

> **Important**
>
> AWS IoT FleetWise 에 S3 버킷에 쓸 수 있는 권한이 있는 경우에만 S3로 데이터를 전송할 수 있습니다. 액세스 권한 부여에 대한 자세한 내용은 AWS IoT FleetWise를 사용한 액세스 제어를 참조하세요.

S3 대상 설정에서 다음을 수행하세요.

1. S3 버킷 에서 에 대한 권한이 AWS IoT FleetWise 있는 버킷을 선택합니다.
2. (선택 사항) S3 버킷에 저장된 데이터 구성에 사용할 수 있는 사용자 지정 접두사를 입력합니다.

3. 출력 형식을 선택합니다. 출력 형식은 S3 버킷에 저장되는 형식 파일입니다.
4. S3 버킷에 저장된 데이터를.gzip 파일로 압축할지 여부를 선택합니다. 데이터를 압축하면 스토리지 비용이 최소화되므로 압축하는 것이 좋습니다.
5. S3 대상 설정에서 선택한 옵션은 예제 S3 객체 URI를 변경합니다. 다음은 S3에 어떤 파일로 저장되는지의 예입니다.

타임스트림 테이블에 차량 데이터를 저장하려면 Amazon Timestream을 선택합니다. Timestream을 사용하여 차량 데이터를 쿼리하여 추세와 패턴을 식별할 수 있습니다. 예를 들어 Timestream을 사용하여 차량 연료 수준에 대한 알람을 만들 수 있습니다. 들어오는 차량 데이터는 거의 실시간으로 Timestream으로 전송됩니다. 자세한 내용은 Amazon Timestream 개발자 가이드에서 Amazon Timestream이란 무엇인가요?를 참조하세요.

Amazon Timestream

> **Important**
>
> AWS IoT FleetWise 에 Timestream에 데이터를 쓸 수 있는 권한이 있는 경우에만 테이블로 데이터를 전송할 수 있습니다. 액세스 권한 부여에 대한 자세한 내용은 AWS IoT FleetWise를 사용한 액세스 제어를 참조하세요.

타임스트림 테이블 설정에서 다음을 수행합니다.

1. 타임스트림 데이터베이스 이름의 경우 드롭다운 목록에서 타임스트림 데이터베이스의 이름을 선택합니다.
2. 타임스트림 테이블 이름의 경우 드롭다운 목록에서 타임스트림 테이블의 이름을 선택합니다.

타임스트림용 서비스 액세스에서 다음을 수행합니다.

- 드롭다운 목록에서 IAM 역할을 선택합니다.

- Next(다음)를 선택합니다.

3단계: 차량 추가

캠페인을 전개할 차량을 선택하려면 차량 목록에서 해당 차량을 선택합니다. 차량을 만들 때 추가한 속성과 값을 검색하거나 차량 이름을 기준으로 차량을 필터링할 수 있습니다.

차량 필터링에서 다음을 수행합니다.

1. 검색 상자에서 속성 또는 차량 이름을 찾아 목록에서 선택합니다.

   > **Note**
   >
   > 각 속성은 한 번만 사용할 수 있습니다.

2. 캠페인을 배포할 대상 차량 이름 또는 속성의 값을 입력합니다. 예를 들어 속성의 완전히 정규화된 이름이 `fuelType`인 경우 해당값으로 `gasoline`을 입력합니다.
3. 다른 차량 속성을 검색하려면 이전 단계를 반복합니다. 최대 5개의 차량 속성과 무제한의 차량 이름을 검색할 수 있습니다.
4. 검색과 일치하는 차량이 차량 이름 아래에 나열됩니다. 캠페인을 배포할 차량을 선택합니다.

   > **Note**
   >
   > 검색 결과에는 최대 100대의 차량이 표시됩니다. 캠페인에 모든 차량을 추가하려면 모두 선택을 선택합니다.

5. 다음을 선택합니다.

4단계: 검토 및 생성

캠페인 구성을 확인한 다음 캠페인 생성을 선택합니다.

> **Note**
>
> 캠페인을 만든 후에는 사용자 또는 팀이 캠페인을 차량에 배포해야 합니다.

5단계: 캠페인을 배포합니다

캠페인을 만든 후에는 사용자 또는 팀이 캠페인을 차량에 배포해야 합니다.

캠페인을 배포하려는 경우

1. 캠페인 요약 페이지에서 배포를 선택합니다.
2. 배포를 시작하고 캠페인에 연결된 차량으로부터 데이터 수집을 시작할지 검토하고 확인합니다.

3. 배포(Deploy)를 선택합니다.

캠페인에 연결된 차량의 데이터 수집을 일시 중지하려면 캠페인 요약 페이지에서 일시 중지를 선택합니다. 캠페인에 연결된 차량에서 데이터 수집을 재개하려면 재개를 선택합니다.

캠페인 생성(AWS CLI)

CreateCampaign API 작업을 사용하여 캠페인을 생성할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

캠페인을 생성할 때 차량에서 수집한 데이터를 Amazon S3(S3) 또는 Amazon Timestream에 저장할 수 있습니다. Timestream을 선택하면 거의 실시간 처리가 필요한 데이터를 저장하는 등 빠르고 확장 가능하며 서버가 필요 없는 시계열 데이터베이스를 사용할 수 있습니다. 업계 최고의 확장성, 데이터 가용성, 보안, 성능을 갖춘 객체 스토리지로 S3를 선택하세요.

> **Important**
>
> AWS IoT FleetWise 에 S3 또는 Timestream에 데이터를 쓸 수 있는 권한이 있는 경우에만 차량 데이터를 전송할 수 있습니다. 액세스 권한 부여에 대한 자세한 내용은 AWS IoT FleetWise를 사용한 액세스 제어를 참조하세요.

캠페인 생성

> **Important**
>
> - 캠페인을 만들기 전에 신호 카탈로그와 차량 또는 플릿이 있어야 합니다. 자세한 내용은 AWS IoT FleetWise 신호 카탈로그 관리, AWS IoT FleetWise 차량 관리, AWS IoT FleetWise에서 플릿 관리 단원을 참조하세요.
> - 캠페인이 생성된 후에는 `UpdateCampaign` API 작업을 사용하여 캠페인을 승인해야 합니다. 자세한 내용은 AWS IoT FleetWise 캠페인 업데이트 단원을 참조하세요.

캠페인을 생성하려면 다음 명령을 실행합니다.

Replace *`file-name`* 캠페인 구성이 포함된 JSON 파일의 이름을 포함합니다.

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

- Replace *campaign-name* 생성하려는 캠페인의 이름이 표시됩니다.
- Replace *signal-catalog-arn* 신호 카탈로그의 Amazon 리소스 이름(ARN)을 사용합니다.
- Replace *target-arn* 생성한 ARN 플릿 또는 차량의 를 사용합니다.
- Replace *bucket-arn* S3 버킷ARN의 를 사용합니다.

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

- Replace *campaign-name* 생성하려는 캠페인의 이름이 표시됩니다.
- Replace *signal-catalog-arn* 신호 카탈로그의 Amazon 리소스 이름(ARN)을 사용합니다.
- Replace *target-arn* 생성한 ARN 플릿 또는 차량의 를 사용합니다.
- Replace *role-arn* Timestream 테이블에 데이터를 전송할 수 있는 AWS IoT FleetWise 권한을 부여하는 ARN 태스크 실행 역할의 를 사용합니다.
- Replace *table-arn* Timestream 테이블ARN의 를 사용합니다.

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

AWS IoT FleetWise 캠페인에 대한 논리적 표현식

AWS IoT FleetWise 는 논리적 표현식을 사용하여 캠페인의 일부로 수집할 데이터를 인식합니다. 표현식에 대한 자세한 내용은 AWS IoT Events 개발자 안내서의 표현식을 참조하세요.

표현식 변수는 수집되는 데이터 유형에 대한 규칙을 준수하도록 구성되어야 합니다. 텔레메트리 시스템 데이터의 경우 표현식 변수는 신호의 정규화된 이름이어야 합니다. 비전 시스템 데이터의 경우 표현식은 신호의 정규화된 이름을 신호의 데이터 유형에서 속성 중 하나로 이어지는 경로와 결합합니다.

예를 들어 신호 카탈로그에 다음 노드가 포함된 경우

```
{
    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"
}
```

노드가 2가지 ROS 정의를 따르는 경우:

```
{
    Vehicle.ADAS.CameraStruct.msg:
    boolean obstaclesExists
    uint8[] image
```

```
    Obstacle[30] obstacles
}
{
    Vehicle.ADAS.Obstacle.msg:
    float32: probability
    uint8 o_type
    float32: distance
}
```

가능한 모든 이벤트 표현식 변수는 다음과 같습니다.

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

AWS IoT FleetWise 캠페인 업데이트

UpdateCampaign API 작업을 사용하여 기존 캠페인을 업데이트할 수 있습니다. 다음 명령은 를 사용합니다 AWS CLI.

- Replace *campaign-name* 업데이트하려는 캠페인의 이름이 표시됩니다.
- Replace *action* 다음 중 하나를 사용합니다.
  - APPROVE - AWS IoT FleetWise 가 차량 또는 플릿에 배포할 수 있도록 캠페인을 승인합니다.
  - SUSPEND— 캠페인을 일시 중단합니다. 캠페인이 차량에서 삭제되고 일시 중단된 캠페인에 포함된 모든 차량은 데이터 전송을 중단합니다.
  - RESUME— SUSPEND 캠페인을 다시 활성화합니다. 캠페인이 모든 차량에 재배치되고 차량이 데이터를 다시 전송합니다.

- UPDATE— 속성을 정의하고 신호와 연결하여 캠페인을 업데이트합니다.

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

AWS IoT FleetWise 캠페인 삭제

AWS IoT FleetWise 콘솔 또는 를 사용하여 캠페인API을 삭제할 수 있습니다.

캠페인 삭제 (콘솔)

캠페인을 삭제하려면 AWS IoT FleetWise 콘솔을 사용합니다.

캠페인을 삭제하려는 경우

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 탐색 창에서 캠페인을 선택합니다.
3. 캠페인 페이지에서 대상 캠페인을 선택합니다.
4. Delete(삭제)를 선택합니다.
5. 삭제 중인가요 **campaign-name**?을(를) 삭제할 캠페인을 입력한 다음 확인을 선택합니다.

캠페인 삭제(AWS CLI)

DeleteCampaign API 작업을 사용하여 캠페인을 삭제할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

캠페인을 삭제하려면 다음 명령을 실행합니다.

Replace *campaign-name* 삭제하려는 차량의 이름이 표시됩니다.

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

Get AWS IoT FleetWise 캠페인 정보

ListCampaigns API 작업을 사용하여 캠페인이 삭제되었는지 확인할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

모든 캠페인을 대상으로 페이지가 매겨진 요약 목록을 검색하려면 다음 명령을 실행합니다.

```
aws iotfleetwise list-campaigns
```

GetCampaign API 작업을 사용하여 차량 정보를 검색할 수 있습니다. 다음 예제에서는 AWS CLI를 사용합니다.

캠페인을 검색하려면 다음 명령을 실행합니다.

Replace *campaign-name* 검색하려는 캠페인의 이름이 표시됩니다.

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

> **Note**
>
> 이 작업은 결과적 일관성을 갖습니다. 다시 말해서 캠페인을 변경하더라도 바로 반영되지 않을 수도 있습니다.

AWS IoT FleetWise 차량 데이터 시각화

Edge Agent for AWS IoT FleetWise 소프트웨어는 선택한 차량 데이터를 Amazon Timestream 또는 Amazon Simple Storage Service(Amazon S3)로 전송합니다. 데이터가 데이터 대상에 도착하면 다른 AWS 서비스를 사용하여 데이터를 시각화하고 공유할 수 있습니다.

Timestream에서 차량 데이터 처리

Timestream은 하루에 수조 개의 시계열 데이터 포인트를 저장하고 분석할 수 있는 완전 관리형 시계열 데이터베이스입니다. 데이터는 고객이 관리하는 타임스트림 테이블에 저장됩니다. Timestream을 사용하여 차량 데이터를 쿼리하여 차량에 대한 통찰력을 얻을 수 있습니다. 자세한 내용은 Amazon Timestream이란 무엇입니까?를 참조하세요

Timestream으로 전송되는 기본 데이터 스키마에는 다음 필드가 포함됩니다.

필드 이름	데이터 유형	설명	
`eventId`	varchar	데이터 수집 이벤트의 ID.	
`vehicleName`	varchar	데이터가 수집된 차량의 ID.	
`name`	varchar	Edge Agent 소프트웨어가 데이터를 수집하는 데 사용하는 캠페인의 이름.	
`time`	타임스탬프	데이터 포인트의 타임스탬프입니다.	
`measure_name`	varchar	신호의 이름입니다.	
`measure_value::bigint`	bigint	정수 유형의 신호 값.	
`measure_value::double`	double	더블 유형의 신호 값.	

필드 이름	데이터 유형	설명	
`measure_value::boolean`	boolean	부울 유형의 신호 값.	

Timestream에 저장된 차량 데이터 시각화

차량 데이터가 Timestream으로 전송되면 다음 AWS 서비스를 사용하여 데이터를 시각화, 모니터링, 분석 및 공유할 수 있습니다.

- Grafana 또는 Amazon Managed Grafana를 사용하여 대시보드의 데이터를 시각화하고 모니터링할 수 있습니다. 단일 Grafana 대시보드를 사용하여 여러 AWS 소스(예: Amazon CloudWatch 및 Timestream) 및 기타 데이터 소스의 데이터를 시각화할 수 있습니다.
- Amazon QuickSight 를 사용하여 대시보드의 데이터를 분석하고 시각화합니다.

Amazon S3에서 차량 데이터 처리

Amazon S3는 원하는 양의 데이터를 저장하고 보호하는 객체 스토리지 서비스입니다. S3는 데이터 레이크, 백업 및 복원, 아카이브, 엔터프라이즈 애플리케이션, AWS IoT 디바이스 및 빅 데이터 분석과 같은 다양한 사용 사례에 사용할 수 있습니다. 데이터는 S3에 버킷의 객체로 저장됩니다. 자세한 내용은 Amazon S3란 무엇인가?를 참조하세요

Amazon S3로 전송되는 기본 데이터 스키마에는 다음 필드가 포함됩니다.

필드 이름	데이터 유형	설명	
`eventId`	varchar	데이터 수집 이벤트의 ID.	
`vehicleName`	varchar	데이터가 수집된 차량의 ID.	
`name`	varchar	Edge Agent 소프트웨어가 데이터를 수집하는 데 사용하는 캠페인의 이름.	

필드 이름	데이터 유형	설명
time	타임스탬프	데이터 포인트의 타임스탬프입니다.
measure_name	varchar	신호의 이름입니다.
measure_value_BIGINT	bigint	정수 유형의 신호 값.
measure_value_DOUBLE	double	더블 유형의 신호 값.
measure_value_BOOLEAN	boolean	부울 유형의 신호 값.
measure_value_STRUCT	struct	구조체 유형의 신호 값.

Amazon S3 객체 형식

AWS IoT FleetWise 는 차량 데이터를 S3로 전송하여 객체로 저장합니다. 데이터를 URI 고유하게 식별하는 객체를 사용하여 캠페인에서 데이터를 찾을 수 있습니다. S3 객체 URI 형식은 수집된 데이터가 비정형 데이터인지 또는 처리된 데이터인지에 따라 달라집니다.

비정형 데이터

비정형 데이터는 미리 정의되지 않은 방식으로 S3에 저장됩니다. 이미지 또는 비디오와 같은 다양한 형식일 수 있습니다.

Amazon Ion 파일의 신호 데이터를 사용하여 AWS IoT FleetWise 로 전달되는 차량 메시지는 디코딩되어 객체로 S3로 전송됩니다. S3 객체는 각 신호를 나타내며 바이너리로 인코딩됩니다.

비정형 데이터 S3 객체는 다음 형식을 URI 사용합니다.

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

처리된 데이터

처리된 데이터는 S3에 저장되며 메시지를 검증, 보강, 변환하는 처리 단계를 거칩니다. 객체 목록과 속도는 처리된 데이터의 예입니다.

S3로 전송된 데이터는 약 10분 동안 버퍼링된 레코드를 나타내는 객체로 저장됩니다. 기본적으로 AWS IoT FleetWise 는 S3에 객체를 쓰기 `year=YYYY/month=MM/date=DD/hour=HH` 전에 형식의 UTC 시간 접두사를 추가합니다. 이 접두사는 버킷에 논리적 계층 구조를 생성하는데, 계층 구조 내에서 슬래시(/) 하나당 한 계층을 생성합니다. 처리된 데이터에는 구조화되지 않은 데이터에 URI 대한 S3 객체도 포함됩니다.

처리된 데이터 S3 객체는 다음 형식을 URI 사용합니다.

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

원시 데이터

프리미티브 데이터라고도 하는 원시 데이터는 Amazon Ion 파일에서 수집된 데이터입니다. 원시 데이터를 사용하여 문제를 해결하거나 오류의 근본 원인을 파악할 수 있습니다.

원시 데이터 S3 객체는 다음 형식을 URI 사용합니다.

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

Amazon S3에 저장된 차량 데이터 분석

차량 데이터가 S3로 전송된 후에는 다음 AWS 서비스를 사용하여 데이터를 모니터링, 분석 및 공유할 수 있습니다.

다운스트림 레이블 지정 및 기계 학습(ML) 워크플로 SageMaker 를 위해 Amazon을 사용하여 데이터를 추출하고 분석합니다.

자세한 내용은 Amazon SageMaker 개발자 안내서의 다음 주제를 참조하세요.

- 데이터 처리
- 기계 학습 모델 훈련
- 이미지 레이블 지정

를 사용하여 데이터를 카탈로그화 AWS Glue 크롤러 하고 Amazon Athena 에서 분석합니다. 기본적으로 S3에 기록된 객체에는 키-값 쌍이 등호로 연결된 데이터 경로를 포함하는 Apache Hive 스타일 시간 파티션이 있습니다.

자세한 내용은 Amazon Athena 사용 설명서에서 다음 주제를 참조하세요.

- Athena에서 데이터 분할
- AWS Glue 를 사용하여 Amazon S3의 데이터 소스에 연결
- Athena를 와 함께 사용할 때의 모범 사례 AWS Glue

Athena 테이블 또는 S3 버킷 QuickSight 을 직접 읽고 Amazon을 사용하여 데이터를 시각화합니다.

> **Tip**
>
> S3에서 직접 읽는 경우 Amazon은 Apache Parquet JSON 형식을 지원하지 QuickSight 않으므로 차량 데이터가 형식인지 확인합니다.

자세한 내용은 Amazon QuickSight 사용 설명서의 다음 주제를 참조하세요.

- 지원되는 데이터 소스
- 데이터 소스 생성

AWS CLI 및 AWS SDKs 를 AWS IoT FleetWise와 함께 사용

이 섹션에서는 AWS IoT FleetWise API 요청 방법에 대한 정보를 제공합니다. AWS IoT FleetWise 작업 및 데이터 유형에 대한 자세한 내용은 AWS IoT FleetWise API 참조를 참조하세요.

다양한 프로그래밍 언어에서 AWS IoT FleetWise 를 사용하려면 다음 자동 기능이 AWS SDKs포함된 를 사용합니다.

- 서비스 요청에 대한 암호화 서명
- 요청 재시도
- 오류 응답 처리

명령줄 액세스의 경우 에서 AWS IoT FleetWise 를 사용합니다AWS CLI. 명령줄에서 AWS IoT FleetWise 및 기타 서비스를 제어하고 스크립트를 통해 자동화할 수 있습니다.

AWS IoT FleetWise 문제 해결

이 섹션의 문제 해결 정보 및 솔루션을 사용하여 AWS IoT FleetWise 관련 문제를 해결할 수 있습니다.

다음 정보는 AWS IoT FleetWise와 관련된 일반적인 문제를 해결하는 데 도움이 될 수 있습니다.

주제

AWS IoT FleetWise 디코더 매니페스트 문제

디코더 매니페스트 문제를 해결합니다.

디코더 매니페스트 API 호출 진단

Error	문제 해결 지침
`UpdateOperationFailure.ConflictingDecoderUpdate`	동일한 디코더 매니페스트에 여러 업데이트 요청이 있습니다. 잠시 기다렸다가 다시 시도하세요.
`UpdateOperationFailure.InternalFailure`	InternalFailure 는 캡슐화된 예외로 시작됩니다. 문제 자체는 캡슐화된 예외에 따라 달라집니다.
`UpdateOperationFailure.ActiveDecoderUpdate`	디코더 매니페스트가 `Active` 상태이므로 업데이트할 수 없습니다. 디코더 매니페스트 상태를 `DRAFT`로 변경한 후 다시 시도하세요.
`UpdateOperationFailure.ConflictingModelUpdate`	AWS IoT FleetWise 는 다른 사람이 수정하는 차량 모델(모델 매니페스트)에 대해 검증을 시도하고 있습니다. 잠시 기다렸다가 다시 시도하세요.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND`	차량 모델에 연결된 신호가 없습니다. 차량 모델에 신호를 추가하고 연결된 신호 카탈로그에서 해당 신호를 찾을 수 있는지 확인합니다.

Error	문제 해결 지침
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE`	차량 모델을 업데이트하여 `ACTIVE` 상태가 되도록 한 후 다시 시도하세요.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_FOUND`	AWS IoT FleetWise 가 디코더 매니페스트와 연결된 차량 모델을 찾을 수 없습니다. 차량 모델의 Amazon 리소스 이름(ARN)을 확인하고 다시 시도합니다.
`UpdateOperationFailure.ModelManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE`	차량 모델의 신호 이름을 신호 카탈로그에서 찾을 수 없기 때문에 차량 모델 검증에 실패했습니다. 차량 모델의 신호가 연결된 신호 카탈로그에 모두 포함되어 있는지 확인하세요.
`UpdateOperationFailure.ValidationFailure`	디코더 매니페스트 업데이트 요청에서 유효하지 않은 신호 또는 네트워크 인터페이스가 발견되었습니다. 예외에서 반환된 모든 신호 및 네트워크 인터페이스가 존재하는지, 사용된 모든 신호가 사용 가능한 인터페이스와 연결되어 있는지, 연결된 신호가 있는 인터페이스를 제거하지 않을지 확인합니다.
`UpdateOperationFailure.KmsKeyAccessDenied`	작업에 사용되는 AWS Key Management Service (AWS KMS) 키에 권한 문제가 있습니다. 키에 액세스할 수 있는 역할을 사용하고 있는지 확인하고 다시 시도하세요.
`UpdateOperationFailure.DecoderDoesNotExist`	디코더 매니페스트가 존재하지 않습니다. 디코더 매니페스트 이름을 확인한 후 다시 시도하세요.

`SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` 이유가 있는 비전 시스템 데이터 오류 메시지에는 요청이 실패한 이유에 대한 정보를 제공하는 힌트가 응답에 포함됩니다. 힌트를 통해 따라야 할 문제 해결 지침을 결정할 수 있습니다.

> **Note**
>
> 비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

디코더 매니페스트 비전 시스템 데이터 검증 진단

Error	문제 해결 지침
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)`	AWS IoT FleetWise 가 신호 카탈로그를 사용하여 신호 디코더에 사용되는 루트 신호 구조를 찾지 못했습니다. 구조의 루트 신호가 신호 카탈로그에 제대로 정의되어 있는지 확인하세요.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	신호 카탈로그의 프리미티브 메시지가 디코더 매니페스트 업데이트 요청에서 동일한 데이터 유형으로 정의되지 않았습니다. 요청에 정의된 프리미티브 메시지가 해당 신호 카탈로그 정의와 일치하는지 확인하세요.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)`	신호 카탈로그의 구조체에 정의된 속성 수가 디코더 매니페스트에서 디코딩하려는 속성의 수와 일치하지 않습니다. 신호 카탈로그에 정의된 신호와 비교하여 디코딩할 신호 수가 정확한지 확인하세요.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	AWS IoT FleetWise 는 디코더 매니페스트 요청에 정의되지 않은 신호 카탈로그에서 로 structuredMessageDefinition 정의된 STRUCT 신호를 발견했습니다. 디코더 매니페스트 업데이트 요청 structuredMessageDefinition 에서 각 구조가 로 정의되어 있는지 확인합니다.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	디코더 매니페스트에 사용된 구조의 루트 신호가 신호 카탈로그에서 구조로 제대로 정의되지 않았습니다. 디코더 매니페스트에 사용되는 루트 신호 구조에는 필드 structFullyQualifiedName이 정의되어 있어야 합니다. 또한 해당

Error	문제 해결 지침
	가 있는 STRUCT 노드가 필요합니다 fullyQualifiedName.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	디코더 매니페스트 요청에 사용된 리프 메시지 중 하나가 프리미티브 메시지로 정의되지 않았습니다. 요청의 모든 리프 객체가 프리미티브 메시지로 정의되었는지 확인하세요.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	신호 카탈로그의 배열 객체가 디코더 매니페스트 업데이트 요청에서 structuredMessageList정의로 정의되지 않았습니다. 디코더 매니페스트 업데이트 요청에서 모든 배열 속성이 structuredMessageList정의로 정의되어 있는지 확인합니다.

Edge Agent for AWS IoT FleetWise 소프트웨어 문제

Edge Agent 소프트웨어 문제를 해결합니다.

문제

문제: Edge Agent 소프트웨어가 시작되지 않습니다.

Edge Agent 소프트웨어가 시작되지 않을 경우 다음 오류가 표시될 수 있습니다.

- ```
  Error from reader: * Line 1, Column 1
  ```
  ```

```
Syntax error: value, object or array expected.
```

해결 방법: Edge Agent for AWS IoT FleetWise 소프트웨어 구성 파일이 유효한 JSON 형식을 사용하고 있는지 확인합니다. 예를 들어, 쉼표가 올바르게 사용되었는지 확인합니다. 구성 파일에 대한 자세한 내용은 다음을 수행하여 Edge Agent for AWS IoT FleetWise 소프트웨어 개발자 안내서를 다운로드하세요.

1. AWS IoT FleetWise 콘솔 로 이동합니다.
2. 서비스 홈 페이지의 AWS IoT FleetWise 시작하기 섹션에서 엣지 에이전트 탐색을 선택합니다.

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

해결 방법: Edge Agent 소프트웨어가 구성 파일에 정의된 네트워크 인터페이스와의 소켓 통신을 설정하지 못할 경우 이 오류가 표시될 수 있습니다.

구성에 정의된 모든 네트워크 인터페이스를 사용할 수 있는지 확인하려면 다음 명령을 실행합니다.

```
ip link show
```

네트워크 인터페이스를 온라인 상태로 전환하려면 다음 명령을 실행합니다. Replace *network-interface-id* 네트워크 인터페이스의 ID를 사용합니다.

```
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]
```

해결 방법: Edge Agent 소프트웨어가 에 대한 MQTT 연결을 설정하지 못하면 이 오류가 발생할 수 있습니다 AWS IoT Core. 다음이 올바르게 구성되었는지 확인하고 Edge Agent 소프트웨어를 다시 시작합니다.

- `mqttConnection::endpointUrl` – AWS 계정의 IoT 디바이스 엔드포인트입니다.
- `mqttConnection::clientID` – Edge Agent 소프트웨어가 실행 중인 차량의 ID.
- `mqttConnection::certificateFilename` – 차량 인증서 파일의 경로입니다.
- `mqttConnection::privateKeyFilename` – 차량 개인 키 파일의 경로.

- AWS IoT Core 를 사용하여 차량을 프로비저닝했습니다. 자세한 내용은 Provision AWS IoT FleetWise 차량 단원을 참조하십시오.

자세한 내용은 AWS IoT Device SDK for C++ FAQ를 참조하세요.

문제: [ERROR] [I oTFleetWiseEngine::connect]: [지속성 라이브러리를 시작하지 못함]

해결 방법: Edge Agent 소프트웨어가 지속성 저장소를 찾지 못할 경우 이 오류가 표시될 수 있습니다. 다음이 올바르게 구성되었는지 확인하고 Edge Agent 소프트웨어를 다시 시작합니다.

`persistency:persistencyPath` – 수집 체계, 디코더 매니페스트 및 데이터 스냅샷을 유지하는 데 사용되는 로컬 경로입니다.

문제: Edge Agent 소프트웨어는 온보드 진단(OBD) II PIDs 및 진단 문제 코드()를 수집하지 않습니다DTCs.

해결 방법: `obdInterface:pidRequestIntervalSeconds` 또는 `obdInterface:dtcRequestIntervalSeconds`가 0으로 구성된 경우 이 오류가 표시될 수 있습니다.

Edge Agent 소프트웨어가 자동 변속기 차량에서 실행 중인 경우 `obdInterface:hasTransmissionEcu`가 `true`으로 구성되어 있는지 확인합니다.

차량이 확장된 컨트롤러 영역 네트워크(CAN 버스) 중재를 지원하는 경우 `obdInterface:useExtendedIds`가 로 구성되어 있는지 IDs확인합니다`true`.

문제: Edge Agent for AWS IoT FleetWise 소프트웨어는 네트워크에서 데이터를 수집하지 않거나 데이터 검사 규칙을 적용할 수 없습니다.

해결 방법: 기본 할당량을 위반한 경우 이 오류가 표시될 수 있습니다.

Resource	할당량	조정 가능	참고
신호의 ID의 값입니다	신호의 ID는 50,000보다 작거나 같아야 합니다	예	Edge Agent 소프트웨어는 ID가 50,000을 초과하는 신호에서 데이

Resource	할당량	조정 가능	참고
			터를 수집하지 않습니다. 이 할당량을 변경하기 전에 신호 카탈로그에 포함된 신호 수를 확인하는 것이 좋습니다.
차량당 활성 데이터 수집 체계의 수	256	예	이 할당량을 변경하기 전에 클라우드에서 만든 캠페인 수와 각 캠페인에 포함된 스키마 수를 확인하는 것이 좋습니다.
신호의 히스토리 버퍼의 크기입니다	20MB	예	할당량이 초과되면 Edge Agent 소프트웨어는 새 데이터 수집을 중단합니다.

문제: [ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error] 또는 [WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]

해결 방법: Edge Agent 소프트웨어가 클라우드에 연결되지 않은 경우 이 오류가 표시될 수 있습니다. 기본적으로 Edge Agent 소프트웨어는 1분 AWS IoT Core 마다 ping 요청을 보내고 3분 동안 기다립니다. 응답이 없으면 Edge Agent 소프트웨어가 자동으로 클라우드 연결을 다시 설정합니다.

AWS IoT FleetWise의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한 는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 규정 준수 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AWS IoT FleetWise 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 AWS 프로그램 범위 내 서비스 규정 준수 프로그램 참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 여러분은 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다

이 설명서는 AWS IoT FleetWise 를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 보안 및 규정 준수 목표에 맞게 AWS IoT FleetWise 를 구성하는 방법을 보여줍니다. 또한 AWS IoT FleetWise 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

내용

AWS IoT FleetWise의 데이터 보호

AWS 공동 책임 모델 AWS IoT FleetWise의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든 를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 섹션을 FAQ참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 책임 공유 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management ()를 사용하여 개별 사용자를 설정하는 것이 좋습니다IAM. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다단계 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2가 필요하며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션과 내의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 FIPS 를 AWS 통해 에 액세스할 때 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 API사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, 또는 를 사용하여 AWS IoT FleetWise 또는 기타 AWS 서비스 로 작업하는 경우가 포함됩니다API AWS CLI AWS SDKs. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL를 제공하는 경우 해당 서버에 대한 요청을 검증URL하기 위해 에 보안 인증 정보를 포함하지 않는 것이 좋습니다.

AWS IoT FleetWise 는 지원되는 차량 하드웨어에 개발 및 설치하는 Edge Agent와 함께 사용하여 차량 데이터를 AWS 클라우드로 전송하기 위한 것입니다. 차량에서 데이터를 추출하는 경우 특정 관할 지역에서는 데이터 개인 정보 보호 규정의 적용 대상일 수 있습니다. AWS IoT FleetWise 를 사용하고 Edge

Agent를 설치하기 전에 관련 법률에 따라 규정 준수 의무를 평가하는 것이 좋습니다. 여기에는 모든 관련 법적 요건을 포함하고 있으며 이는 법적으로 적절한 개인정보 고지를 제공하고 차량 데이터 추출에 필요한 법적 동의를 획득하기 위한 용도로 사용됩니다.

AWS IoT FleetWise에서 저장 시 암호화

차량에서 수집된 데이터는 MQTT 메시지 프로토콜이 포함된 AWS IoT Core 메시지를 통해 클라우드로 전송됩니다. AWS IoT FleetWise 는 Amazon Timestream 데이터베이스로 데이터를 전송합니다. Timestream 에서는 데이터가 암호화됩니다. 저장된 모든 데이터는 기본적으로 AWS 서비스 암호화됩니다.

저장 시 암호화는 AWS Key Management Service (AWS KMS)와 통합되어 데이터를 암호화하는 데 사용되는 암호화 키를 관리합니다. 고객 관리형 키를 사용하여 AWS IoT FleetWise에서 수집한 데이터를 암호화하도록 선택할 수 있습니다. 를 통해 암호화 키를 생성, 관리 및 볼 수 있습니다 AWS KMS. 자세한 내용은 AWS Key Management Service 개발자 안내서의 란 무엇입니까 AWS Key Management Service?를 참조하세요.

전송 중 암호화

AWS IoT 서비스와 교환되는 모든 데이터는 전송 계층 보안()을 사용하여 전송 중에 암호화됩니다TLS. 자세한 정보는 AWS IoT 개발자 안내서의 전송 보안을 참조하세요.

또한 AWS IoT FleetWise 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 되는 인증 및 권한 부여를 AWS IoT Core 지원합니다. 차량은 X.509 인증서를 사용하여 인증(로그인)을 통해 AWS IoT FleetWise 를 사용하고 AWS IoT Core 정책을 사용하여 지정된 작업을 수행하도록 권한 부여(권한 부여)를 받을 수 있습니다. 자세한 내용은 the section called “차량 공급” 단원을 참조하십시오.

AWS IoT FleetWise의 데이터 암호화

데이터 암호화는 전송 중(AWS IoT FleetWise 로 오가는 동안, 게이트웨이와 서버 간에) 및 저장 중(로컬 디바이스 또는 에 저장되는 동안) 데이터를 보호하는 것을 의미합니다 AWS 서비스. 클라이언트측 암호화를 사용하여 저장 데이터를 보호할 수 있습니다.

> **Note**
>
> AWS IoT FleetWise 엣지 처리는 AWS IoT FleetWise 게이트웨이 내에서 호스팅되고 로컬 네트워크를 통해 액세스할 수 APIs 있는 를 노출합니다. 이는 AWS IoT FleetWise Edge 커넥터가 소유한 서버 인증서가 지원하는 TLS 연결을 통해 노출APIs됩니다. 클라이언트 인증의 경우 액세스 제어 암호를 APIs 사용합니다. 서버 인증서 프라이빗 키와 액세스 제어 암호는 모두 디스

크에 저장됩니다. AWS IoT FleetWise 엣지 처리는 저장 시 이러한 보안 인증 정보를 보호하기 위해 파일 시스템 암호화에 의존합니다.

서버 측 암호화 및 클라이언트측 암호화에 대한 자세한 내용은 다음 주제를 검토하십시오.

내용

AWS IoT FleetWise에서 저장 시 암호화

AWS IoT FleetWise 는 AWS 클라우드와 게이트웨이에 데이터를 저장합니다.

AWS 클라우드에 저장된 데이터

AWS IoT FleetWise AWS 서비스 는 기본적으로 저장 데이터를 암호화하는 다른 에 데이터를 저장합니다. 저장 시 암호화는 AWS IoT FleetWise 에서 자산 속성 값과 집계 값을 암호화하는 데 사용되는 암호화 키를 관리하기 위해 AWS Key Management Service (AWS KMS)와 통합됩니다. 고객 관리형 키를 사용하여 AWS IoT FleetWise 에서 자산 속성 값과 집계 값을 암호화하도록 선택할 수 있습니다. 를 통해 암호화 키를 생성, 관리 및 볼 수 있습니다 AWS KMS.

AWS 소유 키 또는 고객 관리형 키를 선택하여 데이터를 암호화할 수 있습니다.

작동 방식

유휴 암호화는 데이터를 암호화하는 데 사용되는 암호화 키를 관리하기 AWS KMS 위해 와 통합됩니다.

- AWS 소유 키 – 기본 암호화 키. AWS IoT FleetWise 가 이 키를 소유합니다. AWS 계정의 키를 확인, 관리 또는 사용할 수 없습니다. 또한 AWS CloudTrail 로그의 키에 대한 작업을 볼 수 없습니다. 추가 비용 없이 이 키를 사용할 수 있습니다.
- 고객 관리형 키 - 키는 사용자가 생성, 소유 및 관리하는 계정에 저장됩니다. KMS 키를 완전히 제어할 수 있습니다. 추가 AWS KMS 요금이 부과됩니다.

AWS 소유 키

AWS 소유 키 는 계정에 저장되지 않습니다. 이는 여러 AWS 계정. AWS 서비스 can 에서 데이터를 보호하는 AWS 소유 키 데 사용할 수 있도록 를 AWS 소유하고 관리하는 KMS 키 모음의 일부입니다.

를 보거나 관리하거나 사용하거나 사용을 AWS 소유 키감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다.

를 사용하는 경우 요금이 부과되지 AWS 소유 키않으며 계정의 AWS KMS 할당량에 포함되지 않습니다.

고객 관리형 키

고객 관리형 키는 생성, 소유 및 관리하는 계정의 KMS 키입니다. 다음과 같이 이러한 KMS 키를 완전히 제어할 수 있습니다.

- 주요 정책, IAM 정책 및 권한 부여 설정 및 유지
- 활성화 및 비활성화
- 암호화 자료 교체
- 태그 추가
- 이를 참조하는 별칭 생성
- KMS 키 삭제 예약

CloudTrail 및 Amazon CloudWatch Logs를 사용하여 AWS IoT FleetWise 가 사용자를 대신하여 보내는 요청을 추적할 수도 AWS KMS 있습니다.

고객 관리형 키를 사용하는 경우 계정에 저장된 KMS 키에 대한 AWS IoT FleetWise 액세스 권한을 부여해야 합니다. AWS IoT FleetWise 는 엔벨로프 암호화 및 키 계층 구조를 사용하여 데이터를 암호화합니다. AWS KMS 암호화 키는 이 키 계층 구조의 루트 키를 암호화하는 데 사용됩니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 봉투 암호화를 참조하세요.

다음 예제 정책은 사용자를 대신하여 고객 관리형 키 생성에 AWS IoT FleetWise 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant",
```

```
        "kms:RevokeGrant"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Important

KMS 키 정책에 새 섹션을 추가할 때 정책의 기존 섹션을 변경하지 마세요. AWS IoT FleetWise에 암호화가 활성화되어 있고 다음 중 하나라도 해당하는 경우 AWS IoT FleetWise 는 데이터에 대한 작업을 수행할 수 없습니다.

- KMS 키가 비활성화되거나 삭제됩니다.
- KMS 키 정책이 서비스에 대해 올바르게 구성되지 않았습니다.

비전 시스템 데이터에 저장 시 암호화 사용

Note

비전 시스템 데이터는 평가판 릴리스이며 변경될 수 있습니다.

AWS IoT FleetWise 계정에서 AWS KMS 키가 활성화된 고객 관리형 암호화를 사용하고 비전 시스템 데이터를 사용하려는 경우 암호화 설정을 복잡한 데이터 유형과 호환되도록 재설정합니다. 이를 통해 AWS IoT FleetWise 는 비전 시스템 데이터에 필요한 추가 권한을 설정할 수 있습니다.

Note

비전 시스템 데이터에 대한 암호화 설정을 재설정하지 않은 경우 디코더 매니페스트가 검증 중 상태로 멈출 수 있습니다.

1. GetEncryptionConfiguration API 작업을 사용하여 AWS KMS 암호화가 활성화되어 있는지 확인합니다. 암호화 유형이 `FLEETWISE_DEFAULT_ENCRYPTION`이면 추가 작업이 필요하지 않습니다.
2. 암호화 유형이 인 경우 PutEncryptionConfiguration API 작업을 `KMS_BASED_ENCRYPTION`사용하여 암호화 유형을 로 재설정합니다`FLEETWISE_DEFAULT_ENCRYPTION`.

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

3. PutEncryptionConfiguration API 작업을 사용하여 암호화 유형을 에 다시 활성화합니다KMS_BASED_ENCRYPTION.

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

암호화 활성화에 대한 자세한 내용은 AWS IoT FleetWise의 키 관리 섹션을 참조하세요.

AWS IoT FleetWise의 키 관리

AWS IoT FleetWise 클라우드 키 관리

기본적으로 AWS IoT FleetWise 는 를 AWS 관리형 키 사용하여 의 데이터를 보호합니다 AWS 클라우드. 고객 관리형 키를 사용하여 AWS IoT FleetWise 에서 데이터를 암호화하도록 설정을 업데이트할 수 있습니다. AWS Key Management Service ()를 통해 암호화 키를 생성, 관리 및 볼 수 있습니다AWS KMS.

AWS IoT FleetWise 는 다음 리소스의 데이터를 암호화 AWS KMS 하기 위해 에 저장된 고객 관리형 키로 서버 측 암호화를 지원합니다.

AWS IoT FleetWise 리소스	데이터 유형	고객 관리 키로 유휴 상태에서 암호화된 필드
신호 카탈로그		설명
	속성	설명, allowedValues, defaultValue, 최소, 최대
	액추에이터	설명, allowedValues, 최소, 최대

AWS IoT FleetWise 리소스	데이터 유형	고객 관리 키로 유휴 상태에서 암호화된 필드
	센서	설명, allowedValues, 최소, 최대
차량 모델(모델 매니페스트)		설명
디코더 매니페스트		설명
	CanInterface	protocolName, protocolVersion
	ObdInterface	requestMessageId, dtcRequestInterval초, hasTransmissionEcu, obdStandard, pidRequestInterval초, useExtendedIds
	CanSignal	인자, isBigEndian, isSigned, 길이, messageId, 오프셋, startBit
	ObdSignal	byteLength, 오프셋, pid, pidResponseLength, 조정, serviceMode, startByte bitMaskLength, bitRightShift
차량		attributes
Campaign		설명
	conditionBasedCollection스키마	표현식, conditionLanguageVersion, minimumTriggerIntervalMs, triggerMode
	TimeBasedCollectionScheme	periodMs

> **Note**
>
> 다른 데이터 및 리소스는 AWS IoT FleetWise 에서 관리하는 키가 있는 기본 암호화를 사용하여 암호화됩니다. 이 키는 생성되어 AWS IoT FleetWise 계정에 저장됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 란 무엇입니까 AWS Key Management Service?를 참조하세요.

KMS 키를 사용하여 암호화 활성화(콘솔)

AWS IoT FleetWise에서 고객 관리형 키를 사용하려면 AWS IoT FleetWise 설정을 업데이트해야 합니다.

KMS 키를 사용하여 암호화를 활성화하려면(콘솔)

1. AWS IoT FleetWise 콘솔 을 엽니다.
2. 설정으로 이동합니다.
3. 암호화에서 편집을 선택하여 암호화 편집 페이지를 엽니다.
4. 암호화 키 유형 에서 다른 AWS KMS 키 선택 을 선택합니다. 이렇게 하면 AWS KMS에 저장된 고객 관리 키로 암호화할 수 있습니다.

   > **Note**
   >
   > AWS IoT FleetWise 리소스에는 고객 관리형 키 암호화만 사용할 수 있습니다. 여기에는 신호 카탈로그, 차량 모델(모델 매니페스트), 디코더 매니페스트, 차량, 플릿 및 캠페인이 포함됩니다.

5. 다음 옵션 중 하나를 사용하여 KMS 키를 선택합니다.

   - 기존 KMS 키를 사용하려면 목록에서 KMS 키 별칭을 선택합니다.
   - 새 KMS 키를 생성하려면 - AWS KMS 키 생성을 선택합니다.

     > **Note**
     >
     > 그러면 AWS KMS 콘솔이 열립니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 생성을 참조하세요.

6. 설정을 저장하려면 저장을 선택합니다.

KMS 키를 사용하여 암호화 활성화(AWS CLI)

PutEncryptionConfiguration API 작업을 사용하여 AWS IoT FleetWise 계정에 대한 암호화를 활성화할 수 있습니다. 다음 예제에서는 를 사용합니다 AWS CLI.

암호화를 활성화하려면 다음 명령을 실행합니다.

- Replace *KMS key id* KMS 키의 ID를 사용합니다.

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

Example 응답

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

KMS 키 정책

KMS 키를 생성한 후 AWS IoT FleetWise에서 작동하려면 최소한 KMS 키 정책에 다음 문을 추가해야 합니다.

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

```
}
```

AWS IoT FleetWise 에 사용할 KMS 키 정책 편집에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 정책 변경을 참조하세요.

> **Important**
>
> KMS 키 정책에 새 섹션을 추가할 때 정책의 기존 섹션을 변경하지 마세요. AWS IoT FleetWise 에 암호화가 활성화되어 있고 다음 중 하나라도 해당하는 경우 AWS IoT FleetWise 는 데이터에 대한 작업을 수행할 수 없습니다.
>
> - KMS 키가 비활성화되거나 삭제됩니다.
> - KMS 키 정책이 서비스에 대해 올바르게 구성되지 않았습니다.

다음을 통한 AWS IoT FleetWise액세스 제어

다음 섹션에서는 AWS IoT FleetWise 리소스에 대한 액세스를 제어하는 방법을 다룹니다. 여기에서 다루는 정보에는 AWS IoT FleetWise 가 캠페인 중에 차량 데이터를 전송할 수 있도록 애플리케이션 액세스 권한을 부여하는 방법이 포함되어 있습니다. 또한 데이터를 저장할 Amazon S3(S3) 버킷 또는 Amazon Timestream 데이터베이스 및 테이블에 대한 AWS IoT FleetWise 액세스 권한을 부여하는 방법도 설명합니다.

이러한 모든 형태의 액세스를 관리하기 위한 기술은 AWS Identity and Access Management ()입니다 IAM. 에 대한 자세한 내용은 란 무엇입니까IAM?를 IAM참조하세요.

내용

- Amazon S3 대상에 대한 AWS IoT FleetWise 액세스 권한 부여
- Amazon Timestream 대상에 대한 AWS IoT FleetWise 액세스 권한 부여

Amazon S3 대상에 대한 AWS IoT FleetWise 액세스 권한 부여

Amazon S3 대상을 사용하는 경우 는 S3 버킷에 차량 데이터를 AWS IoT FleetWise 전달하고 선택적으로 데이터 암호화에 소유한 AWS KMS 키를 사용할 수 있습니다. 오류 로깅이 활성화된 경우 는 CloudWatch 로그 그룹 및 스트림으로 데이터 전송 오류 AWS IoT FleetWise 도 전송합니다. 전송 스트림을 생성할 때는 IAM 역할이 있어야 합니다.

AWS IoT FleetWise 는 S3 대상에 대한 서비스 보안 주체와 함께 버킷 정책을 사용합니다. 버킷 정책 추가에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 콘솔을 사용하여 버킷 정책 추가를 참조하세요.

다음 액세스 정책을 사용하여 가 S3 버킷 AWS IoT FleetWise 에 액세스하도록 활성화합니다. S3 버킷을 소유하지 않은 경우 Amazon S3 작업 목록에 `s3:PutObjectAcl`을 추가합니다. 이렇게 하면 버킷 소유자가 에서 제공하는 객체에 대한 전체 액세스 권한을 부여합니다 AWS IoT FleetWise. 버킷의 객체 내 액세스 보호 방법에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 버킷 정책 예제를 참조하세요.

```
{
  "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"
        }
      }
```

```
    }
  ]
}
```

다음 버킷 정책은 AWS 리전의 계정에 있는 모든 캠페인에 적용됩니다.

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

S3 버킷에 KMS 키가 연결되어 있는 경우 키에 다음 정책이 필요합니다. 키 관리에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 AWS Key Management Service 키(-SSEKMS)를 사용하여 서버 측 암호화를 사용하여 데이터 보호를 참조하세요.

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

Important

버킷을 생성할 때 S3는 리소스 소유자에게 리소스에 대한 전체 제어 권한을 부여하는 기본 액세스 제어 목록(ACL)을 생성합니다. AWS IoT FleetWise 가 S3에 데이터를 전송할 수 없는 경우 S3 버킷ACL에서 를 비활성화해야 합니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 모든 새 버킷에 ACLs 대해 비활성화 및 객체 소유권 적용을 참조하세요.

Amazon Timestream 대상에 대한 AWS IoT FleetWise 액세스 권한 부여

Timestream 대상을 사용하는 경우 는 차량 데이터를 Timestream 테이블에 AWS IoT FleetWise 전달합니다. 가 Timestream으로 데이터를 AWS IoT FleetWise 전송할 수 있도록 IAM 역할에 정책을 연결해야 합니다.

콘솔을 사용하여 캠페인을 생성하는 경우 AWS IoT FleetWise 는 필요한 정책을 역할에 자동으로 연결합니다.

시작하기 전에 다음 사항에 유의하세요.

Important

- AWS IoT FleetWise Timestream 리소스를 생성할 때 동일한 AWS 리전을 사용해야 합니다. AWS 리전을 전환하는 경우 Timestream 리소스에 액세스하는 데 문제가 있을 수 있습니다.

- AWS IoT FleetWise 는 미국 동부(버지니아 북부)와 유럽(프랑크푸르트)에서 사용할 수 있습니다.
- 지원되는 리전의 전체 목록은 AWS 일반 참조에서 Timestream 엔드포인트 및 할당량을 참조하세요.

- Timestream 데이터베이스가 있어야 합니다. 튜토리알의 경우, Amazon Timestream 개발자 안내서의 데이터베이스 생성을 참조하세요.
- 지정된 Timestream 데이터베이스에 테이블을 생성해야 합니다. 튜토리알의 경우, Amazon Timestream 개발자 가이드의 테이블 생성을 참조하세요.

AWS CLI 를 사용하여 Timestream에 대한 신뢰 정책으로 IAM 역할을 생성할 수 있습니다. IAM 역할을 생성하려면 다음 명령을 실행합니다.

신뢰 정책으로 IAM 역할을 생성하려면

- Replace *TimestreamExecutionRole* 생성하려는 역할의 이름이 표시됩니다.
- Replace *trust-policy* 신뢰 정책이 포함된 JSON 파일을 포함합니다.

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

Timestream에 데이터를 쓸 수 있는 AWS IoT FleetWise 권한을 부여하는 권한 정책을 생성합니다. 서비스 역할 권한이 있는 정책을 만들려면 다음 명령을 실행합니다.

권한 정책 생성을 생성하려는 경우

- Replace *AWSIoTFleetwiseAccessTimestreamPermissionsPolicy* 생성하려는 정책의 이름이 표시됩니다.
- Replace *permissions-policy* 권한 정책이 포함된 JSON 파일의 이름을 사용합니다.

```
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": "*"
```

```
    }
  ]
}
```

권한 정책을 IAM 역할에 연결하려면

1. 출력에서 권한 정책의 Amazon 리소스 이름(ARN)을 복사합니다.
2. IAM 권한 정책을 IAM 역할에 연결하려면 다음 명령을 실행합니다.

   - Replace *permissions-policy-arn* 이전 단계에서 복사ARN한 를 사용합니다.
   - Replace *TimestreamExecutionRole* 생성한 IAM 역할의 이름이 표시됩니다.

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

자세한 내용은 IAM 사용 설명서의 AWS 리소스에 대한 액세스 관리를 참조하세요.

AWS IoT FleetWise용 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 AWS IoT FleetWise 리소스를 사용할 수 있는 인증(로그인) 및 권한 부여(권한 보유)를 받을 수 있는지 제어합니다. IAM 는 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 고객
- ID를 통한 인증
- 정책을 사용한 액세스 관리
- AWS IoT FleetWise 의 작동 방식 IAM
- AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제
- AWS IoT FleetWise 자격 증명 및 액세스 문제 해결

고객

AWS Identity and Access Management (IAM) 사용 방법은 AWS IoT FleetWise에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 - AWS IoT FleetWise 서비스를 사용하여 작업을 수행하는 경우 관리자는 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS IoT FleetWise 기능을 사용하여 작업을 수행하려면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS IoT FleetWise 의 기능에 액세스할 수 없는 경우 섹션을 참조하세요AWS IoT FleetWise 자격 증명 및 액세스 문제 해결.

서비스 관리자 - 회사에서 AWS IoT FleetWise 리소스를 담당하는 경우 AWS IoT FleetWise에 대한 전체 액세스 권한이 있을 수 있습니다. 서비스 사용자가 액세스해야 하는 AWS IoT FleetWise 기능과 리소스를 결정하는 것은 사용자의 작업입니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 의 기본 개념을 이해합니다IAM. 회사에서 AWS IoT FleetWiseIAM를 사용하는 방법에 대한 자세한 내용은 섹션을 참조하세요AWS IoT FleetWise 의 작동 방식 IAM.

IAM 관리자 - IAM 관리자인 경우 AWS IoT FleetWise 에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대한 세부 정보를 알고 싶을 수 있습니다. 에서 사용할 수 있는 예제 AWS IoT FleetWise 자격 증명 기반 정책을 보려면 섹션을 IAM참조하세요AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제.

ID를 통한 인증

인증은 자격 증명 AWS 으로 에 로그인하는 방법입니다. 로 AWS 계정 루트 사용자, IAM 사용자로 또는 IAM 역할을 수임하여 인증(에 로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 에 페더레이션 자격 증명 AWS 으로 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션 자격 증명으로 로그인하면 관리자가 이전에 IAM 역할을 사용하여 자격 증명 페더레이션을 설정했습니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 에 로그인하는 방법을 AWS 계정 AWS참조하세요.

AWS 프로그래밍 방식으로 에 액세스하는 경우 는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공하여 자격 증명을 사용하여 요청에 암호화 방식으로 서명합니다. AWS 도구를 사용

하지 않는 경우 직접 요청에 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS 서명 버전 4에서 API 요청을 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것이 AWS 좋습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다단계 인증 및 사용 설명서의 AWS 다단계 인증을 IAM 참조하세요IAM.

AWS 계정 루트 사용자

를 생성하면 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, , AWS Directory Service Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명 AWS 계정이 에 액세스하면 역할을 수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 및 애플리케이션에서 사용할 수 있도록 자체 자격 증명 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 한 사람 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내 자격 증명입니다. 가능한 경우 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신 임시 보안 인증 정보를 사용하는 것이 좋습니다. 그러나 IAM 사용자와 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례에 대한 액세스 키 정기적으로 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용

자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 라는 이름의 그룹을 지정IAMAdmins하고 해당 그룹에 IAM 리소스를 관리할 수 있는 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내 자격 증명입니다. IAM 사용자와 비슷하지만 특정 사람과는 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 역할(콘솔)로 전환할 AWS Management Console수 있습니다. IAM 또는 AWS API 작업을 호출 AWS CLI 하거나 사용자 지정 를 사용하여 역할을 수임할 수 있습니다URL. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자(페더레이션)에 대한 역할 생성을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 자격 증명이 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 의 역할과 상호 연관시킵니다IAM. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 특정 작업에 대해 일시적으로 다른 권한을 맡을 IAM 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 누군가(신뢰할 수 있는 보안 주체)가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.
- 교차 서비스 액세스 - 일부 는 다른 에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어 서비스에서 호출할 때 해당 서비스가 Amazon에서 애플리케이션을 실행EC2하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 에서 작업을 수행하면 보안 주체로 AWS간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수

행할 수 있습니다. FAS 는 를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용을 완료해야 하는 요청을 수신할 때만 수행됩니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 액세스 세션 전달을 참조하세요.

  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다IAM. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.
  - 서비스 연결 역할 - 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon에서 실행되는 애플리케이션 EC2 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 보안 인증을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장하는 것보다 좋습니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행 중인 프로그램이 임시 보안 인증을 가져올 수 있습니다. 자세한 내용은 IAM 사용 설명서EC2의 IAM 역할 사용을 참조하세요.

정책을 사용한 액세스 관리

정책을 AWS 생성하고 AWS 자격 증명 또는 리소스에 연결하여 의 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는 의 객체입니다. 는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 필요한 리소스에 대한 작업을 수행할 수 있는 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성할 수 있습니다. 그런 다음 관리자는 IAM 정책을 역할에 추가하고 사용자는 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI또는 에서 역할 정보를 가져올 수 있습니다 AWS API.

보안 인증 기반 정책

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책 중에서 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 이 포함될 수 있습니다 AWS 서비스.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책IAM에서는 의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACLs)

액세스 제어 목록(ACLs)은 리소스에 액세스할 수 있는 권한이 있는 보안 주체(계정 멤버, 사용자 또는 역할)를 제어합니다. ACLs 는 리소스 기반 정책과 유사하지만 JSON 정책 문서 형식을 사용하지는 않습니다.

Amazon S3 AWS WAF및 AmazonVPC은 를 지원하는 서비스의 예입니다ACLs. 에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 ACLs참조하세요.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책이 IAM엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻

는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 IAM엔터티에 대한 권한 경계를 참조하세요.

- 서비스 제어 정책(SCPs) - 의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책 SCPs입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유 AWS 계정 한 여러 을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직의 모든 기능을 활성화하면 서비스 제어 정책(SCPs)을 계정의 일부 또는 전체에 적용할 수 있습니다. 는 각 를 포함하여 멤버 계정의 엔터티에 대한 권한을 SCP 제한합니다 AWS 계정 루트 사용자. 조직 및 에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 SCPs참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. AWS 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS IoT FleetWise 의 작동 방식 IAM

IAM 를 사용하여 AWS IoT FleetWise 에 대한 액세스를 관리하기 전에 AWS IoT FleetWise에 사용할 수 있는 IAM 기능에 대해 알아봅니다.

IAM AWS IoT FleetWise와 함께 사용할 수 있는 기능

IAM 기능	AWS IoT FleetWise 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예

IAM 기능	AWS IoT FleetWise 지원
정책 리소스	예
정책 조건 키	예
ACLs	아니요
ABAC (정책의 태그)	부분
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	아니요
서비스 연결 역할	아니요

대부분의 IAM 기능에서 AWS IoT FleetWise 및 기타 AWS 서비스가 작동하는 방식을 자세히 알아보려면 IAM 사용 설명서의 에서 AWS 를 사용하는 서비스를 IAM 참조하세요.

AWS IoT FleetWise에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부된 작업 및 리소스와 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제

AWS IoT FleetWise 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제.

AWS IoT FleetWise 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 가 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려면 리소스 기반 정책의 보안 주체로 전체 계정 또는 다른 계정의 IAM엔터티를 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 에 있는 경우 신뢰할 수 AWS 계정있는 계정의 IAM 관리자는 보안 주체 엔터티(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 에서 크로스 계정 리소스 액세스를 IAM 참조하세요.

AWS IoT FleetWise에 대한 정책 작업

정책 작업 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업과 같은 몇 가지 예외가 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

AWS IoT FleetWise 작업 목록을 보려면 서비스 승인 참조의 AWS IoT FleetWise 에서 정의한 작업을 참조하세요.

AWS IoT FleetWise 의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
iotfleetwise
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

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

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, List라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

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

AWS IoT FleetWise 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제.

AWS IoT FleetWise용 정책 리소스

정책 리소스 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 객체를 지정합니다. 문장에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 가장 좋은 방법은 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정하는 것입니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

AWS IoT FleetWise 리소스 유형 및 해당 의 목록을 보려면 서비스 승인 참조의 AWS IoT FleetWise 에서 정의한 리소스를 ARNs참조하세요. 각 리소스ARN의 를 지정할 수 있는 작업을 알아보려면 AWS IoT FleetWise 에서 정의한 작업을 참조하세요.

AWS IoT FleetWise 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제.

AWS IoT FleetWise의 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 작업을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어 IAM 리소스에 사용자 IAM 이름으로 태그가 지정된 경우에만 사용자에게 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS IoT FleetWise 조건 키 목록을 보려면 서비스 승인 참조의 AWS IoT FleetWise 용 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면 AWS IoT FleetWise 에서 정의한 작업을 참조하세요.

AWS IoT FleetWise 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제.

AWS IoT FleetWise의 액세스 제어 목록(ACLs)

지원 ACLs: 아니요

액세스 제어 목록(ACLs)은 리소스에 액세스할 수 있는 권한이 있는 보안 주체(계정 멤버, 사용자 또는 역할)를 제어합니다. ACLs 는 리소스 기반 정책과 유사하지만 JSON 정책 문서 형식을 사용하지는 않습니다.

AWS IoT FleetWise를 사용한 속성 기반 액세스 제어(ABAC)

지원ABAC(정책의 태그): 부분

속성 기반 액세스 제어(ABAC)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그 라고 합니다. IAM 엔터티(사용자 또는 역할)와 많은 AWS 리소스에 태그를 연결할 수 있습니다. 엔터티 및 리소스에 태그를 지정하는 것은 의 첫 번째 단계입니다ABAC. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계합니다.

ABAC 는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로워지는 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 사용하여 권한 정의를 ABAC참조하세요. 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어 사용(ABAC)을 ABAC참조하세요.

> **Note**
>
> AWS IoT FleetWise `iam:PassRole`는 `CreateCampaign` API 작업에 필요한 만 지원합니다.

AWS IoT FleetWise에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명을 사용하여 로그인할 때 작동하지 AWS 서비스 않는 경우도 있습니다. 임시 자격 증명으로 AWS 서비스 작업하는 경우를 비롯한 자세한 내용은 IAM 사용 설명서의 AWS 서비스 에서 작업하는 IAM 섹션을 참조하세요.

사용자 이름 및 암호를 제외한 방법을 AWS Management Console 사용하여 에 로그인하는 경우 임시 보안 인증 정보를 사용합니다. 예를 들어 회사의 Single Sign-On(SSO) 링크를 AWS 사용하여 에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할(콘솔)로 전환을 참조하세요.

AWS CLI 또는 를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS API. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 AWS. AWS recommends에 액세스할 수 있습니다. 자세한 내용은 의 임시 보안 자격 증명을 IAM참조하세요.

AWS IoT FleetWise에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션 지원(FAS): 예

IAM 사용자 또는 역할을 사용하여 에서 작업을 수행하는 경우 보안 주체로 AWS간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS 는 를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용을 완료해야 하는 요청을 수신할 때만 수행됩니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 액세스 세션 전달을 참조하세요.

AWS IoT FleetWise의 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다IAM. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 AWS IoT FleetWise 기능이 중단될 수 있습니다. AWS IoT FleetWise 가 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

AWS IoT FleetWise의 서비스 연결 역할

서비스 링크 역할 지원: 아니요

서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 AWS 에서 작동하는 서비스를 참조하세요IAM. 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

AWS IoT FleetWise에 서비스 연결 역할 사용

AWS IoT FleetWise 는 AWS Identity and Access Management (IAM) 서비스 연결 역할 을 사용합니다. 서비스 연결 역할은 AWS IoT FleetWise에 직접 연결되는 고유한 유형의 IAM 역할입니다. 서비스 연결

역할은 AWS IoT FleetWise 에서 사전 정의하며 AWS IoT FleetWise 가 Amazon 에 지표를 보내는 데 필요한 권한을 포함합니다 CloudWatch. 자세한 내용은 Amazon을 통한 Monitor AWS IoT FleetWise CloudWatch 단원을 참조하십시오.

서비스 연결 역할은 필요한 권한을 수동으로 추가할 필요가 없으므로 AWS IoT FleetWise 를 더 빠르게 설정합니다. AWS IoT FleetWise 는 서비스 연결 역할의 권한을 정의하며, 달리 정의되지 않는 한 AWS IoT FleetWise 만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함됩니다. 이 권한 정책은 다른 IAM엔터티에 연결할 수 없습니다.

먼저 관련 리소스를 삭제한 후에만 서비스 연결 역할을 삭제할 수 있습니다. 이렇게 AWS 하면 리소스에 대한 액세스 권한을 실수로 제거할 수 없으므로 IoT FleetWise 리소스를 보호합니다.

서비스 연결 역할을 지원하는 다른 서비스에 대한 자세한 내용은 AWS 에서 작동하는 서비스를 IAM참조하고 서비스 연결 역할 열에서 Yes인 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes 링크를 선택합니다.

AWS IoT FleetWise에 대한 서비스 연결 역할 권한

AWS IoT FleetWise 는 AWSServiceRoleForIoTFleetWise IoT 에 대한 모든 out-of-the-box 권한에 사용되는 AWS 관리형 정책이라는 서비스 연결 역할을 사용합니다AWSIoT FleetWise.

AWSServiceRoleForIoTFleetWise 서비스 연결 역할은 다음 서비스를 신뢰하여 역할을 수임합니다.

- `IoTFleetWise`

라는 역할 권한 정책은 AWS IoT FleetWise 가 지정된 리소스에 대해 다음 작업을 완료할 수 있도록 AWSIoTFleetwiseServiceRolePolicy 허용합니다.

- 작업: `cloudwatch:PutMetricData` 리소스의 경우: `*`

IAM 엔터티(예: 사용자, 그룹 또는 역할)가 서비스 연결 역할을 생성, 편집 또는 삭제할 수 있도록 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

AWS IoT FleetWise에 대한 서비스 연결 역할 생성

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS IoT FleetWise 콘솔, AWS CLI또는 에 계정을 등록하면 AWS API AWS IoT FleetWise 가 서비스 연결 역할을 생성합니다. 자세한 내용은 AWS IoT FleetWise 설정 구성 단원을 참조하십시오.

AWS IoT FleetWise 에서 서비스 연결 역할 생성(콘솔)

서비스 링크 역할은 수동으로 생성할 필요가 없습니다. AWS IoT FleetWise 콘솔, AWS CLI또는 에 계정을 등록하면 AWS API AWS IoT FleetWise 가 서비스 연결 역할을 생성합니다.

AWS IoT FleetWise에 대한 서비스 연결 역할 편집

AWS IoT FleetWise 에서는 AWSServiceRoleForIoTFleetWise 서비스 연결 역할을 편집할 수 없습니다. 생성한 서비스 연결 역할을 여러 엔터티가 참조할 수 있기 때문에 역할의 이름을 변경할 수 없습니다. 그러나 를 사용하여 역할에 대한 설명을 편집할 수 있습니다IAM. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

서비스 연결 역할을 정리

IAM 를 사용하여 서비스 연결 역할을 삭제하려면 먼저 역할에서 사용하는 리소스를 삭제해야 합니다.

> **Note**
>
> 리소스를 삭제하려고 할 때 AWS IoT FleetWise 가 역할을 사용하는 경우 삭제에 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요. 콘솔 AWS CLI또는 AWS 를 service-linked-role 통해 를 삭제하는 방법을 알아보려면 IAM 사용 설명서의 서비스 연결 역할 사용을 API참조하세요.

이 서비스 연결 역할을 삭제한 다음 다시 생성해야 하는 경우 AWS IoT FleetWise에 계정을 등록할 수 있습니다. AWS 그런 다음 IoT FleetWise 가 서비스 연결 역할을 다시 생성합니다.

AWS IoT FleetWise에 대한 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 AWS IoT FleetWise 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 를 사용하여 작업을 수행할 수 없습니다 AWS API. 사용자에게 필요한 리소스에 대한 작업을 수행할 수 있는 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성할 수 있습니다. 그런 다음 관리자는 IAM 정책을 역할에 추가하고 사용자는 역할을 수임할 수 있습니다.

이러한 예제 정책 문서를 사용하여 IAM 자격 증명 기반 JSON 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 형식 등 AWS IoT FleetWise에서 정의한 작업 및 리소스 유형에 ARNs 대한 자세한 내용은 서비스 승인 참조의 AWS IoT FleetWise 에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- AWS IoT FleetWise 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- Amazon Timestream의 리소스에 액세스

정책 모범 사례

자격 증명 기반 정책은 계정에서 누군가 AWS IoT FleetWise 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 관리AWS 형 정책 또는 AWS 작업 함수에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 - IAM 정책으로 권한을 설정할 때 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. 를 사용하여 권한을 적용하는 IAM 방법에 대한 자세한 내용은 IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 정책 조건을 작성하여 를 사용하여 모든 요청을 전송하도록 지정할 수 있습니다SSL. AWS 서비스와 같은 특정 를 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 AWS CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. IAM Access Analyzer는 정책이 정책 언어(JSON) 및 IAM 모범 사례를 준수하도록 새 정책 및 기존 IAM 정책을 검증합니다. IAM Access Analyzer는 안전하고 기능적인 정책을 작성하는 데 도움이 되는 100개 이상의 정책 확인 및 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer를 사용한 정책 검증을 참조하세요.
- 다중 인증 필요(MFA) - 에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 MFA 위해 를 AWS 계정켭니다. API 작업을 호출할 MFA 때 를 요구하려면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 를 사용한 보안 API 액세스를 MFA 참조하세요.

의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 의 보안 모범 사례를 IAM IAM참조하세요.

AWS IoT FleetWise 콘솔 사용

AWS IoT FleetWise 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 의 AWS IoT FleetWise 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 에만 전화를 거는 사용자에 대해서는 최소 콘솔 권한을 허용할 필요가 없습니다 AWS API. 대신 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 허용합니다.

사용자와 역할이 AWS IoT FleetWise 콘솔을 계속 사용할 수 있도록 하려면 엔터티에 AWS IoT FleetWise `ConsoleAccess` 또는 `ReadOnly` AWS 관리형 정책도 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제에서는 IAM 사용자가 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함되어 있습니다 AWS API.

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

Amazon Timestream의 리소스에 액세스

AWS IoT FleetWise를 사용하기 전에 AWS 계정, IAM및 Amazon Timestream 리소스를 등록하여 AWS 클라우드 사용자를 대신하여 차량 데이터를 에 전송할 수 있는 AWS IoT FleetWise 권한을 부여해야 합니다. 등록하려면 다음이 필요합니다.

- Amazon Timestream 데이터베이스.
- 지정된 Amazon Timestream 데이터베이스에서 생성된 테이블.
- AWS IoT FleetWise 가 Amazon Timestream으로 데이터를 전송할 수 있도록 허용하는 IAM 역할입니다.

절차 및 예제 정책을 포함한 자세한 내용은 구성 설정을 참조하세요.

AWS IoT FleetWise 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS IoT FleetWise 및 작업 시 발생할 수 있는 일반적인 문제를 진단하고 해결할 수 있습니다IAM.

주제

- AWS IoT FleetWise에서 작업을 수행할 권한이 없음
- iam을 수행할 권한이 없습니다.PassRole
- 외부의 사람이 my AWS IoT FleetWise 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

AWS IoT FleetWise에서 작업을 수행할 권한이 없음

에 작업을 수행할 권한이 없다고 AWS Management Console 표시되면 관리자에게 문의하여 지원을 받아야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *myVehicle* 리소스에 대한 세부 정보를 보려고 하지만 iotfleetwise:*GetVehicleStatus* 권한이 없는 경우에 발생합니다.

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

이 경우 Mateo는 *myVehicle* 작업을 사용하여 iotfleetwise:*GetVehicleStatus* 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

iam을 수행할 권한이 없습니다.PassRole

iam:PassRole 작업을 수행할 권한이 없다는 오류가 발생하면 역할을 AWS IoT FleetWise에 전달할 수 있도록 정책을 업데이트해야 합니다.

일부는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있도록 AWS 서비스 허용합니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 이름이 인 IAM 사용자가 콘솔을 사용하여 AWS IoT FleetWise에서 작업을 수행하려고 marymajor 할 때 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부의 사람이 my AWS IoT FleetWise 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제

어 목록(ACLs)을 지원하는 서비스의 경우 이러한 정책을 사용하여 사용자에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- AWS IoT FleetWise 가 이러한 기능을 지원하는지 알아보려면 섹션을 참조하세요AWS IoT FleetWise 의 작동 방식 IAM.
- 소유 AWS 계정 한 의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유 AWS 계정 한 다른 의 IAM 사용자에게 액세스 권한 제공을 참조하세요.
- 타사에 리소스에 대한 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 소유 에 대한 액세스 제공을 AWS 계정참조하세요.
- 자격 증명 페더레이션을 통해 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부 인증 사용자(자격 증명 페더레이션)에 대한 액세스 제공을 참조하세요.
- 교차 계정 액세스를 위한 역할 및 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.

AWS IoT FleetWise에 대한 규정 준수 검증

> **Note**
>
> AWS IoT FleetWise는 규정 AWS 준수 프로그램의 범위에 속하지 않습니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 의 보고서 다운로드 AWS Artifact.

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 AWS 서비스 결정됩니다. 는 규정 준수를 돕기 위해 다음 리소스를 AWS 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 - 이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 에 기준 환경을 배포 AWS 하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 아키텍처 - 이 백서에서는 기업이 AWS 를 사용하여 HIPAA적격 애플리케이션을 생성하는 방법을 설명합니다.

> **Note**
>
> 모두 HIPAA 적합한 AWS 서비스 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하세요.

- AWS 규정 준수 리소스 - 이 워크북 및 가이드 모음은 해당 산업 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드는 여러 프레임워크(미국 국립표준기술연구소(), 결제카드 산업보안표준위원회(NIST), PCI국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례를 요약합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 - 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - AWS 서비스 에서 보안 상태를 포괄적으로 볼 수 있습니다 AWS. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 의심스러운 활동 및 악의적인 활동이 있는지 환경을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. 는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족DSS하여 PCI 와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 GuardDuty 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험과 규정 및 업계 표준 준수를 관리하는 방법을 간소화할 수 있습니다.

AWS IoT FleetWise의 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다. 리전은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며, 이러한 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크를 통해 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

> **Note**
>
> AWS IoT FleetWise 에서 처리하는 데이터는 Amazon Timestream 데이터베이스에 저장됩니다. Timestream은 다른 AWS 가용 영역 또는 리전으로의 백업을 지원합니다. 그러나

Timestream을 사용하여 자체 애플리케이션을 작성하여 데이터를 SDK 쿼리하고 원하는 대상에 저장할 수 있습니다.
Amazon Timestream 에 대한 자세한 내용은 Amazon Timestream 개발자 가이드를 참조하세요.

AWS IoT FleetWise의 인프라 보안

관리형 서비스인 AWS IoT FleetWise 는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안 섹션을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 AWS IoT FleetWise 에 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS). TLS 1.2가 필요하며 TLS 1.3을 권장합니다.
- DHE (Ephemeral Diffie-HellmanPFS) 또는 (Elliptic Curve Ephemeral Diffie-Hellman)과 같은 완벽한 순방향 보안ECDHE()이 포함된 Cipher 제품군입니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 IAM 보안 주체와 연결된 보안 액세스 키를 사용하여 요청에 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

모든 네트워크 위치에서 이러한 API 작업을 호출할 수 있지만 AWS IoT FleetWise 는 소스 IP 주소에 따른 제한을 포함할 수 있는 리소스 기반 액세스 정책을 지원합니다. AWS IoT FleetWise 정책을 사용하여 특정 Amazon Virtual Private Cloud(AmazonVPC) 엔드포인트 또는 특정 의 액세스를 제어할 수도 있습니다VPCs. 효과적으로, 이는 주어진 AWS IoT FleetWise 리소스에 대한 네트워크 액세스를 AWS 네트워크 VPC 내의 특정 에서만 격리합니다.

주제

- 인터페이스 VPC 엔드포인트를 통해 AWS IoT FleetWise 에 연결

인터페이스 VPC 엔드포인트를 통해 AWS IoT FleetWise 에 연결

인터넷을 통해 연결하는 VPC대신 Virtual Private Cloud(AWS PrivateLink)의 인터페이스 VPC 엔드포인트()를 사용하여 AWS IoT FleetWise 에 직접 연결할 수 있습니다. 인터페이스 VPC 엔드포인트를 사용하면 VPC 및 AWS IoT FleetWise 간의 통신이 전적으로 AWS 네트워크 내에서 수행됩니다. 각 VPC 엔드포인트는 VPC 서브넷에 프라이빗 IP 주소가 있는 하나 이상의 탄력적 네트워크 인터페이스(ENIs)로 표시됩니다.

인터페이스 VPC 엔드포인트는 인터넷 게이트웨이, NAT 디바이스, VPN 연결 또는 연결 없이 를 AWS IoT FleetWise 에 VPC 직접 AWS Direct Connect 연결합니다. 의 인스턴스는 AWS IoT FleetWise 와 통신하는 데 퍼블릭 IP 주소가 필요하지 VPC 않습니다API.

를 통해 AWS IoT FleetWise 를 사용하려면 에 있는 인스턴스에서 VPC 연결하거나 AWS Virtual Private Network (VPN) 또는 를 사용하여 프라이빗 네트워크를 VPC에 연결해야 VPC합니다 AWS Direct Connect. Amazon 에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 VPN 연결을 VPN참조하세요. 에 대한 자세한 내용은 AWS Direct Connect 사용 설명서의 연결 생성을 AWS Direct Connect참조하세요.

콘솔 또는 AWS Command Line Interface (AWS CLI) 명령을 사용하여 AWS IoT FleetWise 에 연결할 인터페이스 VPC 엔드포인트를 AWS 생성할 수 있습니다. 자세한 내용은 인터페이스 엔드포인트 생성을 참조하세요.

인터페이스 VPC 엔드포인트를 생성한 후 엔드포인트에 대한 프라이빗 DNS 호스트 이름을 활성화하면 default AWS IoT FleetWise 엔드포인트가 VPC 엔드포인트로 확인됩니다. AWS IoT FleetWise 의 기본 서비스 이름 엔드포인트는 다음 형식입니다.

```
iotfleetwise.Region.amazonaws.com
```

프라이빗 DNS 호스트 이름을 활성화하지 않으면 Amazon은 다음 형식으로 사용할 수 있는 DNS 엔드포인트 이름을 VPC 제공합니다.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 VPC 엔드포인트(AWS PrivateLink)를 참조하세요.

AWS IoT FleetWise 는 내 모든 API 작업에 대한 호출을 지원합니다VPC.

VPC 엔드포인트에 VPC 엔드포인트 정책을 연결하여 IAM 보안 주체에 대한 액세스를 제어할 수 있습니다. 또한 보안 그룹을 VPC 엔드포인트와 연결하여 다양한 IP 주소와 같은 네트워크 트래픽의 오리진

및 대상을 기반으로 인바운드 및 아웃바운드 액세스를 제어할 수 있습니다. 자세한 내용은 VPC 엔드포인트를 사용하여 서비스에 대한 액세스 제어를 참조하세요.

AWS IoT FleetWise에 대한 VPC 엔드포인트 정책 생성

Amazon VPC 엔드포인트 for AWS IoT FleetWise 에 대한 정책을 생성하여 다음을 지정할 수 있습니다.

- 작업을 수행할 수 있거나 수행할 수 없는 보안 주체
- 수행 가능 작업 또는 수행 불가 작업

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 사용하여 서비스에 대한 액세스 제어를 참조하세요.

Example - 지정된 AWS 계정의 모든 액세스를 거부하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책이 AWS 계정을 거부합니다.*123456789012* 엔드포인트를 사용하는 모든 API 통화입니다.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example - 지정된 IAM 보안 주체(사용자)에 대한 VPC 액세스만 허용하는 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 사용자에게만 전체 액세스를 허용합니다.*lijuan* AWS 계정 내 *123456789012*. 엔드포인트에 대한 다른 모든 IAM 보안 주체의 액세스를 거부합니다.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
                ]
            }
        }]
}
```

Example - AWS IoT FleetWise 작업에 대한 VPC 엔드포인트 정책

다음은 AWS IoT FleetWise에 대한 엔드포인트 정책의 예입니다. 엔드포인트에 연결되면 이 정책은 IAM 사용자에게 나열된 AWS IoT FleetWise 작업에 대한 액세스 권한을 부여합니다.*fleetWise* 의 AWS 계정 *123456789012*.

```
{
    "Statement": [
        {
             "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/fleetWise"
                },
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "iotfleetwise:ListFleets",
                "iotfleetwise:ListCampaigns",
                "iotfleetwise:CreateVehicle",
            ]
           }
    ]
}
```

AWS IoT FleetWise 의 구성 및 취약성 분석

IoT 환경은 다양한 기능을 수행하고 장기적으로 사용되며 지리적으로 분산된 다수의 장치로 구성될 수 있습니다. 이러한 특성으로 인해 장치 설정이 복잡해지고 오류가 발생하기 쉬워집니다. 디바이스의 컴퓨팅 파워, 메모리 및 스토리지 기능이 한정된 경우가 많으므로 이에 따라 디바이스 자체에서 암호화 및 다른 형태의 보안을 사용하는 데 제약이 있습니다. 디바이스는 종종 취약점이 알려진 소프트웨어를 사용합니다. 이러한 요인으로 인해 해커의 매력적인 대상인 IoT용 데이터를 수집하는 차량을 비롯한 AWS IoT FleetWise 디바이스가 지속적으로 보안을 유지하기가 어렵습니다.

구성 및 IT 제어는 고객과 AWS 고객 간의 공동 책임입니다. 자세한 내용은 AWS 공동 책임 모델 을 참조하세요.

AWS IoT FleetWise에 대한 보안 모범 사례

AWS IoT FleetWise 는 자체 보안 정책을 개발하고 구현할 때 고려해야 할 여러 보안 기능을 제공합니다. 다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용하십시오.

의 보안에 대한 자세한 내용은 AWS IoT 개발자 안내서의 의 보안 모범 사례를 AWS IoT Core AWS IoT 참조하세요.

가능한 최소 권한 부여

IAM 역할에서 최소 권한 세트를 사용하여 최소 권한 원칙을 따릅니다. IAM 정책의 `Action` 및 `Resource` 속성에 와일드*카드 사용을 제한합니다. 대신, 가능한 경우 한정된 작업과 리소스를 선언합니다. 최소 권한 및 기타 정책 모범 사례에 대한 자세한 내용은 the section called “정책 모범 사례”을 참조하세요.

민감한 정보를 기록하지 않음

자격 증명 및 기타 개인 식별 정보()의 로깅을 방지해야 합니다PII. 다음과 같은 보호 장치를 구현하는 것이 좋습니다.

- 디바이스 이름에 민감한 정보를 사용하지 않습니다.
- 캠페인, 디코더 매니페스트, 차량 모델 및 신호 카탈로그의 이름, 차량 및 플릿의 이름 등 IDs AWS IoT FleetWise 리소스IDs의 이름 및 에 민감한 정보를 사용하지 마세요.

API 통화 기록을 보는 AWS CloudTrail 데 사용

보안 분석 및 운영 문제 해결을 위해 계정에서 수행된 AWS IoT FleetWise API 호출 기록을 볼 수 있습니다. 계정에서 수행된 AWS IoT FleetWise API 통화 기록을 수신하려면 CloudTrail 에서 를 켜면 됩니다 AWS Management Console. 자세한 내용은 the section called “CloudTrail 로그” 단원을 참조하십시오.

장치의 시계를 동기화 상태로 유지

장치에서는 정확한 시간을 유지하는 것이 중요합니다. X.509 인증서에는 만료 날짜와 시간이 있습니다. 장치의 시계는 서버 인증서가 여전히 유효한지 확인하는 데 사용됩니다. 장치 시계는 시간이 지나 드리프트 상태가 되거나 배터리가 방전될 수 있습니다.

자세한 내용은 AWS IoT Core 개발자 가이드의 디바이스 시계를 동기화하기 모범 사례를 참조하세요.

Monitor AWS IoT FleetWise

모니터링은 AWS IoT FleetWise 및 기타 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 는 AWS IoT FleetWise를 감시하고, 문제가 발생하면 보고하고, 적절한 경우 자동 조치를 취할 수 있는 다음 모니터링 도구를 AWS 제공합니다.

- Amazon CloudWatch은 AWS 리소스와 실행 중인 애플리케이션을 AWS 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요한 경우 새 인스턴스를 자동으로 시작할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서 섹션을 참조하세요.
- Amazon CloudWatch Logs는 Amazon EC2 인스턴스 및 기타 소스에서 로그 파일을 모니터링, 저장 CloudTrail및 액세스하는 데 사용할 수 있습니다. CloudWatch Logs는 로그 파일의 정보를 모니터링하고 특정 임계값이 충족되면 사용자에게 알릴 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서 를 참조하세요.
- AWS CloudTrail 는 에 의해 또는 를 대신하여 수행된 API 통화 및 관련 이벤트를 캡처합니다 AWS 계정. 그러고 나서 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 라는 사용자 및 계정 AWS, 호출이 수행된 원본 IP 주소, 호출이 발생한 시점을 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

Amazon을 통한 Monitor AWS IoT FleetWise CloudWatch

Amazon CloudWatch 지표는 AWS 리소스와 리소스의 성능을 모니터링하는 방법입니다. AWS IoT FleetWise 는 지표를 로 전송합니다 CloudWatch. AWS Management Console, AWS CLI또는 를 사용하여 AWS IoT FleetWise 가 API에 보내는 지표를 나열할 수 있습니다 CloudWatch. 자세한 내용은 Amazon CloudWatch 사용 설명서 섹션을 참조하세요.

> **Important**
>
> AWS IoT FleetWise 가 에 지표를 보낼 수 있도록 설정을 구성해야 합니다 CloudWatch. 자세한 내용은 AWS IoT FleetWise 설정 구성 단원을 참조하십시오.

`AWS/IoTFleetWise` 네임스페이스에 포함된 지표는 다음과 같습니다.

신호 지표

지표	설명
IllegalMessageFromEdge	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지가 필요한 형식과 일치하지 않습니다. 단위: 개 차원: VehicleName 유효 통계: Sum
MessageThrottled	차량에서 AWS IoT FleetWise 로 전송된 메시지가 제한되었습니다. 이는 현재 지역에서 이 계정의 서비스 한도를 초과했기 때문입니다. 단위: 개 차원: VehicleName 유효 통계: Sum
ModelingError	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지에는 차량 모델에 대해 검증하지 못하는 신호가 포함되어 있습니다. 단위: 개 차원: ModelManifestName 유효 통계: Sum
DecodingError	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지에는 차량의 디코더 매니페스트에 대해 디코더에 실패하는 신호가 포함되어 있습니다. 단위: 개 차원: DecoderName

지표	설명
	유효 통계: Sum

캠페인 지표

지표	설명
VehicleNotFound	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 차량을 알 수 없습니다. 단위: 개 차원: VehicleName 유효 통계: Sum
CampaignInvalid	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 캠페인이 유효하지 않습니다. 단위: 개 차원: CampaignName 유효 통계: Sum
CampaignNotFound	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 캠페인을 알 수 없습니다. 단위: 개 차원: CampaignName 유효 통계: Sum

캠페인 데이터 대상 지표

지표	설명
TimestreamWriteError	AWS IoT FleetWise 가 차량의 메시지를 Amazon Timestream 테이블에 쓸 수 없습니다. 단위: 개 차원: DatabaseName, TableName 유효 통계: Sum
S3WriteError	AWS IoT FleetWise 가 차량에서 Amazon Simple Storage Service(Amazon S3) 버킷으로 메시지를 작성할 수 없습니다. 단위: 개 차원: BucketName 유효 통계: Sum
S3ReadError	AWS IoT FleetWise 가 Amazon Simple Storage Service(Amazon S3) 버킷의 차량에서 객체 키를 읽을 수 없습니다. 단위: 개 차원: BucketName 유효 통계: Sum

고객 관리형 AWS KMS 키 지표

지표	설명
KMSKeyAccessDenied	AWS KMS 키 액세스 거부 오류로 인해AWS IoT FleetWise 가 차량의 메시지를 Timestream 테이블 또는 Amazon S3 버킷에 쓸 수 없습니다.

지표	설명
	단위: 개 차원: KMSKeyId 유효 통계: Sum

Amazon CloudWatch Logs를 사용한 Monitor AWS IoT FleetWise

Amazon CloudWatch Logs는 리소스에서 발생하는 이벤트를 모니터링하고 문제가 있는 경우 사용자에게 알립니다. 알림을 받으면 로그 파일에 액세스하여 특정 이벤트에 대한 정보를 얻을 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서 를 참조하세요.

CloudWatch 콘솔에서 AWS IoT FleetWise 로그 보기

> **Important**
>
> CloudWatch 콘솔에서 AWS IoT FleetWise 로그 그룹을 보려면 먼저 다음 사항이 맞는지 확인하세요.
>
> - AWS IoT FleetWise에서 로깅을 활성화했습니다. 로깅에 대한 자세한 내용은 AWS IoT FleetWise 로깅 구성 섹션을 참조하세요.
> - AWS IoT 작업에서 작성한 로그 항목이 이미 있습니다.

CloudWatch 콘솔에서 AWS IoT FleetWise 로그를 보려면

1. CloudWatch 콘솔 을 엽니다.
2. 탐색 창에서 로그, 로그 그룹을 선택합니다.
3. 로그 그룹을 선택합니다.
4. 로그 그룹 검색을 선택합니다. 계정에 대해 생성된 로그 이벤트의 전체 목록이 표시됩니다.
5. 확장 아이콘을 선택하면 개별 스트림을 살펴보고 로그 수준이 ERROR와 같은 모든 로그를 찾을 수 있습니다.

   이벤트 필터링 검색 상자에 쿼리를 입력할 수도 있습니다. 예를 들면, 다음 쿼리를 수행할 수 있습니다.

```
{ $.logLevel = "ERROR" }
```

필터 표현식 생성에 대한 자세한 내용은 Amazon Logs 사용 설명서의 필터 및 패턴 구문을 참조하세요. CloudWatch

Example 로그 항목

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

신호 이벤트 유형

이벤트 유형	설명
MODELING_ERROR	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지에는 차량 모델에 대해 검증하지 못하는 신호가 포함되어 있습니다. 속성: vehicleName, campainName, signalCatalogName, signalId, signalValue, signalValueRangeMin, signalValueRangeMax, modelManifestName
ILLEGAL_MESSAGE_FROM_EDGE	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지가 필요한 형식과 일치하지 않습니다. 속성: vehicleName, campaignName, signalCatalogName

이벤트 유형	설명
DECODING_ERROR	차량에서 전송되고 AWS IoT FleetWise 에서 수신한 메시지에는 차량의 디코더 매니페스트에 대해 디코더에 실패하는 신호가 포함되어 있습니다. 속성: campaignName, signalCatalogName, decoderManifestName,(선택 사항) signalName, (선택 사항) s3URI

캠페인 이벤트 유형

이벤트 유형	설명
VEHICLE_NOT_FOUND	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 차량을 알 수 없습니다. 속성: vehicleName, campaignName
CAMPAIGN_NOT_FOUND	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 캠페인을 알 수 없습니다. 속성: vehicleName (선택 사항), campaignName
CAMPAIGN_INVALID	차량에서 전송되고 AWS IoT FleetWise에서 수신한 메시지로, 캠페인이 유효하지 않습니다. 속성: vehicleName (선택 사항), campaignName

캠페인 데이터 대상 이벤트 유형

이벤트 유형	설명
TIMESTREAM_WRITE_ERROR	AWS IoT FleetWise 가 차량의 메시지를 Amazon Timestream 테이블에 쓸 수 없습니다.

이벤트 유형	설명
	속성: vehicleName, campaignName, timestreamDatabaseName, timestreamTableName
S3_WRITE_ERROR	AWS IoT FleetWise 가 차량에서 Amazon Simple Storage Service(Amazon S3) 버킷으로 메시지를 작성할 수 없습니다. 속성: campaignName, destinationName
S3_READ_ERROR	AWS IoT FleetWise 가 Amazon Simple Storage Service(Amazon S3) 버킷의 차량에서 객체 키를 읽을 수 없습니다. 속성: campaignName, destinationName

고객 관리형 AWS KMS 키 이벤트 유형

이벤트 유형	설명
KMS_KEY_ACCESS_DENIED	AWS KMS 키 액세스 거부 오류로 인해AWS IoT FleetWise 가 차량의 메시지를 Timestream 테이블 또는 Amazon S3 버킷에 쓸 수 없습니다.

속성

모든 CloudWatch 로그 항목에는 다음과 같은 속성이 포함됩니다.

accountId

AWS 계정 ID.

eventType

로그가 작성된 이벤트 유형입니다. 이벤트 유형의 값은 로그 항목을 생성한 이벤트에 따라 다릅니다. 각 로그 항목 설명에는 해당 로그 항목의 `eventType` 값이 포함됩니다.

logLevel

사용 중인 로그 수준. 자세한 내용은 AWS IoT Core 개발자 가이드의 로그 레벨 섹션을 참조하세요.

message

로그에 대한 특정 세부 정보가 들어 있습니다.

타임스탬프

AWS IoT FleetWise 가 로그를 처리한 시점의 에폭 밀리초 타임스탬프입니다.

선택적 속성

CloudWatch 로그 항목에는 에 따라 eventType다음과 같은 속성이 선택적으로 포함됩니다.

decoderManifestName

신호가 포함된 디코더 매니페스트의 이름입니다.

destinationName

차량 데이터의 대상의 이름입니다. Amazon S3 버킷 이름입니다.

campaignName

캠페인의 이름입니다.

signalCatalogName

신호가 포함된 신호 카탈로그의 이름입니다.

signalId

오류 신호의 ID입니다.

signalIds

오류 신호 목록입니다IDs.

signalName

신호의 이름입니다.

signalTimestampEpochMs

오류 신호의 타임스탬프.

signalValue

오류 신호의 값입니다.

signalValueRange최대

오류 신호의 최대 범위입니다.

signalValueRange최소

오류 신호의 최소 범위.

s3URI

차량 메시지에 있는 Amazon Ion 파일의 Amazon S3 고유 식별자입니다.

timestreamDatabaseName

Timestream 데이터베이스의 이름입니다.

timestreamTableName

Timestream 테이블의 이름입니다.

vehicleName

차량 모델의 이름입니다.

AWS IoT FleetWise 로깅 구성

AWS IoT FleetWise 로그 데이터를 CloudWatch 로그 그룹으로 전송할 수 있습니다. CloudWatch AWS IoT FleetWise 가 차량의 메시지를 처리하지 못하는 경우 로그를 통해 가시성을 확보할 수 있습니다. 예를 들어 잘못된 구성이나 기타 클라이언트 오류로 인해 이러한 문제가 발생할 수 있습니다. 오류가 발생하면 알림을 받게 되므로 문제를 식별하고 완화할 수 있습니다.

로 로그를 전송하려면 먼저 CloudWatch 로그 그룹을 생성 CloudWatch해야 합니다. AWS IoT FleetWise 에 사용한 것과 동일한 계정과 리전으로 로그 그룹을 구성합니다. AWS IoT FleetWise 에서 로깅을 활성화하면 로그 그룹 이름을 입력합니다. 로깅이 활성화된 후 AWS IoT FleetWise 는 로그 스트림의 CloudWatch 로그 그룹에 로그를 전달합니다.

CloudWatch 콘솔에서 AWS IoT FleetWise 에서 전송된 로그 데이터를 볼 수 있습니다. CloudWatch 로그 그룹 구성 및 로그 데이터 보기에 대한 자세한 내용은 로그 그룹 작업을 참조하세요.

로그를 게시할 수 있는 권한 CloudWatch

CloudWatch 로그 그룹에 대한 로깅을 구성하려면 이 섹션에 설명된 권한 설정이 필요합니다. 권한 관리에 대한 자세한 내용은 IAM 사용 설명서의 AWS 리소스에 대한 액세스 관리를 참조하세요.

이러한 권한을 사용하면 로깅 구성을 변경하고, 에 대한 로그 전송을 구성하고 CloudWatch, 로그 그룹에 대한 정보를 검색할 수 있습니다.

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

모든 AWS 리소스에서 작업이 허용되는 경우 정책에 로 "Resource" 설정되어 표시됩니다"*". 즉, 각 작업이 를 지원하는 모든 AWS 리소스에 대해 작업이 허용됩니다.

in AWS IoT FleetWise 로깅 구성(콘솔)

이 섹션에서는 AWS IoT FleetWise 콘솔을 사용하여 로깅을 구성하는 방법을 설명합니다.

AWS IoT FleetWise 콘솔을 사용하여 로깅을 구성하려면

1. AWS IoT FleetWise 콘솔 을 엽니다.
2. 왼쪽 창에서 설정을 선택합니다.
3. 설정페이지의 로깅 섹션에서 편집을 선택합니다.
4. CloudWatch 로깅 섹션에서 로그 그룹 을 입력합니다.
5. 변경 사항을 저장하려면 제출을 선택합니다.

로깅을 활성화한 후 CloudWatch 콘솔 에서 로그 데이터를 볼 수 있습니다.

AWS IoT FleetWise 에서 기본 로깅 구성(CLI)

이 섹션에서는 를 사용하여 AWS IoT FleetWise 에 대한 로깅을 구성하는 방법을 설명합니다CLI.

여기에 표시된 CLI 명령에 해당하는 의 메 AWS API서드를 사용하여 를 API 사용하여 이 절차를 수행할 수도 있습니다. GetLoggingOptions API 작업을 사용하여 현재 구성을 가져오고 PutLoggingOptions API 작업을 사용하여 구성을 수정할 수 있습니다.

CLI 를 사용하여 AWS IoT FleetWise에 대한 로깅을 구성하려면

1. 계정에 대한 로깅 옵션을 설정하려면, get-logging-options 명령을 사용합니다.

```
aws iotfleetwise get-logging-options
```

2. 로깅을 활성화하려면 put-logging-options 명령을 사용합니다.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
 logType=ERROR,logGroupName=MyLogGroup
```

여기서 각 항목은 다음과 같습니다.

logType

로그로 데이터를 전송할 CloudWatch 로그 유형입니다. 로깅을 비활성화하려면 값을 OFF로 변경합니다.

logGroupName

작업이 데이터를 보내는 CloudWatch 로그 그룹입니다. AWS IoT FleetWise 에 대한 로깅을 활성화하기 전에 로그 그룹 이름을 생성해야 합니다.

로깅을 활성화한 후 를 사용하여 로그 항목 검색을 AWS CLI 참조하세요.

를 사용한 Log AWS IoT FleetWise API 호출 AWS CloudTrail

AWS IoT FleetWise 는 사용자 AWS CloudTrail, 역할 또는 서비스에서 수행한 작업 기록을 제공하는 AWS 서비스인 와 통합됩니다. AWS IoT FleetWise는 이벤트로 AWS IoT FleetWise 에 대한 모든 API 호출을 CloudTrail 캡처합니다. 캡처된 호출에는 AWS IoT FleetWise 콘솔의 호출과 AWS IoT FleetWise API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하는 경우 AWS IoT FleetWise용 이벤트를 포함하여 Amazon S3 버킷으로 이벤트를 지속적으로 전송할 CloudTrail 수 있습니다. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록 에서 최신 이벤트를 볼 수 있습니다. 에서 수집한 정보를 사용하여 AWS IoT FleetWise에 수행된 요청 CloudTrail, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

에 대한 자세한 내용은 AWS CloudTrail 사용 설명서 섹션을 CloudTrail참조하세요.

의AWS IoT FleetWise 정보 CloudTrail

CloudTrail 는 AWS 계정을 생성할 때 계정에서 활성화됩니다. AWS IoT FleetWise에서 활동이 발생하면 해당 활동은 CloudTrail 이벤트 기록 의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 이벤트 기록을 사용하여 CloudTrail 이벤트 보기를 참조하세요.

AWS IoT FleetWise용 이벤트를 포함하여 AWS 계정의 이벤트에 대한 지속적인 기록을 위해 추적을 생성합니다. 추적을 사용하면 CloudTrail 가 Amazon S3 버킷에 로그 파일을 전달할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 기록하고 지정한 Amazon S3 버킷에 로그 파일을 전달합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 작업하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail

- 여러 리전에서 CloudTrail 로그 파일 수신
- 여러 계정에서 CloudTrail 로그 파일 수신

모든 AWS IoT FleetWise 작업은 에 의해 기록 CloudTrail 되고 AWS IoT FleetWise API 참조 에 문서화됩니다. 예를 들어 `CreateCampaign`, `AssociateVehicleFleet`및 `GetModelManifest` 작업에 대한 호출은 CloudTrail 로그 파일에 항목을 생성합니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 IAM 사용자 자격 증명으로 이루어졌는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청을 했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

AWS IoT FleetWise 로그 파일 항목 이해

추적은 사용자가 지정한 Amazon S3 버킷에 로그 파일로 이벤트를 전달할 수 있도록 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 파라미터 등에 대한 정보를 포함합니다. CloudTrail log 파일은 퍼블릭 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 *`AssociateVehicleFleet`* 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.


```
{
      "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 개발자 안내서의 문서 기록

다음 표에서는 AWS IoT FleetWise용 설명서 릴리스를 설명합니다.

변경 사항	설명	날짜
비전 시스템 데이터 미리 보기	AWS IoT FleetWise 의 비전 시스템 데이터 미리 보기를 사용하여 카메라, 레이더 및 덮개를 포함한 차량 비전 시스템에서 데이터를 수집하고 구성할 수 있습니다. IoT FleetWise는 정형 및 비정형 비전 시스템 데이터, 메타데이터(이벤트 ID, 캠페인, 차량), 표준 센서 데이터(텔레메트리 데이터)를 클라우드에서 자동으로 동기화된 상태로 유지합니다.	2023년 11월 26일
AWS KMS 고객 관리형 키	AWS IoT FleetWise 는 이제 AWS KMS 고객 관리형 키를 지원합니다. KMS 키를 사용하여 에 저장된 AWS IoT FleetWise 리소스(신호 카탈로그, 차량 모델, 디코더 매니페스트, 차량 및 데이터 수집 캠페인 구성)와 관련된 서버 측 데이터를 암호화할 수 있습니다 AWS 클라우드.	2023년 10월 16일
Amazon S3의 객체 스토리지	이제AWS IoT FleetWise 는 Amazon Simple Storage Service(Amazon S3)를 사용하여 데이터 저장을 지원합니다. Amazon Timestream 외에도 Amazon S3에 캠페인 중에 수	2023년 6월 1일

	집된 데이터를 저장할 수 있습니다.	
정식 출시	AWS IoT FleetWise 의 퍼블릭 릴리스입니다.	2022년 9월 27일
최초 릴리스	AWS IoT FleetWise 개발자 안내서의 미리 보기 릴리스입니다.	2021년 11월 30일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.