Guía para desarrolladores

AWS IoT FleetWise

AWS IoT FleetWise: Guía para desarrolladores

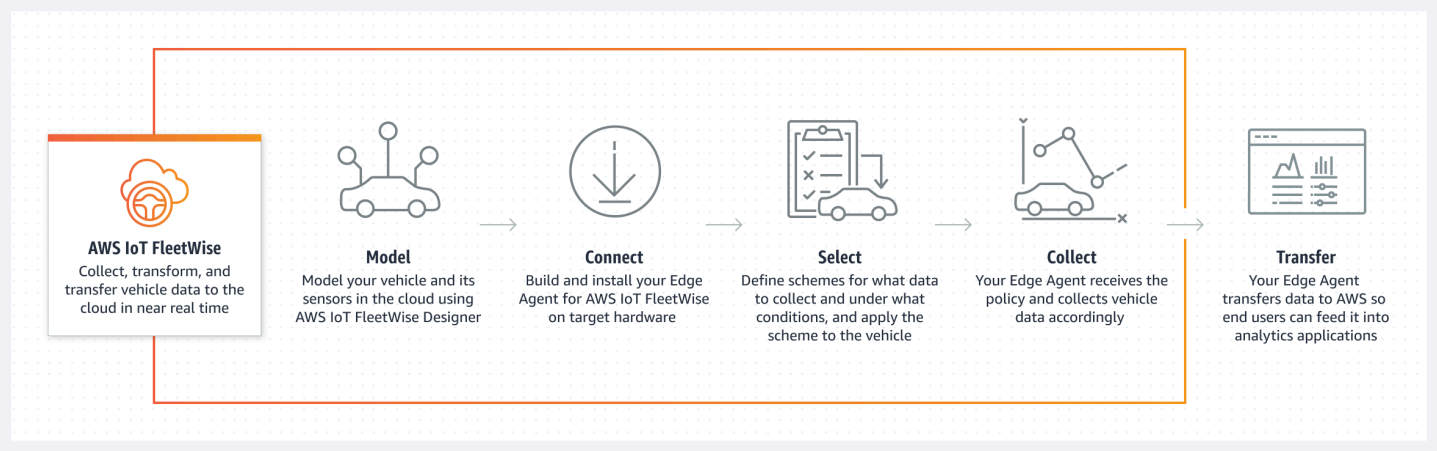
Table of Contents

¿Qué es AWS IoT FleetWise?

AWS IoT FleetWise es un servicio administrado que puede utilizar para recopilar datos de vehículos y organizarlos en la nube. Puede utilizar los datos recopilados para mejorar la calidad, el rendimiento y la autonomía del vehículo. Con AWS IoT FleetWise, puede recopilar y organizar datos de vehículos que utilizan diferentes protocolos y formatos de datos. AWS IoT FleetWise lo ayuda a transformar los mensajes de bajo nivel en valores en lenguaje natural y a estandarizar el formato de los datos en la nube para los análisis de datos. También le permite definir campañas de recopilación de datos para controlar qué datos del vehículo deben recopilarse y cuándo deben transferirse a la nube.

Cuando los datos del vehículo están en la nube, puede utilizarlos para aplicaciones que analicen el estado de la flota de vehículos. Estos datos pueden ayudarlo a identificar posibles problemas de mantenimiento, hacer que los sistemas de información y entretenimiento integrados en los vehículos sean más inteligentes y mejorar las tecnologías avanzadas, como la conducción autónoma y los sistemas de asistencia al conductor, mediante análisis y machine learning (ML).

En el siguiente diagrama se muestra la arquitectura básica de AWS IoT FleetWise.



Temas

- Servicios de relacionados

Ventajas

Los principales beneficios de AWS IoT FleetWise son:

Recopilación de los datos del vehículo de forma más inteligente

Mejore la relevancia de los datos con una recopilación de datos inteligente que envía solo los datos que necesita a la nube para su análisis.

Análisis de forma sencilla de los datos estandarizados de toda la flota

Analice los datos estandarizados de una flota de vehículos sin necesidad de desarrollar un sistema personalizado de registro o recopilación de datos.

Sincronización automática de datos en la nube

Obtenga una vista unificada de los datos recopilados tanto de los sensores estándar (datos de telemetría) como de los sistemas de visión (datos de cámaras, radares y LIDAR) y manténgalos sincronizados automáticamente en la nube. AWS IoT FleetWise mantiene los datos de sistemas de visión estructurados y no estructurados, los metadatos y los datos de sensores estándar sincronizados automáticamente en la nube. Esto agiliza el proceso para obtener una vista panorámica completa de los eventos y obtener información valiosa.

> **Note**
>
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Casos de uso

Los escenarios en los que puede utilizar AWS IoT FleetWise son los siguientes:

Entrenamiento de modelos de IA/ML

Mejore continuamente los modelos de machine learning utilizados en los sistemas de asistencia al conductor autónomos y avanzados mediante la recopilación de datos de los vehículos de producción.

Mejora de la experiencia digital del cliente

Utilice los datos de los sistemas de información para hacer que el contenido audiovisual integrado y la información en la aplicación sean más relevantes.

Mantenimiento del estado de la flota de vehículos

Utilice la información obtenida de los datos de la flota para monitorear el estado de las baterías de los vehículos eléctricos y los niveles de carga, administrar los programas de mantenimiento, analizar el consumo de combustible, etc.

¿Es la primera vez que utiliza AWS IoT FleetWise?

Si es nuevo en AWS IoT FleetWise, le recomendamos que empiece leyendo las siguientes secciones:

- Cómo funciona AWS IoT FleetWise
- Configuración de AWS IoT FleetWise
- Demostración del software Edge Agent
- Ingesta de datos en la nube

Acceso a AWS IoT FleetWise

Para acceder a AWS IoT FleetWise, puede utilizar su consola o API de AWS IoT FleetWise.

Precios de AWS IoT FleetWise

Los vehículos envían datos a la nube mediante mensajes MQTT. Al final de cada mes, pagará por los vehículos que haya creado en AWS IoT FleetWise, así como por los mensajes que haya recopilado de los vehículos. Para obtener información actualizada sobre los precios, consulte la página Precios de AWS IoT FleetWise. Para obtener más información sobre el protocolo de mensajería MQTT, consulte MQTT en la Guía para desarrolladores de AWS IoT Core.

Cómo funciona AWS IoT FleetWise

En las siguientes secciones se incluye información general sobre los componentes del servicio de AWS IoT FleetWise y las interacciones entre ellos.

Tras leer esta introducción, consulte la sección Configuración de AWS IoT FleetWise para aprender a configurar AWS IoT FleetWise.

Temas

Conceptos clave

AWS IoT FleetWise proporciona un marco de modelado de vehículos para que pueda modelar el vehículo y sus sensores y actuadores en la nube. Para permitir la comunicación segura entre el vehículo y la nube, AWS IoT FleetWise también proporciona una implementación de referencia para ayudarlo a desarrollar el software Edge Agent que puede instalar en su vehículo. Puede definir esquemas de recopilación de datos en la nube e implementarlos en el vehículo. El software Edge Agent que se ejecuta en el vehículo utiliza esquemas de recopilación de datos para controlar qué datos deben recopilarse y cuándo deben transferirse a la nube.

A continuación se enumeran los conceptos básicos de AWS IoT FleetWise:

Señal

Las señales son estructuras fundamentales que se definen para contener los datos del vehículo y sus metadatos. Una señal puede ser un atributo, una ramificación, un sensor o un actuador. Por ejemplo, puede crear un sensor para recibir los valores de temperatura del vehículo y almacenar sus metadatos, incluidos el nombre del sensor, un tipo de datos y una unidad. Para obtener más información, consulte Creación y administración de catálogos de señales.

Atributo

Los atributos representan información estática que, por lo general, no cambia, como el fabricante y la fecha de fabricación.

Crear ramificaciones

Las ramificaciones representan señales en una estructura anidada. Las ramificaciones muestran las jerarquías de señales. Por ejemplo, la ramificación `Vehicle` tiene una ramificación secundaria, `Powertrain`. La ramificación `Powertrain` tiene una ramificación secundaria, `combustionEngine`. Para localizar la ramificación `combustionEngine`, utilice la expresión `Vehicle.Powertrain.combustionEngine`.

Sensor

Los datos de los sensores indican el estado actual del vehículo y cambian con el tiempo, a medida que el estado del vehículo cambia también, como los niveles de líquido, las temperaturas, las vibraciones o el voltaje, por ejemplo.

Actuador

Los datos del actuador indican el estado de los dispositivos del vehículo, como los motores, la calefacción y las cerraduras de las puertas. Al cambiar el estado del dispositivo de un vehículo, es posible actualizar los datos del actuador. Por ejemplo, puede definir un actuador para que represente la calefacción. El actuador recibe datos nuevos al encender o apagar la calefacción.

Estructura personalizada

Una estructura personalizada (también conocida como estructura) representa una estructura de datos compleja o de orden superior. Facilita el enlace lógico o la agrupación de datos que se originan en la misma fuente. Una estructura se utiliza cuando los datos se leen o escriben en una operación atómica, por ejemplo, para representar un tipo de datos complejo o una forma de orden superior.

Una señal de tipo estructura se define en el catálogo de señales mediante una referencia a un tipo de datos de estructura en lugar de a un tipo de datos primitivo. Las estructuras se pueden utilizar para todo tipo de señales, incluidos los sensores, los atributos, los actuadores y los tipos de datos de sistemas de visión. Si se envía o recibe una señal de tipo estructural, AWS IoT FleetWise espera que todos los elementos incluidos tengan valores válidos, por lo que todos los elementos son obligatorios. Por ejemplo, si una estructura contiene los elementos Vehicle.Camera.Image.height, Vehicle.Camera.Image.width y Vehicle.Camera.Image.data, se espera que la señal enviada contenga valores para todos estos elementos.

> **Note**
>
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Propiedad personalizada

Una propiedad personalizada representa un miembro de la estructura de datos compleja. El tipo de datos de la propiedad puede ser primitivo u otra estructura.

Al representar una forma de orden superior mediante una estructura y una propiedad personalizada, la forma de orden superior deseada siempre se define y visualiza como una estructura de árbol. La propiedad personalizada se usa para definir todos los nodos de hoja, mientras que la estructura se usa para definir todos los nodos que no son de hoja.

Catálogo de señales

Un catálogo de señales contiene una colección de señales. Las señales de un catálogo de señales se pueden utilizar para modelar vehículos que utilizan diferentes protocolos y formatos de datos. Por ejemplo, supongamos que hay dos automóviles fabricados por distintos fabricantes de automóviles: uno usa el protocolo de red de área de control (bus CAN) y el otro usa el protocolo de diagnóstico a bordo (OBD). Puede definir un sensor en el catálogo de señales para recibir los valores de temperatura de los vehículos. Este sensor se puede utilizar para representar los termopares de ambos coches. Para obtener más información, consulte Creación y administración de catálogos de señales.

Modelo de vehículo (manifiesto del modelo)

Los modelos de vehículo son estructuras declarativas que puede utilizar para estandarizar el formato de los vehículos y definir las relaciones entre sus señales. Los modelos de vehículo permiten hacer que la información sea coherente en varios vehículos del mismo tipo. Para crear modelos de vehículo, debe agregar señales. Para obtener más información, consulte Creación y administración de modelos de vehículo.

Manifiesto del decodificador

Los manifiestos del decodificador contienen información de decodificación para cada señal en los modelos de vehículo. Los sensores y actuadores de los vehículos transmiten mensajes de bajo nivel (datos binarios). Mediante los manifiestos del decodificador, AWS IoT FleetWise puede transformar datos binarios en valores legibles. Cada manifiesto del decodificador está asociado a un modelo de vehículo. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Interfaz de red

Contiene información sobre el protocolo que utiliza la red integrada en el vehículo. AWS IoT FleetWise admite los siguientes protocolos:

Red de área de control (bus CAN)

Protocolo que define cómo se comunican los datos entre las unidades de control electrónico (ECU). Las ECU pueden ser la unidad de control del motor, los airbags o el sistema de audio.

Diagnóstico a bordo (OBD) II

Un protocolo perfeccionado que define cómo se comunican los datos de autodiagnóstico entre las ECU. Proporciona una serie de códigos de diagnóstico de problemas (DTC) estándar que ayudan a identificar qué es lo que falla en el vehículo.

Middleware de vehículos

El middleware de vehículos se define como un tipo de interfaz de red. Algunos ejemplos de middleware de vehículos incluyen Robot Operating System (ROS 2) y el middleware escalable orientado a servicios sobre IP (SOME/IP).

> **Note**
>
> AWS IoT FleetWise es compatible con el middleware de ROS 2 para los datos de sistemas de visión.

Señal del decodificador

Proporciona información de decodificación detallada para una señal específica. Todas las señales especificadas en el modelo de vehículo deben estar emparejadas con una señal del decodificador. Si el manifiesto del decodificador contiene interfaces de red CAN, debe contener señales del decodificador CAN. Si el manifiesto del decodificador contiene interfaces de red OBD, debe contener señales del decodificador OBD.

El manifiesto del decodificador debe contener las señales del decodificador de mensajes si también contiene interfaces de middleware de vehículos.

Vehículo

Una representación virtual del vehículo físico, como un automóvil o un camión. Los vehículos son instancias de modelos de vehículo. Los vehículos creados a partir del mismo modelo de vehículo heredan el mismo grupo de señales. Cada vehículo corresponde a un objeto de AWS IoT.

Flota

Una flota representa un grupo de vehículos. Para poder administrar fácilmente una flota de vehículos, debe asociar los vehículos individuales a una flota.

Campaña

Contiene esquemas de recopilación de datos. Puede definir una campaña en la nube e implementarla en un vehículo o una flota. Las campañas proporcionan al software Edge Agent instrucciones sobre cómo deben seleccionarse, recopilarse y transferirse datos a la nube.

Esquema de recopilación de datos

Los esquemas de recopilación de datos proporcionan al software Edge Agent instrucciones sobre cómo deben recopilarse datos. Actualmente, AWS IoT FleetWise admite el esquema de recopilación basado en la condición y el esquema de recopilación basado en el tiempo.

Esquema de recopilación basado en la condición

Utilice una expresión lógica para reconocer qué datos deben recopilarse. El software Edge Agent recopila datos cuando se cumple la condición. Por ejemplo, si la expresión es `$variable.myVehicle.InVehicleTemperature >35.0`, el software Edge Agent recopila valores de temperatura superiores a 35,0.

Esquema de recopilación basado en el tiempo

Especifique un periodo en milisegundos para definir la frecuencia de recopilación de datos. Por ejemplo, si el periodo es de 10 000 milisegundos, el software Edge Agent recopila datos una vez cada 10 segundos.

Características de AWS IoT FleetWise

Las principales características de AWS IoT FleetWise son las siguientes:

Modelado de vehículos

Cree representaciones virtuales de los vehículos y aplique un formato común para organizar sus señales. AWS IoT FleetWise es compatible con la especificación de señales de vehículos (VSS) que puede utilizar para estandarizar las señales de los vehículos.

Recopilación de datos basada en esquemas

Defina esquemas para transferir solo datos de vehículos de gran valor a la nube. Puede definir esquemas basados en la condición para controlar los datos que deben recopilarse, como los valores de temperatura en el interior del vehículo superiores a 40 grados. También puede definir esquemas basados en el tiempo para controlar con qué frecuencia deben recopilarse los datos.

Software Edge Agent para AWS IoT FleetWise

El software Edge Agent que se ejecuta en los vehículos facilita la comunicación entre los vehículos y la nube. Mientras los vehículos están conectados a la nube, el software Edge Agent recibe continuamente esquemas de recopilación de datos y recopila los datos en consecuencia.

Servicios de relacionados

AWS IoT FleetWise se integra con los siguientes servicios de AWS para mejorar la disponibilidad y escalabilidad de sus soluciones en la nube.

- AWS IoT Core: registre y controle los dispositivos de AWS IoT que cargan datos del vehículo a AWS IoT FleetWise. Para obtener más información, consulte ¿Qué es AWS IoT? en la Guía para desarrolladores de AWS IoT.
- Amazon Timestream: utilice una base de datos de serie temporal para almacenar y analizar los datos del vehículo. Para obtener más información, consulte ¿Qué es Amazon Timestream? en la Guía para desarrolladores de Amazon Timestream.
- Amazon S3: utilice un servicio de almacenamiento de objetos para almacenar y administrar los datos del vehículo. Para obtener más información, consulte ¿Qué es Amazon S3? en la Guía del usuario de Amazon Simple Storage Service.

Configuración de AWS IoT FleetWise

Antes de usar el AWS IoT FleetWise por primera vez, complete los pasos de las siguientes secciones.

Temas

Configura tu Cuenta de AWS

Complete las siguientes tareas para registrarse AWS y crear un usuario administrativo.

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

> **Note**
>
> Puedes usar un rol vinculado a un servicio con IoT AWS . FleetWise Las funciones vinculadas a los servicios están predefinidas por la AWS IoT FleetWise e incluyen los permisos que la AWS IoT FleetWise necesita para enviar métricas a Amazon. CloudWatch Para obtener más información, consulte Uso de roles vinculados a servicios para AWS IoT FleetWise.

Empezar a trabajar con la consola

Si aún no has iniciado sesión en tu Cuenta de AWS, inicia sesión y, a continuación, abre la FleetWise consola de AWS IoT. Para empezar con el AWS IoT FleetWise, cree un modelo de vehículo. Un modelo de vehículo estandariza el formato de los vehículos.

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En Comenzar con AWS IoT FleetWise, selecciona Comenzar.

Para obtener más información acerca de la creación de un modelo de vehículo, consulte Creación de un modelo de vehículo (consola).

Configuración de los ajustes

Puede usar la FleetWise consola o la API de AWS IoT para configurar los ajustes de las métricas de Amazon CloudWatch Logs, Amazon CloudWatch Logs y cifrar los datos con un Clave administrada de AWS.

Con CloudWatch las métricas, puede monitorear el AWS IoT FleetWise y otros AWS recursos. Puede usar CloudWatch las métricas para recopilar y realizar un seguimiento de las métricas, por ejemplo, para determinar si se ha superado el límite de servicio. Para obtener más información sobre CloudWatch las métricas, consulteSupervisión de AWS IoT FleetWise con Amazon CloudWatch.

Con CloudWatch los registros, el AWS IoT FleetWise envía los datos de CloudWatch registro a un grupo de registros, donde puede usarlos para identificar y mitigar cualquier problema. Para obtener más información sobre CloudWatch los registros, consulteConfiguración del registro de AWS IoT FleetWise.

Con el cifrado de datos, el AWS IoT se FleetWise utiliza Claves administradas por AWS para cifrar los datos. También puede optar por crear y administrar claves con AWS KMS. Para obtener más información sobre el cifrado, consulte Cifrado de datos.

Configuración (consola)

Si aún no has iniciado sesión en tu Cuenta de AWS, inicia sesión y, a continuación, abre la FleetWiseconsola de AWS IoT.

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel izquierdo, elija Configuración.
3. En Metrics, selecciona Activar. AWS El IoT FleetWise asocia automáticamente una política CloudWatch gestionada a la función vinculada al servicio y habilita las métricas. CloudWatch
4. In Registro, elija Editar.
    a. En la sección de CloudWatch registro, introduzca el grupo de registros.
    b. Para guardar los cambios, elija Enviar.
5. En la sección Cifrado, elija Editar.
    a. Elija el tipo de clave que desee usar. Para obtener más información, consulte Administración de claves.

i. Use AWS la clave: AWS IoT FleetWise posee y administra la clave.

ii. Elige una AWS Key Management Service clave diferente: tú administras las AWS KMS keys que están en tu cuenta.

b. Para guardar los cambios, elija Enviar.

Configuración de los ajustes (AWS CLI)

En el AWS CLI, registre la cuenta para configurar los ajustes.

1. Para configurar los ajustes, ejecute el siguiente comando.

```
aws iotfleetwise register-account
```

2. Para comprobar la configuración, ejecute el siguiente comando para recuperar el estado de registro.

> **Note**
>
> La función vinculada al servicio solo se usa para publicar FleetWise métricas de AWS IoT en. CloudWatch Para obtener más información, consulte Uso de roles vinculados a servicios para AWS IoT FleetWise.

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

Example Respuesta

```
{
    "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",
    }
```

```
}
```

El estado de registro puede ser uno de los siguientes valores:

- `REGISTRATION_SUCCESS`— El AWS recurso se ha registrado correctamente.
- `REGISTRATION_PENDING`— AWS IoT FleetWise está procesando la solicitud de registro. Este proceso tarda aproximadamente cinco minutos en completarse.
- `REGISTRATION_FAILURE`— AWS IoT no FleetWise puede registrar el AWS recurso. Inténtelo de nuevo más tarde.

Cómo empezar con el AWS IoT FleetWise

Con el AWS IoT FleetWise, puede recopilar, transformar y transferir los datos de su vehículo. Utilice los tutoriales de esta sección para empezar con el AWS IoT FleetWise.

Consulte los siguientes temas para obtener más información sobre el AWS IoT FleetWise:

- Ingesta de datos en la nube
- Modelización de vehículos
- Creación, aprovisionamiento y administración de vehículos
- Creación y administración de flotas
- Recopilación y transferencia de datos con campañas

Requisitos

Debe tener una Cuenta de AWS para empezar con el AWS IoT FleetWise. Si no dispone de una, consulte Configuración de AWS IoT FleetWise.

Utilice una región en la que FleetWise esté disponible el AWS IoT. Para obtener más información, consulte FleetWise Puntos finales y cuotas de AWS IoT. Puede utilizar el selector de regiones de la AWS Management Console para cambiar a una de estas regiones.

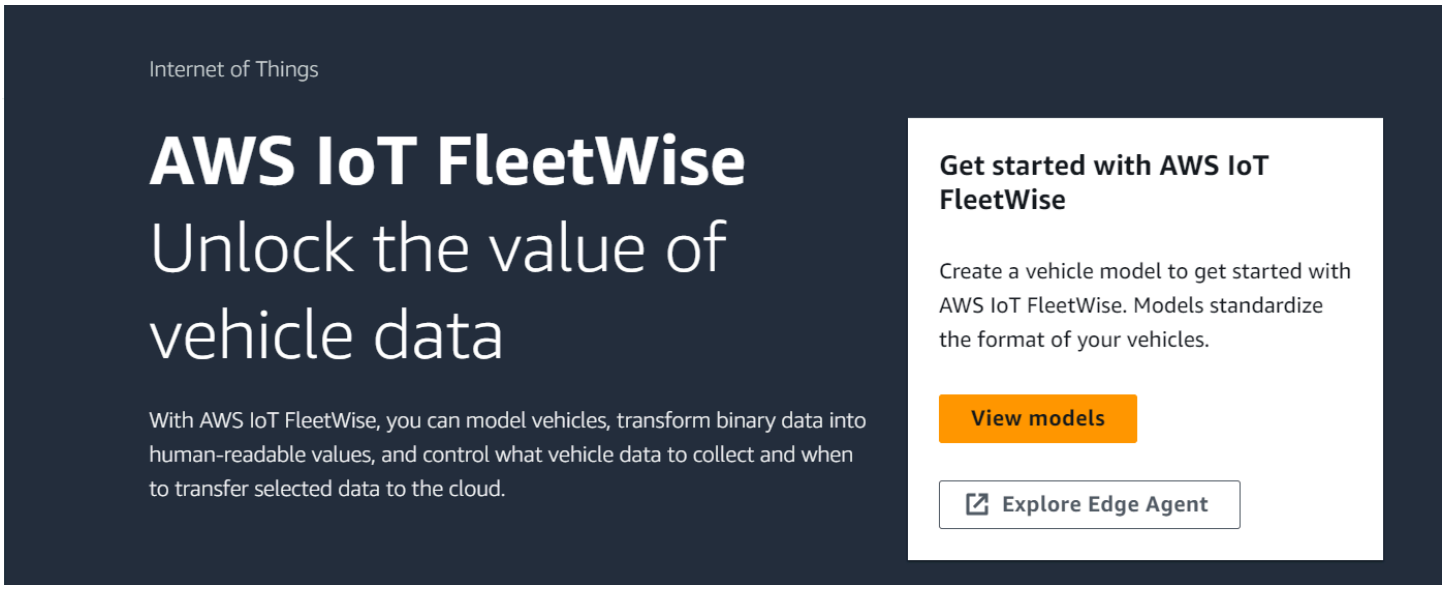
Demostración del software Edge Agent

Puede utilizar la demostración de inicio rápido de Explore Edge Agent para explorar el AWS IoT FleetWise y aprender a desarrollar el software Edge Agent para AWS IoT FleetWise. En esta demostración se utiliza una AWS CloudFormation plantilla. que lo guía por la revisión de la implementación de referencia de Edge Agent, el desarrollo de Edge Agent y, a continuación, la implementación del software Edge Agent en un Graviton de Amazon EC2 y la generación de datos de muestra del vehículo. La demostración también incluye un script que puede utilizar para crear un catálogo de señales, un modelo de vehículo, un manifiesto del decodificador, un vehículo, una flota y una campaña, todo ello en la nube. Para obtener más información sobre la demostración de inicio rápido, haga lo siguiente para descargar la guía para desarrolladores del software Edge Agent.

Descarga de la demostración de inicio rápido

1. Navegue hasta la FleetWiseconsola de AWS IoT.

2. En la página de inicio del servicio, en la FleetWise sección Comenzar con AWS IoT, elija Explore Edge Agent.



Tutorial: Primeros pasos con el AWS IoT FleetWise (consola)

Utilice el AWS IoT FleetWise para recopilar, transformar y transferir el formato de datos exclusivo de los vehículos automatizados a la nube casi en tiempo real. Tiene acceso a información de toda la flota. Esto puede ayudarle a detectar y mitigar de forma eficiente los problemas relacionados con el estado de los vehículos, a transferir señales de datos de gran valor y a diagnosticar problemas de forma remota, todo ello a la vez que reduce los costos.

En este tutorial, se muestra cómo empezar a utilizar el AWS IoT FleetWise. Aprenderá a crear un modelo de vehículo (manifiesto del modelo), un manifiesto del decodificador, un vehículo y una campaña.

Para obtener más información sobre los componentes y conceptos clave de la AWS IoT FleetWise, consulteCómo funciona AWS IoT FleetWise.

Tiempo estimado: 45 minutos, aproximadamente.

> **Important**
>
> Se le cobrará por los FleetWise recursos de AWS IoT que cree y consuma esta demostración. Para obtener más información, consulte AWS IoT FleetWise en la página de FleetWise precios de AWS IoT.

Temas

Requisitos previos

Para completar este tutorial de introducción, primero necesita lo siguiente:

- Un Cuenta de AWS. Si no tiene uno Cuenta de AWS, consulte Creación de uno Cuenta de AWS en la Guía de AWS Account Management referencia.
- Acceso a una plataforma Región de AWS compatible con el AWS IoT FleetWise. Actualmente, el AWS IoT FleetWise es compatible en EE. UU. Este (Virginia del Norte) y Europa (Frankfurt).
- Recursos de Amazon Timestream:
  - Una base de datos de Amazon Timestream. Para obtener más información, consulte Crear una base de datos en la Guía para desarrolladores de Amazon Timestream.
  - Una tabla de Amazon Timestream creada en Amazon Timestream que contendrá sus datos. Para obtener más información, consulte Crear una tabla en la Guía para desarrolladores de Amazon Timestream.

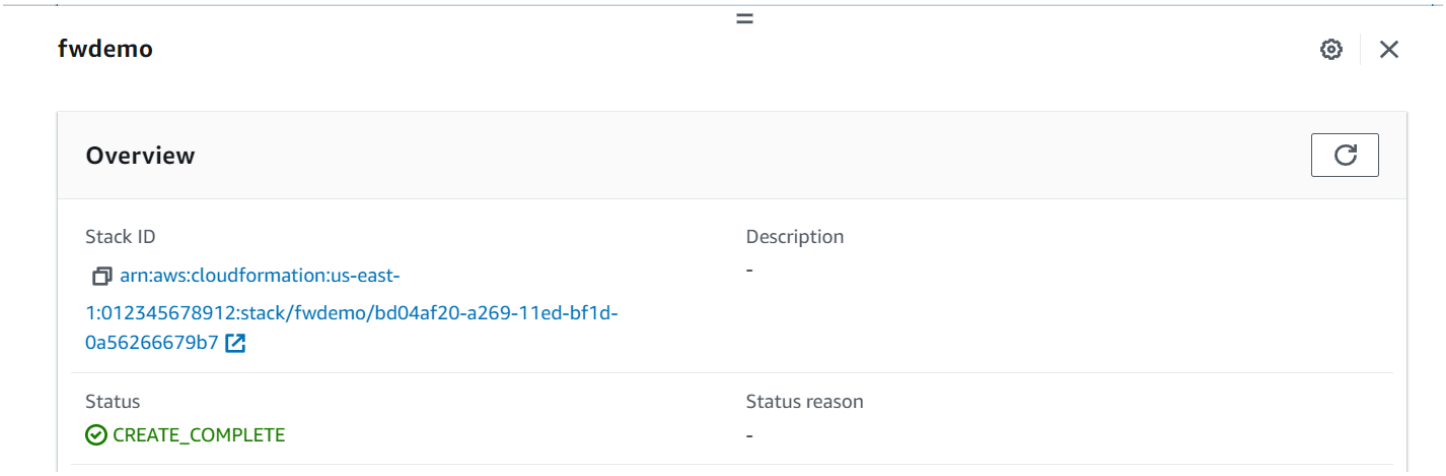
Paso 1: Configurar el software Edge Agent para AWS IoT FleetWise

> **Note**
>
> La CloudFormation pila de este paso usa datos de telemetría. También puede crear una CloudFormation pila con los datos del sistema de visión. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Su software Edge Agent para AWS IoT FleetWise facilita la comunicación entre los vehículos y la nube. Recibe instrucciones de los esquemas de recopilación de datos sobre cómo deben recopilarse datos de vehículos conectados a la nube.

Para configurar el software Edge Agent, haga lo siguiente en Información general:

1. Abra la CloudFormation plantilla de lanzamiento.
2. En la página Creación rápida de una pila, en Nombre de pila, introduce el nombre de tu pila de FleetWise recursos de AWS IoT. Una pila es un nombre descriptivo que aparece como prefijo en los nombres de los recursos que crea esta AWS CloudFormation plantilla.
3. En Parámetros, introduzca los valores personalizados para los parámetros relacionados con la pila.
   a. Fleetsize: puede aumentar el número de vehículos de la flota actualizando el parámetro Fleetsize.
   b. IoT CoreRegion - Puede especificar la región en la AWS IoT que se crea la cosa actualizando el CoreRegion parámetro de IoT. Debes usar la misma región que usaste para crear tus FleetWise vehículos de AWS IoT. Para obtener más información Regiones de AWS, consulte Regiones y zonas: Amazon Elastic Compute Cloud.
4. En la sección Capacidades, selecciona la casilla para confirmar que AWS CloudFormation crea recursos de IAM.
5. Elija Crear pila y espere aproximadamente 15 minutos hasta que el estado de la pila muestre CREATE_COMPLETE.
6. Para confirmar que se ha creado la pila, elija la pestaña Información de la pila, actualice la vista y busque CREATE_COMPLETE.



> **Important**
>
> Se le cobrará por los FleetWise recursos de AWS IoT que cree y consuma esta demostración. Para obtener más información, consulte AWS IoT FleetWise en la página de FleetWise precios de AWS IoT.

Paso 2: Crear un modelo de vehículo

> **Important**
>
> No se puede crear un modelo de vehículo con señales de datos del sistema de visión en la FleetWise consola de AWS IoT. En su lugar, utilice el AWS CLI.

Puede utilizar los modelos de vehículo para estandarizar el formato de los vehículos y para ayudar a definir la relación entre las señales de los vehículos que crea. Al crear un modelo de vehículo, también se crea un catálogo de señales. Un catálogo de señales es una colección de señales estandarizadas que se pueden reutilizar para crear modelos de vehículo. Las señales son estructuras fundamentales que se definen para contener los datos del vehículo y sus metadatos. En este momento, el FleetWise servicio de AWS IoT solo admite un catálogo de señales Región de AWS por cuenta. Esto ayuda a comprobar la coherencia de los datos procesados desde una flota de vehículos.

Para crear un modelo de vehículo:

1. Abre la FleetWise consola AWS de IoT.

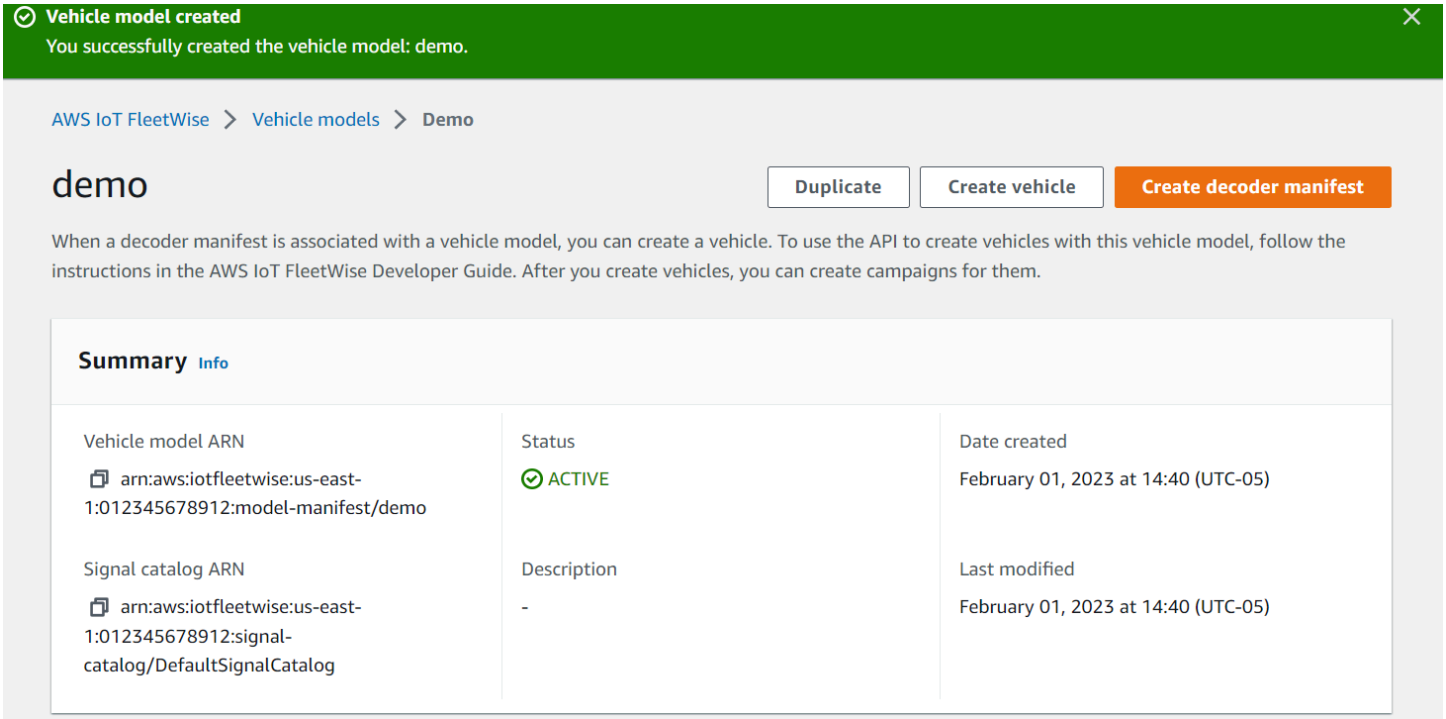
2. En el panel de navegación, elija Modelos de vehículo.
3. En la página Modelos de vehículo, elija Crear modelo de vehículo.
4. En la sección Información general, introduzca el nombre del modelo del vehículo, como Vehículo1, y una descripción opcional. A continuación, elija Siguiente.
5. Elija una o más señales del catálogo de señales. Puede filtrar las señales por nombre en el catálogo de búsqueda o bien elegirlas de la lista. Por ejemplo, puede elegir señales para la presión de los neumáticos y la presión de frenado para poder recopilar datos relacionados con ellas. Elija Siguiente.
6. Elija los archivos .dbc y cárguelos desde su dispositivo local. Elija Siguiente.

   > **Note**
   >
   > Para completar este tutorial, puede descargar un archivo .dbc de muestra para subirlo en este paso.

7. Agregue atributos al modelo de vehículo y, a continuación, elija Siguiente.

   a. Nombre: introduzca el nombre del atributo del vehículo, como el nombre del fabricante o la fecha de fabricación.
   b. Tipo de datos: en el menú Tipo de datos, elija un tipo de datos.
   c. Unidad: (opcional) introduzca un valor unitario, como kilómetros o grados Celsius.
   d. Ruta: (opcional) introduzca un nombre para la ruta a una señal, como `Vehicle.Engine.Light.` El punto (.) indica que se trata de una señal secundaria.
   e. Valor predeterminado: (opcional) introduzca un valor predeterminado.
   f. Descripción: (opcional) introduzca una descripción del atributo.

8. Revise la configuración. Cuando haya terminado, elija Create (Crear). Aparecerá una notificación para indicar que el modelo de vehículo se ha creado correctamente.



Paso 3: Crear un manifiesto del decodificador

Los manifiestos del decodificador están asociados a los modelos de vehículo que cree. Contienen información que ayuda al AWS IoT a FleetWise decodificar y transformar los datos del vehículo desde un formato binario en valores legibles por humanos que se pueden analizar. Las interfaces de red y las señales del decodificador son componentes que ayudan a configurar los manifiestos del decodificador. Una interfaz de red contiene información sobre el protocolo CAN u OBD que utiliza la red del vehículo. La señal del decodificador proporciona información de decodificación para una señal específica.

Para crear un manifiesto del decodificador:

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. En la sección Modelos de vehículo, elija el modelo de vehículo que quiera usar para crear un manifiesto del decodificador.
4. Elija Crear manifiesto del decodificador.

Paso 4: Configurar un manifiesto del decodificador

Para configurar un manifiesto del decodificador:

> **Important**
>
> No se pueden configurar las señales de datos del sistema de visión en los manifiestos del decodificador mediante la FleetWise consola de AWS IoT. En su lugar, utilice el AWS CLI. Para obtener más información, consulte Creación de un manifiesto del decodificador (AWS CLI).

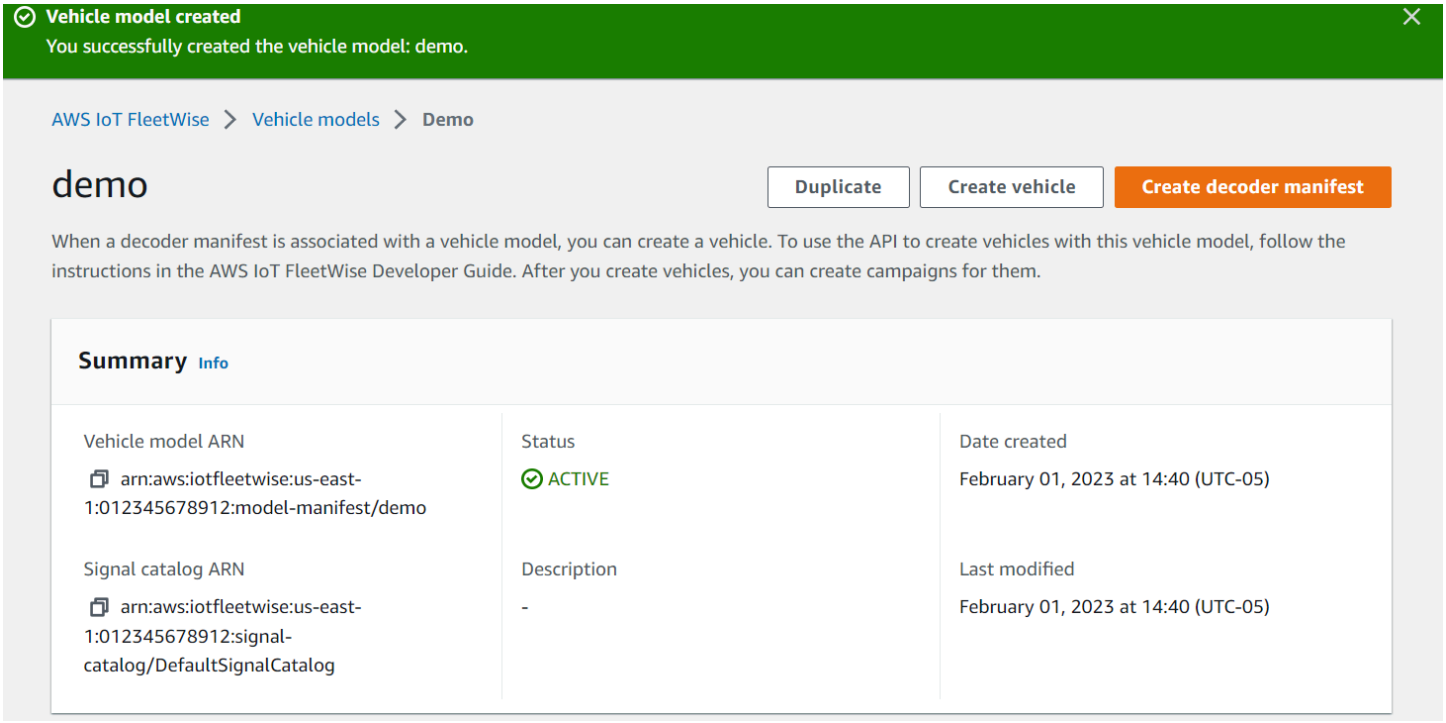
1. Para ayudarlo a identificar el manifiesto del decodificador, introduzca un nombre y una descripción opcional para el mismo. A continuación, elija Siguiente.
2. Para agregar una o más interfaces de red, elija el tipo CAN_INTERFACE u OBD_INTERFACE.
   - Interfaz de diagnóstico a bordo (OBD) integrada: elija este tipo de interfaz si desea un protocolo que defina cómo deben comunicarse los datos de autodiagnóstico entre las unidades de control electrónico (ECU). Este protocolo proporciona una serie de códigos de diagnóstico de problemas (DTC) estándar que pueden ayudarlo a solucionar problemas en el vehículo.
   - Interfaz de red de área de control (bus CAN): elija este tipo de interfaz si desea un protocolo que defina cómo deben comunicarse los datos entre las ECU. Las ECU pueden ser unidades de control del motor, airbags o sistemas de audio.
3. Introduzca un nombre de interfaz de red.
4. Para agregar señales a la interfaz de red, elija una o más señales de la lista.
5. Elija una señal decodificadora para la señal agregada en el paso anterior. Para proporcionar información de decodificación, cargue un archivo .dbc. Cada señal del modelo del vehículo debe estar emparejada con una señal del decodificador que puede elegir de la lista.
6. Para agregar otra interfaz de red secundaria, elija Agregar interfaz de red. Cuando haya terminado de agregar las interfaces de red, elija Siguiente.
7. Revise la configuración y elija Crear. Aparecerá una notificación para indicar que el manifiesto del decodificador se ha creado correctamente.

Paso 5: Crear un vehículo

En el AWS IoT FleetWise, los vehículos son representaciones virtuales de un vehículo físico real. Todos los vehículos creados a partir del mismo modelo de vehículo heredan el mismo grupo de señales y cada vehículo creado corresponde a un objeto de IoT de nueva creación. Debe asociar todos los vehículos a un manifiesto del decodificador.

Requisitos previos

1. Compruebe que ya ha creado el modelo de vehículo y el manifiesto del decodificador. Además, compruebe que el estado del modelo de vehículo sea ACTIVO.
   a. Para comprobar que el estado del modelo de vehículo es ACTIVO, abra la FleetWise consola de AWS IoT.
   b. En el panel de navegación, elija Modelos de vehículo.
   c. En la sección Resumen, en Estado, compruebe el estado del vehículo.



Para crear un vehículo:

1. Abra la FleetWise consola de AWS.
2. En el panel de navegación, elija Vehículos.

3. Elija Crear vehículo.
4. Para definir las propiedades del vehículo, introduzca el nombre del vehículo y, a continuación, elija un manifiesto del modelo (modelo de vehículo) y un manifiesto del decodificador.
5. (Opcional) Para definir los atributos del vehículo, introduzca un par clave-valor y, a continuación, elija Agregar atributos.
6. (Opcional) Para etiquetar el recurso de AWS, agregue etiquetas y, a continuación, elija Agregar etiqueta nueva.
7. Elija Siguiente.
8. Para configurar el certificado del vehículo, puede cargar su propio certificado o elegir Generar automáticamente un nuevo certificado. Le recomendamos que genere el certificado automáticamente para proceder de forma más rápida con la configuración. Si ya dispone de un certificado, puede utilizarlo en lugar de generar uno nuevo.
9. Descargue los archivos de clave pública y privada y, a continuación, elija Siguiente.
10. Para adjuntar una política al certificado del vehículo, puede introducir el nombre de una política existente o bien crear una nueva. Para crear una nueva política, seleccione Crear política y, a continuación, elija Siguiente.
11. Revise la configuración. Cuando haya terminado, elija Crear vehículo.

Paso 6: Crear una campaña

En el AWS IoT FleetWise, las campañas se utilizan para facilitar la selección, la recopilación y la transferencia de datos de los vehículos a la nube. Las campañas contienen esquemas de recopilación de datos que proporcionan al software Edge Agent instrucciones sobre cómo deben recopilarse datos con un esquema de recopilación basado en la condición o un esquema de recopilación basado en el tiempo.

Para crear una campaña:

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Campañas.
3. Elija Crear una campaña.
4. Introduzca el nombre de la campaña y una descripción opcional.
5. A la hora de configurar el esquema de recopilación de datos de la campaña, puede definir de forma manual el esquema de recopilación de datos o cargar un archivo .json desde el dispositivo local. Al cargar un archivo .json se define automáticamente el esquema de recopilación de datos.

a. Para definir de forma manual el esquema de recopilación de datos, seleccione Definir el esquema de recopilación de datos y elija el tipo de esquema que desee utilizar para la campaña. Puede elegir un esquema de recopilación basado en la condición o bien basado en el tiempo.

b. Si elige un esquema de recopilación basado en el tiempo, debe especificar durante cuánto tiempo la campaña recopilará los datos del vehículo.

c. Si elige un esquema de recopilación basado en la condición, debe especificar una expresión para reconocer qué datos deben recopilarse. Asegúrese de especificar el nombre de la señal como una variable, un operador de comparación y un valor de comparación.

d. (Opcional) Elija la versión lingüística de la expresión o manténgala como el valor predeterminado de 1.

e. (Opcional) Especifique el intervalo de activación entre dos eventos de recopilación de datos.

f. Para recopilar datos, elija la condición del modo del desencadenador para el software Edge Agent. De forma predeterminada, el FleetWise software Edge Agent para AWS IoT siempre recopila datos siempre que se cumpla la condición. O bien solo puede recopilar datos cuando se cumple la condición por primera vez, En el primer desencadenador.

g. (Opcional) Puede elegir opciones de esquema más avanzadas.

6. Para especificar las señales de las que el esquema de recopilación de datos recopilará datos, busque el nombre de la señal en el menú.
7. (Opcional) Puede elegir un recuento máximo de muestras o un intervalo mínimo de muestreo. También puede agregar más señales.
8. Elija Siguiente.
9. Defina el destino de almacenamiento al que desea que la campaña transfiera los datos. Puede almacenar datos en Amazon S3 o en Amazon Timestream.

   a. Amazon S3: elija el bucket de S3 para el que AWS IoT FleetWise tiene permisos.

   b. Amazon Timestream: elija la base de datos de Timestream y el nombre de la tabla. Introduzca una función de IAM que permita enviar datos AWS IoT FleetWise a Timestream.

10. Elija Siguiente.
11. Elija los atributos o los nombres de los vehículos en el cuadro de búsqueda.
12. Introduzca el valor relacionado con el atributo o el nombre que haya elegido para el vehículo.
13. Elija los vehículos de los que la campaña recopilará datos. A continuación, elija Siguiente.

14. Revise las configuraciones de la campaña y, a continuación, elija Crear campaña. Usted o su equipo deberán implementar la campaña en los vehículos.

Paso 7: limpiar

Para evitar que se te cobre más por los recursos que has utilizado durante este tutorial, elimina la AWS CloudFormation pila y todos los recursos de la pila.

Para eliminar la AWS CloudFormation pila

1. Abra la consola de AWS CloudFormation.
2. En la lista de pilas, elige la pila que creaste en el paso 1.
3. Elija Eliminar.
4. Para confirmar la eliminación, elija Delete (Eliminar). La pila tarda unos 15 minutos en borrarse.

Siguientes pasos

1. Puede procesar y visualizar los datos de los vehículos que recopila la campaña. Para obtener más información, consulte Procesamiento y visualización de los datos del vehículo.
2. Puede solucionar y resolver problemas con el AWS IoT FleetWise. Para obtener más información, consulte Solución de problemas de AWS IoT FleetWise.

Ingesta de datos en la nube

El software Edge Agent para AWS IoT FleetWise está diseñado para facilitar la comunicación segura entre los vehículos y la nube al instalarlo y ejecutarlo en vehículos.

> **Note**
>
> - AWS IoT FleetWise no está diseñado para usarse en entornos peligrosos o sistemas críticos, ni en asociación con ellos, cuyo funcionamiento pueda causar lesiones corporales graves o la muerte o daños ambientales o a la propiedad. Los datos del vehículo recopilados mediante el uso de AWS IoT FleetWise tienen solo fines informativos, y no puede usar AWS IoT FleetWise para controlar u operar las funciones del vehículo.
> - Los datos del vehículo recopilados mediante el uso de AWS IoT FleetWise deben evaluarse con precisión según corresponda a su caso de uso, incluso con el fin de cumplir con cualquier obligación de cumplimiento que pueda tener en virtud de la normativa de seguridad de los vehículos aplicable (como las obligaciones de supervisión de la seguridad y presentación de informes). Dicha evaluación debe incluir la recopilación y revisión de la información a través de otros medios y fuentes estándares del sector (como los informes de los conductores de vehículos).

Para ingerir datos en la nube, haga lo siguiente:

1. Desarrolle e instale el software Edge Agent para AWS IoT FleetWise en el vehículo. Para obtener más información sobre cómo trabajar con el software Edge Agent, haga lo siguiente para descargar la guía para desarrolladores del software Edge Agent para AWS IoT FleetWise.
   1. Vaya a la consola de AWS IoT FleetWise.
   2. En la página de inicio del servicio, en la sección Introducción a AWS IoT FleetWise, elija Conozca el agente de borde.
2. Cree o importe un catálogo de señales con las señales que utilizará para crear un modelo de vehículo. Para más información, consulte Creación de un catálogo de señales (AWS CLI) y Importación de un catálogo de señales (AWS CLI).

> **Note**
>
> - Si utiliza la consola de AWS IoT FleetWise para crear el primer modelo de vehículo, no es necesario que cree manualmente un catálogo de señales. Al crear el primer modelo de vehículo, AWS IoT FleetWise crea automáticamente un catálogo de señales. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
> - Actualmente, AWS IoT FleetWise admite un catálogo de señales para cada cuenta de AWS por Región de AWS.

3. Utilice las señales del catálogo de señales para crear un modelo de vehículo. Para obtener más información, consulte Creación de un modelo de vehículo.

> **Note**
>
> - Si utiliza la consola de AWS IoT FleetWise para crear un modelo de vehículo, puede cargar archivos .dbc para importar señales. .dbc es un formato de archivo compatible con las bases de datos de la red de área de control (bus CAN). Una vez creado el modelo de vehículo, las nuevas señales se agregan automáticamente al catálogo de señales. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
> - Si utiliza la operación de la API `CreateModelManifest` para crear un modelo de vehículo, debe usar la operación de la API `UpdateModelManifest` para activarlo. Para obtener más información, consulte Actualización de un modelo de vehículo (AWS CLI).
> - Si utiliza la consola de AWS IoT FleetWise para crear un modelo de vehículo, AWS IoT FleetWise lo activará automáticamente.

4. Cree un manifiesto del decodificador. El manifiesto del decodificador contiene información de decodificación para cada señal especificada en el modelo de vehículo creado en el paso anterior. El manifiesto del decodificador está asociado al modelo de vehículo creado. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

> **Note**
>
> - Si usa la operación de la API `CreateDecoderManifest` para crear un manifiesto del decodificador, debe usar la operación de la API `UpdateDecoderManifest` para

activarlo. Para obtener más información, consulte Actualización de un manifiesto del decodificador (AWS CLI).

- Si utiliza la consola de AWS IoT FleetWise para crear un manifiesto del decodificador, AWS IoT FleetWise lo activará automáticamente.

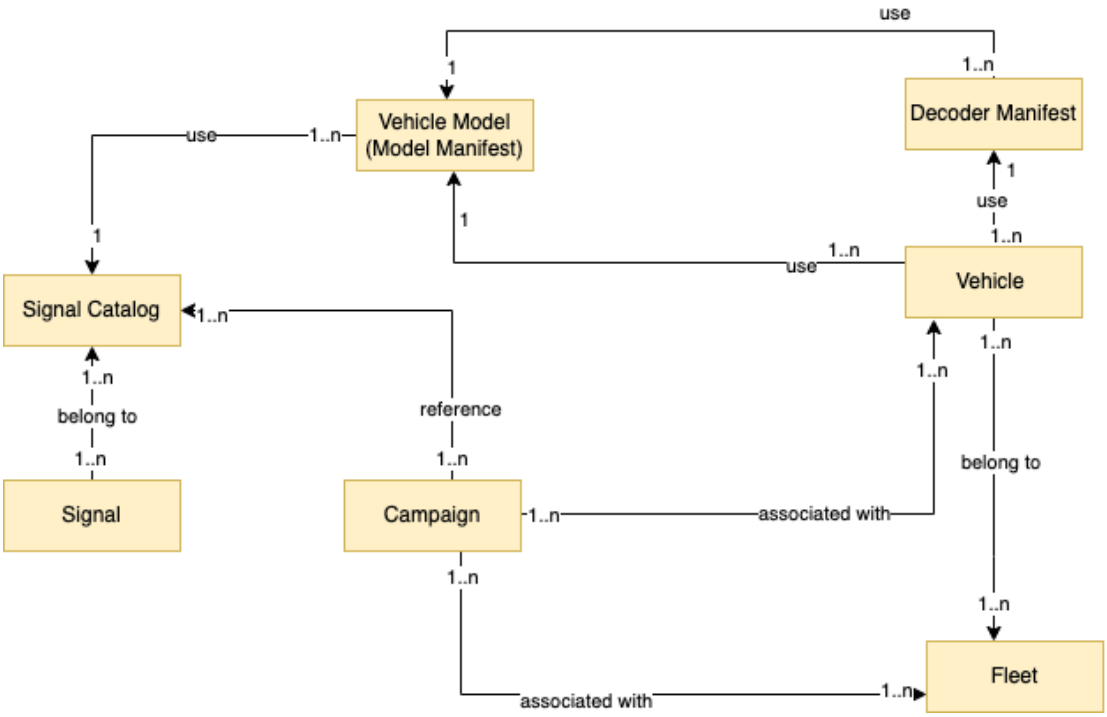
5. Cree vehículos a partir del modelo del vehículo. Los vehículos creados a partir del mismo modelo de vehículo heredan el mismo grupo de señales. Debe utilizar AWS IoT Core para aprovisionar el vehículo antes de poder ingerir datos en la nube. Para obtener más información, consulte Creación, aprovisionamiento y administración de vehículos.
6. (Opcional) Cree una flota para representar un grupo de vehículos y, a continuación, asocie cada uno de los vehículos a la flota. Esto lo ayudará a administrar varios vehículos al mismo tiempo. Para obtener más información, consulte Creación y administración de flotas.
7. Cree campañas. Las campañas se implementan en un vehículo o en una flota de vehículos. Las campañas proporcionan al software Edge Agent instrucciones sobre cómo deben seleccionarse, recopilarse y transferirse datos a la nube. Para obtener más información, consulte Recopilación y transferencia de datos con campañas.

> **Note**
>
> Debe usar la operación de la API `UpdateCampaign` para aprobar la campaña antes de que AWS IoT FleetWise pueda implementarla en el vehículo o en la flota. Para obtener más información, consulte Actualización de una campaña (AWS CLI).

El software Edge Agent transfiere los datos del vehículo a AWS IoT Core usando el tema reservado `$aws/iotfleetwise/vehicles/vehicleName/signals`, que envía los datos a AWS IoT FleetWise. A continuación, AWS IoT FleetWise envía los datos a una tabla de Timestream o a un bucket de Amazon S3. Puede utilizar Timestream para consultar los datos y Amazon QuickSight o Grafana para visualizarlos. Para obtener más información, consulte Procesamiento y visualización de los datos del vehículo.

Modelización de vehículos

AWS FleetWise El IoT proporciona un marco de modelado de vehículos que puede utilizar para crear representaciones virtuales de sus vehículos en la nube. Las señales, los catálogos de señales, los modelos de vehículo y los manifiestos del decodificador son los componentes principales con los que trabaja para modelar los vehículos.



Señales

Las señales son estructuras fundamentales que se definen para contener los datos del vehículo y sus metadatos. Una señal puede ser un atributo, una ramificación, un sensor o un actuador. Por ejemplo, puede crear un sensor para recibir los valores de temperatura del vehículo y almacenar sus metadatos, incluidos el nombre del sensor, un tipo de datos y una unidad. Para obtener más información, consulte Creación y administración de catálogos de señales.

Catálogo de señales

Un catálogo de señales contiene una colección de señales. Las señales de un catálogo de señales se pueden utilizar para modelar vehículos que utilizan diferentes protocolos y formatos de datos. Por ejemplo, supongamos que hay dos automóviles fabricados por distintos fabricantes de automóviles: uno usa el protocolo de red de área de control (bus CAN) y el otro usa el protocolo de diagnóstico a bordo (OBD). Puede definir un sensor en el catálogo de señales para recibir los valores de temperatura de los vehículos. Este sensor se puede utilizar para representar los

termopares de ambos coches. Para obtener más información, consulte Creación y administración de catálogos de señales.

Modelo de vehículo (manifiesto del modelo)

Los modelos de vehículo son estructuras declarativas que puede utilizar para estandarizar el formato de los vehículos y definir las relaciones entre sus señales. Los modelos de vehículo permiten hacer que la información sea coherente en varios vehículos del mismo tipo. Para crear modelos de vehículo, debe agregar señales. Para obtener más información, consulte Creación y administración de modelos de vehículo.

Manifiesto del decodificador

Los manifiestos del decodificador contienen información de decodificación para cada señal en los modelos de vehículo. Los sensores y actuadores de los vehículos transmiten mensajes de bajo nivel (datos binarios). Con los manifiestos del decodificador, el AWS IoT FleetWise puede transformar los datos binarios en valores legibles por humanos. Cada manifiesto del decodificador está asociado a un modelo de vehículo. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Puedes usar la FleetWise consola de AWS IoT o la API para modelar vehículos de la siguiente manera.

1. Cree o importe un catálogo de señales con las señales que utilizará para crear un modelo de vehículo. Para más información, consulte Creación de un catálogo de señales (AWS CLI) y Importación de un catálogo de señales (AWS CLI).

   > **Note**
   >
   > - Si utilizas la FleetWise consola AWS IoT para crear el primer modelo de vehículo, no necesitas crear manualmente un catálogo de señales. Cuando creas tu primer modelo de vehículo, el AWS IoT crea FleetWise automáticamente un catálogo de señales para ti. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
   > - AWS FleetWise Actualmente, IoT admite un catálogo de señales para cada AWS cuenta por Región de AWS.

2. Utilice las señales del catálogo de señales para crear un modelo de vehículo. Para obtener más información, consulte Creación de un modelo de vehículo.

> **Note**
>
> - Si utiliza la FleetWise consola de AWS IoT para crear un modelo de vehículo, puede cargar archivos.dbc para importar señales. .dbc es un formato de archivo compatible con las bases de datos de la red de área del controlador (bus CAN). Una vez creado el modelo de vehículo, las nuevas señales se agregan automáticamente al catálogo de señales. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
> - Si utiliza la operación de la API `CreateModelManifest` para crear un modelo de vehículo, debe usar la operación de la API `UpdateModelManifest` para activarlo. Para obtener más información, consulte Actualización de un modelo de vehículo (AWS CLI).
> - Si utilizas la FleetWise consola de AWS IoT para crear un modelo de vehículo, AWS IoT activa FleetWise automáticamente el modelo de vehículo por ti.

3. Cree un manifiesto del decodificador. El manifiesto del decodificador contiene información de decodificación para cada señal especificada en el modelo de vehículo creado en el paso anterior. El manifiesto del decodificador está asociado al modelo de vehículo creado. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

> **Note**
>
> - Si usa la operación de la API `CreateDecoderManifest` para crear un manifiesto del decodificador, debe usar la operación de la API `UpdateDecoderManifest` para activarlo. Para obtener más información, consulte Actualización de un manifiesto del decodificador (AWS CLI).
> - Si utilizas la FleetWise consola de AWS IoT para crear un manifiesto de decodificador, AWS IoT lo activa FleetWise automáticamente.

Las bases de datos de bus CAN admiten el formato de archivo .dbc. Puede cargar archivos .dbc para importar señales y señales del decodificador. Para obtener un archivo .dbc de ejemplo, haga lo siguiente.

Para obtener un archivo .dbc:

1. Descarga el archivo.zipEngineSignals.
2. Desplácese hasta el directorio en el que descargó el archivo `EngineSignals.zip`.
3. Descomprima el contenido y guárdelo localmente como `EngineSignals.dbc`.

Temas

- Creación y administración de catálogos de señales
- Creación y administración de modelos de vehículo
- Creación y administración de manifiestos del decodificador

Creación y administración de catálogos de señales

> **Note**
>
> Puede descargar un script de demostración para convertir los mensajes ROS 2 en archivos JSON de VSS compatibles con el catálogo de señales. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.

Un catálogo de señales es una colección de señales estandarizadas que se pueden reutilizar para crear modelos de vehículos. AWS FleetWise El IoT es compatible con la especificación de señales de vehículos (VSS) que puede seguir para definir señales. Una señal puede ser de cualquiera de los siguientes tipos:

Atributo

Los atributos representan información estática que, por lo general, no cambia, como el fabricante y la fecha de fabricación.

Rama

Las ramificaciones representan señales en una estructura anidada. Las ramificaciones muestran las jerarquías de señales. Por ejemplo, la ramificación `Vehicle` tiene una ramificación secundaria, `Powertrain`. La ramificación `Powertrain` tiene una ramificación secundaria, `combustionEngine`. Para localizar la ramificación `combustionEngine`, utilice la expresión `Vehicle.Powertrain.combustionEngine`.

Sensor

Los datos de los sensores indican el estado actual del vehículo y cambian con el tiempo, a medida que el estado del vehículo cambia también, como los niveles de líquido, las temperaturas, las vibraciones o el voltaje, por ejemplo.

Actuador

Los datos del actuador indican el estado de los dispositivos del vehículo, como los motores, la calefacción y las cerraduras de las puertas. Al cambiar el estado del dispositivo de un vehículo, es posible actualizar los datos del actuador. Por ejemplo, puede definir un actuador para que represente la calefacción. El actuador recibe datos nuevos al encender o apagar la calefacción.

Estructura personalizada

Una estructura personalizada (también conocida como estructura) representa una estructura de datos compleja o de orden superior. Facilita el enlace lógico o la agrupación de datos que se originan en la misma fuente. Una estructura se utiliza cuando los datos se leen o escriben en una operación atómica, por ejemplo, para representar un tipo de datos complejo o una forma de orden superior.

Una señal de tipo estructura se define en el catálogo de señales mediante una referencia a un tipo de datos de estructura en lugar de a un tipo de datos primitivo. Las estructuras se pueden utilizar para todo tipo de señales, incluidos los sensores, los atributos, los actuadores y los tipos de datos de sistemas de visión. Si se envía o recibe una señal de tipo estructural, AWS IoT FleetWise espera que todos los elementos incluidos tengan valores válidos, por lo que todos los elementos son obligatorios. Por ejemplo, si una estructura contiene los elementos Vehicle.Camera.Image.height, Vehicle.Camera.Image.width y Vehicle.Camera.Image.data, se espera que la señal enviada contenga valores para todos estos elementos.

> **Note**
>
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Propiedad personalizada

Una propiedad personalizada representa un miembro de la estructura de datos compleja. El tipo de datos de la propiedad puede ser primitivo u otra estructura.

Al representar una forma de orden superior mediante una estructura y una propiedad personalizada, la forma de orden superior deseada siempre se define y visualiza como una estructura de árbol. La propiedad personalizada se usa para definir todos los nodos de hoja, mientras que la estructura se usa para definir todos los nodos que no son de hoja.

> **Note**
>
> - Si utilizas la FleetWise consola AWS IoT para crear el primer modelo de vehículo, no necesitas crear manualmente un catálogo de señales. Cuando creas tu primer modelo de vehículo, el AWS IoT crea FleetWise automáticamente un catálogo de señales para ti. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
> - Si utiliza la FleetWise consola de AWS IoT para crear un modelo de vehículo, puede cargar archivos.dbc para importar señales. .dbc es un formato de archivo compatible con las bases de datos de la red de área del controlador (bus CAN). Una vez creado el modelo de vehículo, las nuevas señales se agregan automáticamente al catálogo de señales. Para obtener más información, consulte Creación de un modelo de vehículo (consola).
> - AWS FleetWise Actualmente, IoT admite un catálogo de señales para Cuenta de AWS cada región.

AWS El IoT FleetWise proporciona las siguientes operaciones de API que puede usar para crear y administrar catálogos de señales.

- CreateSignalCatalog— Crea un nuevo catálogo de señales.
- ImportSignalCatalog— Importa señales para crear un catálogo de señales cargando un archivo JSON. Las señales deben definirse siguiendo la VSS y guardarse en formato JSON.
- UpdateSignalCatalog— Actualiza un catálogo de señales existente actualizando, eliminando o añadiendo señales.
- DeleteSignalCatalog— Elimina un catálogo de señales existente.
- ListSignalCatalogs— Recupera una lista paginada de resúmenes de todos los catálogos de señales.
- ListSignalCatalogNodes— Recupera una lista paginada de resúmenes de todas las señales (nodos) de un catálogo de señales determinado.
- GetSignalCatalog— Recupera información sobre un catálogo de señales.

Tutoriales

- Configuración de señales
- Creación de un catálogo de señales (AWS CLI)
- Importación de un catálogo de señales
- Actualización de un catálogo de señales (AWS CLI)
- Eliminación de un catálogo de señales (AWS CLI)
- Obtención de información del catálogo de señales (AWS CLI)

Configuración de señales

Esta sección muestra cómo configurar ramificaciones, atributos, sensores y actuadores.

Temas

- Configurar ramificaciones
- Configuración de atributos
- Configuración de sensores o actuadores
- Configuración de tipos de datos complejos

Configurar ramificaciones

Para agregar una ramificación, especifique la siguiente información:

- `fullyQualifiedName`: el nombre completo de la ramificación es la ruta a la ramificación más el nombre de dicha ramificación. Use un punto (.) para hacer referencia a una ramificación secundaria. Por ejemplo, `Vehicle.Chassis.SteeringWheel` es el nombre completo de la ramificación `SteeringWheel`. `Vehicle.Chassis.` es la ruta a dicha ramificación.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, dos puntos (:) y guion bajo (_).
- (Opcional) `Description`: la descripción de la ramificación.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `deprecationMessage`: el mensaje de obsolescencia del nodo o la ramificación que se va a mover o eliminar.

El mensaje de obsolescencia puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

- (Opcional) `comment`: un comentario además de la descripción. Puede utilizarse un comentario para proporcionar información adicional sobre la ramificación, como su razón de ser o referencias a ramificaciones relacionadas.

  El comentario puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

Configuración de atributos

Para configurar un atributo, especifique la siguiente información:

- `dataType`— El tipo de datos del atributo debe ser uno de los siguientes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, INT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, UNKNOWN o una estructura personalizada definida en la rama de tipos de datos. fullyQualifiedName
- `fullyQualifiedName`: el nombre completo del atributo es la ruta al atributo más el nombre del atributo. Utilice un punto (.) para hacer referencia a una señal secundaria. Por ejemplo, `Vehicle.Chassis.SteeringWheel.Diameter` es el nombre completo del atributo `Diameter`. `Vehicle.Chassis.SteeringWheel.` es la ruta a dicho atributo.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
- (Opcional) `Description`: la descripción del atributo.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `unit`: la unidad científica del atributo, como km o grados Celsius.
- (Opcional) `min`: el valor mínimo del atributo.
- (Opcional) `max`: el valor máximo del atributo.
- (Opcional) `defaultValue`: el valor predeterminado del atributo.
- (Opcional) `assignedValue`: el valor asignado al atributo.

- (Opcional) `allowedValues`: una lista de valores que acepta el atributo.
- (Opcional) `deprecationMessage`: el mensaje de obsolescencia del nodo o la ramificación que se va a mover o eliminar.

  El mensaje de obsolescencia puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `comment`: un comentario además de la descripción. Puede utilizarse un comentario para proporcionar información adicional sobre el atributo, como su justificación o referencias a atributos relacionados.

  El comentario puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

Configuración de sensores o actuadores

Para configurar un sensor o actuador, especifique la siguiente información.

- `dataType`— El tipo de datos de la señal debe ser uno de los siguientes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, INT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, UNKNOWN o una estructura personalizada definida en la rama de tipos de datos. fullyQualifiedName
- `fullyQualifiedName`: el nombre completo de la señal es la ruta a la señal más el nombre de la señal. Utilice un punto (.) para hacer referencia a una señal secundaria. Por ejemplo, `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` es el nombre completo del actuador `HandsOffSteeringState`. `Vehicle.Chassis.SteeringWheel.HandsOff.` es la ruta a dicho actuador.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
- (Opcional) `Description`: la descripción de la señal.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `unit`: la unidad científica de la señal, como km o grados Celsius.
- (Opcional) `min`: el valor mínimo de la señal.

- (Opcional) `max`: el valor máximo de la señal.
- (Opcional) `assignedValue`: el valor asignado a la señal.
- (Opcional) `allowedValues`: lista de valores que acepta la señal.
- (Opcional) `deprecationMessage`: el mensaje de obsolescencia del nodo o la ramificación que se va a mover o eliminar.

  El mensaje de obsolescencia puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `comment`: un comentario además de la descripción. Puede utilizarse un comentario para proporcionar información adicional sobre el sensor o el actuador, como su justificación o referencias a sensores o actuadores relacionados.

  El comentario puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

Configuración de tipos de datos complejos

Los tipos de datos complejos se utilizan al modelar sistemas de visión. Además de ramificaciones, estos tipos de datos se componen de estructuras y propiedades. Una estructura es una señal que se describe mediante varios valores, como una imagen. Una propiedad representa un miembro de la estructura, como un tipo de datos primitivo (como UINT8) u otra estructura (como una marca temporal). Por ejemplo, Vehicle.Cameras.Front representa una ramificación, Vehicle.Cameras.Front.Image representa una estructura y Vehicle.Cameras.Timestamp representa una propiedad.

El siguiente ejemplo de tipo de datos complejo muestra cómo se exportan las señales y los tipos de datos a un único archivo JSON.

Example tipos de datos complejos

```
{
  "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**
>
> Puede descargar un script de demostración para convertir los mensajes ROS 2 en archivos JSON de VSS compatibles con el catálogo de señales. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Configuración de estructura

Para configurar una estructura personalizada, especifique la siguiente información.

- `fullyQualifiedName`: el nombre totalmente cualificado de la estructura personalizada. Por ejemplo, el nombre totalmente cualificado de una estructura personalizada podría ser `ComplexDataTypes.VehicleDataTypes.SVMCamera`.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).

- (Opcional) `Description`: la descripción de la señal.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `deprecationMessage`: el mensaje de obsolescencia del nodo o la ramificación que se va a mover o eliminar.

  El mensaje de obsolescencia puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `comment`: un comentario además de la descripción. Puede utilizarse un comentario para proporcionar información adicional sobre el sensor o el actuador, como su justificación o referencias a sensores o actuadores relacionados.

  El comentario puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

Configuración de la propiedad

Para configurar una propiedad personalizada, especifique la siguiente información.

- `dataType`: el tipo de datos de la señal debe ser uno de los siguientes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, STRUCT, STRUCT_ARRAY o UNKNOWN.
- `fullyQualifiedName`: el nombre totalmente cualificado de la propiedad personalizada. Por ejemplo, el nombre totalmente cualificado de una propiedad personalizada podría ser `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
- (Opcional) `Description`: la descripción de la señal.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `deprecationMessage`: el mensaje de obsolescencia del nodo o la ramificación que se va a mover o eliminar.

El mensaje de obsolescencia puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

- (Opcional) `comment`: un comentario además de la descripción. Puede utilizarse un comentario para proporcionar información adicional sobre el sensor o el actuador, como su justificación o referencias a sensores o actuadores relacionados.

  El comentario puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).
- (Opcional) `dataEncoding`: indica si la propiedad es de datos binarios. La codificación de datos de la propiedad personalizada debe ser una de las siguientes: BINARY o TYPED.
- (Opcional)`structFullyQualifiedName` : el nombre completo del nodo de estructura (estructura) de la propiedad personalizada si el tipo de datos de la propiedad personalizada es Estructura o StructArray.

  El nombre completo puede tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).

Creación de un catálogo de señales (AWS CLI)

Puede utilizar la operación de CreateSignalCatalogAPI para crear un catálogo de señales. En el siguiente ejemplo se utiliza AWS CLI.

Para crear un catálogo de señales, ejecute el siguiente comando.

*`signal-catalog-configuration`*Sustitúyalo por el nombre del archivo JSON que contiene la configuración.

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

- *`signal-catalog-name`*Sustitúyalo por el nombre del catálogo de señales que está creando.
- (Opcional) Reemplace *`description`* por una descripción que lo ayude a identificar el catálogo de señales.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de señales.

```
{
    "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**
>
> Puede descargar un script de demostración para convertir los mensajes ROS 2 en archivos JSON de VSS compatibles con el catálogo de señales. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Importación de un catálogo de señales

Puede usar la FleetWise consola o la API de AWS IoT para importar un catálogo de señales.

Temas

- Importación de un catálogo de señales (consola)
- Importación de un catálogo de señales (AWS CLI)

Importación de un catálogo de señales (consola)

Puede usar la FleetWise consola de AWS IoT para importar un catálogo de señales.

> **Important**
>
> Puede tener un catálogo de señales como máximo. Si ya tiene un catálogo de señales, no verá la opción para importar uno en la consola.

Para importar un catálogo de señales:

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Catálogo de señales.
3. En la página de resumen del catálogo de señales, elija Importar catálogo de señales.
4. Importe el archivo que contiene las señales.
   - Para cargar un archivo en un bucket de S3:
     a. Elija Import from S3 (Importar desde S3).
     b. Elija Browse S3 (Examinar S3).

c. En Buckets, introduzca el nombre o el objeto del bucket, selecciónelo de la lista y, a continuación, elija el archivo de la lista. Pulse el botón Elegir archivo.

O bien, en URI de S3, introduzca un URI de Amazon Simple Storage Service. Para obtener más información, consulte Métodos para acceder a un bucket en la Guía del usuario de Amazon S3.

- Para cargar un archivo desde el equipo:

a. Elija Importar de un archivo.

b. Cargue un archivo .json en formato de especificación de señales de vehículos (VSS).

5. Compruebe el catálogo de señales y, a continuación, elija Importar archivo.

Importación de un catálogo de señales (AWS CLI)

Puede utilizar la operación de ImportSignalCatalogAPI para cargar un archivo JSON que ayude a crear un catálogo de señales. Debe seguir la especificación de señales de vehículos (VSS) para guardar las señales en el archivo JSON. En el siguiente ejemplo se utiliza AWS CLI.

Para importar un catálogo de señales, ejecute el siguiente comando.

- *signal-catalog-name*Sustitúyalo por el nombre del catálogo de señales que está creando.
- (Opcional) Reemplace description por una *descripción* que lo ayude a identificar el catálogo de señales.
- *signal-catalog-configuration-vss*Sustitúyalo por el nombre del archivo de cadenas JSON que contiene las señales definidas en VSS.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de señales.

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

El JSON debe estar en cadena y transferido a través del campo `vssJson`. A continuación, se muestra un ejemplo de las señales definidas en la VSS.

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

}
```

El siguiente ejemplo muestra las mismas señales definidas en la VSS en una cadena JSON.

```
{
   "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**
>
> Puede descargar un script de demostración para convertir los mensajes ROS 2 en archivos JSON de VSS compatibles con el catálogo de señales. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Actualización de un catálogo de señales (AWS CLI)

Puede utilizar la operación de UpdateSignalCatalogAPI para actualizar un catálogo de señales existente. En el siguiente ejemplo se utiliza AWS CLI.

Para actualizar un catálogo de señales existente, ejecute el siguiente comando.

*signal-catalog-configuration*Sustitúyalo por el nombre del archivo JSON que contiene la configuración.

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

*signal-catalog-name*Sustitúyalo por el nombre del catálogo de señales que está actualizando.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de señales.

> **Important**
>
> Las estructuras personalizadas son inmutables. Si necesita reordenar o insertar propiedades en una estructura personalizada existente, elimine la estructura y cree una estructura nueva con el orden de propiedades deseado.
> Para eliminar una estructura personalizada, agregue el nombre totalmente cualificado de la estructura en `nodesToRemove`. No se puede eliminar una estructura si alguna señal hace referencia a ella. Todas las señales que hagan referencia a la estructura (el tipo de datos se define como la estructura de destino) se deben actualizar o eliminar antes de solicitar la actualización del catálogo de señales.

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

Eliminación de un catálogo de señales (AWS CLI)

Puede utilizar la operación DeleteSignalCatalogAPI para eliminar un catálogo de señales. En el siguiente ejemplo se utiliza AWS CLI.

> **Important**
>
> Antes de eliminar un catálogo de señales, asegúrese de que no tenga modelos de vehículo, manifiestos del decodificador, vehículos, flotas ni campañas asociados. Para obtener instrucciones, consulte lo siguiente:
>
> - Eliminación de un modelo de vehículo
> - Eliminación de un manifiesto del decodificador
> - Eliminación de un vehículo
> - Eliminación de una flota (AWS CLI)
> - Eliminación de una campaña

Para eliminar un catálogo de señales existente, ejecute el siguiente comando. *signal-catalog-name*Sustitúyalo por el nombre del catálogo de señales que va a eliminar.

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

> **Note**
>
> Este comando no proporciona ningún resultado.

Obtención de información del catálogo de señales (AWS CLI)

Puede utilizar la operación de la ListSignalCatalogsAPI para comprobar si se ha eliminado un catálogo de señales. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todos los catálogos de señales, ejecute el siguiente comando.

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

Puede utilizar la operación ListSignalCatalogNodesAPI para comprobar si se ha actualizado un catálogo de señales. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todas las señales (nodos) de un catálogo de señales determinado, ejecute el siguiente comando.

*signal-catalog-name*Sustitúyalo por el nombre del catálogo de señales que está comprobando.

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

Puede utilizar la operación de la GetSignalCatalogAPI para recuperar la información del catálogo de señales. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar información sobre un catálogo de señales, ejecute el siguiente comando.

*signal-catalog-name*Sustitúyalo por el nombre del catálogo de señales que desea recuperar.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en el catálogo de señales podrían no reflejarse inmediatamente.

Creación y administración de modelos de vehículo

Puede utilizar las señales para crear modelos de vehículo que lo ayudarán a estandarizar el formato de los vehículos. Los modelos de vehículo permiten hacer que la información sea coherente en varios vehículos del mismo tipo, a fin de que pueda procesar los datos de las flotas de vehículos. Los vehículos creados a partir del mismo modelo de vehículo heredan el mismo grupo de señales. Para obtener más información, consulte Creación, aprovisionamiento y administración de vehículos.

Cada modelo de vehículo tiene un campo de estado que contiene el estado de dicho modelo de vehículo. El estado puede ser uno de los siguientes valores:

- `ACTIVE`: el modelo de vehículo está activo.
- `DRAFT`: se guarda la configuración del modelo de vehículo.

> **Important**
>
> - Si quiere utilizar la operación de la API `CreateModelManifest` para crear el primer modelo de vehículo, primero debe crear un catálogo de señales. Para obtener más información, consulte Creación de un catálogo de señales (AWS CLI).
> - Si utilizas la FleetWise consola de AWS IoT para crear un modelo de vehículo, AWS IoT activa FleetWise automáticamente el modelo de vehículo por ti.
> - Si utiliza la operación de la API `CreateModelManifest` para crear un modelo de vehículo, el modelo de vehículo permanece en el estado `DRAFT`.
> - No puede crear vehículos a partir de modelos de vehículo que se encuentren en el estado `DRAFT`. Utilice la operación de la API `UpdateModelManifest` para cambiar los modelos de vehículo al estado `ACTIVE`.
> - No puede editar los modelos de vehículo que se encuentren en el estado `ACTIVE`.

Temas

Creación de un modelo de vehículo

Puedes usar la FleetWise consola de AWS IoT o la API para crear modelos de vehículos.

> **Important**
>
> Para poder crear un modelo de vehículo mediante la operación de la API `CreateModelManifest`, debe disponer de un catálogo de señales.

Temas

Creación de un modelo de vehículo (consola)

En la FleetWise consola de AWS IoT, puede crear un modelo de vehículo de las siguientes maneras:

Utilice una plantilla proporcionada por AWS

AWS FleetWise El IoT proporciona una plantilla de diagnóstico integrado (OBD) II, J1979, que crea automáticamente un catálogo de señales, un modelo de vehículo y un manifiesto del decodificador. La plantilla también agrega las interfaces de red OBD al manifiesto del decodificador. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Para crear un modelo de vehículo mediante una plantilla:

1. Navegue hasta la FleetWiseconsola de AWS IoT.

2. En el panel de navegación, elija Modelos de vehículo.
3. En la página Modelos de vehículo, elija Agregar la plantilla proporcionada.
4. Elija Diagnóstico a bordo (OBD) II.
5. Introduzca un nombre para la interfaz de red OBD que FleetWise está creando el AWS IoT.
6. Elija Añadir.

Creación manual de un modelo de vehículo

Puede agregar señales del catálogo de señales o importarlas cargando uno o más archivos .dbc. Un archivo .dbc es un formato de archivo admitido por las bases de datos de la red de área de control (bus CAN).

> **Important**
>
> No se puede crear un modelo de vehículo con señales de datos del sistema de visión mediante la FleetWise consola de AWS IoT. En su lugar, utilice el AWS CLI para crear un modelo de vehículo.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Para crear manualmente un modelo de vehículo:

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. En la página Modelos de vehículo, elija Crear modelo de vehículo y, a continuación, haga lo siguiente.

Temas

Paso 1: Configurar el modelo de vehículo

En la sección Información general, haga lo siguiente:

1. Introduzca un nombre para el modelo de vehículo.
2. (Opcional) Introduzca una descripción.
3. Elija Siguiente.

Paso 2: Agregar señales

> **Note**
>
> - Si es la primera vez que utilizas AWS IoT FleetWise, este paso no estará disponible hasta que tengas un catálogo de señales. Cuando se crea el primer modelo de vehículo, el AWS IoT crea FleetWise automáticamente un catálogo de señales con señales añadidas al primer modelo de vehículo.
> - Si tiene experiencia con el AWS IoT FleetWise, puede añadir señales al modelo de su vehículo seleccionando señales del catálogo de señales o cargando archivos.dbc para importar señales.
> - Para poder crear un modelo de vehículo, debe tener al menos una señal.

Para agregar señales:

1. Elija una o más señales del catálogo de señales que vaya a agregar al modelo de vehículo. Puede revisar las señales seleccionadas en el panel derecho.

   > **Note**
   >
   > Solo las señales seleccionadas se agregarán al modelo de vehículo.

2. Elija Siguiente.

Paso 3: Importar señales

> **Note**
>
> - Si es la primera vez que utiliza AWS IoT FleetWise, debe cargar al menos un archivo.dbc para importar señales.
> - Si tiene experiencia con el AWS IoT FleetWise, puede añadir señales al modelo de su vehículo seleccionando señales del catálogo de señales o cargando archivos.dbc para importar señales.
> - Para poder crear un modelo de vehículo, debe tener al menos una señal.

Para importar señales:

1. Elija Elegir archivos.
2. En el cuadro de diálogo, elija el archivo .dbc que contiene las señales. Puede cargar varios archivos .dbc a la vez.
3. AWS IoT FleetWise analiza sus archivos.dbc para recuperar señales.

   En la sección Señales, especifique los siguientes metadatos para cada señal:

   - Nombre: el nombre de la señal.

     El nombre de la señal debe ser único. El nombre de la señal más la ruta pueden tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
   - Tipo de datos: el tipo de datos de la señal debe ser INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY o UNKNOWN.
   - Tipo de señal: el tipo de señal, que puede ser sensor o actuador.
   - (Opcional) Unidad: la unidad científica de la señal, como km o grados Celsius.
   - (Opcional) Ruta: la ruta a la señal. Al igual que en JSONPath, utilice un punto (.) para hacer referencia a una señal secundaria. Por ejemplo, **`Vehicle.Engine.Light`**.

     El nombre de la señal más la ruta pueden tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).

- (Opcional) Mín.: el valor mínimo de la señal.
- (Opcional) Máx.: el valor máximo de la señal.
- (Opcional) Descripción: la descripción de la señal.

  La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

4. Elija Siguiente.

(Opcional) Paso 4: Agregar atributos

Puede agregar hasta 100 atributos, incluidos los atributos existentes en el catálogo de señales.

Para agregar atributos:

1. En Agregar atributos, especifique los siguientes metadatos para cada atributo:

   - Nombre: el nombre del atributo.

     El nombre de la señal debe ser único. El nombre y la ruta de la señal pueden tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
   - Tipo de datos: el tipo de datos del atributo debe ser INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, UINT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY o UNKNOWN.
   - (Opcional) Unidad: la unidad científica del atributo, como km o grados Celsius.
   - (Opcional) Ruta: la ruta a la señal. Al igual que en JSONPath, utilice un punto (.) para hacer referencia a una señal secundaria. Por ejemplo, **`Vehicle.Engine.Light`**.

     El nombre de la señal más la ruta pueden tener hasta 150 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo).
   - (Opcional) Mín.: valor mínimo del atributo.
   - (Opcional) Máx.: el valor máximo del atributo.
   - (Opcional) Descripción: la descripción del atributo.

     La descripción puede tener hasta 2048 caracteres. Caracteres válidos: a-z, A-Z, 0-9, : (dos puntos), _ (guion bajo) y - (guion).

2. Elija Siguiente.

Paso 5: Revisar y crear

Compruebe las configuraciones del modelo de vehículo y, a continuación, elija Crear.

Duplicación de un modelo de vehículo

AWS El IoT FleetWise puede copiar las configuraciones de un modelo de vehículo existente para crear un modelo nuevo. Las señales especificadas en el modelo de vehículo seleccionado se copian en el nuevo modelo de vehículo.

Para duplicar un modelo de vehículo:

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. Elija un modelo de la lista de modelos de vehículo y, a continuación, elija Modelo duplicado.

Para configurar el modelo del vehículo, siga el tutorial Creación manual de un modelo de vehículo.

El AWS IoT puede tardar unos minutos en FleetWise procesar tu solicitud para crear el modelo de vehículo. Una vez que el modelo de vehículo se haya creado correctamente, la columna Estado en la página Modelos de vehículo se mostrará como ACTIVO. Cuando el modelo de vehículo se activa, no puede editarlo.

Creación de un modelo de vehículo (AWS CLI)

Puede utilizar la operación de la CreateModelManifestAPI para crear modelos de vehículos (manifiestos de modelos). El siguiente ejemplo utiliza AWS CLI.

> **Important**
>
> Si desea utilizar la FleetWise API de AWS IoT para crear el primer modelo de vehículo, primero debe crear un catálogo de señales. Para obtener más información acerca de cómo crear un catálogo de señales, consulte Creación de un catálogo de señales (AWS CLI).

Para crear un modelo de vehículo, ejecute el siguiente comando.

*vehicle-model-configuration*Sustitúyalo por el nombre del archivo JSON que contiene la configuración.

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

- *vehicle-model-name*Reemplácelo por el nombre del modelo de vehículo que está creando.
- Reemplace *signal-catalog-ARN* por el Nombre de recurso de Amazon (ARN) del catálogo de señales.
- (Opcional) Reemplace *description* por una descripción que lo ayude a identificar el modelo de vehículo.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de señales.

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

Actualización de un modelo de vehículo (AWS CLI)

Puedes usar la operación de la UpdateModelManifestAPI para actualizar un modelo de vehículo existente (manifiestos de modelos). El siguiente ejemplo utiliza AWS CLI.

Para actualizar un modelo de vehículo existente, ejecute el siguiente comando:

*update-vehicle-model-configuration*Sustitúyalo por el nombre del archivo JSON que contiene la configuración.

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

- *vehicle-model-name*Reemplácelo por el nombre del modelo de vehículo que está actualizando.

- (Opcional) Para activar el modelo de vehículo, *vehicle-model-status*sustitúyalo porACTIVE.

> **Important**
>
> Una vez que el modelo de vehículo esté activado, no podrá cambiarlo.

- (Opcional) Reemplace *description* por una descripción actualizada para ayudarlo a identificar el modelo de vehículo.

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

Eliminación de un modelo de vehículo

Puedes usar la FleetWise consola de AWS IoT o la API para eliminar modelos de vehículos.

> **Important**
>
> Los vehículos y los manifiestos del decodificador asociados al modelo de vehículo deben eliminarse primero. Para más información, consulte Eliminación de un vehículo y Eliminación de un manifiesto del decodificador.

Eliminación de un modelo de vehículo (consola)

Para eliminar un modelo de vehículo, usa la FleetWise consola de AWS IoT.

Para eliminar un modelo de vehículo:

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. En la página Modelos de vehículo, elija el modelo de vehículo de destino.
4. Elija Eliminar.

5. En ¿Eliminar **`vehicle-model-name`**?, introduzca el nombre del modelo de vehículo que desea eliminar y, a continuación, elija Confirmar.

Eliminación de un modelo de vehículo (AWS CLI)

Puedes usar la operación de la DeleteModelManifestAPI para eliminar un modelo de vehículo existente (manifiestos de modelo). El siguiente ejemplo utiliza AWS CLI.

Para eliminar un modelo de vehículo, ejecute el siguiente comando:

*`model-manifest-name`*Sustitúyelo por el nombre del modelo de vehículo que vaya a eliminar.

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

> **Note**
>
> Este comando no proporciona ningún resultado.

Obtención de información sobre el modelo de vehículo (AWS CLI)

Puedes usar la operación de la ListModelManifestsAPI para verificar si se ha eliminado un modelo de vehículo. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todos los modelos de vehículo, ejecute el siguiente comando.

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

Puede utilizar la operación de la ListModelManifestNodesAPI para verificar si se ha actualizado un modelo de vehículo. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todas las señales (nodos) de un modelo de vehículo determinado, ejecute el siguiente comando:

*`vehicle-model-name`*Reemplácelo por el nombre del modelo de vehículo que está comprobando.

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

```
                --name vehicle-model-name
```

Para recuperar información sobre un modelo de vehículo, ejecute el siguiente comando:

Reemplace *vehicle-model* por el nombre del modelo de vehículo que desea recuperar.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en el modelo de vehículo podrían no reflejarse inmediatamente.

Creación y administración de manifiestos del decodificador

Los manifiestos del decodificador contienen información de decodificación que el AWS IoT FleetWise utiliza para transformar los datos del vehículo (datos binarios) en valores legibles por humanos y para preparar los datos para el análisis de datos. La interfaz de red y las señales del decodificador son los componentes principales con los que se trabaja para configurar los manifiestos del decodificador.

Interfaz de red

Contiene información sobre el protocolo que utiliza la red integrada en el vehículo. AWS El IoT FleetWise admite los siguientes protocolos.

Red de área de control (bus CAN)

Protocolo que define cómo se comunican los datos entre las unidades de control electrónico (ECU). Las ECU pueden ser la unidad de control del motor, los airbags o el sistema de audio.

Diagnóstico a bordo (OBD) II

Un protocolo perfeccionado que define cómo se comunican los datos de autodiagnóstico entre las ECU. Proporciona una serie de códigos de diagnóstico de problemas (DTC) estándar que ayudan a identificar qué es lo que falla en el vehículo.

Middleware de vehículos

El middleware de vehículos se define como un tipo de interfaz de red. Algunos ejemplos de middleware de vehículos incluyen Robot Operating System (ROS 2) y el middleware escalable orientado a servicios sobre IP (SOME/IP).

> **Note**
>
> AWS IoT FleetWise es compatible con el middleware ROS 2 para los datos del sistema de visión.

Señal del decodificador

Proporciona información de decodificación detallada para una señal específica. Todas las señales especificadas en el modelo de vehículo deben estar emparejadas con una señal del decodificador. Si el manifiesto del decodificador contiene interfaces de red CAN, debe contener señales del decodificador CAN. Si el manifiesto del decodificador contiene interfaces de red OBD, debe contener señales del decodificador OBD.

El manifiesto del decodificador debe contener las señales del decodificador de mensajes si también contiene interfaces de middleware de vehículos.

Cada manifiesto del decodificador debe estar asociado a un modelo de vehículo. AWS El IoT FleetWise utiliza el manifiesto del decodificador asociado para decodificar los datos de los vehículos creados en función del modelo del vehículo.

Cada manifiesto del decodificador incluye un campo que indica su estado. El estado puede ser uno de los siguientes valores:

- `ACTIVE`: el manifiesto del decodificador está activo.
- `DRAFT`: la configuración del manifiesto del decodificador no se guarda.
- `VALIDATING`: el manifiesto del decodificador está en proceso de validación para determinar su idoneidad. Esto solo se aplica a los manifiestos del decodificador que contienen al menos una señal de datos de sistemas de visión.
- `INVALID`: el manifiesto del decodificador no se validó y aún no se puede activar. Esto solo se aplica a los manifiestos del decodificador que contienen al menos una señal de datos de sistemas de visión. Puede usar las GetDecoderManifest API ListDecoderManifests y para comprobar el motivo de una validación fallida.

Important

- Si utilizas la FleetWise consola de AWS IoT para crear un manifiesto de decodificador, AWS IoT lo activa FleetWise automáticamente.
- Si utiliza la operación de la API `CreateDecoderManifest` para crear un manifiesto del decodificador, este permanece en el estado `DRAFT`.
- No puede crear vehículos a partir de modelos de vehículo que estén asociados a un manifiesto del decodificador `DRAFT`. Utilice la operación de la API `UpdateDecoderManifest` para cambiar el manifiesto del decodificador al estado `ACTIVE`.
- No puede editar los manifiestos del decodificador que se encuentren en el estado `ACTIVE`.

Temas

Configuración de las interfaces de red y las señales del decodificador

Cada manifiesto del decodificador tiene al menos una interfaz de red y las señales del decodificador están emparejadas con las señales especificadas en el modelo de vehículo asociado.

Si el manifiesto del decodificador contiene interfaces de red CAN, debe contener señales del decodificador CAN. Si el manifiesto del decodificador contiene interfaces de red OBD, debe contener señales del decodificador OBD.

Temas

Configuración de interfaces de red

Para configurar una interfaz de red CAN, especifique la siguiente información:

- `name`: el nombre de la interfaz CAN.

  El nombre de la interfaz debe ser único y puede tener entre 1 y 100 caracteres.
- (Opcional) `protocolName`: el nombre del protocolo.

  Valores válidos: `CAN-FD` y `CAN`
- (Opcional)`protocolVersion`: AWS IoT FleetWise actualmente es compatible con CAN-FD y CAN 2.0b.

  Valores válidos: `1.0` y `2.0b`

Para configurar una interfaz de red OBD, especifique la siguiente información:

- `name`: el nombre de la interfaz OBD.

  El nombre de la interfaz debe ser único y puede tener entre 1 y 100 caracteres.
- `requestMessageId`: el ID del mensaje que solicita los datos.
- (Opcional) `dtcRequestIntervalSeconds`: con qué frecuencia se solicitan códigos de diagnóstico de problemas (DTC) al vehículo en segundos. Por ejemplo, si el valor especificado es 120, el software Edge Agent recopila los DTC almacenados cada dos minutos.
- (Opcional) `hasTransmissionEcu`: si el vehículo tiene un módulo de control de la transmisión (TCM).

  Valores válidos: `true` y `false`
- (Opcional)`obdStandard`: el estándar OBD FleetWise compatible con AWS IoT. AWS FleetWise Actualmente, el IoT es compatible con la norma ISO15765-4 de diagnóstico integrado de armonización mundial (WWH-OBD).
- (Opcional) `pidRequestIntervalSeconds`: con qué frecuencia se solicitan los PID de OBD II del vehículo. Por ejemplo, si el valor especificado es 120, el software Edge Agent recopila los PID de OBD II cada dos minutos.
- (Opcional) `useExtendedIds`: si se deben usar los ID extendidos en el mensaje.

  Valores válidos: `true` y `false`

Para configurar una interfaz de red de middleware de vehículos, especifique la siguiente información.

- `name`: el nombre de la interfaz de middleware del vehículo.

  El nombre de la interfaz debe ser único y puede tener entre 1 y 100 caracteres.
- `protocolName`: el nombre del protocolo.

  Valores válidos: `ROS_2`

Configuración de las señales del decodificador

Para configurar una señal del decodificador CAN, especifique la siguiente información:

- `factor`: el multiplicador que se utiliza para decodificar el mensaje.
- `isBigEndian`: si el orden de bytes del mensaje es big-endian. Si lo es, el valor más significativo de la secuencia se almacena primero, en la dirección de almacenamiento más baja.
- `isSigned`: si el mensaje está firmado. Si lo está, puede representar números positivos y negativos.
- `length`: la longitud total en bytes del mensaje.
- `messageId`: el ID del mensaje.
- `offset`: el desplazamiento utilizado para calcular el valor de la señal. Combinado con el factor, el cálculo es `value = raw_value * factor + offset`.
- `startBit`: indica la ubicación del primer bit del mensaje.
- (Opcional) `name`: el nombre de la señal.

Para configurar una señal del decodificador OBD, especifique la siguiente información:

- `byteLength`: la longitud total en bytes del mensaje.
- `offset`: el desplazamiento utilizado para calcular el valor de la señal. Combinado con el escalado, el cálculo es `value = raw_value * scaling + offset`.
- `pid`: el código de diagnóstico utilizado para solicitar un mensaje de un vehículo para esta señal.
- `pidResponseLength`: la longitud del mensaje solicitado.
- `scaling`: el multiplicador que se utiliza para decodificar el mensaje.
- `serviceMode`: el modo de funcionamiento (servicio de diagnóstico) de un mensaje.
- `startByte`: indica el principio del mensaje.

- (Opcional) `bitMaskLength`: la cantidad de bits que están enmascarados en un mensaje.
- (Opcional) `bitRightShift`: el número de posiciones desplazadas hacia la derecha.

Para configurar una señal del decodificador de mensajes, especifique la siguiente información.

- `topicName`: el nombre del tema de la señal del mensaje. Corresponde a los temas de ROS 2. Para obtener más información sobre el objeto de mensaje estructurado, consulte. StructuredMessage
- `structuredMessage`: el mensaje estructurado para la señal del mensaje. Se puede definir con una primitiveMessageDefinition, structuredMessageList Definición o de structuredMessageDefinition forma recursiva.

Creación de un manifiesto del decodificador

Puedes usar la FleetWise consola de AWS IoT o la API para crear un manifiesto de decodificador para tu modelo de vehículo.

> **Important**
>
> Para poder crear un manifiesto del decodificador, debe disponer de un modelo de vehículo. Cada manifiesto del decodificador debe estar asociado a un modelo de vehículo. Para obtener más información, consulte Creación y administración de modelos de vehículo.

Temas

- Creación de un manifiesto del decodificador (consola)
- Creación de un manifiesto del decodificador (AWS CLI)

Creación de un manifiesto del decodificador (consola)

Puedes usar la FleetWise consola de AWS IoT para crear un manifiesto de decodificador asociado al modelo de tu vehículo.

> **Important**
>
> No se pueden configurar las señales de datos del sistema de visión en los manifiestos del decodificador mediante la FleetWise consola de AWS IoT. En su lugar, utilice el AWS CLI. Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Para crear un manifiesto del decodificador:

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. Elija el modelo de vehículo de destino.
4. En la página de resumen del modelo de vehículo, elija Crear manifiesto del decodificador y, a continuación, haga lo siguiente.

Temas

Paso 1: Configurar el manifiesto del decodificador

En la sección Información general, haga lo siguiente:

1. Escriba un nombre único para el manifiesto del decodificador.
2. (Opcional) Introduzca una descripción.
3. Elija Siguiente.

Paso 2: Agregar interfaces de red

Cada manifiesto del decodificador debe tener al menos una interfaz de red. Puede agregar varias interfaces de red a un manifiesto del decodificador.

Creación de una interfaz de red

- En Interfaz de red, haga lo siguiente:

a. Para Tipo de interfaz de red, elija CAN_INTERFACE o OBD_INTERFACE.
b. Escriba un nombre único para la interfaz de red.
c. Introduzca un ID único para la interfaz de red. Puedes usar el ID generado por el AWS IoT FleetWise.
d. Seleccione una o más señales especificadas en el modelo de vehículo para emparejarlas con las señales del decodificador.
e. Para proporcionar información de decodificación, cargue un archivo .dbc. AWS IoT FleetWise analiza el archivo.dbc para recuperar las señales del decodificador.
f. En la sección Señales emparejadas, asegúrese de que todas las señales estén emparejadas con una señal del decodificador.
g. Elija Siguiente.

> **Note**
>
> - Puede cargar solo un archivo .dbc para cada interfaz de red.
> - Asegúrese de que todas las señales especificadas en el modelo de vehículo estén emparejadas con una señal del decodificador.
> - Una vez haya decidido agregar otra interfaz de red, no podrá editar la que esté editando. Puede eliminar cualquier interfaz de red existente.

Paso 3: Revisar y crear

Compruebe las configuraciones del manifiesto del decodificador y, a continuación, elija Crear.

Creación de un manifiesto del decodificador (AWS CLI)

Puede utilizar la operación de la CreateDecoderManifestAPI para crear manifiestos del decodificador. El siguiente ejemplo utiliza AWS CLI.

> **Important**
>
> Antes de crear un manifiesto del decodificador, cree primero un modelo de vehículo. Para obtener más información, consulte Creación de un modelo de vehículo.

Para crear un manifiesto del decodificador, ejecute el siguiente comando:

*decoder-manifest-configuration*Sustitúyalo por el nombre del archivo JSON que contiene la configuración.

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

- *decoder-manifest-name*Sustitúyalo por el nombre del manifiesto del decodificador que estás creando.
- Reemplace *vehicle-model-ARN* por el Nombre de recurso de Amazon (ARN) del modelo del vehículo.
- (Opcional) Reemplace *description* por una descripción que lo ayude a identificar el manifiesto del decodificador.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de las interfaces de red y las señales del decodificador.

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

- *decoder-manifest-name*Sustitúyalo por el nombre del manifiesto del decodificador que estás creando.
- Reemplace *vehicle-model-ARN* por el Nombre de recurso de Amazon (ARN) del modelo del vehículo.
- (Opcional) Reemplace *description* por una descripción que lo ayude a identificar el manifiesto del decodificador.

El orden de los nodos de propiedades dentro de una estructura debe permanecer coherente, tal como se define en el catálogo de señales y en el modelo del vehículo (manifiesto del modelo). Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configuración de las interfaces de red y las señales del decodificador.

```
{
```

```
"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**
>
> Puede descargar un script de demostración para crear un manifiesto del decodificador con las señales del sistema de visión. Para obtener más información, consulte la Guía para desarrolladores de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Actualización de un manifiesto del decodificador (AWS CLI)

Puedes usar la operación de la UpdateDecoderManifestAPI para actualizar un manifiesto de decodificador. Puede agregar, eliminar y actualizar interfaces de red y decodificadores de señal. También puede cambiar el estado del manifiesto del decodificador. El siguiente ejemplo utiliza AWS CLI.

Para actualizar el manifiesto del decodificador, ejecute el siguiente comando:

*decoder-manifest-name*Sustitúyelo por el nombre del manifiesto del decodificador que estás actualizando.

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

> **Important**
>
> Una vez activado el manifiesto del decodificador, no puede editarlo.

Eliminación de un manifiesto del decodificador

Puedes usar la FleetWise consola o la API de AWS IoT para eliminar un manifiesto de decodificador.

> **Important**
>
> Los vehículos asociados al manifiesto del decodificador deben eliminarse primero. Para obtener más información, consulte Eliminación de un vehículo.

Temas

Eliminación de un manifiesto del decodificador (consola)

Puedes usar la FleetWise consola de AWS IoT para eliminar un manifiesto de decodificador.

Para eliminar un manifiesto del decodificador:

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Modelos de vehículo.
3. Elija el modelo de vehículo de destino.
4. En la página de resumen del modelo de vehículo, elija la pestaña Manifiestos del decodificador.
5. Elija el manifiesto del decodificador de destino y, a continuación, elija Eliminar.
6. En ¿Eliminar **`decoder-manifest-name`**?, introduzca el nombre del manifiesto del decodificador que desea eliminar y, a continuación, elija Confirmar.

Eliminación de un manifiesto del decodificador (AWS CLI)

Puedes usar la operación de la DeleteDecoderManifestAPI para eliminar el manifiesto de un decodificador. En el siguiente ejemplo se utiliza AWS CLI.

> **Important**
>
> Antes de eliminar el manifiesto del decodificador, elimine primero los vehículos asociados. Para obtener más información, consulte Eliminación de un vehículo.

Para eliminar un manifiesto del decodificador, ejecute el siguiente comando.

*`decoder-manifest-name`*Sustitúyalo por el nombre del manifiesto del decodificador que vas a eliminar.

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

Obtención de información sobre un manifiesto del decodificador (AWS CLI)

Puede utilizar la operación de la ListDecoderManifestsAPI para comprobar si se ha eliminado un manifiesto del decodificador. En el siguiente ejemplo se utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todos los manifiestos del decodificador, ejecute el siguiente comando:

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

Puede utilizar la operación ListDecoderManifestSignalsAPI para comprobar si las señales del decodificador del manifiesto del decodificador se han actualizado. En el siguiente ejemplo se utiliza. AWS CLI

Para recuperar una lista paginada de resúmenes de todas las señales del decodificador (nodos) de un manifiesto del decodificador determinado, ejecute el siguiente comando.

*decoder-manifest-name*Sustitúyalo por el nombre del manifiesto del decodificador que estás comprobando.

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

Puedes usar la operación de la ListDecoderManifestNetworkInterfacesAPI para comprobar si las interfaces de red del manifiesto del decodificador se han actualizado. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todas las interfaces de red en un manifiesto del decodificador determinado, ejecute el siguiente comando.

*decoder-manifest-name*Sustitúyalo por el nombre del manifiesto del decodificador que estás comprobando.

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

Puedes usar la operación de la GetDecoderManifestAPI para verificar si las interfaces de red y las señales del decodificador del manifiesto del decodificador se han actualizado. En el siguiente ejemplo se utiliza. AWS CLI

Para recuperar información sobre un manifiesto del decodificador, ejecute el siguiente comando:

Reemplace *decoder-manifest* por el nombre del manifiesto del decodificador que desea recuperar.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en el manifiesto del decodificador podrían no reflejarse de inmediato.

Creación, aprovisionamiento y administración de vehículos

Los vehículos son instancias de modelos de vehículo. Los vehículos deben crearse a partir de un modelo de vehículo y asociarse a un manifiesto del decodificador. Los vehículos cargan uno o más flujos de datos a la nube. Por ejemplo, un vehículo puede enviar datos sobre el kilometraje, la temperatura del motor y el estado de la calefacción a la nube. Cada vehículo contiene la siguiente información:

`vehicleName`

ID que identifica el vehículo.

No agregue información de identificación personal (PII) ni ningún otro dato confidencial o sensible en el nombre del vehículo. Otros AWS servicios, incluido Amazon, pueden acceder a los nombres de los vehículos CloudWatch. Los nombres de los vehículos no se han diseñado para utilizarse con información privada o confidencial.

`modelManifestARN`

El Nombre de recurso de Amazon (ARN) de un modelo de vehículo (manifiesto del modelo). Cada vehículo se crea a partir de un modelo de vehículo. Los vehículos creados a partir del mismo modelo de vehículo constan del mismo grupo de señales heredadas del modelo de vehículo. Estas señales están definidas y estandarizadas en el catálogo de señales.

`decoderManifestArn`

El ARN del manifiesto del decodificador. Un manifiesto de decodificador proporciona información de decodificación que el AWS IoT FleetWise puede utilizar para transformar datos de señal sin procesar (datos binarios) en valores legibles por humanos. El manifiesto del decodificador debe estar asociado a un modelo de vehículo. AWS El IoT FleetWise utiliza el mismo manifiesto del decodificador para decodificar los datos sin procesar de los vehículos creados en función del mismo modelo de vehículo.

`attributes`

Los atributos son pares clave-valor que contienen información estática. Los vehículos pueden contener atributos heredados del modelo del vehículo. Puede agregar atributos adicionales para distinguir un vehículo individual de otros vehículos creados a partir del mismo modelo de vehículo. Por ejemplo, si tiene un coche negro, puede especificar el siguiente valor para un atributo: `{"color": "black"}`.

> **Important**
>
> Los atributos deben definirse en el modelo de vehículo asociado antes de poder agregarlos a vehículos individuales.

Para obtener más información sobre los modelos de vehículo, los manifiestos del decodificador y los atributos, consulte Modelización de vehículos.

AWS El IoT FleetWise proporciona las siguientes operaciones de API que puedes usar para crear y administrar vehículos.

- CreateVehicle— Crea un vehículo nuevo.
- BatchCreateVehicle— Crea uno o más vehículos nuevos.
- UpdateVehicle— Actualiza un vehículo existente.
- BatchUpdateVehicle— Actualiza uno o más vehículos existentes.
- DeleteVehicle— Elimina un vehículo existente.
- ListVehicles— Recupera una lista paginada de resúmenes de todos los vehículos.
- GetVehicle— Recupera información sobre un vehículo.

Tutoriales

- Aprovisionamiento de vehículos
- Temas reservados
- Creación de un vehículo
- Actualización de un vehículo (AWS CLI)
- Actualización de varios vehículos (AWS CLI)
- Eliminación de un vehículo
- Obtención de información sobre un vehículo (AWS CLI)

Aprovisionamiento de vehículos

El FleetWise software Edge Agent para AWS IoT que se ejecuta en su vehículo recopila y transfiere datos a la nube. AWS El IoT FleetWise se integra AWS IoT Core para respaldar la comunicación segura entre el software Edge Agent y la nube a través de MQTT. Cada vehículo corresponde a una

AWS IoT cosa. Puedes usar un elemento existente AWS IoT para crear un vehículo o configurar el AWS IoT FleetWise para que cree automáticamente AWS IoT algo para tu vehículo. Para obtener más información, consulte Creación de un vehículo (AWS CLI).

AWS IoT Core admite la autenticación y la autorización que ayudan a controlar de forma segura el acceso a FleetWise los recursos de AWS IoT. Los vehículos pueden usar certificados X.509 para autenticarse (iniciar sesión) para usar el AWS IoT FleetWise y AWS IoT Core políticas para obtener autorización (tener permisos) para realizar acciones específicas.

Autenticación de vehículos

Puede crear AWS IoT Core políticas para autenticar sus vehículos.

Para autenticar un vehículo

- Para crear una AWS IoT Core política, ejecute el siguiente comando.
  - Reemplace *policy-name* por el nombre de la política que desea crear.
  - Sustituya *file-name* por el nombre del archivo JSON que contiene la AWS IoT Core política.

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

Antes de utilizar la política de ejemplo, haga lo siguiente:

- Sustituya la *región* por la AWS región en la que creó FleetWise los recursos de AWS IoT.
- Sustituya *AWSAccount* por su ID de AWS cuenta.

En este ejemplo, se incluyen temas reservados para el AWS IoT FleetWise. Debe agregar los temas a la política. Para obtener más información, consulte Temas reservados.

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

```
}
```

Autorización de vehículos

Puede crear certificados X.509 para autorizar vehículos.

Para autorizar un vehículo

> **Important**
>
> Le recomendamos que cree un certificado nuevo para cada vehículo.

1. Para crear un par de claves RSA y emitir un certificado X.509, ejecute el siguiente comando:

   - Reemplace *cert* por el nombre del archivo que guarda el contenido del resultado del comando de certificatePem.
   - Sustituya *public-key* por el nombre del archivo que guarda el contenido del resultado del comando de KeyPair. PublicKey.
   - Sustituya *private-key* por el nombre del archivo que guarda el contenido de los resultados de los comandos de KeyPair. PrivateKey.

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

2. Copie el Nombre de recurso de Amazon (ARN) del certificado del resultado.
3. Para asociar la política al certificado, ejecute el siguiente comando:

   - Sustituya *el nombre de la política* por el nombre de la AWS IoT Core política que creó.
   - Reemplace *certificate-arn* por el ARN del certificado que ha copiado.

   ```
   aws iot attach-policy \
       --policy-name policy-name\
   ```

```
    --target "certificate-arn"
```

4. Para adjuntar el certificado al objeto, ejecute el siguiente comando:

   - *Sustituya el nombre de* la cosa por el nombre de la AWS IoT cosa o el identificador de su vehículo.
   - Reemplace *certificate-arn* por el ARN del certificado que ha copiado.

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

Temas reservados

AWS IoT FleetWise se reserva el uso de los siguientes temas. Si el tema reservado lo permite, puede suscribirse o publicar en el mismo. Sin embargo, no puede crear nuevos temas que comiencen por el símbolo de dólar ($). Si utiliza operaciones de publicación o suscripción no compatibles con temas reservados, es posible que la conexión finalice.

Tema	Operación de cliente permitida	Descripción	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/checkins`	Publicación	El software Edge Agent publica información sobre el estado del vehículo en este tema. La información sobre el estado del vehículo se intercambia en formato de búferes de protocolo (protobuf). Para obtener más información, consulte la Guía para desarroll	

Tema	Operación de cliente permitida	Descripción	
		adores de FleetWise software Edge Agent para AWS IoT.	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/signals`	Publicación	El software Edge Agent publica señales en este tema. La información sobre la señal se intercambia en formato de búferes de protocolo (protobuf). Para obtener más información, consulte la Guía para desarrolladores de FleetWise software Edge Agent para AWS IoT.	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/collection_schemes`	Suscribirse	AWS IoT FleetWise publica esquemas de recopilación de datos sobre este tema. Los vehículos utilizan estos esquemas de recopilación de datos.	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/decoder_manifests`	Suscribirse	AWS IoT FleetWise publica manifiestos de decodificadores sobre este tema. Los vehículos consumen estos manifiestos del decodificador.	

Creación de un vehículo

Puedes usar la FleetWise consola de AWS IoT o la API para crear un vehículo.

> **Important**
>
> Antes de comenzar, compruebe lo siguiente:
>
> - Debe disponer de un modelo de vehículo y su estado debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de modelos de vehículo.
> - El modelo de vehículo debe estar asociado a un manifiesto del decodificador y el estado de dicho manifiesto debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Temas

Creación de un vehículo (consola)

Puedes usar la FleetWise consola AWS IoT para crear un vehículo.

> **Important**
>
> Antes de comenzar, compruebe lo siguiente:
>
> - Debe disponer de un modelo de vehículo y su estado debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de modelos de vehículo.
> - El modelo de vehículo debe estar asociado a un manifiesto del decodificador y el estado de dicho manifiesto debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Para crear un vehículo

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Vehículos.
3. En la página de resumen del vehículo, elija Crear vehículo y, a continuación, siga los siguientes pasos.

Temas

Paso 1: Definir las propiedades del vehículo

En este paso, debe asignar un nombre al vehículo y asociarlo al manifiesto del modelo y al manifiesto del decodificador.

1. Escriba un nombre único para el vehículo.

   > **Important**
   >
   > Un vehículo corresponde a cualquier AWS IoT cosa. Si ya existe un objeto con ese nombre, elija Asociar el vehículo a un objeto de IoT para actualizarlo con el vehículo. O bien, elige un nombre de vehículo diferente y el AWS IoT FleetWise creará automáticamente algo nuevo para el vehículo.

2. Elija un modelo de vehículo (manifiesto del modelo) de la lista.
3. Elija un manifiesto del decodificador de la lista. El manifiesto del decodificador está asociado al modelo de vehículo.
4. (Opcional) Para asociar los atributos del vehículo, elija Agregar atributos. Si se salta este paso, tendrá que añadir atributos una vez creado el vehículo para poder implementarlo en las campañas.
5. (Opcional) Para asociar etiquetas al vehículo, elija Agregar etiqueta nueva. También puede agregar etiquetas una vez creado el vehículo.

6. Elija Siguiente.

Paso 2: Configurar el certificado del vehículo

Para usar tu vehículo como una AWS IoT cosa, debes configurar un certificado de vehículo con una política adjunta. Si se salta este paso, debe configurar un certificado tras crear el vehículo para poder distribuirlo en las campañas.

1. Elija Generar automáticamente un nuevo certificado (opción recomendada).
2. Elija Siguiente.

Paso 3: Asociar políticas al certificado

Asocie una política al certificado que ha configurado en el paso anterior.

1. En Políticas, introduzca el nombre de una política existente. Para crear una nueva política, elija Crear política.
2. Elija Siguiente.

Paso 4: Revisar y crear

Verifique las configuraciones del vehículo y, a continuación, elija Crear vehículo.

> **Important**
>
> Una vez creado el vehículo, debe descargar el certificado y las claves. Utilizará el certificado y la clave privada para conectar el vehículo en el FleetWise software Edge Agent para AWS IoT.

Creación de un vehículo (AWS CLI)

Al crear un vehículo, debe utilizar un modelo de vehículo que esté asociado a un manifiesto del decodificador. Puedes usar la operación de la CreateVehicleAPI para crear un vehículo. El siguiente ejemplo utiliza AWS CLI.

Important

Antes de comenzar, compruebe lo siguiente:

- Debe disponer de un modelo de vehículo y su estado debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de modelos de vehículo.
- El modelo de vehículo debe estar asociado a un manifiesto del decodificador y el estado de dicho manifiesto debe ser `ACTIVE`. Para obtener más información, consulte Creación y administración de manifiestos del decodificador.

Para crear un vehículo, ejecute el siguiente comando:

Reemplace *`file-name`* por el nombre del archivo JSON que contiene la configuración del vehículo.

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

Example configuración de un vehículo

- (Opcional) El valor `associationBehavior` puede ser uno de los siguientes:
  - `CreateIotThing`— Cuando se crea tu vehículo, el AWS IoT crea FleetWise automáticamente una AWS IoT cosa con el nombre del identificador de tu vehículo.
  - `ValidateIotThingExists`: se utiliza un objeto de AWS IoT existente para crear un vehículo.

    Para crear AWS IoT algo, ejecuta el siguiente comando. Reemplace *`thing-name`* por el nombre del objeto que desea crear.

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

  Si no se especifica, el AWS IoT crea FleetWise automáticamente AWS IoT algo para tu vehículo.

  Important

  Asegúrese de que la AWS IoT cosa esté aprovisionada una vez creado el vehículo. Para obtener más información, consulte Aprovisionamiento de vehículos.

- Reemplace *`vehicle-name`* por uno de los siguientes.

  - El nombre de AWS IoT lo que quieres, si `associationBehavior` está configurado para`ValidateIotThingExists`.
  - El ID del vehículo que debe crearse si `associationBehavior` está configurado como `CreateIotThing`.

    El ID del vehículo puede tener de 1 a 100 caracteres. Caracteres válidos: a-z, A-Z, 0-9, guion (-), guion bajo (_) y dos puntos (:).
- Reemplace *`model-manifest-ARN`* por el ARN de su modelo de vehículo (manifiesto del modelo).
- Reemplace *`decoder-manifest-ARN`* por el ARN del manifiesto del decodificador asociado al modelo de vehículo especificado.
- (Opcional) Puede agregar atributos adicionales para distinguir este vehículo de otros vehículos creados a partir del mismo modelo de vehículo. Por ejemplo, si tiene un automóvil eléctrico, puede especificar el siguiente valor para un atributo: `{"fuelType": "electric"}`.

> **Important**
>
> Los atributos deben definirse en el modelo de vehículo asociado antes de poder agregarlos a vehículos individuales.

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

Creación de varios vehículos (AWS CLI)

Puedes usar la operación de la BatchCreateVehicleAPI para crear varios vehículos a la vez. El siguiente ejemplo utiliza AWS CLI.

Para crear varios vehículos, ejecute el siguiente comando:

Reemplace *file-name* por el nombre del archivo JSON que contiene las configuraciones de varios vehículos.

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

Example configuraciones de vehículos

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

Puede crear hasta 10 vehículos por cada operación por lotes. Para obtener más información sobre la configuración de vehículos, consulte Creación de un vehículo (AWS CLI).

Actualización de un vehículo (AWS CLI)

Puedes usar la operación de la UpdateVehicleAPI para actualizar un vehículo existente. El siguiente ejemplo utiliza AWS CLI.

Para actualizar un vehículo, ejecute el siguiente comando:

Reemplace *file-name* por el nombre del archivo JSON que contiene la configuración del vehículo.

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

Example configuración de un vehículo

- Reemplace *vehicle-name* por el ID del vehículo que desea actualizar.
- (Opcional) Reemplace *model-manifest-ARN* por el ARN del modelo de vehículo (manifiesto del modelo) que utiliza para reemplazar el modelo de vehículo en uso.
- (Opcional) Reemplace *decoder-manifest-ARN* por el ARN del manifiesto del decodificador asociado al nuevo modelo de vehículo que ha especificado.
- (Opcional) *attribute-update-mode*Sustitúyalo por los atributos del vehículo.
  - `Merge`: los atributos nuevos se combinan con los atributos existentes actualizándolos con nuevos valores y agregando nuevos atributos si no existen.

    Por ejemplo, si un vehículo tiene los atributos `{"color": "black", "fuelType": "electric"}` y lo actualiza con los atributos `{"color": "", "fuelType": "gasoline", "model": "x"}`, el vehículo actualizado tendrá los siguientes atributos: `{"fuelType": "gasoline", "model": "x"}`.
  - `Overwrite`: los atributos existentes se sustituyen por atributos nuevos.

    Por ejemplo, si un vehículo tiene los atributos `{"color": "black", "fuelType": "electric"}` y lo actualiza con el atributo `{"model": "x"}`, el vehículo actualizado tendrá el atributo `{"model": "x"}`.

  Esto es obligatorio si la entrada contiene atributos.
- (Opcional) Para agregar nuevos atributos o actualizar los existentes con nuevos valores, configure `attributes`. Por ejemplo, si tiene un automóvil eléctrico, puede especificar el siguiente valor para un atributo: `{"fuelType": "electric"}`.

  Para eliminar atributos, configure `attributeUpdateMode` como `Merge`.

> **Important**
>
> Los atributos deben definirse en el modelo de vehículo asociado antes de poder agregarlos a vehículos individuales.

```
{
```

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

Actualización de varios vehículos (AWS CLI)

Puedes usar la operación de la BatchUpdateVehicleAPI para actualizar varios vehículos existentes a la vez. El siguiente ejemplo utiliza AWS CLI.

Para actualizar varios vehículos, ejecute el siguiente comando:

Reemplace *file-name* por el nombre del archivo JSON que contiene las configuraciones de varios vehículos.

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

Example configuraciones de vehículos

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

```
}
```

Puede actualizar hasta 10 vehículos en cada operación por lotes. Para obtener más información acerca la configuración de cada vehículo, consulte Actualización de un vehículo (AWS CLI).

Eliminación de un vehículo

Puedes usar la FleetWise consola o la API de AWS IoT para eliminar vehículos.

> **Important**
>
> Una vez que se elimina un vehículo, el AWS IoT lo elimina FleetWise automáticamente de las flotas y campañas asociadas. Para obtener más información, consulte Creación y administración de flotas y Recopilación y transferencia de datos con campañas. Sin embargo, el vehículo sigue existiendo como una cosa o sigue asociado a una cosa dentro de ella. AWS IoT Core Para obtener instrucciones sobre cómo eliminar un objeto, consulte Eliminar un objeto en la Guía para desarrolladores de AWS IoT Core .

Eliminación de un vehículo (consola)

Puedes usar la FleetWise consola AWS IoT para eliminar un vehículo.

Para eliminar un vehículo

1. Navegue hasta la FleetWiseconsola de AWS IoT.
2. En el panel de navegación, elija Vehículos.
3. En la página Vehículos, seleccione el botón situado junto al vehículo que desea eliminar.
4. Elija Eliminar.
5. En Eliminar **`vehicle-name`**, introduzca el nombre del vehículo y, a continuación, elija Eliminar.

Eliminación de un vehículo (AWS CLI)

Puedes usar la operación de la DeleteVehicleAPI para eliminar un vehículo. En el siguiente ejemplo se utiliza AWS CLI.

Para eliminar un vehículo, ejecute el siguiente comando:

Reemplace *vehicle-name* por el ID del vehículo que desea eliminar.

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

Obtención de información sobre un vehículo (AWS CLI)

Puedes usar la operación de la ListVehiclesAPI para verificar si se ha eliminado un vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todos los vehículos, ejecute el siguiente comando:

```
aws iotfleetwise list-vehicles
```

Puedes usar la operación de la GetVehicleAPI para recuperar la información del vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar los metadatos de un vehículo, ejecute el siguiente comando:

Reemplace *vehicle-name* por el ID del vehículo que desea recuperar.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en el vehículo podrían no reflejarse de inmediato.

Creación y administración de flotas

Una flota representa un grupo de vehículos. Una flota sin vehículos asociados es una entidad vacía. Antes de poder utilizar una flota para administrar varios vehículos al mismo tiempo, debe asociar los vehículos a dicha flota. Un vehículo puede pertenecer a varias flotas. Puede controlar qué datos se recopilarán de una flota de vehículos y cuándo deberán recopilarse mediante la implementación de una campaña. Para obtener más información, consulte Recopilación y transferencia de datos con campañas.

Una flota contiene la siguiente información:

`fleetId`

El ID de la flota.

`description` (opcional)

Una descripción para ayudarlo a encontrar la flota.

`signalCatalogArn`

El nombre de recurso de Amazon (ARN) del catálogo de señales.

AWS IoT FleetWise proporciona las siguientes operaciones de API que puede utilizar para crear y administrar flotas:

- CreateFleet: crea un grupo de vehículos que contienen el mismo grupo de señales.
- AssociateVehicleFleet: asocia un vehículo a una flota.
- DisassociateVehicleFleet: elimina la asociación de un vehículo a una flota.
- UpdateFleet: actualiza la descripción de una flota existente.
- DeleteFleet: elimina una flota existente.
- ListFleets: recupera una lista paginada de resúmenes de todas las flotas.
- ListFleetsForVehicle: recupera una lista paginada de los ID de todas las flotas a las que pertenece el vehículo.
- ListVehiclesInFleet: recupera una lista paginada de resúmenes de todos los vehículos de una flota.
- GetFleet: recupera información sobre una flota.

Temas

Creación de una flota (AWS CLI)

Puede utilizar la operación de la API CreateFleet para crear una flota de vehículos. El siguiente ejemplo utiliza AWS CLI.

> **Important**
>
> Para poder crear una flota, debe disponer de un catálogo de señales. Para obtener más información, consulte Creación de un catálogo de señales (AWS CLI).

Para crear una flota, ejecute el siguiente comando:

- Reemplace *fleet-id* por el ID de la flota que está creando.

  El ID de la flota debe ser único y debe tener entre 1 y 100 caracteres. Caracteres válidos: letras (A-Z y a-z), números (0-9), dos puntos (:), guiones (-) y guiones bajos (_).
- (Opcional) Reemplace *description* por una descripción.

  La descripción puede contener entre 1 y 2048 caracteres.
- Reemplace *signal-catalog-arn* por el ARN del catálogo de señales.

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

Asociación de un vehículo a una flota (AWS CLI)

Puede utilizar la operación de la API AssociateVehicleFleet para asociar un vehículo a una flota. El siguiente ejemplo utiliza AWS CLI.

> **Important**
>
> - Para poder asociar un vehículo a una flota, debe disponer de un vehículo y de una flota. Para obtener más información, consulte Creación, aprovisionamiento y administración de vehículos.
> - Si asocia un vehículo a una flota a la que se dirige una campaña, AWS IoT FleetWise implementa automáticamente dicha campaña en el vehículo.

Para asociar un vehículo a una flota, ejecute el siguiente comando:

- Reemplace *fleet-id* por el ID de la flota.
- Reemplace *vehicle-name* por el ID del vehículo.

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

Anulación de la asociación de un vehículo a una flota (AWS CLI)

Puede utilizar la operación de la API DisassociateVehicleFleet para anular la asociación de un vehículo a una flota. El siguiente ejemplo utiliza AWS CLI.

Para anular la asociación de un vehículo a una flota, ejecute el siguiente comando:

- Reemplace *fleet-id* por el ID de la flota.
- Reemplace *vehicle-name* por el ID del vehículo.

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

Actualización de una flota (AWS CLI)

Puede utilizar la operación de la API UpdateFleet para actualizar la descripción de una flota. El siguiente ejemplo utiliza AWS CLI.

Para actualizar una flota, ejecute el siguiente comando:

- Reemplace *fleet-id* por el ID de la flota que esté actualizando.
- Reemplace *description* por una descripción nueva.

  La descripción puede contener entre 1 y 2048 caracteres.

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

Eliminación de una flota (AWS CLI)

Puede utilizar la operación de la API DeleteFleet para eliminar una flota. El siguiente ejemplo utiliza AWS CLI.

> **Important**
>
> Antes de eliminar una flota, asegúrese de que no tenga vehículos asociados. Para obtener instrucciones acerca de cómo anular la asociación de un vehículo a una flota, consulte Anulación de la asociación de un vehículo a una flota (AWS CLI).

Para eliminar una flota, ejecute el siguiente comando:

Reemplace *fleet-id* por el ID de la flota que vaya a eliminar.

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

Obtención de información sobre una flota (AWS CLI)

Puede utilizar la operación de la API ListFleets para verificar si se ha eliminado una flota. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todas las flotas, ejecute el siguiente comando:

```
aws iotfleetwise list-fleets
```

Puede utilizar la operación de la API ListFleetsForVehicle para recuperar una lista paginada de los ID de todas las flotas a las que pertenece el vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de los ID de todas las flotas a las que pertenece el vehículo, ejecute el siguiente comando:

Reemplace *vehicle-name* por el ID del vehículo.

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

Puede utilizar la operación de la API ListVehiclesInFleet para recuperar una lista paginada de resúmenes de todos los vehículos de una flota. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todos los vehículos de una flota, ejecute el siguiente comando:

Reemplace *fleet-id* por el ID de la flota.

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

Puede utilizar la operación de la API GetFleet para recuperar la información de la flota. El siguiente ejemplo utiliza AWS CLI.

Para recuperar los metadatos de una flota, ejecute el siguiente comando:

Reemplace *fleet-id* por el ID de la flota.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en la flota podrían no reflejarse inmediatamente.

Recopilación y transferencia de datos con campañas

Una campaña es una orquestación de reglas de recopilación de datos. Las campañas proporcionan al software Edge Agent para AWS IoT FleetWise instrucciones sobre cómo seleccionar, recopilar y transferir datos a la nube.

Usted crea campañas en la nube. Una vez que usted o su equipo han aprobado una campaña, AWS IoT FleetWise las implementa automáticamente en los vehículos. Puede optar por implementar una campaña en un vehículo o en una flota de vehículos. El software Edge Agent no comienza a recopilar datos hasta que se implementa una campaña activa en el vehículo.

> **Note**
>
> Para que las campañas funcionen, debe disponer de lo siguiente:
>
> - El software Edge Agent ejecutándose en el vehículo. Para obtener más información sobre cómo desarrollar e instalar el software Edge Agent, así como sobre cómo trabajar con él, haga lo siguiente:
>
> 1. Vaya a la consola de AWS IoT FleetWise.
> 2. En la página de inicio del servicio, en la sección Introducción a AWS IoT FleetWise, elija Conozca el agente de borde.
>
> - AWS IoT Core configurado para aprovisionar el vehículo. Para obtener más información, consulte Aprovisionamiento de vehículos.

Cada elemento de la lista contiene la siguiente información:

`signalCatalogArn`

El Nombre de recurso de Amazon (ARN) del catálogo de señales asociado a la campaña.

(Opcional) `tags`

Las etiquetas son metadatos que se pueden utilizar para administrar la campaña. Puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados.

`TargetArn`

El ARN de un vehículo o una flota donde se implementa la campaña.

`name`

Un nombre único que ayuda a identificar la campaña.

`collectionScheme`

Los esquemas de recopilación de datos proporcionan al software Edge Agent instrucciones sobre qué datos deben recopilarse y cuándo debe hacerse. Actualmente, AWS IoT FleetWise admite el esquema de recopilación basado en la condición y el esquema de recopilación basado en el tiempo.

`conditionBasedCollectionScheme`

El esquema de recopilación basado en la condición utiliza una expresión lógica para reconocer qué datos deben recopilarse. El software Edge Agent recopila datos cuando se cumple la condición.

`expression`

La expresión lógica que se utiliza para reconocer qué datos deben recopilarse. Por ejemplo, si se especifica la expresión `$variable.`myVehicle.InVehicleTemperature` > 50.0`, el software Edge Agent recopila valores de temperatura superiores a 50,0. Para obtener instrucciones acerca de cómo escribir expresiones, consulte Expresiones lógicas para campañas.

(Opcional) `triggerMode` puede ser uno de los siguientes valores:

- `RISING_EDGE`: el software Edge Agent recopila datos solo cuando se cumple la condición por primera vez. Por ejemplo, `$variable.`myVehicle.AirBagDeployed` == true`.
- `ALWAYS`: el software Edge Agent recopila datos siempre que se cumpla la condición.

(Opcional) `minimumTriggerIntervalMs`

Cantidad de tiempo mínima (en milisegundos) entre dos eventos de recopilación de datos. Si una señal cambia con frecuencia, es conveniente recopilar datos a un ritmo más lento.

(Opcional) `conditionLanguageVersion`

La versión del lenguaje de expresiones condicionales.

`timeBasedCollectionScheme`

Si define un esquema de recopilación basado en el tiempo, debe especificar un periodo en milisegundos. El software Edge Agent utiliza el periodo para decidir con qué frecuencia deben

recopilarse datos. Por ejemplo, si el periodo es de 120 000 milisegundos, el software Edge Agent recopila datos una vez cada dos minutos.

(Opcional) `compression`

Para ahorrar ancho de banda inalámbrico y reducir el tráfico de la red, puede especificar SNAPPY para comprimir los datos en los vehículos.

De forma predeterminada (`OFF`), el software Edge Agent no comprime los datos.

`dataDestinationConfigs`

Elija el destino al que la campaña transferirá los datos del vehículo. Puede elegir guardar datos en Amazon S3 o en Amazon Timestream.

S3 es un mecanismo de almacenamiento de datos rentable que ofrece capacidades duraderas de administración de datos y servicios de datos descendentes. Puede utilizar S3 para obtener datos relacionados con los comportamientos de conducción o para analizar el mantenimiento a largo plazo.

Timestream es un mecanismo de persistencia de datos que puede ayudarlo a identificar tendencias y patrones casi en tiempo real. Puede utilizar Timestream para obtener datos de serie temporal; por ejemplo, para analizar las tendencias históricas en la velocidad o el frenado del vehículo.

(Opcional) `dataExtraDimensions`

Puede agregar uno o más atributos para proporcionar información adicional sobre una señal.

(Opcional) `description`

Puede agregar una descripción para ayudar a identificar el propósito de la campaña.

(Opcional) `diagnosticsMode`

Cuando el modo de diagnóstico está configurado como `SEND_ACTIVE_DTCS`, la campaña envía códigos de diagnóstico de problemas (DTC) estándar y almacenados que ayudan a identificar qué es lo que falla en el vehículo. Por ejemplo, P0097 indica que el módulo de control del motor (ECM) ha determinado que la entrada del sensor de temperatura del aire de admisión 2 (IAT2) es inferior al rango normal del sensor.

De forma predeterminada (`OFF`), el software Edge Agent no envía códigos de diagnóstico.

(Opcional) `expiryTime`

Puede definir la fecha de caducidad de una campaña. Cuando una campaña caduca, el software Edge Agent deja de recopilar los datos tal y como se especifica en dicha campaña. Si se implementan varias campañas en el vehículo, el software Edge Agent utiliza otras campañas para recopilar datos.

Valor predeterminado: `253402243200` (31 de diciembre de 9999 a las 00:00:00 UTC)

(Opcional) `postTriggerCollectionDuration`

Puede definir una duración para la recopilación posterior a la activación, de modo que el software Edge Agent siga recopilando datos durante un periodo específico tras invocar un esquema. Por ejemplo, si se invoca un esquema de recopilación basado en la condición con la expresión `$variable.`myVehicle.Engine.RPM` > 7000.0`, el software Edge Agent sigue recopilando los valores de revoluciones por minuto (RPM) del motor. Incluso si las RPM solo suben por encima de 7000 una vez, esto podría indicar la presencia de un problema mecánico. En este caso, es posible que desee que el software Edge Agent continúe recopilando datos para ayudar a supervisar el estado del vehículo.

Valor predeterminado: `0`

(Opcional) `priority`

Puede especificar un número entero para indicar el nivel de prioridad de la campaña. Las campañas con un número menor tienen mayor prioridad. Si implementa varias campañas en un vehículo, las campañas con mayor prioridad se inician primero.

Valor predeterminado: `0`

(Opcional) `signalsToCollect`

Una lista de señales a partir de las cuales se recopilan datos cuando se invoca el esquema de recopilación de datos.

> **Important**
>
> Las señales utilizadas en la expresión del esquema de recopilación basado en la condición deben especificarse en este campo.

`name`

El nombre de la señal desde la que se recopilan los datos cuando se invoca el esquema de recopilación de datos.

(Opcional) `maxSampleCount`

El número máximo de muestras de datos que el software Edge Agent recopila y transfiere a la nube cuando se invoca el esquema de recopilación de datos.

(Opcional) `minimumSamplingIntervalMs`

Cantidad de tiempo mínima (en milisegundos) que debe transcurrir entre dos eventos de recopilación de muestras de datos. Si una señal cambia con frecuencia, puede utilizar este parámetro para recopilar datos a un ritmo más lento.

Rango válido: 0-4294967295

(Opcional) `spoolingMode`

Si `spoolingMode` está configurado para `TO_DISK`, el software Edge Agent almacena temporalmente los datos de forma local cuando el vehículo no está conectado a la nube. Una vez restablecida la conexión, los datos almacenados localmente se transfieren de forma automática a la nube.

Valor predeterminado: `OFF`

(Opcional) `startTime`

Una campaña aprobada se activa a la hora de inicio.

Valor predeterminado: `0`

El estado de una campaña puede ser uno de los siguientes valores:

- `CREATING`: AWS IoT FleetWise está procesando la solicitud para crear la campaña.
- `WAITING_FOR_APPROVAL`: una vez creada una campaña, esta entra en el estado `WAITING_FOR_APPROVAL`. Para aprobar la campaña, utilice la operación de la API `UpdateCampaign`. Una vez aprobada la campaña, AWS IoT FleetWise la implementa automáticamente en el vehículo o la flota de destino. Para obtener más información, consulte Actualización de una campaña (AWS CLI).
- `RUNNING` : la campaña está activa.

- `SUSPENDED`: la campaña está suspendida. Para reanudar la campaña, utilice la operación de la API `UpdateCampaign`.

AWS IoT FleetWise proporciona las siguientes operaciones de API que puede utilizar para crear y administrar campañas.

- CreateCampaign: crea una nueva campaña.
- UpdateCampaign: actualiza una campaña existente. Una vez creada una campaña, debe usar esta operación de API para aprobarla.
- DeleteCampaign: elimina una campaña existente.
- ListCampaigns: recupera una lista paginada de resúmenes de todas las campañas.
- GetCampaign: recupera información sobre una campaña.

Tutoriales

- Creación de una campaña
- Actualización de una campaña (AWS CLI)
- Eliminación de una campaña
- Obtención de información sobre una campaña (AWS CLI)

Creación de una campaña

Puede utilizar la consola o la API de AWS IoT FleetWise para crear campañas que recopilen datos de vehículos.

> **Important**
>
> Para que la campaña funcione, debe disponer de lo siguiente:
>
> - El software Edge Agent ejecutándose en el vehículo. Para obtener más información sobre cómo desarrollar e instalar el software Edge Agent, así sobre cómo trabajar con él, haga lo siguiente:
>
> 1. Vaya a la consola de AWS IoT FleetWise.
> 2. En la página de inicio del servicio, en la sección Introducción a AWS IoT FleetWise, elija Conozca el agente de borde.

- AWS IoT Core configurado para aprovisionar el vehículo. Para obtener más información, consulte Aprovisionamiento de vehículos.

Temas

- Creación de una campaña (consola)
- Creación de una campaña (AWS CLI)
- Expresiones lógicas para campañas

Creación de una campaña (consola)

Puede usar la consola de AWS IoT FleetWise para crear una campaña a fin de seleccionar, recopilar y transferir los datos del vehículo a la nube.

Para crear una campaña:

1. Vaya a la consola de AWS IoT FleetWise.
2. En el panel de navegación, elija Campañas.
3. En la página Campañas, elija Crear campaña y, a continuación, complete los pasos de los siguientes temas.

Temas

- Paso 1: Configurar la campaña
- Paso 2: Definir el destino de almacenamiento
- Paso 3: Agregar vehículos
- Paso 4: Revisar y crear
- Paso 5: Implementar una campaña

Important

- Para poder crear una campaña, debe disponer de un catálogo de señales y de un vehículo. Para más información, consulte Creación y administración de catálogos de señales y Creación, aprovisionamiento y administración de vehículos.

- Una vez creada una campaña, debe aprobarla. Para obtener más información, consulte Paso 5: Implementar una campaña.

Paso 1: Configurar la campaña

En la sección Información general, haga lo siguiente:

1. Introduzca un nombre para la campaña.
2. (Opcional) Introduzca una descripción.

Configure el esquema de recopilación de datos de la campaña. Un esquema de recopilación de datos proporciona al software Edge Agent instrucciones sobre qué datos deben recopilarse y cuándo debe hacerse. En la consola de AWS IoT FleetWise, puede configurar un esquema de recopilación de datos de las siguientes formas:

- Defina de forma manual el esquema de recopilación de datos.
- Cargue un archivo para definir automáticamente el esquema de recopilación de datos.

En Opción de configuración, elija cualquiera de las siguientes opciones:

- Para especificar manualmente el tipo de esquema de recopilación de datos y definir las opciones para personalizarlo, elija Definir el esquema de recopilación de datos.

  Especifique manualmente el tipo de esquema de recopilación de datos y defina las opciones para personalizar el esquema.

  1. En la sección Detalles del esquema de recopilación de datos, elija el tipo de esquema de recopilación de datos que desea que use la campaña. Si desea usar una expresión lógica para reconocer qué datos del vehículo deben recopilarse, elija Basado en condición. Para utilizar un periodo específico a fin de decidir con qué frecuencia deben recopilarse datos del vehículo, elija En función del tiempo.
  2. Defina el tiempo durante el que la campaña recopila datos.

> **Note**
>
> De forma predeterminada, una campaña aprobada se activa de inmediato y no tiene una hora de finalización establecida. Para evitar cargos adicionales, debe especificar un intervalo de tiempo.

3. Si ha especificado un esquema de recopilación de datos basado en la condición, debe definir una expresión lógica para reconocer qué datos deben recopilarse. AWS IoT FleetWise utiliza una expresión lógica para reconocer qué datos deben recopilarse para un esquema basado en la condición. La expresión debe especificar el nombre completo de una señal como variable, un operador de comparación y un valor de comparación.

   Por ejemplo, si especifica la expresión `$variable.`myVehicle.InVehicleTemperature` > 50.0`, AWS IoT FleetWise recopila valores de temperatura superiores a 50,0. Para obtener instrucciones acerca de cómo escribir expresiones, consulte Expresiones lógicas para campañas.

   Introduzca la expresión lógica que se utiliza para reconocer qué datos deben recopilarse.

4. (Opcional) Puede especificar la versión del lenguaje de la expresión condicional. El valor predeterminado es 1.
5. (Opcional) Puede especificar el intervalo de desencadenador mínimo, que es la duración mínima entre dos eventos de recopilación de datos. Por ejemplo, si una señal cambia con frecuencia, es conveniente recopilar datos a un ritmo más lento.
6. Especifique la condición del Modo del desencadenador para que el software Edge Agent recopile datos. De forma predeterminada, el software Edge Agent para AWS IoT FleetWise siempre recopila datos cuando se cumple la condición. O bien solo puede recopilar datos cuando se cumple la condición por primera vez, En el primer desencadenador.
7. Si ha especificado un esquema de recopilación de datos basado en el tiempo, debe especificar un Periodo en milisegundos, de 10 000 a 60 000 milisegundos. El software Edge Agent utiliza el periodo para decidir con qué frecuencia deben recopilarse datos.
8. (Opcional) Puede editar las Opciones de esquema avanzadas.

   a. Para ahorrar ancho de banda inalámbrico y reducir el tráfico de red mediante la compresión de datos, elija Snappy.

b. (Opcional) Para definir cuánto tiempo, en milisegundos, se deben seguir recopilando datos tras un evento de recopilación de datos, puede especificar la Duración de la recopilación posterior al desencadenador.

c. (Opcional) Para indicar el nivel de prioridad de la campaña, puede especificar la Prioridad de la campaña. Las campañas con un número de prioridad menor se implementan primero y se considera que tienen una prioridad más alta.

d. El software Edge Agent puede almacenar temporalmente datos de forma local cuando un vehículo no está conectado a la nube. Una vez restablecida la conexión, los datos almacenados localmente se transfieren de forma automática a la nube. Especifique si desea que Edge Agent almacene los datos de forma local durante una pérdida de conexión.

e. (Opcional) Para proporcionar información adicional para una señal, agregue hasta cinco atributos como Dimensiones de datos adicionales.

- Para cargar un archivo a fin de definir el esquema de recopilación de datos, seleccione Cargar un archivo .json desde un dispositivo local. AWS IoT FleetWise define automáticamente las opciones que puede definir en el archivo. Puede revisar y actualizar las opciones seleccionadas.

  Cargue un archivo .json con detalles sobre el esquema de recopilación de datos.

  1. Para importar información sobre el esquema de recopilación de datos, elija Elegir archivos. Para obtener más información acerca del formato de archivo requerido, consulte la documentación de la API CreateCampaign.

     > **Note**
     >
     > Actualmente, AWS IoT FleetWise admite la extensión de formato de archivo .json.

  2. AWS IoT FleetWise define automáticamente el esquema de recopilación de datos en función de la información del archivo. Revise las opciones seleccionadas por AWS IoT FleetWise. Puede actualizar las opciones si es necesario.

Especificación de señales

Puede especificar las señales desde las que se recopilarán los datos al invocarse el esquema de recopilación de datos.

> **Important**
>
> Las señales utilizadas en la expresión del esquema de recopilación basado en la condición deben especificarse en este campo.

Para especificar las señales de las que se van a recopilar datos:

1. Busque el nombre completo de la señal.

   > **Note**
   >
   > El nombre completo de la señal es la ruta a la señal más el nombre de la señal. Utilice un punto (.) para hacer referencia a una señal secundaria.
   > Por ejemplo, `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` es el nombre completo del actuador `HandsOffSteeringState`. `Vehicle.Chassis.SteeringWheel.HandsOff.` es la ruta a dicho actuador.

2. (Opcional) Para obtener el Recuento máximo de muestras, introduzca el recuento máximo de muestras de datos que el software Edge Agent recopila y transfiere a la nube al invocarse el esquema de recopilación de datos.
3. (Opcional) En Intervalo de muestreo mínimo, introduzca el tiempo mínimo entre dos eventos de recopilación de muestras de datos, en milisegundos. Si una señal cambia con frecuencia, puede utilizar este parámetro para recopilar datos a un ritmo más lento.
4. Para agregar otra señal, elija Agregar más señales. Puede agregar hasta 999 señales.
5. Elija Siguiente.

Paso 2: Definir el destino de almacenamiento

> **Note**
>
> Solo puede transferir los datos del vehículo a Amazon S3 si la campaña contiene señales de datos de sistemas de visión.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Elija el destino en el que desea almacenar los datos recopilados por la campaña. Puede transferir los datos del vehículo a Amazon S3 o a Amazon Timestream.

En Configuración de destino, haga lo siguiente:

- Elija S3 o Timestream en la lista desplegable.

Para guardar los datos del vehículo en un bucket de S3, elija Amazon S3. S3 es un servicio de almacenamiento de objetos que almacena datos como objetos dentro de buckets. Para obtener más información, consulte Creación, configuración y trabajo con buckets de Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

S3 optimiza el costo del almacenamiento de datos y proporciona mecanismos adicionales para utilizar los datos de los vehículos, como los lagos de datos, el almacenamiento centralizado de datos, los procesos de procesamiento de datos y los análisis. Puede usar S3 para almacenar datos para su procesamiento y análisis por lotes. Por ejemplo, puede crear informes de eventos de frenada brusca para el modelo de machine learning (ML). Los datos entrantes del vehículo se almacenan en un búfer durante 10 minutos antes de la entrega.

Amazon S3

> **Important**
>
> No puede transferir solo datos a S3 si AWS IoT FleetWise tiene permisos para escribir en el bucket de S3. Para obtener más información sobre la concesión de acceso, consulte Control del acceso con AWS IoT FleetWise.

En Configuración de destino de S3, haga lo siguiente:

1. En Bucket de S3, elija un bucket para el que AWS IoT FleetWise tenga permisos.
2. (Opcional) Introduzca un prefijo personalizado que puede usar para organizar los datos almacenados en el bucket de S3.
3. Elija el formato de salida, que es el formato de los archivos que se guardan en el bucket de S3.
4. Elija si desea comprimir los datos almacenados en el bucket de S3 como un archivo .gzip. Recomendamos comprimir los datos para minimizar los costes de almacenamiento.
5. Las opciones seleccionadas en la configuración de destino de S3 cambian el URI del objeto S3 del ejemplo. Este es un ejemplo del tipo de archivos que se guardan en S3.

Para almacenar los datos del vehículo en una tabla de Timestream, elija Amazon Timestream. Puede utilizar Timestream para consultar los datos del vehículo y así identificar tendencias y patrones. Por ejemplo, puede utilizar Timestream para crear una alarma para el nivel de combustible del vehículo. Los datos entrantes del vehículo se transfieren a Timestream casi en tiempo real. Para obtener más información, consulte ¿Qué es Amazon Timestream? en la Guía para desarrolladores de Amazon Timestream.

Amazon Timestream

> **Important**
>
> Solo puede transferir datos a una tabla si AWS IoT FleetWise tiene permisos para escribir datos en Timestream. Para obtener más información sobre la concesión de acceso, consulte Control del acceso con AWS IoT FleetWise.

En Configuración de la tabla de Timestream, haga lo siguiente:

1. En Nombre de base de datos de Timestream, elija el nombre de la base de datos de Timestream en la lista desplegable.
2. En Nombre de tabla de Timestream, elija el nombre de la tabla de Timestream en la lista desplegable.

En Acceso al servicio para Timestream, haga lo siguiente:

- Elija un rol de IAM en la lista desplegable.

- Elija Siguiente.

Paso 3: Agregar vehículos

Para elegir en qué vehículos desea implementar la campaña, selecciónelos en la lista de vehículos. Filtre los vehículos buscando los atributos y valores que les agregó al crearlos, o bien buscando por nombre de vehículo.

En Filtrar vehículos, haga lo siguiente:

1. En el cuadro de búsqueda, busque el atributo o el nombre del vehículo y elíjalo de la lista.

> **Note**
>
> Cada atributo puede utilizarse solo una vez.

2. Introduzca el valor del atributo o el nombre del vehículo en el que desee implementar la campaña. Por ejemplo, si el nombre completo del atributo es `fuelType`, introduzca `gasoline` como valor.
3. Para buscar otro atributo del vehículo, repita los pasos anteriores. Puede buscar hasta cinco atributos de vehículo y un número ilimitado de nombres de vehículo.
4. Los vehículos que coinciden con la búsqueda aparecen en Nombre del vehículo. Elija los vehículos en los que desee implementar la campaña.

> **Note**
>
> En los resultados de la búsqueda se muestran hasta 100 vehículos. Elija Seleccionar todo para agregar todos los vehículos a la campaña.

5. Elija Siguiente.

Paso 4: Revisar y crear

Verifique las configuraciones de la campaña y, a continuación, elija Crear campaña.

> **Note**
>
> Tras crear una campaña, usted o su equipo deben implementarla en los vehículos.

Paso 5: Implementar una campaña

Tras crear una campaña, usted o su equipo deben implementarla en los vehículos.

Para implementar una campaña:

1. En la página Resumen de la campaña, elija Implementar.
2. Revise y confirme que desea iniciar la implementación y empezar a recopilar datos de los vehículos conectados a la campaña.

3. Elija Implementar.

Si desea detener la recopilación de datos de los vehículos conectados a la campaña, en la página Resumen de la campaña elija Suspender. Para reanudar la recopilación de datos de los vehículos conectados a la campaña, elija Reanudar.

Creación de una campaña (AWS CLI)

Puede utilizar la operación de la API CreateCampaign para crear una campaña. El siguiente ejemplo utiliza AWS CLI.

Al crear una campaña, los datos recopilados de los vehículos se pueden almacenar en Amazon S3 (S3) o en Amazon Timestream. Elija Timestream para obtener una base de datos de serie temporal rápida, escalable y sin servidor; por ejemplo, para almacenar datos que requieren un procesamiento casi en tiempo real. Elija S3 para un almacenamiento de objetos con una escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector.

> **Important**
>
> Solo puede transferir los datos del vehículo si AWS IoT FleetWise tiene permisos para escribir datos en S3 o Timestream. Para obtener más información sobre la concesión de acceso, consulte Control del acceso con AWS IoT FleetWise.

Creación de una campaña

> **Important**
>
> - Para poder crear una campaña, debe disponer de un catálogo de señales y de un vehículo o una flota. Para más información, consulte Creación y administración de catálogos de señales, Creación, aprovisionamiento y administración de vehículos y Creación y administración de flotas.
> - Una vez creada una campaña, debe utilizar la operación de la API `UpdateCampaign` para aprobarla. Para obtener más información, consulte Actualización de una campaña (AWS CLI)

Para crear una campaña, ejecute el siguiente comando:

Reemplace *file-name* por el nombre del archivo JSON que contiene la configuración de la campaña.

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

- Reemplace *campaign-name* por el nombre de la campaña que está creando.
- Reemplace *signal-catalog-arn* por el Nombre de recurso de Amazon (ARN) del catálogo de señales.
- Reemplace *target-arn* por el ARN de una flota o vehículo que haya creado.
- Reemplace *bucket-arn* por el ARN del bucket de S3.

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

- Reemplace *campaign-name* por el nombre de la campaña que está creando.
- Reemplace *signal-catalog-arn* por el Nombre de recurso de Amazon (ARN) del catálogo de señales.
- Reemplace *target-arn* por el ARN de una flota o vehículo que haya creado.
- Reemplace *role-arn* por el ARN del rol de ejecución de tareas que otorga permiso a AWS IoT FleetWise para entregar datos a la tabla de Timestream.
- Reemplace *table-arn* por el ARN de la tabla de Timestream.

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

Expresiones lógicas para campañas

AWS IoT FleetWise utiliza una expresión lógica simple para reconocer qué datos recopilar como parte de una campaña. Para obtener más información acerca de expresiones, consulte Expresiones en la Guía para desarrolladores de AWS IoT Events.

La variable de expresión se debe construir de manera que cumpla con las reglas del tipo de datos que se recopila. Para los datos del sistema de telemetría, la variable de expresión debe ser el nombre totalmente cualificado de la señal. Para datos de sistemas de visión, la expresión combina el nombre totalmente cualificado de la señal con la ruta que va desde el tipo de datos de la señal hasta una de sus propiedades.

Por ejemplo, si el catálogo de la señal contiene los siguientes nodos:

```
{
    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"
}
```

Si los nodos siguen la definición de ROS 2:

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

A continuación, se muestran todas las variables de expresión de eventos posibles:

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

Actualización de una campaña (AWS CLI)

Puede utilizar la operación de la API UpdateCampaign para actualizar una campaña existente. El siguiente comando utiliza la AWS CLI.

- Reemplace *campaign-name* por el nombre de la campaña que está actualizando.
- Sustituya *action* por una de las acciones siguientes:
  - APPROVE: aprueba la campaña para permitir que AWS IoT FleetWise la implemente en un vehículo o flota.
  - SUSPEND: suspende la campaña. La campaña se elimina de los vehículos y todos los vehículos de la campaña suspendida dejarán de enviar datos.
  - RESUME: reactiva la SUSPEND campaña. La campaña se volverá a implementar en todos los vehículos y estos volverán a enviar datos.
  - UPDATE: actualiza la campaña definiendo los atributos y asociándolos a una señal.

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

Eliminación de una campaña

Para eliminar campañas, puede usar la consola o la API de AWS IoT FleetWise.

Eliminación de una campaña (consola)

Para eliminar una campaña, use la consola de AWS IoT FleetWise.

Para eliminar una campaña:

1. Vaya a la consola de AWS IoT FleetWise.
2. En el panel de navegación, elija Campañas.
3. En la página Campañas, elija la campaña de destino.
4. Elija Eliminar.
5. En ¿Eliminar **campaign-name**?, introduzca el nombre de la campaña que desea eliminar y, a continuación, elija Confirmar.

Eliminación de una campaña (AWS CLI)

Puede utilizar la operación de la API DeleteCampaign para eliminar una campaña. El siguiente ejemplo utiliza AWS CLI.

Para eliminar una campaña, ejecute el siguiente comando:

Reemplace *campaign-name* por el nombre del vehículo que está eliminando.

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

Obtención de información sobre una campaña (AWS CLI)

Puede utilizar la operación de la API ListCampaigns para comprobar si se ha eliminado una campaña. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada de resúmenes de todas las campañas, ejecute el siguiente comando:

```
aws iotfleetwise list-campaigns
```

Puede utilizar la operación de la API GetCampaign para recuperar la información del vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar los metadatos de una campaña, ejecute el siguiente comando:

Reemplace *campaign-name* por el nombre de la campaña que desea recuperar.

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

> **Note**
>
> Esta operación es a largo plazo coherente. En otras palabras, los cambios que se hagan en la campaña podrían no reflejarse de inmediato.

Procesamiento y visualización de los datos del vehículo

El software Edge Agent para AWS IoT FleetWise transfiere datos seleccionados del vehículo a Amazon Timestream o a Amazon Simple Storage Service (Amazon S3). Cuando los datos hayan llegado al destino, podrá utilizar otros servicios de AWS para visualizarlos y compartirlos.

Procesamiento de los datos del vehículo en Timestream

Amazon Timestream es una base de datos de serie temporal completamente administrada que puede almacenar y analizar billones de puntos de datos de serie temporal por día. Los datos se almacenan en una tabla de Timestream administrada por el cliente. Puede utilizar Timestream para consultar los datos de los vehículos y obtener información sobre ellos. Para obtener más información, consulte ¿Qué es Amazon Timestream?

El esquema de datos predeterminado que se transfiere a Timestream contiene los siguientes campos:

Nombre del campo	Tipo de datos	Descripción
`eventId`	varchar	El ID del evento de recopilación de datos.
`vehicleName`	varchar	El ID del vehículo del que se han recopilado los datos.
`name`	varchar	El nombre de la campaña que utiliza el software Edge Agent para recopilar datos.
`time`	Marca de tiempo	La marca temporal del punto de datos.
`measure_name`	varchar	El nombre de la señal.
`measure_value::bigint`	bigint	Valores de señal de tipo Entero.

Nombre del campo	Tipo de datos	Descripción	
`measure_value::double`	double	Valores de señal de tipo Doble.	
`measure_value::boolean`	boolean	Valores de señal de tipo Booleano.	

Visualización de los datos del vehículo almacenados en Timestream

Una vez transferidos los datos del vehículo a Timestream, puede usar los siguientes servicios de AWS para visualizarlos, supervisarlos, analizarlos y compartirlos.

- Visualice y supervise los datos en los paneles mediante Grafana o Amazon Managed Grafana. Puede visualizar datos de varios orígenes de AWS (como Amazon CloudWatch y Timestream) y de otros orígenes de datos con un único panel de Grafana.
- Analice y visualice los datos en paneles mediante Amazon QuickSight.

Procesamiento de datos de vehículos en S3

Amazon S3 es un servicio de almacenamiento de objetos que almacena y protege cualquier cantidad de datos. Puede utilizar S3 para diversos casos de uso, tales como lagos de datos, copias de seguridad y restauración, archivado, aplicaciones empresariales, dispositivos de AWS IoT y análisis de macrodatos. Los datos se almacenan en S3 como objetos en buckets. Para obtener más información, consulte ¿Qué es Amazon S3?

El esquema de datos predeterminado que se transfiere a Amazon S3 contiene los siguientes campos:

Nombre del campo	Tipo de datos	Descripción	
`eventId`	varchar	El ID del evento de recopilación de datos.	

Nombre del campo	Tipo de datos	Descripción
`vehicleName`	varchar	El ID del vehículo del que se han recopilado los datos.
`name`	varchar	El nombre de la campaña que utiliza el software Edge Agent para recopilar datos.
`time`	Marca de tiempo	La marca temporal del punto de datos.
`measure_name`	varchar	El nombre de la señal.
`measure_value_BIGINT`	bigint	Valores de señal de tipo Entero.
`measure_value_DOUBLE`	double	Valores de señal de tipo Doble.
`measure_value_BOOLEAN`	boolean	Valores de señal de tipo Booleano.
`measure_value_STRUCT`	struct	Valores de señal de tipo estructura.

Formato de objeto S3

AWS IoT FleetWise transfiere los datos del vehículo a S3, donde se guardan como un objeto. Puede usar el URI del objeto que identifica los datos de forma exclusiva para buscar datos de la campaña. El formato del URI del objeto de S3 depende de si los datos recopilados son datos no estructurados o procesados.

Datos no estructurados

Los datos no estructurados se almacenan en S3 de una manera no predefinida. Puede estar en varios formatos, como imágenes o vídeos.

Los mensajes de vehículos enviados a AWS IoT FleetWise con datos de señal de los archivos de Amazon Ion se decodifican y se transfieren a S3 como objetos. Los objetos de S3 representan cada señal y están codificados en binario.

El URI del objeto de S3 de datos utiliza el formato siguiente:

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

Datos procesados

Los datos procesados se almacenan en S3 y se someten a pasos de procesamiento que validan, enriquecen y transforman los mensajes. Las listas de objetos y la velocidad son ejemplos de datos procesados.

Los datos transferidos a S3 se almacenan como objetos que representan registros almacenados en búfer durante un periodo de unos 10 minutos. De forma predeterminada, AWS IoT FleetWise agrega la hora UTC en formato `year=YYYY/month=MM/date=DD/hour=HH` como prefijo antes de escribir los objetos en S3. Este prefijo crea una jerarquía lógica en el bucket en la que cada barra inclinada (/) crea un nivel jerárquico. Los datos procesados también contienen el URI del objeto de S3 para los datos no estructurados.

El URI del objeto de S3 de datos procesados usa el formato siguiente:

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

Datos sin procesar

Los datos sin procesar, también conocidos como datos primarios, son datos recopilados de los archivos de Amazon Ion. Puede utilizar los datos sin procesar para solucionar cualquier problema o para determinar la causa raíz de los errores.

El URI del objeto de S3 de datos sin procesar usa el formato siguiente:

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

Análisis de los datos del vehículo almacenados en S3

Una vez transferidos los datos del vehículo a S3, puede usar los siguientes servicios de AWS para monitorearlos, analizarlos y compartirlos.

Extraiga y analice datos con Amazon SageMaker para flujos de trabajo posteriores de etiquetado y machine learning (ML).

Para obtener más información, consulte los siguientes temas en la Guía para desarrolladores de Amazon SageMaker:

- Procesamiento de datos
- Formación de modelos de machine learning
- Etiquetar imágenes

Catalogue los datos con Rastreador de AWS Glue y analícelos en Amazon Athena. De forma predeterminada, los objetos escritos en S3 tienen particiones temporales al estilo de Apache Hive, con rutas de datos que contienen pares clave-valor conectados por signos iguales.

Para obtener más información, consulte los siguientes temas en la Guía del usuario de Amazon Athena.

- Particiones de datos en Athena
- Uso de AWS Glue para conectarse a orígenes de datos en Amazon S3
- Prácticas recomendadas para utilizar Athena con AWS Glue

Visualice los datos con Amazon QuickSight leyendo directamente la tabla de Athena o el bucket de S3.

> **Tip**
>
> Si está leyendo directamente desde S3, confirme que los datos del vehículo estén en formato JSON, ya que Amazon QuickSight no admite el formato Apache Parquet.

Para obtener más información, consulte los siguientes temas en la Guía del usuario de Amazon QuickSight.

- Orígenes de datos admitidos
- Creación de un origen de datos

SDK de AWS CLI y AWS

En esta sección se proporciona información sobre la realización de solicitudes API de AWS IoT FleetWise. Para obtener más información sobre las operaciones y los tipos de datos de AWS IoT FleetWise, consulte la referencia de la API de AWS IoT FleetWise.

Si desea usar AWS IoT FleetWise con diferentes lenguajes de programación, puede utilizar los SDK de AWS, que contienen la siguiente funcionalidad automática:

- Firmar criptográficamente sus solicitudes de servicio
- Reintentar solicitudes
- Tratar las respuestas a errores

Para acceder a la línea de comandos, utilice AWS IoT FleetWise con AWS CLI. Puede controlar AWS IoT FleetWise y el resto de los servicios desde la línea de comandos y automatizarlos mediante scripts.

Solución de problemas de AWS IoT FleetWise

Use la información de solución de problemas y las posibles soluciones de esta sección para resolver problemas que pueden presentarse en AWS IoT FleetWise.

La siguiente información puede ayudarle a solucionar problemas comunes con AWS IoT FleetWise.

Temas

Problemas con el manifiesto del decodificador

Solucione problemas con el manifiesto del decodificador.

Diagnóstico de las llamadas a la API del manifiesto del decodificador

Error	Directrices para solucionar problemas
`UpdateOperationFailure.ConflictingDecoderUpdate`	El mismo manifiesto del decodificador tiene varias solicitudes de actualización. Espere e inténtelo de nuevo.
`UpdateOperationFailure.InternalFailure`	InternalFailure se lanza como una excepción encapsulada. El problema en sí depende de la excepción encapsulada.
`UpdateOperationFailure.ActiveDecoderUpdate`	El manifiesto del decodificador está en un estado `Active` y no se puede actualizar. Cambie el estado manifiesto del decodificador a `DRAFT` y, a continuación, vuelva a intentarlo.
`UpdateOperationFailure.ConflictingModelUpdate`	AWS IoT FleetWise está intentando validarlo con un modelo de vehículo (manifiesto de modelo) modificado por otra persona. Espere e inténtelo de nuevo.

Error	Directrices para solucionar problemas
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND`	El modelo de vehículo no tiene ninguna señal asociada. Agregue señales al modelo del vehículo y compruebe que las señales se encuentran en el catálogo de señales asociado.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE`	Actualice el modelo de vehículo para que esté en estado `ACTIVE` y, a continuación, vuelva a intentarlo.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_FOUND`	AWS IoT FleetWise no puede encontrar el modelo de vehículo asociado al manifiesto del decodificador. Compruebe el nombre de recurso de Amazon (ARN) del modelo de vehículo y, a continuación, vuelva a intentarlo.
`UpdateOperationFailure.ModelManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE`	No se ha podido validar el modelo de vehículo porque no se encontraron los nombres de las señales del modelo de vehículo en el catálogo de señales. Compruebe que todas las señales del modelo de vehículo estén incluidas en el catálogo de señales asociado.
`UpdateOperationFailure.ValidationFailure`	Se encontraron señales o interfaces de red que no son válidas en la solicitud de actualización del manifiesto del decodificador. Compruebe que existan todas las señales e interfaces de red devueltas por la excepción, que todas las señales utilizadas estén asociadas a una interfaz disponible y que no vaya a eliminar ninguna interfaz que tenga señales asociadas.
`UpdateOperationFailure.KmsKeyAccessDenied`	Hay un problema de permiso con la clave de AWS Key Management Service (AWS KMS) utilizada para la operación. Compruebe que utiliza un rol que tiene acceso a la clave e inténtelo de nuevo.

Error	Directrices para solucionar problemas
`UpdateOperationFailure.DecoderDoesNotExist`	El manifiesto del decodificador no existe. Compruebe el nombre del manifiesto del decodificador y, a continuación, vuelva a intentarlo.

Los mensajes de error de datos de sistemas de visión con el motivo `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` incluirán una pista en la respuesta que proporcionará información sobre el motivo del error en la solicitud. Puede usar la sugerencia para determinar qué directrices de solución de problemas debe seguir.

> **Note**
>
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Diagnóstico de la validación de datos de sistemas de visión del manifiesto del decodificador

Error	Directrices para solucionar problemas
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)`	AWS IoT FleetWise no encontró la estructura de señal raíz utilizada en el decodificador de señales mediante el catálogo de señales. Compruebe que la señal raíz de la estructura esté correctamente definida en el catálogo de señales.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	No se definió un mensaje primitivo del catálogo de señales con el mismo tipo de datos en la solicitud de actualización del manifiesto del decodificador. Compruebe que los mensajes primitivos definidos en la solicitud coincidan con la definición del catálogo de señales correspondiente.

Error	Directrices para solucionar problemas
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)`	El número de propiedades definidas en una estructura del catálogo de señales no coincide con el número de propiedades que está intentando decodificar en el manifiesto del decodificador. Compruebe que tiene el número correcto de señales para decodificar en comparación con las señales definidas en el catálogo de señales.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	AWS IoT FleetWise encontró una señal definida como una ESTRUCTURA en el catálogo de señales sin una structuredMessageDefinition definida en la solicitud de manifiesto del decodificador. Asegúrese de que cada estructura esté definida como structuredMessageDefinition en la solicitud de actualización del manifiesto del decodificador.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	La señal raíz de la estructura utilizada en el manifiesto del decodificador no está definida correctamente como una estructura del catálogo de señales. La estructura de señal raíz utilizada en el manifiesto del decodificador debe tener definido su campo structFullyQualifiedName. También necesita un nodo de ESTRUCTURA con ese fullyQualifiedName.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	Uno de los mensajes de hoja utilizados en la solicitud de manifiesto del decodificador no está definido como mensaje primitivo. Compruebe que todos los objetos de hoja de la solicitud estén definidos como mensajes primitivos.

Error	Directrices para solucionar problemas
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	En la solicitud de actualización del manifiesto del decodificador, no se definió un objeto de matriz del catálogo de señales como structuredMessageListDefinition. Compruebe que todas las propiedades de la matriz estén definidas como structuredMessageListDefinition en la solicitud de actualización del manifiesto del decodificador.

Problemas con el software Edge Agent para AWS IoT FleetWise

Solucione problemas con el software Edge Agent.

Problemas

- Problema: el software Edge Agent no se inicia.
- Problema: [ERROR] [IoTFleetWiseEngine::connect]: [Failed to init persistency library]
- Problema: el software Edge Agent no recopila los PID de diagnóstico a bordo (OBD) II ni los códigos de diagnóstico de problemas (DTC).
- Problema: el software Edge Agent para AWS IoT FleetWise no recopila datos de la red o no puede aplicar las normas de inspección de datos.
- Problema: [ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error] o [WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]

Problema: el software Edge Agent no se inicia.

Es posible que se muestren los siguientes errores si el software Edge Agent no se inicia.

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

  Solución: asegúrese de que el archivo de configuración del software Edge Agent para AWS IoT FleetWise utilice un formato JSON válido. Por ejemplo, compruebe que las comas se hayan utilizado correctamente. Para obtener más información sobre el archivo de configuración, haga
  ```

lo siguiente para descargar la Guía para desarrolladores del software Edge Agent para AWS IoT FleetWise.

1. Vaya a la consola de AWS IoT FleetWise.
2. En la página de inicio del servicio, en la sección Introducción a AWS IoT FleetWise, elija Conozca el agente de borde.

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

  Solución: es posible que se muestre este error cuando el software Edge Agent no pueda establecer la comunicación por socket con las interfaces de red definidas en el archivo de configuración.

  Para verificar que todas las interfaces de red definidas en la configuración están disponibles, ejecute el siguiente comando:

  ```
  ip link show
  ```

  Para conectar una interfaz de red, ejecute el siguiente comando: Reemplace *network-interface-id* por el ID la interfaz de red.

  ```
  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]
  ```

  Solución: es posible que se muestre este error cuando el software Edge Agent no pueda establecer una conexión MQTT con AWS IoT Core. Compruebe que los siguientes elementos estén configurados correctamente y reinicie el software Edge Agent.

  - `mqttConnection::endpointUrl`: punto de conexión del dispositivo con IoT de la cuenta de AWS.
  - `mqttConnection::clientID`: el ID del vehículo en el que se ejecuta el software Edge Agent.
  - `mqttConnection::certificateFilename`: la ruta al archivo del certificado del vehículo.
  - `mqttConnection::privateKeyFilename`: la ruta al archivo de la clave privada del vehículo.

- Ha utilizado AWS IoT Core para aprovisionar el vehículo. Para obtener más información, consulte Aprovisionamiento de vehículos.

Para obtener más información, consulte AWS IoT Device SDK para C++Preguntas frecuentes en .

Problema: [ERROR] [IoTFleetWiseEngine::connect]: [Failed to init persistency library]

Solución: es posible que se muestre este error si el software Edge Agent no encuentra el almacenamiento de persistencia. Compruebe que los siguientes elementos estén configurados correctamente y reinicie el software Edge Agent.

`persistency:persistencyPath`: una ruta local que se utiliza para conservar los esquemas de recopilación, los manifiestos del decodificador y las instantáneas de datos.

Problema: el software Edge Agent no recopila los PID de diagnóstico a bordo (OBD) II ni los códigos de diagnóstico de problemas (DTC).

Solución: es posible que aparezca este error si `obdInterface:pidRequestIntervalSeconds` o `obdInterface:dtcRequestIntervalSeconds` están configurados en 0.

Si el software Edge Agent se ejecuta en un vehículo con transmisión automática, asegúrese de que `obdInterface:hasTransmissionEcu` está configurado como `true`.

Si el vehículo admite ID de arbitraje de red de área de control extendida (bus CAN), asegúrese de que `obdInterface:useExtendedIds` esté configurado como `true`.

Problema: el software Edge Agent para AWS IoT FleetWise no recopila datos de la red o no puede aplicar las normas de inspección de datos.

Solución:: es posible que se muestre este error si se incumplen las cuotas predeterminadas.

Resource	Cuota	Ajustable	Nota
Valor del ID de la señal	El ID de la señal debe ser igual o inferior a 50 000	Sí	El software Edge Agent no recopilar á datos de señales

Resource	Cuota	Ajustable	Nota
			que tengan un ID superior a 50 000. Le recomendamos que compruebe cuántas señales contiene el catálogo de señales antes de cambiar esta cuota.
Número de esquemas activos de recopilación de datos por vehículo	256	Sí	Le recomendamos que compruebe cuántas campañas ha creado en la nube y cuántos esquemas contiene cada campaña antes de cambiar esta cuota.
Tamaño del búfer de historial de señales	20 MB	Sí	Si se supera la cuota, el software Edge Agent deja de recopilar nuevos datos.

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

Solución: es posible que se muestre este error cuando el software Edge Agent no esté conectado a la nube. De forma predeterminada, el software Edge Agent envía una solicitud de ping a AWS IoT Core cada minuto y espera tres minutos. Si no hay respuesta, el software Edge Agent restablece automáticamente la conexión a la nube.

Seguridad en el AWS IoT FleetWise

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a la AWS IoT FleetWise, consulte Servicios de AWS dentro del alcance por programa de conformidad .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar el AWS IoT FleetWise. Le muestra cómo configurar el AWS IoT FleetWise para cumplir sus objetivos de seguridad y conformidad. También aprenderá a usar otros AWS servicios que lo ayudan a monitorear y proteger sus FleetWise recursos de AWS IoT.

Contenido

Protección de datos en AWS IoT FleetWise

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en el AWS IoT FleetWise. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabajas con AWS IoT FleetWise u otro tipo de Servicios de AWS aplicaciones mediante la consola, la API o AWS los SDK. AWS CLI Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear

para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

AWS FleetWise El IoT está diseñado para usarse con un agente perimetral que usted desarrolle e instale en el hardware de un vehículo compatible para transmitir los datos del vehículo a la AWS nube. La extracción de los datos de los vehículos puede estar sujeta a las normas de privacidad de datos en determinadas jurisdicciones. Antes de usar AWS IoT FleetWise e instalar su Edge Agent, le recomendamos encarecidamente que evalúe sus obligaciones de cumplimiento en virtud de la legislación aplicable. Esto incluye cualquier requisito legal aplicable para proporcionar avisos de privacidad legalmente adecuados y obtener cualquier consentimiento necesario para extraer los datos del vehículo.

Cifrado en reposo

Los datos recopilados de un vehículo se transmiten a la nube a través de un AWS IoT Core mensaje con el protocolo de mensajes MQTT. AWS El IoT FleetWise entrega los datos a su base de datos de Amazon Timestream. En Timestream, los datos están cifrados. Todos los datos en reposo Servicios de AWS cifran de forma predeterminada.

El cifrado en reposo se integra con AWS Key Management Service (AWS KMS) para administrar la clave de cifrado que se utiliza para cifrar los datos. Puede optar por utilizar una clave gestionada por el cliente para cifrar los datos recopilados por el AWS IoT FleetWise. Puede crear, administrar y ver su clave de cifrado mediante AWS KMS ella. Para obtener más información, consulte ¿Qué es AWS Key Management Service? en la Guía para AWS Key Management Service desarrolladores.

Cifrado en tránsito

Todos los datos intercambiados con AWS IoT los servicios se cifran en tránsito mediante Transport Layer Security (TLS). Para obtener más información, consulte Seguridad de transporte en la Guía del desarrollador de AWS IoT .

Además, AWS IoT Core admite la autenticación y la autorización para ayudar a controlar de forma segura el acceso a FleetWise los recursos de AWS IoT. Los vehículos pueden usar certificados X.509 para autenticarse (iniciar sesión) para usar AWS IoT FleetWise y usar AWS IoT Core políticas para obtener autorización (tener permisos) para realizar acciones específicas. Para obtener más información, consulte the section called “Aprovisionamiento de vehículos”.

Cifrado de datos

El cifrado de datos se refiere a la protección de los datos mientras están en tránsito (mientras viajan hacia y desde el AWS IoT FleetWise, y entre las puertas de enlace y los servidores) y en reposo (mientras están almacenados en dispositivos locales o dentro). Servicios de AWS Los datos en reposo pueden protegerse mediante el cifrado del cliente.

> **Note**
>
> AWS El procesamiento FleetWise perimetral de IoT expone las API que están alojadas en las FleetWise puertas de enlace de AWS IoT y a las que se puede acceder a través de la red local. Estas API se exponen a través de una conexión TLS respaldada por un certificado de servidor propiedad del conector AWS IoT FleetWise Edge. Para la autenticación de los clientes, estas API utilizan una contraseña de control de acceso. Tanto la clave privada del certificado del servidor como la contraseña de control de acceso se almacenan en el disco. AWS El procesamiento FleetWise perimetral de IoT se basa en el cifrado del sistema de archivos para garantizar la seguridad de estas credenciales en reposo.

Para obtener más información sobre el cifrado del lado del servidor y el cifrado del cliente, revise los temas siguientes.

Contenido

Cifrado en reposo

AWS El IoT FleetWise almacena sus datos en la AWS nube y en pasarelas.

Datos en reposo en la nube AWS

AWS El IoT FleetWise almacena los datos en otros Servicios de AWS que cifran los datos en reposo de forma predeterminada. Encryption at rest se integra con AWS Key Management Service (AWS KMS) para administrar la clave de cifrado que se utiliza para cifrar los valores de las propiedades de sus activos y los valores agregados en el AWS IoT FleetWise. Puede optar por utilizar una clave gestionada por el cliente para cifrar los valores de las propiedades de los activos y los valores

agregados en el AWS IoT FleetWise. Puede crear, administrar y ver su clave de cifrado mediante AWS KMS ella.

Puede elegir una clave gestionada por el cliente Clave propiedad de AWS o una clave gestionada por el cliente para cifrar sus datos.

Funcionamiento

El cifrado en reposo se integra AWS KMS para administrar la clave de cifrado que se utiliza para cifrar los datos.

- Clave propiedad de AWS — Clave de cifrado predeterminada. AWS El IoT FleetWise es el propietario de esta clave. No puede ver, administrar ni utilizar esta clave en su Cuenta de AWS. Tampoco puedes ver las operaciones de la clave en AWS CloudTrail los registros. Puede usar esta clave sin cargo adicional.
- Clave administrada por el cliente: la clave se almacena en la cuenta y usted la crea, posee y administra. Usted controla plenamente la clave KMS. Se aplican AWS KMS cargos adicionales.

Claves propiedad de AWS

Claves propiedad de AWS no están almacenados en tu cuenta. Forman parte de una colección de claves de KMS que AWS posee y administra para su uso en múltiples direcciones Cuentas de AWS. Servicios de AWS se pueden utilizar Claves propiedad de AWS para proteger sus datos.

No puede ver, administrar Claves propiedad de AWS, usar ni auditar su uso. Sin embargo, no necesita realizar ninguna acción ni cambiar ningún programa para proteger las claves que cifran los datos.

No se te cobrará ninguna comisión si las utilizas Claves propiedad de AWS y no se tienen en cuenta las AWS KMS cuotas de tu cuenta.

Claves administradas por el cliente

Las claves administradas por el cliente son claves KMS en su cuenta que usted ha creado, posee y administra. Tiene el control total sobre estas claves KMS, lo que significa que puede hacer lo siguiente:

- Establecer y mantener sus políticas de claves, políticas de IAM y concesiones.
- Activarlas y desactivarlas.
- Rotar sus materiales criptográficos.

- Agregar etiquetas.
- Crear alias que hagan referencia a ellas.
- Programar su eliminación.

También puedes usar CloudTrail Amazon CloudWatch Logs para realizar un seguimiento de las solicitudes que el AWS IoT FleetWise envía AWS KMS en tu nombre.

Si utilizas claves administradas por el cliente, debes conceder FleetWise acceso de AWS IoT a la clave de KMS almacenada en tu cuenta. AWS El IoT FleetWise utiliza el cifrado de sobres y la jerarquía de claves para cifrar los datos. Su clave de cifrado de AWS KMS se utiliza para cifrar la clave raíz de esta jerarquía de claves. Para obtener más información, consulte Cifrado de sobre en la Guía para desarrolladores de AWS Key Management Service .

El siguiente ejemplo de política otorga FleetWise permisos de AWS IoT para crear una clave gestionada por el cliente en su nombre.

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

Important

Cuando añada las nuevas secciones a su política de claves de KMS, no cambie ninguna sección existente en la política. AWS El IoT no FleetWise puede realizar operaciones con

sus datos si el cifrado está habilitado para la AWS IoT FleetWise y se cumple alguna de las siguientes condiciones:

- Se deshabilita o se elimina la clave KMS.
- La política de claves de KMS no está configurada correctamente para el servicio.

Uso de datos de sistemas de visión con cifrado en reposo

> **Note**
>
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.

Si tiene el cifrado gestionado por el cliente con AWS KMS claves habilitadas en su FleetWise cuenta de AWS IoT y desea utilizar los datos del sistema de visión, restablezca la configuración de cifrado para que sea compatible con tipos de datos complejos. Esto permite FleetWise al AWS IoT establecer los permisos adicionales necesarios para los datos del sistema de visión.

> **Note**
>
> El manifiesto del decodificador podría quedar bloqueado en estado de validación si no ha restablecido la configuración de cifrado de los datos de sistemas de visión.

1. Utilice la operación GetEncryptionConfigurationde la API para comprobar si el AWS KMS cifrado está activado. No es necesario realizar ninguna otra acción si el tipo de cifrado es `FLEETWISE_DEFAULT_ENCRYPTION`.
2. Si el tipo de cifrado es`KMS_BASED_ENCRYPTION`, utilice la operación de PutEncryptionConfigurationAPI para restablecer el tipo de cifrado a`FLEETWISE_DEFAULT_ENCRYPTION`.

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

3. Utilice la operación PutEncryptionConfigurationde API para volver a habilitar el tipo de cifrado en. `KMS_BASED_ENCRYPTION`

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

Para obtener más información acerca de cómo habilitar el cifrado, consulte Administración de claves.

Administración de claves

AWS Gestión de claves FleetWise en la nube de IoT

De forma predeterminada, AWS IoT FleetWise utiliza Claves administradas por AWS para proteger sus datos en el Nube de AWS. Puede actualizar la configuración para usar una clave administrada por el cliente para cifrar los datos en el AWS IoT FleetWise. Puede crear, administrar y ver su clave de cifrado mediante AWS Key Management Service (AWS KMS).

AWS FleetWise El IoT admite el cifrado del lado del servidor con claves administradas por el cliente almacenadas AWS KMS para cifrar los datos de los siguientes recursos.

AWS FleetWise Recurso de IoT	Tipo de datos	Campos que están cifrados en reposo con claves administradas por el cliente
Catálogo de señales		description
	Atributo	description, allowedValues, defaultValue, min, max
	Actuador	description, allowedValues, min, max
	Sensor	description, allowedValues, min, max
Modelo de vehículo (manifiesto del modelo)		description

AWS FleetWise Recurso de IoT	Tipo de datos	Campos que están cifrados en reposo con claves administradas por el cliente
Manifiesto del decodificador		description
	CanInterface	protocolName, protocolVersion
	ObdInterface	requestMessageId, dtcReques tInterval Segundos hasTransm issionEcu, estándar OBD, segundos, pidRequestInterval useExtendedIds
	CanSignal	factor, isBigEndian isSigned, longitud, messageID, offset, StartBit
	ObdSignal	ByteLength, offset, pid, scaling, ServiceMode, pidResponseLength StartByte, bitMaskLength bitRightS hift
Vehículo		attributes
Campaña		description
	conditionBasedCollectionEsquema	expresión, minimumTriggerInte rval Ms conditionLanguageVersion, TriggerMode
	TimeBasedCollectionScheme	periodMs

> **Note**
>
> Otros datos y recursos se cifran mediante el cifrado predeterminado con claves gestionadas por el AWS IoT FleetWise. Esta clave se crea y almacena en la FleetWise cuenta de AWS IoT.

Para obtener más información, consulta ¿Qué es AWS Key Management Service? en la Guía para AWS Key Management Service desarrolladores.

Activación del cifrado mediante claves KMS (consola)

Para usar claves administradas por el cliente con AWS IoT FleetWise, debes actualizar tu FleetWise configuración de AWS IoT.

Para habilitar el cifrado mediante claves KMS (consola)

1. Abre la FleetWise consola AWS de IoT.
2. Vaya a Configuración.
3. En Cifrado, elija Editar para abrir la página Editar cifrado.
4. En Tipo de clave de cifrado, selecciona Elegir una AWS KMS clave diferente. Esto habilita el cifrado con las claves administradas por el cliente almacenadas en AWS KMS.

   > **Note**
   >
   > Solo puede usar el cifrado de claves gestionado por el cliente para FleetWise los recursos de AWS IoT. Esto incluye el catálogo de señales, el modelo del vehículo (manifiesto del modelo), el manifiesto del decodificador, el vehículo, la flota y la campaña.

5. Elija la clave KMS con una de las siguientes opciones:

   - Para usar una clave KMS existente: elija el alias de su clave KMS de la lista.
   - Para crear una nueva clave KMS, selecciona Crear una AWS KMS clave.

     > **Note**
     >
     > Esto abre la AWS KMS consola. Para obtener más información sobre la creación de una clave KMS, consulte Creación de claves en la Guía para desarrolladores de AWS Key Management Service .

6. Elija Guardar para actualizar la configuración.

Habilitación del cifrado mediante claves KMS (AWS CLI).

Puedes usar la operación de la PutEncryptionConfigurationAPI para habilitar el cifrado de tu FleetWise cuenta de AWS IoT. En el siguiente ejemplo, se utiliza AWS CLI.

Para activar el cifrado, ejecute el siguiente comando:

- Reemplace *KMS key id* por el ID de la clave KMS.

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

Example Respuesta

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

Política de claves de KMS

Después de crear una clave de KMS, debe añadir, como mínimo, la siguiente declaración a su política de claves de KMS para que funcione con el AWS IoT FleetWise.

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

```
}
```

Para obtener más información sobre cómo editar una política de claves de KMS para usarla con AWS IoT FleetWise, consulte Cambiar una política clave en la Guía para AWS Key Management Service desarrolladores.

> **Important**
>
> Cuando añada las nuevas secciones a su política de claves de KMS, no cambie ninguna sección existente de la política. AWS El IoT no FleetWise puede realizar operaciones con sus datos si el cifrado está habilitado para la AWS IoT FleetWise y se cumple alguna de las siguientes condiciones:
>
> - Se deshabilita o se elimina la clave KMS.
> - La política de claves de KMS no está configurada correctamente para el servicio.

Controlar el acceso con AWS IoT FleetWise

En las siguientes secciones, se explica cómo controlar el acceso a los AWS IoT FleetWise recursos y desde ellos. La información que cubren incluye cómo conceder acceso a tu aplicación para que el AWS IoT FleetWise pueda transferir los datos de los vehículos durante las campañas. También describen cómo puede conceder AWS IoT FleetWise acceso a su bucket de Amazon S3 (S3) o a la base de datos y tabla de Amazon Timestream para almacenar datos.

La tecnología para gestionar todas estas formas de acceso es la AWS Identity and Access Management (IAM). Para obtener más información acerca de IAM, consulte ¿Qué es IAM?.

Contenido

Conceder AWS IoT FleetWise acceso a un destino de Amazon S3

Cuando utiliza un destino de Amazon S3, AWS IoT FleetWise entrega los datos del vehículo a su depósito de S3 y, si lo desea, puede utilizar una AWS KMS clave de su propiedad para el cifrado de datos. Si el registro de errores está activado, AWS IoT FleetWise también envía los errores de

entrega de datos a su grupo de CloudWatch registros y a sus transmisiones. Es obligatorio contar con un rol de IAM; al crear un flujo de entrega.

AWS IoT FleetWise usa una política de bucket con el principal de servicio para el destino S3. Para obtener más información sobre cómo agregar o modificar políticas de bucket, consulte Agregar una política de bucket mediante la consola de Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

Use la siguiente política de acceso para permitir el acceso AWS IoT FleetWise a su bucket de S3. Si no es el propietario del bucket de S3, agregue `s3:PutObjectAcl` a la lista de acciones de Amazon S3. Esto otorga al propietario del bucket acceso total a los objetos entregados por AWS IoT FleetWise. Para obtener más información sobre cómo proteger el acceso a los objetos de sus buckets, consulte los ejemplos de políticas de bucket en la Guía del usuario de Amazon Simple Storage Service.

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

La siguiente política de bucket se aplica a todas las campañas de una cuenta de una AWS región.

```
{
  "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"
```

```
        }
      }
    }
  ]
}
```

Si tiene una clave KMS adjunta a su bucket de S3, la clave necesitará la siguiente política. Para obtener información sobre la administración de claves, consulte Protección de datos mediante cifrado con AWS Key Management Service claves del lado del servidor (SSE-KMS) en la Guía del usuario de Amazon Simple Storage Service.

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

> **Important**
>
> Cuando crea un bucket, S3 crea unas listas de control de acceso (ACL) predeterminadas que conceden al propietario del recurso control total sobre el recurso. Si el AWS IoT no FleetWise puede entregar datos a S3, asegúrese de deshabilitar la ACL en el bucket de S3. Para obtener más información, consulte Desactivación de las ACL para todos los buckets nuevos y aplicación de la propiedad de objetos en la Guía del usuario de Amazon Simple Storage Service.

Conceder AWS IoT FleetWise acceso a un destino de Amazon Timestream

Cuando utilizas un destino de Timestream, AWS IoT FleetWise entrega los datos del vehículo a una tabla de Timestream. Debe adjuntar las políticas a la función de IAM para poder enviar datos a AWS IoT FleetWise Timestream.

Si utilizas la consola para crear una campaña, AWS IoT adjunta FleetWise automáticamente la política requerida al rol.

Antes de comenzar, compruebe lo siguiente:

> **Important**
>
> - Debe usar la misma AWS región al crear recursos de Timestream para IoT AWS . FleetWise Si cambias de AWS región, es posible que tengas problemas para acceder a los recursos de Timestream.
> - AWS FleetWise El IoT está disponible en EE. UU. Este (Virginia del Norte) y Europa (Fráncfort).
> - Para ver una lista completa de las regiones admitidas, consulte Puntos de conexión y cuotas de Timestream en la Referencia general de AWS.

- Debe contar con una base de datos de Timestream. Para ver un tutorial, consulte Crear una base de datos en la Guía para desarrolladores de Amazon Timestream.
- Debe tener una tabla creada en la base de datos de Timestream especificada. Para ver un tutorial, consulte Crear una tabla en la Guía para desarrolladores de Amazon Timestream.

Puede utilizarla AWS CLI para crear un rol de IAM con una política de confianza para Timestream. Para crear un rol de IAM, ejecute el siguiente comando.

Creación de un rol de IAM con una política de confianza

- *TimestreamExecutionRole*Sustitúyalo por el nombre del rol que está creando.
- Reemplace *trust-policy* por el archivo JSON que contenga la política de confianza.

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

Cree una política de permisos para conceder a AWS IoT FleetWise permisos para escribir datos en Timestream. Para crear una política de permisos, ejecute el siguiente comando.

Creación de una política de permisos

- *AWSIoTFleetwiseAccessTimestreamPermissionsPolicy*Sustitúyala por el nombre de la política que estás creando.
- Reemplace *permissions-policy* por el nombre del archivo JSON que contenga la política de permisos.

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

Asociación de la política de permisos a su rol de IAM.

1. En el resultado, copie el Nombre de recurso de Amazon (ARN) de la política de permisos.
2. Para asociar la política de permisos de IAM a su rol de IAM, ejecute el comando siguiente.

   - *permissions-policy-arn*Sustitúyalo por el ARN que copió en el paso anterior.
   - *TimestreamExecutionRole*Sustitúyalo por el nombre del rol de IAM que creaste.

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

Para obtener más información, consulte Administración de accesos para recursos de AWS en la Guía del usuario de IAM.

Identity and Access Management para AWS IoT FleetWise

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos)

para usar los recursos de AWS IoT FleetWise . La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que se realice en el AWS IoT FleetWise.

Usuario del servicio: si utilizas el FleetWise servicio de AWS IoT para realizar tu trabajo, el administrador te proporcionará las credenciales y los permisos que necesitas. A medida que vaya utilizando más FleetWise funciones de AWS IoT para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función de AWS IoT FleetWise, consulteSolución de problemas AWS de FleetWise identidad y acceso a IoT.

Administrador de servicios: si está a cargo de FleetWise los recursos de AWS IoT en su empresa, es probable que tenga acceso total a la AWS IoT FleetWise. Es su trabajo determinar a qué FleetWise funciones y recursos de AWS IoT deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM con el AWS IoT FleetWise, consulteCómo FleetWise funciona el AWS IoT con IAM.

Administrador de IAM: si es administrador de IAM, puede que le interese obtener más información sobre cómo redactar políticas para administrar el acceso a la IoT AWS . FleetWise Para ver ejemplos de políticas de AWS IoT FleetWise basadas en la identidad que puede usar en IAM, consulte. Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Autenticación con identidades

La autenticación es la forma en que inicias sesión para AWS usar tus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWSen la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los

permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre

la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad

principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de

Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations .

- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo FleetWise funciona el AWS IoT con IAM

Antes de utilizar la IAM para gestionar el acceso a la AWS IoT FleetWise, infórmese sobre las funciones de IAM disponibles para su uso con la IoT AWS . FleetWise

Funciones de IAM que puede usar con IoT AWS FleetWise

Característica de IAM	AWS FleetWise Soporte de IoT
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial

Característica de IAM	AWS FleetWise Soporte de IoT
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan el AWS IoT FleetWise y otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para el IoT AWS FleetWise

Compatibilidad con las políticas basadas en identidad	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Para ver ejemplos de políticas de AWS IoT FleetWise basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Políticas basadas en recursos dentro del IoT AWS FleetWise

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones políticas para el AWS IoT FleetWise

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no

tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de FleetWise acciones de AWS IoT, consulte Acciones definidas por AWS IoT FleetWise en la Referencia de autorización de servicios.

Las acciones políticas en AWS IoT FleetWise usan el siguiente prefijo antes de la acción:

```
iotfleetwise
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `List`, incluya la siguiente acción:

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

Para ver ejemplos de políticas de AWS IoT FleetWise basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Recursos de políticas para AWS IoT FleetWise

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica

recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de FleetWise recursos de AWS IoT y sus ARN, consulte Recursos definidos por AWS IoT FleetWise en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por el IoT AWS. FleetWise

Para ver ejemplos de políticas de AWS IoT FleetWise basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Claves de condiciones de políticas para AWS IoT FleetWise

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de FleetWise condición de AWS IoT, consulte Claves de condición para AWS IoT FleetWise en la Referencia de autorización de servicios. Para saber con qué acciones y recursos puede usar una clave de condición, consulte Acciones definidas por el AWS IoT FleetWise .

Para ver ejemplos de políticas de AWS IoT FleetWise basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

Listas de control de acceso (ACL) en IoT AWS FleetWise

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con IoT AWS FleetWise

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

> **Note**
>
> AWS IoT FleetWise solo es compatible`iam:PassRole`, lo cual es necesario para el funcionamiento de la `CreateCampaign` API.

Uso de credenciales temporales con AWS IoT FleetWise

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda

generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales de servicios cruzados para IoT AWS FleetWise

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Funciones de servicio para AWS IoT FleetWise

Compatible con roles de servicio	No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría afectar la FleetWise funcionalidad de AWS IoT. Edite las funciones de servicio solo cuando el AWS IoT FleetWise proporcione instrucciones para hacerlo.

Funciones vinculadas a servicios para el IoT AWS FleetWise

Compatible con roles vinculados al servicio	No

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Uso de roles vinculados a servicios para AWS IoT FleetWise

AWS IoT FleetWise utiliza funciones AWS Identity and Access Management vinculadas a servicios (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a AWS IoT. FleetWise AWS IoT predefine las funciones vinculadas al servicio FleetWise e incluyen los permisos que AWS IoT FleetWise necesita para enviar métricas a Amazon. CloudWatch Para obtener más información, consulte Supervisión de AWS IoT FleetWise con Amazon CloudWatch.

Un rol vinculado a un servicio FleetWise agiliza la configuración de AWS IoT, ya que no es necesario añadir manualmente los permisos necesarios. AWS IoT FleetWise define los permisos de sus funciones vinculadas a servicios y, a menos que se defina lo contrario, solo AWS IoT FleetWise puede asumir sus funciones. Los permisos definidos incluyen la política de confianza y la política de permisos. Dicha política de permisos no se puede asociar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. Esto protege sus FleetWise recursos de AWS IoT porque no puede eliminar el permiso de acceso a los recursos de forma inadvertida.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Roles vinculados a servicios. Para ver la documentación del rol vinculado a un servicio, elija una opción Sí correspondiente con un enlace.

Permisos de roles vinculados a servicios para AWS IoT FleetWise

AWS IoT FleetWise usa el rol vinculado a un servicio denominado AWSServiceRoleForIoTFleetWise: una política administrada por AWS que se usa para todos los out-of-the-box permisos de AWS IoT. FleetWise

El rol AWSServiceRoleForIoTFleetWise vinculado al servicio confía en los siguientes servicios para asumir el rol:

- `IoTFleetWise`

La política de permisos de roles denominada AWSIoTFleetwiseServiceRolePolicy permite FleetWise a AWS IoT realizar las siguientes acciones en los recursos especificados:

- Acción: `cloudwatch:PutMetricData` en el recurso *

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio para AWS IoT FleetWise

No necesita crear manualmente un rol vinculado a servicios. Cuando registra una cuenta en la FleetWise consola, la o la AWS CLI AWS API de AWS IoT, AWS IoT FleetWise crea el rol vinculado al servicio para usted. Para obtener más información, consulte Configuración de los ajustes.

Creación de un rol vinculado a un servicio en AWS IoT FleetWise (consola)

No necesita crear manualmente un rol vinculado a servicios. Cuando registra una cuenta en la FleetWise consola de AWS IoT, la AWS CLI o la AWS API, AWS IoT FleetWise crea el rol vinculado al servicio para usted.

Edición de un rol vinculado a un servicio para AWS IoT FleetWise

No puede editar el rol AWSServiceRoleForIoTFleetWise vinculado al servicio en AWS IoT. FleetWise Dado que diversas entidades podrían hacer referencia a cualquier rol vinculado al servicio que haya creado, no puede cambiar el nombre del rol. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Edición de un rol vinculado a un servicio en la Guía del usuario de IAM.

Saneamiento de un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si AWS IoT FleetWise utiliza el rol al intentar eliminar los recursos, es posible que la eliminación no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación. Para obtener información sobre cómo eliminarlos a service-linked-role través de la consola, la AWS CLI o la AWS API, consulte Uso de funciones vinculadas a servicios en la Guía del usuario de IAM.

Si elimina este rol vinculado a un servicio y, a continuación, necesita volver a crearlo, puede registrar una cuenta en AWS IoT. FleetWise FleetWise A continuación, AWS IoT vuelve a crear el rol vinculado al servicio para usted.

Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar FleetWise recursos de AWS IoT. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener detalles sobre las acciones y los tipos de recursos definidos por AWS IoT FleetWise, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición para AWS IoT FleetWise en la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la FleetWise consola AWS de IoT
- Cómo permitir a los usuarios consultar sus propios permisos
- Acceso a los recursos de Amazon Timestream

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar FleetWise los recursos de AWS IoT de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus

políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la FleetWise consola AWS de IoT

Para acceder a la FleetWise consola de AWS IoT, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los FleetWise recursos de AWS IoT que tiene Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la FleetWise consola de AWS IoT, adjunte también la política de AWS IoT FleetWise `ConsoleAccess` o `ReadOnly` AWS gestionada a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

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

Acceso a los recursos de Amazon Timestream

Antes de usar AWS IoT FleetWise, debes registrar tu AWS cuenta, IAM y los recursos de Amazon Timestream para AWS conceder el permiso de FleetWise IoT a los que enviar Nube de AWS datos del vehículo en tu nombre. Para registrarse, necesita:

- Una base de datos de Amazon Timestream.
- Una tabla creada en la base de datos de Amazon Timestream especificada.
- Una función de IAM que permite FleetWise al AWS IoT enviar datos a Amazon Timestream.

Para obtener más información, incluidos procedimientos y políticas de ejemplo, consulte Configurar ajustes.

Solución de problemas AWS de FleetWise identidad y acceso a IoT

Utilice la siguiente información como ayuda para diagnosticar y solucionar problemas comunes que pueden surgir al trabajar con AWS IoT FleetWise e IAM.

Temas

No estoy autorizado a realizar una acción en el AWS IoT FleetWise

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`myVehicle`*, pero no tiene los permisos `iotfleetwise:`*`GetVehicleStatus`*.

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

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *`myVehicle`* mediante la acción `iotfleetwise:`*`GetVehicleStatus`*.

No estoy autorizado a realizar lo siguiente: PassRole

Si recibes un error que indica que no estás autorizado a realizar la `iam:PassRole` acción, tus políticas deben actualizarse para que puedas transferir una función al AWS IoT FleetWise.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un usuario de IAM denominado `marymajor` intenta usar la consola para realizar una acción en el AWS IoT FleetWise. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis FleetWise recursos de AWS IoT

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si el AWS IoT FleetWise admite estas funciones, consulteCómo FleetWise funciona el AWS IoT con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Validación de conformidad para AWS IoT FleetWise

> **Note**
>
> AWS FleetWiseEl IoT no está en el ámbito de ningún programa de AWS cumplimiento.

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

  > **Note**
  >
  > No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en el AWS IoT FleetWise

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja demora. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura AWS global.

> **Note**
>
> Los datos procesados por el AWS IoT FleetWise se almacenan en una base de datos de Amazon Timestream. Timestream admite copias de seguridad en otras zonas o regiones de AWS disponibilidad. Sin embargo, puede escribir su propia aplicación con el SDK de Timestream para consultar los datos y guardarlos en el destino que prefiera.

Para obtener más información sobre Amazon Timestream, consulte la Guía para desarrolladores de Amazon Timestream.

Seguridad de la infraestructura en el AWS IoT FleetWise

Como servicio gestionado, el AWS IoT FleetWise está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder al AWS IoT FleetWise a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puedes llamar a estas operaciones de API desde cualquier ubicación de la red, pero el AWS IoT FleetWise admite políticas de acceso basadas en recursos, que pueden incluir restricciones basadas en la dirección IP de origen. También puede usar FleetWise políticas de AWS IoT para controlar el acceso desde puntos de enlace específicos de Amazon Virtual Private Cloud (Amazon VPC) o VPC específicas. En efecto, esto aísla el acceso a la red a un FleetWise recurso de AWS IoT determinado únicamente de la VPC específica de la red. AWS

Temas

- Conexión al AWS IoT a FleetWise través de un punto final de VPC de interfaz

Conexión al AWS IoT a FleetWise través de un punto final de VPC de interfaz

Puede conectarse directamente al AWS IoT FleetWise mediante un punto de enlace de VPC de interfaz (AWS PrivateLink) en su Virtual Private Cloud (VPC), en lugar de conectarse a través de Internet. Cuando utiliza un punto final de VPC de interfaz, la comunicación entre su VPC e AWS FleetWise IoT se lleva a cabo completamente dentro de la red. AWS Cada punto de conexión de VPC está representado por una o varias Interfaces de red elásticas (ENI) con direcciones IP privadas en las subredes de la VPC.

El punto final de la interfaz de la VPC conecta la VPC directamente al AWS IoT FleetWise sin necesidad de una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión. AWS Direct Connect Las instancias de tu VPC no necesitan direcciones IP públicas para comunicarse con la API de AWS IoT FleetWise.

Para usar el AWS IoT FleetWise a través de su VPC, debe conectarse desde una instancia que esté dentro de la VPC o conectar su red privada a su VPC mediante una (VPN) o. AWS Virtual Private Network AWS Direct Connect Para obtener más información sobre Amazon VPN, consulte Conexiones VPN en la Guía del usuario de Amazon Virtual Private Cloud. Para obtener más información AWS Direct Connect, consulte Crear una conexión en la Guía del AWS Direct Connect usuario.

Puede crear un punto final de VPC de interfaz para conectarse a AWS IoT FleetWise mediante la AWS consola o los comandos AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Creación de un punto de conexión de interfaz.

Después de crear un punto de enlace de VPC de interfaz, si habilita los nombres de host de DNS privados para el punto de enlace, el punto de enlace de AWS IoT predeterminado pasa a ser su FleetWise punto de enlace de VPC. El punto final del nombre de servicio predeterminado para AWS IoT FleetWise tiene el siguiente formato.

```
iotfleetwise.Region.amazonaws.com
```

Si no habilita nombres de alojamiento de DNS privados, Amazon VPC proporciona un nombre de punto de conexión de DNS que puede utilizar en el siguiente formato.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

Para obtener más información, consulte Interface VPC endpoints (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

AWS FleetWise El IoT permite realizar llamadas a todas sus acciones de API dentro de su VPC.

Puede adjuntar políticas de punto de conexión de VPC a un punto de conexión de VPC para controlar el acceso de las entidades principales de IAM. También puede asociar grupos de seguridad con un punto de conexión de VPC para controlar el acceso de entrada y salida en función del origen y el destino del tráfico de red, como un rango de direcciones IP. Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de VPC.

Creación de una política de puntos finales de VPC para IoT AWS FleetWise

Puede crear una política para los puntos de enlace de Amazon VPC para AWS IoT FleetWise a fin de especificar lo siguiente:

- La entidad principal que puede o no puede realizar acciones
- Las acciones que se pueden realizar o no

Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

Example — Política de punto final de VPC para denegar todo acceso desde una cuenta específica AWS

La siguiente política de punto final de VPC deniega a la AWS cuenta *123456789012* todas las llamadas a la API que utilicen el punto final.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
```

```
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example : política de punto de conexión de VPC para permitir el acceso de VPC solo a una entidad principal de IAM especificada (usuario).

La siguiente política de puntos finales de VPC permite el acceso total solo a la lijuan de un usuario en AWS la cuenta 123456789012. A las demás entidades principales de IAM se les deniega el acceso al punto de conexión.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
                ]
            }
        }]
}
```

Example — Política de puntos finales de VPC para acciones de IoT AWS FleetWise

El siguiente es un ejemplo de una política de puntos finales para el AWS IoT FleetWise. *Cuando se conecta a un punto final, esta política otorga acceso a FleetWise las acciones de AWS IoT enumeradas para el usuario de IAM FleetWise en el 123456789012. Cuenta de AWS*

```
{
    "Statement": [
        {
             "Principal": {
                "AWS": [
```

```
                    "arn:aws:iam::123456789012:user/fleetWise"
                },
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "iotfleetwise:ListFleets",
                "iotfleetwise:ListCampaigns",
                "iotfleetwise:CreateVehicle",
            ]
           }
    ]
}
```

Análisis de configuración y vulnerabilidad en AWS IoT FleetWise

Los entornos de IoT pueden constar de un gran número de dispositivos que tienen diversas capacidades, son de larga duración y están distribuidos geográficamente. Estas características hacen que la configuración del dispositivo sea compleja y propensa a errores. Además, dado que los dispositivos a menudo tienen limitaciones de potencia informática, memoria y capacidad de almacenamiento, el uso del cifrado y otras formas de seguridad en los propios dispositivos queda acotado. Los dispositivos a menudo usan software con vulnerabilidades conocidas. Estos factores hacen que los dispositivos de IoT, incluidos los vehículos que recopilan datos para la AWS IoT FleetWise, sean un objetivo atractivo para los piratas informáticos y dificultan su protección de forma continua.

La configuración y los controles de TI son una responsabilidad compartida entre usted AWS y usted, nuestro cliente. Para obtener más información, consulte el modelo de responsabilidad AWS compartida.

Mejores prácticas de seguridad para AWS IoT FleetWise

AWS FleetWise El IoT proporciona una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

Para obtener más información sobre la seguridad, AWS IoT consulte las prácticas recomendadas de seguridad AWS IoT Core en la Guía para AWS IoT desarrolladores

Conceda los mínimos permisos posibles

Siga el principio de privilegios mínimos utilizando el conjunto mínimo de permisos en los roles de IAM. Limite el uso del comodín * para las propiedades `Action` y `Resource` en las políticas de IAM. En su lugar, declare un conjunto finito de acciones y recursos cuando sea posible. Para obtener más información acerca de los privilegios mínimos y otras prácticas recomendadas de política, consulte the section called “Prácticas recomendadas sobre las políticas”.

No registre información confidencial

Debe evitar el registro de credenciales y otra información de identificación personal (PII). Le recomendamos que implemente las siguientes medidas de seguridad:

- No utilice información confidencial en los nombres de los dispositivos.
- No utilices información confidencial en los nombres e identificadores de los FleetWise recursos de AWS IoT, por ejemplo, en los nombres de las campañas, los manifiestos de decodificadores, los modelos de vehículos y los catálogos de señales, o en los identificadores de vehículos y flotas.

Úselo para ver el AWS CloudTrail historial de llamadas de la API

Puedes ver un historial de las llamadas a la FleetWise API de AWS IoT realizadas en tu cuenta con fines de análisis de seguridad y solución de problemas operativos. Para recibir un historial de las llamadas a la FleetWise API de AWS IoT realizadas en tu cuenta, solo tienes que activar CloudTrail la AWS Management Console. Para obtener más información, consulte the section called “Registros de CloudTrail”.

Mantenga sincronizado el reloj del dispositivo

Es importante que la hora del dispositivo sea precisa. Los certificados X.509 tienen una fecha y una hora de caducidad. El reloj del dispositivo se utiliza para comprobar que un certificado de servidor sigue siendo válido. Los relojes de dispositivos pueden variar con el tiempo o las baterías pueden descargarse.

Para obtener más información, consulte las prácticas recomendadas Mantener sincronizado el reloj del dispositivo en la Guía del desarrollador de AWS IoT Core .

Supervisión de AWS IoT FleetWise

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS IoT FleetWise y de sus otras soluciones de AWS. AWS ofrece las siguientes herramientas de supervisión para vigilar AWS IoT FleetWise, informar cuando algo no va bien y tomar medidas automáticamente cuando proceda:

- Amazon CloudWatch supervisa los recursos de AWS y las aplicaciones que ejecuta en AWS en tiempo real. Puede recopilar métricas y realizar un seguimiento de ellas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica alcanza el umbral que se especifique. Por ejemplo, puede hacer que CloudWatch haga un seguimiento del uso de la CPU u otras métricas de las instancias de Amazon EC2 y lanzar nuevas instancias automáticamente cuando sea necesario. Para más información, consulte la Guía del usuario de Amazon CloudWatch.
- Registros de Amazon CloudWatch puede usarse para supervisar y almacenar los archivos de registro desde instancias de Amazon EC2, CloudTrail u otras fuentes, así como para tener acceso a ellos. Registros de Amazon CloudWatch puede supervisar información en los registros y enviarle una notificación cuando se llega a determinados umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulte la Guía del usuario de Registros de Amazon CloudWatch.
- AWS CloudTrail captura las llamadas a la API y los eventos relacionados realizados por su cuenta de Cuenta de AWS o en su nombre. A continuación, entrega los archivos de registro al bucket de Amazon S3 que se especifique. También puede identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para más información, consulte la Guía del usuario de AWS CloudTrail.

Supervisión de AWS IoT FleetWise con Amazon CloudWatch

Las métricas de Amazon CloudWatch son una forma de supervisar los recursos de AWS y su rendimiento. AWS IoT FleetWise envía las métricas a CloudWatch. Puede usar la AWS Management Console, AWS CLI o una API para obtener una lista de las métricas que AWS IoT FleetWise envía a CloudWatch. Para más información, consulte la Guía del usuario de Amazon CloudWatch.

> **Important**
>
> Debe configurar los ajustes para que AWS IoT FleetWise pueda enviar métricas a CloudWatch. Para obtener más información, consulte Configuración de los ajustes.

El espacio de nombres de `AWS/IoTFleetWise` incluye las siguientes métricas.

Métricas de señal

Métrica	Descripción
IllegalMessageFromEdge	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise no coincidía con el formato requerido. Unidades: recuento Dimensiones: VehicleName Estadísticas válidas: suma
MessageThrottled	Un mensaje enviado desde el vehículo a AWS IoT FleetWise se retrasó. Esto se debe a que ha superado los límites de servicio de esta cuenta en la región actual. Unidades: recuento Dimensiones: VehicleName Estadísticas válidas: suma
ModelingError	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise contiene señales que no se validan con el modelo del vehículo. Unidades: recuento Dimensiones: ModelManifestName

Métrica	Descripción
	Estadísticas válidas: suma
DecodingError	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise contiene señales que producen un error para decodific ar con el manifiesto del decodificador del vehículo. Unidades: recuento Dimensiones: DecoderName Estadísticas válidas: suma

Métricas de campaña

Métrica	Descripción
VehicleNotFound	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise en el que el vehículo es desconocido. Unidades: recuento Dimensiones: VehicleName Estadísticas válidas: suma
CampaignInvalid	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise en el que la campaña no es válida. Unidades: recuento Dimensiones: CampaignName Estadísticas válidas: suma

Métrica	Descripción
CampaignNotFound	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise en el que la campaña es desconocida. Unidades: recuento Dimensiones: CampaignName Estadísticas válidas: suma

Métricas de destino de los datos de campaña

Métrica	Descripción
TimestreamWriteError	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en la tabla de Amazon Timestream. Unidades: recuento Dimensiones: DatabaseName, TableName Estadísticas válidas: suma
S3WriteError	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en el bucket de Amazon Simple Storage Service (Amazon S3). Unidades: recuento Dimensiones: BucketName Estadísticas válidas: suma
S3ReadError	AWS IoT FleetWise no pudo leer una clave de objeto desde el vehículo en el bucket de Amazon Simple Storage Service (Amazon S3).

Métrica	Descripción
	Unidades: recuento Dimensiones: BucketName Estadísticas válidas: suma

Métricas clave de AWS KMS administradas por el cliente

Métrica	Descripción
KMSKeyAccessDenied	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en la tabla de Timestream o en el bucket de Amazon S3 debido a un error de acceso denegado a la clave de AWS KMS. Unidades: recuento Dimensiones: KMSKeyId Estadísticas válidas: suma

Supervisión de AWS IoT FleetWise con Registros de Amazon CloudWatch

Registros de Amazon CloudWatch supervisa los eventos que se producen en los recursos y lo avisa si hay algún problema. Si recibe una alerta, puede acceder a los archivos de registro para obtener información sobre el evento específico. Para obtener más información, consulte la Guía del usuario de Registros de Amazon CloudWatch.

Visualización de registros de AWS IoT FleetWise en la consola de CloudWatch

> **Important**
>
> Para poder visualizar el grupo de registros de AWS IoT FleetWise en la consola de CloudWatch, asegúrese de que se cumple lo siguiente:
>
> - Ha activado el registro en AWS IoT FleetWise. Para obtener más información acerca del registro, consulte Configuración del registro de AWS IoT FleetWise.
> - Ya hay entradas de registro escritas por operaciones de AWS IoT.

Para visualizar los registros de AWS IoT FleetWise en la consola de CloudWatch:

1. Abra la consola de CloudWatch.
2. En el panel de navegación, elija Registros, Grupos de registros.
3. Elija el grupo de registros.
4. Elija Buscar grupos de registros. Verá una lista completa de los eventos de registro generados para la cuenta.
5. Elija el icono de expansión para ver un flujo individual y buscar todos los registros que tengan un nivel de registro de `ERROR`.

   También puede escribir una consulta en el cuadro de búsqueda Filtrar eventos. Por ejemplo, puede probar lo siguiente:

   ```
   { $.logLevel = "ERROR" }
   ```

Para obtener más información sobre la creación de expresiones de filtro, consulte Sintaxis de patrones y filtros en la Guía del usuario de Registros de Amazon CloudWatch.

Example entrada de registro

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
```

```
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

Tipos de eventos de señal

Tipo de evento	Descripción
MODELING_ERROR	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise contiene señales que no se validan con el modelo del vehículo. Atributos: vehicleName, campaignName, signalCatalogName, signalId, signalValue, signalValueRangeMin, signalValueRangeMax, modelManifestName
ILLEGAL_MESSAGE_FROM_EDGE	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise no coincidía con el formato requerido. Atributos: vehicleName, campaignName, signalCatalogName
DECODING_ERROR	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise contiene señales que producen un error para decodific ar con el manifiesto del decodificador del vehículo. Atributos: campaignName, signalCatalogName, decoderManifestName, (opcional) signalName, (opcional) s3URI

Tipos de eventos de campaña

Tipo de evento	Descripción
VEHICLE_NOT_FOUND	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que se desconocía el vehículo. Atributos: vehicleName, campaignName
CAMPAIGN_NOT_FOUND	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que se desconocía la campaña. Atributos: vehicleName (opcional), campaignName
CAMPAIGN_INVALID	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que la campaña no era válida. Atributos: vehicleName (opcional), campaignName

Tipos de eventos de destino de datos de campaña

Tipo de evento	Descripción
TIMESTREAM_WRITE_ERROR	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en la tabla de Amazon Timestream. Atributos: vehicleName, campaignName, timestreamDatabaseName, timestreamTableName
S3_WRITE_ERROR	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en el bucket de Amazon Simple Storage Service (Amazon S3).

Tipo de evento	Descripción
	Atributos: campaignName, destinationName
S3_READ_ERROR	AWS IoT FleetWise no pudo leer una clave de objeto desde el vehículo en el bucket de Amazon Simple Storage Service (Amazon S3). Atributos: campaignName, destinationName

Tipos de eventos clave de AWS KMS administrados por el cliente

Tipo de evento	Descripción
KMS_KEY_ACCESS_DENIED	AWS IoT FleetWise no pudo escribir un mensaje desde el vehículo en la tabla de Timestream o en el bucket de Amazon S3 debido a un error de acceso denegado a la clave de AWS KMS.

Atributos

Todas las entradas de registro de CloudWatch incluyen estos atributos:

accountId

El ID de su Cuenta de AWS.

eventType

El tipo de evento para el que se generó el registro. El valor del tipo de evento depende del evento que generó la entrada de registro. Cada descripción de entrada de registro incluye el valor de `eventType` para esa entrada de registro.

logLevel

El nivel de registro que se está utilizando. Para obtener más información, consulte Niveles de registro en la Guía para desarrolladores de AWS IoT Core.

message

Contiene detalles específicos sobre el registro.

Marca de tiempo

La marca temporal en el formato de milisegundos de epoch en la que AWS IoT FleetWise IoT procesó el registro.

Atributos opcionales

Las entradas de Registros de CloudWatch incluyen opcionalmente estos atributos, en función del `eventType`:

decoderManifestName

El nombre del manifiesto del decodificador que contiene la señal.

destinationName

El nombre del destino de los datos del vehículo. Por ejemplo, el nombre del bucket de Amazon S3.

campaignName

Nombre de la campaña.

signalCatalogName

Nombre del catálogo de señales que contiene la señal.

signalId

ID de la señal de error.

signalIds

Lista de los ID de las señales de error.

signalName

El nombre de la señal.

signalTimestampEpochMs

Marca de tiempo de la señal de error.

signalValue

Valor de la señal de error.

signalValueRangeMax

Rango máximo de la señal de error.

signalValueRangeMin

Rango mínimo de la señal de error.

s3URI

El identificador único de Amazon S3 de un archivo de Amazon Ion de un mensaje de vehículo.

timestreamDatabaseName

Nombre de la base de datos de Timestream.

timestreamTableName

Nombre de la tabla de Timestream.

vehicleName

Nombre del vehículo.

Configuración del registro de AWS IoT FleetWise

Puede enviar los datos de registro de AWS IoT FleetWise a un grupo de registro de CloudWatch. Registros de CloudWatch ofrece visibilidad en caso de que AWS IoT FleetWise no procese los mensajes de los vehículos. Por ejemplo, esto puede ocurrir debido a una configuración defectuosa u a otros errores del cliente. Se le notificará cualquier error para que pueda identificar y mitigar los problemas.

Para poder enviar registros a CloudWatch, debe crear un grupo de registro de CloudWatch. Configure el grupo de registro con la misma cuenta y en la misma región que utilizó con AWS IoT FleetWise. Cuando habilite el registro en AWS IoT FleetWise, proporcione el nombre del grupo de registro. Una vez activado el registro, AWS IoT FleetWise envía los registros al grupo de registro de CloudWatch en flujos de registro.

Puede ver los datos de registro enviados desde AWS IoT FleetWise en la consola de CloudWatch. Para obtener más información sobre la configuración de un grupo de registro de CloudWatch y la visualización de los datos de registro, consulte Trabajo con grupos de registros.

Permisos para publicar registros en Registros de CloudWatch

Para configurar un registro para un grupo de registro de CloudWatch, deben configurarse los permisos según se describe en esta sección. Para obtener más información sobre los permisos de administración, consulte Administración de accesos para recursos de AWS en la Guía del usuario de IAM.

Estos permisos le permiten cambiar la configuración de registro, configurar la entrega de registros para CloudWatch y recuperar información sobre su grupo de registro.

```
{
   "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"
      }
   ]
```

```
}
```

Si se permiten acciones en todos los recursos de AWS, esto se indica en la política con una ajuste `"Resource"` con el valor `"*"`. Esto significa que las acciones están permitidas en todos los recursos de AWS compatibles con cada acción.

Configuración del registro en AWS IoT FleetWise (consola)

En esta sección se describe cómo utilizar la consola de AWS IoT FleetWise para configurar el registro.

Para usar la consola de AWS IoT FleetWise para configurar el registro:

1. Abra la consola de AWS IoT FleetWise.
2. En el panel izquierdo, seleccione Configuración.
3. En la sección Registro de la página Configuración, elija Editar.
4. En la sección Registro de CloudWatch, introduzca el grupo de registro.
5. Para guardar los cambios, elija Enviar.

Tras habilitar el registro, podrá ver los datos de registro en la consola de CloudWatch.

Configuración del registro predeterminado en AWS IoT FleetWise (CLI)

En esta sección se describe cómo configurar el registro para AWS IoT FleetWise mediante la CLI.

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí. Puede usar la operación de la API GetLoggingOptions para obtener la configuración actual y la operación de la API PutLoggingOptions para modificar la configuración.

Uso de la CLI para configurar el registro para AWS IoT FleetWise:

1. Utilice el comando get-logging-options para establecer las opciones de registro para la cuenta.

   ```
   aws iotfleetwise get-logging-options
   ```

2. Para habilitar el registro, utilice el comando put-logging-options.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
 logType=ERROR,logGroupName=MyLogGroup
```

donde:

logType

Tipo de registro para enviar datos a Registros de CloudWatch. Para deshabilitar el registro, cambie el valor a OFF.

logGroupName

Grupo de Registros de CloudWatch al que envía los datos la operación. Asegúrese de crear el nombre del grupo de registro antes de habilitar el registro para AWS IoT FleetWise.

Una vez habilitado el registro, consulte Buscar entradas de registro mediante AWS CLI.

Registro de llamadas a la API de AWS IoT FleetWise con AWS CloudTrail

AWS IoT FleetWise se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en AWS IoT FleetWise. CloudTrail captura las llamadas a la API de AWS IoT FleetWise como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de AWS IoT FleetWise y las llamadas desde el código a las operaciones de la API de AWS IoT FleetWise. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para AWS IoT FleetWise. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información que recopila CloudTrail, puede determinar la solicitud que se hizo a AWS IoT FleetWise, la dirección IP desde la que se hizo dicha solicitud y quién la hizo y cuándo, además de información adicional.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información sobre AWS IoT FleetWise en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce actividad en AWS IoT FleetWise, esta se registra en un evento de CloudTrail junto con otros eventos de servicio de

AWS en el Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte Ver eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de AWS IoT FleetWise, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recepción de archivos de registro de CloudTrail de varias regiones
- Recepción de archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones de AWS IoT FleetWise, que están documentadas en la referencia de la API de AWS IoT FleetWise. Por ejemplo, las llamadas a las acciones `CreateCampaign`, `AssociateVehicleFleet` y `GetModelManifest` generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el elemento userIdentity de CloudTrail.

Descripción de las entradas de los archivos de registro de AWS IoT FleetWise

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registro en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

En el ejemplo que sigue se muestra una entrada de registro de CloudTrail que ilustra la operación *AssociateVehicleFleet*.


```
{
      "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"
    }
```

Historial de documentos de la Guía para desarrolladores de AWS IoT FleetWise

En la siguiente tabla se describen las versiones de la documentación de AWS IoT FleetWise.

Cambio	Descripción	Fecha
Vista previa de los datos de sistemas de visión	Puede usar la vista previa de los datos de sistemas de visión de AWS IoT FleetWise para recopilar y organizar los datos de los sistemas de visión de los vehículos , incluidos los de cámaras, radares y lídares. Mantiene los datos de sistemas de visión estructurados y no estructur ados, los metadatos (ID de evento, campaña, vehículo) y el sensor estándar (datos de telemetría) sincronizados automáticamente en la nube.	26 de noviembre de 2023
Claves administradas por el cliente de AWS KMS	AWS IoT FleetWise ahora admite claves administr adas por el cliente de AWS KMS. Puede usar la clave de KMS para cifrar los datos del servidor relacionados con los recursos de AWS IoT FleetWise (catálogo de señales, modelo de vehículo, manifiesto del decodificador, vehículos y configuraciones de campañas de recopilación de	16 de octubre de 2023

	datos) almacenados en Nube de AWS.	
Almacenamiento de objetos en Amazon S3	AWS IoT FleetWise ahora admite el almacenamiento de datos con Amazon Simple Storage Service (Amazon S3). Puede almacenar los datos recopilados durante las campañas en Amazon S3, además de en Amazon Timestream.	1 de junio de 2023
Disponibilidad general	Esta es la versión pública de AWS IoT FleetWise.	27 de septiembre de 2022
Versión inicial	Esta es la versión prelimina r de la Guía para desarroll adores de AWS IoT FleetWise.	30 de noviembre de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.