Guía para desarrolladores

AWS IoT FleetWise

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

AWS IoT FleetWise: Guía para desarrolladores

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS IoT FleetWise? 1
Ventajas 2
Casos de uso 3
¿Eres nuevo en el AWS IoT FleetWise? 4
Acceso al AWS IoT FleetWise 4
Precios para AWS IoT FleetWise 4
Servicios relacionados 4
Conceptos clave 5
Conceptos clave 5
Características del AWS IoT FleetWise 10
AWS Regiones compatibles 11
Configura el AWS IoT FleetWise 13
Configura tu Cuenta de AWS 13
Inscríbase en una Cuenta de AWS 13
Creación de un usuario con acceso administrativo 14
Empezar a trabajar con la consola 15
Configure sus ajustes 16
Configuración (consola) 16
Configuración de los ajustes (AWS CLI) 17
Uso IPv6 con AWS IoT FleetWise 18
IPv6 requisitos previos para los puntos finales del plano de control 18
IPv6 soporte para puntos finales AWS PrivateLink 18
Probando IPv6 la compatibilidad de direcciones 19
Uso de IPv6 direcciones en las políticas de IAM 19
Usar puntos de enlace de doble pila 20
Introducción 22
Introducción 22
Requisitos previos 23
Paso 1: Configurar el software Edge Agent para AWS IoT FleetWise 24
Paso 2: Crear un modelo de vehículo 25
Paso 3: Crear un manifiesto del decodificador 27
Paso 4: Configurar un manifiesto del decodificador 28
Paso 5: Crear un vehículo 29
Paso 6: Crear una campaña 30

Paso 7: limpiar 32
Pasos a seguir a continuación 32
Ingesta de datos 33
Vehículos modelo 37
Catálogos de señales 40
Configuración de señales 43
Cree un catálogo de señales 49
Importación de un catálogo de señales 54
Actualice un catálogo de señales 64
Eliminar un catálogo de señales 67
Obtenga información del catálogo de señales 69
Modelos de vehículo 70
Creación de un modelo de vehículo 71
Actualiza un modelo de vehículo 78
Eliminación de un modelo de vehículo 80
Obtenga información sobre el modelo del vehículo 81
Manifiestos del decodificador 82
Configure las interfaces y las señales 85
Creación de un manifiesto del decodificador 88
Actualizar un manifiesto de decodificador 98
Eliminación de un manifiesto del decodificador 101
Obtenga información sobre el manifiesto del decodificador 103
Administre vehículos 105
Aprovisionamiento de vehículos 106
Autenticación de vehículos 107
Autorización de vehículos 109
Temas reservados 110
Creación de un vehículo 116
Creación de un vehículo (consola) 116
Creación de un vehículo (AWS CLI) 118
Crea varios vehículos 121
Actualiza un vehículo 123
Actualice varios vehículos 125
Eliminación de un vehículo 127
Eliminación de un vehículo (consola) 127
Eliminación de un vehículo (AWS CLI) 128

Obtenga información sobre el vehículo 129
Gestione flotas 131
Creación de una flota 132
Asocie un vehículo a una flota 133
Desvincular un vehículo de una flota 134
Actualice una flota 135
Eliminación de una flota 136
Verifica la eliminación de la flota 136
Obtenga información sobre la flota 137
Gestione los datos con campañas 140
Creación de una campaña 146
Creación de una campaña (consola) 147
Creación de una campaña (AWS CLI) 156
Expresiones lógicas para FleetWise campañas AWS de IoT 161
Actualizar una campaña 163
Eliminar una campaña 164
Eliminación de una campaña (consola) 164
Eliminación de una campaña (AWS CLI) 164
Verifica la eliminación de la campaña 165
Obtenga información sobre la campaña 165
Almacena y reenvía 166
Cree particiones de datos 167
Sube los datos de la campaña 170
Sube datos mediante AWS IoT Jobs 171
Recopile datos de códigos de diagnóstico de problemas 173
Palabras clave de códigos de diagnóstico de problemas 175
Cree una campaña de recopilación de datos para los códigos de diagnóstico de problemas 177
Casos de uso de códigos de diagnóstico de problemas 180
Visualice los datos del vehículo 183
Procesar los datos del vehículo enviados a un tema de MQTT 184
Procesa los datos del vehículo en Timestream 185
Visualice los datos del vehículo almacenados en Timestream 186
Procese los datos del vehículo en Amazon S3 186
Formato de objeto Amazon S3 187
Analice los datos del vehículo almacenados en Amazon S3 189

Comandos 191
Conceptos de comandos 192
Comandos y conceptos clave 193
Estado de ejecución del comando 196
Vehículos y comandos 202
Información general sobre el flujo de trabajo 202
Flujo de trabajo de 204
flujo de trabajo de comandos 206
(Opcional) Comandos y notificaciones 208
Creación y administración de comandos 209
Cree un recurso de comandos 210
Recupera información sobre un comando 212
Enumere los comandos de su cuenta 213
Actualiza o desactiva un recurso de comando 214
Eliminar un recurso de comandos 216
Iniciar y supervisar las ejecuciones de comandos 217
Actualiza el resultado de la ejecución del comando 221
Obtenga la ejecución de comandos 223
Enumere las ejecuciones de comandos en su cuenta 224
Eliminar la ejecución de un comando 226
Ejemplo: uso de comandos 227
Ejemplo de descripción general del modo de dirección del vehículo 227
Requisitos previos 228
Política de IAM para el uso de comandos remotos 228
Ejecute AWS IoT comandos (AWS CLI) 231
Limpieza 236
Escenarios de uso de comandos 238
Crear un comando sin parámetros 239
Crear un comando con valores predeterminados para los parámetros 240
Crear un comando con valores de parámetros 241
Uso de comandos con plantillas de estado 242
Último estado conocido 245
Cree una plantilla de estado 246
Asociar una plantilla de FleetWise estado de AWS IoT a un vehículo 248
Actualizar una plantilla de estado 249
Eliminar una plantilla de estado 250

Obtenga información sobre la plantilla de estado 251
Operaciones con plantillas de estados 252
Active y desactive la recopilación de datos de estado 253
Obtenga una instantánea del estado del vehículo 259
Procese los datos del último estado conocido del vehículo mediante la mensajería MQTT .. 261
Configure la recopilación de datos independiente de la red 265
Introducción 265
Configuración del entorno 265
Modelos de datos 266
Actualizaciones del catálogo de señales 266
Modelo y decodificador del vehículo 268
Enviar comando 271
AWS CLI y SDKs 272
Solución de problemas 273
Problemas con el manifiesto del decodificador 273
Problemas con los agentes perimetrales 277
Problema: el software Edge Agent no se inicia. 278
Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [No se pudo iniciar la biblioteca de persistencia] 279
Problema: el software Edge Agent no recopila los códigos de diagnóstico integrados (OBD) II PIDs ni los códigos de diagnóstico de problemas (). DTCs 279
Problema: el FleetWise software Edge Agent para AWS IoT no recopila datos de la red o no puede aplicar reglas de inspección de datos. 280
Problema: [ERROR] [AwsIotConnectivityModule: :connect]: [La conexión falló por error] o [WARN] [AwsIotChannel: :send]: [No hay una conexión MQTT activa.] 281
Almacena y reenvía las incidencias 281
Problema: Recibir un anuncio `AccessDeniedException` con todos los permisos de IAM necesarios 281
Problema: Los datos subidos a AWS IoT Jobs ignoran las `endTime` 281
Problema: La carga de datos a AWS IoT Jobs tiene un `REJECTED` estado de ejecución. 282
Seguridad 283
Protección de los datos 284
El cifrado en reposo en el AWS IoT FleetWise 285
Cifrado en tránsito 285
Cifrado de datos en AWS IoT FleetWise 286
Control del acceso 298

Otorgue AWS IoT FleetWise permiso para enviar y recibir datos sobre un tema de MQTT .. 299
Conceder AWS IoT FleetWise acceso a un destino de Amazon S3 302
Conceder AWS IoT FleetWise acceso a un destino de Amazon Timestream 305
Conceda AWS IoT Device Management permiso para generar la carga útil de los comandos con AWS IoT FleetWise 308
Identity and Access Management 312
Público 313
Autenticación con identidades 313
Administración de acceso mediante políticas 317
Cómo FleetWise funciona el AWS IoT con IAM 320
Ejemplos de políticas basadas en identidades 329
Solución de problemas 333
Referencia de permisos de la API 335
Validación de conformidad 344
Resiliencia 346
Seguridad de la infraestructura 346
Conexión al AWS IoT a FleetWise través de un punto final de VPC de interfaz 347
Configuración y análisis de vulnerabilidades 351
Prácticas recomendadas de seguridad 351
Conceda los mínimos permisos posibles 351
No registre información confidencial 352
Úselo para ver el AWS CloudTrail historial de llamadas de la API 352
Mantenga sincronizado el reloj del dispositivo 352
Monitorización de AWS IoT FleetWise 353
Monitorización con CloudWatch 353
Supervise con CloudWatch registros 358
Ver FleetWise los registros de AWS IoT en la CloudWatch consola 358
Configuración del registro 365
CloudTrail registros 368
AWS FleetWise Información sobre IoT en CloudTrail 368
Comprenda las entradas del archivo de registro de 369
Historial de documentos 371
..... ccclxxiv

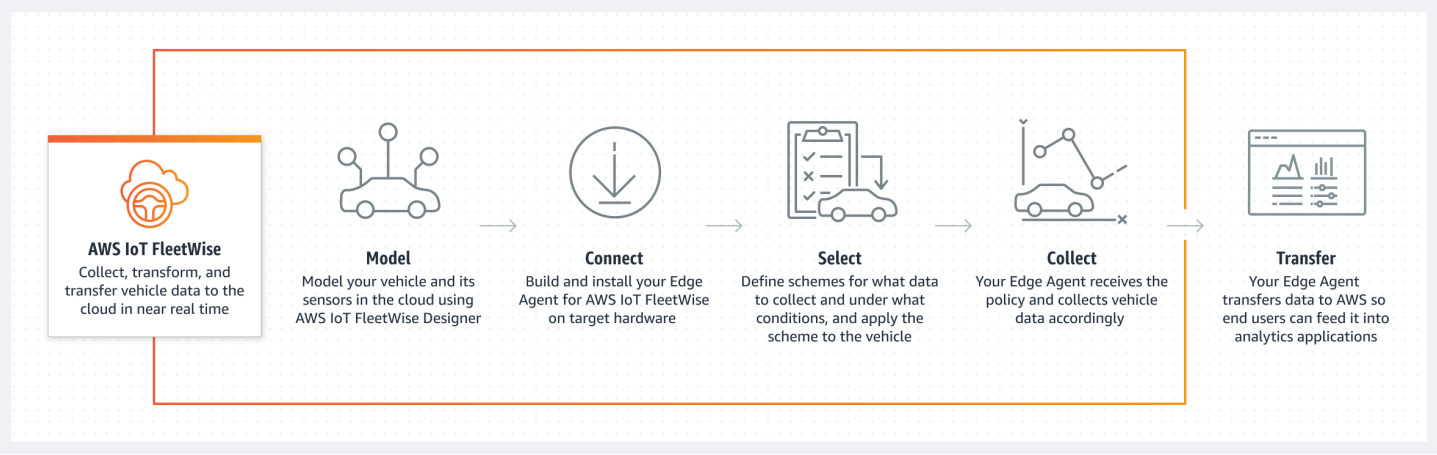
¿Qué es el AWS IoT FleetWise?

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

AWS FleetWise El IoT es un servicio gestionado 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 el AWS IoT FleetWise, puede recopilar y organizar datos de vehículos que utilizan diferentes protocolos y formatos de datos. AWS FleetWise El IoT ayuda a transformar los mensajes de bajo nivel en valores legibles para las personas 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).

El siguiente diagrama muestra la arquitectura básica de AWS IoT FleetWise.



Temas

- Ventajas
- Casos de uso
- ¿Eres nuevo en el AWS IoT FleetWise?
- Acceso al AWS IoT FleetWise
- Precios para AWS IoT FleetWise
- Servicios relacionados
- Conceptos y características clave del AWS IoT FleetWise
- AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise

Ventajas

Los principales beneficios del 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 visión 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 FleetWise El IoT mantiene los datos del sistema 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.

Almacene los datos en Edge y envíelos en condiciones óptimas

> Reduzca los costos de transmisión almacenando temporalmente los datos en los vehículos. Puede reenviar los datos seleccionados a la nube en condiciones óptimas y específicas, como cuando los vehículos se conectan a una red Wi-Fi.

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

Casos de uso

Entre los escenarios en los que puede utilizar el AWS IoT se FleetWise incluyen 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.

Cree y administre comandos

Utilice comandos para ejecutar comandos en un vehículo desde la nube. Puede enviar comandos a un vehículo de forma remota y, en unos segundos, el vehículo ejecutará el comando.
Por ejemplo, puedes configurar comandos para cerrar la puerta de un vehículo o ajustar la temperatura.

Cree y gestione plantillas de estado

Las plantillas de estado proporcionan un mecanismo para que los propietarios de vehículos rastreen el estado de su vehículo. El AWS IoT FleetWise Edge Agent que se ejecuta en el vehículo recopila y envía actualizaciones de señales a la nube.

¿Eres nuevo en el AWS IoT FleetWise?

Si eres nuevo en el AWS IoT FleetWise, te recomendamos que comiences leyendo las siguientes secciones:

- Conceptos y características clave del AWS IoT FleetWise
- Configura el AWS IoT FleetWise
- Tutorial: Cómo empezar con el AWS IoT FleetWise
- Transfiera FleetWise datos de AWS IoT a la nube

Acceso al AWS IoT FleetWise

Puede usar la FleetWise consola o la API de AWS IoT para acceder a AWS IoT FleetWise.

Precios para AWS IoT FleetWise

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

Servicios relacionados

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

- AWS IoT Core— Registre y controle AWS IoT los dispositivos que cargan los datos del vehículo a la AWS IoT FleetWise y envíen comandos a un vehículo de forma remota. 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.

Conceptos y características clave del AWS IoT FleetWise

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En las siguientes secciones se proporciona una descripción general de los componentes del FleetWise servicio de AWS IoT y de cómo interactúan.

Después de leer esta introducción, consulta la Configura el AWS IoT FleetWise sección para aprender a configurar el AWS IoT FleetWise.

Temas

- Conceptos clave
- Características del AWS IoT FleetWise

Conceptos clave

AWS FleetWise El IoT proporciona un marco de modelado de vehículos para modelar su vehículo y sus sensores y actuadores en la nube. Para permitir la comunicación segura entre su vehículo y la nube, el 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.

Los siguientes son los conceptos básicos del AWS IoT FleetWise.

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 Gestione los catálogos FleetWise de señales de AWS IoT.

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.

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 Gestione los catálogos FleetWise de señales de AWS IoT.

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 Gestione los modelos de FleetWise vehículos de AWS IoT.

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 para los humanos. Cada manifiesto del

decodificador está asociado a un modelo de vehículo. Para obtener más información, consulte Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

Interfaz de red

Contiene información sobre el protocolo que utiliza la red del 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 (ECUs). ECUs puede ser la unidad de control del motor, los airbags o el sistema de audio.

Diagnóstico a bordo (OBD) II

Un protocolo desarrollado aún más que define cómo se comunican los datos de autodiagnóstico entre sí. ECUs Proporciona una serie de códigos de diagnóstico de problemas estándar (DTCs) que ayudan a identificar qué es lo que está mal en tu 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.

Interfaces personalizadas

También puede utilizar su propia interfaz para decodificar las señales en el Edge. Esto puede ahorrarle tiempo, ya que no necesita crear reglas de decodificación en la nube.

Decodificador de señales

Proporciona información de decodificación detallada para una señal específica. Todas las señales especificadas en el modelo del vehículo deben estar emparejadas con un decodificador de señales. 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 decodificadores de señal OBD.

El manifiesto del decodificador debe contener decodificadores de señales de mensajes si también contiene interfaces de middleware para vehículos. O bien, si el manifiesto del decodificador contiene interfaces de decodificación personalizadas, también debe contener señales de decodificación personalizadas.

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.

Partición de datos

Configure los datos particionados en una campaña para almacenar temporalmente los datos de señal. Tú configuras cuándo y cómo reenviar los 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, el AWS IoT FleetWise admite el esquema de recolección basado en condiciones y el esquema de recolección 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.

Comandos

Los comandos ejecutan comandos en un vehículo desde la nube. Puedes enviar comandos a un vehículo de forma remota y, en unos segundos, el vehículo ejecutará el comando. Por ejemplo, puedes configurar comandos para cerrar la puerta de un vehículo o ajustar la temperatura.

El comando es un recurso gestionado por AWS IoT Device Management. Contiene configuraciones reutilizables que se aplican al enviar la ejecución de un comando al vehículo. Para obtener más información, consulte AWS IoT los comandos de la Guía para AWS IoT Core desarrolladores.

Plantillas de estados

Las plantillas estatales proporcionan un mecanismo para que los propietarios de vehículos rastreen el estado de su vehículo. El agente de software Edge Agent que se ejecuta en el vehículo recopila las actualizaciones de señales y las envía a la nube. Cada plantilla de estado contiene una lista de señales a partir de las cuales se recopilan los datos.

Características del AWS IoT FleetWise

Las siguientes son las principales características del AWS IoT FleetWise.

Modelado de vehículos

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

Edge Agent para FleetWise software AWS de IoT

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.

AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise

Para obtener una lista de AWS las regiones que admiten AWS IoT FleetWise, consulte FleetWise Puntos finales y cuotas de AWS IoT. AWS FleetWise Las funciones de IoT difieren en cuanto a su compatibilidad regional.

> **Note**
>
> El acceso a la región de Asia Pacífico (Bombay) y a algunas FleetWise funciones de AWS IoT están actualmente bloqueados. Para solicitar acceso a esta AWS región y a todas las funciones cerradas, póngase en contacto con su administrador de cuentas o con el AWS Support Center.

En la siguiente tabla se muestra el soporte de funciones por región:

Funciones/regiones	Este de EE. UU. (Norte de Virginia)	Europa (Fráncfort)	Asia-Pacífico (Bombay) NOTA: Solo acceso cerrado
Catálogos de señales	Sí	Sí	Restringido
Modelos de vehículos	Sí	Sí	Restringido
Manifiestos decodificadores	Sí	Sí	Restringido
Vehículos	Sí	Sí	Restringido
Flotas	Sí	Sí	Restringido
Campaigns (Campañas)	Sí	Sí	Restringido
Datos del sistema de visión (en versión preliminar)	Sí	Sí	Restringido

Funciones/regiones	Este de EE. UU. (Norte de Virginia)	Europa (Fráncfort)	Asia-Pacífico (Bombay) NOTA: Solo acceso cerrado
El tema MQTT como destino de los datos de la campaña	Restringido	Restringido	Restringido
Almacena y reenvía	Restringido	Restringido	Restringido
Comandos	Restringido	Restringido	Restringido
Último estado conocido	Restringido	Restringido	Restringido
Recopilación de datos independiente de la red mediante una interfaz de decodificación personalizada	Restringido	Restringido	Restringido
Obtención del código de diagnóstico de problemas (DTC) *	Restringido	Restringido	Restringido

*La obtención de DTC ofrece una gama de funciones que van más allá de la recuperación básica de datos de DTC. Esta funcionalidad incluye funciones personalizadas que te permiten definir funciones en la periferia e invocarlas por su nombre en las expresiones de campaña basadas en condiciones. Además, admite la recopilación de cadenas ilimitadas, lo que proporciona un manejo flexible de los tipos de datos de cadenas. El Edge Agent puede buscar datos de forma periódica o en función de condiciones específicas, lo que mejora su adaptabilidad y eficiencia en los procesos de recopilación de datos. Para obtener más información, consulte la guía de funciones personalizadas y la implementación de referencia sobre la recopilación de datos de DTC en la Guía para desarrolladores de Edge Agent.

Configura el 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
- Empezar a trabajar con la consola
- Configura tus FleetWise ajustes AWS de IoT
- Realización de solicitudes a AWS IoT FleetWise mediante IPv6

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. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro implica recibir una llamada telefónica o un mensaje de texto e introducir 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. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a 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 Iniciar sesión como usuario raíz 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.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use 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. Abre la FleetWise consola AWS de IoT.
2. En Comenzar con AWS IoT FleetWise, selecciona Comenzar.

Para obtener más información acerca de la creación de un modelo de vehículo, consulte Cree un modelo AWS de FleetWise vehículo de IoT.

Configura tus FleetWise ajustes AWS de IoT

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, consulteSupervisa el 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, consulteConfigurar el FleetWise registro AWS de IoT.

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 en AWS IoT FleetWise.

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. Abre la FleetWise consola AWS de 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 Gestión de claves en AWS IoT FleetWise.

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.

Realización de solicitudes a AWS IoT FleetWise mediante IPv6

Puede comunicarse con AWS IoT FleetWise a través del Protocolo de Internet versión 6 (IPv6) y IPv4 administrar sus recursos. Los puntos finales de doble pila admiten solicitudes de AWS IoT a FleetWise APIs través IPv6 de y. IPv4 No hay cargos adicionales por la comunicación a través de. IPv6

El IPv6 protocolo es el estándar IP de próxima generación con funciones de seguridad adicionales. Ofrece un espacio de direcciones de 128 bits de longitud, mientras que IPv4 tiene una dirección de 32 bits. IPv4 puede generar 4,29 x 10^9 direcciones, mientras que IPv6 puede tener 3,4 x 10^38 direcciones.

IPv6 requisitos previos para los puntos finales del plano de control

IPv6 el soporte de protocolos se habilita automáticamente para los puntos finales del plano de control. Al utilizar los puntos finales para los clientes del plano de control, debe proporcionar la extensión de indicación del nombre del servidor (SNI). Los clientes pueden usar la extensión SNI para indicar el nombre del servidor con el que se contactan y si utilizan los puntos finales normales o los puntos finales de doble pila. Consulte Usar puntos de enlace de doble pila.

IPv6 soporte para puntos finales AWS PrivateLink

AWS El IoT FleetWise admite IPv6 la comunicación para interconectar los puntos finales de la VPC mediante. AWS PrivateLink

Probando IPv6 la compatibilidad de direcciones

Si utilizas use Linux/Unix o Mac OS X, puedes probar si puedes acceder a un punto final de doble pila IPv6 mediante el comando curl, como se muestra en el siguiente ejemplo:

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

Usted obtiene información similar a la del siguiente ejemplo. Si estás conectado a través de una red IPv6, la dirección IP conectada será una IPv6 dirección.

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

Si utilizas Microsoft Windows 7 o Windows 10, puedes probar si puedes acceder a un punto final de doble pila a través IPv6 o IPv4 mediante el comando ping, como se muestra en el siguiente ejemplo.

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

Uso de IPv6 direcciones en las políticas de IAM

Antes de utilizarlos IPv6 para sus recursos, debe asegurarse de que todas las políticas de IAM que se utilicen para el filtrado de direcciones IP incluyan IPv6 rangos de direcciones. Para obtener más información acerca de cómo administrar los permisos de acceso con IAM, consulte Identity and Access Management para AWS IoT FleetWise.

Las políticas de IAM que filtran direcciones IP utilizan operadores de condición de dirección IP. La siguiente política identifica el `54.240.143.*` rango de IPv4 direcciones permitidas mediante operadores de condiciones de direcciones IP. Como todas IPv6 las direcciones están fuera del rango permitido, esta política impide la comunicación mediante IPv6 direcciones.

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

```
      "Effect": "Allow",
      "Principal": "*",
      "Action": "iotfleetwise:*",
      "Resource": "arn:aws:iotfleetwise:*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Para incluir IPv6 direcciones, puede modificar el elemento Condition de la política para permitir intervalos de direcciones IPv4 (54.240.143.0/24) y IPv6 (2001:: 1234:5678DB8: :/64), como se muestra en el siguiente ejemplo.

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

Usar puntos de enlace de doble pila

AWS Los puntos finales FleetWise de doble pila de IoT admiten solicitudes de AWS IoT a FleetWise APIs través IPv6 de y. IPv4 Cuando realizas una solicitud a un punto final de doble pila, esta se resuelve automáticamente en una IPv4 o una dirección. IPv6 En el modo de doble pila, se aceptan tanto IPv4 las conexiones como las de IPv6 cliente.

Si utilizas la API REST, puedes acceder directamente a un FleetWise punto final de AWS IoT mediante el nombre del punto final (URI). AWS FleetWise El IoT solo admite nombres de terminales regionales de doble pila, lo que significa que debe Región de AWS especificarlos como parte del nombre.

La siguiente tabla muestra el formato de los puntos finales del plano de control para AWS IoT FleetWise cuando se utilizan IPv4 y los modos de doble pila. Para obtener más información sobre estos puntos de conexión, consulte Puntos de conexión de AWS IoT FleetWise .

Punto de conexión	IPv4 dirección	Modo de pila doble
Plano de control	iotfleetwise. *<region>*.amazonaws.com	IoT en cuanto a flota. *<region>*.api.aws

Al usar AWS CLI y AWS SDKs, puede usar una variable de `AWS_USE_DUALSTACK_ENDPOINT` entorno o el `use_dualstack_endpoint` parámetro, que es una configuración de un archivo de configuración compartido, para cambiar a un punto final de doble pila. También puede especificar el punto final de doble pila directamente como una anulación del FleetWise punto final de AWS IoT en el archivo de configuración. Para obtener más información, consulte Dual-stack and FIPS endpoints.

Cuando usas el AWS CLI, puedes establecer el valor de configuración `use_dualstack_endpoint` como `true` en un perfil de tu archivo AWS Config. Esto dirigirá todas las FleetWise solicitudes de AWS IoT realizadas por los comandos al punto final de doble pila de la región especificada. Puede especificar la región en el archivo de configuración o en un comando utilizando la opción `--region`.

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

En lugar de utilizar los puntos finales de doble pila para todos los comandos, utilice estos puntos finales para comandos específicos:

- Puede usar el punto final de doble pila para comandos específicos configurando el `--endpoint-url` parámetro para esos comandos. Por ejemplo, en el siguiente comando, puede reemplazar el comando *<endpoint-url>* to`iotfleetwise.`*<region>*`.api.aws`.

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

- Puedes configurar perfiles independientes en tu archivo AWS Config. Por ejemplo, crea un perfil que `use_dualstack_endpoint` se establezca en verdadero y otro que no se establezca`use_dualstack_endpoint`. Al ejecutar un comando, especifique qué perfil quiere usar, en función de si quiere usar el punto de conexión de doble pila o no.

Tutorial: Cómo empezar con el AWS IoT FleetWise

Con el AWS IoT FleetWise, puede recopilar, transformar y transferir los datos de su vehículo. Usa el tutorial de esta sección para empezar a usar el AWS IoT FleetWise.

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

- Transfiera FleetWise datos de AWS IoT a la nube
- Modelos de FleetWise vehículos AWS IoT
- Administre FleetWise vehículos de AWS IoT
- Gestione flotas en IoT AWS FleetWise
- Recopile FleetWise datos de AWS IoT con campañas

Introducción

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, consulteConceptos y características clave del 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.

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). Puede utilizar el selector de regiones de la AWS Management Console para cambiar a una de estas regiones. Para obtener más información, consulte FleetWise Puntos finales y cuotas de AWS IoT.
- 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.
- La demostración del software Edge Agent. (Las instrucciones para configurar la demostración se encuentran en el siguiente paso).
  - 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. Le guía a través de la revisión de la implementación de referencia de Edge Agent, el desarrollo de su Edge Agent y, a continuación, la implementación del software Edge Agent en un Amazon EC2 Graviton 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 descargar la demostración, navegue hasta la FleetWise consola de AWS IoT. En la página de inicio del servicio, en la FleetWise sección Comenzar con AWS IoT, elija Explore Edge Agent.

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.

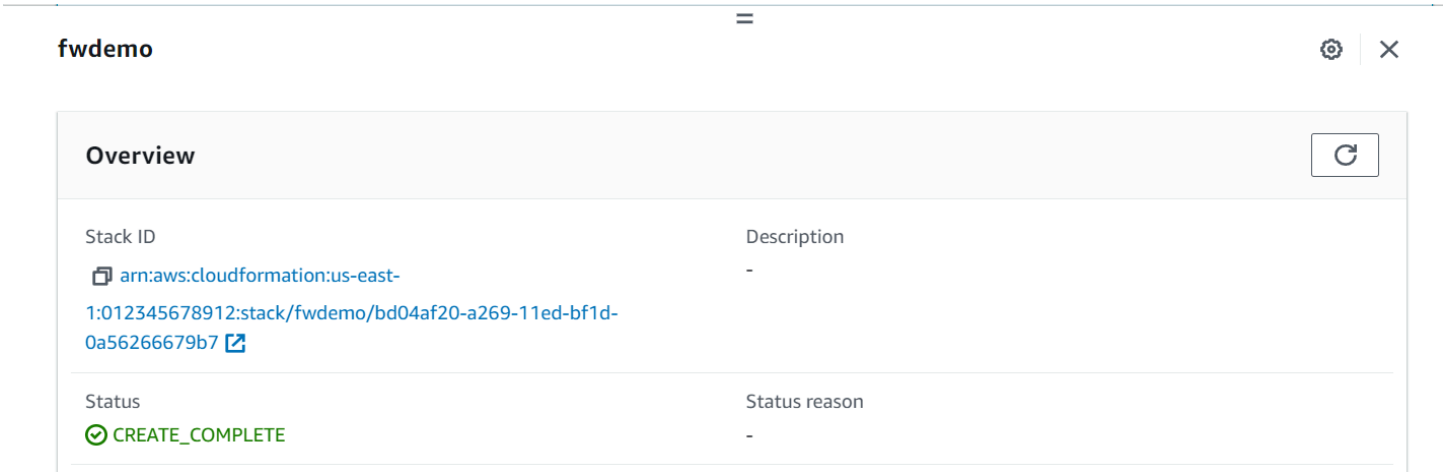
Su software Edge Agent para AWS IoT FleetWise facilita la comunicación entre los vehículos y la nube. Recibe instrucciones de los esquemas de recopilación de datos sobre cómo deben recopilarse datos de vehículos conectados a la nube.

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. TCoreRegión Io - Puede especificar la región en la AWS IoT que se crea la cosa actualizando el parámetro TCore Región Io. 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.

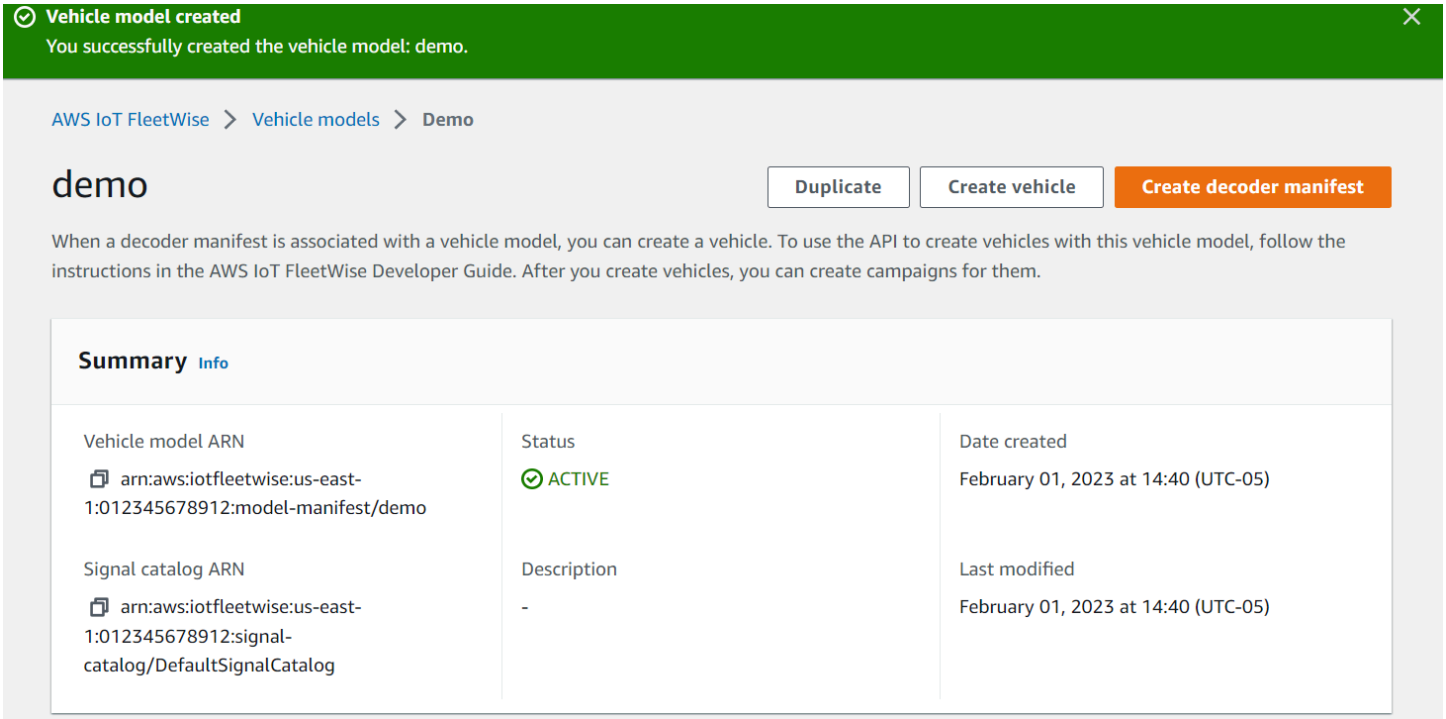
2. En el panel de navegación, elija Modelos de vehículo.
3. En la página Modelos de vehículo, elija Crear modelo de vehículo.
4. En la sección Información general, introduzca el nombre del modelo del vehículo, como Vehículo1, y una descripción opcional. A continuación, elija Siguiente.
5. Elija una o 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 Next (Siguiente).
6. Elija los archivos .dbc y cárguelos desde su dispositivo local. Elija Next (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).

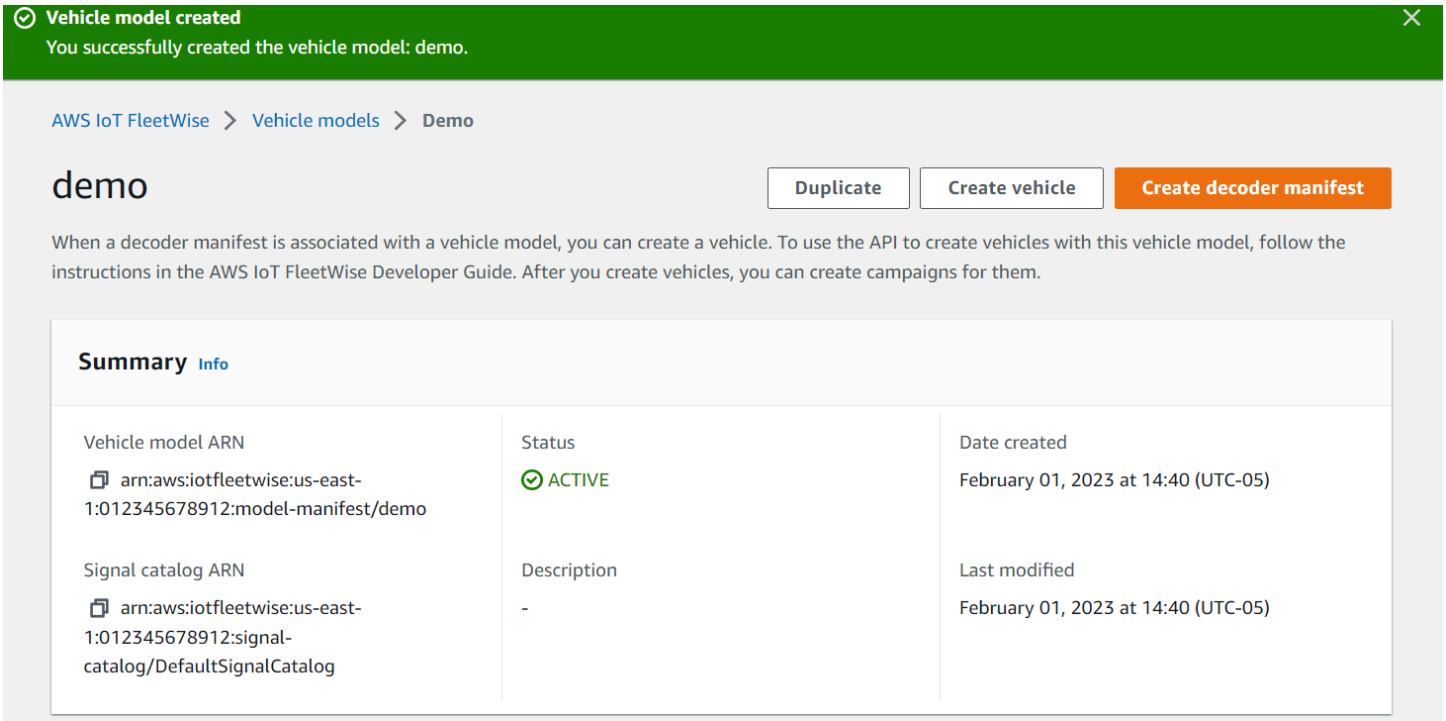
1. Para ayudarlo a identificar el manifiesto del decodificador, introduzca un nombre y una descripción opcional para el mismo. A continuación, elija Siguiente.
2. Para agregar una o más interfaces de red, elija el tipo CAN_INTERFACE u OBD_INTERFACE.
   - Interfaz de diagnóstico integrada (OBD) - Elija este tipo de interfaz si desea un protocolo que defina cómo se comunican los datos de autodiagnóstico entre las unidades de control electrónico (). ECUs Este protocolo proporciona una serie de códigos de diagnóstico de problemas estándar (DTCs) que pueden ayudarte a solucionar problemas con tu vehículo.
   - Interfaz de red de área del controlador (bus CAN) - Elija este tipo de interfaz si desea un protocolo que defina cómo se comunican los datos entre ECUs sí. ECUs pueden ser las unidades de control del motor, los airbags o el sistema 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 Next (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 Next (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 Next (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.

Pasos a seguir a continuación

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

Transfiera FleetWise datos de AWS IoT a la nube

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

El FleetWise software Edge Agent para AWS IoT, cuando se instala y ejecuta en vehículos, está diseñado para facilitar la comunicación segura entre sus vehículos y la nube.

> **Note**
>
> - AWS FleetWise El IoT no está diseñado para usarse en entornos peligrosos o sistemas críticos, ni en asociación con ellos, que puedan provocar lesiones corporales graves o la muerte o causar daños ambientales o a la propiedad. Los datos del vehículo recopilados mediante el uso de la AWS IoT FleetWise tienen únicamente fines informativos y no puede utilizar la AWS IoT FleetWise para controlar u operar las funciones del vehículo.
> - Los datos del vehículo recopilados mediante el uso del AWS IoT FleetWise deben evaluarse para determinar su 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 las normas de seguridad de los vehículos aplicables (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 su FleetWise software Edge Agent para AWS IoT en su vehículo. Para obtener más información sobre cómo trabajar con el software Edge Agent, haga lo siguiente para descargar la Guía para desarrolladores de FleetWise software Edge Agent para AWS IoT.
    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.

2. Cree o importe un catálogo de señales con las señales que utilizará para crear un modelo de vehículo. Para obtener más información, consulte Cree un catálogo AWS de FleetWise señales de IoT 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 Cree un modelo AWS de FleetWise vehículo de IoT.
   > - AWS FleetWise Actualmente, IoT admite un catálogo de señales para cada AWS cuenta por Región de AWS.

3. Utilice las señales del catálogo de señales para crear un modelo de vehículo. Para obtener más información, consulte Cree un modelo AWS de FleetWise vehículo de IoT.

   > **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 Cree un modelo AWS de FleetWise vehículo de IoT.
   > - 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 Actualizar un modelo AWS de FleetWise vehículo de IoT.
   > - 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.

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 Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

> **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 Actualizar un manifiesto de FleetWise decodificador de AWS IoT.
> - Si utilizas la FleetWise consola de AWS IoT para crear un manifiesto de decodificador, AWS IoT lo activa FleetWise 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 utilizarla AWS IoT Core para aprovisionar su vehículo antes de poder transferir datos a la nube. Para obtener más información, consulte Administre FleetWise vehículos de AWS IoT.
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 Gestione flotas en IoT AWS FleetWise.
7. (Opcional) Crea 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 Recopile FleetWise datos de AWS IoT con campañas. Puedes crear campañas, plantillas de estados (a continuación) o ambas opciones para recopilar datos.

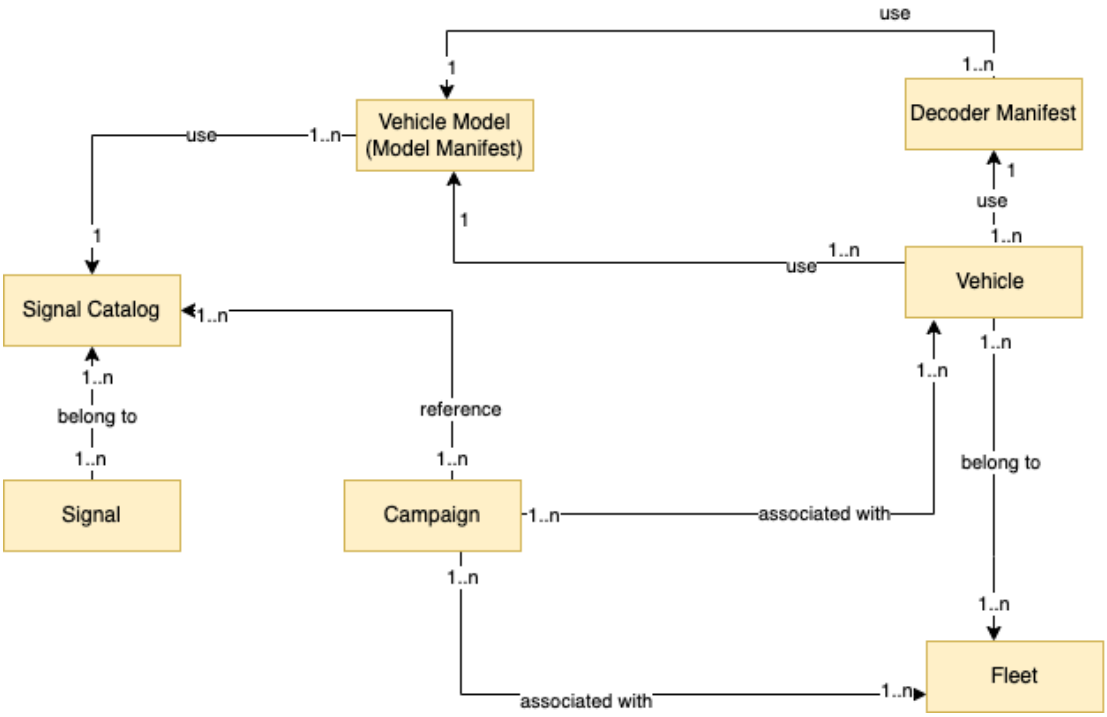
   > **Note**
   >
   > Debe usar la operación de la `UpdateCampaign` API para aprobar la campaña antes de que el AWS IoT FleetWise pueda implementarla en el vehículo o la flota. Para obtener más información, consulte Actualizar una FleetWise campaña AWS de IoT.

8. (Opcional) Cree plantillas de estado. Las plantillas de estado se despliegan en un vehículo. Las plantillas de estados proporcionan un mecanismo para que los propietarios de vehículos rastreen el estado de su vehículo. Para obtener más información, consulte Supervisa el último estado conocido de tus vehículos.

El software Edge Agent transfiere los datos del vehículo a AWS IoT Core través del tema MQTT que tú elijas. Para enviar los datos al AWS IoT FleetWise para las campañas, utiliza el tema reservado`$aws/iotfleetwise/vehicles/vehicleName/signals`. Para el último estado conocido, el agente perimetral usa el tema reservado`$aws/iotfleetwise/vehicles/vehicleName/last_known_states/data`. Para obtener más información sobre cómo se procesan los datos ingeridos, consulteVisualice los datos de FleetWise vehículos de AWS IoT.

Modelos de FleetWise vehículos AWS IoT

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 Gestione los catálogos FleetWise de señales de AWS IoT.

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 Gestione los catálogos FleetWise de señales de AWS IoT.

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 Gestione los modelos de FleetWise vehículos de AWS IoT.

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 Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

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 obtener más información, consulte Cree un catálogo AWS de FleetWise señales de IoT 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 Cree un modelo AWS de FleetWise vehículo de IoT.
   > - 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 Cree un modelo AWS de FleetWise vehículo de IoT.

> **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 Cree un modelo AWS de FleetWise vehículo de IoT.
> - 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 Actualizar un modelo AWS de FleetWise vehículo de IoT.
> - 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 Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

> **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 Actualizar un manifiesto de FleetWise decodificador de AWS IoT.
> - 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 decodificadores de señales. Para obtener un archivo .dbc de ejemplo, haga lo siguiente.

Para obtener un archivo .dbc

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

Temas

- Gestione los catálogos FleetWise de señales de AWS IoT
- Gestione los modelos de FleetWise vehículos de AWS IoT
- Gestione los AWS manifiestos de FleetWise decodificadores de IoT

Gestione los catálogos FleetWise de señales de AWS IoT

> **Note**
>
> Puede descargar un script de demostración para convertir los mensajes ROS 2 en archivos VSS .json 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 admite 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 Cree un modelo AWS de FleetWise vehículo de IoT.
> - 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 Cree un modelo AWS de FleetWise vehículo de IoT.
> - 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

- Configurar FleetWise señales AWS de IoT
- Cree un catálogo AWS de FleetWise señales de IoT
- Importar un catálogo AWS de FleetWise señales de IoT
- Actualizar un catálogo AWS de FleetWise señales de IoT
- Eliminar un catálogo AWS de FleetWise señales de IoT
- Obtenga información sobre el catálogo FleetWise de señales de AWS IoT

Configurar FleetWise señales AWS de IoT

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,,,,, BOOLEAN UINT8, FLOAT INT16 UINT16, DOUBLE INT32 UINT32, STRING INT64 UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, _ARRAY, UINT8 INT16 BOOLEAN_ARRAY, UINT16 FLOAT_ARRAY, INT32 UINT32 DOUBLE_ARRAY, INT64 STRING_ARRAY, UINT64 UNIX_TIMESTAMP_ARRAY, UNKNOWN o personalizado estructura 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, BOOLEAN UINT32 INT64 UINT64, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, _ARRAY, INT8 _ARRAY, UINT8 _ARRAY, INT16 _ARRAY, UINT16 INT32 _ARRAY, UINT32 _ARRAY, INT64 _ARRAY, UINT64 BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY fullyQualifiedName, UNKNOWN o personalizado estructura definida en la rama de tipos de datos.
- `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 (por ejemplo UINT8) u otra estructura (como una marca de tiempo). 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 demuestra 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 VSS .json 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, BOOLEAN UINT16 INT32 UINT32, FLOAT INT64 UINT64, DOUBLE, STRING, UNIX_TIMESTAMP, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 _ARRAY, INT16 _ARRAY, UINT16 _ARRAY, INT32 _ARRAY, UINT32 INT64 BOOLEAN_ARRAY, UINT64 FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, 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).

Cree un catálogo AWS de FleetWise señales de IoT

Puede utilizar la operación 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`*Reemplácelo por el nombre del catálogo de señales que está creando.
- (Opcional) *`description`* Sustitúyalo por una descripción que le ayude a identificar el catálogo de señales.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configurar FleetWise señales AWS de IoT.

```
{
    "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 VSS .json 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.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `CreateSignalCatalog` API.

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

Importar un catálogo AWS de FleetWise señales de IoT

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 FleetWiseconsola 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 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 usar la operación de la 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) Sustituya la descripción por una *description* para ayudarle a identificar el catálogo de señales.
- *signal-catalog-configuration-vss*Sustitúyala 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 Configurar FleetWise señales AWS de IoT.

```
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.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ImportSignalCatalog` API.

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

Actualizar un catálogo AWS de FleetWise señales de IoT

Puede utilizar la operación de la 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*Reemplácelo por el nombre del catálogo de señales que está actualizando.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configurar FleetWise señales AWS de IoT.

> **Important**
>
> Las estructuras personalizadas son inmutables. Si necesita reordenar o insertar propiedades en una estructura personalizada (estructura) existente, elimine la estructura y cree una estructura completamente 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
      }
     }]
    }
```

Si ha activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la API. `UpdateSignalCatalog`

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

```
}
```

Compruebe la actualización del catálogo de señales

Puede usar la operación de la ListSignalCatalogNodesAPI para verificar si se actualizó 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
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListSignalCatalogNodes` API.

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

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

Eliminar un catálogo AWS de FleetWise señales de IoT

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:

- Eliminar un modelo AWS de FleetWise vehículo de IoT
- Eliminar un manifiesto de FleetWise decodificador de AWS IoT
- Eliminar un FleetWise vehículo AWS de IoT
- Eliminar una FleetWise flota AWS de IoT
- Eliminar una FleetWise campaña AWS de IoT

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
```

Verifica la eliminación del catálogo de señales

Puede utilizar la operación de la ListSignalCatalogsAPI para verificar 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
```

Si habilitó el cifrado mediante una AWS KMS clave administrada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `ListSignalCatalogs` API.

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

    {
```

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

Obtenga información sobre el catálogo FleetWise de señales de AWS IoT

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 desee recuperar.

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

Si habilitó el cifrado mediante una AWS KMS clave gestionada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `GetSignalCatalog` API.

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

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

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

Gestione los modelos de FleetWise vehículos de AWS IoT

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 Administre FleetWise vehículos de AWS IoT.

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**
>
> - Debe disponer de un catálogo de señales antes de poder crear un modelo de vehículo mediante la operación de la `CreateModelManifest` API. Para obtener más información, consulte Cree un catálogo AWS de FleetWise señales de IoT.
> - 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

- Cree un modelo AWS de FleetWise vehículo de IoT
- Actualizar un modelo AWS de FleetWise vehículo de IoT
- Eliminar un modelo AWS de FleetWise vehículo de IoT
- Obtenga información sobre modelos de FleetWise vehículos de AWS IoT

Cree un modelo AWS de FleetWise vehículo de IoT

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

Temas

- Creación de un modelo de vehículo (consola)
- Creación de un modelo de vehículo (AWS CLI)

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
- Creación manual de un modelo de vehículo
- Duplicación de un modelo de vehículo

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 Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

Para crear un modelo de vehículo mediante una plantilla

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 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 Agregar.

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. 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 y, a continuación, haga lo siguiente.

Temas

- Paso 1: Configurar el modelo de vehículo
- Paso 2: Agregar señales
- Paso 3: Importar señales
- (Opcional) Paso 4: Agregar atributos
- Paso 5: Revisar y crear

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 uno de los siguientes: INT8,,,,, BOOLEAN UINT8, FLOAT INT16 UINT16 INT32, DOUBLE UINT32 INT64, STRING UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 _ARRAY, INT16 BOOLEAN_ARRAY, UINT16 INT32 FLOAT_ARRAY, UINT32 DOUBLE_ARRAY, INT64 UINT64 STRING_ARRAY, UNIX_TIMESTAMP_ARRAY o UNKNOWN.
   - Tipo de señal: el tipo de señal, que puede ser sensor o actuador.
   - (Opcional) Unit: la unidad científica de la señal, como km o grados Celsius.
   - (Opcional) Path: la ruta a la señal. Similar a JSONPath, usa 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) Min: el valor mínimo de la señal.
   - (Opcional) Max: el valor máximo de la señal.
   - (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).
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 uno de los siguientes: INT8,,,,, BOOLEAN UINT8 INT16, FLOAT UINT16 INT32 UINT32, DOUBLE INT64 UINT64, STRING, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 _ARRAY, INT16 _ARRAY, UINT16 INT32 BOOLEAN_ARRAY, UINT32 FLOAT_ARRAY, INT64 UINT64 DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY o UNKNOWN
   - (Opcional) Unit: la unidad científica del atributo, como km o grados Celsius.
   - (Opcional) Path: la ruta a la señal. Similar a JSONPath, usa 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) Min: el valor mínimo del atributo.
   - (Opcional) Max: el valor máximo del atributo.
   - (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).
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. Abre la FleetWise consola AWS de 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**
>
> Debe disponer de un catálogo de señales antes de poder crear un modelo de vehículo mediante la operación de la `CreateModelManifest` API. Para obtener más información acerca de cómo crear un catálogo de señales, consulte Cree un catálogo AWS de FleetWise señales de IoT.

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.
- *signal-catalog-ARN*Sustitúyalo por el Amazon Resource Name (ARN) del catálogo de señales.
- (Opcional) *description* Sustitúyalo por una descripción que le ayude a identificar el modelo del vehículo.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configurar FleetWise señales AWS de IoT.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `CreateModelManifest` API.

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

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

Actualizar un modelo AWS de FleetWise vehículo de IoT

Puedes usar la operación de la UpdateModelManifestAPI para actualizar un modelo de vehículo existente (manifiestos de modelo). 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) *description* Sustitúyala por una descripción actualizada que te ayude a identificar el modelo del vehículo.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `UpdateModelManifest` API.

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

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

Verifica la actualización del modelo del vehículo

Puedes usar la operación de la ListModelManifestNodesAPI para verificar si se actualizó 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
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListModelManifestNodes` API.

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

```
      ]
    },
  ]
}
```

Eliminar un modelo AWS de FleetWise vehículo de IoT

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 obtener más información, consulte Eliminar un FleetWise vehículo AWS de IoT y Eliminar un manifiesto de FleetWise decodificador de AWS IoT.

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. 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 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)

Puede utilizar 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
```

Verifica la eliminación del modelo de vehículo

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
```

Si habilitó el cifrado mediante una AWS KMS clave administrada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `ListModelManifests` API.

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

Obtenga información sobre modelos de FleetWise vehículos de AWS IoT

Puede utilizar la operación de la GetModelManifestAPI para recuperar información sobre un modelo de vehículo. En el siguiente ejemplo se utiliza AWS CLI.

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

*vehicle-model*Sustitúyalo 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.

Si ha activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `GetModelManifest` API.

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

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

Gestione los AWS manifiestos de FleetWise decodificadores de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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 los decodificadores de señal son los componentes principales con los que se trabaja para configurar los manifiestos de los decodificadores.

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 (ECUs). ECUs puede ser la unidad de control del motor, los airbags o el sistema de audio.

Diagnóstico a bordo (OBD) II

Un protocolo desarrollado aún más que define cómo se comunican los datos de autodiagnóstico entre sí. ECUs Proporciona una serie de códigos de diagnóstico de problemas estándar (DTCs) que ayudan a identificar qué es lo que está mal en tu 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.

Interfaces personalizadas

También puede utilizar su propia interfaz para decodificar las señales en el Edge. Esto puede ahorrarle tiempo, ya que no necesita crear reglas de decodificación en la nube.

Decodificador de señales

Proporciona información de decodificación detallada para una señal específica. Todas las señales especificadas en el modelo del vehículo deben estar emparejadas con un decodificador de señales. 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 decodificadores de señal OBD.

El manifiesto del decodificador debe contener decodificadores de señales de mensajes si también contiene interfaces de middleware para vehículos. O bien, si el manifiesto del decodificador contiene interfaces de decodificación personalizadas, también debe contener señales de decodificación personalizadas.

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 utilizar las teclas ListDecoderManifests y GetDecoderManifest APIs 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

- Configure AWS las interfaces FleetWise de red de IoT y las señales del decodificador
- Crea un manifiesto de FleetWise decodificador de AWS IoT
- Actualizar un manifiesto de FleetWise decodificador de AWS IoT
- Eliminar un manifiesto de FleetWise decodificador de AWS IoT
- Obtenga información sobre el manifiesto del FleetWise decodificador de AWS IoT

Configure AWS las interfaces FleetWise de red de IoT y las señales del decodificador

Important

El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Cada manifiesto de decodificador tiene al menos una interfaz de red y decodificadores de señal emparejados con las señales especificadas en el modelo de vehículo asociado.

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

Temas

- Configuración de interfaces de red
- Configure los decodificadores de señal

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 (DTCs) al vehículo en cuestión de segundos. Por ejemplo, si el valor especificado es 120, el software Edge Agent recopila los datos almacenados DTCs una vez cada 2 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 el estándar World Wide Harmonisation On-Board Diagnostics (WWH-OBD) -4. ISO15765
- (Opcional)`pidRequestIntervalSeconds`: con qué frecuencia se debe solicitar el OBD II desde el vehículo. PIDs Por ejemplo, si el valor especificado es 120, el software Edge Agent recopila el OBD II PIDs una vez cada 2 minutos.
- (Opcional)`useExtendedIds`: si se debe utilizar la extensión IDs 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`

Para configurar una interfaz de decodificación personalizada, especifique la siguiente información.

- `name`— El nombre del decodificador que utiliza para decodificar las señales en el Edge.

  El nombre de la interfaz del decodificador puede tener entre 1 y 100 caracteres.

Configure los decodificadores de señal

Para configurar un decodificador de señales 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.
- (Opcional)`signalValueType`: el tipo de valor de la señal. El tipo de valor predeterminado es entero.

Para configurar un decodificador de señales 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.
- (Opcional)`isSigned`: si el mensaje está firmado. Si lo está, puede representar números positivos y negativos. El mensaje no está firmado de forma predeterminada (`false`).
- (Opcional)`signalValueType`: el tipo de valor de la señal. El tipo de valor predeterminado es entero.

Para configurar un decodificador de señales 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.

Para configurar una señal de decodificación personalizada, especifique la siguiente información.

- (Opcional)`id`: el ID de la señal que usted mismo decodifica mediante la interfaz del decodificador. El ID de la señal puede tener entre 1 y 150 caracteres. Si no se especifica, el `id` valor predeterminado es el `fullyQualifiedName` de la señal.

Crea un manifiesto de FleetWise decodificador de AWS IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puedes usar la FleetWise consola de AWS IoT o la API para crear un manifiesto de decodificador para tu modelo 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. Abre la FleetWise consola AWS de 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
- Paso 2: Mapear la interfaz CAN
- Paso 3: Revisar y crear

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.

Agregue 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

1. Cargue un archivo de interfaz de red. Puede cargar un archivo.dbc para los protocolos CAN o un archivo.json para ROS 2 o interfaces personalizadas.
2. Introduzca un nombre para la interfaz de red. Si ha cargado una interfaz personalizada, el nombre ya está proporcionado.

Mapee las señales faltantes

Si hay señales en el modelo de vehículo a las que les faltan decodificadores de señal emparejados en las interfaces de red cargadas, puede crear un decodificador personalizado predeterminado que mapee las señales faltantes. Esto es opcional, ya que puede mapear las señales manualmente en el siguiente paso.

Para crear un decodificador personalizado predeterminado

1. Seleccione Crear un decodificador personalizado predeterminado para las señales que faltan.
2. Elija Siguiente.

Paso 2: Mapear la interfaz CAN

Puede mapear las señales CAN con decodificadores de señales CAN. Si ha seleccionado la casilla Crear un decodificador personalizado predeterminado para las señales ausentes, cualquier señal a la que le falte una señal del decodificador se mapea automáticamente a los decodificadores de señales personalizados predeterminados.

Para mapear señales CAN

1. En el mapeo de señales CAN, seleccione un decodificador de señales.

2. Elija Siguiente.

> **Note**
>
> Si ha añadido una ROS 2 o una interfaz personalizada, puede verificar las asignaciones antes de crear el manifiesto del decodificador.

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 los manifiestos del decodificador. El siguiente ejemplo utiliza AWS CLI.

> **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 Cree un modelo AWS de FleetWise vehículo de IoT.

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.
- *vehicle-model-ARN*Sustitúyalo por el nombre de recurso de Amazon (ARN) del modelo de vehículo.
- (Opcional) *description* Sustitúyalo por una descripción que le ayude a identificar el manifiesto del decodificador.

Para obtener más información acerca de cómo configurar ramificaciones, atributos, sensores y actuadores, consulte Configure AWS las interfaces FleetWise de red de IoT 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.
- *vehicle-model-ARN*Sustitúyalo por el nombre de recurso de Amazon (ARN) del modelo de vehículo.
- (Opcional) *description* Sustitúyalo por una descripción que le 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 Configure AWS las interfaces FleetWise de red de IoT 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"
         }
        }
       }
      ]
     }
    }
   ]
}
```

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

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

```
{
```

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

> **Note**
>
> 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.

Si habilitaste el cifrado con una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `CreateDecoderManifest` API.

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

Actualizar un manifiesto de FleetWise decodificador de AWS IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puedes usar la operación de la UpdateDecoderManifestAPI para actualizar el manifiesto de un 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
```

Si las señales no tienen reglas de decodificación especificadas, puede crear reglas de decodificación predeterminadas. Las señales se añaden a una interfaz decodificada personalizada `CustomDecodingSignal$id` con el nombre completo de la señal. Para actualizar el manifiesto de un decodificador con las reglas de decodificación predeterminadas, ejecute el siguiente comando.

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

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

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

Si has activado el cifrado con una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `UpdateDecoderManifest` API.

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

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

Verifica la actualización del manifiesto del decodificador

Puede utilizar la operación de la 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
```

Si has activado el cifrado con una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListDecoderManifestSignals` API.

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

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

Puede utilizar la operación de 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
```

Si has activado el cifrado con una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListDecoderManifestNetworkInterfaces` API.

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

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

Eliminar un manifiesto de FleetWise decodificador de AWS IoT

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 Eliminar un FleetWise vehículo AWS de IoT.

Temas

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

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. Abre la FleetWise consola AWS de 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 Eliminar un FleetWise vehículo AWS de IoT.

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
```

Verifica la eliminación del manifiesto del decodificador

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
```

Si habilitó el cifrado mediante una AWS KMS clave administrada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `ListDecoderManifests` API.

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

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

Obtenga información sobre el manifiesto del FleetWise decodificador de AWS IoT

Puede utilizar la operación de la GetDecoderManifestAPI para comprobar si se han actualizado las interfaces de red y los decodificadores de señal del manifiesto del decodificador. En el siguiente ejemplo se utiliza. AWS CLI

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

*decoder-manifest*Sustitúyalo por el nombre del manifiesto del decodificador que desee 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.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `GetDecoderManifest` API.

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

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

Administre FleetWise vehículos de AWS IoT

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 Modelos de FleetWise vehículos AWS IoT.

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 AWS de FleetWise vehículos de IoT
- Temas reservados en AWS IoT FleetWise
- Crea un FleetWise vehículo AWS de IoT
- Cree varios FleetWise vehículos AWS de IoT
- Actualizar un FleetWise vehículo AWS de IoT
- Actualice varios FleetWise vehículos AWS de IoT
- Eliminar un FleetWise vehículo AWS de IoT
- Obtenga información sobre FleetWise vehículos de AWS IoT

Aprovisionamiento AWS de FleetWise vehículos de IoT

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 Crea un FleetWise vehículo AWS de IoT.

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.
  - *policy-name*Sustitúyalo por el nombre de la política que desee crear.
  - *file-name*Sustitúyalo 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:

- *region*Sustitúyala por la AWS región en la que creaste FleetWise los recursos de AWS IoT.
- *awsAccount*Sustitúyalo por tu 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 en AWS IoT FleetWise.

```
{
    "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:

   - *cert*Sustitúyalo por el nombre del archivo que guarda el contenido del resultado del comando de CertificatePEM.
   - *public-key*Sustitúyalo por el nombre del archivo que guarda el contenido de la salida del comando de KeyPair. PublicKey.
   - *private-key*Sustitúyalo por el nombre del archivo que guarda el contenido de la salida del comando 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:

   - *policy-name*Sustitúyalo por el nombre de la AWS IoT Core política que creó.
   - *certificate-arn*Sustitúyalo por el ARN del certificado que copió.

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

```
    --policy-name policy-name\
    --target "certificate-arn"
```

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

   - *thing-name*Sustitúyalo por el nombre de AWS IoT tu objeto o el identificador de tu vehículo.
   - *certificate-arn*Sustitúyalo por el ARN del certificado que copió.

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

Temas reservados en AWS IoT FleetWise

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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`	Publish	El software Edge Agent publica información sobre el estado del vehículo en este tema. La información sobre el estado del vehículo	

Tema	Operación de cliente permitida	Descripción	
		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.	
`$aws/iotfleetwise/vehicles/ vehicleName /signals`	Publish	El software Edge Agent publica señales en este tema. La información de la señal se intercambia en formato de búferes de protocolo (Protobuf). Para obtener más información, consulte la Guía para desarrolladores de FleetWise software Edge Agent para AWS IoT.	
`$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.	

Tema	Operación de cliente permitida	Descripción	
`$aws/iotfleetwise/vehicles/vehicleName/decoder_manifests`	Suscribirse	AWS IoT FleetWise publica manifiestos de decodificadores sobre este tema. Los vehículos consumen estos manifiestos del decodificador.	
`$aws/iotfleetwise/vehicles/vehicleName/command/request`	Suscribirse	AWS IoT FleetWise publica solicitudes para ejecutar comandos en este tema. Luego, los vehículos consumen estas solicitudes de comando.	

Tema	Operación de cliente permitida	Descripción	
`$aws/iotfleetwise/vehicles/vehicleName/command/response`	Publish	El software Edge Agent publica las respuestas a los comandos del vehículo sobre este tema. Las respuestas a los comandos se intercambian en formato de búferes de protocolo (Protobuf). Para obtener más información, consulte la Guía para desarrolladores de FleetWise software Edge Agent para AWS IoT.	
`$aws/iotfleetwise/vehicles/vehicleName/command/notification`	Suscribirse	AWS IoT FleetWise publica actualizaciones del estado de los comandos sobre este tema. Las notificaciones se envían en formato JSON.	

Tema	Operación de cliente permitida	Descripción	
`$aws/iotfleetwise/vehicles/${vehicle_name}/last_known_states/config`	Suscribirse	AWS IoT FleetWise publica las configuraciones de plantillas de estado sobre este tema. Los vehículos consumen estas configuraciones de plantillas de estado.	
`$aws/iotfleetwise/vehicles/${vehicle_name}/last_known_states/data`	Publish	El software Edge Agent publica los datos recopilados a partir de las señales relacionadas con este tema.	

Tema	Operación de cliente permitida	Descripción	
`$aws/iotfleetwise/vehicles/` *`$vehicle_name`* `/last_known_state/` *`$state_template_name`* `/data`	Suscribirse	AWS IoT FleetWise publica los datos recopilados de las señales configuradas en lo especificado *`$state_template_name`* en este tema. Las actualizaciones pueden ser parciales. Por ejemplo, si una asociación de plantillas de estados contiene varias señales con la estrategia de actualización en caso de cambio, un mensaje determinado solo incluirá las señales que hayan cambiado. La información de 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.	

Crea un FleetWise vehículo AWS de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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 Gestione los modelos de FleetWise vehículos de AWS IoT.
> - 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 Gestione los AWS manifiestos de FleetWise decodificadores de IoT.

Temas

- Creación de un vehículo (consola)
- Creación de un vehículo (AWS CLI)

Creación de un vehículo (consola)

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

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
- Paso 2: Configurar el certificado del vehículo
- Paso 3: Asociar políticas al certificado
- Paso 4: Revisar y crear

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.

Para crear un vehículo, ejecute el siguiente comando:

*file-name*Sustitúyalo 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 del 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. *`thing-name`*Sustitúyalo por el nombre del objeto que desee 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 AWS de FleetWise vehículos de IoT.

- *`vehicle-name`*Sustitúyalo por uno de los siguientes.
  - El nombre del objeto, AWS IoT si `associationBehavior` está configurado para ello`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 (:).
- *`model-manifest-ARN`*Sustitúyalo por el ARN del modelo de su vehículo (manifiesto del modelo).
- *`decoder-manifest-ARN`*Sustitúyalo 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"
    }
}
```

Example — asociar una plantilla de estado al vehículo

Puede asociar plantillas de estado al vehículo para poder recopilar las actualizaciones de estado del vehículo en la nube mediante el uso del `stateTemplates` campo.

En este ejemplo, *`stateTemplateUpdateStrategy`* puede ser una de las siguientes:

- `periodic`: le permite especificar una velocidad fija a la que el software Edge Agent enviará las actualizaciones de señal a la nube (el software Edge Agent enviará las actualizaciones incluso si el valor de la señal no ha cambiado entre las actualizaciones).
- `onChange`: El software Edge Agent enviará actualizaciones de señal cada vez que la señal cambie.

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

Dónde contiene el *`create-vehicle.json`* archivo (por ejemplo):

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

```
                    "stateTemplateUpdateRate": {
                        "unit": "SECOND",
                        "value": 10
                    }
                }
            }
        }
    ]
}
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `CreateVehicle` API.

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

Cree varios FleetWise vehículos AWS de IoT

Puede utilizar 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:

*file-name*Sustitúyalo 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 Crea un FleetWise vehículo AWS de IoT.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `BatchCreateVehicle` API.

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

```
      ]
    },
  ]
}
```

Actualizar un FleetWise vehículo AWS de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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:

*file-name*Sustitúyalo por el nombre del archivo.json que contiene la configuración de tu vehículo.

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

Example — configuración del vehículo

- *vehicle-name*Sustitúyala por la ID del vehículo que deseas actualizar.
- (Opcional) *model-manifest-ARN* Sustitúyalo por el ARN del modelo de vehículo (manifiesto del modelo) que utilizas para sustituir el modelo de vehículo en uso.
- (Opcional) *decoder-manifest-ARN* Sustitúyalo por el ARN del manifiesto del decodificador asociado al nuevo modelo de vehículo que especificaste.
- (Opcional) Sustitúyalo por *attribute-update-mode* 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"
        }
}
```

Example — añadir o eliminar plantillas de estado asociadas al vehículo

Puede asociar plantillas de estado adicionales o eliminar las asociaciones existentes del vehículo mediante los siguientes campos:

- `stateTemplatesToAdd`
- `stateTemplatesToRemove`

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

Dónde contiene el *update-vehicle.json* archivo (por ejemplo):

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `UpdateVehicle` API.

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

Actualice varios FleetWise vehículos AWS de IoT

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:

*file-name*Sustitúyalo 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 Actualizar un FleetWise vehículo AWS de IoT.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `BatchUpdateVehicle` API.

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

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

Eliminar un FleetWise vehículo AWS de IoT

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

> **Important**
>
> Tras eliminar un vehículo, el AWS IoT lo elimina FleetWise automáticamente de las flotas y campañas asociadas. Para obtener más información, consulte Gestione flotas en IoT AWS FleetWise y Recopile FleetWise datos de AWS IoT 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. Abre la FleetWise consola AWS de 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:

*`vehicle-name`*Sustitúyalo por el identificador del vehículo que desee eliminar.

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

Verifica la eliminación del vehículo

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
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListVehicles` API.

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

```
}
```

Obtenga información sobre FleetWise vehículos de AWS IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede utilizar 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:

*vehicle-name*Sustitúyala por la ID del vehículo que deseas 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.

Puede utilizar la operación de la GetVehicleStatusAPI para recuperar el estado de los recursos asociados a un vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar el estado de los recursos asociados a un vehículo, ejecuta el siguiente comando.

- *vehicle-name*Sustitúyalo por el identificador del vehículo al que están asociados los recursos.
- *type*Sustitúyalo por el tipo de recurso cuyo estado desee recuperar. Los valores válidos para `type` son `CAMPAIGN`, `STATE_TEMPLATE` y `DECODER`.

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

Si habilitó el cifrado mediante una AWS KMS clave administrada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar las operaciones `GetVehicle` o las de la `GetVehicleStatus` API.

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

Gestione flotas en IoT AWS FleetWise

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 Recopile FleetWise datos de AWS IoT con campañas.

Una flota contiene la siguiente información:

`fleetId`

El ID de la flota.

(Opcional) `description`

Una descripción para ayudarlo a encontrar la flota.

`signalCatalogArn`

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

AWS El IoT FleetWise proporciona las siguientes operaciones de API que puede usar 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— Disocia un vehículo de 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 todas las flotas a IDs 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

- Cree una FleetWise flota AWS de IoT
- Asocie un FleetWise vehículo de AWS IoT a una flota
- Disociar un FleetWise vehículo de AWS IoT de una flota
- Actualice una FleetWise flota AWS de IoT
- Eliminar una FleetWise flota AWS de IoT
- Obtenga información sobre la FleetWise flota de AWS IoT

Cree una FleetWise flota AWS de IoT

Puede utilizar la operación de la CreateFleetAPI para crear una flota de vehículos. En el siguiente ejemplo se utiliza AWS CLI.

> **Important**
>
> Para poder crear una flota, debe disponer de un catálogo de señales. Para obtener más información, consulte Cree un catálogo AWS de FleetWise señales de IoT.

Para crear una flota, ejecute el siguiente comando:

- *fleet-id*Reemplácelo 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) *description* Sustitúyalo por una descripción.

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

```
aws iotfleetwise create-fleet \
   --fleet-id fleet-id \
   --description description \
```

```
    --signal-catalog-arn signal-catalog-arn
```

Si habilitó el cifrado mediante una AWS KMS clave gestionada por el cliente, incluya la siguiente declaración de política para que su función pueda invocar la operación de la `CreateFleet` API.

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

Asocie un FleetWise vehículo de AWS IoT a una flota

Puede utilizar la operación de la AssociateVehicleFleetAPI para asociar un vehículo a una flota. En el siguiente ejemplo se 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 Administre FleetWise vehículos de AWS IoT.
> - Si asocias un vehículo a una flota a la que apunta una campaña, el AWS IoT despliega FleetWise automáticamente la campaña en el vehículo.

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

- *fleet-id*Sustitúyala por el identificador de la flota.
- *vehicle-name*Sustitúyalo por el identificador del vehículo.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `AssociateVehicleFleet` API.

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

Disociar un FleetWise vehículo de AWS IoT de una flota

Puede utilizar la operación de la DisassociateVehicleFleetAPI para desvincular un vehículo de una flota. En el siguiente ejemplo se utiliza AWS CLI.

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

- *fleet-id*Sustitúyalo por el identificador de la flota.
- *vehicle-name*Sustitúyalo por el identificador del vehículo.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `DisassociateVehicleFleet` API.

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

Actualice una FleetWise flota AWS de IoT

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

Para actualizar una flota, ejecute el siguiente comando:

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

  La descripción puede contener entre 1 y 2048 caracteres.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `UpdateFleet` API.

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

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

Eliminar una FleetWise flota AWS de IoT

Puede utilizar la operación de la DeleteFleetAPI para eliminar una flota. En el siguiente ejemplo se 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 Disociar un FleetWise vehículo de AWS IoT de una flota.

Para eliminar una flota, ejecute el siguiente comando:

*fleet-id*Sustitúyalo por el identificador de la flota que vas a eliminar.

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

Verifica la eliminación de la flota

Puedes usar la operación de la ListFleetsAPI 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
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListFleets` API.

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

Obtenga información sobre la FleetWise flota de AWS IoT

Puedes usar la operación de la ListFleetsForVehicleAPI para recuperar una lista paginada IDs de todas las flotas a las que pertenece el vehículo. El siguiente ejemplo utiliza AWS CLI.

Para recuperar una lista paginada IDs de todas las flotas a las que pertenece el vehículo, ejecuta el siguiente comando.

*vehicle-name*Sustitúyala por la ID del vehículo.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListFleetsForVehicle` API.

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

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

Puede utilizar la operación de la ListVehiclesInFleetAPI 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:

*fleet-id*Sustitúyala por la ID de la flota.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `ListVehiclesInFleet` API.

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

Puedes usar la operación de la GetFleetAPI para recuperar la información de la flota. El siguiente ejemplo utiliza AWS CLI.

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

*fleet-id*Sustitúyala por la 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.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `GetFleet` API.

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

Recopile FleetWise datos de AWS IoT con campañas

Important

El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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

Usted crea campañas en la nube. Una vez que tú o tu equipo hayan aprobado una campaña, el AWS IoT FleetWise establece la campaña como lista para su despliegue y se desplegará la próxima vez que se registre el vehículo. 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.

Important

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, instalar y trabajar con el software Edge Agent, haga lo siguiente.
  1. Abre la FleetWise consola AWS de IoT.
  2. En la página de inicio del servicio, en la FleetWise sección Comenzar con AWS IoT, elija Explore Edge Agent.
- Lo has configurado AWS IoT Core para aprovisionar tu vehículo. Para obtener más información, consulte Aprovisionamiento AWS de FleetWise vehículos de IoT.

Note

También puedes utilizar plantillas de estado Supervisa el último estado conocido de tus vehículos (no de flotas) prácticamente en tiempo real, que te permiten transmitir datos

de telemetría con una estrategia de actualización «en caso de cambio» o «periódica».
La función también incluye funciones «bajo demanda» para activar o desactivar las plantillas desplegadas anteriormente o solicitar el estado actual del vehículo una sola vez (recuperarlo).
El acceso al último estado conocido está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

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 recopilar o cuándo recopilarlos. AWS FleetWise Actualmente, el IoT admite el esquema de recolección basado en condiciones y el esquema de recolección basado en el tiempo.

- `conditionBasedCollectionScheme`— el esquema de recopilación basado en condiciones utiliza una expresión lógica para reconocer qué datos recopilar. El software Edge Agent recopila datos cuando se cumple la condición.
  - `expression`— la expresión lógica utilizada para reconocer qué datos recopilar. 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 FleetWise campañas AWS de IoT.

- (Opcional)`conditionLanguageVersion`: la versión del lenguaje de expresiones condicionales.
- (Opcional)`minimumTriggerIntervalMs`: el tiempo mínimo entre dos eventos de recopilación de datos, en milisegundos. Si una señal cambia con frecuencia, es conveniente recopilar datos a un ritmo más lento.
- (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.

- `timeBasedCollectionScheme`— cuando defina un esquema de recopilación basado en el tiempo, especifique un período de tiempo 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.
  - `periodMs`— el período de tiempo (en milisegundos) para decidir con qué frecuencia se recopilarán los datos.

(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`

Elige el único destino al que la campaña transferirá los datos del vehículo. Puede enviar los datos a un tema de MQTT o almacenarlos en Amazon S3 o Amazon Timestream.

El MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y ampliamente adoptado. Puede publicar datos en un tema de MQTT para crear sus propias arquitecturas basadas en eventos mediante reglas. AWS IoT AWS IoT La compatibilidad con MQTT se basa en la especificación MQTT v3.1.1 y la especificación MQTT v5.0, con algunas diferencias. Para obtener más información, consulte Diferencias entre MQTT.

S3 puede ser un mecanismo de almacenamiento de datos rentable que ofrece capacidades duraderas de administración de datos y servicios de datos posteriores. 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.

> **Note**
>
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

(Opcional) `dataExtraDimensions`

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

(Opcional) `dataPartitions`

Cree una partición de datos para almacenar temporalmente los datos de señal de un vehículo. Usted configura cuándo y cómo enviar los datos a la nube.

- Especifique cómo el AWS IoT FleetWise almacena los datos en un vehículo o flota definiendo el tamaño máximo de almacenamiento, el tiempo mínimo de vida y la ubicación de almacenamiento.
- La campaña `spoolingMode` debe serlo`TO_DISK`.
- Las configuraciones de carga incluyen la definición de la versión del lenguaje de condiciones y la expresión lógica.

(Opcional) `description`

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

(Opcional) `diagnosticsMode`

Cuando el modo de diagnóstico está configurado para ello`SEND_ACTIVE_DTCS`, la campaña envía códigos de problemas de diagnóstico estándar y almacenados que ayudan a identificar qué es lo que está mal en tu vehículo. DTCs Por ejemplo, el número 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`

Defina la fecha de caducidad de su 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`

Especifique 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.

- `name`— el nombre de la señal desde la que se recopilan los datos cuando se invoca el esquema de recopilación de datos.
- `dataPartitionId`— el ID de la partición de datos que se va a utilizar en la señal. El ID debe coincidir con uno de los IDs proporcionados`dataPartitions`. Si carga una señal como condición en su partición de datos, deberá incluir esas mismas señales`signalsToCollect`.
- (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`: 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.

  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 tu 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 despliega FleetWise automáticamente la campaña en el vehículo o la flota objetivo. Para obtener más información, consulte Actualizar una FleetWise campaña AWS de IoT.
- `RUNNING` : la campaña está activa.
- `SUSPENDED`: la campaña está suspendida. Para reanudar la campaña, utilice la operación de la API `UpdateCampaign`.

AWS El IoT FleetWise proporciona las siguientes operaciones de API que puedes usar 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

- Crea una FleetWise campaña AWS de IoT
- Actualizar una FleetWise campaña AWS de IoT
- Eliminar una FleetWise campaña AWS de IoT
- Obtenga información sobre FleetWise la campaña de AWS IoT
- Almacene y reenvíe los datos de la campaña
- Recopile datos de códigos de problemas de diagnóstico mediante AWS IoT FleetWise
- Visualice los datos de FleetWise vehículos de AWS IoT

Crea una FleetWise campaña AWS de IoT

Important

El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puedes usar la FleetWise consola de AWS IoT o la API 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, instalar y trabajar con el software Edge Agent, haga lo siguiente:

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

- Lo has configurado AWS IoT Core para aprovisionar tu vehículo. Para obtener más información, consulte Aprovisionamiento AWS de FleetWise vehículos de IoT.

Temas

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

Creación de una campaña (consola)

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

Para crear una campaña

1. Abre la FleetWise consola AWS de IoT.
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: Especifique las condiciones de almacenamiento y carga
- Paso 3: Configurar el destino de los datos
- Paso 4: Añadir vehículos
- Paso 5: Revisar y crear
- Paso 6: Despliega una campaña

⚠ Important

- Para poder crear una campaña, debe disponer de un catálogo de señales y de un vehículo. Para obtener más información, consulte Gestione los catálogos FleetWise de señales de AWS IoT y Administre FleetWise vehículos de AWS IoT.

- Una vez creada una campaña, debe aprobarla. Para obtener más información, consulte Actualizar una FleetWise campaña AWS de IoT.

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 FleetWise consola de AWS IoT, puede configurar un esquema de recopilación de datos de las siguientes maneras:

- 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 especificó un esquema de recopilación de datos basado en condiciones, debe definir una expresión lógica para reconocer qué datos recopilar. AWS El IoT FleetWise utiliza una expresión lógica para reconocer qué datos recopilar para un esquema basado en condiciones. La expresión debe especificar el nombre completo de una señal como variable, un operador de comparación y un valor de comparación.

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

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

4. (Opcional) Especifique la versión de idioma de la expresión condicional. El valor predeterminado es 1.
5. (Opcional) Especifique el intervalo de activación 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 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.
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) Edite las opciones avanzadas del esquema.

   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, especifique 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 para definir el esquema de recopilación de datos, selecciona Cargar un archivo.json desde tu dispositivo local. AWS El IoT define FleetWise automáticamente las opciones que se pueden 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 sobre el formato de archivo requerido, consulte la documentación de la CreateCampaignAPI.

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

  2. AWS El IoT define FleetWise automáticamente el esquema de recopilación de datos en función de la información de su archivo. Revisa las opciones que AWS IoT FleetWise seleccionó para ti. Puede actualizar las opciones si es necesario.

Paso 2: Especifique las condiciones de almacenamiento y carga

Para elegir si el software Edge Agent almacenará temporalmente los datos de forma local cuando el vehículo no esté conectado a la nube, especifique el modo de bobina.

- En el modo de transferencia de datos, elige una de las siguientes opciones:

- No almacenado: el software Edge Agent recopila datos, pero no los almacena temporalmente de forma local cuando un vehículo está desconectado. El software Edge Agent transfiere los datos a la nube cuando el vehículo se vuelve a conectar.
- Almacenado en disco: el software Edge Agent recopila y almacena temporalmente los datos de forma local cuando un vehículo está desconectado. Los datos recopilados se almacenan temporalmente en una ubicación definida en la sección de «persistencia» del archivo de configuración de Edge Agent. El Edge Agent transfiere los datos a la nube cuando el vehículo se vuelve a conectar.
- Almacenados en un disco con particiones: el vehículo siempre almacena temporalmente los datos en el Edge, en la partición de datos especificada. Puedes elegir cuándo quieres reenviar los datos almacenados a la nube.
  1. (Opcional) Introduzca un identificador de partición para designar un conjunto de datos concreto.
  2. Introduzca un nombre de carpeta como ubicación en la que se almacenarán los datos. La ruta absoluta de la ubicación de almacenamiento es`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.
  3. Introduzca el tamaño máximo de almacenamiento de los datos almacenados en la partición. Los datos más recientes sobrescriben los datos más antiguos cuando la partición alcanza el tamaño máximo.
  4. Introduzca el tiempo mínimo durante el que los datos de esta partición permanecerán en el disco.
  5. (Opcional) Introduzca las condiciones de carga de la partición.

Especificación de señales

Puede especificar las señales de las que se recopilarán los datos durante la campaña.

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

1. Selecciona el nombre de la señal.
2. (Opcional) Para obtener el número máximo de muestras, introduce la cantidad máxima de muestras de datos que el software Edge Agent recopila y transfiere a la nube durante la campaña.

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 3: Configurar el destino de los datos

> **Note**
>
> Si la campaña contiene señales de datos del sistema de visión, solo puede almacenar los datos del vehículo en Amazon S3. No puede almacenarlos en Timestream ni enviarlos a un tema de MQTT.
> Los datos de sistemas de visión están en versión de vista previa y sujetos a cambios.
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

Elija el destino al que desea enviar o almacenar los datos recopilados por la campaña. Puede enviar los datos del vehículo a un tema de MQTT o almacenarlos en Amazon S3 o Amazon Timestream.

En Configuración de destino, haga lo siguiente:

- Elija el tema Amazon S3, Amazon Timestream o MQTT en la lista desplegable.

Amazon S3

> **Important**
>
> Solo puedes transferir datos a S3 si el AWS IoT FleetWise tiene permisos para escribir en el bucket de S3. Para obtener más información sobre la concesión de acceso, consulte Controlar el acceso con AWS IoT FleetWise.

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 Crear, configurar y trabajar 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.

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 costos 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.

Amazon Timestream

> **Important**
>
> Solo puedes transferir datos a una tabla si el AWS IoT FleetWise tiene permisos para escribir datos en Timestream. Para obtener más información sobre la concesión de acceso, consulte Controlar el acceso con AWS IoT FleetWise.
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

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.

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.

Tema MQTT

> **Important**
>
> Solo puedes enrutar datos a un tema de MQTT si AWS IoT FleetWise tiene permisos para los AWS IoT temas. Para obtener más información sobre la concesión de acceso, consulte Controlar el acceso con AWS IoT FleetWise.

Para enviar datos del vehículo a un tema de MQTT, elija un tema de MQTT.

Los datos del vehículo enviados por la mensajería MQTT se envían prácticamente en tiempo real y te permiten usar reglas para tomar medidas o enviar los datos a otros destinos. Para obtener más información sobre el uso de MQTT, consulte las reglas y protocolos de comunicación de los dispositivosAWS IoT en la Guía para AWS IoT Core desarrolladores.

1. En el tema MQTT, introduzca el nombre del tema.
2. En el tema Acceso a servicios para MQTT, elija si quiere permitir que AWS IoT FleetWise cree y use un nuevo rol de servicio por usted. Si desea utilizar un rol de servicio existente, seleccione el rol en la lista desplegable bajo Seleccionar rol.

- Elija Siguiente.

Paso 4: Añadir 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 5: 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 6: Despliega 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)

Puedes usar la operación de la CreateCampaignAPI para crear una campaña. El siguiente ejemplo utiliza AWS CLI.

Al crear una campaña, los datos recopilados de los vehículos pueden enviarse a un tema de MQTT o almacenarse en Amazon S3 (S3) o 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 almacenar objetos con una escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector. Elija MQTT para entregar los datos prácticamente en tiempo real y utilice las reglas para realizar las acciones que AWS IoT defina o enrutar los datos a otros destinos.

> **Important**
>
> Solo puedes transferir los datos del vehículo a un tema de MQTT, Amazon S3 o Amazon AWS Timestream FleetWise si el IoT tiene permisos para enviar mensajes de MQTT en tu nombre o para escribir datos en S3 o Timestream. Para obtener más información sobre la concesión de acceso, consulte Controlar el acceso con AWS IoT FleetWise.
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

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 obtener más información, consulte Gestione los catálogos FleetWise de

señales de AWS IoT, Administre FleetWise vehículos de AWS IoT y Gestione flotas en IoT AWS FleetWise.

- Una vez creada una campaña, debe utilizar la operación de la API `UpdateCampaign` para aprobarla. Para obtener más información, consulte Actualizar una FleetWise campaña AWS de IoT

Para crear una campaña, ejecute el siguiente comando:

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

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

- *campaign-name*Sustitúyalo por el nombre de la campaña que estás creando.
- *signal-catalog-arn*Sustitúyalo por el Amazon Resource Name (ARN) del catálogo de señales.
- *target-arn*Sustitúyalo por el ARN de una flota o vehículo que haya creado.
- *bucket-arn*Sustitúyalo por el ARN del cucharón 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"
      }
    }
  ],
     "dataPartitions": [
      { ...  }
  ]
}
```

> **Note**
>
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

- *campaign-name*Reemplácelo por el nombre de la campaña que está creando.
- *signal-catalog-arn*Sustitúyalo por el ARN del catálogo de señales.
- *target-arn*Sustitúyalo por el ARN de una flota o vehículo que haya creado.
- *role-arn*Sustitúyalo por el ARN del rol de ejecución de tareas que otorga el FleetWise permiso de AWS IoT para entregar datos a la tabla Timestream.
- *table-arn*Sustitúyalo por el ARN de la tabla 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"
      }
    }
  ],
   "dataPartitions": [
      { ...  }
  ]
}
```

- *campaign-name*Sustitúyalo por el nombre de la campaña que estás creando.

- *signal-catalog-arn*Sustitúyalo por el Amazon Resource Name (ARN) del catálogo de señales.
- *target-arn*Sustitúyalo por el ARN de una flota o vehículo que haya creado.
- *topic-arn*Sustitúyalo por el ARN del tema MQTT que especificó como destino de los mensajes que contienen datos del vehículo.
- *role-arn*Sustitúyalo por el ARN del rol de ejecución de tareas que otorga FleetWise permiso a AWS IoT para enviar, recibir y realizar acciones en los mensajes del tema MQTT que especificaste.

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

```
            "mqttTopicConfig": {
                "mqttTopicArn": "topic-arn",
                "executionRoleArn": "role-arn"
            }
        }
    ]
}
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `CreateCampaign` API.

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

Expresiones lógicas para FleetWise campañas AWS de IoT

AWS El IoT FleetWise utiliza una expresión lógica 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`
}
```

Actualizar una FleetWise campaña AWS de IoT

Puedes usar la operación de la UpdateCampaignAPI para actualizar una campaña existente. El siguiente comando usa AWS CLI.

- *campaign-name*Sustitúyelo por el nombre de la campaña que está actualizando.
- Reemplace *action* por uno de los siguientes:
  - APPROVE— Aprueba la campaña para permitir que el AWS IoT FleetWise lo despliegue 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. Está previsto que la campaña se redistribuya en todos los vehículos la próxima vez que se registren y los vehículos volverán a enviar datos.
  - UPDATE— Actualiza la campaña definiendo los atributos y asociándolos a la campaña.
- *description*Sustitúyala por una nueva descripción.

  La descripción puede tener un máximo de 2.048 caracteres.
- *data-extra-dimensions*Sustitúyala por atributos específicos del vehículo para enriquecer los datos recopilados durante la campaña. Por ejemplo, puedes añadir la marca y el modelo del vehículo a la campaña, y el AWS IoT FleetWise asociará los datos con esos atributos como dimensiones en Amazon Timestream. A continuación, puede consultar los datos comparándolos con la marca y el modelo del vehículo.

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

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la UpdateCampaign API.

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

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

Eliminar una FleetWise campaña AWS de IoT

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

Eliminación de una campaña (consola)

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

Para eliminar una campaña

1. Abre la FleetWise consola AWS de IoT.
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)

Puedes usar la operación de la DeleteCampaignAPI para eliminar una campaña. En el siguiente ejemplo se utiliza AWS CLI.

Para eliminar una campaña, ejecute el siguiente comando:

*campaign-name*Sustitúyalo por el nombre del vehículo que vas a eliminar.

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

> **Las particiones de datos eliminadas no se pueden recuperar**
>
> Al eliminar una campaña, se eliminan todos los datos de los dispositivos y los datos de una partición no se subirán a la nube.

Verifica la eliminación de la campaña

Puedes usar la operación de la ListCampaignsAPI para verificar 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
```

Obtenga información sobre FleetWise la campaña de AWS IoT

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

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

*campaign-name*Sustitúyala por el nombre de la campaña que deseas 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.

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar la operación de la `GetCampaign` API.

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

Almacene y reenvíe los datos de la campaña

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Utilice particiones de datos en las campañas para almacenar temporalmente los datos de señales de vehículos y flotas en el Edge. Al configurar las opciones de carga y almacenamiento para las particiones de datos, puede optimizar las condiciones ideales para el reenvío de datos a los destinos de datos designados (como un bucket de Amazon S3). Por ejemplo, puede configurar la partición de datos para almacenar datos en un vehículo hasta que se conecte a una red Wi-Fi. Luego, una vez que el vehículo se conecta, la campaña activa el envío de los datos de esa partición concreta a la nube. Como alternativa, puede recopilar datos mediante AWS IoT Jobs.

Temas

- Cree particiones de datos
- Sube los datos de la campaña

- Sube datos mediante AWS IoT Jobs

Cree particiones de datos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Una partición de datos de una campaña almacena temporalmente los datos de las señales. Usted configura cuándo y cómo reenviar los datos a la nube.

Una partición de datos funciona designando primero un conjunto de datos concreto y utilizándolo `dataPartitionId` para una campaña. Luego, puede definir con más detalle las opciones de almacenamiento de la partición, como el tamaño máximo, el tiempo mínimo para mantener la partición de datos activa (en el disco) y dónde almacenar los datos en el Edge. Puede determinar la ubicación de almacenamiento en el vehículo utilizando`storageLocation`. La ubicación de almacenamiento determina el nombre de la carpeta de la partición de datos situada en la carpeta de almacenamiento de la campaña. La carpeta de almacenamiento de la campaña se encuentra en una carpeta que lleva el nombre del vehículo y sigue una ruta de persistencia definida en el archivo de configuración de Edge. Esta es la ruta absoluta de la ubicación de almacenamiento:`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.

El modo de bobina configurado para `TO_DISK` especificar que los datos particionados deben guardarse en un disco del vehículo. El almacenamiento de datos para las particiones de datos funciona según el método FIFO (primero en entrar, primero en salir). Si eliminas una campaña, también eliminas los datos de la partición de datos asociada. Si no especificas una partición de datos para los casos de uso de activación/desactivación de la conectividad, el AWS IoT FleetWise seguirá almacenando los datos en un búfer circular del vehículo cuando no haya conectividad. Cuando se reanuda la conectividad, el AWS IoT FleetWise carga los datos a la nube. Este comportamiento se puede configurar en el FleetWise software Edge Agent para AWS IoT.

> **Important**
>
> Si la partición de datos supera el límite máximo de almacenamiento establecido, los datos más recientes sobrescriben los más antiguos cuando la partición alcanza el tamaño máximo. Los datos perdidos en el Edge no se pueden recuperar. El tamaño de almacenamiento viene determinado por el límite de almacenamiento de Edge.
> Cuando los datos se cargan en la nube, se pueden eliminar una vez transcurrido el tiempo mínimo de vida. Establezca el tiempo mínimo de vida adecuado para evitar su eliminación involuntaria.

Las opciones de carga determinan las expresiones variables y condicionan el idioma. Si se especifican las opciones de carga, también debe especificar las opciones de almacenamiento. También puede solicitar que las señales de las particiones de datos se carguen en la nube. Para obtener más información, consulte Sube los datos de la campaña.

Una vez definidas las condiciones de partición de datos, `signalsToCollect` ayuda a especificar qué señales se deben tener en cuenta en la partición de datos. Puede especificar IDs las particiones de datos o configurarlas `dataPartitionId` `default` para que utilicen una partición de datos predeterminada establecida. Una señal sin una señal especificada se `dataPartitionId` asociará a la señal predeterminada`dataPartition`.

Para crear una partición de datos

Con el siguiente ejemplo, cree una campaña con una condición de almacenamiento de particiones de datos. Esta campaña de ejemplo está configurada para almacenar los datos del vehículo en Amazon Timestream.

1. *`campaign-name`*Sustitúyala por el nombre de la campaña que estás creando.
2. (Opcional) Proporciona una descripción.
3. *`role-arn`*Sustitúyalo por el nombre de recurso de Amazon (ARN) del rol de ejecución de tareas que otorga el FleetWise permiso de AWS IoT para entregar datos a la tabla Timestream.
4. *`table-arn`*Sustitúyalo por el ARN de la tabla Timestream.
5. *`signal-catalog-arn`*Sustitúyalo por el ARN del catálogo de señales.
6. *`data-partition-id`*Sustitúyalos por el `dataPartitions` ID y por el ID con el que se va a asociar`signalsToCollect`. En primer lugar, sustituya el ID de la partición de datos que

se va a utilizar en la señal. Pues`signalsToCollect`, el ID debe coincidir con uno de los IDs proporcionados`dataPartitions`.

> **Note**
>
> Establezca una partición de datos predeterminada para una campaña `default` utilizándola como ID.

7. *`target-arn`*Sustitúyalo por el ARN de una flota o vehículo que haya creado.

```
{
    "name": "campaign-name",
    "description": "Measurement of SOC, SOH, thermal, and power optimization for Fleet
 2704",
    "targetArn": "target-arn",
    "collectionScheme": {
        "conditionBasedCollectionScheme": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS` > 50",
            "minimumTriggerIntervalMs": 1000,
            "triggerMode": "ALWAYS"
        }
    },
    "compression": "SNAPPY",
    "dataDestinationConfigs": [{
        "timestreamConfig": {
            "executionRoleArn": "role-arn",
            "timestreamTableArn": "table-arn"
        }
    }],
    "dataPartitions": [{
        "id": "data-partition-id",
        "storageOptions": {
            "maximumSize": {
                "unit": "GB",
                "value": 1024
            },
            "minimumTimeToLive": {
                "unit": "WEEKS",
                "value": 6
            },
```

```
            "storageLocation": "string"
        },
        "uploadOptions": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS.PowerOptimization` > 90"
        }
    }],
    "signalCatalogArn": "signal-catalog-arn",
    "signalsToCollect": [{
        "dataPartitionId": "data-partition-id",
        "maxSampleCount": 50000,
        "minimumSamplingIntervalMs": 100,
        "name": "Below-90-percent"
    }],
    "spoolingMode": "TO_DISK",
    "tags": [{
        "Key": "BMS",
        "Value": "Under-90"
    }]
}
```

Tras cumplir todas las condiciones especificadas, los datos particionados se reenvían a la nube, lo que permite recopilar y almacenar nuevas señales particionadas.

A continuación, llamará a la `UpdateCampaign` API para implementarla en el FleetWise software Edge Agent para AWS IoT. Para obtener más información, consulte Sube los datos de la campaña.

Sube los datos de la campaña

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Hay dos formas de cargar los datos de la campaña en Edge:

- Las campañas que cumplan tus condiciones de carga subirán los datos automáticamente a la nube una vez que se aprueben. Para aprobar una campaña, usa la operación de la `updateCampaign` API.
- A través de AWS IoT Jobs, puedes forzar la carga de datos incluso cuando no se cumplan las condiciones especificadas. Para obtener más información, consulte Sube datos mediante AWS IoT Jobs.

Para cargar los datos de la campaña mediante la operación **`UpdateCampaign`** de la API

Una vez creada la campaña, el estado de la campaña se muestra como tal `WAITING_FOR_APPROVAL` hasta que cambies `action` a`APPROVED`.

- Usa el siguiente ejemplo para actualizar la campaña `action` recurriendo a la operación de la UpdateCampaignAPI.

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

Sube datos mediante AWS IoT Jobs

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Con AWS IoT Jobs, puedes configurar campañas para subir los datos de vehículos almacenados a la nube siempre que los necesites.

Para crear un documento de trabajo para tu campaña

- Utilice el siguiente ejemplo para crear un documento de trabajo para la campaña. Un documento de trabajo es un archivo.json que contiene información sobre los vehículos o las flotas

necesarios para realizar un trabajo. Para obtener más información sobre la creación de un documento de trabajo, consulte Crear y administrar trabajos mediante la AWS CLI Guía para AWS IoT desarrolladores.

Para solicitar que solo un vehículo cargue datos, defina el objetivo de la tarea en AWS IoT lo que esté asociado al vehículo. Para solicitar que varios vehículos (de la misma campaña) suban datos, crea un grupo de cosas con todas las cosas correspondientes a los vehículos y, a continuación, establece el objetivo de trabajo en ese grupo de cosas.

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

a. `CampaignArn`Sustitúyalo por el nombre de recurso de Amazon (ARN) de una campaña de la misma región y cuenta. Se requiere el ARN de la campaña.

b. (Opcional) `endTime` Sustitúyalo por la marca de tiempo de los datos recopilados en el vehículo en formato ISO 8601 UTC (sin milisegundos). Por ejemplo, `2024-03-05T23:00:00Z`. La marca de tiempo es exclusiva y determina el último punto de datos que se va a cargar. Si lo omite`endTime`, el software Edge Agent seguirá cargándose hasta que se carguen todos los datos almacenados en la campaña. Una vez cargados todos los datos, actualiza el estado de ejecución del trabajo a`SUCCEEDED`. El estado del trabajo se actualiza a`COMPLETED`.

Para crear un trabajo mediante una plantilla de trabajo gestionado

1. Elija IoT-IO TFleet Wise- CollectCampaignData de la lista de plantillas administradas. Para obtener más información, consulte Crear un trabajo a partir de plantillas AWS administradas en la AWS IoT Guía para desarrolladores.
2. La plantilla gestionada tiene los `endTime` parámetros `CampaignArn` y.

   a. `CampaignArn`Sustitúyalo por el nombre de recurso de Amazon (ARN) de una campaña de la misma región y cuenta. Se requiere el ARN de la campaña.

b. (Opcional) `endTime` Sustitúyalo por la marca de tiempo de los datos recopilados en el vehículo en formato ISO 8601 UTC (sin milisegundos). Por ejemplo, `2024-03-05T23:00:00Z`. La marca de tiempo es exclusiva y determina el último punto de datos que se va a cargar. Si lo omite`endTime`, el software Edge Agent seguirá cargándose hasta que se carguen todos los datos almacenados en la campaña. Una vez cargados todos los datos, actualiza el estado de ejecución del trabajo a`SUCCEEDED`. El estado del trabajo se actualiza a`COMPLETED`.

Para ver temas relacionados con la solución de problemas, consulteAlmacena y reenvía las incidencias.

Para obtener más información sobre los AWS IoT trabajos, consulte los trabajos en la guía para AWS IoT desarrolladores.

Recopile datos de códigos de problemas de diagnóstico mediante AWS IoT FleetWise

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Cuando un vehículo detecta un error, genera un código de diagnóstico de averías (DTC) y graba una instantánea de los sensores o actuadores afectados. DTCs le ayudan a conocer los errores casi en tiempo real, a entender su causa y a tomar medidas correctivas. AWS El IoT FleetWise admite la recopilación de DTCs, incluidas las correspondientes instantáneas de DTC y datos ampliados, a través de una campaña de recopilación de datos. En este tema se presentan los conceptos, los flujos de trabajo y las palabras clave que facilitan la recopilación de datos de DTC, ilustrados con ejemplos.

A continuación se muestran los conceptos clave para el uso de los DTC.

Funciones definidas a medida

Una función definida a medida es la capacidad de invocar y ejecutar sus propias funciones predefinidas en el Edge Agent, lo que amplía el concepto de decodificación personalizada. Estas

funciones se utilizan en coordinación con el FleetWise agente de AWS IoT. El FleetWise software Edge Agent para AWS IoT proporciona funciones integradas para calcular las estadísticas de señal, como los valores mínimo, máximo y promedio. Una función personalizada amplía esta capacidad al permitirle crear una lógica personalizada para casos de uso específicos. Para recopilar datos sobre los códigos de diagnóstico de problemas (DTC), los desarrolladores pueden aprovechar las funciones personalizadas para implementar mecanismos avanzados de recuperación de datos, como la obtención de códigos DTC, instantáneas y datos ampliados directamente desde la periferia del vehículo mediante los Servicios de Diagnóstico Unificado (UDS) o interfaces de diagnóstico alternativas.

Para obtener más información, consulte la guía de funciones personalizadas y la implementación de referencia sobre la recopilación de datos de DTC en la Guía para desarrolladores de Edge Agent.

Obtención de señales

En las campañas de recopilación de datos, las señales generalmente se recopilan de forma continua desde un dispositivo y se almacenan en búfer en el software Edge Agent. Luego, las señales se cargan o almacenan periódicamente en campañas basadas en el tiempo o se activan en función de condiciones específicas en las campañas basadas en las condiciones. Sin embargo, debido a la preocupación por la congestión del tráfico de los dispositivos, las señales DTC no se pueden recopilar de los dispositivos y almacenarlas en búfer de forma continua. Para solucionar este problema, el AWS IoT FleetWise proporciona la captación de señales, lo que garantiza que la señal objetivo se obtenga de forma discontinua desde un dispositivo.

La obtención de señales admite acciones tanto periódicas como condicionadas. Puede definir el método, las condiciones y las acciones exactas basadas en la búsqueda mediante funciones personalizadas definidas para cada señal que no se deben recopilar de un dispositivo de forma continua. En el caso de las señales gestionadas mediante el mecanismo de captación de señales, el tipo de activación y las condiciones de almacenamiento local o carga en la nube siguen rigiéndose por los dos `CollectionScheme` factores `timeBasedCollectionScheme` y `conditionBasedCollectionScheme` son compatibles, lo que equivale a lo que ocurre con las señales normales.

Los siguientes temas le muestran cómo puede crear y usar DTCs.

Temas

- Palabras clave de códigos de diagnóstico de problemas

- Cree una campaña de recopilación de datos para los códigos de diagnóstico de problemas
- Casos de uso de códigos de diagnóstico de problemas

Palabras clave de códigos de diagnóstico de problemas

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

`signalsToFetch`parámetro para crear una campaña

Utilice la signalsToFetchsintaxis para configurar cómo se puede obtener la información de la señal en el Edge. La obtención de señales estándar se controla mediante el modelado mediante reglas definidas explícitamente en un manifiesto del decodificador o definidas de forma personalizada mediante Edge First Modeling. Con las señales que recuperar, puedes definir cuándo y cómo se obtienen los datos durante las campañas.

Las señales que se van a buscar permiten recopilar información sobre los DTC. Por ejemplo, puede crear una señal denominada tipo cadena `DTC_Info` que contenga información sobre los DTC de cada unidad de control del motor (ECU). O bien, puede filtrar por una ECU específica.

- `SignalFetchInformation`definiciones de estructura y parámetros.

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

  - `fullyQualifiedName`: el nombre completo (FQDN) de la señal para la que desea utilizar la búsqueda personalizada.

- `signalFetchConfig`: define las reglas sobre cómo se deben obtener las señales definidas anteriormente. Admite la búsqueda basada en el tiempo y en la condición.
- `conditionLanguageVersion`: la versión del lenguaje condicional utilizada para analizar la expresión en la configuración.
- `actions`: una lista de todas las expresiones de acción evaluadas en Edge. The Edge obtendrá el valor de la señal definida.

> **Important**
>
> Las acciones solo se pueden usar`custom_function`.

Palabras clave de expresión de campaña

La siguiente expresión toma el nombre completo de una señal compatible con el vehículo y vuelve a ser verdadera si la señal no tiene ningún dato en los búferes de señal del Edge. Por otro lado, devuelve falso.

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

Example uso

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

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

Esta expresión toma la siguiente entrada:

Nombre de la función: cadena

El nombre de la función personalizada que admite Edge

parámetros: varargs ***Expression***

Parámetros para. `functionName` Puede ser cualquier lista de expresiones.

Los parámetros admiten tipos literales: String, Int, Boolean o Double.

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

Example uso

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

Cree una campaña de recopilación de datos para los códigos de diagnóstico de problemas

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En este tema se describe cómo crear una campaña de recopilación de datos para códigos de diagnóstico de problemas (DTC).

1. Defina una señal personalizada en el Edge. Debe definir las reglas de decodificación para la señal DTC en el Edge como una señal decodificada personalizada. Para obtener más información, consulte Tutorial: Configurar la recopilación de datos independiente de la red mediante una interfaz de decodificación personalizada.
2. Defina una función personalizada en el Edge. Debe definir una función personalizada para recopilar las señales DTC en el Edge en un momento compilado.

   Para obtener más información, consulte la guía de funciones personalizadas y la implementación de referencia sobre la recopilación de datos de DTC en la Guía para desarrolladores de Edge Agent.

> **Note**
>
> Un ejemplo de función definida personalizada es el `DTC_QUERY` que se muestra en el script de demostración.

3. Cree un catálogo de señales que modele una señal DTC como un tipo de cadena.

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

4. Cree y active un modelo de vehículo con la señal DTC añadida.
5. Cree y active un manifiesto del decodificador con la señal DTC añadida. La señal DTC debe ser del tipo de decodificador `CUSTOM_DECODING_SIGNAL` de señal con un `CUSTOM_DECODING_INTERFACE` tipo de interfaz de red.

   Example decodificador de señal

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

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

Example interfaz de red

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

> **Note**
>
> Las señales de la red de área del controlador (CAN) no admiten el tipo de datos de cadena.

6. Aprovisiona y crea vehículos. Los vehículos deben utilizar un modelo de vehículo (manifiesto de modelo) y un manifiesto de decodificador que se activaron en los pasos anteriores.
7. Crea y aprueba la campaña. Debe crear una campaña definiendo las señales DTC (opcionalmente, con señales de telemetría) e implementarlas en los vehículos.
8. Acceda a los datos en el destino definido. Los datos de los DTC incluyen `DTCCodeDTCSnapshot`, y `DTCExtendedDatastrings` como una cadena sin procesar en el destino de datos definido en la campaña.

Casos de uso de códigos de diagnóstico de problemas

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En los siguientes casos de uso, se supone que la `DTC_QUERY` función se definió en el script de demostración.

Búsqueda periódica

Obtenga una colección de DTC a intervalos configurados.

El siguiente ejemplo es una campaña en la que se obtienen señales periódicas `Vehicle.DTC_INFO` para todos DTCs con una máscara de estado para todos. ECUs Existe una condición para la recopilación de datos. `Vehicle.DTC_INFO`

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "timeBased": {
        // The FleetWise Edge Agent will query the UDS module for all DTCs every five
 seconds.
          "executionFrequencyMs": 5000
        }
      },
      "actions": [
      // Every five seconds, this action is called and its output is stored in the
      // signal history buffer of Vehicle.DTC_INFO
        "custom_function(\"DTC_QUERY\", -1, 2, -1)"
      ]
    }
  ],
  "signalsToCollect": [
    {
```

```
      "name": "Vehicle.ECU1.DTC_INFO"
    }
  ],
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      // Whenever a new DTC is filled into the signal, the data is ingested.
      "expression": "!isNull($variable.`Vehicle.ECU1.DTC_INFO`)",
      "minimumTriggerIntervalMs": 1000,
      // Make sure that data is ingested only when there are new DTCs.
      "triggerMode": "RISING_EDGE"
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Búsqueda basada en condiciones

Obtenga un conjunto de DTC cuando se cumpla una condición. Por ejemplo, cuando la señal CAN lo estéVehicle.Ignition == 1, busque y cargue los datos del DTC.

En la siguiente campaña de ejemplo, se obtiene una señal basada en una condición Vehicle.ECU1.DTC_INFO para comprobar si el DTC (» AAA123 «) está pendiente con el número de registro 1 para la ECU-1. Esta campaña recopila y carga de datos en función del tiempo.

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "conditionBased": {
```

```
        // The action will only run when the ignition is on.
          "conditionExpression": "$variable.`Vehicle.Ignition` == 1",
          "triggerMode": "ALWAYS"
        }
      },
      // The UDS module is only requested for the specific ECU address and the specific
 DTC Number/Status.
      "actions": ["custom_function(\"DTC_QUERY\", 1, 2, 8, \"0xAAA123\")"]
    }
  ],
  "signalsToCollect": [
    {
      "name": "Vehicle.ECU1.DTC_INFO"
    },
    {
      "name": "Vehicle.Ignition"
    }
  ],
  "collectionScheme": {
    "timeBasedCollectionScheme": {
      "periodMs": 10000
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Búsqueda bajo demanda

Busca un DTC específico para una flota.

Para un caso de uso bajo demanda, puedes usar la misma campaña que se define en la búsqueda periódica. El efecto bajo demanda se consigue suspendiendo la campaña poco después de desplegarla mediante la FleetWise consola de AWS IoT o ejecutando el siguiente comando CLI.

- *command-name*Sustitúyalo por el nombre del comando.

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

A continuación, suspenda la campaña cuando lleguen los datos del DTC.

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

Puedes reanudar la campaña de nuevo para obtener los datos de los DTC.

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

Visualice los datos de FleetWise vehículos de AWS IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

El FleetWise software Edge Agent para AWS IoT envía los datos del vehículo seleccionados a un tema de MQTT o los transfiere a Amazon Timestream o Amazon Simple Storage Service (Amazon S3). Una vez que los datos lleguen al destino de los datos, puede utilizar otros AWS servicios para procesarlos, redireccionarlos, visualizarlos y compartirlos.

> **Note**
>
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

Procesar los datos del vehículo enviados a un tema de MQTT

Los datos del vehículo enviados por la mensajería MQTT se envían prácticamente en tiempo real y te permiten utilizar las reglas para tomar medidas o enviar los datos a otros destinos. Para obtener más información sobre el uso de MQTT, consulte las reglas y protocolos de comunicación de los dispositivosAWS IoT en la Guía para AWS IoT Core desarrolladores.

El esquema de datos predeterminado que se envía en un mensaje MQTT 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 de tiempo 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.

Nombre del campo	Tipo de datos:	Descripción
`measure_value::varchar`	varchar	Valores de señal de tipo varchar.

Procesa 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 de tiempo 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.
`measure_value::varchar`	varchar	Valores de señal de tipo varchar.

Visualice 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 múltiples AWS fuentes (como Amazon CloudWatch y Timestream) y otras fuentes de datos con un único panel de Grafana.
- Analice y visualice los datos en los paneles de control mediante el uso de. QuickSight

Procese los datos del vehículo en Amazon S3

Amazon S3 es un servicio de almacenamiento de objetos que almacena y protege cualquier cantidad de datos. Puede usar S3 para diversos casos de uso, como lagos de datos, copias de seguridad y restauración, archivado, aplicaciones empresariales, AWS IoT dispositivos 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 de tiempo 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.
`measure_value_VARCHAR`	varchar	Valores de señal de tipo varchar.

Formato de objeto Amazon S3

AWS El 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 los vehículos que se transmiten al 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 añade un prefijo de hora UTC en el formato `year=YYYY/month=MM/date=DD/hour=HH` antes de escribir 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
```

Analice los datos del vehículo almacenados en Amazon 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 AI para flujos de trabajo posteriores de etiquetado y aprendizaje automático (ML).

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

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

Cataloga tus datos con Amazon Athena Rastreador de AWS Glue y analízalos. 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 AWS Glue para conectarse a fuentes de datos en Amazon S3
- Mejores prácticas al usar Athena con AWS Glue

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

> **Tip**
>
> Si estás leyendo directamente desde S3, confirma que los datos de tu vehículo estén en formato JSON, ya que QuickSight no son compatibles con el formato Apache Parquet.

Para obtener más información, consulta los siguientes temas de la Guía del QuickSight usuario de Amazon:

- Orígenes de datos admitidos

- Creación de un origen de datos

Comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.
> Esta documentación describe cómo utilizar la función de comandos para AWS IoT FleetWise. Para obtener información sobre el uso de la función de comandos en AWS IoT Device Management, consulte comandos.
> Usted es el único responsable de implementar los comandos de forma segura y que cumpla con las leyes aplicables. Para obtener más información sobre sus responsabilidades, consulte las condiciones AWS de servicio de los AWS IoT servicios.

Utilice la función de comandos para ejecutar comandos en un vehículo desde la nube. Los comandos se dirigen a un dispositivo a la vez y se pueden utilizar para aplicaciones de baja latencia y alto rendimiento, por ejemplo, para recuperar los registros del dispositivo o para iniciar un cambio de estado del dispositivo.

El comando es un recurso administrado por. AWS IoT Device Management Contiene configuraciones reutilizables que se aplican al enviar la ejecución de un comando al vehículo. Puede predefinir un conjunto de comandos para casos de uso específicos o usarlos para crear configuraciones reutilizables para casos de uso recurrentes. Por ejemplo, puedes configurar comandos que una aplicación puede utilizar para cerrar la puerta de un vehículo o para cambiar la temperatura de forma remota.

Con la función de AWS IoT comandos, puede:

- Cree un recurso de comandos y reutilice la configuración para enviar varios comandos al dispositivo de destino y, a continuación, ejecutarlos en el dispositivo.
- Controle la granularidad con la que desea que se ejecute cada comando en el dispositivo. Por ejemplo, puede aprovisionar un vehículo como una AWS IoT cosa y, a continuación, enviar una orden para bloquear o desbloquear las puertas del vehículo.
- Ejecuta varios comandos simultáneamente en el dispositivo objetivo sin esperar a que se complete el anterior.

- Seleccione activar las notificaciones de los eventos de comandos y recupere la información de estado y los resultados del dispositivo a medida que ejecuta el comando y una vez que se completa.

En los temas siguientes, se muestra cómo crear, enviar, recibir y administrar comandos.

Temas

- Conceptos de comandos
- Vehículos y comandos
- Creación y administración de comandos
- Iniciar y supervisar las ejecuciones de comandos
- Ejemplo: Uso de comandos para controlar el modo de dirección de un vehículo (AWS CLI)
- Escenarios de uso de comandos

Conceptos de comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Los comandos son instrucciones que se envían desde la nube al dispositivo de destino. El dispositivo objetivo puede ser un vehículo y debe estar registrado como una AWS IoT cosa en el registro de objetos. El comando puede contener parámetros que definen una acción que deben realizar los actuadores del vehículo. A continuación, el vehículo analiza el comando y sus parámetros y los procesa para realizar la acción correspondiente. A continuación, responde a la aplicación en la nube con el estado de la ejecución del comando.

Para ver el flujo de trabajo detallado, consulteVehículos y comandos.

Temas

- Comandos y conceptos clave
- Estado de ejecución del comando

Comandos y conceptos clave

A continuación, se muestran algunos conceptos clave sobre el uso de la función de comandos y su funcionamiento con las plantillas de estado del último estado conocido (LKS).

Comando

Un comando es una entidad que puedes usar para enviar instrucciones a un vehículo físico para que realice acciones como encender el motor o cambiar la posición de las ventanillas. Puedes predefinir un conjunto de comandos para casos de uso específicos o usarlos para crear configuraciones reutilizables para casos de uso recurrentes. Por ejemplo, puedes configurar comandos que una aplicación puede utilizar para cerrar la puerta de un vehículo o para cambiar la temperatura de forma remota.

Namespace

Al utilizar la función de comandos, debe especificar el espacio de nombres del comando. Cuando creas un comando en AWS IoT FleetWise, debes elegirlo `AWS-IoT-FleetWise` como espacio de nombres. Al utilizar este espacio de nombres, debe proporcionar los parámetros que se utilizarán para ejecutar el comando en el vehículo. Si desea crear un comando en AWS IoT Device Management su lugar, debe usar el espacio de `AWS-IoT` nombres en su lugar. Para obtener más información, consulta los comandos de la guía AWS IoT Device Management para desarrolladores.

Estados de comandos

Los comandos que cree estarán en un estado disponible, lo que significa que se pueden usar para iniciar la ejecución de un comando en el vehículo. Si un comando queda obsoleto, puedes dejarlo obsoleto. En el caso de un comando en estado obsoleto, las ejecuciones de los comandos existentes se ejecutarán hasta completarse. No puede actualizar el comando ni ejecutar ninguna ejecución nueva. Para enviar nuevas ejecuciones, debe restaurar el comando para que esté disponible.

También puede eliminar un comando si ya no es necesario. Al marcar un comando para su eliminación, si el comando ha quedado obsoleto durante un período superior al tiempo de espera máximo de 24 horas, el comando se eliminará inmediatamente. Si el comando no está obsoleto o ha estado en desuso durante un período inferior al tiempo de espera máximo, estará en un estado pendiente de eliminación. El comando se eliminará automáticamente de tu cuenta después de 24 horas.

Parámetros

Al crear un comando, si lo desea, puede especificar los parámetros que desea que ejecute el vehículo objetivo al ejecutar el comando. El comando que cree es una configuración reutilizable y puede usarse para enviar múltiples ejecuciones de comandos a su vehículo y ejecutarlas simultáneamente. Como alternativa, también puedes especificar los parámetros solo en tiempo de ejecución y elegir realizar una sola operación: crear un comando y enviarlo a tu vehículo.

Vehículo objetivo

Cuando quieras ejecutar el comando, debes especificar un vehículo objetivo que recibirá el comando y realizará acciones específicas. El vehículo objetivo ya debe estar registrado como un objeto AWS IoT. Cuando envíes el comando al vehículo, este empezará a ejecutar una instancia del comando en función de los parámetros y valores que hayas especificado.

Actuadores

Cuando desee ejecutar el comando, debe especificar los actuadores del vehículo que recibirá el comando y sus valores, que determinan las acciones que se van a realizar. Si lo desea, puede configurar los valores predeterminados de los actuadores para evitar el envío de comandos inexactos. Por ejemplo, puede utilizar el valor por defecto de un actuador de `LockDoor` cerradura de puerta para que el comando no desbloquee accidentalmente las puertas. Para obtener información general sobre los actuadores, consulteConceptos clave.

Soporte de tipos de datos

Los actuadores que se utilizan para la función de comandos admiten los siguientes tipos de datos.

> **Note**
>
> Las matrices no son compatibles con los datos telemáticos, los comandos ni el último estado conocido (LKS). Solo puede utilizar el tipo de datos de matriz para los datos de los sistemas de visión.

- Tipos de punto flotante. Se admiten los siguientes tipos:
  - Flotante (32 bits)
  - Doble (64 bits)
- Entero (con y sin signo). Se admiten los siguientes tipos de enteros.

    - int8 y uint8
    - int16 y uint16
    - int32 y uint32
- Largo. Se admiten los siguientes tipos largos.
    - Largo (int64)
    - Largo sin signo (uint64)
- Cadena
- Booleano

Ejecución de comandos

La ejecución de un comando es una instancia de un comando que se ejecuta en un dispositivo de destino. El vehículo ejecuta el comando utilizando los parámetros que especificó al crear el comando o al iniciar la ejecución del comando. A continuación, el vehículo realiza las operaciones especificadas y devuelve el estado de la ejecución.

> **Note**
>
> Para un vehículo determinado, puede ejecutar varios comandos al mismo tiempo. Para obtener información sobre el número máximo de ejecuciones simultáneas que puedes ejecutar para cada vehículo, consulta las cuotas de AWS IoT Device Management comandos.

Plantillas de estado del último estado conocido (LKS)

Las plantillas de estados proporcionan un mecanismo para que los propietarios de vehículos rastreen el estado de su vehículo. Para monitorizar el último estado conocido (LKS) de tus vehículos prácticamente en tiempo real, puedes crear plantillas de estado y asociarlas a tus vehículos.

Con la función de comandos, puede realizar operaciones «bajo demanda» que se pueden utilizar para recopilar y procesar datos de estado. Por ejemplo, puedes solicitar el estado actual del vehículo una sola vez (recuperar), o activar o desactivar las plantillas de estado LKS desplegadas anteriormente para iniciar o dejar de informar sobre los datos del vehículo. Para ver ejemplos que muestran cómo utilizar comandos con plantillas de estado, consulte. Escenarios de uso de comandos

Estado de ejecución del comando

Tras iniciar la ejecución del comando, el vehículo puede publicar el estado de la ejecución y proporcionar los motivos del estado como información adicional sobre la ejecución. En las siguientes secciones se describen los distintos estados de ejecución de los comandos y los códigos de estado.

Temas

- Estado de ejecución del comando, motivo, código y descripción
- Estado de ejecución de comandos y códigos de estado
- Estado de tiempo de espera de ejecución del comando

Estado de ejecución del comando, motivo, código y descripción

Para informar sobre las actualizaciones del estado de ejecución de los comandos, sus vehículos pueden utilizar la `UpdateCommandExecution` API para publicar la información de estado actualizada en la nube, utilizando los temas reservados a los comandos que se describen en la guía para AWS IoT Core desarrolladores. Al enviar la información de estado, sus dispositivos pueden proporcionar un contexto adicional sobre el estado de cada ejecución de comandos mediante el `StatusReason` objeto `reasonCode` y `reasonDescription` los campos contenidos en el objeto.

Estado de ejecución de comandos y códigos de estado

La siguiente tabla muestra los distintos códigos de estado de ejecución de comandos y los estados permitidos a los que puede pasar la ejecución de un comando. También muestra si la ejecución de un comando es «terminal» (es decir, si no habrá más actualizaciones de estado), si el cambio lo ha iniciado el vehículo o la nube, y los diferentes códigos de estado predefinidos y cómo se corresponden con los estados que informa la nube.

- Para obtener información sobre cómo se AWS IoT FleetWise utilizan los códigos de estado predefinidos y el `statusReason` objeto, consulte el estado del comando en la documentación del FleetWise software Edge Agent para AWS IoT.
- Para obtener información adicional sobre las ejecuciones terminales y no terminales y las transiciones entre los estados, consulte el estado de ejecución de los comandos en la guía para AWS IoT Core desarrolladores.

Estado y fuente de la ejecución de los comandos

Estado de ejecución del comando	Descripción	¿Iniciado por el dispositivo o la nube?	¿Ejecución terminal?	Transiciones de estado permitidas	Códigos de estado predefinidos
`CREATED`	Cuando la solicitud de la API para iniciar la ejecución del comando (`StartCommandExecution` API) se realiza correctamente, el estado de ejecución del comando cambia a`CREATED`.	Cloud	No	• `IN_PROGRESS` • `SUCCEEDED` • `FAILED` • `REJECTED` • `TIMED_OUT`	Ninguno
`IN_PROGRESS`	Cuando el vehículo comience a ejecutar el comando, podrá publicar un mensaje en el tema de respuesta para actualizar el	Dispositivo	No	• IN_PROGRESS • SUCCEEDED • ERROR • REJECTED • TIMED_OUT	`COMMAND_STATUS_COMMAND_IN_PROGRESS`

Estado de ejecución del comando	Descripción	¿Iniciado por el dispositivo o la nube?	¿Ejecución terminal?	Transiciones de estado permitidas	Códigos de estado predefinidos
	estadoIN_PROCESS .				
SUCCEEDED	Cuando el vehículo haya procesado correctamente el comando y completado la ejecución, puede publicar un mensaje en el tema de respuesta para actualizar el estadoSUCCEEDED.	Dispositivo	Sí	No aplicable	COMMAND_STATUS_SUCCEEDED
FAILED	Si el vehículo no ha ejecutado el comando, puede publicar un mensaje en el tema de respuesta para actualizar el estadoFAILED.	Dispositivo	Sí	No aplicable	COMMAND_STATUS_EXECUTION_FAILED

Estado de ejecución del comando	Descripción	¿Iniciado por el dispositivo o la nube?	¿Ejecución terminal?	Transiciones de estado permitidas	Códigos de estado predefinidos
`REJECTED`	Si el vehículo no acepta la orden, puede publicar un mensaje en el tema de respuesta para actualizar el estado`REJECTE`	Dispositivo	Sí	No aplicable	Ninguno

Estado de ejecución del comando	Descripción	¿Iniciado por el dispositivo o la nube?	¿Ejecución terminal?	Transiciones de estado permitidas	Códigos de estado predefinidos
`TIMED_OUT`	El estado de ejecución del comando puede cambiar a `TIMED_OUT` uno debido a cualquiera de los siguientes motivos. • No se recibió el resultado de la ejecución del comando y la nube informa automáticamente `TIMED_OUT` del estado. • El vehículo informa de que se ha agotado el tiempo de espera al intentar ejecutar el	Dispositivo y nube	No	• SUCCEEDED • ERROR • REJECTED • TIMED_OUT	`COMMAND_STATUS_EXECUTION_TIMEOUT`

Estado de ejecución del comando	Descripción	¿Iniciado por el dispositivo o la nube?	¿Ejecución terminal?	Transiciones de estado permitidas	Códigos de estado predefinidos
	comando. En este caso, la ejecución del comando pasa a ser terminal. Para obtener más información sobre este estado, consulteEstado de tiempo de espera de ejecución del comando.				

Estado de tiempo de espera de ejecución del comando

Tanto la nube como el dispositivo pueden informar sobre el tiempo de espera de la ejecución de un comando. Una vez que se envía el comando al dispositivo, se inicia un temporizador. Si no se recibió ninguna respuesta del dispositivo dentro del período especificado, la nube informa del `TIMED_OUT` estado. En este caso, la ejecución del comando en `TIMED_OUT` estado no es terminal.

El dispositivo puede sustituir este estado por un estado de terminal, como `SUCCEEDEDFAILED`, o. `REJECTED` También puede informar de que se ha agotado el tiempo de espera al ejecutar el comando. En este caso, el estado de ejecución del comando permanece en, `TIMED_OUT` pero los campos del `StatusReason` objeto se actualizan en función de la información proporcionada por el dispositivo. La ejecución del comando en ese `TIMED_OUT` estado pasa ahora a ser terminal.

Para obtener información adicional, consulte Consideraciones sobre el tiempo de espera de la ejecución de comandos en la guía para AWS IoT Core desarrolladores.

Vehículos y comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.
> Usted es el único responsable de implementar los comandos de forma segura y que cumpla con las leyes aplicables.

Para usar la función de comandos:

1. En primer lugar, cree un recurso de comandos. Si lo desea, especifique los parámetros que contienen la información necesaria para ejecutar el comando.
2. Especifique el vehículo objetivo que recibirá el comando y realizará las acciones especificadas.
3. Ahora puedes ejecutar el comando en el dispositivo objetivo y comprobar los detalles de ejecución del comando para recuperar el estado y utilizar CloudWatch los registros para seguir solucionando cualquier problema.

En las siguientes secciones, se muestra el flujo de trabajo entre los vehículos y los comandos.

Temas

- Información general sobre el flujo de trabajo
- Flujo de trabajo de
- flujo de trabajo de comandos
- (Opcional) Comandos y notificaciones

Información general sobre el flujo de trabajo

Los siguientes pasos proporcionan una visión general del flujo de trabajo de comandos entre sus vehículos y los comandos. Cuando utilizas cualquiera de los comandos de las operaciones de la API HTTP, la solicitud se firma con las credenciales Sigv4.

> **Note**
>
> A excepción de la operación de `StartCommandExecution` API, todas las operaciones que se realizan a través del protocolo HTTP utilizan el punto final del plano de control.

1. Establezca una conexión MQTT y suscríbase a los temas de comandos

   Para prepararse para el flujo de trabajo de los comandos, los dispositivos deben establecer una conexión MQTT con el `iot:Data-ATS` terminal y suscribirse al tema de solicitud de comandos mencionado anteriormente. De forma opcional, sus dispositivos también pueden suscribirse a los temas de respuesta sobre los comandos aceptados y rechazados.
2. Cree un modelo de vehículo y un recurso de mando

   Ahora puede crear un vehículo y un recurso de comando mediante las operaciones de la API `CreateVehicle` y del plano de `CreateCommand` control. El recurso de comando contiene las configuraciones que se aplicarán cuando el comando se ejecute en el vehículo.
3. Inicie la ejecución del comando en el dispositivo de destino

   Inicie la ejecución del comando en el vehículo mediante la API del plano de `StartCommandExecution` datos con el punto final específico de su cuenta`iot:Jobs`. La API publica un mensaje de carga útil codificado por protobug en el tema de solicitud de comandos.
4. Actualiza el resultado de la ejecución del comando

   El vehículo procesa el comando y la carga útil recibida y, a continuación, publica el resultado de la ejecución del comando en el tema de respuesta mediante la `UpdateCommandExecution` API. Si tu vehículo se ha suscrito a los temas de respuesta con los comandos de aceptación y rechazo, recibirá un mensaje en el que se indicará si el servicio en la nube ha aceptado o rechazado la respuesta.
5. (Opcional) Recupera el resultado de la ejecución del comando

   Para recuperar el resultado de la ejecución del comando, puede utilizar la operación API del plano de `GetCommandExecution` control. Una vez que tu vehículo publique el resultado de la ejecución del comando en el tema de respuesta, esta API devolverá la información actualizada.
6. (Opcional) Suscríbase y gestione los eventos de comandos

   Para recibir notificaciones sobre las actualizaciones del estado de ejecución de los comandos, puede suscribirse al tema de eventos de comandos. A continuación, puede utilizar la API del

plano de `CreateTopicRule` control para enrutar los datos de eventos de comandos a otras aplicaciones, como AWS Lambda funciones o Amazon SQS, y crear aplicaciones a partir de ellos.

Flujo de trabajo de

Los siguientes pasos describen en detalle el flujo de trabajo del vehículo cuando se utiliza la función de comandos.

> **Note**
>
> Las operaciones que se describen en esta sección utilizan el protocolo MQTT.

1. Establezca una conexión MQTT

   Para preparar sus vehículos para usar la función de comandos, primero deben conectarse al intermediario de AWS IoT Core mensajes. Su vehículo debe poder realizar la `iot:Connect` acción necesaria para conectarse AWS IoT Core y establecer una conexión MQTT con el intermediario de mensajes. Para encontrar el punto final del plano de datos para usted Cuenta de AWS, utilice la `DescribeEndpoint` API o el comando `describe-endpoint` CLI como se muestra a continuación.

   ```
   aws iot describe-endpoint --endpoint-type iot:Data-ATS
   ```

   Al ejecutar este comando, se devuelve el punto final del plano de datos específico de la cuenta, como se muestra a continuación.

   ```
   account-specific-prefix.iot.region.amazonaws.com
   ```

2. Suscríbase al tema de solicitud de comandos

   Una vez establecida la conexión, sus dispositivos pueden suscribirse al tema de solicitud de AWS IoT comandos en MQTT. Al crear un comando e iniciar la ejecución del comando en el dispositivo de destino, el intermediario de mensajes publicará un mensaje de carga útil codificado en protobuf en el tema de solicitud. A continuación, el dispositivo podrá recibir el mensaje de carga útil y procesar el comando. En este ejemplo, *<DeviceID>* sustitúyalo por el

identificador único del vehículo objetivo. Este identificador puede ser el identificador único de tu vehículo o el nombre de una cosa

> **Note**
>
> El mensaje de carga que se envía al dispositivo debe usar el formato protobuf.

```
$aws/commands/things/<DeviceID>/executions/+/request/protobuf
```

3. (Opcional) Suscríbase a los temas de respuesta a los comandos

   Si lo desea, puede suscribirse a estos temas de respuesta a los comandos para recibir un mensaje que indique si el servicio en la nube ha aceptado o rechazado la respuesta del dispositivo.

   > **Note**
   >
   > Es opcional que sus vehículos se suscriban a los temas `/accepted` y a los temas de `/rejected` respuesta. Tus vehículos recibirán automáticamente estos mensajes de respuesta, incluso si no se han suscrito explícitamente a estos temas.

   ```
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/accepted
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/rejected
   ```

4. Actualiza el resultado de la ejecución de un comando

   A continuación, el vehículo objetivo procesa el comando. A continuación, utiliza la `UpdateCommandExecution` API para publicar el resultado de la ejecución en el siguiente tema de respuesta de MQTT.

   > **Note**
   >
   > Para ejecutar un vehículo y un comando determinados, *<DeviceID>* debe coincidir con el campo correspondiente del tema de solicitud al que se haya suscrito el dispositivo.

```
$aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf
```

La `UpdateCommandExecution` API es una operación de API de plano de datos a través de MQTT que se autentica con TLS.

- Si el servicio en la nube procesó correctamente el resultado de la ejecución del comando, se publicará un mensaje en el tema aceptado por MQTT. El tema aceptado utiliza el siguiente formato.

  ```
  $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
  accepted
  ```

- Si el servicio en la nube no ha podido procesar el resultado de la ejecución del comando, se publica una respuesta al tema de MQTT rechazado. El tema rechazado utiliza el siguiente formato.

  ```
  $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
  rejected
  ```

Para obtener más información sobre esta API y un ejemplo, consulteActualiza el resultado de la ejecución del comando.

flujo de trabajo de comandos

En los pasos siguientes se describe en detalle el flujo de trabajo de los comandos.

> **Note**
>
> Las operaciones que se describen en esta sección utilizan el protocolo HTTP.

1. Registre su vehículo

   Ahora que ha preparado su vehículo para usar la función de comandos, puede preparar su solicitud registrando su vehículo y, a continuación, creando un comando que se enviará al vehículo. Para registrar el vehículo, cree una instancia de un modelo de vehículo (manifiesto

del modelo) mediante la operación API del plano de `CreateVehicle`control. Para obtener más información y ejemplos, consulte Crear un vehículo.

2. Cree un comando

   Usa la operación de la API del plano de control `CreateCommand`HTTP para modelar los comandos que se apliquen al vehículo al que te diriges. Especifique los parámetros y valores predeterminados que se utilizarán al ejecutar el comando y asegúrese de que utilice el espacio de `AWS-IoT-FleetWise` nombres. Para obtener más información y ejemplos sobre el uso de esta API, consulte. Cree un recurso de comandos

3. Inicie la ejecución del comando

   Ahora puede ejecutar el comando que creó en el vehículo mediante la operación API del plano de `StartCommandExecution`datos. AWS IoT Device Management obtiene el comando y los parámetros del comando y valida la solicitud entrante. A continuación, invoca la AWS IoT FleetWise API con los parámetros necesarios para generar la carga útil específica del vehículo. Luego, la carga útil se envía al dispositivo mediante MQTT al tema AWS IoT Device Management de solicitud de comando al que está suscrito el dispositivo. Para obtener más información y ejemplos sobre el uso de esta API, consulte. Envía un comando ()AWS CLI

   ```
   $aws/commands/things/<DeviceID>/executions/+/request/protobuf
   ```

   > **Note**
   >
   > Si el dispositivo estaba desconectado cuando se envió el comando desde la nube y se estaban utilizando sesiones persistentes de MQTT, el comando espera en el intermediario de mensajes. Si el dispositivo vuelve a conectarse antes de que se agote el tiempo de espera y se ha suscrito al tema de solicitud de comandos, el dispositivo podrá procesar el comando y publicar el resultado en el tema de respuesta. Si el dispositivo no vuelve a conectarse antes de que se agote el tiempo de espera, se agotará el tiempo de espera de la ejecución del comando y el mensaje de carga caducará.

4. Recupera la ejecución del comando

   Tras ejecutar el comando en el dispositivo, utilice la operación de la API del plano de `GetCommandExecution`control para recuperar y supervisar el resultado de la ejecución del comando. También puede usar la API para obtener información adicional sobre los datos de

ejecución, como cuándo se actualizaron por última vez, cuándo se completó la ejecución y los parámetros especificados.

> **Note**
>
> Para recuperar la información de estado más reciente, el dispositivo debe haber publicado el resultado de la ejecución del comando en el tema de respuesta.

Para obtener más información y ejemplos sobre el uso de esta API, consulteObtenga la ejecución de comandos.

(Opcional) Comandos y notificaciones

Puede suscribirse a los eventos de comandos para recibir notificaciones cuando cambie el estado de la ejecución de un comando. En los pasos siguientes se muestra cómo suscribirse a los eventos de comandos y, a continuación, procesarlos.

1. Creación de una regla del tema

   Puede suscribirse al tema de eventos de comandos y recibir notificaciones cuando cambie el estado de la ejecución de un comando. También puede crear una regla temática para dirigir los datos procesados por el vehículo a otras aplicaciones, como AWS Lambda las funciones. Puede crear una regla temática mediante la AWS IoT consola o la operación de la API del plano de `CreateTopicRule` AWS IoT Core control. Para obtener más información, consulte Creación de una AWS IoT regla.

   En este ejemplo, *`<CommandID>`* sustitúyalo por el identificador del comando del que quieres recibir notificaciones y *`<CommandExecutionStatus>`* por el estado de la ejecución del comando.

   ```
   $aws/events/commandExecution/<CommandID>/<CommandExecutionStatus>
   ```

   > **Note**
   >
   > Para recibir notificaciones de todos los comandos y estados de ejecución de los comandos, puede utilizar caracteres comodín y suscribirse al tema siguiente.

```
$aws/events/commandExecution/+/#
```

2. Reciba y procese los eventos de comandos

   Si creaste una regla temática en el paso anterior para suscribirte a los eventos de comandos, puedes administrar las notificaciones push de comandos que recibas. Si lo desea, también puede crear aplicaciones sobre ella, por ejemplo, con Amazon SQS AWS Lambda, Amazon SNS o Step AWS Functions mediante la regla de tema que ha creado.

El siguiente código muestra un ejemplo de carga útil para las notificaciones de eventos de comandos que recibirá.

```
{
    "executionId": "2bd65c51-4cfd-49e4-9310-d5cbfdbc8554",
    "status":"FAILED",
    "statusReason": {
         "reasonCode": "4",
         "reasonDescription": ""
    },
    "eventType": "COMMAND_EXECUTION",
    "commandArn":"arn:aws:iot:us-east-1:123456789012:command/0b9d9ddf-
e873-43a9-8e2c-9fe004a90086",
    "targetArn":"arn:aws:iot:us-east-1:123456789012:thing/5006c3fc-
de96-4def-8427-7eee36c6f2bd",
    "timestamp":1717708862107
}
```

Creación y administración de comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puedes configurar acciones remotas reutilizables o enviar instrucciones únicas e inmediatas a tus dispositivos. Al utilizar esta función, puede especificar las instrucciones que sus dispositivos pueden ejecutar prácticamente en tiempo real. Un comando te permite configurar acciones remotas reutilizables para el vehículo objetivo. Tras crear un comando, puedes iniciar la ejecución de un comando dirigido a un vehículo específico.

En este tema se muestra cómo puede crear y administrar un recurso de comando mediante la AWS IoT Core API o el AWS CLI. En él se muestra cómo realizar las siguientes acciones en un recurso de comandos.

Temas

- Cree un recurso de comandos
- Recupera información sobre un comando
- Enumere los comandos de su cuenta
- Actualiza o desactiva un recurso de comando
- Eliminar un recurso de comandos

Cree un recurso de comandos

Puede usar la operación de la API del plano de `CreateCommand` AWS IoT Core control o la FleetWise consola de AWS IoT para crear un comando.

Cree un comando (consola)

Puede usar la FleetWise consola de AWS IoT para crear un comando.

Para crear un comando

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, selecciona Comandos.
3. Elija el comando Crear.
4. Especifique un identificador de comando único que le ayude a identificar el comando que desea ejecutar en el vehículo.
5. (Opcional) Especifique un nombre y una descripción opcionales para mostrar.
6. (Opcional) Seleccione el actuador y el valor del parámetro por defecto. Los parámetros especifican las acciones que el vehículo objetivo puede realizar al recibir el comando. Si no añades parámetros, tendrás que proporcionarlos al ejecutar el comando.

7. Elija una función de IAM que conceda permisos para generar la carga útil de los comandos. Consulte Controlar el acceso.
8. Seleccione el comando Crear.

Cree un comando (AWS CLI)

El siguiente ejemplo muestra cómo crear un comando con un parámetro.

Consideraciones a la hora de crear un comando

Al crear un comando en AWS IoT FleetWise:

- Debe especificar el `roleArn` que otorga el permiso para crear y ejecutar comandos en su vehículo. Para obtener más información y ejemplos de políticas, incluidos los casos en que las claves KMS están habilitadas, consulteConceda AWS IoT Device Management permiso para generar la carga útil de los comandos con AWS IoT FleetWise.
- Debe especificarlo `AWS-IoT-FleetWise` como espacio de nombres.
- Puede omitir el `mandatory-parameters` campo y, en su lugar, especificarlos en tiempo de ejecución. Como alternativa, puede crear un comando con parámetros y, si lo desea, especificar valores por defecto para ellos. Si especificó valores por defecto, en tiempo de ejecución, puede utilizarlos o anularlos especificando sus propios valores. Para ver estos ejemplos adicionales, consulteEscenarios de uso de comandos.
- Puede especificar hasta tres pares de nombre-valor para el `mandatory-parameters` campo. Sin embargo, al ejecutar el comando en el vehículo, solo se acepta un par nombre-valor y el `name` campo debe usar el nombre completo con el prefijo. `$actuatorPath.`

- *`command-id`*Sustitúyalo por un identificador único para el comando. Puede usar el UUID, los caracteres alfanuméricos, «-» y «_».
- *`role-arn`*Sustitúyalo por el rol de IAM que te otorga permiso para crear y ejecutar comandos, por ejemplo,. `"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"`
- (Opcional) *`display-name`* Sustitúyalo por un nombre fácil de usar para el comando y *`description`* por una descripción significativa del comando.
- Sustituya *`name`* y *`value`* del `mandatory-parameters` objeto por la información necesaria para el comando que se va a crear. El `name` campo es el nombre completo, tal como se define en el catálogo de señales, con `$actuatorPath.` el prefijo. Por ejemplo, `name` puede ser *`$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode`*

y `value` puede ser un booleano que indica el estado de un modo de dirección similar. *`{"B": false}`*

```
aws iot create-command --command-id command-id \
    --role-arn role-arn \
    --description description \
    --display-name display-name \
    --namespace "AWS-IoT-FleetWise" \
    --mandatory-parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

La operación de la `CreateCommand` API devuelve una respuesta que contiene el ID y el ARN (Amazon Resource Name) del comando.

```
{
    "commandId": "HandsOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"
}
```

Recupera información sobre un comando

Puede utilizar la operación de la API del plano de `GetCommand` AWS IoT Core control para recuperar información sobre un recurso de comando.

Para obtener información sobre un recurso de comando, ejecute el siguiente comando. *`command-id`*Sustitúyalo por el identificador que se utilizó al crear el comando.

```
aws iot get-command --command-id command-id
```

La operación de la `GetCommand` API devuelve una respuesta que contiene la siguiente información.

- El ID y el ARN (Amazon Resource Name) del comando.
- La fecha y la hora en que se creó el comando y se actualizó por última vez.
- El estado del comando, que indica si está disponible para ejecutarse en el vehículo.
- Todos los parámetros que especificó al crear el comando.

```
{
    "commandId": "HandsOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode",
            "value": {"B": false }
        }
    ],
    "createdAt": "2024-03-23T11:24:14.919000-07:00",
    "lastUpdatedAt": "2024-03-23T11:24:14.919000-07:00",
    "deprecated": false,
    "pendingDeletion": false
}
```

Enumere los comandos de su cuenta

Puede utilizar la operación de la API del plano de `ListCommands` AWS IoT Core control para enumerar todos los comandos de la cuenta que haya creado.

Para enumerar los comandos de tu cuenta, ejecuta el siguiente comando. De forma predeterminada, la API devuelve los comandos que se crearon para ambos espacios de nombres. Para filtrar la lista y mostrar solo los comandos para los que se crearon AWS IoT FleetWise, ejecuta el siguiente comando.

> **Note**
>
> También puede ordenar la lista en orden ascendente o descendente, o bien filtrarla para que muestre solo los comandos que tengan un nombre de parámetro de comando específico.

```
aws iot list-commands --namespace "AWS-IoT-FleetWise"
```

La operación `ListCommands` de la API devuelve una respuesta que contiene la siguiente información.

- El ID y el ARN (Amazon Resource Name) de los comandos.

- La fecha y la hora en que se creó el comando y se actualizó por última vez.
- El estado del comando, que indica si los comandos están disponibles para ejecutarse en el vehículo.

Actualiza o desactiva un recurso de comando

Puede usar la operación de la API del plano de `UpdateCommand` AWS IoT Core control o la FleetWise consola de AWS IoT para actualizar un recurso de comando. Puede actualizar el nombre mostrado y la descripción de un comando. También puedes dejar de usar un comando si no se está utilizando actualmente.

> **Note**
>
> No puede modificar la información del espacio de nombres ni los parámetros que se van a utilizar al ejecutar el comando.

Actualizar un comando (consola)

Actualizar un comando

Para actualizar un comando desde la consola, vaya a la página Comandos de la FleetWise consola de AWS IoT y lleve a cabo los siguientes pasos.

1. Elija el comando que desee actualizar y, a continuación, elija Editar.
2. Edite los detalles del comando y, a continuación, elija Guardar cambios.

Desactivar un comando

Para anular un comando de la consola, vaya a la página Comandos de la FleetWise consola de AWS IoT y lleve a cabo los siguientes pasos.

1. Elija el comando que desee desaprobar y, a continuación, elija Deprecar.
2. Confirme la obsolescencia y, a continuación, elija Deprecatar.

Actualizar un comando ()AWS CLI

Actualizar un comando

Para actualizar un recurso de comandos, ejecute el siguiente comando. *command-id*Sustitúyalo por el identificador del comando que desee actualizar y proporcione el *display-name* y actualizado*description*.

```
aws iot update-command \
    --command-id command-id \
    --display-name display-name \
    --description description
```

La operación UpdateCommand de la API devuelve la siguiente respuesta.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": false,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Desactivar un comando

Se desaprueba un comando cuando se pretende dejar de usarlo en el dispositivo o cuando está desactualizado. En el siguiente ejemplo, se muestra cómo desaprobar un comando.

```
aws iot update-command \
    --command-id command-id \
    --deprecated
```

La operación de la UpdateCommand API devuelve una respuesta que contiene el ID y el ARN (Amazon Resource Name) del comando.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": true,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Una vez que un comando ha quedado obsoleto, las ejecuciones de comandos existentes seguirán ejecutándose en el vehículo hasta que pasen a ser inútiles. Para ejecutar cualquier comando nuevo, debes usar la UpdateCommand API para restaurar el comando de forma que esté disponible. Para obtener información adicional sobre la obsolescencia y la restauración de un comando y las

consideraciones al respecto, consulte Cómo desaprobar un recurso de comandos en la AWS IoT Core Guía para desarrolladores.

Eliminar un recurso de comandos

Puede usar la operación de la API del plano de `DeleteCommand` AWS IoT Core control o la FleetWise consola de AWS IoT para eliminar un recurso de comando.

> **Note**
>
> Las acciones de eliminación son permanentes y no se pueden deshacer. El comando se eliminará permanentemente de tu cuenta.

Eliminar un comando (consola)

Para eliminar un comando de la consola, vaya a la página Comandos de la FleetWise consola de AWS IoT y lleve a cabo los siguientes pasos.

1. Elija el comando que desee eliminar y, a continuación, elija Eliminar.
2. Confirme que desea eliminar el comando y, a continuación, elija Eliminar.

Eliminar un comando (AWS CLI)

Para eliminar un recurso de comando, ejecute el siguiente comando. *command-id*Sustitúyalo por el identificador del comando que desee eliminar. En el siguiente ejemplo, se muestra cómo eliminar un recurso de comando.

```
aws iot delete-command --command-id command-id
```

Si la solicitud de eliminación se ha realizado correctamente:

- Si el comando ha quedado obsoleto durante un período superior al tiempo de espera máximo de 24 horas, se eliminará inmediatamente y aparecerá un HTTP `statusCode` de 204.
- Si el comando no está en desuso o ha estado en desuso durante un tiempo inferior al tiempo de espera máximo, el comando estará en un `pending deletion` estado y aparecerá un HTTP de 202. `statusCode` El comando se eliminará automáticamente de tu cuenta una vez transcurrido el tiempo de espera máximo de 24 horas.

Iniciar y supervisar las ejecuciones de comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Una vez que hayas creado un recurso de comando, puedes iniciar la ejecución de un comando en el vehículo objetivo. Una vez que el vehículo comience a ejecutar el comando, podrá empezar a actualizar el resultado de la ejecución del comando y publicar las actualizaciones de estado y la información sobre los resultados en los temas reservados de MQTT. A continuación, podrá recuperar el estado de la ejecución del comando y supervisar el estado de las ejecuciones en su cuenta.

En este tema se muestra cómo puede enviar un comando a su vehículo mediante la AWS CLI FleetWise consola de AWS IoT. También se muestra cómo supervisar y actualizar el estado de la ejecución del comando.

Temas

- Actualiza el resultado de la ejecución del comando
- Obtenga la ejecución de comandos
- Enumere las ejecuciones de comandos en su cuenta
- Eliminar la ejecución de un comando

Enviar un comando (consola)

Para enviar un comando desde la consola, vaya a la página Vehículos de la FleetWise consola AWS IoT y lleve a cabo los siguientes pasos.

1. Elige el vehículo al que quieres enviar un comando.
2. Elija Run command (Ejecutar comando).
3. Selecciona el ID del comando.
4. Especifique el tiempo de espera de ejecución del comando y, a continuación, elija Ejecutar comando.

Envía un comando ()AWS CLI

Puede utilizar la operación de la API del plano de `StartCommandExecution` AWS IoT datos para enviar un comando a un vehículo. A continuación, el vehículo reenvía el comando a un servicio de software intermedio para automoción (por ejemplo SOME/IP , software intermedio escalable orientado a servicios sobre IP) o lo publica en la red de un vehículo (como la interfaz de un dispositivo de red de área de controlador (CAN)). El siguiente ejemplo utiliza AWS CLI.

Temas

- Consideraciones a la hora de enviar un comando
- Obtenga el punto final del plano de datos específico de la cuenta
- Envía un ejemplo de comando

Consideraciones a la hora de enviar un comando

Al iniciar la ejecución de un comando en AWS IoT FleetWise:

- Debe aprovisionar cualquier AWS IoT cosa para el vehículo. Para obtener más información, consulte Aprovisionamiento AWS de FleetWise vehículos de IoT.
- Debe haber creado ya un comando `AWS-IoT-FleetWise` como espacio de nombres y haber proporcionado uno `role-Arn` que le otorgue permiso para crear y ejecutar comandos en el IoT AWS . FleetWise Para obtener más información, consulte Cree un recurso de comandos.
- Puede omitir el `parameters` campo si decide usar cualquier valor predeterminado que se especificó para los parámetros al crear el comando. Si `mandatory-parameters` no se especificó en el momento de la creación, o si desea anular cualquier valor predeterminado especificando sus propios valores para los parámetros, debe especificar el `parameters` campo. Para ver estos ejemplos adicionales, consulteEscenarios de uso de comandos.
- Puede especificar hasta tres pares de nombre-valor para el `mandatory-parameters` campo. Sin embargo, al ejecutar el comando en el vehículo, solo se acepta un par nombre-valor y el `name` campo debe usar el nombre completo con el prefijo. `$actuatorPath.`

Obtenga el punto final del plano de datos específico de la cuenta

Antes de ejecutar el comando de la API, debe obtener la URL del punto final específica de la cuenta para el punto final. `iot:Jobs` Por ejemplo, si ejecuta este comando:

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

Devolverá la URL del punto final específica de la cuenta, tal y como se muestra en el ejemplo de respuesta que aparece a continuación.

```
{
    "endpointAddress": "<account-specific-prefix>.jobs.iot.<region>.amazonaws.com"
}
```

Envía un ejemplo de comando

Para enviar un comando a un vehículo, ejecute el siguiente comando.

- *command-arn*Sustituya el comando que desee ejecutar por el ARN. Puede obtener esta información a partir de la respuesta del comando `create-command` CLI.
- *target-arn*Sustitúyalo por el ARN del dispositivo o AWS IoT elemento de destino para el que desee ejecutar el comando.

  > **Note**
  >
  > Puede especificar el ARN objetivo de cualquier AWS IoT cosa (FleetWise vehículo AWS IoT). Actualmente, no se admiten grupos de cosas ni flotas.

- *endpoint-url*Sustitúyalo por el punto final específico de la cuenta que obtuvisteObtenga el punto final del plano de datos específico de la cuenta, con el prefijo, por ejemplo`https://`,. `https://`*123456789012abcd*`.jobs.iot.`*ap-south-1*`.amazonaws.com`
- Sustituya *name* y *value* por el `mandatory-parameters` campo que especificó al crear el comando mediante la `create-command` CLI.

  El `name` campo es el nombre completo definido en el catálogo de señales con `$actuatorPath.` el prefijo. Por ejemplo, `name` puede ser *`$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode`* y `value` puede ser un booleano que indica el estado de un modo de dirección similar. *`{"B": false}`*
- (Opcional) También puede especificar un parámetro adicional,. `executionTimeoutSeconds` Este campo opcional especifica el tiempo en segundos dentro del cual el dispositivo debe responder con el resultado de la ejecución. Puede configurar el tiempo de espera en un valor máximo de 24 horas.

Cuando se ha creado la ejecución del comando, se inicia un temporizador. Antes de que caduque el temporizador, si el estado de ejecución del comando no cambia a un estado que lo convierta en terminal (por ejemploFAILED, SUCCEEDED o), el estado cambia automáticamente aTIMED_OUT.

> **Note**
>
> El dispositivo también puede informar de un TIMED_OUT estado o anular este estado a un estado comoSUCCEEDED, o FAILEDREJECTED, y la ejecución del comando pasará a ser terminal. Para obtener más información, consulte Estado de tiempo de espera de ejecución del comando.

```
aws iot-jobs-data start-command-execution \
    --command-arn command-arn \
    --target-arn target-arn \
    --execution-timeout-seconds 30 \
    --endpoint-url endpoint-url \
    --parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

La operación StartCommandExecution de la API devuelve un identificador de ejecución de comandos. Puede usar este ID para consultar el estado de ejecución del comando, los detalles y el historial de ejecución del comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
 }
```

Tras ejecutar el comando, los dispositivos recibirán una notificación con la siguiente información. El issued_timestamp_ms campo corresponde a la hora en que se invocó la StartCommandExecution API. timeout_msCorresponde al valor de tiempo de espera que se configura mediante el executionTimeoutSeconds parámetro al invocar la StartCommandExecution API.

```
timeout_ms: 9000000
issued_timestamp_ms: 1723847831317
```

Actualiza el resultado de la ejecución del comando

Para actualizar el estado de la ejecución del comando, el dispositivo debe haber establecido una conexión MQTT y estar suscrito al siguiente tema de solicitud de comandos.

En este ejemplo, *<device-id>* sustitúyalo por el identificador único del dispositivo de destino, que puede ser el nombre `VehicleId` o el nombre de la cosa, y *<execution-id>* por el identificador de la ejecución del comando.

> **Note**
>
> - La carga útil debe usar el formato protobuf.
> - Es opcional que sus dispositivos se suscriban a los temas `/accepted` y `/rejected` respondan a ellos. Tus dispositivos recibirán estos mensajes de respuesta aunque no se hayan suscrito a ellos de forma explícita.

```
// Request topic
$aws/devices/<DeviceID>/command_executions/+/request/protobuf

// Response topics (Optional)
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/accepted/protobuf
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/rejected/protobuf
```

El dispositivo puede publicar un mensaje en el tema de respuesta a los comandos. Tras procesar el comando, envía una respuesta codificada en protobuf a este tema. El *<DeviceID>* campo debe coincidir con el campo correspondiente del tema de la solicitud.

```
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/<PayloadFormat>
```

Una vez que tu dispositivo publique una respuesta a este tema, podrás recuperar la información de estado actualizada mediante la `GetCommandExecution` API. El estado de la ejecución de un comando puede ser cualquiera de los que se muestran aquí.

- `IN_PROGRESS`

- `SUCCEEDED`
- `FAILED`
- `REJECTED`
- `TIMED_OUT`

Tenga en cuenta que un comando se ejecuta en cualquiera de los estados `SUCCEEDED` y `REJECTED` es terminal, y el estado lo informa el dispositivo. `FAILED` Cuando la ejecución de un comando es terminal, esto significa que no se realizarán más actualizaciones en su estado ni en los campos relacionados. El dispositivo o la nube pueden informar de un `TIMED_OUT` estado. Si lo informa la nube, el dispositivo podrá actualizar posteriormente el campo de motivo del estado.

Por ejemplo, a continuación se muestra un ejemplo de mensaje MQTT publicado por el dispositivo.

> **Note**
>
> En cuanto al estado de ejecución del comando, si sus dispositivos utilizan el `statusReason` objeto para publicar la información de estado, debe asegurarse de que:
>
> - `reasonCode`Utiliza el patrón `[A-Z0-9_-]+` y su longitud no supera los 64 caracteres.
> - La longitud `reasonDescription` no supera los 1024 caracteres. Puede utilizar cualquier carácter excepto los caracteres de control, como las líneas nuevas.

```
{
    "deviceId": "",
    "executionId": "",
    "status": "CREATED",
    "statusReason": {
        "reasonCode": "",
        "reasonDescription": ""
    }
}
```

Para ver un ejemplo que muestre cómo utilizar el cliente de pruebas de AWS IoT Core MQTT para suscribirse a los temas y ver los mensajes de ejecución de comandos, consulte Visualización de las actualizaciones de comandos mediante el cliente de pruebas de MQTT en la guía para AWS IoT Core desarrolladores.

Obtenga la ejecución de comandos

Puede utilizar la operación de la API del plano de `GetCommandExecution` AWS IoT control para recuperar información sobre la ejecución de un comando. Debe haber ejecutado ya este comando mediante la operación `StartCommandExecution` de API.

Para recuperar los metadatos de un comando ejecutado, ejecute el siguiente comando.

- *`execution-id`*Sustitúyalo por el ID del comando. Puede obtener esta información a partir de la respuesta del comando `start-command-execution` CLI.
- *`target-arn`*Sustitúyalo por el ARN del vehículo o AWS IoT objeto objetivo para el que desee ejecutar el comando.

```
aws iot get-command-execution --execution-id execution-id \
    --target-arn target-arn
```

La operación de la `GetCommandExecution` API devuelve una respuesta que contiene información sobre el ARN de la ejecución del comando, el estado de la ejecución y la hora en que el comando comenzó a ejecutarse y se completó. El código siguiente muestra un ejemplo de respuesta de la solicitud de la API.

Para proporcionar un contexto adicional sobre el estado de la ejecución de cada comando, la función de comandos proporciona un `statusReason` objeto. El objeto contiene dos campos: `reasonCode` y`reasonDescription`. Con estos campos, sus dispositivos pueden proporcionar información adicional sobre el estado de la ejecución de un comando. Esta información anulará cualquier valor predeterminado `reasonCode` `reasonDescription` que se informe desde la nube.

Para reportar esta información, sus dispositivos pueden publicar la información de estado actualizada en la nube. Luego, cuando recuperes el estado de ejecución del comando mediante la `GetCommandExecution` API, verás los códigos de estado más recientes.

> **Note**
>
> El `completedAt` campo de la respuesta de ejecución corresponde al momento en que el dispositivo notifica el estado de un terminal a la nube. En el caso del `TIMED_OUT` estado, este campo solo se configurará cuando el dispositivo indique que se ha agotado el tiempo de espera. Cuando el `TIMED_OUT` estado lo establece la nube, el `TIMED_OUT` estado no

se actualiza. Para obtener más información sobre el comportamiento del tiempo de espera, consulteEstado de tiempo de espera de ejecución del comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/LockDoor",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myFrontDoor",
    "status": "SUCCEEDED",
    "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "Parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode":

         { "B": true }
    }'
}
```

Enumere las ejecuciones de comandos en su cuenta

Usa la operación de la API HTTP del plano de `ListCommandExecutions` AWS IoT Core control para enumerar todas las ejecuciones de comandos en tu cuenta. Este ejemplo usa AWS CLI.

Temas

- Consideraciones al enumerar las ejecuciones de comandos
- Ejemplo de ejecuciones de comandos de lista

Consideraciones al enumerar las ejecuciones de comandos

Las siguientes son algunas consideraciones a tener en cuenta a la hora de utilizar la `ListCommandExecutions` API.

- Debe especificar al menos las ejecuciones de un comando en particular `targetArn` o de un vehículo objetivo, `commandArn` dependiendo de si desea enumerar las ejecuciones. La solicitud de API no puede estar vacía ni puede contener ambos campos en la misma solicitud.

- Debe proporcionar únicamente la información `startedTimeFilter` o la `completedTimeFilter` información. La solicitud de API no puede estar vacía ni puede contener ambos campos en la misma solicitud. Puede usar los `after` campos `before` y del objeto para enumerar las ejecuciones de comandos que se crearon o completaron en un período de tiempo específico.
- Los `after` campos `before` y no deben ser posteriores a la hora actual. De forma predeterminada, si no especificas ningún valor, el `before` campo es la hora actual y `after` la hora actual (6 meses). Es decir, según el filtro que utilices, la API mostrará una lista de todas las ejecuciones que se hayan creado o completado en los últimos seis meses.
- Puede usar el `sort-order` parámetro para especificar si desea enumerar las ejecuciones en orden ascendente. De forma predeterminada, las ejecuciones se muestran en orden descendente si no especificas este campo.
- No puede filtrar las ejecuciones de comandos en función de su estado al enumerar las ejecuciones de comandos de un ARN de comando.

Ejemplo de ejecuciones de comandos de lista

El siguiente ejemplo muestra cómo enumerar las ejecuciones de comandos en su Cuenta de AWS.

Al ejecutar el comando, debe especificar si desea filtrar la lista para que muestre únicamente las ejecuciones de comandos que se hayan creado para un dispositivo concreto mediante el`targetArn`, o las ejecuciones de un comando concreto especificado mediante el`commandArn`.

En este ejemplo, sustituya:

- *`<target-arn>`*con el número de recurso de Amazon (ARN) del dispositivo al que se dirige la ejecución, por ejemplo. `arn:aws:iot:`*`us-east-1`*`:`*`123456789012`*`:thing/`*`b8e4157c98f332cffb37627f`*
- *`<target-arn>`*con el número de recurso de Amazon (ARN) del dispositivo al que se dirige la ejecución, por ejemplo. `arn:aws:iot:`*`us-east-1`*`:`*`123456789012`*`:thing/`*`b8e4157c98f332cffb37627f`*
- *`<after>`*con el tiempo transcurrido el cual desea enumerar las ejecuciones que se crearon, por ejemplo,`2024-11-01T03:00`.

```
aws iot list-command-executions \
```

```
--target-arn <target-arn> \
--started-time-filter '{after=<after>}' \
--sort-order "ASCENDING"
```

Al ejecutar este comando, se genera una respuesta que contiene una lista de las ejecuciones de comandos que ha creado y la hora en que las ejecuciones comenzaron a ejecutarse y en que finalizaron. También proporciona información de estado y el `statusReason` objeto que contiene información adicional sobre el estado.


```
{
    "commandExecutions": [
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "b2b654ca-1a71-427f-9669-e74ae9d92d24",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "TIMED_OUT",
            "createdAt": "2024-11-24T14:39:25.791000-08:00",
            "startedAt": "2024-11-24T14:39:25.791000-08:00"
        },
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "34bf015f-ef0f-4453-acd0-9cca2d42a48f",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "IN_PROGRESS",
            "createdAt": "2024-11-24T14:05:36.021000-08:00",
            "startedAt": "2024-11-24T14:05:36.021000-08:00"
        }
    ]
}
```


Eliminar la ejecución de un comando

Si ya no quieres usar la ejecución de un comando, puedes eliminarla permanentemente de tu cuenta.

> **Note**
>
> La ejecución de un comando solo se puede eliminar si ha pasado a un estado terminal`SUCCEEDED`, como`FAILED`, o`REJECTED`.

En el siguiente ejemplo, se muestra cómo eliminar la ejecución de un comando mediante el `delete-command-execution` AWS CLI comando. *<execution-id>*Sustitúyalo por el identificador de la ejecución del comando que va a eliminar.

```
aws iot delete-command-execution --execution-id <execution-id>
```

Si la solicitud a la API se realiza correctamente, la ejecución del comando genera un código de estado de 200. Puedes usar la `GetCommandExecution` API para comprobar que la ejecución del comando ya no existe en tu cuenta.

Ejemplo: Uso de comandos para controlar el modo de dirección de un vehículo (AWS CLI)

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En el siguiente ejemplo, se muestra cómo utilizar la función de comandos mediante el AWS CLI. En este ejemplo, se utiliza un AWS IoT FleetWise vehículo como dispositivo objetivo para mostrar cómo se puede enviar un comando para controlar de forma remota el modo de dirección.

Temas

- Ejemplo de descripción general del modo de dirección del vehículo
- Requisitos previos
- Política de IAM para el uso de comandos remotos
- Ejecute AWS IoT comandos (AWS CLI)
- Limpieza

Ejemplo de descripción general del modo de dirección del vehículo

En este ejemplo, harás lo siguiente:

1. Cree un recurso de comando para la operación y `create-command` AWS CLI utilícelo para cambiar el modo de dirección del vehículo.
2. Recupere información sobre el comando, como la hora en que se creó o se actualizó por última vez mediante el `get-command` AWS CLI.
3. Envíe el comando al vehículo utilizando el `start-command-execution` AWS CLI modo de dirección como parámetro obligatorio, que luego se ejecutará en el dispositivo.
4. Obtenga el resultado de la ejecución del comando mediante el `get-command-execution` AWS CLI. Puede comprobar cuándo se completa la ejecución y recuperar detalles adicionales, como el resultado de la ejecución y el tiempo que se tardó en completar la ejecución del comando.
5. Realice actividades de limpieza eliminando todos los comandos y las ejecuciones de comandos que ya no desee utilizar.

Requisitos previos

Antes de ejecutar este ejemplo:

- Aprovisione su AWS IoT FleetWise vehículo como una AWS IoT cosa inscrita en el AWS IoT registro. También debes añadir un certificado a lo tuyo y activarlo, y adjuntar una política a lo tuyo. Luego, su dispositivo puede conectarse a la nube y ejecutar los comandos. Para obtener más información, consulte Aprovisionar vehículos.
- Cree un usuario de IAM y una política de IAM que le conceda permiso para realizar las operaciones de la API relacionadas con el uso de comandos, tal y como se muestra en. Política de IAM para el uso de comandos remotos

Política de IAM para el uso de comandos remotos

En la siguiente tabla se muestra un ejemplo de política de IAM que permite el acceso a todas las operaciones de la API del plano de control y del plano de datos para la función de comandos. El usuario de la aplicación tendrá permisos para realizar todas las operaciones de la API de comandos remotos, como se muestra en la tabla.

Operación de la API

Acción de la API	Plano de control/datos	Protocolo	Descripción	Recurso
`CreateCommand`	Plano de control	HTTP	Crea un recurso de comandos	• comando
`GetCommand`	Plano de control	HTTP	Recupera información sobre un comando	• comando
`UpdateCommand`	Plano de control	HTTP	Actualiza la información sobre un comando o lo deja obsoleto	• comando
`ListCommands`	Plano de control	HTTP	Muestra los comandos de tu cuenta	• comando
`DeleteCommand`	Plano de control	HTTP	Elimina un comando	• comando
`StartCommandExecution`	Plano de datos	HTTP	Comienza a ejecutar un comando	• comando • thing
`UpdateCommandExecution`	Plano de datos	MQTT	Actualiza la ejecución de un comando	• comando • thing
`GetCommandExecution`	Plano de control	HTTP	Recupera información sobre la ejecución de un comando	• comando • thing
`ListCommandExecutions`	Plano de control	HTTP	Muestra las ejecuciones de comandos en su cuenta	• comando • thing
`DeleteCommandExecution`	Plano de control	HTTP	Elimina la ejecución de un comando	• comando • thing

En este ejemplo, sustituya:

- *region*con su Región de AWS, por ejemplo. `ap-south-1`
- *account-id*con tu Cuenta de AWS número, por ejemplo`57EXAMPLE833`.
- *command-idcommand-id1*, y *command-id2* con su identificador de comando único, como `LockDoor` o`TurnOffAC`.
- *thing-name*con el nombre de tu AWS IoT cosa, como`my_car`.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
        "Action": [
            "iot:CreateCommand",
            "iot:GetCommand",
            "iot:ListCommands",
            "iot:UpdateCommand",
            "iot:DeleteCommand"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id1",
            "arn:aws:iot:<region>:<account-id>:command/command-id2",
        ]
     },
     {
        "Action": [
            "iot:GetCommandExecution",
            "iot:ListCommandExecutions",
            "iot:DeleteCommandExecution"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
        ]
     },
     {
        "Action": "iot:StartCommandExecution",
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
```

```
        ]
      }
    ]
}
```

Ejecute AWS IoT comandos (AWS CLI)

A continuación, se muestra cómo se puede utilizar AWS CLI para ejecutar operaciones con comandos y cambiar el modo de dirección del vehículo.

1. Cree un recurso de comando para el funcionamiento del modo de dirección

   Cree el comando que desee enviar a su dispositivo mediante la `create-command` CLI. En este ejemplo, especifique:

   - `command-id` como *`TurnOffSteeringMode`*
   - `role-arn`ya que `role-arn` debe proporcionarse `"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"` la función de IAM, ya que es la función de IAM la que otorga los permisos para crear y ejecutar comandos en el vehículo. Para obtener más información, consulte Conceda AWS IoT Device Management permiso para generar la carga útil de los comandos con AWS IoT FleetWise.
   - `display-name`como "» *`Turn off steering mode`*
   - `namespace`debe ser `AWS-IoT-FleetWise`
   - `mandatory-parameters`como un par nombre-valor, con `name` como "*`$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode`*" y DefaultValue como *`{ "S": "true" }`*

   > **Note**
   >
   > También puede crear un comando sin especificar ningún parámetro obligatorio. A continuación, debe especificar los parámetros que se utilizarán al ejecutar el comando mediante la `start-command-execution` CLI. Para ver un ejemplo, consulta Escenarios de uso de comandos.

> **Important**
>
> Al utilizar el espacio de `AWS-IoT-FleetWise` nombres, debe asegurarse de que el `Name` campo especificado como parte del mismo `mandatory-parameters` utilice el `$actuatorPath.` prefijo y que el `Value` campo utilice el tipo de datos de cadena.

```
aws iot create-command \
    --command-id TurnOffSteeringMode \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --display-name "Turn off steering mode" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
        "name": "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
        "defaultValue": { "S": "true" }
      }
    ]'
```

El siguiente resultado muestra un ejemplo de respuesta de la CLI, donde `ap-south-1` y `123456789012` son ejemplos del Cuenta de AWS ID Región de AWS y.

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode"
}
```

Para ver ejemplos adicionales sobre el uso de este comando, consulteCree un recurso de comandos.

2. Recupere información sobre el comando

   Ejecute el siguiente comando para recuperar información sobre el comando, donde `command-id` aparece el identificador del comando en el resultado de la `create-command` operación desde arriba.

> **Note**
>
> Si creas más de un comando, puedes usar la `ListCommands` API para enumerar todos los comandos de tu cuenta y, a continuación, usar la `GetCommand` API para obtener información adicional sobre un comando específico. Para obtener más información, consulte Enumere los comandos de su cuenta.

```
aws iot get-command --command-id TurnOffSteeringMode
```

La ejecución de este comando genera la siguiente respuesta. Verá la hora en que se creó el comando y cuándo se actualizó por última vez, los parámetros que especificó y si el comando está disponible para ejecutarse en el dispositivo.

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
            "defaultValue": {"S": "true" }
        }
    ],
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "lastUpdatedAt": "2024-03-23T00:50:10.095000-07:00",
    "deprecated": false
}
```

Para ver ejemplos adicionales sobre el uso de este comando, consulteRecupera información sobre un comando.

3. Inicie la ejecución del comando

   Ejecute el siguiente comando para iniciar la ejecución del comando, donde `command-arn` aparece el comando ARN en la salida de la `get-command` operación desde arriba. `target-`

`arn`Es el ARN del dispositivo de destino para el que está ejecutando el comando, por ejemplo,. *myVehicle*

En este ejemplo, dado que proporcionó valores predeterminados para los parámetros al crear el comando, la `start-command-execution` CLI puede usar estos valores al ejecutar el comando. También puede optar por anular el valor predeterminado especificando un valor diferente para los parámetros cuando utilice la CLI.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode \
    --target-arn arn:aws:iot:ap-south-1:123456789012:thing/myVehicle
```

La ejecución de este comando devuelve un identificador de ejecución del comando. Puede usar este ID para consultar el estado de ejecución del comando, los detalles y el historial de ejecución del comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
}
```

Para ver ejemplos adicionales sobre el uso de la CLI, consulteEnvía un comando ()AWS CLI.

4. Recupere información sobre la ejecución del comando

Ejecute el siguiente comando para recuperar información sobre el comando que ejecutó en el dispositivo de destino. Especifica el`execution-id`, que obtuviste como resultado de la `start-command-execution` operación desde arriba, y el`target-arn`, que es el ARN del dispositivo al que te diriges.

> **Note**
>
> - Para obtener la información de estado más reciente, sus dispositivos deben haber publicado la información de estado actualizada en el tema de respuesta reservada de MQTT para los comandos que utilizan la API de `UpdateCommandExecution` MQTT. Para obtener más información, consulte Actualiza el resultado de la ejecución del comando.
> - Si inicia más de una ejecución de comandos, puede utilizar la `ListCommandExecutions` API para enumerar todas las ejecuciones de comandos

de su cuenta y, a continuación, utilizarla `GetCommandExecution` para obtener información adicional sobre una ejecución específica. Para obtener más información, consulte Enumere las ejecuciones de comandos en su cuenta.

```
aws iot get-command-execution \
    --execution-id <"07e4b780-7eca-4ffd-b772-b76358da5542"> \
    --target-arn arn:aws:iot:<region>:<account>:thing/myVehicle
```

Al ejecutar este comando, se devuelve información sobre la ejecución del comando, el estado de la ejecución, la hora en que comenzó a ejecutarse y la hora en que se completó. Por ejemplo, la siguiente respuesta muestra que la ejecución del comando se realizó correctamente en el dispositivo de destino y que el modo de dirección estaba desactivado.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myVehicle",
    "result": "SUCCEEDED",
     "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "result": {
        "KeyName": {
            "S": "",
            "B": true,
            "BIN": null
        }
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode":
         { "S": "true" }
    }'
}
```

Limpieza

Ahora que ha creado un comando y lo ha ejecutado en su dispositivo, si ya no tiene intención de usarlo, puede eliminarlo. Las ejecuciones de comandos pendientes que estén en curso seguirán ejecutándose sin que se vean afectadas por la solicitud de eliminación.

> **Note**
>
> Como alternativa, también puedes dejar de usar un comando si está desactualizado y es posible que tengas que usarlo más adelante para ejecutarlo en el dispositivo de destino.

1. (Opcional) Elimine el recurso de comandos

   Ejecute el siguiente comando para dejarlo obsoleto, donde `command-id` aparece el identificador del comando que aparece arriba en el resultado de la `get-command` operación.

   ```
   aws iot update-command \
      --command-id TurnOffSteeringMode \
      --deprecated
   ```

   La ejecución de este comando devuelve un resultado que indica que el comando ha quedado obsoleto. También puede usar la CLI para restaurar el comando.

   > **Note**
   >
   > También puede usar la `update-command` CLI para actualizar el nombre mostrado y la descripción de un comando. Para obtener información adicional, consulta Actualiza o desactiva un recurso de comando.

   ```
   {
       "commandId": "TurnOffSteeringMode",
       "deprecated": true,
       "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
   }
   ```

2. Elimine el comando

   Ejecute el siguiente comando para eliminar el comando especificado por`command-id`.

   > **Note**
   >
   > La acción de eliminación es permanente y no se puede deshacer.

   ```
   aws iot delete-command --command-id TurnOffSteeringMode
   ```

   Si la solicitud de eliminación se realiza correctamente, verás un HTTP `statusCode` de 202 o 204, dependiendo de si marcaste el comando como obsoleto y de cuándo lo hizo. Para obtener más información y un ejemplo, consulte Eliminar un recurso de comandos.

   Puede usar la `get-command` CLI para comprobar que el comando se ha eliminado de su cuenta.

3. (Opcional) Elimine las ejecuciones de comandos

   De forma predeterminada, todas las ejecuciones de comandos se eliminarán en un plazo de seis meses a partir de la fecha en que las haya creado. Puede ver esta información mediante el `timeToLive` parámetro de la `GetCommandExecution` API.

   Como alternativa, si la ejecución del comando ha pasado a ser terminal, por ejemplo, cuando el estado de ejecución es uno de `SUCCEEDEDFAILED`, o bien`REJECTED`, puede eliminar la ejecución del comando. Ejecute el siguiente comando para eliminar la ejecución, donde `execution-id` aparece el identificador de ejecución que aparece arriba en el resultado de la `get-command-execution` operación.

   ```
   aws iot delete-command-execution \
               --execution-id "07e4b780-7eca-4ffd-b772-b76358da5542"
   ```

   Puede usar la `get-command-execution` CLI para comprobar que la ejecución del comando se ha eliminado de su cuenta.

Escenarios de uso de comandos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Al utilizar la función de comandos, puede crear y ejecutar comandos en los siguientes escenarios:

- Puede omitir los parámetros durante la creación y especificar solo el ID del comando. En este caso, debe especificar los parámetros que se utilizarán al ejecutar el comando en el dispositivo de destino.
- Puede especificar uno o más parámetros y configurar sus valores predeterminados al crear el comando. Proporcionar valores predeterminados le ayudará a evitar el envío de comandos imprecisos.
- Puede especificar uno o más parámetros y configurar sus valores al crear el comando. Se puede proporcionar más de un parámetro, pero solo se ejecutará uno de ellos, y el `Name` campo de este parámetro debe usar el `$actuatorPath` prefijo.

En esta sección se proporcionan algunos escenarios de uso para la `StartCommandExecution` API `CreateCommand` y el uso de los parámetros. También se muestran algunos ejemplos del uso de comandos con plantillas de estado.

Temas

- Crear un comando sin parámetros
- Crear un comando con valores predeterminados para los parámetros
- Crear un comando con valores de parámetros
- Uso de comandos con plantillas de estado

Crear un comando sin parámetros

El siguiente caso de uso muestra cómo puede usar la `CreateCommand` API o la `create-command` CLI para crear un comando sin parámetros. Al crear un comando, solo necesita proporcionar un ID de comando y un ARN de rol.

Este caso de uso es especialmente útil en casos de uso recurrentes, como cuando se quiere enviar el mismo comando varias veces a un vehículo. En este caso, el comando no está vinculado a un actuador específico y ofrece la flexibilidad necesaria para ejecutar el comando en cualquier actuador. En lugar de ello, debe especificar los parámetros en tiempo de ejecución al ejecutar el comando mediante la `StartCommandExecution` API o la `start-command-execution` CLI, que incluye los valores de los actuadores y las señales físicas.

Crear un comando sin entrada **`mandatory-parameters`**

Este caso de uso muestra cómo crear un comando sin introducir parámetros obligatorios.

```
aws iot create-command \
    --command-id "UserJourney1" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --description "UserJourney1 - No mandatory parameters" \
    --namespace "AWS-IoT-FleetWise"
```

Ejecutar un comando creado sin **`mandatory-parameters`** entrada

En este primer ejemplo, el comando que se creó anteriormente le permite ejecutar un comando en cualquier actuador sin restricciones. `actuator1`Para establecer un valor de 10, ejecute:

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

Del mismo modo, puede ejecutar un comando que se establezca `actuator3` en un valor de`true`.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
```

```
    --parameters '{
        "$actuatorPath.Vehicle.actuator3": {"S": "true"}
    }'
```

Crear un comando con valores predeterminados para los parámetros

Este comando solo permite ejecutar un comando en el actuador especificado. Proporcionar valores predeterminados le ayudará a evitar el envío de comandos imprecisos. Por ejemplo, un `LockDoor` comando que bloquea y abre puertas se puede configurar con un valor predeterminado para evitar que el comando desbloquee puertas accidentalmente.

Este caso de uso resulta especialmente útil cuando se desea enviar el mismo comando varias veces y realizar diferentes acciones en el mismo actuador, como bloquear y desbloquear las puertas de un vehículo. Si desea configurar el actuador en el valor predeterminado, no necesita pasar ninguno `parameters` a la `start-command-execution` CLI. Si especifica un valor diferente para el `parameters` en la `start-command-execution` CLI, se anulará el valor predeterminado.

Crear un comando con valores predeterminados para **`mandatory-parameters`**

El siguiente comando muestra cómo proporcionar un valor por defecto para el actuador1.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1",
            "defaultValue": {"S": "0"}
        }
    ]'
```

Ejecutar un comando creado con valores predeterminados para **`mandatory-parameters`**

El comando `UserJourney2` permite ejecutar un comando sin necesidad de pasar un valor de entrada durante el tiempo de ejecución. En este caso, la ejecución en tiempo de ejecución utilizará los valores predeterminados especificados durante la creación.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
```

```
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle
```

También puede pasar un valor diferente para el mismo actuador, el actuator1, durante el tiempo de ejecución, lo que anulará el valor predeterminado.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "139"}
    }'
```

Crear un comando con valores de parámetros

Este comando solo permite ejecutar un comando en el actuador especificado. También le obliga a establecer un valor para el actuador durante el tiempo de ejecución.

Este caso de uso es especialmente útil cuando se quiere que el usuario final solo realice determinadas acciones específicas en algunos de los actuadores cuando los utilice en el vehículo.

> **Note**
>
> Puede tener más de pares nombre-valor para la `mandatory-parameters` entrada, con valores predeterminados para algunos o todos ellos. En tiempo de ejecución, puede determinar el parámetro que desea usar cuando se ejecuta en el actuador, siempre que el nombre del actuador utilice el nombre completo con el prefijo. `$actuatorPath.`

Crear un comando sin valores por defecto para **`mandatory-parameters`**

Este comando solo permite ejecutar un comando en el actuador especificado. También le obliga a establecer un valor para el actuador durante el tiempo de ejecución.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1"
```

```
        }
    ]'
```

Ejecutar un comando creado sin valores predeterminados para **`mandatory-parameters`**

Al ejecutar el comando, en este caso, debe especificar un valor para el actuador1. La ejecución del comando que se muestra a continuación establecerá correctamente el valor de `actuator1` en. `10`

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney2 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

Uso de comandos con plantillas de estado

También puede utilizar las operaciones de la API de comandos para la recopilación y el procesamiento de datos de estado. Por ejemplo, puede obtener una instantánea del estado de un solo uso o activar o desactivar las plantillas de estado para iniciar o detener la recopilación de datos del estado del vehículo. Los siguientes ejemplos muestran cómo utilizar la función de comandos con plantillas de estado. Para obtener más información, consulte Indique las operaciones de la plantilla para la recopilación y el procesamiento de datos

> **Note**
>
> El campo Nombre especificado como parte de la `mandatory-parameters` entrada debe usar el `$stateTemplate` prefijo.

Ejemplo 1: Crear comandos para plantillas de estado con valores predeterminados

En este ejemplo, se muestra cómo utilizar la `create-command` CLI para activar las plantillas de estado.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
```

```
      {
          "name": "$stateTemplate.name"
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      }
    ]'
```

Del mismo modo, el siguiente comando muestra un ejemplo de cómo puede utilizar la `start-command-execution` CLI para las plantillas de estado.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
       "$stateTemplate.name": {"S": "ST345"}
    }'
```

Ejemplo 2: Crear comandos para plantillas de estado sin valores predeterminados

El siguiente comando crea varias plantillas de estado sin valores predeterminados para ninguno de los parámetros. Obliga a ejecutar el comando con estos parámetros y sus valores.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name",
          "defaultValue": {"S": "ST123"}
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      },
      {
          "name": "$stateTemplate.deactivateAfterSeconds",
          "defaultValue": {"L": "120"}
      }
    ]'
```

El siguiente comando muestra cómo puede utilizar la `start-command-execution` CLI para el ejemplo anterior.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
        "$stateTemplate.name": {"S": "ST345"},
        "$stateTemplate.operation": {"S": "activate"},
        "$stateTemplate.deactivateAfterSeconds" : {"L": "120"}
```

Supervisa el último estado conocido de tus vehículos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puedes monitorizar el último estado conocido de tus vehículos casi en tiempo real creando plantillas de estado y asociándolas a tus vehículos. Los vehículos asociados a las plantillas de estado transmiten datos de telemetría con una estrategia o con una `onChange` estrategia de actualización. `periodic` Con una estrategia de actualización en caso de cambio, los vehículos asociados transmiten datos de telemetría cuando se produce un cambio. Durante una estrategia de actualización periódica, los vehículos asociados transmiten datos de telemetría durante un período de tiempo específico.

Con las operaciones bajo demanda, puedes solicitar el estado actual del vehículo de una sola vez (buscar). También puedes activar o desactivar las plantillas de estado implementadas anteriormente para iniciar o detener el envío de datos sobre el estado del vehículo. Las últimas operaciones de estado conocidas se realizan mediante el AWS IoT comando APIs.

Cada plantilla de estado contiene la siguiente información.

`name`

El alias único de la plantilla de estado.

`signalCatalogArn`

El nombre de recurso de Amazon (ARN) del catálogo de señales asociado a la plantilla de estado.

`stateTemplateProperties`

Una lista de señales a partir de las cuales se recopilan datos. Las propiedades de la plantilla de estado determinan las actualizaciones de señal específicas que el vehículo envía a la nube.

`dataExtraDimensions`

Una lista de los atributos del vehículo que se incluirán en los búferes del protocolo (Protobuf) codifica los datos procesados.

`metadataExtraDimensions`

Una lista de los atributos del vehículo que se publicará con los datos procesados como una propiedad de usuario de MQTT 5.

`id`

Un identificador único generado por el servicio.

Para conocer los métodos para recopilar los datos enviados por un vehículo que utiliza el FleetWise software Edge Agent para AWS IoT, consulteProcese los datos del último estado conocido del vehículo mediante la mensajería MQTT. Para obtener más información sobre cómo asociar una plantilla de estado a un vehículo, consulteCrea un FleetWise vehículo AWS de IoT.

Temas

- Cree una plantilla AWS de FleetWise estado de IoT
- Actualizar una plantilla AWS de FleetWise estado de IoT
- Eliminar una plantilla AWS de FleetWise estado de IoT
- Obtenga información sobre plantillas FleetWise de estado de AWS IoT
- Indique las operaciones de la plantilla para la recopilación y el procesamiento de datos

Cree una plantilla AWS de FleetWise estado de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede usar la FleetWise API o la consola de AWS IoT para crear una plantilla de estado. Las plantillas de estado proporcionan un mecanismo para rastrear el estado de sus vehículos. El FleetWise software Edge Agent para AWS IoT que se ejecuta en el vehículo recopila y envía actualizaciones de señales a la nube.

Cree una plantilla de estado (consola)

Puede usar la FleetWise consola de AWS IoT para crear una plantilla de estado.

Para crear una plantilla de estado

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, selecciona Plantillas de estado.
3. En la página de plantillas de estado, elija Crear plantilla de estado.
4. En Detalles de la plantilla de estado, introduzca un nombre para la plantilla de estado y, si lo desea, introduzca una descripción.
5. En Elegir señales, añada las señales de las que desee obtener información sobre el estado del vehículo.
6. Seleccione Crear plantilla de estado.

Una vez que haya creado correctamente una plantilla de estado, la verá en la página de plantillas de estados. Ahora puede asociarla a un vehículo.

Crea una plantilla de estado (AWS CLI)

Puede utilizar la operación de la CreateStateTemplateAPI para crear una plantilla de estado. El siguiente ejemplo utiliza AWS CLI.

Para crear una plantilla de estado, ejecute el siguiente comando.

*create-state-template*Sustitúyalo por el nombre del archivo.json que contiene la configuración de la plantilla de estado.

```
aws iotfleetwise create-state-template \
    --cli-input-json file://create-state-template.json
```

Example configuración de la plantilla de estado

`stateTemplateProperties`debe contener los nombres completos de las señales.

`dataExtraDimensions`y `metadataExtraDimensions` debe contener los nombres completos de los atributos del vehículo. Las dimensiones especificadas sustituyen a cualquier valor de dimensión existente en la plantilla de estado.

```
{
    "name": "state-template-name",
```

```
    "signalCatalogArn": "arn:aws:iotfleetwise:region:account:signal-catalog/catalog-
name",
    "stateTemplateProperties": [
        "Vehicle.Signal.One",
        "Vehicle.Signal.Two"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Asociar una plantilla de FleetWise estado de AWS IoT a un vehículo

Asocie una plantilla de estado a un vehículo (consola)

Puede usar la FleetWise consola de AWS IoT para agregar plantillas de estado asociadas a un vehículo.

Para asociar una plantilla de estado

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Vehículos.
3. Elige un vehículo de la lista para abrir su página de detalles.
4. En la pestaña Plantillas de estado, selecciona Administrar plantillas de estado.
5. Seleccione Añadir plantilla de estado.
6. Seleccione una plantilla de estado y elija su método de presentación de informes.
    a. En caso de cambio: la plantilla de estado informará de los cambios en el estado del vehículo.
    b. Periódico: la plantilla de estado informará de las actualizaciones en el intervalo de tiempo especificado.
7. Seleccione Save changes (Guardar cambios).

Asociar una plantilla de FleetWise estado de AWS IoT a un vehículo (AWS CLI)

Asocie la plantilla de estado creada a un vehículo para poder recopilar las actualizaciones de estado del vehículo en la nube. Para ello, utilice:

- Al crear un vehículo, utilice el `stateTemplates` campo del `create-vehicle` comando. Para obtener más información, consulte Crea un FleetWise vehículo AWS de IoT.
- Al actualizar un vehículo, utilice los `stateTemplatesToRemove` campos `stateTemplatesToAdd` o del `update-vehicle` comando. Para obtener más información, consulte Actualizar un FleetWise vehículo AWS de IoT.

Actualizar una plantilla AWS de FleetWise estado de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede usar la operación de la UpdateStateTemplateAPI o la FleetWise consola de AWS IoT para actualizar una plantilla de estado existente.

Actualizar una plantilla de estado (consola)

Para actualizar una plantilla de estado desde la consola, vaya a la página de plantillas de estado de la FleetWise consola de AWS IoT y lleve a cabo los siguientes pasos.

1. Elija la plantilla de estado que desee actualizar y, a continuación, elija Editar.
2. Edita los detalles de la plantilla de estado y, a continuación, selecciona Guardar cambios.

Actualizar una plantilla de estado (AWS CLI)

Para actualizar una plantilla de estado, ejecute el siguiente comando.

*`update-state-template`*Sustitúyalo por el nombre del archivo.json que contiene la configuración de la plantilla de estado.

```
aws iotfleetwise update-state-template \
    --cli-input-json file://update-state-template.json
```

Example configuración de la plantilla de estado

`stateTemplateProperties`Debe contener los nombres completos de las señales.

El `dataExtraDimensions` y `metadataExtraDimensions` deben contener los nombres completos de los atributos del vehículo.

```
{
    "identifier": "state-template-name",
    "stateTemplatePropertiesToAdd": [
        "Vehicle.Signal.Three"
    ],
    "stateTemplatePropertiesToRemove": [
        "Vehicle.Signal.One"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Eliminar una plantilla AWS de FleetWise estado de IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede usar la operación de la DeleteStateTemplateAPI o la FleetWise consola de AWS IoT para eliminar una plantilla de estado.

Eliminar una plantilla de estado (consola)

Para eliminar una plantilla de estado de la consola, vaya a la página de plantillas de estado de la FleetWise consola de AWS IoT y lleve a cabo los siguientes pasos.

1. Elija la plantilla de estado que desee eliminar y, a continuación, elija Eliminar.
2. Confirme que desea eliminar la plantilla de estado y, a continuación, seleccione Eliminar.

Eliminar una plantilla de estado (AWS CLI)

Para eliminar una plantilla de estado, ejecute el siguiente comando.

*identifier*Sustitúyala por el nombre o el identificador de la plantilla de estado.

```
aws iotfleetwise delete-state-template \
    --identifier idenitfier
```

Obtenga información sobre plantillas FleetWise de estado de AWS IoT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede utilizar la operación de la GetStateTemplateAPI para recuperar información sobre una plantilla de estado. El siguiente ejemplo utiliza AWS CLI.

*identifier*Sustitúyala por el nombre de la plantilla de estado.

```
aws iotfleetwise get-state-template \
    --identifier idenitfier
```

Puede utilizar la operación de ListStateTemplatesAPI para recuperar una lista de las plantillas de estado que ha creado. El siguiente ejemplo utiliza AWS CLI.

```
aws iotfleetwise list-state-templates
```

Si has activado el cifrado mediante una AWS KMS clave gestionada por el cliente, incluye la siguiente declaración de política para que tu función pueda invocar las operaciones `GetStateTemplate` o las de la `ListStateTemplates` API.

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

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

Indique las operaciones de la plantilla para la recopilación y el procesamiento de datos

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En las siguientes secciones se describe cómo utilizar las plantillas de estado para activar y desactivar la recopilación de datos, realizar una operación de recuperación y procesar los datos de estado de sus vehículos.

Temas

- Active y desactive la recopilación de datos de estado mediante plantillas de estado

- Obtenga una instantánea del estado del vehículo mediante plantillas de estado
- Procese los datos del último estado conocido del vehículo mediante la mensajería MQTT

Active y desactive la recopilación de datos de estado mediante plantillas de estado

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En las siguientes secciones se describe cómo activar y desactivar la ingesta de datos con plantillas de estado mediante el. AWS CLI

> **Important**
>
> Antes de empezar, asegúrate de haber creado una plantilla de estado y de haberla asociado, junto con su estrategia de actualización, a un vehículo.

Debes activar una plantilla de estado para que el Edge Agent pueda enviar actualizaciones de señales a la nube.

Para realizar estas operaciones con plantillas de estado, primero cree un recurso de comando y, a continuación, inicie la ejecución del comando en el vehículo. En la siguiente sección, se describe cómo utilizar esta API y cómo activar y desactivar la ingesta de datos.

Temas

- Mediante la API de CreateCommand
- Ejemplo: activar una plantilla de estado
- Ejemplo: desactivar una plantilla de estado

Mediante la API de `CreateCommand`

Cree un recurso de comandos en el espacio de nombres `AWS-IoTFleetwise` «» y utilice los siguientes parámetros al crear o enviar un recurso de comandos para una plantilla de estado:

- `$stateTemplate.name`— El nombre de la plantilla de estado en la que se va a realizar la operación. La plantilla de estado debe aplicarse al vehículo antes de poder realizar una operación. Para obtener más información, consulte Asociar una plantilla de FleetWise estado de AWS IoT a un vehículo.
- `$stateTemplate.operation`— La operación que se va a realizar en la plantilla de estado. Utilice uno de los siguientes valores para este parámetro:
  - `activate`— El Edge Agent comienza a enviar actualizaciones de señales a la nube en función de las `stateTemplateUpdateStrategy` que especificaste (con cambios o periódicas) al aplicar la plantilla de estado al vehículo. Para obtener más información, consulte Asociar una plantilla de FleetWise estado de AWS IoT a un vehículo.

    Además, puedes definir una hora de desactivación automática de la plantilla de estado para detener las actualizaciones después de un período de tiempo específico. Si no se proporciona una hora de desactivación automática, las plantillas de estado seguirán enviando actualizaciones hasta que se emita una llamada de desactivación.

    Tan pronto como se reciba el `activate` comando, el dispositivo debería enviar las señales especificadas en la plantilla de estado de acuerdo con la estrategia de actualización. AWS IoT FleetWise recomienda que cuando el dispositivo reciba un comando de activación, el primer mensaje que envíe contenga una instantánea de todas las señales de la plantilla de estado. Los mensajes siguientes deben enviarse de acuerdo con la estrategia de actualización.
  - `deactivate`— El Edge Agent deja de enviar actualizaciones de señales a la nube.
  - `fetchSnapshot`— El Edge Agent envía una sola vez una instantánea de las señales definidas en la plantilla de estado, independientemente de las `stateTemplateUpdateStrategy` que haya especificado al aplicar la plantilla de estado al vehículo.
- (Opcional)`$stateTemplate.deactivateAfterSeconds`: la plantilla de estado se desactiva automáticamente tras el tiempo especificado. Este parámetro solo se puede utilizar cuando el valor del `$stateTemplate.operation` parámetro es «activar». Si no se especifica este parámetro, o si el valor de este parámetro es 0, el Edge Agent sigue enviando actualizaciones de señales a la nube hasta que se reciba una operación de «desactivación» de la plantilla de estado. La plantilla de estado nunca se desactiva automáticamente.

Valor mínimo: 0, valor máximo: 4294967295.

> **Note**
>
> - La API devuelve el resultado correcto en respuesta a una solicitud de activación de una plantilla que ya está en estado activo.
> - La API devuelve el resultado correcto en respuesta a una solicitud de desactivación de una plantilla que ya se encuentra en estado de desactivación.
> - La solicitud más reciente que realices en una plantilla de estado es la que surte efecto. Por ejemplo, si solicita la desactivación de una plantilla de estado en una hora y, a continuación, hace una segunda solicitud para que la misma plantilla se desactive en cuatro horas, la desactivación de cuatro horas entrará en vigor debido a que es la solicitud más reciente.

> **Important**
>
> Se puede producir una excepción de validación en cualquiera de los siguientes escenarios:
>
> - Se proporciona una plantilla de estado que no corresponde `ASSOCIATED` a un vehículo.
> - Se ha realizado una solicitud para activar una plantilla de estado, pero no ha estado `DEPLOYED` en un vehículo.
> - Se hace una solicitud a una plantilla de estado, pero se hace `DELETED` en un vehículo.

Ejemplo: activar una plantilla de estado

Para activar una plantilla de estado, cree primero un recurso de comando. A continuación, puede enviar el siguiente comando al vehículo en el que desee activar la plantilla de estado. En este ejemplo se muestra cómo se pueden especificar los valores por defecto para los parámetros al crear un comando. Estos parámetros y sus valores se utilizan al iniciar la ejecución del comando para activar la plantilla de estado.

1. Cree un recurso de comando

   Antes de poder enviar un comando al vehículo, debe crear un recurso de comando. Puede especificar valores alternativos para los parámetros obligatorios al enviar el comando al vehículo. Para obtener más información, consulte Cree un recurso de comandos.

   > **Important**
   >
   > `$stateTemplate.name`y `$stateTemplate.operation` los parámetros deben proporcionarse como un tipo de datos de cadena. Si se proporciona cualquier otro tipo de datos, o si falta alguno de estos dos parámetros, se produce un error en la ejecución del comando y se produce una excepción de validación. El `$stateTemplate.deactivateAfterSeconds` parámetro debe proporcionarse como un tipo `Long` de datos.

```
aws iot create-command \
    --description "This command activates a state template on a vehicle"
    --command-id ActivateStateTemplate \
    --display-name "Activate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "activate"}
    },
    {
        "name": "$stateTemplate.deactivateAfterSeconds",
        "defaultValue": {"L": "120"}
    }
]'
```

2. Inicie la ejecución del comando en el vehículo

   Una vez creado el comando, envíelo al vehículo. Si no especificó valores para los parámetros obligatorios al crear el recurso de comando, debe especificarlos ahora. Para obtener más información, consulte Envía un comando ()AWS CLI.

   > **Important**
   >
   > Asegúrese de utilizar el punto final de la API del plano de datos de AWS IoT trabajos específico de la cuenta para la operación de la API.

   ```
   aws iot-jobs-data start-command-execution \
       --endpoint-url <endpoint-url> \
       --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/ActivateStateTemplate \
       --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
   ```

3. Recupera el estado de la operación de la plantilla de estado

   Tras iniciar la ejecución del comando, puede utilizar la `GetCommandExecution` API para recuperar la plantilla de estado.

   ```
   aws iot get-command-execution --execution-id <EXECUTION_ID>
   ```

Ejemplo: desactivar una plantilla de estado

Para desactivar una plantilla de estado, cree primero un recurso de comando. A continuación, puede enviar el siguiente comando al vehículo en el que desee desactivar la plantilla de estado. En este ejemplo se muestra cómo se pueden especificar los valores por defecto para los parámetros al crear un comando. Estos parámetros y sus valores se utilizan al iniciar la ejecución del comando para desactivar la plantilla de estado.

1. Cree un recurso de comando

   Antes de poder enviar un comando al vehículo, debe crear un recurso de comando. Puede especificar valores alternativos para los parámetros obligatorios al enviar el comando al vehículo. Para obtener más información, consulte Cree un recurso de comandos.

```
aws iot create-command \
    --description "This command deactivates a state template on a vehicle"
    --command-id DeactivateStateTemplate \
    --display-name "Deactivate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "deactivate"}
    }
]'
```

2. Inicie la ejecución del comando en el vehículo

   Una vez creado el comando, envíelo al vehículo. Si no especificó valores para los parámetros obligatorios al crear el recurso de comando, debe especificarlos ahora. Para obtener más información, consulte Envía un comando ()AWS CLI.

```
aws iot-jobs-data start-command-execution \
    --endpoint-url <endpoint-url> \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/DeactivateStateTemplate
 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

3. Recupera el estado de la operación de la plantilla de estado

   Tras iniciar la ejecución del comando, puede utilizar la `GetCommandExecution` API para recuperar la plantilla de estado.

```
aws iot get-command-execution  --execution-id <EXECUTION_ID>
```

Obtenga una instantánea del estado del vehículo mediante plantillas de estado

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Puede obtener el último estado conocido de un vehículo mediante la operación de la API del plano de `CreateCommand` AWS IoT Core control o la FleetWise consola AWS IoT.

> **Important**
>
> Se puede producir una excepción de validación en cualquiera de los siguientes escenarios:
>
> - Se proporciona una plantilla de estado que no corresponde `ASSOCIATED` a un vehículo.
> - Se ha realizado una solicitud para activar una plantilla de estado, pero no ha estado `DEPLOYED` en un vehículo.
> - Se hace una solicitud a una plantilla de estado, pero se hace `DELETED` en un vehículo.

Obtenga una instantánea del estado del vehículo (consola)

Puedes usar la FleetWise consola AWS IoT para buscar el último estado conocido de un vehículo. AWS IoT FleetWise creará un comando para que obtengas datos.

Para obtener el estado de un vehículo

1. Abre la FleetWise consola AWS de IoT.
2. En el panel de navegación, elija Vehículos.
3. Elige un vehículo de la lista para abrir su página de detalles.
4. En la pestaña Plantillas de estado, selecciona Obtener datos.
5. Seleccione el rol de IAM que otorga FleetWise permisos de AWS IoT para enviar un comando y obtener datos. Consulte Controlar el acceso.
6. Seleccione Fetch state.

Obtenga una instantánea del estado del vehículo ()AWS CLI

Para obtener una instantánea de estado, primero cree un recurso de comando. A continuación, puede enviar el siguiente comando al vehículo del que quiere obtener la instantánea de estado. Para obtener más información sobre el uso de la `CreateCommand` API y sus parámetros, consulteMediante la API de `CreateCommand`.

1. Cree un recurso de comandos

   El siguiente ejemplo muestra cómo crear el recurso de comandos para realizar la operación de recuperación. Puede especificar valores alternativos para los parámetros obligatorios al enviar el comando al vehículo. Para obtener más información, consulte Cree un recurso de comandos.

   ```
   aws iot create-command \
       --command-id <COMMAND_ID> \
       --display-name "FetchSnapshot State Template" \
       --namespace AWS-IoTFleetWise \
       --mandatory-parameters '[
         {
             "name": "$stateTemplate.name",
             "defaultValue": {"S": "ST123"}
         },
         {
             "name": "$stateTemplate.operation",
             "defaultValue": {"S": "fetchSnapshot"}
         }
       ]'
   ```

   Respuesta:

   ```
   {
       "commandId": "<COMMAND_ID>",
       "commandArn": "arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID>"
   }
   ```

2. Inicie la ejecución del comando para obtener una instantánea del estado

   Una vez creado el comando, envíelo al vehículo. Si no especificó valores para los parámetros obligatorios al crear el recurso de comando, debe especificarlos ahora. Para obtener más información, consulte Envía un comando ()AWS CLI.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

Respuesta:

```
{
    "executionId": "<UNIQUE_UUID>"
}
```

3. Recupera el estado de la operación de la plantilla de estado

   Tras iniciar la ejecución del comando, puede utilizar la `GetCommandExecution` API para recuperar la plantilla de estado.

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

Procese los datos del último estado conocido del vehículo mediante la mensajería MQTT

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Para recibir actualizaciones de su vehículo y procesar sus datos, suscríbase al siguiente tema de MQTT. Para obtener más información, consulte Temas MQTT en la Guía para desarrolladores de AWS IoT Core .

```
$aws/iotfleetwise/vehicles/$vehicle_name/last_known_state/$state_template_name/data
```

Es posible que los últimos mensajes de actualización de la señal de estado conocidos se reciban fuera de servicio, ya que MQTT no garantiza el pedido. Cualquier cliente que utilice MQTT para recibir y procesar los datos del vehículo debe encargarse de ello. Los últimos mensajes de actualización de señal de estado conocidos siguen el protocolo de mensajería MQTT 5.

El encabezado de cada mensaje MQTT tiene las siguientes propiedades de usuario:

- VehicleName: un identificador único de los vehículos.
- stateTemplateName— Un identificador único de la última plantilla de estado conocida.

Además, puede especificar los atributos del vehículo que se incluirán en el encabezado del mensaje MQTT especificando el parámetro de `metadataExtraDimensions` solicitud al actualizar o crear una plantilla de estado. (Consulte las plantillas de estado).

Las propiedades de usuario del encabezado de los mensajes de MQTT son útiles para enrutar los mensajes a diferentes destinos sin inspeccionar la carga útil.

La carga útil de los mensajes MQTT contiene los datos recopilados de los vehículos. Puede especificar los atributos del vehículo que se incluirán en la carga útil del mensaje MQTT especificando el parámetro de `extraDimensions` solicitud al crear o actualizar una plantilla de estado (consulte). Cree una plantilla AWS de FleetWise estado de IoT Las dimensiones adicionales enriquecen los datos recopilados de los vehículos al asociarles dimensiones adicionales.

La carga útil del mensaje MQTT está codificada en búferes de protocolo (Protobuf) y el encabezado del mensaje MQTT contiene un indicador de tipo de contenido definido como aplicación/flujo de octetos. El esquema de codificación de Protobuf es el siguiente:

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

syntax = "proto3";

option java_package = "com.amazonaws.iot.autobahn.schemas.lastknownstate";
package Aws.IoTFleetWise.Schemas.CustomerMessage;

message LastKnownState {

  /*
   * The absolute timestamp in milliseconds since Unix Epoch of when the event was
 triggered in vehicle.
   */
  uint64 time_ms = 1;

  /*
   * This field is deprecated, use signals instead
   */
```

```
  repeated Signal signal = 2 [ deprecated = true ];

  repeated Signal signals = 3;

  repeated ExtraDimension extra_dimensions = 4;
}

message Signal {

  /*
   * The Fully Qualified Name of the signal is the path to the signal plus the signal's
 name.
   * For example, Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  /*
   * The FWE reported signal value can be one of the following data types.
   */
  oneof SignalValue {
    double double_value = 2;

    bool boolean_value = 3;

    sint32 int8_value = 4;

    uint32 uint8_value = 5;

    sint32 int16_value = 6;

    uint32 uint16_value = 7;

    sint32 int32_value = 8;

    uint32 uint32_value = 9;

    sint64 int64_value = 10;

    uint64 uint64_value = 11;

    float float_value = 12;
    /*
```

```
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 13;
  }
}

message ExtraDimension {
  /*
   * The Fully Qualified Name of the attribute is the path to the attribute plus the
 attribute's name.
   * For example, Vehicle.Model.Color
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  oneof ExtraDimensionValue {
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 2;
  }
}
```

Donde:

- `time_ms`:

  La marca temporal absoluta (en milisegundos desde la época de Unix) del momento en que se desencadenó el evento en el vehículo. El software Edge Agent se utiliza en el reloj del vehículo para esta marca de tiempo.
- `signal`:

  Matriz de `Signal` s que contiene la información de la señal: `name` (cadena) y `signalValue` que admite los siguientes tipos de datos: `doublebool`,`int8`,`uint8`,`int16`,,`uint16`,`int32`,`uint32`, `int64uint64`,`float`. `string`
- `extra_dimensions`:

  Una matriz `ExtraDimensions` que contiene la información sobre los atributos del vehículo: `name` (cadena) y `extraDimensionValue` que actualmente solo admite el tipo de `string` datos.

Tutorial: Configurar la recopilación de datos independiente de la red mediante una interfaz de decodificación personalizada

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Introducción

Este tutorial describe cómo configurar el AWS IoT FleetWise para recopilar datos y ejecutar comandos mediante la recopilación de datos independiente de la red, que utiliza una interfaz de decodificación personalizada. Con la recopilación de datos independiente de la red, puede utilizar sus propios métodos para decodificar las señales antes de enviarlas al destino de datos especificado. Esto ahorra tiempo, ya que no es necesario crear decodificadores de señal específicos para el AWS IoT FleetWise. Puedes hacer que un subconjunto de señales se decodifique con tu propia implementación o puedes usarlo `defaultForUnmappedSignals` al crear o actualizar un manifiesto de decodificador. Esto también proporciona flexibilidad para recopilar señales y activadores a través de una amplia gama de fuentes dentro del vehículo.

Este tutorial está destinado a las señales de vehículos que no se encuentran en una interfaz de red de área de controlador (bus CAN) estándar. Por ejemplo, datos codificados en un formato o esquema personalizado para el vehículo.

Configuración del entorno

En este tutorial se asume que ha seguido los pasos necesarios para configurar sus entornos para acceder a la FleetWise nube de AWS IoT y a la implementación APIs y la base de código de Edge.

Modelos de datos

La siguiente sección ilustra cómo modelar las propiedades del vehículo mediante una interfaz de decodificación personalizada. Esto se aplica a la recopilación de datos, así como a los casos de uso de comandos. También se aplica a cualquier modelo de fuente de datos subyacente utilizado en el vehículo, por ejemplo IDLs.

En el ejemplo, hay dos propiedades del vehículo: un sensor del vehículo (posición actual del vehículo) que se recopila y un actuador del vehículo (aire acondicionado) que se controla de forma remota. Ambas se definen en este esquema:

```
// Vehicle WGS84 Coordinates
double Latitude;
double Longitude;

// Vehicle AC
Boolean ActivateAC;
```

El siguiente paso es importar estas definiciones al AWS IoT FleetWise mediante la interfaz de decodificación personalizada APIs.

Actualizaciones del catálogo de señales

Importe estas definiciones a su catálogo de señales. Si FleetWise ya dispone de un catálogo de señales en el AWS IoT, utilice directamente la API de actualización. Si no tiene uno, cree primero un catálogo de señales y, a continuación, llame a la API de actualización.

En primer lugar, debe crear la representación VSS de estas señales de vehículos. El VSS se utiliza como taxonomía para representar los datos de los vehículos en el IoT AWS . FleetWise Cree un archivo json llamado «vehicle-signals.json» con el siguiente contenido:

```
// vehicle-signals.json
// Verify that branches and nodes are unique in terms of fully qualified name
// in the signal catalog.
[
 {
    "branch": {
      "fullyQualifiedName": "Vehicle",
      "description": "Vehicle Branch"
```

```
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.CurrentLocation",
      "description": "CurrentLocation"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "description": "Latitude"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "description": "Longitude"
    }
  },
  {
    "actuator": {
      "fullyQualifiedName": "Vehicle.ActivateAC",
      "description": "AC Controller",
      "dataType": "BOOLEAN"
    }
  }
]
```

Si no tiene un catálogo de señales, debe invocar: `create-signal-catalog`

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise create-signal-catalog \
        --name my-signal-catalog \
        --nodes "${VEHICLE_NODES}"
```

Si ya tiene un catálogo de señales, puede añadir esas señales mediante la `update-signal-catalog` API:

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise update-signal-catalog \
```

```
        --name my-signal-catalog \
        --nodes-to-add "${VEHICLE_NODES}"
```

Modelo y decodificador del vehículo

Tras insertar las señales en el catálogo de señales, el siguiente paso es crear un modelo de vehículo e instanciar esas señales. Para ello, se utilizan las teclas y. `create-model-manifest` `create-decoder-manifest` APIs

Primero, formatea los nombres de las señales que deseas insertar en el modelo del vehículo:

```
# Prepare the signals for insertion into the vehicle model.
VEHICLE_NODES=`cat vehicle-signals.json`
VEHICLE_NODES=`echo ${VEHICLE_NODES} | jq -r ".[] | .actuator,.sensor
 | .fullyQualifiedName" | grep Vehicle\\.`
VEHICLE_NODES=`echo "${VEHICLE_NODES}" | jq -Rn [inputs]`
# This is how the vehicle model input looks.
echo $VEHICLE_NODES
# [ "Vehicle.CurrentLocation.Latitude",
#   "Vehicle.CurrentLocation.Longitude",
#   "Vehicle.ActivateAC" ]
# Create the vehicle model with those signals.
aws iotfleetwise create-model-manifest \
    --name my-model-manifest \
    --signal-catalog-arn arn:xxxx:signal-catalog/my-signal-catalog \
    --nodes "${VEHICLE_NODES}"

# Activate the vehicle model.
 aws iotfleetwise update-model-manifest \
    --name my-model-manifest --status ACTIVE
```

Ahora, utilice la interfaz de decodificación personalizada para crear un manifiesto del decodificador.

> **Note**
>
> Solo necesita crear interfaces y señales de red si desea especificar una configuración personalizada IDs, lo cual no forma parte de este ejemplo.
> Para obtener información sobre cómo mapear la información de decodificación cuando el nombre completo (FQN) difiere del ID de la señal de decodificación personalizada, consulte la Guía para desarrolladores de Edge Agent.

```
// Create a network interface that is of type : CUSTOM_DECODING_INTERFACE
// custom-interface.json
[
  {
    "interfaceId": "NAMED_SIGNAL",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  },
  {
    "interfaceId": "AC_ACTUATORS",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
// custom-decoders.json
// Refer to the fully qualified names of the signals, make them of
// type CUSTOM_DECODING_SIGNAL, and specify them as part of the same interface ID
// that was defined above.
[
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Longitude"
      }
    },
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Latitude"
      }
    },
    {
        "fullyQualifiedName": "Vehicle.ActivateAC",
        "interfaceId": "AC_ACTUATORS",
        "type": "CUSTOM_DECODING_SIGNAL",
```

```
        "customDecodingSignal": {
          "id": "Vehicle.ActivateAC"
        }
    }
]
# Create the decoder manifest.
 CUSTOM_INTERFACE=`cat custom-interface.json`
 CUSTOM_DECODERS=`cat custom-decoders.json`

aws iotfleetwise create-decoder-manifest \
   --name my-decoder-manifest \
   --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
   --network-interfaces "${CUSTOM_INTERFACE}" \
   --signal-decoders "${CUSTOM_DECODERS}"

 # Activate the decoder manifest.
 aws iotfleetwise update-decoder-manifest \
    --name my-decoder-manifest \
    --status ACTIVE
```

En este punto, ya ha modelado completamente estas señales en el AWS IoT FleetWise. A continuación, crea el vehículo y lo asocia al modelo que ha creado. Para ello, utilizas la `create-vehicle` API:

```
aws iotfleetwise create-vehicle \
    --decoder-manifest-arn arn:xxx:decoder-manifest/my-decoder-manifest \
    --association-behavior ValidateIotThingExists \
    --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
    --vehicle-name "my-vehicle"
```

El siguiente paso es centrarse en la base de código de AWS IoT FleetWise Edge y escribir la extensión de código necesaria.

> **Note**
>
> Para obtener información sobre la implementación de Edge, consulte la Guía para desarrolladores de Edge Agent.

Enviar comando

Ahora, compile el software (asegúrese de añadir los encabezados y los archivos de C++ al CMake archivo) y, a continuación, regrese a la nube APIs para probar un comando en este actuador:

```
// Create a command targeting your vehicle.
aws iot create-command --command-id activateAC \
    --namespace "AWS-IoT-Fleetwise" \
    --endpoint-url endpoint-url \
    --role-arn ${SERVICE_ROLE_ARN} \
    --mandatory-parameters '[ { "name": "$actuatorPath.Vehicle.ActivateAC",
 "defaultValue": {"B": "false"} } ]' \
// You will receive the command ARN.

{
    "commandId": "activateAC",
    "commandArn": "arn:aws:iot:xxx:command/activateAC"
}

// You can send the command to activate the AC targeting your vehicle.

JOBS_ENDPOINT_URL=`aws iot describe-endpoint --endpoint-type iot:Jobs | jq -
j .endpointAddress`
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:xxx:command/activateAC \
    --target-arn arn:xxx:vehicle/my-vehicle \
    --parameters '{ "$actuatorPath.Vehicle.ActivateAC" : {"B": "true"}}' \
    --endpoint-url https://${JOBS_ENDPOINT_URL}
// You will receive the corresponding execution ID.
{
    "executionId": "01HSK4ZH6ME7D43RB2BV8JC51D"
}

// If you have the AWS IoT FleetWise Edge Agent running, you can see the logs.
[AcCommandDispatcher.cpp:26] [setActuatorValue()]:
[Actuator Vehicle.ActivateAC executed successfully for command ID
 01HSK4ZH6ME7D43RB2BV8JC51D]
```

Uso AWS CLI y AWS SDKs con el AWS IoT FleetWise

En esta sección se proporciona información sobre cómo realizar solicitudes a la FleetWise API de AWS IoT. Para obtener más información sobre FleetWise las operaciones y los tipos de datos de AWS IoT, consulta la Referencia de FleetWise API de AWS IoT.

Para utilizar el AWS IoT FleetWise con una variedad de lenguajes de programación, utilice el AWS SDKs, que contiene las siguientes funciones automáticas:

- 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 el AWS IoT FleetWise y sus otros servicios desde la línea de comandos y automatizarlos mediante scripts.

Solución de problemas de AWS IoT FleetWise

Utilice la información y las soluciones de solución de problemas de esta sección para ayudar a resolver los problemas relacionados con el AWS IoT FleetWise.

La siguiente información puede ayudarte a solucionar problemas comunes relacionados con el AWS IoT FleetWise.

Temas

- AWS Problemas con el manifiesto del FleetWise decodificador de IoT
- Edge Agent para problemas AWS de FleetWise software de IoT
- Almacena y reenvía las incidencias

AWS Problemas con el manifiesto del FleetWise decodificador de IoT

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 FleetWise El IoT intenta realizar la validación con respecto a un modelo de

Error	Directrices para solucionar problemas
	vehículo (manifiesto de modelo) que está siendo modificado por otra persona. Espere e inténtelo de nuevo.
`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 El IoT no FleetWise encuentra 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.

Error	Directrices para solucionar problemas
`UpdateOperationFailure.KmsKeyAccessDenied`	Hay un problema de permiso con la tecla 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.
`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 El IoT FleetWise no encontró la estructura de señal raíz utilizada en el decodificador de señales utilizando 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_IN`	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

Error	Directrices para solucionar problemas
`COMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	primitivos definidos en la solicitud coincidan con la definición del catálogo de señales correspondiente.
`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 El IoT FleetWise encontró una señal definida como STRUCT 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 nombre de campo structFullyQualified. También necesita un nodo STRUCT con eso. fullyQualifiedName

Error	Directrices para solucionar problemas
`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.
`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 en el catálogo de señales como structuredMessageList definición. Compruebe que todas las propiedades de la matriz estén definidas como structuredMessageList definición en la solicitud de actualización del manifiesto del decodificador.

Edge Agent para problemas AWS de FleetWise software de IoT

Solucione problemas con el software Edge Agent.

Problemas

- Problema: el software Edge Agent no se inicia.
- Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [No se pudo iniciar la biblioteca de persistencia]
- Problema: el software Edge Agent no recopila los códigos de diagnóstico integrados (OBD) II PIDs ni los códigos de diagnóstico de problemas (). DTCs
- Problema: el FleetWise software Edge Agent para AWS IoT no recopila datos de la red o no puede aplicar reglas de inspección de datos.
- Problema: [ERROR] [AwsIotConnectivityModule: :connect]: [La conexión falló por error] o [WARN] [AwsIotChannel: :send]: [No hay una conexión MQTT activa.]

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 FleetWise software Edge Agent para AWS IoT 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 de FleetWise software Edge Agent para AWS IoT.

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

- ```
  [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: *network-interface-id*Sustitúyalo por el ID de 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 final del dispositivo IoT de la AWS cuenta.
- `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.
- Has utilizado AWS IoT Core para aprovisionar el vehículo. Para obtener más información, consulte Aprovisionamiento AWS de FleetWise vehículos de IoT.

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

Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [No se pudo iniciar la biblioteca de persistencia]

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 códigos de diagnóstico integrados (OBD) II PIDs ni los códigos de diagnóstico de problemas (). DTCs

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 tu vehículo admite el arbitraje de red de área de controlador extendida (bus CAN) IDs, asegúrate de que `obdInterface:useExtendedIds` esté configurado para ello. `true`

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

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

Recurso	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 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 recopilac ión 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]: [La conexión falló por error] o [WARN] [AwsIotChannel: :send]: [No hay una conexión MQTT activa.]

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 AWS IoT Core cada minuto y espera tres minutos. Si no hay respuesta, el software Edge Agent restablece automáticamente la conexión a la nube.

Almacena y reenvía las incidencias

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Problema: Recibir un anuncio **`AccessDeniedException`** con todos los permisos de IAM necesarios

Solución: la versión de acceso anticipado de la función de almacenar y reenviar para dividir los datos en las campañas requiere la opción Perlowlisting. Ponte en contacto con el equipo de servicio para asegurarte de que tus recursos cuentan con los permisos adecuados mediante la inclusión en la lista de permisos.

Problema: Los datos subidos a AWS IoT Jobs ignoran las **`endTime`**

Solución: especificó un elemento no válido `endtime` en el documento de trabajo. Por ejemplo, `endtime` no sigue el formato ISO 8601 (UTC). En los registros del AWS IoT FleetWise agente, puede haber una declaración de nivel de advertencia que diga:. `Malformed IoT Job endTime: customer configured endTime. Not setting endTime`

Problema: La carga de datos a AWS IoT Jobs tiene un **REJECTED** estado de ejecución.

Solución: especificó un elemento no válido `campaignArn` en el documento de trabajo. Por ejemplo, si especificas un ARN para una campaña que no se está ejecutando en un vehículo, podría haber una declaración de nivel de error que diga: `CampaignArn value in the received job document does not match the ARN of a Store and Forward campaign` en los registros del agente. AWS IoT FleetWise

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
- Controlar el acceso con AWS IoT FleetWise
- Identity and Access Management para AWS IoT FleetWise
- Validación de conformidad para AWS IoT FleetWise
- Resiliencia en el AWS IoT FleetWise
- Seguridad de la infraestructura en el AWS IoT FleetWise
- Análisis de configuración y vulnerabilidad en AWS IoT FleetWise
- Mejores prácticas de seguridad para el AWS IoT FleetWise

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. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres 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, consulta las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. 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. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza 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 140-3 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, consulta Estándar de procesamiento de la información federal (FIPS) 140-3.

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 trabaja con AWS IoT FleetWise u otro tipo Servicios de AWS mediante la consola, la API o AWS SDKs. AWS CLI Cualquier dato que ingrese

en etiquetas o campos de texto 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.

El cifrado en reposo en el AWS IoT FleetWise

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. Para obtener más información, consulte Protección de datos mediante cifrado en la Guía del usuario de Amazon S3 y Protección de datos en Timestream for. LiveAnalytics

Encryption at rest 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 en AWS IoT FleetWise

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 qué están alojados en FleetWise pasarelas de AWS IoT y a los APIs que se puede acceder a través de la red local. APIs Se exponen a través de una conexión TLS respaldada por un certificado de servidor propiedad del conector AWS IoT FleetWise Edge. Para la autenticación del cliente, APIs 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

- El cifrado en reposo en el AWS IoT FleetWise
- Gestión de claves en AWS IoT FleetWise

El cifrado en reposo en el AWS IoT FleetWise

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 guardados 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 usar tu AWS KMS clave.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Important**
>
> 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.

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 Gestión de claves en AWS IoT FleetWise.

Gestión de claves en AWS IoT FleetWise

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

AWS Gestión de claves FleetWise en la nube de IoT

De forma predeterminada, AWS IoT Claves administradas por AWS se FleetWise utiliza 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

AWS FleetWise Recurso de IoT	Tipo de datos:	Campos que están cifrados en reposo con claves administradas por el cliente
	Actuador	description, allowedValues, min, max
	Sensor	description, allowedValues, min, max
Modelo de vehículo (manifiesto del modelo)		description
Manifiesto del decodificador		description
	CanInterface	protocolName, protocolVersion
	ObdInterface	requestMessageId, dtcReques tInterval Segundos hasTransm issionEcu, estándar OBD, segundos, pidRequestInterval useExtendedIds
	CanSignal	factor, isBigEndian isSigned, length, 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

AWS FleetWise Recurso de IoT	Tipo de datos:	Campos que están cifrados en reposo con claves administradas por el cliente
Plantilla de estado		description

> **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:

- *kms_key_id*Sustitúyala por el ID de la clave KMS.

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

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. El principio del FleetWise

servicio de AWS IoT `iotfleetwise.amazonaws.com` en la declaración de política clave del KMS permite que el AWS IoT acceda FleetWise a la clave del KMS.

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

Como práctica recomendada de seguridad, añada `aws:SourceArn` y `aws:SourceAccount` condicione las claves a la política de claves del KMS. La clave de condición global de IAM `aws:SourceArn` ayuda a garantizar que AWS IoT FleetWise utilice la clave KMS solo para el recurso Amazon Resource Names () específico del servicio. ARNs

Si estableces el valor de`aws:SourceArn`, siempre debe ser así. `arn:aws:iotfleetwise:us-east-1:account_id:*` Esto permite que la clave KMS acceda a todos los FleetWise recursos de AWS IoT para ello Cuenta de AWS. AWS IoT FleetWise admite una clave KMS por cuenta para todos los recursos que contiene Región de AWS. Si se utiliza cualquier otro valor para el `SourceArn` campo de recursos del ARN o no se utiliza el comodín (*) en el campo de recursos del ARN, se impide que el AWS FleetWise IoT acceda a la clave KMS.

El valor de `aws:SourceAccount` es el identificador de su cuenta, que se utiliza para restringir aún más la clave de KMS, de modo que solo pueda usarse para su cuenta específica. Si agrega claves a la clave de KMS `aws:SourceAccount` y las `aws:SourceArn` condiciona, asegúrese de que ningún otro servicio o cuenta utilice la clave. Esto ayuda a evitar errores.

La siguiente política incluye una entidad principal de servicio (un identificador de un servicio), así como `aws:SourceAccount` una `aws:SourceArn` configuración para su uso en función de la ID de su cuenta Región de AWS y de la suya.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:SourceAccount": "AWS-account-ID"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iotfleetwise:region:AWS-account-ID:*"
    }
  }
}
```

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.

Permisos de AWS KMS cifrado

Si habilitó el AWS KMS cifrado, debe especificar los permisos en la política de roles para poder llamar a AWS IoT FleetWise APIs. La siguiente política permite el acceso a todas FleetWise las acciones de AWS IoT, así como a permisos AWS KMS específicos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:*",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

La siguiente declaración de política es necesaria para que su función pueda invocar el cifrado APIs. Esta declaración de política permite `PutEncryptionConfiguration` y `GetEncryptionConfiguration` actúa desde el AWS IoT FleetWise.

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

```
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Recuperación tras la eliminación de AWS KMS la clave

Si eliminas una AWS KMS clave después de habilitar el cifrado con AWS IoT FleetWise, debes restablecer tu cuenta eliminando todos los datos antes de FleetWise volver a usar AWS IoT. Puedes usar la lista y eliminar las operaciones de la API para limpiar los recursos de tu cuenta.

Para limpiar los recursos de tu cuenta

1. Utilice la lista APIs con el `listResponseScope` parámetro establecido en`METADATA_ONLY`. Esto proporciona una lista de recursos, incluidos los nombres de los recursos y otros metadatos, como ARNs las marcas de tiempo.
2. Use delete APIs para eliminar recursos individuales.

Debe limpiar los recursos en el siguiente orden.

1. Campañas
   a. Enumera todas las campañas con el `listResponseScope` parámetro establecido en`METADATA_ONLY`.
   b. Elimine las campañas.
2. Flotas y vehículos
   a. Enumere todas las flotas con el `listResponseScope` parámetro establecido en. `METADATA_ONLY`
   b. Enumere todos los vehículos de cada flota con el `listResponseScope` parámetro establecido en. `METADATA_ONLY`
   c. Desvincule todos los vehículos de cada flota.

   d. Elimine las flotas.
   e. Elimine los vehículos.
3. Manifiestos del decodificador
   a. Enumere todos los manifiestos del decodificador con el `listResponseScope` parámetro establecido en. `METADATA_ONLY`
   b. Elimine todos los manifiestos del decodificador.
4. Modelos de vehículos (manifiestos de modelos)
   a. Enumere todos los modelos de vehículos con el `listResponseScope` parámetro establecido en`METADATA_ONLY`.
   b. Elimine todos los modelos de vehículos.
5. Plantillas de estados
   a. Enumere todas las plantillas de estado con el `listResponseScope` parámetro establecido en`METADATA_ONLY`.
   b. Elimine todas las plantillas de estado.
6. Catálogos de señales
   a. Enumere todos los catálogos de señales.
   b. Elimine todos los catálogos de señales.

Controlar el acceso con AWS IoT FleetWise

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

En las siguientes secciones, se explica cómo controlar el acceso a AWS IoT FleetWise los 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, o a los mensajes MQTT que se utilizan para enviar datos desde los vehículos.

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

- Otorgue AWS IoT FleetWise permiso para enviar y recibir datos sobre un tema de MQTT
- Conceder AWS IoT FleetWise acceso a un destino de Amazon S3
- Conceder AWS IoT FleetWise acceso a un destino de Amazon Timestream
- Conceda AWS IoT Device Management permiso para generar la carga útil de los comandos con AWS IoT FleetWise

Otorgue AWS IoT FleetWise permiso para enviar y recibir datos sobre un tema de MQTT

Cuando utilizas un tema de MQTT, tus vehículos envían datos mediante el intermediario de mensajes de AWS IoT MQTT. Debe conceder AWS IoT FleetWise permiso para suscribirse al tema de MQTT que especifique. Si también utilizas AWS IoT Rules para tomar medidas o enrutar datos a otros destinos, debes adjuntar políticas a un rol de IAM para permitir el reenvío de datos AWS IoT FleetWise a las Reglas de IoT.

Además, tus otras aplicaciones o dispositivos pueden suscribirse al tema que especifiques para recibir datos del vehículo prácticamente en tiempo real, y a estas aplicaciones o dispositivos se les deben conceder los permisos y el acceso necesarios.

Para obtener más información sobre el uso de MQTT y las funciones y permisos necesarios, consulta:

- Protocolos de comunicación del dispositivo
- Reglas para AWS IoT
- Otorgar a una AWS IoT regla el acceso que requiere
- Transfiera los permisos de rol

Antes de comenzar, compruebe lo siguiente:

Important

- Debes usar la misma AWS región al crear recursos de campañas de vehículos para el AWS IoT FleetWise. Si cambias de AWS región, es posible que tengas problemas para acceder a los recursos.
- AWS FleetWise El IoT está disponible en EE. UU. Este (Virginia del Norte) y Europa (Fráncfort).

Puede utilizarla AWS CLI para crear una función de IAM con una política de confianza para la mensajería MQTT. Para crear un rol de IAM, ejecute el siguiente comando.

Creación de un rol de IAM con una política de confianza

- *IotTopicExecutionRole*Sustitúyalo por el nombre del rol que está creando.
- *trust-policy*Sustitúyalo por el archivo JSON que contiene la política de confianza.

```
aws iam create-role --role-name IotTopicExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "mqttTopicTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:iotfleetwise:region:account-id:campaign/campaign-name"
          ],
          "aws:SourceAccount": [
            "account-id"
          ]
```

```
        }
      }
    }
  ]
}
```

Cree una política de permisos para conceder a AWS IoT FleetWise permisos para publicar mensajes en el tema de MQTT que especificó. Para crear una política de permisos, ejecute el siguiente comando.

Creación de una política de permisos

- *AWSIoTFleetwiseAccessIotTopicPermissionsPolicy*Sustitúyala por el nombre de la política que estás creando.
- *permissions-policy*Sustitúyalo por el nombre del archivo JSON que contiene la política de permisos.

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessIotTopicPermissionsPolicy --
policy-document file://permissions-policy.json
```

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Publish"
        ],
        "Resource": [
          "topic-arn"
        ]
      }
    ]
}
```

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.
- *IotTopicExecutionRole*Sustitúyalo por el nombre del rol de IAM que creaste.

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

Para obtener más información, consulte Administración de accesos para recursos de AWS en la Guía del usuario de IAM.

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 región de AWS .

```
{
  "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, consulta Cómo deshabilitar ACLs todos los depósitos nuevos y hacer cumplir 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.

Note

Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

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), Europa (Fráncfort) y Asia Pacífico (Bombay).
- 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.
- *trust-policy*Sustitúyalo por el archivo.json que contiene 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.
- *permissions-policy*Sustitúyalo por el nombre del archivo JSON que contiene 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.

Conceda AWS IoT Device Management permiso para generar la carga útil de los comandos con AWS IoT FleetWise

Cuando utilice la función de comandos para iniciar la ejecución de un comando, AWS IoT Device Management obtendrá el comando y los parámetros del comando de la solicitud entrante. Luego, requiere permisos para acceder a FleetWise los recursos de AWS IoT para validar la solicitud y generar la carga útil. Luego, la carga útil se envía al vehículo a AWS IoT Device Management través de MQTT al tema de solicitud de comando al que esté suscrito tu vehículo.

Primero debes crear un rol de IAM que otorgue AWS IoT Device Management los permisos necesarios para generar la carga útil. A continuación, proporciona el ARN de este rol a la `CreateCommand`API mediante el `roleArn` campo. A continuación se muestran algunos ejemplos de políticas.

> **Important**
>
> Para el rol de IAM, debes usar el Región de AWS mismo rol en el que creaste el vehículo y los recursos de mando. Si cambias Región de AWS, es posible que tengas problemas para acceder a los recursos.

El rol de IAM debe tener la siguiente política de confianza.

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

```
    {
      "Sid": "RemoteCommandsTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Otorga permisos a todos los vehículos (cosas de IoT)

El siguiente ejemplo muestra cómo conceder permisos para generar la carga útil para todos los vehículos registrados como AWS IoT cosas.

> **Note**
>
> - Esta política puede ser demasiado permisiva. Utilice el principio del privilegio mínimo para asegurarse de conceder solo los permisos necesarios.
> - Para denegar los permisos en su lugar, `"Effect": "Allow"` `"Effect": "Deny"` cámbielo a la política de IAM.

En este ejemplo, sustituya:

- *<AWS_REGION>*con su Región de AWS lugar de uso de los FleetWise recursos de AWS IoT.
- *<ACCOUNT_ID>*con tu Cuenta de AWS número.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        }
    ]
}
```

Otorgar permiso a un vehículo específico (cosa de IoT)

El siguiente ejemplo muestra cómo conceder permisos para generar la carga útil de un vehículo específico registrado como tal AWS IoT .

En este ejemplo, sustituya:

- *<AWS_REGION>*con su Región de AWS lugar de uso de los FleetWise recursos de AWS IoT.
- *<ACCOUNT_ID>*con tu Cuenta de AWS número.
- *<VEHICLE_NAME>*con el nombre de IoT para tu vehículo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>"
        }
    ]
}
```

Otorga permisos a vehículos y señales específicos

El siguiente ejemplo muestra cómo conceder permisos para generar la carga útil del actuador de un vehículo específico.

En este ejemplo, sustituya:

- *<AWS_REGION>*con su Región de AWS lugar de uso de los FleetWise recursos de AWS IoT.
- *<ACCOUNT_ID>*con tu Cuenta de AWS número.
- *<VEHICLE_NAME>*con el nombre de IoT para tu vehículo.
- *<SIGNAL_FQN>*con el nombre de la señal, como*<Vehicle.actuator2>*.

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

```
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iotfleetwise:Signals": ["<SIGNAL_FQN>"]
                }
            }
        }
    ]
}
```

Otorga permisos a vehículos y plantillas estatales específicos

El siguiente ejemplo muestra cómo conceder permisos para generar la carga útil para un vehículo y una plantilla de estado específicos.

En este ejemplo, sustituya:

- *<AWS_REGION>*es tu Región de AWS lugar donde utilizas los FleetWise recursos de AWS IoT.
- *<ACCOUNT_ID>*es tu Cuenta de AWS número.
- *<VEHICLE_NAME>*es el nombre de IoT para su vehículo.
- *<STATE_TEMPLATE_ID>*con el identificador de la plantilla de su estado.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": [
                "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
                "arn:aws:iotfleetwise:<AWS_REGION>:<ACCOUNT_ID>:state-
template/<STATE_TEMPLATE_ID>"]
        }
    ]
}
```

Otorgue permisos para usar las claves de KMS administradas por el cliente

Si has activado las claves de KMS administradas por el cliente AWS IoT FleetWise, en el siguiente ejemplo se muestra cómo conceder permisos para generar la carga útil.

En este ejemplo, sustituya:

- *<AWS_REGION>*con su Región de AWS lugar de uso de los FleetWise recursos de AWS IoT.
- *<ACCOUNT_ID>*con tu Cuenta de AWS número.
- *<KMS_KEY_ID>*con el ID de su clave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:<AWS_REGION>:<ACCOUNT_ID>:key/<KMS_KEY_ID>"
        }
    ]
}
```

Identity and Access Management 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 AWS los 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

- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo FleetWise funciona el AWS IoT con IAM
- Ejemplos de políticas basadas en la identidad para el IoT AWS FleetWise
- Solución de problemas AWS de FleetWise identidad y acceso a IoT
- AWS Referencia de acciones y recursos de la FleetWise API de IoT

Público

La forma en que se usa 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 ayudarle 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 gestionador 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 gestionador 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 la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes 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 Autenticación multifactor AWS en IAM en 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 utiliza 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, consulta 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 utiliza 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, consulta ¿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 puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos 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 de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM 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 persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario

a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol 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 de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el 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, consulta 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, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual 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 AWS ellas, se te considera principal. 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: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo 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 que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon 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 utiliza 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 obtener 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. Los 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 () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs 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 compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la 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). Puedes 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 (SCPs): SCPs 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 administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus 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 sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- 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 se AWS determina 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
ACLs	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No

Característica de IAM	AWS FleetWise Soporte de IoT
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 asociada. Para obtener 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

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los 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 basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones políticas para el AWS IoT FleetWise

Compatibilidad con las acciones 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 `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

Compatibilidad con los 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 puedes 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). Puedes 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, utiliza 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 ARNs respectivos tipos, 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

Compatibilidad con 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 puedes 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. Puedes crear expresiones condicionales que utilizan 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 puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes 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, consulta 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 (ACLs) en AWS IoT FleetWise

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs 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

Compatibilidad con 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 Definición de permisos con la autorización de 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

Compatibilidad con 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 obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (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 obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales de servicios cruzados para IoT AWS FleetWise

Admite sesiones de acceso directo (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

Compatibilidad 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 puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta 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 Supervisa el 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 funciones vinculadas a servicios, consulte los AWS servicios que funcionan con IAM y busque los servicios con la palabra Sí en la columna Funciones vinculadas 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 vinculado al servicio de AWSService RoleForIo TFleet Wise confía en los siguientes servicios para que lo asuman:

- `IoTFleetWise`

La política de permisos de roles denominada AWSIo TFleetwise ServiceRolePolicy 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 Configura tus FleetWise ajustes AWS de IoT.

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 vinculado al servicio de AWSService RoleForIo TFleet Wise 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 puedes añadir las políticas de IAM a roles y los usuarios puedes 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 (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por AWS IoT FleetWise, incluido el ARNs formato de 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, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea 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, consulta 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, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

- Utiliza 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 Validación de políticas con el Analizador de acceso de IAM en la Guía del 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 exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte 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 el 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 obtener 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 gestionadas 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 ejemplos de políticas, consulte. Configura tus FleetWise ajustes AWS de IoT

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
- No estoy autorizado a realizar lo siguiente: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis FleetWise recursos de AWS IoT

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 gestionador 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 gestionador 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 admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede utilizar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener 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, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

AWS Referencia de acciones y recursos de la FleetWise API de IoT

Cuando gestione el acceso mediante políticas para una identidad de IAM, puede utilizar la siguiente tabla como referencia. En la tabla se muestra cada FleetWise API de AWS IoT, las acciones correspondientes para las que puede conceder permisos para realizar la acción y el AWS recurso para el que puede conceder los permisos.

Especifique las acciones en el campo `Action` de la política y el valor del recurso en el campo `Resource` de la política. Para especificar una acción, usa el `iotfleetwise:` prefijo seguido del nombre de la acción. Por ejemplo, `iotfleetwise:CreateSignalCatalog`.

Actualmente, AWS IoT FleetWise admite las siguientes acciones, recursos y claves de condición.

Usa la barra de desplazamiento para ver el resto de la tabla.

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
AssociateVehicleFleet	IoT en cuanto a flota: AssociateVehicleFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id* arn ::iotfleetwise*partition* : :vehicle/ *region account_id vehicle-id*
CreateCampaign	IoT en cuanto a flota: CreateCampaign	arn :iotfleetwise: :campaign/ *partition region account_id campaign-name* arn ::iotfleetwise*partition* : :signal-catalog/ *region account_id name*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
		arn ::iotfleetwise*partition* : ::vehicle/ *region account_id vehicle-id* arn ::iotfleetwise: :fleet/ *partition region account_id fleet-id*
CreateDecoderManifest	IoT en cuanto a flota: CreateDecoderManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name* ejecutar ::iotfleetwise: *partition* ::model-manifest/ *region account_id name*
CreateFleet	iotfleetwise: CreateFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id* arn ::iotfleetwise*partition* : :signal-catalog/ *region account_id name*
CreateModelManifest	iotfleetwise: CreateModelManifest	arn :iotfleetwise: :model-manifest/ *partition region account_id name* arn ::iotfleetwise*partition* : :signal-catalog/ *region account_id name*
CreateSignalCatalog	iotfleetwise: CreateSignalCatalog	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*
CreateStateTemplate	iotfleetwise: CreateStateTemplate	arn:: iotfleetwise: :state-template/ *partition region account_id state-template-id* arn ::iotfleetwise*partition* : :signal-catalog/ *region account_id name*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
CreateVehicle	iotfleetwise: CreateVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise*partition* : :model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name*
BatchCreateVehicle	IoT Fleetwise: CreateVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise*partition* : :model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name*
DeleteCampaign	IoT Fleetwise: DeleteCampaign	arn :iotfleetwise: :campaign/ *partition region account_id campaign-name*
DeleteDecoderManifest	IoT Fleetwise: DeleteDecoderManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
DeleteFleet	IoT Fleetwise: DeleteFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id*
DeleteModelManifest	IoT en cuanto a flota: DeleteModelManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
DeleteSignalCatalog	IoT Fleetwise: DeleteSignalCatalog	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
DeleteStateTemplate	iotfleetwise: DeleteStateTemplate	arn:: iotfleetwise: :state-template/ *partition region account_id state-template-id*
DeleteVehicle	IoT en cuanto a flota: DeleteVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id*
DisassociateVehicleFleet	IoT en cuanto a flota: DisassociateVehicleFleet	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise: :fleet/ *partition region account_id fleet-id*
GetCampaign	IoT en cuanto a flota: GetCampaign	arn :iotfleetwise: :campaign/ *partition region account_id campaign-name*
GetDecoderManifest	IoT Fleetwise: GetDecoderManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
GetEncryptionConfiguration	IoT Fleetwise: GetEncryptionConfiguration	
GetFleet	IoT en cuanto a flota: GetFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id*
GetLoggingOptions	IoT en cuanto a flota: GetLoggingOptions	
GetModelManifest	IoT en cuanto a flota: GetModelManifest	arn :iotfleetwise: :model-manifest/ *partition region account_id name*
GetRegisterAccountStatus	iotfleetwise: GetRegisterAccountStatus	

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
GetSignalCatalog	IoT en cuanto a flota: GetSignalCatalog	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*
GetStateTemplate	iotfleetwise: GetStateTemplate	arn:: iotfleetwise: :state-template/ *partition region account_id state-template-id*
GetVehicle	IoT en cuanto a flota: GetVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id*
GetVehicleStatus	IoT en cuanto a flota: GetVehicleStatus	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id*
ImportDecoderManifest	IoT en cuanto a flota: ImportDecoderManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
ImportSignalCatalog	IoT Fleetwise: ImportSignalCatalog	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*
ListCampaigns	iotfleetwise: ListCampaigns	
ListDecoderManifestNetworkInterfaces	IoT en cuanto a flota: ListDecoderManifestNetworkInterfaces	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
ListDecoderManifests	IoT Fleetwise: ListDecoderManifests	
ListDecoderManifestSignals	IoT en cuanto a flota: ListDecoderManifestSignals	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
ListFleets	IoT Fleetwise: ListFleets	

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
ListFleetsForVehicle	IoT en cuanto a flota: ListFleetsForVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id*
ListModelManifestNodes	IoT en cuanto a flota: ListModelManifestNodes	arn :iotfleetwise: :model-manifest/ *partition region account_id name*
ListModelManifests	iotfleetwise: ListModelManifests	
ListSignalCatalogNodes	IoT en cuanto a flota: ListSignalCatalogNodes	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*
ListSignalCatalogs	iotfleetwise: ListSignalCatalogs	
ListStateTemplates	IoT en cuanto a flota: ListStateTemplates	
ListVehicles	IoT en cuanto a flota: ListVehicles	
ListVehiclesInFleet	IoT en cuanto a flota: ListVehiclesInFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
ListTagsForResource	IoT en cuanto a flota: ListTagsForResource	arn :iotfleetwise: :signal-catalog/ *partition region account_id name* arn :iotfleetwise: *partition* ::model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name* ejecutar ::iotfleetwise*partition* : ::vehicle/ *region account_id vehicle-id* arn ::iotfleetwise: :fleet/ *partition region account_id fleet-id* arn ::iotfleetwise*partition* : :campaign/ *region account_id campaign-name* arn ::iotfleetwise*partition* : :state-template/ *region account_id state-template-id*
PutEncryptionConfiguration	IoT en cuanto a flota: PutEncryptionConfiguration	
PutLoggingOptions	IoT en cuanto a flota: PutLoggingOptions	
RegisterAccount	IoT en cuanto a flota: RegisterAccount	

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
TagResource	IoT en cuanto a flota: TagResource	arn :iotfleetwise: :signal-catalog/ *partition region account_id name* arn :iotfleetwise: *partition* ::model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name* ejecutar ::iotfleetwise*partition* : ::vehicle/ *region account_id vehicle-id* arn ::iotfleetwise: :fleet/ *partition region account_id fleet-id* arn ::iotfleetwise*partition* : :campaign/ *region account_id campaign-name* arn ::iotfleetwise*partition* : :state-template/ *region account_id state-template-id*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
UntagResource	IoT en cuanto a flota: UntagResource	arn :iotfleetwise: :signal-catalog/ *partition region account_id name* arn :iotfleetwise: *partition* ::model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name* ejecutar ::iotfleetwise*partition* : ::vehicle/ *region account_id vehicle-id* arn ::iotfleetwise: :fleet/ *partition region account_id fleet-id* arn ::iotfleetwise*partition* : :campaign/ *region account_id campaign-name* arn ::iotfleetwise*partition* : :state-template/ *region account_id state-template-id*
UpdateCampaign	IoT en cuanto a flota: UpdateCampaign	arn :iotfleetwise: :campaign/ *partition region account_id campaign-name*
UpdateDecoderManifest	IoT Fleetwise: UpdateDecoderManifest	arn :iotfleetwise: :decoder-manifest/ *partition region account_id name*
UpdateFleet	IoT Fleetwise: UpdateFleet	arn :iotfleetwise: :fleet/ *partition region account_id fleet-id*
UpdateModelManifest	IoT en cuanto a flota: UpdateModelManifest	arn :iotfleetwise: :model-manifest/ *partition region account_id name*

AWS FleetWise Acciones de IoT	Permisos necesarios	Recursos
UpdateSignalCatalog	iotfleetwise: UpdateSignalCatalog	arn :iotfleetwise: :signal-catalog/ *partition region account_id name*
UpdateStateTemplate	iotfleetwise: UpdateStateTemplate	arn:: iotfleetwise: :state-template/ *partition region account_id state-template-id*
UpdateVehicle	IoT en cuanto a flota: UpdateVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise*partition* : :model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name*
BatchUpdateVehicle	IoT Fleetwise: UpdateVehicle	arn ::iotfleetwise: :vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise*partition* : :model-manifest/ *region account_id name* arn ::iotfleetwise*partition* : :decoder-manifest/ *region account_id name*

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:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la 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.
> Amazon Timestream no está disponible en la región Asia Pacífico (Bombay).

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 principal de IAM. También puedes 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 utilizar FleetWise las políticas de AWS IoT para controlar el acceso desde puntos de conexión específicos o específicos de Amazon Virtual Private Cloud (Amazon VPC). VPCs 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 final de la VPC está representado por una o más interfaces de red elásticas (ENIs) 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 Puntos de conexión de VPC de interfaz (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.

> **Note**
>
> AWS El IoT FleetWise admite todos los puntos finales de VPC con el modo de doble pila. Para obtener información sobre los puntos de enlace de servicio, consulte puntos de enlace y FleetWise cuotas de AWS IoT.

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 utilizan 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 un usuario *lijuan* de la cuenta. AWS *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 AWS IoT FleetWise. Cuando se conecta a un punto final, esta política otorga acceso a las FleetWise acciones de AWS IoT enumeradas para el usuario de IAM *fleetWise* en el Cuenta de AWS *123456789012*.

```
{
    "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 el 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, plantéeselas 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 y los FleetWise recursos IDs 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 nombres IDs de los vehículos y las 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 “CloudTrail registros”.

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 .

Supervise el AWS IoT FleetWise

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento del AWS IoT FleetWise y sus otras AWS soluciones. AWS proporciona las siguientes herramientas de supervisión para vigilar el AWS IoT FleetWise, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch supervisa tus AWS recursos y las aplicaciones en las que ejecutas 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, puedes CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Logs se puede utilizar para supervisar, almacenar y acceder a tus archivos de registro desde EC2 instancias de Amazon y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcanzan ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.
- 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. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la AWS CloudTrail Guía del usuario de .

Supervisa el AWS IoT FleetWise con Amazon CloudWatch

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

CloudWatch Las métricas de Amazon son una forma de supervisar tus AWS recursos y su rendimiento. AWS IoT FleetWise envía métricas a CloudWatch. Puedes usar la AWS Management Console AWS CLI, la o una API para enumerar las métricas a las que FleetWise envía AWS IoT CloudWatch. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

> **Important**
>
> Debe configurar los ajustes para que el AWS IoT FleetWise pueda enviar métricas a CloudWatch. Para obtener más información, consulte Configura tus FleetWise ajustes AWS de IoT.

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: None Estadísticas válidas: Sum
MessageThrottled	Se FleetWise limitó un mensaje enviado desde el vehículo al AWS IoT. Esto se debe a que ha superado los límites de servicio de esta cuenta en la región actual. Unidades: recuento Dimensiones: None Estadísticas válidas: Sum
ModelingError	Un mensaje enviado desde el vehículo y recibido por el AWS IoT FleetWise contiene

Métrica	Descripción
	señales que no se validan con el modelo del vehículo. Unidades: recuento Dimensiones: ModelName StateTemplateName (opcional), SignalCatalogName (opcional)
DecodingError	Un mensaje enviado desde el vehículo y recibido por el AWS IoT FleetWise contiene señales que no se decodifican según el manifiesto del decodificador del vehículo. Unidades: recuento Dimensiones: DecoderName Estadísticas válidas: Sum
MessageSizeLimitExceeded	Se descartó un mensaje enviado desde el vehículo a AWS IoT FleetWise . Esto se debe a que has superado el límite máximo de un servicio de mensajes para esta cuenta en la región actual. Unidades: recuento Dimensiones: None Estadísticas válidas: Sum

Métricas de vehículos

Métrica	Descripción
VehicleNotFound	Un mensaje recibido por AWS IoT FleetWise, donde se desconoce el vehículo.

Métrica	Descripción
	Unidades: recuento Dimensiones: None Estadísticas válidas: Sum

Métricas de campaña

Métrica	Descripción
CampaignInvalid	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, donde la campaña no es válida. Unidades: recuento Dimensiones: CampaignName Estadísticas válidas: Sum
CampaignNotFound	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que se desconoce la campaña. Unidades: recuento Dimensiones: CampaignName Estadísticas válidas: Sum

Métricas de plantillas estatales

Métrica	Descripción
NoStateTemplatesAssociated	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que no hay plantillas de estado asociadas al vehículo.

Métrica	Descripción
	Unidades: recuento Estadísticas válidas: Sum

Métricas de destino de los datos de campaña

Métrica	Descripción
TimestreamWriteError	AWS El IoT no FleetWise pudo escribir un mensaje desde el vehículo a la tabla de Amazon Timestream. Unidades: recuento Dimensiones:, DatabaseName TableName Estadísticas válidas: Sum
S3 WriteError	AWS El IoT no FleetWise pudo escribir un mensaje desde el vehículo al depósito de Amazon Simple Storage Service (Amazon S3). Unidades: recuento Dimensiones: BucketName Estadísticas válidas: Sum
S3 ReadError	AWS El IoT no FleetWise pudo leer una clave de objeto del vehículo en el depósito de Amazon Simple Storage Service (Amazon S3). Unidades: recuento Dimensiones: BucketName Estadísticas válidas: Sum

Métricas AWS KMS clave gestionadas por el cliente

Métrica	Descripción
KMSKeyAccessDenied	AWS El IoT no FleetWise pudo escribir un mensaje desde el vehículo en la tabla Timestream o en el bucket de Amazon S3 debido a un error de acceso a la AWS KMS clave denegada. Unidades: recuento Dimensiones: ID KMSKey Estadísticas válidas: suma

Supervise el AWS IoT FleetWise con Amazon CloudWatch Logs

> **Important**
>
> El acceso a ciertas FleetWise funciones de AWS IoT está actualmente restringido. Para obtener más información, consulte AWS Disponibilidad regional y de funciones en el AWS IoT FleetWise.

Amazon CloudWatch Logs supervisa los eventos que se producen en sus recursos y le avisa si hay algún problema. Si recibe una alerta, puede acceder a los archivos de registro para obtener información sobre el evento en cuestión. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.

Ver FleetWise los registros de AWS IoT en la CloudWatch consola

> **Important**
>
> Antes de que pueda ver el grupo de FleetWise registros de AWS IoT en la CloudWatch consola, asegúrese de que se cumpla lo siguiente:
>
> - Has activado el inicio de sesión en AWS IoT FleetWise. Para obtener más información acerca del registro, consulte Configurar el FleetWise registro AWS de IoT.

- Ya hay entradas de registro escritas por AWS IoT las operaciones.

Para ver los FleetWise registros de AWS IoT en la CloudWatch consola

1. Abra la consola de CloudWatch .
2. En el panel de navegación, elija Registros, Grupos de registros.
3. Elija el grupo de registro.
4. Elija Buscar grupos de registro. 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 filtros y patrones en la Guía del usuario de Amazon CloudWatch Logs.

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 el AWS IoT FleetWise contiene señales que no se validan con el modelo del vehículo. Atributos: VehicleName, CampaignName (opcional) signalCatalogName, SignalID (opcional), SignalValue (opcional), signalVal ueRange Min (opcional), signalValueRange Max (opcional), SignalID, modelManifestName stateTemplateName
ILLEGAL_MESSAGE_FROM_EDGE	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise no coincidía con el formato requerido. Atributos: nombre del vehículo, nombre de la campaña, signalCatalogName
DECODING_ERROR	Un mensaje enviado desde el vehículo y recibido por el AWS IoT FleetWise contiene señales que no se decodifican según el manifiesto del decodificador del vehículo. Atributos: CampaignName, signalCat alogName, (opcional) SignalName decoderMa nifestName, (opcional) S3uri
MESSAGE_THROTTLED	Se FleetWise limitó un mensaje enviado desde el vehículo al AWS IoT. Esto se debe a que ha superado los límites de servicio de esta cuenta en la región actual. Atributos: AccountID, VehicleName, message, EventType, LogLevel, timestamp

Tipo de evento	Descripción
SE HA SUPERADO EL LÍMITE DE TAMAÑO DEL MENSAJE	Un mensaje enviado desde el vehículo y recibido mediante AWS IoT FleetWise supera el tamaño máximo del límite de un servicio de mensajes. Atributos: AccountID, VehicleName

Tipos de eventos de vehículos

Tipo de evento	Descripción
VEHICLE_NOT_FOUND	Un mensaje recibido por AWS IoT FleetWise, donde se desconocía el vehículo. Atributos: VehicleName, CampaignName (opcional), (opcional) stateTemplateName

Tipos de eventos de campaña

Tipo de evento	Descripción
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, donde 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 El IoT no FleetWise pudo escribir un mensaje desde el vehículo a la tabla de Amazon Timestream. Atributos: Nombre del vehículo, Nombre de la campaña, timestreamDatabaseName timestrea mTableName
S3_WRITE_ERROR	AWS El IoT no FleetWise pudo escribir un mensaje desde el vehículo al depósito de Amazon Simple Storage Service (Amazon S3). Atributos: campaignName, destinationName
S3_READ_ERROR	AWS El IoT no FleetWise pudo leer una clave de objeto del vehículo en el depósito de Amazon Simple Storage Service (Amazon S3). Atributos: campaignName, destinationName

Tipos de eventos de plantillas de estados

Tipo de evento	Descripción
STATE_TEMPLATE_NOT_FOUND	Un mensaje enviado desde el vehículo y recibido por AWS IoT FleetWise, en el que se desconocía la plantilla de estado. Atributos: nombre del vehículo (opcional), stateTemplateName

Tipos de eventos AWS KMS clave gestionados por el cliente

Tipo de evento	Descripción
KMS_KEY_ACCESS_DENIED	AWS El IoT no FleetWise pudo escribir un mensaje desde el vehículo en la tabla Timestream o en el bucket de Amazon S3 debido a un error de acceso a la AWS KMS clave denegada. Atributos: kmsKeyId (opcional), ResourceArn (opcional)

Atributos

Todas las entradas CloudWatch de los registros incluyen los siguientes atributos:

accountId

Tu Cuenta de AWS ID.

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 de milisegundos de la época en la que el AWS IoT FleetWise procesó el registro.

Atributos opcionales

CloudWatch Las entradas de los registros incluyen opcionalmente estos atributos, en función de: `eventType`

decoderManifestName

El nombre del manifiesto del decodificador que contiene la señal.

destinationName

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

Una lista de señales de error IDs.

signalName

El nombre de la señal.

signalTimestampEpochSrta.

Marca de tiempo de la señal de error.

signalValue

Valor de la señal de error.

signalValueRangeMax

Rango máximo de la señal de error.

signalValueRangeMín

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.

Configurar el FleetWise registro AWS de IoT

Puede enviar sus datos de FleetWise registro de AWS IoT a un grupo de CloudWatch registros. CloudWatch Los registros proporcionan visibilidad en caso de que el AWS FleetWise IoT no procese los mensajes de los vehículos. Por ejemplo, esto puede ocurrir debido a una configuración defectuosa o a otros errores del cliente. Se le notificará cualquier error para que pueda identificar y mitigar los problemas.

Para poder enviar los registros a CloudWatch, debe crear un grupo de CloudWatch registros. Configure el grupo de registros con la misma cuenta y en la misma región que utilizó con AWS IoT FleetWise. Cuando habilite el inicio de sesión en AWS IoT FleetWise, proporcione el nombre del grupo de registros. Una vez que se habilita el registro, el AWS CloudWatch IoT FleetWise entrega los registros al grupo de registros en flujos de registros.

Puede ver los datos de registro enviados desde el AWS IoT FleetWise en la CloudWatch consola. Para obtener más información sobre la configuración de un grupo de CloudWatch registros y la visualización de los datos de registro, consulte Trabajar con grupos de registros.

Permisos para publicar registros en CloudWatch

La configuración del registro para un grupo de CloudWatch registros requiere la configuración de permisos descrita en esta sección. Para obtener información sobre la administración de permisos, consulte Administración del acceso a los AWS recursos en la Guía del usuario de IAM.

Con estos permisos, puede cambiar la configuración de registro, configurar la entrega de CloudWatch registros y recuperar información sobre su grupo de registros.

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

Cuando se permiten acciones en todos AWS los recursos, se indica en la política con una `"Resource"` configuración de`"*"`. Esto significa que las acciones están permitidas en todos los AWS recursos compatibles con cada acción.

Configurar el registro en AWS IoT FleetWise (consola)

En esta sección se describe cómo usar la FleetWise consola de AWS IoT para configurar el registro.

Para usar la FleetWise consola de AWS IoT para configurar el registro

1. Abre la FleetWiseconsola AWS de IoT.
2. En el panel izquierdo, seleccione Configuración.
3. En la sección Registro de la página Configuración, elija Editar.
4. En la sección de CloudWatch registro, introduzca el grupo de registros.
5. Para guardar los cambios, elija Enviar.

Tras activar el registro, podrá ver los datos de registro en la CloudWatch consola.

Configurar el 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 mediante los métodos de la AWS API que corresponden a los comandos CLI que se muestran aquí. Puede utilizar la operación de GetLoggingOptionsAPI para obtener la configuración actual y la operación de PutLoggingOptionsAPI para modificarla.

Para usar 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

   El tipo de registro para enviar datos a CloudWatch los registros. Para deshabilitar el registro, cambie el valor a `OFF`.

logGroupName

El grupo CloudWatch de registros al que la operación envía los datos. Asegúrese de crear el nombre del grupo de registros antes de habilitar el registro para AWS IoT FleetWise.

Después de habilitar el registro, consulte Buscar entradas de registro mediante la AWS CLI.

Registre AWS las llamadas a FleetWise la API de IoT mediante AWS CloudTrail

AWS FleetWise El IoT está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en el AWS IoT FleetWise. CloudTrail captura todas las llamadas a la API para el AWS IoT FleetWise como eventos. Las llamadas capturadas incluyen llamadas desde la FleetWise consola de AWS IoT y llamadas en código a las operaciones de la FleetWise API de AWS IoT. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para el AWS IoT FleetWise. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a AWS IoT FleetWise, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario.

AWS FleetWise Información sobre IoT en CloudTrail

CloudTrail está habilitada en tu AWS cuenta al crearla. Cuando se produce actividad en el AWS IoT FleetWise, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS . Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los relacionados con el AWS IoT FleetWise, crea un registro. Un rastro permite CloudTrail entregar 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. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a

fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail servicios e integraciones compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones
- Recibir archivos de CloudTrail registro de varias cuentas

Todas FleetWise las acciones de AWS IoT se registran CloudTrail y se documentan en la Referencia de FleetWise API de AWS IoT. Por ejemplo, las llamadas a las `CreateCampaign` `GetModelManifest` acciones y las llamadas generan entradas en los archivos de CloudTrail registro. `AssociateVehicleFleet`

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 AWS servicio.

Para obtener más información, consulte el elemento userIdentity de CloudTrail .

Comprenda AWS las entradas de los archivos de FleetWise registro de IoT

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la *`AssociateVehicleFleet`* operación.

```
{
```

```
    "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 FleetWise desarrolladores de AWS IoT

En la siguiente tabla se describen las versiones de documentación para AWS IoT FleetWise.

Cambio	Descripción	Fecha
Ampliación de las regiones	AWS FleetWise El IoT ya está disponible en la región de Asia Pacífico (Bombay) (solo con acceso cerrado).	21 de noviembre de 2024
Disponibilidad general limitada de las nuevas funciones	AWS El IoT FleetWise ahora admite el acceso restringido para que las campañas almacenen y reenvíen datos, configuren un tema MQTT como destino de datos y recopilen datos de códigos de diagnóstico de problemas. Ahora también admite el acceso restringido para la recopilación de datos independiente de la red mediante interfaces de decodificación personalizadas, la configuración de comandos y la supervisión del último estado conocido de los vehículos.	21 de noviembre de 2024
Envía los datos de la campaña a un tema de MQTT	AWS El IoT FleetWise ahora permite enviar los datos recopilados durante las campañas a un tema de MQTT que especifique,	1 de mayo de 2024

	además de la capacidad de almacenar los datos en Amazon S3 o Amazon Timestream.	
Vista previa de los datos de sistemas de visión	Puede utilizar la vista previa de los datos del sistema de visión del AWS IoT FleetWise para recopilar y organizar los datos de los sistemas de visión de los vehículos , incluidos los de cámaras, radares y LIDAR. 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
AWS KMS claves gestionadas por el cliente	AWS El IoT FleetWise ahora admite claves gestionadas por el AWS KMS cliente. Puede usar la clave KMS para cifrar los datos del lado del servidor relacionados con los FleetWise recursos de AWS IoT (catálogo de señales, modelo de vehículo, manifiesto del decodificador, vehículos y configuraciones de campañas de recopilación de datos) almacenados en. Nube de AWS	16 de octubre de 2023

Almacenamiento de objetos en Amazon S3	AWS El IoT FleetWise ahora admite el almacenamiento de datos mediante 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 preliminar de la Guía para desarrolladores 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.