



Entwicklerhandbuch

# AWS IoT FleetWise



# AWS IoT FleetWise: Entwicklerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Was ist AWS IoT FleetWise? .....	1
Vorteile .....	2
Anwendungsfälle .....	2
Sind Sie neu im AWS Internet der Dinge FleetWise? .....	3
Zugriff auf AWS IoT FleetWise .....	3
Preise für AWS IoT FleetWise .....	3
So FleetWise funktioniert AWS IoT .....	4
Die wichtigsten Konzepte .....	4
Funktionen von AWS IoT FleetWise .....	8
Zugehörige Services .....	9
Einrichtung AWS IoT FleetWise .....	10
Richten Sie Ihre ein AWS-Konto .....	10
Melde dich an für ein AWS-Konto .....	10
Erstellen eines Benutzers mit Administratorzugriff .....	11
Erste Schritte in der Konsole .....	12
Einstellungen konfigurieren .....	13
Konfigurieren der Einstellungen (Konsole) .....	13
Einstellungen konfigurieren (AWS CLI) .....	14
Erste Schritte .....	16
Voraussetzungen .....	16
Verwenden der Edge Agent-Softwaredemo .....	16
Erste Schritte (Konsole) .....	17
Voraussetzungen .....	18
Schritt 1: Richten Sie die Edge Agent-Software für AWS IoT ein FleetWise .....	18
Schritt 2: Erstellen Sie ein Fahrzeugmodell .....	20
Schritt 3: Erstellen Sie ein Decoder-Manifest .....	22
Schritt 4: Konfigurieren Sie ein Decoder-Manifest .....	23
Schritt 5: Erstellen Sie ein Fahrzeug .....	24
Schritt 6: Erstelle eine Kampagne .....	25
Schritt 7: Bereinigen .....	27
Nächste Schritte .....	27
Daten in die Cloud aufnehmen .....	28
Fahrzeuge modellieren .....	31
Signalkataloge .....	34

Signale konfigurieren .....	37
Erstellen AWS CLI Sie einen Signalkatalog () .....	43
Importieren Sie einen Signalkatalog .....	48
Aktualisieren Sie einen Signalkatalog (AWS CLI) .....	58
Löscht einen Signalkatalog (AWS CLI) .....	60
Ruft Informationen zum Signalkatalog ab (AWS CLI) .....	61
Fahrzeugmodelle .....	61
Erstellen Sie ein Fahrzeugmodell .....	62
Aktualisieren Sie ein Fahrzeugmodell ()AWS CLI .....	69
Löschen Sie ein Fahrzeugmodell .....	70
Informationen zum Fahrzeugmodell abrufen (AWS CLI) .....	71
Decoder-Manifeste .....	72
Konfigurieren Sie Netzwerkschnittstellen und Decodersignale .....	74
Erstellen Sie ein Decoder-Manifest .....	77
Aktualisieren Sie ein Decoder-Manifest ()AWS CLI .....	85
Löschen Sie ein Decoder-Manifest .....	85
Ruft Informationen zum Decoder-Manifest ab ()AWS CLI .....	87
Fahrzeuge .....	89
Fahrzeuge bereitstellen .....	90
Fahrzeuge authentifizieren .....	91
Fahrzeuge autorisieren .....	93
Reservierte Themen .....	94
Erstelle ein Fahrzeug .....	96
Erstellen Sie ein Fahrzeug (Konsole) .....	96
Erstellen Sie ein Fahrzeug (AWS CLI) .....	99
Erstellen Sie mehrere Fahrzeuge (AWS CLI) .....	101
Ein Fahrzeug aktualisieren (AWS CLI) .....	102
Aktualisieren Sie mehrere Fahrzeuge (AWS CLI) .....	103
Ein Fahrzeug löschen .....	104
Lösche ein Fahrzeug (Konsole) .....	105
Lösche ein Fahrzeug (AWS CLI) .....	105
Fahrzeuginformationen abrufen ()AWS CLI .....	105
Flotten .....	107
Erstelle eine Flotte (AWS CLI) .....	108
Ordnen Sie ein Fahrzeug einer Flotte zu (AWS CLI) .....	109
Ein Fahrzeug von einer Flotte trennen (AWS CLI) .....	109

Aktualisieren Sie eine Flotte (AWS CLI) .....	110
Löschen Sie eine Flotte (AWS CLI) .....	110
Flotteninformationen abrufen (AWS CLI) .....	111
Kampagnen .....	113
Erstellen einer Kampagne .....	118
Erstellen Sie eine Kampagne (Konsole) .....	119
Erstelle eine Kampagne (AWS CLI) .....	127
Logische Ausdrücke für Kampagnen .....	130
Aktualisieren Sie eine Kampagne (AWS CLI) .....	132
Löscht eine Kampagne .....	132
Löschen Sie eine Kampagne (Konsole) .....	132
Löschen Sie eine Kampagne (AWS CLI) .....	133
Kampagneninformationen abrufen () AWS CLI .....	133
Verarbeitung und Visualisierung von Fahrzeugdaten .....	135
Verarbeitung von Fahrzeugdaten in Timestream .....	135
Visualisierung der in Timestream gespeicherten Fahrzeugdaten .....	136
Verarbeitung von Fahrzeugdaten in S3 .....	136
S3-Objektformat .....	137
Analyse der in S3 gespeicherten Fahrzeugdaten .....	137
AWS CLI und AWS-SDKs .....	141
Fehlerbehebung .....	142
Probleme mit dem Decoder-Manifest .....	142
FleetWise Softwareprobleme mit dem Edge-Agent für AWS IoT .....	146
Problem: Die Edge Agent-Software startet nicht. ....	146
Problem: [FEHLER] [IoTFleetWiseEngine: :connect]: [Persistenzbibliothek konnte nicht initialisiert werden] .....	148
Problem: Die Edge Agent-Software erfasst keine PIDs und Diagnose-Fehlercodes (DTCs) für die integrierte Diagnose (OBD) II. ....	148
Problem: Die Edge Agent for AWS FleetWise IoT-Software sammelt keine Daten aus dem Netzwerk oder kann keine Dateninspektionsregeln anwenden. ....	149
Problem: [FEHLER] [AwsIotConnectivityModule: :connect]: [Verbindung mit Fehler fehlgeschlagen] oder [WARN] [AwsIotChannel: :send]: [Keine aktive MQTT-Verbindung.] ...	150
Sicherheit .....	151
Datenschutz .....	152
Verschlüsselung im Ruhezustand .....	153
Verschlüsselung während der Übertragung .....	153

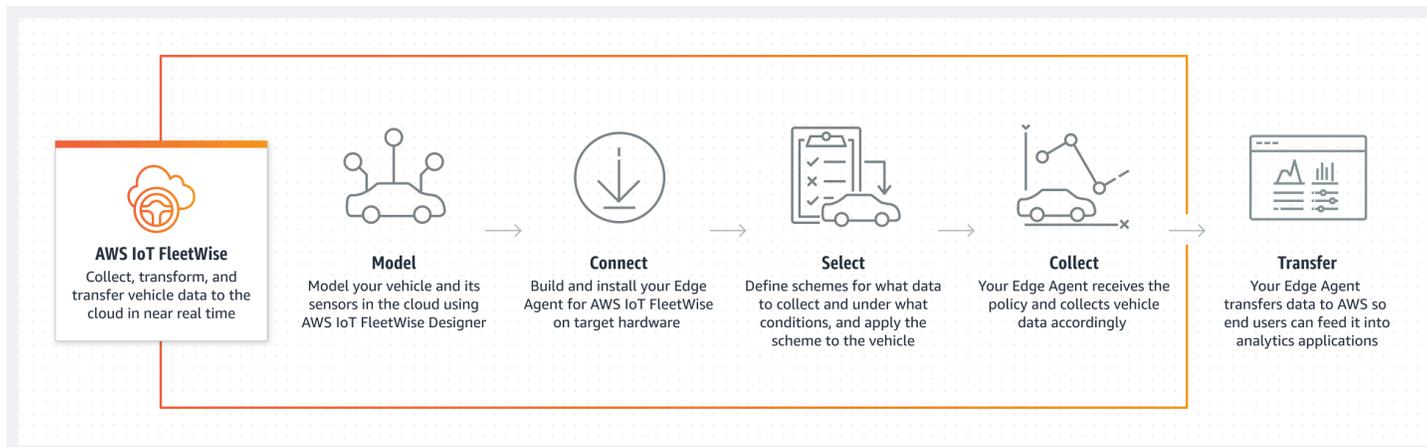
Datenverschlüsselung .....	154
Steuern des Zugriffs .....	162
Gewährung AWS IoT FleetWise Zugriff auf ein Amazon S3 S3-Ziel .....	163
Gewährung AWS IoT FleetWise Zugriff auf ein Amazon Timestream Timestream-Ziel .....	166
Identitäts- und Zugriffsverwaltung .....	169
Zielgruppe .....	169
Authentifizierung mit Identitäten .....	170
Verwalten des Zugriffs mit Richtlinien .....	174
Wie AWS IoT FleetWise funktioniert mit IAM .....	177
Beispiele für identitätsbasierte Richtlinien .....	187
Fehlerbehebung .....	190
Compliance-Validierung .....	192
Ausfallsicherheit .....	194
Sicherheit der Infrastruktur .....	194
Verbindung herstellen zu AWS IoT FleetWise über einen VPC Schnittstellenendpunkt .....	195
Konfigurations- und Schwachstellenanalyse .....	199
Bewährte Methoden für die Gewährleistung der Sicherheit .....	199
Erteilen von Mindestberechtigungen .....	199
Keine Protokollierung sensibler Informationen .....	200
Verwenden Sie AWS CloudTrail um die Anrufliste einzusehen API .....	200
Synchronisieren der internen Uhr Ihres Geräts .....	200
Überwachen .....	201
Überwachung mit CloudWatch .....	201
Überwachung mit CloudWatch Protokollen .....	205
AWS FleetWise IoT-Protokolle in der CloudWatch Konsole anzeigen .....	205
Konfigurieren der Protokollierung .....	211
CloudTrail-Protokolle .....	214
AWSIoT FleetWise Informationen in CloudTrail .....	214
Verständnis AWS IoT FleetWise Logdateieinträge .....	215
Dokumentverlauf .....	217
.....	ccxix

# Was ist AWS IoT FleetWise?

AWS IoT FleetWise ist ein verwalteter Dienst, mit dem Sie Fahrzeugdaten sammeln und in der Cloud organisieren können. Sie können die gesammelten Daten verwenden, um die Fahrzeugqualität, Leistung und Autonomie zu verbessern. Mit AWS IoT FleetWise können Sie Daten von Fahrzeugen sammeln und organisieren, die unterschiedliche Protokolle und Datenformate verwenden. AWS IoT FleetWise hilft dabei, Nachrichten auf niedriger Ebene in für Menschen lesbare Werte umzuwandeln und das Datenformat in der Cloud für Datenanalysen zu standardisieren. Sie können auch Datenerfassungskampagnen definieren, um zu kontrollieren, welche Fahrzeugdaten gesammelt werden sollen und wann diese Daten in die Cloud übertragen werden sollen.

Wenn sich die Fahrzeugdaten in der Cloud befinden, können Sie sie für Anwendungen verwenden, die den Zustand der Fahrzeugflotte analysieren. Diese Daten können Ihnen helfen, potenzielle Wartungsprobleme zu erkennen, Infotainmentsysteme im Fahrzeug intelligenter zu machen und fortschrittliche Technologien wie autonomes Fahren und Fahrerassistenzsysteme mit Analysen und maschinellem Lernen (ML) zu verbessern.

Das folgende Diagramm zeigt die grundlegende Architektur von AWS IoT FleetWise.



## Themen

- [Vorteile](#)
- [Anwendungsfälle](#)
- [Sind Sie neu im AWS Internet der Dinge FleetWise?](#)
- [Zugriff auf AWS IoT FleetWise](#)
- [Preise für AWS IoT FleetWise](#)

- [So FleetWise funktioniert AWS IoT](#)
- [Zugehörige Services](#)

## Vorteile

Die wichtigsten Vorteile von AWS IoT FleetWise sind:

Sammeln Sie Fahrzeugdaten intelligenter

Verbessern Sie die Datenrelevanz mit intelligenter Datenerfassung, bei der nur die Daten, die Sie benötigen, zur Analyse in die Cloud gesendet werden.

Analysieren Sie auf einfache Weise standardisierte, flottenweite Daten

Analysieren Sie standardisierte Daten aus einer Fahrzeugflotte, ohne ein individuelles Datenerfassungs- oder Protokollierungssystem entwickeln zu müssen.

Automatische Datensynchronisierung in der Cloud

Verschaffen Sie sich einen einheitlichen Überblick über Daten, die sowohl von Standardsensoren (Telemetriedaten) als auch von Bildverarbeitungssystemen (Daten von Kameras, Radaren und Lidaren) gesammelt wurden, und synchronisieren Sie sie automatisch in der Cloud. AWS IoT FleetWise sorgt dafür, dass sowohl strukturierte als auch unstrukturierte Bildverarbeitungssystemdaten, Metadaten und Standardsensordaten automatisch in der Cloud synchronisiert werden. Dadurch wird der Prozess optimiert, um sich ein Gesamtbild der Ereignisse zu verschaffen und Erkenntnisse zu gewinnen.

### Note

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Anwendungsfälle

Zu den Szenarien, in denen Sie AWS IoT verwenden können, FleetWise gehören die folgenden:

## Trainieren Sie KI/ML-Modelle

Verbessern Sie kontinuierlich die Modelle für maschinelles Lernen, die für autonome und fortschrittliche Fahrerassistenzsysteme verwendet werden, indem Sie Daten aus Serienfahrzeugen sammeln.

## Verbessern Sie das digitale Kundenerlebnis

Nutzen Sie Daten aus Infotainmentsystemen, um audiovisuelle Inhalte im Fahrzeug und In-App-Einblicke relevanter zu machen.

## Sorgen Sie für eine gesunde Fahrzeugflotte

Nutzen Sie Erkenntnisse aus Flottendaten, um den Zustand und den Ladezustand der EV-Batterie zu überwachen, Wartungspläne zu verwalten, den Kraftstoffverbrauch zu analysieren und vieles mehr.

# Sind Sie neu im AWS Internet der Dinge FleetWise?

Wenn Sie mit AWS IoT noch nicht vertraut sind FleetWise, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- [So FleetWise funktioniert AWS IoT](#)
- [Einrichtung AWS IoT FleetWise](#)
- [Demo der Edge Agent-Software](#)
- [Daten in die Cloud aufnehmen](#)

## Zugriff auf AWS IoT FleetWise

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um auf AWS IoT zuzugreifen FleetWise.

## Preise für AWS IoT FleetWise

Fahrzeuge senden Daten über MQTT-Nachrichten an die Cloud. Sie zahlen am Ende eines jeden Monats für die Fahrzeuge, die Sie im AWS IoT erstellt haben FleetWise. Sie zahlen auch für Nachrichten, die Sie von Fahrzeugen sammeln. Aktuelle Informationen zur Preisgestaltung finden Sie auf der Seite mit den [AWS FleetWise IoT-Preisen](#). Weitere Informationen zum MQTT-Messaging-Protokoll finden Sie unter [MQTT](#) im AWS IoT CoreDeveloper Guide.

# So FleetWise funktioniert AWS IoT

Die folgenden Abschnitte bieten einen Überblick über AWS FleetWise IoT-Servicekomponenten und deren Zusammenspiel.

Nachdem Sie diese Einführung gelesen haben, erfahren Sie im [Einrichtung AWS IoT FleetWise](#) Abschnitt, wie Sie AWS IoT einrichten FleetWise.

## Themen

- [Die wichtigsten Konzepte](#)
- [Funktionen von AWS IoT FleetWise](#)

## Die wichtigsten Konzepte

AWS IoT FleetWise bietet ein Framework zur Fahrzeugmodellierung, mit dem Sie Ihr Fahrzeug und seine Sensoren und Aktuatoren in der Cloud modellieren können. Um die sichere Kommunikation zwischen Ihrem Fahrzeug und der Cloud zu ermöglichen, bietet AWS IoT FleetWise auch eine Referenzimplementierung, die Sie bei der Entwicklung von Edge Agent-Software unterstützt, die Sie in Ihrem Fahrzeug installieren können. Sie können Datenerfassungsschemata in der Cloud definieren und sie in Ihrem Fahrzeug bereitstellen. Die in Ihrem Fahrzeug ausgeführte Edge Agent-Software verwendet Datenerfassungsschemata, um zu steuern, welche Daten erfasst und wann sie in die Cloud übertragen werden sollen.

Im Folgenden sind die Kernkonzepte von AWS IoT aufgeführt FleetWise.

### Signal

Signale sind grundlegende Strukturen, die Sie so definieren, dass sie Fahrzeugdaten und deren Metadaten enthalten. Ein Signal kann ein Attribut, ein Zweig, ein Sensor oder ein Aktuator sein. Sie können beispielsweise einen Sensor erstellen, der die Temperaturwerte im Fahrzeug empfängt und dessen Metadaten, einschließlich eines Sensornamens, eines Datentyps und einer Einheit, speichert. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#).

### Attribut

Attribute stellen statische Informationen dar, die sich im Allgemeinen nicht ändern, z. B. Hersteller und Herstellungsdatum.

## Verzweigen

Zweige stellen Signale in einer verschachtelten Struktur dar. Zweige zeigen Signalhierarchien. Zum Beispiel hat der `Vehicle` Zweig einen untergeordneten Zweig, `Powertrain`. Der `Powertrain` Zweig hat einen untergeordneten Zweig, `combustionEngine`. Verwenden Sie den `Vehicle.Powertrain.combustionEngine` Ausdruck, um den `combustionEngine` Zweig zu finden.

## Sensor

Sensordaten geben den aktuellen Zustand des Fahrzeugs an und ändern sich im Laufe der Zeit, wenn sich der Zustand des Fahrzeugs ändert, z. B. Flüssigkeitsstand, Temperaturen, Vibrationen oder Spannung.

## Aktuator

Aktuatorendaten geben Auskunft über den Zustand von Fahrzeuggeräten wie Motoren, Heizungen und Türschlössern. Durch Ändern des Zustands eines Fahrzeuggeräts können Aktuatorendaten aktualisiert werden. Sie können beispielsweise einen Aktuator definieren, der die Heizung darstellt. Der Aktuator empfängt neue Daten, wenn Sie die Heizung ein- oder ausschalten.

## Benutzerdefinierter Aufbau

Eine benutzerdefinierte Struktur (auch als Struktur bezeichnet) stellt eine komplexe Datenstruktur oder Datenstruktur höherer Ordnung dar. Sie erleichtert das logische Binden oder Gruppieren von Daten, die aus derselben Quelle stammen. Eine Struktur wird verwendet, wenn Daten in einer atomaren Operation gelesen oder geschrieben werden, z. B. um einen komplexen Datentyp oder eine Form höherer Ordnung darzustellen.

Ein Signal vom Strukturtyp wird im Signalkatalog definiert, indem ein Verweis auf einen Strukturdatentyp anstelle eines primitiven Datentyps verwendet wird. Strukturen können für alle Arten von Signalen verwendet werden, einschließlich Sensoren, Attributen, Aktuatoren und Datentypen für Bildverarbeitungssysteme. Wenn ein Signal vom Typ `Structure` gesendet oder empfangen wird, erwartet AWS IoT, dass alle enthaltenen Elemente gültige Werte haben, sodass alle Elemente obligatorisch sind. Wenn eine Struktur beispielsweise die Elemente `Vehicle.Camera.Image.Height`, `Vehicle.Camera.Image.Width` und `Vehicle.Camera.Image.Data` enthält, wird erwartet, dass das gesendete Signal Werte für all diese Elemente enthält.

**Note**

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Benutzerdefiniertes Eigentum

Eine benutzerdefinierte Eigenschaft stellt ein Element der komplexen Datenstruktur dar. Der Datentyp der Eigenschaft kann entweder primitiv oder eine andere Struktur sein.

Bei der Darstellung einer Form höherer Ordnung mithilfe einer Struktur und einer benutzerdefinierten Eigenschaft wird die beabsichtigte Form höherer Ordnung immer als Baumstruktur definiert und betrachtet. Die benutzerdefinierte Eigenschaft wird verwendet, um alle Blattknoten zu definieren, während die Struktur verwendet wird, um alle Knoten zu definieren, die keine Blattknoten sind.

## Signalkatalog

Ein Signalkatalog enthält eine Sammlung von Signalen. Signale in einem Signalkatalog können verwendet werden, um Fahrzeuge zu modellieren, die unterschiedliche Protokolle und Datenformate verwenden. Beispielsweise gibt es zwei Fahrzeuge, die von verschiedenen Autoherstellern hergestellt werden: eines verwendet das Control Area Network (CAN-Bus) - Protokoll, das andere das On-Board Diagnostics (OBD) -Protokoll. Sie können im Signalkatalog einen Sensor für den Empfang von Fahrzeugtemperaturwerten definieren. Dieser Sensor kann zur Darstellung der Thermoelemente in beiden Fahrzeugen verwendet werden. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#).

## Fahrzeugmodell (Modellmanifest)

Fahrzeugmodelle sind deklarative Strukturen, mit denen Sie das Format Ihrer Fahrzeuge standardisieren und Beziehungen zwischen Signalen in den Fahrzeugen definieren können. Fahrzeugmodelle sorgen für konsistente Informationen für mehrere Fahrzeuge desselben Typs. Sie fügen Signale hinzu, um Fahrzeugmodelle zu erstellen. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).

## Decoder-Manifest

Decoder-Manifeste enthalten Dekodierungsinformationen für jedes Signal in Fahrzeugmodellen. Sensoren und Aktuatoren in Fahrzeugen übertragen Nachrichten auf niedriger Ebene

(Binärdaten). Mit Decoder-Manifesten FleetWise ist AWS IoT in der Lage, Binärdaten in menschenlesbare Werte umzuwandeln. Jedes Decoder-Manifest ist einem Fahrzeugmodell zugeordnet. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

## Netzwerkschnittstelle

Enthält Informationen über das Protokoll, das das bordeigene Netzwerk verwendet. AWS IoT FleetWise unterstützt die folgenden Protokolle.

### Controller Area Network (CAN-Bus)

Ein Protokoll, das definiert, wie Daten zwischen elektronischen Steuergeräten (ECUs) übertragen werden. Bei Steuergeräten kann es sich um das Motorsteuergerät, Airbags oder das Audiosystem handeln.

### On-Board-Diagnose (OBD) II

Ein weiterentwickeltes Protokoll, das definiert, wie Selbstdiagnosedaten zwischen Steuergeräten übertragen werden. Es bietet eine Reihe von Standard-Diagnosefehlercodes (DTCs), mit deren Hilfe Sie feststellen können, was mit Ihrem Fahrzeug nicht stimmt.

### Middleware für Fahrzeuge

Die Fahrzeug-Middleware ist als eine Art Netzwerkschnittstelle definiert. Zu den Beispielen für Fahrzeug-Middleware gehören das Robot Operating System (ROS 2) und die skalierbare serviceorientierte Middleware over IP (SOME/IP).

#### Note

AWS IoT FleetWise unterstützt ROS 2-Middleware für Bildverarbeitungssystemdaten.

## Decodersignal

Bietet detaillierte Dekodierungsinformationen für ein bestimmtes Signal. Jedes im Fahrzeugmodell angegebene Signal muss mit einem Decodersignal gepaart werden. Wenn das Decoder-Manifest CAN-Netzwerkschnittstellen enthält, muss es CAN-Decodersignale enthalten. Wenn das Decoder-Manifest OBD-Netzwerkschnittstellen enthält, muss es OBD-Decodersignale enthalten.

Das Decoder-Manifest muss Nachrichtendecodersignale enthalten, wenn es auch Fahrzeug-Middleware-Schnittstellen enthält.

## Fahrzeug

Eine virtuelle Darstellung Ihres physischen Fahrzeugs, z. B. eines Autos oder eines Lastwagens. Fahrzeuge sind Beispiele für Fahrzeugmodelle. Fahrzeuge, die aus demselben Fahrzeugmodell hergestellt wurden, erben dieselbe Gruppe von Signalen. Jedes Fahrzeug entspricht einer AWS IoT Sache.

## Flotte

Eine Flotte steht für eine Gruppe von Fahrzeugen. Bevor Sie eine Fahrzeugflotte einfach verwalten können, müssen Sie einzelne Fahrzeuge einer Flotte zuordnen.

## Kampagne

Enthält Datenerfassungsschemata. Sie definieren eine Kampagne in der Cloud und stellen sie für ein Fahrzeug oder eine Flotte bereit. Kampagnen geben der Edge Agent-Software Anweisungen zur Auswahl, Erfassung und Übertragung von Daten in die Cloud.

## Schema der Datenerfassung

Datenerfassungsschemata geben der Edge Agent-Software Anweisungen zum Sammeln von Daten. Derzeit FleetWise unterstützt AWS IoT das zustandsbasierte Sammelschema und das zeitbasierte Sammelschema.

## Bedingungsabhängiges Sammelsystem

Verwenden Sie einen logischen Ausdruck, um zu erkennen, welche Daten gesammelt werden sollen. Die Edge Agent-Software sammelt Daten, wenn die Bedingung erfüllt ist. Wenn der Ausdruck beispielsweise lautet `$variable.myVehicle.InVehicleTemperature >35.0`, erfasst die Edge Agent-Software Temperaturwerte, die über 35,0 liegen.

## Zeitbasiertes Erfassungsschema

Geben Sie einen Zeitraum in Millisekunden an, um zu definieren, wie oft Daten gesammelt werden sollen. Wenn der Zeitraum beispielsweise 10.000 Millisekunden beträgt, erfasst die Edge Agent-Software alle 10 Sekunden Daten.

## Funktionen von AWS IoT FleetWise

Im Folgenden sind die wichtigsten Funktionen von AWS IoT aufgeführt FleetWise.

## Modellierung von Fahrzeugen

Erstellen Sie virtuelle Darstellungen Ihrer Fahrzeuge und wenden Sie ein einheitliches Format an, um Fahrzeugsignale zu organisieren. AWS IoT FleetWise unterstützt die [Vehicle Signal Specification \(VSS\)](#), mit der Sie Fahrzeugsignale standardisieren können.

## Schemabasierte Datenerfassung

Definieren Sie Schemata, um nur hochwertige Fahrzeugdaten in die Cloud zu übertragen. Sie können zustandsabhängige Schemata definieren, um zu steuern, welche Daten erfasst werden sollen, z. B. Daten, die im Fahrzeug über 40 Grad liegen. Sie können auch zeitbasierte Schemata definieren, um zu steuern, wie oft Daten erfasst werden.

## Edge-Agent für AWS FleetWise IoT-Software

Die in Fahrzeugen ausgeführte Edge Agent-Software erleichtert die Kommunikation zwischen Fahrzeugen und der Cloud. Während Fahrzeuge mit der Cloud verbunden sind, empfängt die Edge Agent-Software kontinuierlich Datenerfassungsschemata und sammelt Daten entsprechend.

## Zugehörige Services

AWS IoT FleetWise lässt sich in die folgenden AWS Dienste integrieren, um die Verfügbarkeit und Skalierbarkeit Ihrer Cloud-Lösungen zu verbessern.

- **AWS IoT Core**— Registrieren und steuern Sie AWS IoT Geräte, die Fahrzeugdaten in das AWS IoT hochladen FleetWise. Weitere Informationen finden Sie unter [Was ist AWS IoT](#) im AWS IoT-Entwicklerhandbuch.
- **Amazon Timestream** — Verwenden Sie eine Zeitreihendatenbank, um Ihre Fahrzeugdaten zu speichern und zu analysieren. Weitere Informationen finden Sie unter [Was ist Amazon Timestream](#) im Amazon Timestream Developer Guide.
- **Amazon S3** — Verwenden Sie einen Objektspeicherservice, um Ihre Fahrzeugdaten zu speichern und zu verwalten. Weitere Informationen finden Sie unter [Was ist Amazon S3](#) im Amazon Simple Storage Service-Benutzerhandbuch.

# Einrichtung AWS IoT FleetWise

Bevor Sie es verwenden AWS Führen Sie FleetWise zum ersten Mal IoT die Schritte in den folgenden Abschnitten aus.

Themen

- [Richten Sie Ihre ein AWS-Konto](#)
- [Erste Schritte in der Konsole](#)
- [Einstellungen konfigurieren](#)

## Richten Sie Ihre ein AWS-Konto

Erledigen Sie die folgenden Aufgaben, um sich anzumelden AWS und erstellen Sie einen Administratorbenutzer.

### Melde dich an für ein AWS-Konto

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie [https://portal.aws.amazon.com/billing/die Anmeldung](https://portal.aws.amazon.com/billing/die-Anmeldung).
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, ein Root-Benutzer des AWS-Kontos wird erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen im Konto. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

## Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie sich Ihre Root-Benutzer des AWS-Kontos, aktivieren AWS IAM Identity Center, und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melde dich an bei [AWS Management Console](#) als Kontoinhaber wählen Sie Root-Benutzer und geben Sie Ihren AWS-Konto E-Mail-Adresse. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Als Root-Benutzer anmelden im AWS-Anmeldung Benutzerleitfaden](#).

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren Sie ein virtuelles MFA Gerät für AWS-Konto Root-Benutzer \(Konsole\)](#) im IAMBenutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) in der AWS IAM Identity Center Benutzerleitfaden.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Für ein Tutorial zur Verwendung des IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) in der AWS IAM Identity Center Benutzerleitfaden.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie unter [Anmelden bei AWS Zugriffsportal](#) im AWS-Anmeldung Benutzerleitfaden.

## Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie unter [Einen Berechtigungssatz erstellen in](#) der AWS IAM Identity Center Benutzerleitfaden.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden Sie unter [Gruppen hinzufügen](#) in der AWS IAM Identity Center Benutzerleitfaden.

### Note

Sie können eine dienstverknüpfte Rolle verwenden mit AWS IoT FleetWise. Servicebezogene Rollen sind vordefiniert von AWS IoT FleetWise und schließen die Berechtigungen ein, die AWS IoT FleetWise muss Metriken an Amazon senden CloudWatch. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für das IoT AWS FleetWise](#).

## Erste Schritte in der Konsole

Wenn Sie noch nicht bei Ihrem angemeldet sind AWS-Konto, melden Sie sich an und öffnen Sie dann [AWS FleetWise IoT-Konsole](#). Um loszulegen mit AWS IoT FleetWise, erstellen Sie ein Fahrzeugmodell. Ein Fahrzeugmodell standardisiert das Format Ihrer Fahrzeuge.

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
2. In Erste Schritte mit AWS IoT FleetWise, wählen Sie Erste Schritte aus.

Weitere Informationen zum Erstellen eines Fahrzeugmodells finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).

## Einstellungen konfigurieren

Sie können das AWS FleetWise IoT-Konsole oder API um Einstellungen für Amazon CloudWatch Logs-Metriken und Amazon CloudWatch Logs zu konfigurieren und Daten mit einem Von AWS verwalteter Schlüssel.

Mit CloudWatch Metriken können Sie Folgendes überwachen AWS IoT FleetWise und andere AWS Ressourcen schätzen. Sie können CloudWatch Metriken verwenden, um Messwerte zu sammeln und nachzuverfolgen, z. B. um festzustellen, ob ein Servicelimit überschritten wurde. Weitere Informationen zu CloudWatch Metriken finden Sie unter [Überwachung des AWS IoT FleetWise mit Amazon CloudWatch](#).

Mit CloudWatch Protokollen AWS IoT FleetWise sendet Protokolldaten an eine CloudWatch Protokollgruppe, wo Sie sie verwenden können, um Probleme zu identifizieren und zu beheben. Weitere Informationen zu CloudWatch Protokollen finden Sie unter [AWS FleetWise IoT-Protokollierung konfigurieren](#).

Mit Datenverschlüsselung AWS FleetWise IoT-Anwendungen Von AWS verwaltete Schlüssel um Daten zu verschlüsseln. Sie können sich auch dafür entscheiden, Schlüssel mit zu erstellen und zu verwalten AWS KMS. Weitere Hinweise zur Verschlüsselung finden Sie unter [Datenverschlüsselung](#).

## Konfigurieren der Einstellungen (Konsole)

Wenn Sie noch nicht bei Ihrem angemeldet sind AWS-Konto, melden Sie sich an und öffnen Sie dann [AWS FleetWiseIoT-Konsole](#).

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im linken Bereich Einstellungen aus.
3. Wählen Sie unter Metriken die Option Aktivieren aus. AWS IoT fügt der serviceverknüpften Rolle FleetWise automatisch eine CloudWatch verwaltete Richtlinie hinzu und aktiviert CloudWatch Metriken.
4. Wählen Sie unter Protokollierung die Option Bearbeiten aus.
  - a. Geben Sie im Abschnitt CloudWatch Protokollierung die Protokollgruppe ein.
  - b. Um Ihre Änderungen zu speichern, wählen Sie Submit.
5. Wählen Sie im Bereich Verschlüsselung die Option Bearbeiten aus.

- a. Wählen Sie den Schlüsseltyp aus, den Sie verwenden möchten. Weitere Informationen finden Sie unter [Schlüsselverwaltung](#).
  - i. Benutze AWS Schlüssel — AWS IoT FleetWise besitzt und verwaltet den Schlüssel.
  - ii. Wählen Sie ein anderes AWS Key Management Service Schlüssel — Du schaffst AWS KMS keys die sind in deinem Konto.
- b. Um Ihre Änderungen zu speichern, wählen Sie Absenden.

## Einstellungen konfigurieren (AWS CLI)

Im AWS CLI, registrieren Sie das Konto, um Einstellungen zu konfigurieren.

1. Führen Sie den folgenden Befehl aus, um Einstellungen zu konfigurieren.

```
aws iotfleetwise register-account
```

2. Um Ihre Einstellungen zu überprüfen, führen Sie den folgenden Befehl aus, um den Registrierungsstatus abzurufen.

### Note

Die mit dem Dienst verknüpfte Rolle wird nur zum Veröffentlichen verwendet AWS FleetWise IoT-Metriken zu CloudWatch. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für das IoT AWS FleetWise](#).

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

### Example response

```
{
  "accountStatus": "REGISTRATION_SUCCESS",
  "creationTime": "2022-07-28T11:31:22.603000-07:00",
  "customerAccountId": "012345678912",
  "iamRegistrationResponse": {
    "errorMessage": "",
    "registrationStatus": "REGISTRATION_SUCCESS",
    "roleArn": "arn:aws:iam::012345678912:role/AWSIoT FleetwiseServiceRole"
  }
}
```

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

Der Registrierungsstatus kann einer der folgenden sein:

- **REGISTRATION\_SUCCESS**— Der AWS Die Ressource wurde erfolgreich registriert.
- **REGISTRATION\_PENDING** – AWS IoT bearbeitet FleetWise die Registrierungsanfrage. Dieser Vorgang dauert ungefähr fünf Minuten.
- **REGISTRATION\_FAILURE** – AWS IoT FleetWise kann das nicht registrieren AWS Ressource. Bitte versuchen Sie es später erneut.

# Erste Schritte mit AWS IoT FleetWise

Mit AWS IoT FleetWise können Sie Ihre Fahrzeugdaten sammeln, transformieren und übertragen. Verwenden Sie die Tutorials in diesem Abschnitt, um mit AWS IoT zu beginnen FleetWise.

In den folgenden Themen erfahren Sie mehr über AWS IoT FleetWise:

- [Daten in die Cloud aufnehmen](#)
- [Fahrzeuge modellieren](#)
- [Fahrzeuge erstellen, bereitstellen und verwalten](#)
- [Flotten erstellen und verwalten](#)
- [Sammeln und übertragen Sie Daten mit Kampagnen](#)

## Voraussetzungen

Sie müssen über eine verfügen AWS-Konto , um mit AWS IoT beginnen zu können FleetWise. Falls Sie noch keines haben, beachten Sie die Informationen unter [Einrichtung AWS IoT FleetWise](#).

Verwenden Sie eine Region, in der AWS IoT verfügbar FleetWise ist. Weitere Informationen finden Sie unter [AWS FleetWise IoT-Endpunkte und Kontingente](#). Sie können die Regionsauswahl in verwenden AWS Management Console , um zu einer dieser Regionen zu wechseln.

## Demo der Edge Agent-Software

Sie können die Explore Edge Agent-Schnellstart-Demo verwenden, um sich mit AWS IoT FleetWise vertraut zu machen und zu erfahren, wie Sie Edge Agent-Software für AWS IoT entwickeln FleetWise. Diese Demo verwendet eine AWS CloudFormation Vorlage. Es führt Sie durch die Überprüfung der Edge Agent-Referenzimplementierung, die Entwicklung Ihres Edge-Agents und die anschließende Bereitstellung Ihrer Edge-Agent-Software auf einem Amazon EC2 Graviton sowie die Generierung von Beispielfahrzeugdaten. Die Demo bietet auch ein Skript, mit dem Sie einen Signalkatalog, ein Fahrzeugmodell, ein Decoder-Manifest, ein Fahrzeug, eine Flotte und eine Kampagne erstellen können — alles in der Cloud. Für weitere Informationen zur Schnellstart-Demo gehen Sie wie folgt vor, um das Edge Agent Software Developer Guide herunterzuladen.

Um die Schnellstart-Demo herunterzuladen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).

2. Wählen Sie auf der Service-Startseite im FleetWise Abschnitt Erste Schritte mit AWS IoT die Option Explore Edge Agent aus.

Internet of Things

# AWS IoT FleetWise

## Unlock the value of vehicle data

With AWS IoT FleetWise, you can model vehicles, transform binary data into human-readable values, and control what vehicle data to collect and when to transfer selected data to the cloud.

### Get started with AWS IoT FleetWise

Create a vehicle model to get started with AWS IoT FleetWise. Models standardize the format of your vehicles.

[View models](#)

[Explore Edge Agent](#)

## Tutorial: Erste Schritte mit AWS IoT FleetWise (Konsole)

Nutzen Sie AWS IoT, FleetWise um das einzigartige Datenformat nahezu in Echtzeit von automatisierten Fahrzeugen zu sammeln, zu transformieren und in die Cloud zu übertragen. Sie haben Zugang zu flottenweiten Erkenntnissen. Dies kann Ihnen helfen, Probleme mit dem Fahrzeugzustand effizient zu erkennen und zu beheben, hochwertige Datensignale zu übertragen und Probleme aus der Ferne zu diagnostizieren und gleichzeitig die Kosten zu senken.

Dieses Tutorial zeigt Ihnen, wie Sie mit AWS IoT beginnen können FleetWise. Sie lernen, wie Sie ein Fahrzeugmodell (Modellmanifest), ein Decoder-Manifest, ein Fahrzeug und eine Kampagne erstellen.

Weitere Informationen zu den wichtigsten Komponenten und Konzepten von AWS IoT FleetWise finden Sie unter [So FleetWise funktioniert AWS IoT](#).

Geschätzte Zeit: Ungefähr 45 Minuten.

### Important

Die AWS FleetWise IoT-Ressourcen, die diese Demo erstellt und verbraucht, werden Ihnen in Rechnung gestellt. Weitere Informationen finden Sie unter [AWS IoT](#) auf FleetWise der Seite mit den AWS FleetWise IoT-Preisen.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Richten Sie die Edge Agent-Software für AWS IoT ein FleetWise](#)
- [Schritt 2: Erstellen Sie ein Fahrzeugmodell](#)
- [Schritt 3: Erstellen Sie ein Decoder-Manifest](#)
- [Schritt 4: Konfigurieren Sie ein Decoder-Manifest](#)
- [Schritt 5: Erstellen Sie ein Fahrzeug](#)
- [Schritt 6: Erstelle eine Kampagne](#)
- [Schritt 7: Bereinigen](#)
- [Nächste Schritte](#)

## Voraussetzungen

Um dieses Tutorial „Erste Schritte“ abzuschließen, benötigen Sie zunächst Folgendes:

- Ein AWS-Konto. Falls Sie noch keinen haben AWS-Konto, finden Sie weitere Informationen unter [Erstellen eines AWS-Konto](#) im AWS Account Management Referenzhandbuch.
- Zugriff auf einen AWS-Region , der AWS IoT unterstützt FleetWise. Derzeit FleetWise wird AWS IoT in den USA Ost (Nord-Virginia) und Europa (Frankfurt) unterstützt.
- Amazon Timestream Timestream-Ressourcen:
  - Eine Amazon Timestream Timestream-Datenbank. Weitere Informationen finden Sie unter [Create a database](#) im Amazon Timestream Developer Guide.
  - Eine in Amazon Timestream erstellte Amazon Timestream Timestream-Tabelle, die Ihre Daten enthält. Weitere Informationen finden Sie unter [Tabelle erstellen](#) im Amazon Timestream Developer Guide.

## Schritt 1: Richten Sie die Edge Agent-Software für AWS IoT ein FleetWise

### Note

Der CloudFormation Stack in diesem Schritt verwendet Telemetriedaten. Sie können einen CloudFormation Stapel auch mithilfe von Bildverarbeitungssystemdaten erstellen. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

Ihre Edge Agent-Software für AWS IoT FleetWise erleichtert die Kommunikation zwischen Fahrzeugen und der Cloud. Sie erhält von Datenerfassungsprogrammen Anweisungen zur Erfassung von Daten aus mit der Cloud verbundenen Fahrzeugen.

Gehen Sie unter Allgemeine Informationen wie folgt vor, um Ihre Edge Agent-Software einzurichten:

1. Öffnen Sie die [CloudFormation Startvorlage](#).
2. Geben Sie auf der Seite Quick create stack unter Stackname den Namen Ihres Stacks von AWS FleetWise IoT-Ressourcen ein. Ein Stack ist ein benutzerfreundlicher Name, der als Präfix in den Namen der Ressourcen erscheint, die diese AWS CloudFormation Vorlage erstellt.
3. Geben Sie unter Parameter Ihre benutzerdefinierten Werte für die Parameter ein, die sich auf Ihren Stack beziehen.
  - a. Flottengröße — Sie können die Anzahl der Fahrzeuge in Ihrer Flotte erhöhen, indem Sie den Parameter Fleetsize aktualisieren.
  - b. IoT CoreRegion - Sie können die Region angeben, in der das AWS IoT Ding erstellt wird, indem Sie den CoreRegion IoT-Parameter aktualisieren. Sie müssen dieselbe Region verwenden, in der Sie Ihre AWS FleetWise IoT-Fahrzeuge erstellt haben. Weitere Informationen finden Sie AWS-Regionen unter [Regionen und Zonen — Amazon Elastic Compute Cloud](#).
4. Wählen Sie im Abschnitt Funktionen das Kästchen aus, um zu bestätigen, dass AWS CloudFormation dadurch IAM-Ressourcen erstellt werden.
5. Wählen Sie Stack erstellen und warten Sie dann etwa 15 Minuten, bis der Status des Stacks CREATE\_COMPLETE anzeigt.
6. Um zu bestätigen, dass der Stack erstellt wurde, wählen Sie den Tab Stack-Info, aktualisieren Sie die Ansicht und suchen Sie nach CREATE\_COMPLETE.

fwdemo ⚙️ | ✕

---

### Overview ↻

Stack ID	Description
<a href="#">arn:aws:cloudformation:us-east-1:012345678912:stack/fwdemo/bd04af20-a269-11ed-bf1d-0a56266679b7</a>	-
Status	Status reason
<span style="color: green;">✔️</span> CREATE_COMPLETE	-

#### ⚠️ Important

Die AWS FleetWise IoT-Ressourcen, die diese Demo erstellt und verbraucht, werden Ihnen in Rechnung gestellt. Weitere Informationen finden Sie unter [AWS IoT](#) auf FleetWise der Seite mit den AWS FleetWise IoT-Preisen.

## Schritt 2: Erstellen Sie ein Fahrzeugmodell

#### ⚠️ Important

Sie können kein Fahrzeugmodell mit Datensignalen des Bildverarbeitungssystems in der AWS FleetWise IoT-Konsole erstellen. Verwenden Sie stattdessen die AWS CLI.

Sie verwenden Fahrzeugmodelle, um das Format Ihrer Fahrzeuge zu standardisieren und die Beziehung zwischen den Signalen in den von Ihnen erstellten Fahrzeugen zu definieren. Ein Signalkatalog wird auch erstellt, wenn Sie ein Fahrzeugmodell erstellen. Ein Signalkatalog ist eine Sammlung standardisierter Signale, die zur Erstellung von Fahrzeugmodellen wiederverwendet werden können. Signale sind grundlegende Strukturen, die Sie so definieren, dass sie Fahrzeugdaten und ihre Metadaten enthalten. Derzeit unterstützt der AWS FleetWise IoT-Dienst nur einen Signalkatalog AWS-Region pro Konto. Auf diese Weise kann überprüft werden, ob die verarbeiteten Daten einer Fahrzeugflotte konsistent sind.

Um ein Fahrzeugmodell zu erstellen

1. Öffnen Sie die AWS FleetWise IoT-Konsole.

2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie auf der Seite Fahrzeugmodelle die Option Fahrzeugmodell erstellen aus.
4. Geben Sie im Abschnitt Allgemeine Informationen den Namen Ihres Fahrzeugmodells, z. B. Vehicle1, und eine optionale Beschreibung ein. Wählen Sie anschließend Weiter.
5. Wählen Sie ein oder mehrere Signale aus dem Signalkatalog aus. Sie können Signale im Suchkatalog nach Namen filtern oder sie aus der Liste auswählen. Sie können beispielsweise Signale für Reifendruck und Bremsdruck auswählen, um Daten zu diesen Signalen zu sammeln. Wählen Sie Weiter aus.
6. Wählen Sie Ihre DBC-Dateien aus und laden Sie sie von Ihrem lokalen Gerät hoch. Wählen Sie Weiter aus.

 Note

Für dieses Tutorial können Sie eine [.dbc-Beispieldatei](#) herunterladen, um sie für diesen Schritt hochzuladen.

7. Fügen Sie Ihrem Fahrzeugmodell Attribute hinzu und wählen Sie dann Weiter.
  - a. Name - Geben Sie den Namen des Fahrzeugattributs ein, z. B. den Herstellernamen oder das Herstellungsdatum.
  - b. Datentyp - Wählen Sie im Menü Datentyp einen Datentyp aus.
  - c. Einheit - (optional) Geben Sie einen Einheitswert ein, z. B. Kilometer oder Celsius.
  - d. Pfad — (Optional) Geben Sie einen Namen für den Pfad zu einem Signal ein, z. B. Vehicle.Engine.Light. Der Punkt (.) gibt an, dass es sich um ein untergeordnetes Signal handelt.
  - e. Standardwert - (Optional) Geben Sie einen Standardwert ein.
  - f. Beschreibung - (Optional) Geben Sie eine Beschreibung des Attributs ein.
8. Überprüfen Sie Ihre Konfigurationen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen). Es wird eine Benachrichtigung angezeigt, dass Ihr Fahrzeugmodell erfolgreich erstellt wurde.

✔ **Vehicle model created**  
You successfully created the vehicle model: demo.

AWS IoT FleetWise > Vehicle models > Demo

## demo

[Duplicate](#) [Create vehicle](#) [Create decoder manifest](#)

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

### Summary [Info](#)

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

## Schritt 3: Erstellen Sie ein Decoder-Manifest

Decoder-Manifeste sind den Fahrzeugmodellen zugeordnet, die Sie erstellen. Sie enthalten Informationen, die dem AWS IoT helfen, Fahrzeugdaten aus einem Binärformat zu FleetWise dekodieren und in menschenlesbare Werte umzuwandeln, die analysiert werden können. Netzwerkschnittstellen und Decodersignale sind Komponenten, die bei der Konfiguration von Decoder-Manifesten helfen. Eine Netzwerkschnittstelle enthält Informationen über das CAN- oder OBD-Protokoll, das Ihr Fahrzeugnetzwerk verwendet. Das Decodersignal liefert Dekodierungsinformationen für ein bestimmtes Signal.

Um ein Decoder-Manifest zu erstellen

1. Öffnen Sie die AWS FleetWise IoT-Konsole.
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie im Abschnitt Fahrzeugmodelle das Fahrzeugmodell aus, das Sie zum Erstellen eines Decoder-Manifests verwenden möchten.
4. Wählen Sie Decoder-Manifest erstellen.

## Schritt 4: Konfigurieren Sie ein Decoder-Manifest

Um ein Decoder-Manifest zu konfigurieren

### Important

Sie können mit der AWS FleetWise IoT-Konsole keine Datensignale von Bildverarbeitungssystemen in Decoder-Manifesten konfigurieren. Verwenden Sie stattdessen die AWS CLI. Weitere Informationen finden Sie unter [Erstellen Sie ein Decoder-Manifest \(AWS CLI\)](#).

1. Geben Sie einen Namen und eine optionale Beschreibung ein, damit Sie Ihr Decoder-Manifest leichter identifizieren können. Wählen Sie anschließend Weiter.
2. Um eine oder mehrere Netzwerkschnittstellen hinzuzufügen, wählen Sie entweder den Typ CAN\_INTERFACE oder den Typ OBD\_INTERFACE.
  - On-Board-Diagnoseschnittstelle (OBD) — Wählen Sie diesen Schnittstellentyp, wenn Sie ein Protokoll benötigen, das definiert, wie Selbstdiagnosedaten zwischen elektronischen Steuergeräten (ECUs) übertragen werden. Dieses Protokoll bietet eine Reihe von Standard-Diagnosefehlercodes (DTCs), die Ihnen bei der Behebung von Problemen mit Ihrem Fahrzeug helfen können.
  - CAN-Bus-Schnittstelle (Controller Area Network) - Wählen Sie diesen Schnittstellentyp, wenn Sie ein Protokoll benötigen, das definiert, wie Daten zwischen Steuergeräten übertragen werden. Bei Steuergeräten kann es sich um Motorsteuergeräte, Airbags oder das Audiosystem handeln.
3. Geben Sie einen Namen für die Netzwerkschnittstelle ein.
4. Um der Netzwerkschnittstelle Signale hinzuzufügen, wählen Sie ein oder mehrere Signale aus der Liste aus.
5. Wählen Sie ein Decodersignal für das Signal, das Sie im vorherigen Schritt hinzugefügt haben. Laden Sie eine DBC-Datei hoch, um Dekodierungsinformationen bereitzustellen. Jedes Signal im Fahrzeugmodell muss mit einem Decodersignal gekoppelt werden, das Sie aus der Liste auswählen können.
6. Um eine weitere Netzwerkschnittstelle hinzuzufügen, wählen Sie Netzwerkschnittstelle hinzufügen. Wenn Sie mit dem Hinzufügen von Netzwerkschnittstellen fertig sind, wählen Sie Weiter.

- Überprüfen Sie Ihre Konfigurationen und wählen Sie dann Erstellen. Es wird eine Benachrichtigung angezeigt, dass Ihr Decoder-Manifest erfolgreich erstellt wurde.

## Schritt 5: Erstellen Sie ein Fahrzeug

Im AWS IoT FleetWise der Dinge sind Fahrzeuge virtuelle Repräsentationen Ihres realen, physischen Fahrzeugs. Alle Fahrzeuge, die mit demselben Fahrzeugmodell erstellt wurden, erben dieselbe Gruppe von Signalen, und jedes Fahrzeug, das Sie erstellen, entspricht einem neu erstellten IoT-Ding. Sie müssen alle Fahrzeuge einem Decoder-Manifest zuordnen.

### Voraussetzungen

- Stellen Sie sicher, dass Sie das Fahrzeugmodell und das Decoder-Manifest bereits erstellt haben. Stellen Sie außerdem sicher, dass der Status des Fahrzeugmodells AKTIV ist.
  - Um zu überprüfen, ob der Status des Fahrzeugmodells AKTIV ist, öffnen Sie die AWS FleetWise IoT-Konsole.
  - Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
  - Überprüfen Sie im Abschnitt Zusammenfassung unter Status den Status Ihres Fahrzeugs.

✔ Vehicle model created
✕

You successfully created the vehicle model: demo.

AWS IoT FleetWise > Vehicle models > Demo

## demo

Duplicate
Create vehicle
Create decoder manifest

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

**Summary** [Info](#)

<p>Vehicle model ARN</p> <p> <code>arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo</code></p>	<p>Status</p> <p><span style="color: green; font-weight: bold;">✔ ACTIVE</span></p>	<p>Date created</p> <p>February 01, 2023 at 14:40 (UTC-05)</p>
<p>Signal catalog ARN</p> <p> <code>arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog</code></p>	<p>Description</p> <p>-</p>	<p>Last modified</p> <p>February 01, 2023 at 14:40 (UTC-05)</p>

## Um ein Fahrzeug zu erstellen

1. Öffnen Sie die FleetWise AWS-Konsole.
2. Wählen Sie im Navigationsbereich Fahrzeuge aus.
3. Wählen Sie Fahrzeug erstellen aus.
4. Um die Fahrzeugeigenschaften zu definieren, geben Sie den Fahrzeugnamen ein und wählen Sie dann ein Modellmanifest (Fahrzeugmodell) und ein Decoder-Manifest aus.
5. (Optional) Um die Fahrzeugattribute zu definieren, geben Sie ein Schlüssel-Wert-Paar ein und wählen Sie dann Attribute hinzufügen.
6. (Optional) Um Ihre AWS-Ressource zu kennzeichnen, fügen Sie Tags hinzu und wählen Sie dann Neues Tag hinzufügen.
7. Wählen Sie Weiter aus.
8. Um das Fahrzeugzertifikat zu konfigurieren, können Sie entweder Ihr eigenes Zertifikat hochladen oder die Option Neues Zertifikat automatisch generieren wählen. Wir empfehlen, Ihr Zertifikat automatisch zu generieren, um die Einrichtung zu beschleunigen. Wenn Sie bereits über ein Zertifikat verfügen, können Sie es stattdessen verwenden.
9. Laden Sie die öffentlichen und privaten Schlüsseldateien herunter und wählen Sie dann Weiter.
10. Um dem Fahrzeugzertifikat eine Richtlinie anzuhängen, können Sie entweder einen vorhandenen Richtliniennamen eingeben oder eine neue Richtlinie erstellen. Um eine neue Richtlinie zu erstellen, wählen Sie Richtlinie erstellen und dann Weiter.
11. Überprüfen Sie Ihre Konfigurationen. Wenn Sie fertig sind, wählen Sie Fahrzeug erstellen.

## Schritt 6: Erstelle eine Kampagne

Im AWS IoT werden Kampagnen verwendet FleetWise, um die Auswahl, Erfassung und Übertragung von Daten von Fahrzeugen in die Cloud zu erleichtern. Kampagnen enthalten Datenerfassungsschemata, die der Edge Agent-Software Anweisungen zur Erfassung von Daten mit einem zustands- oder zeitbasierten Erfassungsschema geben.

### So erstellen Sie eine Kampagne

1. Öffnen Sie die AWS FleetWise IoT-Konsole.
2. Wählen Sie im Navigationsbereich Kampagnen aus.
3. Wählen Sie Create campaign (Kampagne erstellen).

4. Geben Sie Ihren Kampagnennamen und eine optionale Beschreibung ein.
5. Um das Datenerfassungsschema Ihrer Kampagne zu konfigurieren, können Sie das Datenerfassungsschema manuell definieren oder eine JSON-Datei von Ihrem lokalen Gerät hochladen. Durch das Hochladen einer.json-Datei wird das Datenerfassungsschema automatisch definiert.
  - a. Um das Datenerfassungsschema manuell zu definieren, wählen Sie Datenerfassungsschema definieren und wählen Sie den Typ des Datenerfassungsschemas aus, das Sie für Ihre Kampagne verwenden möchten. Sie können entweder ein auf Bedingungen basierendes Erfassungsschema oder ein zeitbasiertes Erfassungsschema wählen.
  - b. Wenn Sie sich für ein zeitbasiertes Erfassungsschema entscheiden, müssen Sie den Zeitraum angeben, für den Ihre Kampagne die Fahrzeugdaten erfasst.
  - c. Wenn Sie sich für ein zustandsabhängiges Erfassungsschema entscheiden, müssen Sie einen Ausdruck angeben, um zu erkennen, welche Daten erfasst werden sollen. Achten Sie darauf, den Namen des Signals als Variable, als Vergleichsoperator und als Vergleichswert anzugeben.
  - d. (Optional) Wählen Sie die Sprachversion Ihres Ausdrucks oder behalten Sie den Standardwert 1 bei.
  - e. (Optional) Geben Sie das Triggerintervall zwischen zwei Datenerfassungsereignissen an.
  - f. Um Daten zu sammeln, wählen Sie die Bedingung Triggermodus für die Edge Agent-Software. Standardmäßig sammelt die Edge Agent for AWS FleetWise IoT-Software immer dann Daten, wenn die Bedingung erfüllt ist. Oder er kann nur Daten sammeln, wenn die Bedingung zum ersten Mal erfüllt ist (beim ersten Trigger).
  - g. (Optional) Sie können erweiterte Schemaoptionen wählen.
6. Um die Signale anzugeben, aus denen das Datenerfassungsschema Daten sammelt, suchen Sie im Menü nach dem Namen des Signals.
7. (Optional) Sie können eine maximale Probenanzahl oder ein minimales Probenintervall wählen. Sie können auch weitere Signale hinzufügen.
8. Wählen Sie Weiter aus.
9. Definieren Sie das Speicherziel, an das die Kampagne Daten übertragen soll. Sie können Daten in Amazon S3 oder Amazon Timestream speichern.
  - a. Amazon S3 — Wählen Sie den S3-Bucket aus, der AWS IoT FleetWise über Berechtigungen verfügt.

- b. Amazon Timestream — wählen Sie die Timestream-Datenbank und den Tabellennamen aus. Geben Sie eine IAM-Rolle ein, die das Senden von Daten AWS IoT FleetWise an Timestream ermöglicht.
10. Wählen Sie Weiter aus.
11. Wählen Sie Fahrzeugattribute oder Fahrzeugnamen aus dem Suchfeld aus.
12. Geben Sie den Wert ein, der sich auf das Attribut oder den Namen bezieht, den Sie für Ihr Fahrzeug ausgewählt haben.
13. Wählen Sie die Fahrzeuge aus, von denen Ihre Kampagne Daten sammeln soll. Wählen Sie anschließend Weiter.
14. Überprüfen Sie die Konfigurationen Ihrer Kampagne und wählen Sie dann Kampagne erstellen. Sie oder Ihr Team müssen die Kampagne für Fahrzeuge bereitstellen.

## Schritt 7: Bereinigen

Um weitere Gebühren für die Ressourcen zu vermeiden, die Sie in diesem Tutorial verwendet haben, löschen Sie den AWS CloudFormation Stapel und alle Stack-Ressourcen.

Um den AWS CloudFormation Stapel zu löschen

1. Öffnen Sie die [AWS CloudFormation -Konsole](#).
2. Wählen Sie aus der Liste der Stacks den Stapel aus, den Sie in Schritt 1 erstellt haben.
3. Wählen Sie Löschen aus.
4. Um die Löschung zu bestätigen, klicken Sie auf Delete (Löschen). Das Löschen des Stacks dauert etwa 15 Minuten.

## Nächste Schritte

1. Sie können die Fahrzeugdaten, die Ihre Kampagne sammelt, verarbeiten und visualisieren. Weitere Informationen finden Sie unter [Verarbeitung und Visualisierung von Fahrzeugdaten](#).
2. Sie können Probleme mit AWS IoT beheben und lösen FleetWise. Weitere Informationen finden Sie unter [AWS IoT-Problemlösung FleetWise](#).

# Daten in die Cloud aufnehmen

Die Edge Agent for AWS FleetWise IoT-Software ist darauf ausgelegt, eine sichere Kommunikation zwischen Ihren Fahrzeugen und der Cloud zu ermöglichen, wenn sie in Fahrzeugen installiert ist und läuft.

## Note

- AWS IoT FleetWise ist nicht für den Einsatz in oder in Verbindung mit dem Betrieb gefährlicher Umgebungen oder kritischer Systeme vorgesehen, die zu schweren Körperverletzungen oder zum Tod führen oder Umwelt- oder Sachschäden verursachen können. Fahrzeugdaten, die durch Ihre Nutzung von AWS IoT FleetWise gesammelt werden, dienen nur zu Informationszwecken, und Sie dürfen AWS IoT FleetWise zur Steuerung oder Bedienung von Fahrzeugfunktionen verwenden.
- Fahrzeugdaten, die durch Ihre Nutzung des AWS IoT erfasst werden, FleetWise sollten auf ihre Genauigkeit geprüft werden, die für Ihren Anwendungsfall angemessen ist, auch zum Zwecke der Erfüllung von Compliance-Verpflichtungen, die Sie möglicherweise gemäß den geltenden Fahrzeugsicherheitsvorschriften haben (wie Sicherheitsüberwachungs- und Berichtspflichten). Eine solche Bewertung sollte die Erfassung und Überprüfung von Informationen mit anderen branchenüblichen Mitteln und Quellen (z. B. Berichte von Fahrzeugführern) umfassen.

Gehen Sie wie folgt vor, um Daten in die Cloud aufzunehmen:

1. Entwickeln und installieren Sie Ihre Edge Agent for AWS FleetWise IoT-Software in Ihrem Fahrzeug. Für weitere Informationen zur Arbeit mit der Edge Agent-Software gehen Sie wie folgt vor, um das [Edge Agent for AWS FleetWise IoT-Softwareentwicklerhandbuch](#) herunterzuladen.
  1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
  2. Wählen Sie auf der Service-Startseite im FleetWise Abschnitt Erste Schritte mit AWS IoT die Option Explore Edge Agent aus.
2. Erstellen oder importieren Sie einen Signalkatalog mit Signalen, die Sie zur Erstellung eines Fahrzeugmodells verwenden werden. Weitere Informationen finden Sie unter [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#) und [Importieren Sie einen Signalkatalog \(AWS CLI\)](#).

**Note**

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um das erste Fahrzeugmodell zu erstellen, müssen Sie keinen Signalkatalog manuell erstellen. Wenn Sie Ihr erstes Fahrzeugmodell erstellen, erstellt AWS IoT FleetWise automatisch einen Signalkatalog für Sie. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- AWS IoT unterstützt FleetWise derzeit einen Signalkatalog für jedes AWS Konto pro AWS-Region.

3. Verwenden Sie Signale im Signalkatalog, um ein Fahrzeugmodell zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell](#).

**Note**

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, können Sie .dbc-Dateien hochladen, um Signale zu importieren. .dbc ist ein Dateiformat, das Controller Area Network (CAN-Bus) -Datenbanken unterstützen. Nachdem das Fahrzeugmodell erstellt wurde, werden dem Signalkatalog automatisch neue Signale hinzugefügt. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- Wenn Sie die CreateModelManifest API-Operation verwenden, um ein Fahrzeugmodell zu erstellen, müssen Sie die UpdateModelManifest API-Operation verwenden, um das Fahrzeugmodell zu aktivieren. Weitere Informationen finden Sie unter [Aktualisieren Sie ein Fahrzeugmodell \(AWS CLI\)](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, aktiviert AWS IoT das Fahrzeugmodell FleetWise automatisch für Sie.

4. Erstellen Sie ein Decoder-Manifest. Das Decoder-Manifest enthält Dekodierungsinformationen für jedes Signal, das im Fahrzeugmodell angegeben ist, das Sie im vorherigen Schritt erstellt haben. Das Decoder-Manifest ist dem Fahrzeugmodell zugeordnet, das Sie erstellt haben. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

**Note**

- Wenn Sie den `CreateDecoderManifest` API-Vorgang verwenden, um ein Decoder-Manifest zu erstellen, müssen Sie den `UpdateDecoderManifest` API-Vorgang verwenden, um das Decoder-Manifest zu aktivieren. Weitere Informationen finden Sie unter [Aktualisieren Sie ein Decoder-Manifest \(AWS CLI\)](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Decoder-Manifest zu erstellen, aktiviert AWS IoT das Decoder-Manifest FleetWise automatisch für Sie.

5. Erstellen Sie Fahrzeuge anhand des Fahrzeugmodells. Fahrzeuge, die mit demselben Fahrzeugmodell erstellt wurden, erben dieselbe Gruppe von Signalen. Sie müssen AWS IoT Core für die Bereitstellung Ihres Fahrzeugs verwenden, bevor Sie Daten in die Cloud aufnehmen können. Weitere Informationen finden Sie unter [Fahrzeuge erstellen, bereitstellen und verwalten](#).
6. (Optional) Erstellen Sie eine Flotte, die eine Gruppe von Fahrzeugen repräsentiert, und ordnen Sie dann einzelne Fahrzeuge der Flotte zu. Auf diese Weise können Sie mehrere Fahrzeuge gleichzeitig verwalten. Weitere Informationen finden Sie unter [Flotten erstellen und verwalten](#).
7. Kampagnen erstellen. Kampagnen werden für ein Fahrzeug oder eine Fahrzeugflotte eingesetzt. Kampagnen geben der Edge Agent-Software Anweisungen zur Auswahl, Erfassung und Übertragung von Daten in die Cloud. Weitere Informationen finden Sie unter [Sammeln und übertragen Sie Daten mit Kampagnen](#).

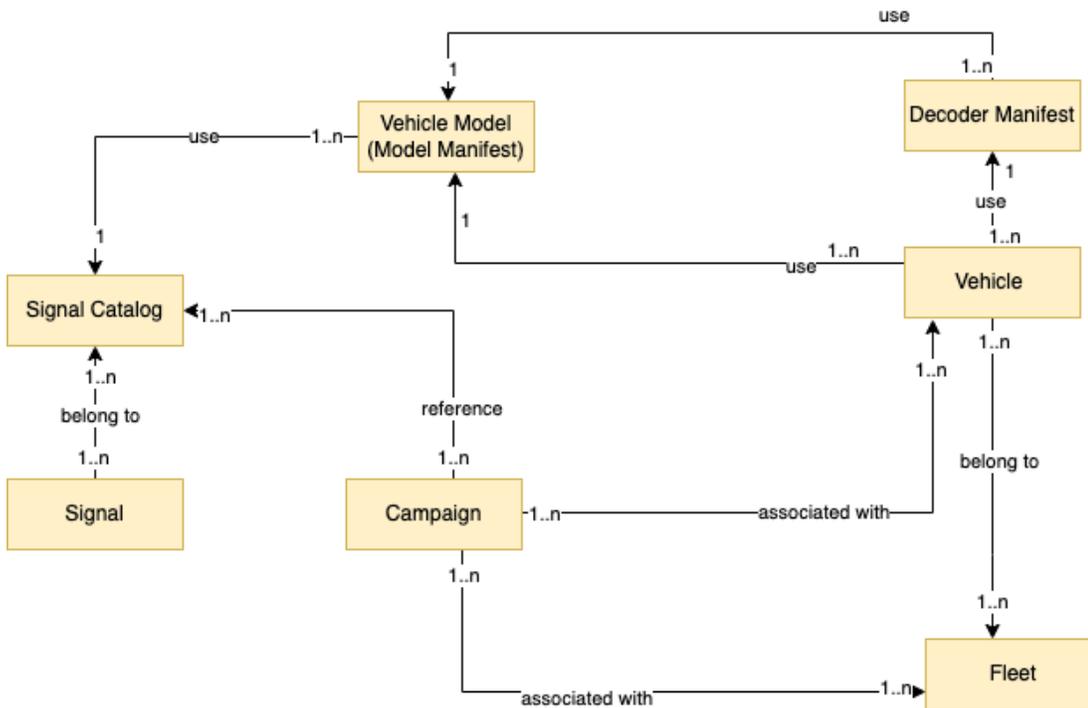
**Note**

Sie müssen den `UpdateCampaign` API-Vorgang verwenden, um die Kampagne zu genehmigen, bevor AWS IoT sie für das Fahrzeug oder die Flotte bereitstellen FleetWise kann. Weitere Informationen finden Sie unter [Aktualisieren Sie eine Kampagne \(AWS CLI\)](#).

Die Edge Agent-Software überträgt Fahrzeugdaten an AWS IoT Core das reservierte Thema `aws/iotfleetwise/vehicles/vehicleName/signals`, das Daten an das AWS IoT sendet FleetWise. AWS IoT liefert die Daten FleetWise dann an eine Timestream-Tabelle oder einen Amazon S3 S3-Bucket. Sie können Timestream verwenden, um Ihre Daten abzufragen, und Amazon QuickSight oder Grafana verwenden, um Ihre Daten zu visualisieren. Weitere Informationen finden Sie unter [Verarbeitung und Visualisierung von Fahrzeugdaten](#).

# Fahrzeuge modellieren

AWS IoT FleetWise bietet ein Framework für die Fahrzeugmodellierung, mit dem Sie virtuelle Darstellungen Ihrer Fahrzeuge in der Cloud erstellen können. Signale, Signalkataloge, Fahrzeugmodelle und Decoder-Manifeste sind die Kernkomponenten, mit denen Sie bei der Modellierung Ihrer Fahrzeuge arbeiten.



## Signal

Signale sind grundlegende Strukturen, die Sie so definieren, dass sie Fahrzeugdaten und ihre Metadaten enthalten. Ein Signal kann ein Attribut, ein Zweig, ein Sensor oder ein Aktuator sein. Sie können beispielsweise einen Sensor erstellen, der die Temperaturwerte im Fahrzeug empfängt und dessen Metadaten, einschließlich eines Sensornamens, eines Datentyps und einer Einheit, speichert. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#).

## Signalkatalog

Ein Signalkatalog enthält eine Sammlung von Signalen. Signale in einem Signalkatalog können verwendet werden, um Fahrzeuge zu modellieren, die unterschiedliche Protokolle und Datenformate verwenden. Beispielsweise gibt es zwei Fahrzeuge, die von verschiedenen Autoherstellern hergestellt werden: eines verwendet das Control Area Network (CAN-Bus) - Protokoll, das andere das On-Board Diagnostics (OBD) -Protokoll. Sie können im Signalkatalog

einen Sensor für den Empfang von Fahrzeugtemperaturwerten definieren. Dieser Sensor kann zur Darstellung der Thermoelemente in beiden Fahrzeugen verwendet werden. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#).

## Fahrzeugmodell (Modellmanifest)

Fahrzeugmodelle sind deklarative Strukturen, mit denen Sie das Format Ihrer Fahrzeuge standardisieren und Beziehungen zwischen Signalen in den Fahrzeugen definieren können. Fahrzeugmodelle sorgen für konsistente Informationen für mehrere Fahrzeuge desselben Typs. Sie fügen Signale hinzu, um Fahrzeugmodelle zu erstellen. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).

## Decoder-Manifest

Decoder-Manifeste enthalten Dekodierungsinformationen für jedes Signal in Fahrzeugmodellen. Sensoren und Aktuatoren in Fahrzeugen übertragen Nachrichten auf niedriger Ebene (Binärdaten). Mit Decoder-Manifesten FleetWise ist AWS IoT in der Lage, Binärdaten in menschenlesbare Werte umzuwandeln. Jedes Decoder-Manifest ist einem Fahrzeugmodell zugeordnet. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Fahrzeuge auf folgende Weise zu modellieren.

1. Erstellen oder importieren Sie einen Signalkatalog mit Signalen, die Sie zur Erstellung eines Fahrzeugmodells verwenden werden. Weitere Informationen finden Sie unter [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#) und [Importieren Sie einen Signalkatalog \(AWS CLI\)](#).

### Note

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um das erste Fahrzeugmodell zu erstellen, müssen Sie keinen Signalkatalog manuell erstellen. Wenn Sie Ihr erstes Fahrzeugmodell erstellen, erstellt AWS IoT FleetWise automatisch einen Signalkatalog für Sie. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- AWS IoT unterstützt FleetWise derzeit einen Signalkatalog für jedes AWS Konto pro AWS-Region.

2. Verwenden Sie Signale im Signalkatalog, um ein Fahrzeugmodell zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell](#).

**Note**

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, können Sie .dbc-Dateien hochladen, um Signale zu importieren. .dbc ist ein Dateiformat, das Controller Area Network (CAN-Bus) -Datenbanken unterstützen. Nachdem das Fahrzeugmodell erstellt wurde, werden dem Signalkatalog automatisch neue Signale hinzugefügt. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- Wenn Sie die `CreateModelManifest` API-Operation verwenden, um ein Fahrzeugmodell zu erstellen, müssen Sie die `UpdateModelManifest` API-Operation verwenden, um das Fahrzeugmodell zu aktivieren. Weitere Informationen finden Sie unter [Aktualisieren Sie ein Fahrzeugmodell \(\)AWS CLI](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, aktiviert AWS IoT das Fahrzeugmodell FleetWise automatisch für Sie.

3. Erstellen Sie ein Decoder-Manifest. Das Decoder-Manifest enthält Dekodierungsinformationen für jedes Signal, das im Fahrzeugmodell angegeben ist, das Sie im vorherigen Schritt erstellt haben. Das Decoder-Manifest ist dem Fahrzeugmodell zugeordnet, das Sie erstellt haben. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

**Note**

- Wenn Sie den `CreateDecoderManifest` API-Vorgang verwenden, um ein Decoder-Manifest zu erstellen, müssen Sie den `UpdateDecoderManifest` API-Vorgang verwenden, um das Decoder-Manifest zu aktivieren. Weitere Informationen finden Sie unter [Aktualisieren Sie ein Decoder-Manifest \(\)AWS CLI](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Decoder-Manifest zu erstellen, aktiviert AWS IoT das Decoder-Manifest FleetWise automatisch für Sie.

CAN-Bus-Datenbanken unterstützen das .dbc-Dateiformat. Sie könnten .dbc-Dateien hochladen, um Signale und Decodersignale zu importieren. Gehen Sie wie folgt vor, um eine Beispiel-DBC-Datei zu erhalten.

Um eine .dbc-Datei abzurufen

1. [Laden Sie die ZIP-Datei herunter. EngineSignals](#)
2. Navigieren Sie zu dem Verzeichnis, in das Sie die Datei EngineSignals.zip heruntergeladen haben.
3. Entpacken Sie die Datei und speichern Sie sie lokal unter EngineSignals.dbc

Themen

- [Signalkataloge erstellen und verwalten](#)
- [Fahrzeugmodelle erstellen und verwalten](#)
- [Decoder-Manifeste erstellen und verwalten](#)

## Signalkataloge erstellen und verwalten

### Note

Sie können ein [Demo-Skript](#) herunterladen, um ROS 2-Nachrichten in VSS-JSON-Dateien zu konvertieren, die mit dem Signalkatalog kompatibel sind. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Ein Signalkatalog ist eine Sammlung standardisierter Signale, die zur Erstellung von Fahrzeugmodellen wiederverwendet werden können. AWS IoT FleetWise unterstützt die [Fahrzeugsignalspezifikation \(VSS\)](#), anhand derer Sie Signale definieren können. Bei einem Signal kann es sich um einen der folgenden Typen handeln.

Attribut

Attribute stellen statische Informationen dar, die sich im Allgemeinen nicht ändern, wie Hersteller und Herstellungsdatum.

Verzweigung

Zweige stellen Signale in einer verschachtelten Struktur dar. Zweige zeigen Signalhierarchien. Zum Beispiel hat der Vehicle Zweig einen untergeordneten Zweig, Powertrain. Der Powertrain Zweig hat einen untergeordneten Zweig, combustionEngine. Verwenden Sie den

`Vehicle.Powertrain.combustionEngine` Ausdruck, um den `combustionEngine` Zweig zu finden.

## Sensor

Sensordaten geben den aktuellen Zustand des Fahrzeugs an und ändern sich im Laufe der Zeit, wenn sich der Zustand des Fahrzeugs ändert, z. B. Flüssigkeitsstand, Temperaturen, Vibrationen oder Spannung.

## Aktuator

Aktuatordaten geben Auskunft über den Zustand von Fahrzeuggeräten wie Motoren, Heizungen und Türschlössern. Durch Ändern des Zustands eines Fahrzeuggeräts können Aktuatordaten aktualisiert werden. Sie können beispielsweise einen Aktuator definieren, der die Heizung darstellt. Der Aktuator empfängt neue Daten, wenn Sie die Heizung ein- oder ausschalten.

## Benutzerdefinierter Aufbau

Eine benutzerdefinierte Struktur (auch als Struktur bezeichnet) stellt eine komplexe Datenstruktur oder Datenstruktur höherer Ordnung dar. Sie erleichtert das logische Binden oder Gruppieren von Daten, die aus derselben Quelle stammen. Eine Struktur wird verwendet, wenn Daten in einer atomaren Operation gelesen oder geschrieben werden, z. B. um einen komplexen Datentyp oder eine Form höherer Ordnung darzustellen.

Ein Signal vom Strukturtyp wird im Signalkatalog definiert, indem ein Verweis auf einen Strukturdatentyp anstelle eines primitiven Datentyps verwendet wird. Strukturen können für alle Arten von Signalen verwendet werden, einschließlich Sensoren, Attributen, Aktuatoren und Datentypen für Bildverarbeitungssysteme. Wenn ein Signal vom Typ `Structure` gesendet oder empfangen wird, erwartet AWS IoT, dass alle enthaltenen Elemente gültige Werte haben, sodass alle Elemente obligatorisch sind. Wenn eine Struktur beispielsweise die Elemente `Vehicle.Camera.Image.Height`, `Vehicle.Camera.Image.Width` und `Vehicle.Camera.Image.Data` enthält, wird erwartet, dass das gesendete Signal Werte für all diese Elemente enthält.

### Note

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Benutzerdefiniertes Eigentum

Eine benutzerdefinierte Eigenschaft stellt ein Element der komplexen Datenstruktur dar. Der Datentyp der Eigenschaft kann entweder primitiv oder eine andere Struktur sein.

Bei der Darstellung einer Form höherer Ordnung mithilfe einer Struktur und einer benutzerdefinierten Eigenschaft wird die beabsichtigte Form höherer Ordnung immer als Baumstruktur definiert und betrachtet. Die benutzerdefinierte Eigenschaft wird verwendet, um alle Blattknoten zu definieren, während die Struktur verwendet wird, um alle Knoten zu definieren, die keine Blattknoten sind.

### Note

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um das erste Fahrzeugmodell zu erstellen, müssen Sie keinen Signalkatalog manuell erstellen. Wenn Sie Ihr erstes Fahrzeugmodell erstellen, erstellt AWS IoT FleetWise automatisch einen Signalkatalog für Sie. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, können Sie .dbc-Dateien hochladen, um Signale zu importieren. .dbc ist ein Dateiformat, das Controller Area Network (CAN-Bus) -Datenbanken unterstützen. Nachdem das Fahrzeugmodell erstellt wurde, werden dem Signalkatalog automatisch neue Signale hinzugefügt. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#).
- AWS IoT unterstützt FleetWise derzeit einen Signalkatalog für AWS-Konto jede Region.

AWS IoT FleetWise bietet die folgenden API-Operationen, mit denen Sie Signalkataloge erstellen und verwalten können.

- [CreateSignalCatalog](#)— Erstellt einen neuen Signalkatalog.
- [ImportSignalCatalog](#)— Importiert Signale, um einen Signalkatalog zu erstellen, indem eine JSON-Datei hochgeladen wird. Signale müssen gemäß VSS definiert und im JSON-Format gespeichert werden.
- [UpdateSignalCatalog](#)— Aktualisiert einen vorhandenen Signalkatalog durch Aktualisierung, Entfernung oder Hinzufügen von Signalen.
- [DeleteSignalCatalog](#)— Löscht einen vorhandenen Signalkatalog.

- [ListSignalCatalogs](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Signalkataloge ab.
- [ListSignalCatalogNodes](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Signale (Knoten) in einem bestimmten Signalkatalog ab.
- [GetSignalCatalog](#)— Ruft Informationen über einen Signalkatalog ab.

## Tutorials

- [Signale konfigurieren](#)
- [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#)
- [Importieren Sie einen Signalkatalog](#)
- [Aktualisieren Sie einen Signalkatalog \(AWS CLI\)](#)
- [Löscht einen Signalkatalog \(AWS CLI\)](#)
- [Ruft Informationen zum Signalkatalog ab \(AWS CLI\)](#)

## Signale konfigurieren

In diesem Abschnitt erfahren Sie, wie Sie Zweige, Attribute, Sensoren und Aktoren konfigurieren.

### Themen

- [Zweige konfigurieren](#)
- [Attribute konfigurieren](#)
- [Sensoren oder Aktoren konfigurieren](#)
- [Konfigurieren Sie komplexe Datentypen](#)

## Zweige konfigurieren

Um einen Zweig zu konfigurieren, geben Sie die folgenden Informationen an.

- `fullyQualifiedName`— Der vollqualifizierte Name des Zweigs besteht aus dem Pfad zum Zweig plus dem Namen des Zweigs. Verwenden Sie einen Punkt (`.`), um auf einen untergeordneten Zweig zu verweisen. Dies `Vehicle.Chassis.SteeringWheel` ist beispielsweise der vollständig qualifizierte Name für den `SteeringWheel` Zweig. `Vehicle.Chassis` ist der Pfad zu diesem Zweig.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9, Doppelpunkt (:) und Unterstrich (\_).

- (Optional) `Description` — Die Beschreibung für den Zweig.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: a—z, A—Z, 0—9,; (Doppelpunkt), \_ (Unterstrich) und - (Bindestrich).

- (Optional) `deprecationMessage` — Die veraltete Meldung für den Knoten oder Zweig, der verschoben oder gelöscht wird.

Die `DeprecationMessage` kann bis zu 2048 Zeichen enthalten. Gültige Zeichen: a—z, A—Z, 0—9,; (Doppelpunkt), \_ (Unterstrich) und - (Bindestrich).

- (Optional) `comment` — Ein Kommentar zusätzlich zur Beschreibung. Ein Kommentar kann verwendet werden, um zusätzliche Informationen über die Filiale bereitzustellen, z. B. die Begründung für die Filiale oder Verweise auf verwandte Zweige.

Der Kommentar kann bis zu 2048 Zeichen lang sein. Zulässige Zeichen: a—z, A—Z, 0—9,; (Doppelpunkt), \_ (Unterstrich) und - (Bindestrich).

## Attribute konfigurieren

Um ein Attribut zu konfigurieren, geben Sie die folgenden Informationen an.

- `dataType`— Der Datentyp des Attributs muss einer der folgenden sein: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY, UNKNOWN, oder eine benutzerdefinierte Struktur, die im Datentypzweig definiert ist. `fullyQualifiedName`
- `fullyQualifiedName`— Der vollqualifizierte Name des Attributs ist der Pfad zum Attribut plus der Name des Attributs. Verwenden Sie einen Punkt (.), um auf ein untergeordnetes Signal zu verweisen. `Vehicle.Chassis.SteeringWheel.Diameter` ist beispielsweise der vollqualifizierte Name für das `Diameter` Attribut. `Vehicle.Chassis.SteeringWheel` ist der Pfad zu diesem Attribut.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9,; (Doppelpunkt) und \_ (Unterstrich).

- (Optional) `Description` — Die Beschreibung für das Attribut.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `unit` — Die wissenschaftliche Einheit für das Attribut, z. B. `km` oder `Celsius`.
- (Optional) `min` — Der Mindestwert des Attributs.
- (Optional) `max` — Der Höchstwert des Attributs.
- (Optional) `defaultValue` — Der Standardwert des Attributs.
- (Optional) `assignedValue` — Der dem Attribut zugewiesene Wert.
- (Optional) `allowedValues` — Eine Liste von Werten, die das Attribut akzeptiert.
- (Optional) `deprecationMessage` — Die Verfallsmeldung für den Knoten oder Zweig, der verschoben oder gelöscht wird.

Die `DeprecationMessage` kann bis zu 2048 Zeichen enthalten. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `comment` — Ein Kommentar zusätzlich zur Beschreibung. Ein Kommentar kann verwendet werden, um zusätzliche Informationen über das Attribut bereitzustellen, z. B. die Begründung für das Attribut oder Verweise auf verwandte Attribute.

Der Kommentar kann bis zu 2048 Zeichen lang sein. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

## Sensoren oder Aktoren konfigurieren

Um einen Sensor oder Aktuator zu konfigurieren, geben Sie die folgenden Informationen an.

- `dataType`— Der Datentyp des Signals muss einer der folgenden sein: `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, `INT64`, `UINT64`, `BOOLEAN`, `FLOAT`, `DOUBLE`, `STRING`, `UNIX_TIMESTAMP`, `INT8_ARRAY`, `UINT8_ARRAY`, `INT16_ARRAY`, `UINT64_ARRAY`, `BOOLEAN_ARRAY`, `FLOAT_ARRAY`, `DOUBLE_ARRAY`, `STRING_ARRAY`, `UNIX_TIMESTAMP_ARRAY`, `UNKNOWN`, oder eine benutzerdefinierte Struktur, die im Datentypzweig definiert ist. `fullyQualifiedName`
- `fullyQualifiedName`— Der vollqualifizierte Name des Signals ist der Pfad zum Signal plus der Name des Signals. Verwenden Sie einen Punkt (`.`), um auf ein untergeordnetes Signal zu verweisen. Dies `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`

ist beispielsweise der vollqualifizierte Name für den HandsOffSteeringState Aktuator. `Vehicle.Chassis.SteeringWheel.HandsOff` ist der Pfad zu diesem Aktuator.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt) und `_` (Unterstrich).

- (Optional) `Description` — Die Beschreibung für das Signal.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `unit` — Die wissenschaftliche Einheit für das Signal, z. B. `km` oder `Celsius`.
- (Optional) `min` — Der Mindestwert des Signals.
- (Optional) `max` — Der Maximalwert des Signals.
- (Optional) `assignedValue` — Der dem Signal zugewiesene Wert.
- (Optional) `allowedValues` — Liste der Werte, die das Signal akzeptiert.
- (Optional) `deprecationMessage` — Die Verfallsmeldung für den Knoten oder Zweig, der verschoben oder gelöscht wird.

Die `DeprecationMessage` kann bis zu 2048 Zeichen enthalten. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `comment` — Ein Kommentar zusätzlich zur Beschreibung. Ein Kommentar kann verwendet werden, um zusätzliche Informationen über den Sensor oder Aktuator bereitzustellen, z. B. dessen Begründung oder Verweise auf verwandte Sensoren oder Aktoren.

Der Kommentar kann bis zu 2048 Zeichen lang sein. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

## Konfigurieren Sie komplexe Datentypen

Bei der Modellierung von Bildverarbeitungssystemen werden komplexe Datentypen verwendet. Diese Datentypen bestehen nicht nur aus Verzweigungen, sondern auch aus Strukturen (auch Struktur genannt) und Eigenschaften. Eine Struktur ist ein Signal, das durch mehrere Werte beschrieben wird, wie ein Bild. Eine Eigenschaft stellt ein Element der Struktur dar, z. B. einen primitiven Datentyp (wie `UINT8`) oder eine andere Struktur (wie `Timestamp`). Beispielsweise steht `Vehicle.Cameras.Front` für einen Zweig, `Vehicle.Cameras.Front.Image` für eine Struktur und `Vehicle.Cameras.Timestamp` für eine Eigenschaft.

Das folgende Beispiel für einen komplexen Datentyp zeigt, wie Signale und Datentypen in eine einzige JSON-Datei exportiert werden.

### Example komplexer Datentyp

```
{
  "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

Sie können ein [Demo-Skript](#) herunterladen, um ROS 2-Nachrichten in VSS-JSON-Dateien zu konvertieren, die mit dem Signalkatalog kompatibel sind. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Struktur konfigurieren

Um eine benutzerdefinierte Struktur (oder Struktur) zu konfigurieren, geben Sie die folgenden Informationen an.

- `fullyQualifiedName`— Der vollqualifizierte Name der benutzerdefinierten Struktur. Der vollqualifizierte Name einer benutzerdefinierten Struktur könnte beispielsweise `lautenComplexDataTypes.VehicleDataTypes.SVMCamera`.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt) und `_` (Unterstrich).

- (Optional) `Description` — Die Beschreibung für das Signal.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `deprecationMessage` — Die veraltete Meldung für den Knoten oder Zweig, der verschoben oder gelöscht wird.

Die `DeprecationMessage` kann bis zu 2048 Zeichen enthalten. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `comment` — Ein Kommentar zusätzlich zur Beschreibung. Ein Kommentar kann verwendet werden, um zusätzliche Informationen über den Sensor oder Aktuator bereitzustellen, z. B. dessen Begründung oder Verweise auf verwandte Sensoren oder Aktoren.

Der Kommentar kann bis zu 2048 Zeichen lang sein. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

## Eigenschaft konfigurieren

Um eine benutzerdefinierte Eigenschaft zu konfigurieren, geben Sie die folgenden Informationen an.

- `dataType`— Der Datentyp des Signals muss einer der folgenden sein: `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, `INT64`, `UINT64`, `BOOLEAN`, `FLOAT`, `DOUBLE`, `STRING`, `UNIX_TIMESTAMP`, `INT8_ARRAY`, `UINT8_ARRAY`, `INT16_ARRAY`,

UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY, STRUCT, STRUCT\_ARRAY oder UNKNOWN.

- `fullyQualifiedName`— Der vollständig qualifizierte Name der benutzerdefinierten Eigenschaft. Der vollqualifizierte Name einer benutzerdefinierten Eigenschaft könnte beispielsweise `lautenComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt) und `_` (Unterstrich)

- (Optional) `Description` — Die Beschreibung für das Signal.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `deprecationMessage` — Die veraltete Meldung für den Knoten oder Zweig, der verschoben oder gelöscht wird.

Die `DeprecationMessage` kann bis zu 2048 Zeichen enthalten. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `comment` — Ein Kommentar zusätzlich zur Beschreibung. Ein Kommentar kann verwendet werden, um zusätzliche Informationen über den Sensor oder Aktuator bereitzustellen, z. B. dessen Begründung oder Verweise auf verwandte Sensoren oder Aktoren.

Der Kommentar kann bis zu 2048 Zeichen lang sein. Zulässige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt), `_` (Unterstrich) und `-` (Bindestrich).

- (Optional) `dataEncoding` — Gibt an, ob es sich bei der Eigenschaft um Binärdaten handelt. Die Datenkodierung der benutzerdefinierten Eigenschaft muss einer der folgenden Werte entsprechen: `BINARY` oder `TYPED`.
- (Optional) `structFullyQualifiedName` — Der vollqualifizierte Name des Strukturknotens (Struct) für die benutzerdefinierte Eigenschaft, wenn der Datentyp der benutzerdefinierten Eigenschaft `Struct` oder `StructArray` ist.

Der vollqualifizierte Name kann bis zu 150 Zeichen lang sein. Gültige Zeichen: `a—z`, `A—Z`, `0—9`, `:` (Doppelpunkt) und `_` (Unterstrich).

## Erstellen AWS CLI Sie einen Signalkatalog ()

Sie können die [CreateSignalCatalog](#) API-Operation verwenden, um einen Signalkatalog zu erstellen. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um einen Signalkatalog zu erstellen.

*signal-catalog-configuration* Ersetzen Sie ihn durch den Namen der JSON-Datei, die die Konfiguration enthält.

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

- *signal-catalog-name* Ersetzen Sie es durch den Namen des Signalkatalogs, den Sie erstellen.
- (Optional) Ersetzen Sie die *Beschreibung* durch eine Beschreibung, damit Sie den Signalkatalog leichter identifizieren können.

Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktoren finden Sie unter [Signale konfigurieren](#).

```
{
  "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

Sie können ein [Demoskript](#) herunterladen, um ROS 2-Nachrichten in VSS-JSON-Dateien zu konvertieren, die mit dem Signalkatalog kompatibel sind. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Importieren Sie einen Signalkatalog

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um einen Signalkatalog zu importieren.

### Themen

- [Importieren Sie einen Signalkatalog \(Konsole\)](#)
- [Importieren Sie einen Signalkatalog \(AWS CLI\)](#)

## Importieren Sie einen Signalkatalog (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um einen Signalkatalog zu importieren.

**⚠ Important**

Sie können maximal einen Signalkatalog haben. Wenn Sie bereits über einen Signalkatalog verfügen, wird die Option zum Importieren eines Signalkatalogs in der Konsole nicht angezeigt.

Um einen Signalkatalog zu importieren

1. Öffnen Sie die [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Signal-Katalog aus.
3. Wählen Sie auf der Übersichtsseite des Signalkatalogs die Option Signalkatalog importieren aus.
4. Importieren Sie die Datei, die die Signale enthält.
  - Um eine Datei aus einem S3-Bucket hochzuladen:
    - a. Wählen Sie Import from S3 (Import aus S3).
    - b. Wählen Sie S3 durchsuchen.
    - c. Geben Sie für Buckets den Bucket-Namen oder das Objekt ein, wählen Sie es aus der Liste aus und wählen Sie dann die Datei aus der Liste aus. Wählen Sie die Schaltfläche „Datei auswählen“.

Oder geben Sie für S3-URI eine Amazon Simple Storage Service-URI ein. Weitere Informationen finden Sie unter [Methoden für den Zugriff auf einen Bucket](#) im Amazon S3 S3-Benutzerhandbuch.

- Um eine Datei von Ihrem Computer hochzuladen:
    - a. Wählen Sie Aus Datei importieren.
    - b. Laden Sie eine JSON-Datei im Format [Vehicle Signal Specification \(VSS\)](#) hoch.
5. Überprüfen Sie den Signalkatalog und wählen Sie dann Datei importieren.

## Importieren Sie einen Signalkatalog (AWS CLI)

Sie können den [ImportSignalCatalog](#) API-Vorgang verwenden, um eine JSON-Datei hochzuladen, die bei der Erstellung eines Signalkatalogs hilft. Sie müssen die [Vehicle Signal Specification \(VSS\)](#) befolgen, um Signale in der JSON-Datei zu speichern. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um einen Signalkatalog zu importieren.

- *signal-catalog-name* Ersetzen Sie ihn durch den Namen des Signalkatalogs, den Sie erstellen.
- (Optional) Ersetzen Sie die Beschreibung durch eine *Beschreibung*, damit Sie den Signalkatalog leichter identifizieren können.
- *signal-catalog-configuration-vss* Ersetzen Sie es durch den Namen der JSON-Zeichenfolgedatei, die in VSS definierte Signale enthält.

Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktuatoren finden Sie unter. [Signale konfigurieren](#)

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

Das JSON muss stringifiziert und durch das Feld übergeben werden. `vssJson` Das Folgende ist ein Beispiel für Signale, die in VSS definiert sind.

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

```

    }
  }
}
}
}

```

Das folgende Beispiel zeigt dieselben Signale, die in VSS definiert sind, in einer JSON-Zeichenfolge.

```

{
  "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

Sie können ein [Demo-Skript](#) herunterladen, um ROS 2-Nachrichten in VSS-JSON-Dateien zu konvertieren, die mit dem Signalkatalog kompatibel sind. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Aktualisieren Sie einen Signalkatalog (AWS CLI)

Sie können den [UpdateSignalCatalog](#) API-Vorgang verwenden, um einen vorhandenen Signalkatalog zu aktualisieren. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um einen vorhandenen Signalkatalog zu aktualisieren.

*signal-catalog-configuration* Ersetzen Sie ihn durch den Namen der JSON-Datei, die die Konfiguration enthält.

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

*signal-catalog-name* Ersetzen Sie ihn durch den Namen des Signalkatalogs, den Sie aktualisieren.

Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktoren finden Sie unter [Signale konfigurieren](#).

 Important

Benutzerdefinierte Strukturen sind unveränderlich. Wenn Sie Eigenschaften einer vorhandenen benutzerdefinierten Struktur (Struktur) neu anordnen oder in sie einfügen müssen, löschen Sie die Struktur und erstellen Sie eine brandneue Struktur mit der gewünschten Reihenfolge der Eigenschaften.

Um eine benutzerdefinierte Struktur zu löschen, fügen Sie den vollständig qualifizierten Namen der Struktur hinzu. `nodesToRemove` Eine Struktur kann nicht gelöscht werden, wenn Signale auf sie verweisen. Alle Signale, die sich auf die Struktur beziehen (ihr Datentyp ist

als Zielstruktur definiert), müssen vor der Anforderung zur Aktualisierung des Signalkatalogs aktualisiert oder gelöscht werden.

```
{
  "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",
    "fullyQualifiedNames": "Vehicle.Chassis.SteeringWheel.Diameter",
    "max": 55
  }
}]
}
```

## Löscht einen Signalkatalog (AWS CLI)

Sie können die [DeleteSignalCatalog](#) API-Operation verwenden, um einen Signalkatalog zu löschen. Das folgende Beispiel verwendet AWS CLI.

### Wichtig

Stellen Sie vor dem Löschen eines Signalkatalogs sicher, dass ihm keine Fahrzeugmodelle, Decoderlisten, Fahrzeuge, Flotten oder Kampagnen zugeordnet sind. Detaillierte Informationen finden Sie hier:

- [Löschen Sie ein Fahrzeugmodell](#)
- [Löschen Sie ein Decoder-Manifest](#)
- [Ein Fahrzeug löschen](#)
- [Löschen Sie eine Flotte \(AWS CLI\)](#)
- [Löscht eine Kampagne](#)

Führen Sie den folgenden Befehl aus, um einen vorhandenen Signalkatalog zu löschen. *signal-catalog-name* Ersetzen Sie ihn durch den Namen des Signalkatalogs, den Sie löschen möchten.

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

### Note

Dieser Befehl erzeugt keine Ausgabe.

## Ruft Informationen zum Signalkatalog ab (AWS CLI)

Sie können den [ListSignalCatalogs](#)API-Vorgang verwenden, um zu überprüfen, ob ein Signalkatalog gelöscht wurde. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Signalkataloge abzurufen.

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

Sie können den [ListSignalCatalogNodes](#)API-Vorgang verwenden, um zu überprüfen, ob ein Signalkatalog aktualisiert wurde. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Signale (Knoten) in einem bestimmten Signalkatalog abzurufen.

*signal-catalog-name* Ersetzen Sie es durch den Namen des Signalkatalogs, den Sie überprüfen.

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

Sie können die [GetSignalCatalog](#)API-Operation verwenden, um Signalkataloginformationen abzurufen. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um Informationen über einen Signalkatalog abzurufen.

*signal-catalog-name* Ersetzen Sie ihn durch den Namen des Signalkatalogs, den Sie abrufen möchten.

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

### Note

Diese Operation ist [letztlich konsistent](#). Mit anderen Worten, Änderungen am Signalkatalog werden möglicherweise nicht sofort übernommen.

## Fahrzeugmodelle erstellen und verwalten

Sie verwenden Signale, um Fahrzeugmodelle zu erstellen, die dabei helfen, das Format Ihrer Fahrzeuge zu standardisieren. Fahrzeugmodelle sorgen für konsistente Informationen für mehrere

Fahrzeuge desselben Typs, sodass Sie Daten von Fahrzeugflotten verarbeiten können. Fahrzeuge, die mit demselben Fahrzeugmodell erstellt wurden, erben dieselbe Gruppe von Signalen. Weitere Informationen finden Sie unter [Fahrzeuge erstellen, bereitstellen und verwalten](#).

Jedes Fahrzeugmodell hat ein Statusfeld, das den Status des Fahrzeugmodells enthält. Der Zustand kann einer der folgenden Werte sein:

- ACTIVE— Das Fahrzeugmodell ist aktiv.
- DRAFT— Die Konfiguration des Fahrzeugmodells wird gespeichert.

#### Important

- Wenn Sie die `CreateModelManifest` API-Operation verwenden möchten, um das erste Fahrzeugmodell zu erstellen, müssen Sie zuerst einen Signalkatalog erstellen. Weitere Informationen finden Sie unter [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#).
- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeugmodell zu erstellen, aktiviert AWS IoT das Fahrzeugmodell FleetWise automatisch für Sie.
- Wenn Sie den `CreateModelManifest` API-Vorgang verwenden, um ein Fahrzeugmodell zu erstellen, bleibt das Fahrzeugmodell im DRAFT Status.
- Sie können keine Fahrzeuge anhand von Fahrzeugmodellen erstellen, die sich in diesem DRAFT Bundesstaat befinden. Verwenden Sie den `UpdateModelManifest` API-Vorgang, um Fahrzeugmodelle auf den jeweiligen ACTIVE Status umzustellen.
- Sie können keine Fahrzeugmodelle bearbeiten, die sich in diesem ACTIVE Bundesstaat befinden.

#### Themen

- [Erstellen Sie ein Fahrzeugmodell](#)
- [Aktualisieren Sie ein Fahrzeugmodell \(\)AWS CLI](#)
- [Löschen Sie ein Fahrzeugmodell](#)
- [Informationen zum Fahrzeugmodell abrufen \(AWS CLI\)](#)

## Erstellen Sie ein Fahrzeugmodell

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Fahrzeugmodelle zu erstellen.

**⚠ Important**

Sie benötigen einen Signalkatalog, bevor Sie mithilfe der `CreateModelManifest` API-Operation ein Fahrzeugmodell erstellen können.

## Themen

- [Erstellen Sie ein Fahrzeugmodell \(Konsole\)](#)
- [Erstellen Sie ein Fahrzeugmodell \(AWS CLI\)](#)

## Erstellen Sie ein Fahrzeugmodell (Konsole)

In der AWS FleetWise IoT-Konsole können Sie auf folgende Weise ein Fahrzeugmodell erstellen:

- [Verwenden Sie eine Vorlage von AWS](#)
- [Manuelles Erstellen Sie ein Fahrzeugmodell](#)
- [Duplizieren Sie ein Fahrzeugmodell](#)

## Verwenden Sie eine Vorlage von AWS

AWS IoT FleetWise bietet eine On-Board Diagnostic (OBD) II, J1979-Vorlage, die automatisch einen Signalkatalog, ein Fahrzeugmodell und ein Decoder-Manifest für Sie erstellt. Die Vorlage fügt dem Decoder-Manifest auch OBD-Netzwerkschnittstellen hinzu. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

Um ein Fahrzeugmodell mithilfe einer Vorlage zu erstellen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie auf der Seite Fahrzeugmodelle die Option Bereitgestellte Vorlage hinzufügen aus.
4. Wählen Sie On-Board-Diagnose (OBD) II aus.
5. Geben Sie einen Namen für die OBD-Netzwerkschnittstelle ein, die AWS IoT FleetWise erstellt.
6. Wählen Sie Hinzufügen aus.

## Manuelles Erstellen Sie ein Fahrzeugmodell

Sie können Signale aus dem Signalkatalog hinzufügen oder Signale importieren, indem Sie eine oder mehrere DBC-Dateien hochladen. Eine.dbc-Datei ist ein Dateiformat, das CAN-Bus-Datenbanken (Controller Area Network) unterstützen.

### Important

Mit der AWS FleetWise IoT-Konsole können Sie kein Fahrzeugmodell mit Datensignalen des Bildverarbeitungssystems erstellen. Verwenden Sie stattdessen die, AWS CLI um ein Fahrzeugmodell zu erstellen.

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

Um ein Fahrzeugmodell manuell zu erstellen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie auf der Seite Fahrzeugmodelle die Option Fahrzeugmodell erstellen aus, und gehen Sie dann wie folgt vor.

### Themen

- [Schritt 1: Fahrzeugmodell konfigurieren](#)
- [Schritt 2: Signale hinzufügen](#)
- [Schritt 3: Signale importieren](#)
- [\(Optional\) Schritt 4: Attribute hinzufügen](#)
- [Schritt 5: Überprüfen und Erstellen](#)

### Schritt 1: Fahrzeugmodell konfigurieren

Gehen Sie unter Allgemeine Informationen wie folgt vor.

1. Geben Sie einen Namen für das Fahrzeugmodell ein.
2. (Optional) Geben Sie eine Beschreibung ein.

### 3. Wählen Sie Weiter aus.

#### Schritt 2: Signale hinzufügen

##### Note

- Wenn Sie AWS IoT zum ersten Mal verwenden FleetWise, ist dieser Schritt erst verfügbar, wenn Sie einen Signalkatalog haben. Wenn das erste Fahrzeugmodell erstellt wird, erstellt AWS IoT FleetWise automatisch einen Signalkatalog mit Signalen, die dem ersten Fahrzeugmodell hinzugefügt wurden.
- Wenn Sie Erfahrung mit AWS IoT haben FleetWise, können Sie Ihrem Fahrzeugmodell Signale hinzufügen, indem Sie Signale aus dem Signalkatalog auswählen oder .dbc-Dateien hochladen, um Signale zu importieren.
- Sie benötigen mindestens ein Signal, um ein Fahrzeugmodell zu erstellen.

#### Um Signale hinzuzufügen

1. Wählen Sie ein oder mehrere Signale aus dem Signalkatalog aus, den Sie dem Fahrzeugmodell hinzufügen möchten. Sie können die ausgewählten Signale im rechten Bereich überprüfen.

##### Note

Nur ausgewählte Signale werden dem Fahrzeugmodell hinzugefügt.

2. Wählen Sie Weiter aus.

#### Schritt 3: Signale importieren

##### Note

- Wenn Sie AWS IoT zum ersten Mal verwenden FleetWise, müssen Sie mindestens eine .dbc-Datei hochladen, um Signale zu importieren.
- Wenn Sie Erfahrung mit AWS IoT haben FleetWise, können Sie Ihrem Fahrzeugmodell Signale hinzufügen, indem Sie Signale aus dem Signalkatalog auswählen oder .dbc-Dateien hochladen, um Signale zu importieren.

- Sie benötigen mindestens ein Signal, um ein Fahrzeugmodell zu erstellen.

## Um Signale zu importieren

1. Wählen Sie Dateien auszuwählen.
2. Wählen Sie im Dialogfeld die .dbc-Datei aus, die Signale enthält. Sie können mehrere .dbc-Dateien hochladen.
3. AWS IoT FleetWise analysiert Ihre DBC-Dateien, um Signale abzurufen.

Geben Sie im Abschnitt Signale die folgenden Metadaten für jedes Signal an.

- Name — Der Name des Signals.

Der Signalname muss eindeutig sein. Der Signalname und der Pfad können bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9,: (Doppelpunkt) und \_ (Unterstrich).

- Datentyp — Der Datentyp des Signals muss einer der folgenden sein: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, INT64\_ARRAY ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY oder UNKNOWN.
- Signaltyp — Der Typ des Signals, bei dem es sich um einen Sensor oder einen Aktuator handeln kann.
- (Optional) Einheit — Die wissenschaftliche Einheit für das Signal, z. B. km oder Celsius.
- (Optional) Pfad — Der Pfad zum Signal. Verwenden Sie ähnlich wie bei JsonPath einen Punkt (.), um auf ein untergeordnetes Signal zu verweisen. z. B. **Vehicle.Engine.Light**.

Der Signalname plus der Pfad können bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9,: (Doppelpunkt) und \_ (Unterstrich).

- (Optional) Min — Der Mindestwert des Signals.
- (Optional) Max — Der Maximalwert des Signals.
- (Optional) Beschreibung — Die Beschreibung für das Signal.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: a—z, A—Z, 0—9,: (Doppelpunkt), \_ (Unterstrich) und - (Bindestrich).

4. Wählen Sie Weiter aus.

## (Optional) Schritt 4: Attribute hinzufügen

Sie können bis zu 100 Attribute hinzufügen, einschließlich der vorhandenen Attribute im Signalkatalog.

Um Attribute hinzuzufügen

1. Geben Sie unter Attribute hinzufügen die folgenden Metadaten für jedes Attribut an.

- Name — Der Name des Attributs.

Der Signalname muss eindeutig sein. Der Signalname und der Pfad können bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9, : (Doppelpunkt) und \_ (Unterstrich)

- Datentyp — Der Datentyp des Attributs muss einer der folgenden sein: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, INT64\_ARRAY ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY oder UNKNOWN
- (Optional) Einheit — Die wissenschaftliche Einheit für das Attribut, z. B. km oder Celsius.
- (Optional) Pfad — Der Pfad zum Signal. Verwenden Sie ähnlich wie bei JsonPath einen Punkt (.), um auf ein untergeordnetes Signal zu verweisen. z. B. **Vehicle.Engine.Light**.

Der Signalname plus der Pfad können bis zu 150 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9, : (Doppelpunkt) und \_ (Unterstrich)

- (Optional) Min — Der Mindestwert des Attributs.
- (Optional) Max — Der Höchstwert des Attributs.
- (Optional) Beschreibung — Die Beschreibung für das Attribut.

Die Beschreibung kann bis zu 2048 Zeichen enthalten. Zulässige Zeichen: a—z, A—Z, 0—9, : (Doppelpunkt), \_ (Unterstrich) und - (Bindestrich).

2. Wählen Sie Weiter.

## Schritt 5: Überprüfen und Erstellen

Überprüfen Sie die Konfigurationen für das Fahrzeugmodell und wählen Sie dann Create.

## Duplizieren Sie ein Fahrzeugmodell

AWS IoT FleetWise kann die Konfigurationen eines vorhandenen Fahrzeugmodells kopieren, um ein neues Modell zu erstellen. Die im ausgewählten Fahrzeugmodell angegebenen Signale werden in das neue Fahrzeugmodell kopiert.

Um ein Fahrzeugmodell zu duplizieren

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie ein Modell aus der Fahrzeugmodellliste und dann Modell duplizieren aus.

Folgen Sie der [Manuelles Erstellen Sie ein Fahrzeugmodell](#) Anleitung, um das Fahrzeugmodell zu konfigurieren.

Es kann einige Minuten dauern, FleetWise bis AWS IoT Ihre Anfrage zur Erstellung des Fahrzeugmodells bearbeitet hat. Nachdem das Fahrzeugmodell erfolgreich erstellt wurde, wird auf der Seite Fahrzeugmodelle in der Spalte Status der Wert AKTIV angezeigt. Wenn das Fahrzeugmodell aktiv wird, können Sie es nicht bearbeiten.

## Erstellen Sie ein Fahrzeugmodell (AWS CLI)

Sie können den [CreateModelManifest](#) API-Vorgang verwenden, um Fahrzeugmodelle (Modellmanifeste) zu erstellen. Das folgende Beispiel verwendet die AWS CLI.

### Important

Wenn Sie die AWS FleetWise IoT-API verwenden möchten, um das erste Fahrzeugmodell zu erstellen, müssen Sie zuerst einen Signalkatalog erstellen. Weitere Informationen zur Erstellung eines Signalkatalogs finden Sie unter [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#).

Führen Sie den folgenden Befehl aus, um ein Fahrzeugmodell zu erstellen.

*vehicle-model-configuration* Ersetzen Sie es durch den Namen der JSON-Datei, die die Konfiguration enthält.

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

- *vehicle-model-name* Ersetzen Sie es durch den Namen des Fahrzeugmodells, das Sie erstellen.
- Ersetzen Sie *Signal-Catalog-ARN* durch den Amazon Resource Name (ARN) des Signalkatalogs.
- (Optional) Ersetzen Sie die *Beschreibung* durch eine Beschreibung, um das Fahrzeugmodell leichter identifizieren zu können.

Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktuatoren finden Sie unter [Signale konfigurieren](#).

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

## Aktualisieren Sie ein Fahrzeugmodell ()AWS CLI

Sie können den [UpdateModelManifest](#) API-Vorgang verwenden, um ein vorhandenes Fahrzeugmodell (Modellmanifeste) zu aktualisieren. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um ein vorhandenes Fahrzeugmodell zu aktualisieren.

*update-vehicle-model-configuration* Ersetzen Sie ihn durch den Namen der JSON-Datei, die die Konfiguration enthält.

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

- *vehicle-model-name* Ersetzen Sie es durch den Namen des Fahrzeugmodells, das Sie aktualisieren möchten.
- (Optional) Um das Fahrzeugmodell zu aktivieren, *vehicle-model-status* ersetzen Sie es durch ACTIVE.

**⚠ Important**

Nachdem das Fahrzeugmodell aktiviert wurde, können Sie das Fahrzeugmodell nicht mehr ändern.

- (Optional) Ersetzen Sie die *Beschreibung* durch eine aktualisierte Beschreibung, um das Fahrzeugmodell leichter identifizieren zu können.

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

## Löschen Sie ein Fahrzeugmodell

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Fahrzeugmodelle zu löschen.

**⚠ Important**

Fahrzeuge und Decoder-Manifeste, die dem Fahrzeugmodell zugeordnet sind, müssen zuerst gelöscht werden. Weitere Informationen finden Sie unter [Ein Fahrzeug löschen](#) und [Löschen Sie ein Decoder-Manifest](#).

## Löschen Sie ein Fahrzeugmodell (Konsole)

Verwenden Sie die AWS FleetWise IoT-Konsole, um ein Fahrzeugmodell zu löschen.

Um ein Fahrzeugmodell zu löschen

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie auf der Seite Fahrzeugmodelle das Zielfahrzeugmodell aus.
4. Wählen Sie Löschen aus.

5. Unter Löschen **vehicle-model-name**? , geben Sie den Namen des Fahrzeugmodells ein, das gelöscht werden soll, und wählen Sie dann Bestätigen.

## Löschen Sie ein Fahrzeugmodell (AWS CLI)

Sie können den [DeleteModelManifest](#) API-Vorgang verwenden, um ein vorhandenes Fahrzeugmodell (Modellmanifeste) zu löschen. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um ein Fahrzeugmodell zu löschen.

*model-manifest-name* Ersetzen Sie es durch den Namen des Fahrzeugmodells, das Sie löschen möchten.

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

### Note

Dieser Befehl erzeugt keine Ausgabe.

## Informationen zum Fahrzeugmodell abrufen (AWS CLI)

Sie können den [ListModelManifests](#) API-Vorgang verwenden, um zu überprüfen, ob ein Fahrzeugmodell gelöscht wurde. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Fahrzeugmodelle abzurufen.

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

Sie können den [ListModelManifestNodes](#) API-Vorgang verwenden, um zu überprüfen, ob ein Fahrzeugmodell aktualisiert wurde. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Signale (Knoten) in einem bestimmten Fahrzeugmodell abzurufen.

*vehicle-model-name* Ersetzen Sie es durch den Namen des Fahrzeugmodells, das Sie überprüfen.

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

Führen Sie den folgenden Befehl aus, um Informationen zu einem Fahrzeugmodell abzurufen.

Ersetzen Sie *vehicle-model* durch den Namen des Fahrzeugmodells, das Sie abrufen möchten.

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

### Note

Diese Operation ist [letztlich konsistent](#). Mit anderen Worten, Änderungen am Fahrzeugmodell werden möglicherweise nicht sofort übernommen.

## Decoder-Manifeste erstellen und verwalten

Decoder-Manifeste enthalten Dekodierungsinformationen, die das AWS IoT FleetWise verwendet, um Fahrzeugdaten (Binärdaten) in für Menschen lesbare Werte umzuwandeln und Ihre Daten für Datenanalysen aufzubereiten. Netzwerkschnittstellen- und Decodersignale sind die Kernkomponenten, mit denen Sie bei der Konfiguration von Decoder-Manifesten arbeiten.

### Netzwerkschnittstelle

Enthält Informationen über das Protokoll, das das bordeigene Netzwerk verwendet. AWS IoT FleetWise unterstützt die folgenden Protokolle.

#### Controller Area Network (CAN-Bus)

Ein Protokoll, das definiert, wie Daten zwischen elektronischen Steuergeräten (ECUs) übertragen werden. Steuergeräte können das Motorsteuergerät, Airbags oder das Audiosystem sein.

#### On-Board-Diagnose (OBD) II

Ein weiterentwickeltes Protokoll, das definiert, wie Selbstdiagnosedaten zwischen Steuergeräten übertragen werden. Es bietet eine Reihe von Standard-Diagnose-Fehlercodes (DTCs), mit deren Hilfe Sie feststellen können, was mit Ihrem Fahrzeug nicht stimmt.

## Middleware für Fahrzeuge

Die Fahrzeug-Middleware ist als eine Art Netzwerkschnittstelle definiert. Zu den Beispielen für Fahrzeug-Middleware gehören das Robot Operating System (ROS 2) und die skalierbare serviceorientierte Middleware over IP (SOME/IP).

### Note

AWS IoT FleetWise unterstützt ROS 2-Middleware für Bildverarbeitungssystemdaten.

## Decodersignal

Bietet detaillierte Dekodierungsinformationen für ein bestimmtes Signal. Jedes im Fahrzeugmodell angegebene Signal muss mit einem Decodersignal gepaart werden. Wenn das Decoder-Manifest CAN-Netzwerkschnittstellen enthält, muss es CAN-Decodersignale enthalten. Wenn das Decoder-Manifest OBD-Netzwerkschnittstellen enthält, muss es OBD-Decodersignale enthalten.

Das Decoder-Manifest muss Nachrichtendecodersignale enthalten, wenn es auch Fahrzeug-Middleware-Schnittstellen enthält.

Jedes Decoder-Manifest muss einem Fahrzeugmodell zugeordnet sein. AWS IoT FleetWise verwendet das zugehörige Decoder-Manifest, um Daten von Fahrzeugen zu dekodieren, die auf der Grundlage des Fahrzeugmodells erstellt wurden.

Jedes Decoder-Manifest hat ein Statusfeld, das den Status des Decoder-Manifests enthält. Der Zustand kann einer der folgenden Werte sein:

- **ACTIVE**— Das Decoder-Manifest ist aktiv.
- **DRAFT**— Die Konfiguration des Decoder-Manifests wurde nicht gespeichert.
- **VALIDATING**— Das Decoder-Manifest wird derzeit auf seine Eignung überprüft. Dies gilt nur für Decoder-Manifeste, die mindestens ein Datensignal des Bildverarbeitungssystems enthalten.
- **INVALID**— Das Decoder-Manifest konnte nicht validiert werden und kann noch nicht aktiviert werden. Dies gilt nur für Decoder-Manifeste, die mindestens ein Datensignal des Bildverarbeitungssystems enthalten. Sie können die `GetDecoderManifest` APIs `ListDecoderManifests` und verwenden, um den Grund für eine fehlgeschlagene Überprüfung zu überprüfen.

### Important

- Wenn Sie die AWS FleetWise IoT-Konsole verwenden, um ein Decoder-Manifest zu erstellen, aktiviert AWS IoT das Decoder-Manifest FleetWise automatisch für Sie.
- Wenn Sie den `CreateDecoderManifest` API-Vorgang verwenden, um ein Decoder-Manifest zu erstellen, bleibt das Decoder-Manifest im Status. DRAFT
- Sie können keine Fahrzeuge anhand von Fahrzeugmodellen erstellen, die mit einem DRAFT Decoder-Manifest verknüpft sind. Verwenden Sie den `UpdateDecoderManifest` API-Vorgang, um das Decoder-Manifest in den ACTIVE Status zu ändern.
- Sie können keine Decoder-Manifeste bearbeiten, die sich im ACTIVE Status befinden.

### Themen

- [Konfigurieren Sie Netzwerkschnittstellen und Decodersignale](#)
- [Erstellen Sie ein Decoder-Manifest](#)
- [Aktualisieren Sie ein Decoder-Manifest \(AWS CLI\)](#)
- [Löschen Sie ein Decoder-Manifest](#)
- [Ruft Informationen zum Decoder-Manifest ab \(AWS CLI\)](#)

## Konfigurieren Sie Netzwerkschnittstellen und Decodersignale

Jedes Decoder-Manifest hat mindestens eine Netzwerkschnittstelle und Decodersignale, die mit Signalen gepaart sind, die im zugehörigen Fahrzeugmodell spezifiziert sind.

Wenn das Decoder-Manifest CAN-Netzwerkschnittstellen enthält, muss es CAN-Decodersignale enthalten. Wenn das Decoder-Manifest OBD-Netzwerkschnittstellen enthält, muss es OBD-Decodersignale enthalten.

### Themen

- [Netzwerkschnittstellen konfigurieren](#)
- [Decodersignale konfigurieren](#)

## Netzwerkschnittstellen konfigurieren

Um eine CAN-Netzwerkschnittstelle zu konfigurieren, geben Sie die folgenden Informationen an.

- `name`— Der Name der CAN-Schnittstelle.

Der Schnittstellenname muss eindeutig sein und kann 1—100 Zeichen lang sein.

- (Optional) `protocolName` — Der Name des Protokolls.

Gültige Werte: `CAN-FD` und `CAN`

- (Optional) `protocolVersion` — AWS IoT unterstützt FleetWise derzeit `CAN-FD` und `CAN 2.0b`.

Gültige Werte: `1.0` und `2.0b`

Um eine OBD-Netzwerkschnittstelle zu konfigurieren, geben Sie die folgenden Informationen an.

- `name`— Der Name der OBD-Schnittstelle.

Der Schnittstellenname muss eindeutig sein und kann 1—100 Zeichen lang sein.

- `requestMessageId`— Die ID der Nachricht, die Daten anfordert.

- (Optional) `dtcRequestIntervalSeconds` — Wie oft innerhalb von Sekunden Diagnose-Fehlercodes (DTCs) vom Fahrzeug angefordert werden sollen. Wenn der angegebene Wert beispielsweise 120 ist, sammelt die Edge Agent-Software alle 2 Minuten gespeicherte Fehlercodes.

- (Optional) `hasTransmissionEcu` — Gibt an, ob das Fahrzeug über ein Getriebesteuergerät (TCM) verfügt.

Gültige Werte: `true` und `false`

- (Optional) `obdStandard` — Der OBD-Standard, den AWS IoT FleetWise unterstützt. AWS IoT unterstützt FleetWise derzeit den ISO15765-4-Standard WWH-OBD (World Wide Harmonization On-Board Diagnostics).

- (Optional) `pidRequestIntervalSeconds` — Wie oft OBD-II-PIDs vom Fahrzeug angefordert werden sollen. Wenn der angegebene Wert beispielsweise 120 ist, sammelt die Edge Agent-Software alle 2 Minuten OBD II-PIDs.

- (Optional) `useExtendedIds` — Ob erweiterte IDs in der Nachricht verwendet werden sollen.

Gültige Werte: `true` und `false`

Geben Sie die folgenden Informationen an, um eine Fahrzeug-Middleware-Netzwerkschnittstelle zu konfigurieren.

- `name`— Der Name der Middleware-Schnittstelle des Fahrzeugs.

Der Schnittstellename muss eindeutig sein und kann 1—100 Zeichen lang sein.

- `protocolName`— Der Name des Protokolls.

Zulässige Werte: `R0S_2`

## Decodersignale konfigurieren

Um ein CAN-Decodersignal zu konfigurieren, geben Sie die folgenden Informationen an.

- `factor`— Der Multiplikator, der zur Dekodierung der Nachricht verwendet wurde.
- `isBigEndian`— Ob die Byte-Reihenfolge der Nachricht Big-Endian ist. Wenn es Big-Endian ist, wird der signifikanteste Wert in der Sequenz zuerst gespeichert, und zwar an der niedrigsten Speicheradresse.
- `isSigned`— Ob die Nachricht signiert ist. Wenn sie signiert ist, kann die Nachricht sowohl positive als auch negative Zahlen enthalten.
- `length`— Die Länge der Nachricht in Byte.
- `messageId`— Die ID der Nachricht.
- `offset`— Der Offset, der zur Berechnung des Signalwerts verwendet wurde. In Kombination mit dem Faktor ist die Berechnung  $value = raw\_value * factor + offset$ .
- `startBit`— Gibt die Position des ersten Bits der Nachricht an.
- (Optional) `name` — Der Name des Signals.

Um ein OBD-Decodersignal zu konfigurieren, geben Sie die folgenden Informationen an.

- `byteLength`— Die Länge der Nachricht in Byte.
- `offset`— Der Offset, der zur Berechnung des Signalwerts verwendet wurde. In Kombination mit der Skalierung ist die Berechnung  $value = raw\_value * scaling + offset$ .
- `pid`— Der Diagnosecode, der verwendet wird, um von einem Fahrzeug eine Nachricht für dieses Signal anzufordern.
- `pidResponseLength`— Die Länge der angeforderten Nachricht.
- `scaling`— Der Multiplikator, der zur Dekodierung der Nachricht verwendet wurde.
- `serviceMode`— Der Betriebsmodus (Diagnosedienst) in einer Nachricht.
- `startByte`— Zeigt den Anfang der Nachricht an.

- (Optional) `bitMaskLength` — Die Anzahl der Bits, die in einer Nachricht maskiert sind.
- (Optional) `bitRightShift` — Die Anzahl der nach rechts verschobenen Positionen.

Um ein Nachrichtendecodersignal zu konfigurieren, geben Sie die folgenden Informationen an.

- `topicName`— Der Themenname für das Nachrichtensignal. Er entspricht den Themen in ROS 2. Weitere Hinweise zum strukturierten Nachrichtenobjekt finden Sie unter [StructuredMessage](#).
- `structuredMessage`— Die strukturierte Nachricht für das Nachrichtensignal. Sie kann entweder mit einer `primitiveMessageDefinition` `structuredMessageList` Definition oder `structuredMessageDefinition` rekursiv definiert werden.

## Erstellen Sie ein Decoder-Manifest

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um ein Decoder-Manifest für Ihr Fahrzeugmodell zu erstellen.

### Important

Sie benötigen ein Fahrzeugmodell, bevor Sie ein Decoder-Manifest erstellen können. Jedes Decoder-Manifest muss einem Fahrzeugmodell zugeordnet sein. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).

### Themen

- [Erstellen Sie ein Decoder-Manifest \(Konsole\)](#)
- [Erstellen Sie ein Decoder-Manifest \(AWS CLI\)](#)

## Erstellen Sie ein Decoder-Manifest (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um ein Decoder-Manifest zu erstellen, das Ihrem Fahrzeugmodell zugeordnet ist.

### Important

Sie können mit der AWS FleetWise IoT-Konsole keine Datensignale von Bildverarbeitungssystemen in Decoder-Manifesten konfigurieren. Verwenden Sie stattdessen

die AWS CLI. Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

Um ein Decoder-Manifest zu erstellen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie das Zielfahrzeugmodell aus.
4. Wählen Sie auf der Seite mit der Zusammenfassung des Fahrzeugmodells die Option Decoder-Manifest erstellen aus, und gehen Sie dann wie folgt vor.

Themen

- [Schritt 1: Decoder-Manifest konfigurieren](#)
- [Schritt 2: Netzwerkschnittstellen hinzufügen](#)
- [Schritt 3: Überprüfen und Erstellen](#)

Schritt 1: Decoder-Manifest konfigurieren

Gehen Sie unter Allgemeine Informationen wie folgt vor.

1. Geben Sie einen eindeutigen Namen für das Decoder-Manifest ein.
2. (Optional) Geben Sie eine Beschreibung ein.
3. Wählen Sie Weiter aus.

Schritt 2: Netzwerkschnittstellen hinzufügen

Jedes Decoder-Manifest muss mindestens eine Netzwerkschnittstelle haben. Sie können einem Decoder-Manifest mehrere Netzwerkschnittstellen hinzufügen.

Um eine Netzwerkschnittstelle hinzuzufügen

- Gehen Sie unter Netzwerkschnittstelle wie folgt vor.
  - a. Wählen Sie als Netzwerkschnittstellentyp CAN\_INTERFACE oder OBD\_INTERFACE.
  - b. Geben Sie einen eindeutigen Namen für Ihre Netzwerkschnittstelle ein.

- c. Geben Sie eine eindeutige Netzwerkschnittstellen-ID ein. Sie können die von AWS IoT generierte ID verwenden FleetWise.
- d. Wählen Sie ein oder mehrere in Ihrem Fahrzeugmodell angegebene Signale aus, um sie mit Decodersignalen zu koppeln.
- e. Laden Sie eine DBC-Datei hoch, um Dekodierungsinformationen bereitzustellen. AWS IoT FleetWise analysiert die .dbc-Datei, um Decodersignale abzurufen.
- f. Stellen Sie im Abschnitt Gepaarte Signale sicher, dass jedes Signal mit einem Decodersignal gepaart ist.
- g. Wählen Sie Weiter aus.

#### Note

- Sie können für jede Netzwerkschnittstelle nur eine .dbc-Datei hochladen.
- Stellen Sie sicher, dass jedes in Ihrem Fahrzeugmodell angegebene Signal mit einem Decodersignal gekoppelt ist.
- Nachdem Sie sich entschieden haben, eine weitere Netzwerkschnittstelle hinzuzufügen, können Sie die Netzwerkschnittstelle, die Sie gerade bearbeiten, nicht mehr bearbeiten. Sie können alle vorhandenen Netzwerkschnittstellen löschen.

### Schritt 3: Überprüfen und Erstellen

Überprüfen Sie die Konfigurationen für das Decoder-Manifest und wählen Sie dann Create.

### Erstellen Sie ein Decoder-Manifest ( )AWS CLI

Sie können den [CreateDecoderManifest](#) API-Vorgang verwenden, um Decoder-Manifeste zu erstellen. Das folgende Beispiel verwendet die AWS CLI.

#### Important

Bevor Sie ein Decoder-Manifest erstellen, müssen Sie zunächst ein Fahrzeugmodell erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeugmodell](#).

Führen Sie den folgenden Befehl aus, um ein Decoder-Manifest zu erstellen.

*decoder-manifest-configuration* Ersetzen Sie es durch den Namen der JSON-Datei, die die Konfiguration enthält.

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

- *decoder-manifest-name* Ersetzen Sie es durch den Namen des Decoder-Manifests, das Sie erstellen.
- Ersetzen Sie *Vehicle-Model-ARN* durch den Amazon Resource Name (ARN) des Fahrzeugmodells.
- (Optional) Ersetzen Sie die Beschreibung durch eine *Beschreibung*, damit Sie das Decoder-Manifest leichter identifizieren können.

Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktuatoren finden Sie unter: [Konfigurieren Sie Netzwerkschnittstellen und Decodersignale](#)

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [
    {
      "canInterface": {
        "name": "myNetworkInterface",
        "protocolName": "CAN",
        "protocolVersion": "2.0b"
      },
      "interfaceId": "Qq1acaenBy0B3sSM39SYm",
      "type": "CAN_INTERFACE"
    }
  ],
  "signalDecoders": [
    {
      "canSignal": {
        "name": "Engine_Idle_Time",
        "factor": 1,
        "isBigEndian": true,
        "isSigned": false,
        "length": 24,
        "messageId": 271343712,

```

```

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

```

- *decoder-manifest-name* Ersetzen Sie es durch den Namen des Decoder-Manifests, das Sie erstellen.
- Ersetzen Sie *Vehicle-Model-ARN* durch den Amazon Resource Name (ARN) des Fahrzeugmodells.
- (Optional) Ersetzen Sie die Beschreibung durch eine *Beschreibung*, damit Sie das Decoder-Manifest leichter identifizieren können.

Die Reihenfolge der Eigenschaftsknoten innerhalb einer Struktur (Struktur) muss konsistent bleiben, wie sie im Signalkatalog und im Fahrzeugmodell (Modellmanifest) definiert ist. Weitere Informationen zur Konfiguration von Verzweigungen, Attributen, Sensoren und Aktuatoren finden Sie unter [Konfigurieren Sie Netzwerkschnittstellen und Decodersignale](#).

```

{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",

```

```
"description": "description",
"networkInterfaces": [{
  "canInterface": {
    "name": "myNetworkInterface",
    "protocolName": "CAN",
    "protocolVersion": "2.0b"
  },
  "interfaceId": "Qq1acaenBy0B3sSM39SYm",
  "type": "CAN_INTERFACE"
}, {
  "type": "VEHICLE_MIDDLEWARE",
  "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
  "vehicleMiddleware": {
    "name": "ROS2_test",
    "protocolName": "ROS_2"
  }
}],
"signalDecoders": [{
  "canSignal": {
    "name": "Engine_Idle_Time",
    "factor": 1,
    "isBigEndian": true,
    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 16
  },
  "fullyQualified_name": "Vehicle.EngineIdleTime",
  "interfaceId": "Qq1acaenBy0B3sSM39SYm",
  "type": "CAN_SIGNAL"
},
{
  "canSignal": {
    "name": "Engine_Run_Time",
    "factor": 1,
    "isBigEndian": true,
    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 40
  },
  "fullyQualified_name": "Vehicle.EngineRunTime",
```

```
"interfaceId": "Qq1acaenBy0B3sSM39SYm",
"type": "CAN_SIGNAL"
},
{
  "fullyQualified_name": "Vehicle.CompressedImageTopic",
  "type": "MESSAGE_SIGNAL",
  "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
  "messageSignal": {
    "topicName": "CompressedImageTopic:sensor_msgs/msg/CompressedImage",
    "structuredMessage": {
      "structuredMessageDefinition": [{
        "fieldName": "header",
        "dataType": {
          "structuredMessageDefinition": [{
            "fieldName": "stamp",
            "dataType": {
              "structuredMessageDefinition": [{
                "fieldName": "sec",
                "dataType": {
                  "primitiveMessageDefinition": {
                    "ros2PrimitiveMessageDefinition": {
                      "primitiveType": "INT32"
                    }
                  }
                }
              ],
            },
            {
              "fieldName": "nanosec",
              "dataType": {
                "primitiveMessageDefinition": {
                  "ros2PrimitiveMessageDefinition": {
                    "primitiveType": "UINT32"
                  }
                }
              }
            }
          ],
        },
        {
          "fieldName": "frame_id",
          "dataType": {
            "primitiveMessageDefinition": {
              "ros2PrimitiveMessageDefinition": {
```

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

 Note

Sie können ein [Demo-Skript](#) herunterladen, um ein Decoder-Manifest mit Signalen des Bildverarbeitungssystems zu erstellen. Weitere Informationen finden Sie im [Vision System Data Developer Guide](#).

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

## Aktualisieren Sie ein Decoder-Manifest ( )AWS CLI

Sie können den [UpdateDecoderManifest](#)API-Vorgang verwenden, um ein Decoder-Manifest zu aktualisieren. Sie können Netzwerkschnittstellen und Signaldecoder hinzufügen, entfernen und aktualisieren. Sie können auch den Status des Decoder-Manifests ändern. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um ein Decoder-Manifest zu aktualisieren.

*decoder-manifest-name* Ersetzen Sie es durch den Namen des Decoder-Manifests, das Sie aktualisieren.

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

 Important

Nachdem Sie das Decoder-Manifest aktiviert haben, können Sie es nicht mehr bearbeiten.

## Löschen Sie ein Decoder-Manifest

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um ein Decoder-Manifest zu löschen.

 **Important**

Fahrzeuge, die dem Decoder-Manifest zugeordnet sind, müssen zuerst gelöscht werden. Weitere Informationen finden Sie unter [Ein Fahrzeug löschen](#).

## Themen

- [Löschen Sie ein Decoder-Manifest \(Konsole\)](#)
- [Löschen Sie ein Decoder-Manifest \(\)AWS CLI](#)

## Löschen Sie ein Decoder-Manifest (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um ein Decoder-Manifest zu löschen.

Um ein Decoder-Manifest zu löschen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeugmodelle aus.
3. Wählen Sie das Zielfahrzeugmodell aus.
4. Wählen Sie auf der Seite mit der Zusammenfassung des Fahrzeugmodells die Registerkarte Decoder-Manifeste aus.
5. Wählen Sie das Ziel-Decoder-Manifest und klicken Sie dann auf Löschen.
6. Unter Löschen? **decoder-manifest-name** , geben Sie den Namen des Decoder-Manifests ein, das gelöscht werden soll, und wählen Sie dann Confirm.

## Löschen Sie ein Decoder-Manifest ()AWS CLI

Sie können den [DeleteDecoderManifest](#)API-Vorgang verwenden, um ein Decoder-Manifest zu löschen. Das folgende Beispiel verwendet AWS CLI.

 **Important**

Bevor Sie das Decoder-Manifest löschen, löschen Sie zuerst die zugehörigen Fahrzeuge. Weitere Informationen finden Sie unter [Ein Fahrzeug löschen](#).

Führen Sie den folgenden Befehl aus, um ein Decoder-Manifest zu löschen.

*decoder-manifest-name* Ersetzen Sie es durch den Namen des Decoder-Manifests, das Sie löschen möchten.

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

## Ruft Informationen zum Decoder-Manifest ab ()AWS CLI

Sie können den [ListDecoderManifests](#) API-Vorgang verwenden, um zu überprüfen, ob ein Decoder-Manifest gelöscht wurde. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Decoder-Manifeste abzurufen.

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

Sie können den [ListDecoderManifestSignals](#) API-Vorgang verwenden, um zu überprüfen, ob die Decodersignale im Decoder-Manifest aktualisiert wurden. Das folgende Beispiel verwendet AWS CLI

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Decodersignale (Knoten) in einem bestimmten Decoder-Manifest abzurufen.

Ersetzen Sie es *decoder-manifest-name* durch den Namen des Decoder-Manifests, das Sie überprüfen.

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

Sie können den [ListDecoderManifestNetworkInterfaces](#) API-Vorgang verwenden, um zu überprüfen, ob die Netzwerkschnittstellen im Decoder-Manifest aktualisiert wurden. Im folgenden Beispiel wird verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Netzwerkschnittstellen in einem bestimmten Decoder-Manifest abzurufen.

*decoder-manifest-name* Ersetzen Sie es durch den Namen des Decoder-Manifests, das Sie überprüfen.

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

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

Sie können den [GetDecoderManifest](#) API-Vorgang verwenden, um zu überprüfen, ob Netzwerkschnittstellen und Decodersignale im Decoder-Manifest aktualisiert wurden. Das folgende Beispiel verwendet AWS CLI

Führen Sie den folgenden Befehl aus, um Informationen über ein Decoder-Manifest abzurufen.

Ersetzen Sie das *Decoder-Manifest* durch den Namen des Decoder-Manifests, das Sie abrufen möchten.

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

#### Note

Diese Operation ist [letztlich konsistent](#). Mit anderen Worten, Änderungen am Decoder-Manifest werden möglicherweise nicht sofort übernommen.

# Fahrzeuge erstellen, bereitstellen und verwalten

Fahrzeuge sind Beispiele für Fahrzeugmodelle. Fahrzeuge müssen anhand eines Fahrzeugmodells erstellt und einem Decoder-Manifest zugeordnet werden. Vehicles lädt einen oder mehrere Datenströme in die Cloud hoch. Beispielsweise kann ein Fahrzeug Daten zu Kilometerstand, Motortemperatur und Zustand der Heizung an die Cloud senden. Jedes Fahrzeug enthält die folgenden Informationen:

## `vehicleName`

Eine ID, die das Fahrzeug identifiziert.

Fügen Sie Ihrem Fahrzeugnamen keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Fahrzeugnamen sind für andere AWS Dienste, einschließlich Amazon, zugänglich CloudWatch. Fahrzeugnamen sind nicht dazu bestimmt, für private oder sensible Daten verwendet zu werden.

## `modelManifestARN`

Der Amazon-Ressourcenname (ARN) eines Fahrzeugmodells (Modellmanifest). Jedes Fahrzeug wird aus einem Fahrzeugmodell erstellt. Fahrzeuge, die mit demselben Fahrzeugmodell erstellt wurden, bestehen aus derselben Gruppe von Signalen, die vom Fahrzeugmodell übernommen wurden. Diese Signale sind im Signalkatalog definiert und standardisiert.

## `decoderManifestArn`

Der ARN des Decoder-Manifests. Ein Decoder-Manifest bietet Dekodierungsinformationen, die das AWS IoT verwenden FleetWise kann, um Rohsignaldaten (Binärdaten) in für Menschen lesbare Werte umzuwandeln. Ein Decoder-Manifest muss einem Fahrzeugmodell zugeordnet sein. AWS IoT FleetWise verwendet dasselbe Decoder-Manifest, um Rohdaten von Fahrzeugen zu dekodieren, die auf der Grundlage desselben Fahrzeugmodells erstellt wurden.

## `attributes`

Attribute sind Schlüssel-Wert-Paare, die statische Informationen enthalten. Fahrzeuge können Attribute enthalten, die vom Fahrzeugmodell übernommen wurden. Sie können zusätzliche Attribute hinzufügen, um ein einzelnes Fahrzeug von anderen Fahrzeugen zu unterscheiden, die mit demselben Fahrzeugmodell erstellt wurden. Wenn Sie beispielsweise ein schwarzes Auto haben, können Sie den folgenden Wert für ein Attribut angeben: `{"color": "black"}`.

**⚠ Important**

Attribute müssen im zugehörigen Fahrzeugmodell definiert werden, bevor Sie sie einzelnen Fahrzeugen hinzufügen können.

Weitere Informationen zu Fahrzeugmodellen, Decoder-Manifesten und Attributen finden Sie unter [Fahrzeuge modellieren](#).

AWS IoT FleetWise bietet die folgenden API-Operationen, mit denen Sie Fahrzeuge erstellen und verwalten können.

- [CreateVehicle](#)— Erstellt ein neues Fahrzeug.
- [BatchCreateVehicle](#)— Erzeugt ein oder mehrere neue Fahrzeuge.
- [UpdateVehicle](#)— Aktualisiert ein vorhandenes Fahrzeug.
- [BatchUpdateVehicle](#)— Aktualisiert ein oder mehrere bestehende Fahrzeuge.
- [DeleteVehicle](#)— Löscht ein vorhandenes Fahrzeug.
- [ListVehicles](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Fahrzeuge ab.
- [GetVehicle](#)— Ruft Informationen über ein Fahrzeug ab.

## Tutorials

- [Fahrzeuge bereitstellen](#)
- [Reservierte Themen](#)
- [Erstelle ein Fahrzeug](#)
- [Ein Fahrzeug aktualisieren \(AWS CLI\)](#)
- [Aktualisieren Sie mehrere Fahrzeuge \(AWS CLI\)](#)
- [Ein Fahrzeug löschen](#)
- [Fahrzeuginformationen abrufen \(AWS CLI\)](#)

## Fahrzeuge bereitstellen

Die Edge Agent for AWS FleetWise IoT-Software, die in Ihrem Fahrzeug ausgeführt wird, sammelt und überträgt Daten in die Cloud. AWS IoT FleetWise lässt sich integrieren AWS IoT Core , um

die sichere Kommunikation zwischen der Edge Agent-Software und der Cloud über MQTT zu unterstützen. Jedes Fahrzeug entspricht einer AWS IoT Sache. Sie können ein vorhandenes Objekt verwenden AWS IoT , um ein Fahrzeug zu erstellen, oder AWS IoT so einstellen FleetWise , dass automatisch ein Objekt AWS IoT für Ihr Fahrzeug erstellt wird. Weitere Informationen finden Sie unter [Erstellen Sie ein Fahrzeug \(AWS CLI\)](#).

AWS IoT Core unterstützt [Authentifizierung](#) und [Autorisierung](#), die helfen, den Zugriff auf AWS FleetWise IoT-Ressourcen sicher zu kontrollieren. Fahrzeuge können X.509-Zertifikate verwenden, um sich zu authentifizieren (anzumelden), um AWS IoT zu nutzen, FleetWise und AWS IoT Core Richtlinien, um autorisiert zu werden (über Berechtigungen zu verfügen), um bestimmte Aktionen auszuführen.

## Fahrzeuge authentifizieren

Sie können AWS IoT Core Richtlinien zur Authentifizierung Ihrer Fahrzeuge erstellen.

Um Ihr Fahrzeug zu authentifizieren

- Führen Sie den folgenden Befehl aus, um eine AWS IoT Core Richtlinie zu erstellen.
  - Ersetzen Sie den *Richtliniennamen* durch den Namen der Richtlinie, die Sie erstellen möchten.
  - Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Richtlinie enthält. AWS IoT Core

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

Bevor Sie die Beispielrichtlinie verwenden, gehen Sie wie folgt vor:

- Ersetzen Sie *Region* durch die AWS Region, in der Sie AWS FleetWise IoT-Ressourcen erstellt haben.
- Ersetzen Sie *AWSAccount* durch Ihre AWS Konto-ID.

Dieses Beispiel beinhaltet Themen, die dem AWS Internet der Dinge vorbehalten sind FleetWise. Sie müssen die Themen zur Richtlinie hinzufügen. Weitere Informationen finden Sie unter [Reservierte Themen](#).

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

## Fahrzeuge autorisieren

Sie können X.509-Zertifikate erstellen, um Ihre Fahrzeuge zu autorisieren.

Um Ihr Fahrzeug zu autorisieren

### Important

Wir empfehlen Ihnen, für jedes Fahrzeug ein neues Zertifikat zu erstellen.

1. Führen Sie den folgenden Befehl aus, um ein RSA-Schlüsselpaar zu erstellen und ein X.509-Zertifikat auszustellen.
  - Ersetzen Sie *cert* durch den Namen der Datei, die den Inhalt der Befehlsausgabe von CertificatePEM speichert.
  - Ersetzen Sie *public-key* durch den Namen der Datei, die den Inhalt der Befehlsausgabe von KeyPair speichert. PublicKey.
  - Ersetzen Sie *private-key* durch den Namen der Datei, die den Inhalt der Befehlsausgabe von KeyPair speichert. 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. Kopieren Sie den Amazon-Ressourcennamen (ARN) des Zertifikats aus der Ausgabe.
3. Führen Sie den folgenden Befehl aus, um die Richtlinie an das Zertifikat anzuhängen.

- Ersetzen Sie den *Richtliniennamen* durch den Namen der AWS IoT Core Richtlinie, die Sie erstellt haben.
- Ersetzen Sie *certificate-arn* durch den ARN des Zertifikats, das Sie kopiert haben.

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

4. Führen Sie den folgenden Befehl aus, um das Zertifikat an das Ding anzuhängen.
  - Ersetzen Sie *thing-name* durch den Namen Ihres AWS IoT Dings oder die ID Ihres Fahrzeugs.
  - Ersetzen Sie *certificate-arn* durch den ARN des Zertifikats, das Sie kopiert haben.

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

## Reservierte Themen

AWS IoT FleetWise behält sich die Verwendung der folgenden Themen vor. Wenn das reservierte Thema es zulässt, können Sie es abonnieren oder veröffentlichen. Sie können jedoch keine neuen Themen erstellen, die mit einem Dollarzeichen (\$) beginnen. Wenn Sie nicht unterstützte Veröffentlichungs- oder Abonnementvorgänge mit reservierten Themen verwenden, kann dies dazu führen, dass die Verbindung beendet wird.

Thema	Client-Betrieb zulässig	Beschreibung
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> /checkins	Veröffentlichen	Die Edge Agent-Software veröffentlicht Fahrzeugstatusinformationen zu diesem Thema.

Thema	Client-Betrieb zulässig	Beschreibung	
		<p>Fahrzeugstatusinformationen werden im Protokollpufferformat (Protobuf) ausgetauscht. Weitere Informationen finden Sie im <a href="#">Edge Agent for AWS FleetWise IoT-Softwareentwicklerhandbuch</a>.</p>	
<pre>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /signals</pre>	Veröffentlichen	<p>Die Edge Agent-Software veröffentlicht Signale zu diesem Thema.</p> <p>Signalinformationen werden im Protokollpufferformat (Protobuf) ausgetauscht. Weitere Informationen finden Sie im <a href="#">Edge Agent for AWS FleetWise IoT-Softwareentwicklerhandbuch</a>.</p>	
<pre>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /collection_schemes</pre>	Abonnieren	<p>AWS IoT FleetWise veröffentlicht Datenerfassungsschemata zu diesem Thema. Fahrzeuge nutzen diese Datenerfassungsschemata.</p>	

Thema	Client-Betrieb zulässig	Beschreibung
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i>/decoder_manifests</code>	Abonnieren	AWS IoT FleetWise veröffentlicht Decoder-Manifeste zu diesem Thema. Fahrzeuge verbrauchen diese Decoder-Manifeste.

## Erstelle ein Fahrzeug

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um ein Fahrzeug zu erstellen.

### Wichtig

Bevor Sie beginnen, überprüfen Sie Folgendes:

- Sie müssen über ein Fahrzeugmodell verfügen und der Status des Fahrzeugmodells muss lauten ACTIVE. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).
- Ihr Fahrzeugmodell muss mit einem Decoder-Manifest verknüpft sein, und der Status des Decoder-Manifests muss lauten ACTIVE. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

### Themen

- [Erstellen Sie ein Fahrzeug \(Konsole\)](#)
- [Erstellen Sie ein Fahrzeug \(AWS CLI\)](#)
- [Erstellen Sie mehrere Fahrzeuge \(AWS CLI\)](#)

## Erstellen Sie ein Fahrzeug (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeug zu erstellen.

**⚠ Important**

Bevor Sie beginnen, überprüfen Sie Folgendes:

- Sie müssen über ein Fahrzeugmodell verfügen und der Status des Fahrzeugmodells muss lauten ACTIVE. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).
- Ihr Fahrzeugmodell muss mit einem Decoder-Manifest verknüpft sein, und der Status des Decoder-Manifests muss lauten ACTIVE. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

Um ein Fahrzeug zu erstellen

1. Öffnen Sie die [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeuge aus.
3. Wählen Sie auf der Seite mit der Fahrzeugübersicht die Option Fahrzeug erstellen aus, und führen Sie dann die folgenden Schritte aus.

Themen

- [Schritt 1: Definieren Sie die Fahrzeugeigenschaften](#)
- [Schritt 2: Fahrzeugzertifikat konfigurieren](#)
- [Schritt 3: Richtlinien an das Zertifikat anhängen](#)
- [Schritt 4: Überprüfen und Erstellen](#)

## Schritt 1: Definieren Sie die Fahrzeugeigenschaften

In diesem Schritt benennen Sie das Fahrzeug und verknüpfen es mit dem Modellmanifest und dem Decoder-Manifest.

1. Geben Sie einen eindeutigen Namen für das Fahrzeug ein.

**⚠ Important**

Ein Fahrzeug entspricht einer AWS IoT Sache. Wenn bereits ein Objekt mit diesem Namen existiert, wählen Sie Fahrzeug mit einem IoT-Ding verknüpfen, um das Ding mit

dem Fahrzeug zu aktualisieren. Oder wählen Sie einen anderen Fahrzeugnamen und AWS IoT erstellt FleetWise automatisch eine neue Sache für das Fahrzeug.

2. Wählen Sie ein Fahrzeugmodell (Modellmanifest) aus der Liste aus.
3. Wählen Sie ein Decoder-Manifest aus der Liste aus. Das Decoder-Manifest ist dem Fahrzeugmodell zugeordnet.
4. (Optional) Um Fahrzeugattribute zuzuordnen, wählen Sie Attribute hinzufügen. Wenn Sie diesen Schritt überspringen, müssen Sie nach der Erstellung des Fahrzeugs Attribute hinzufügen, bevor Sie es für Kampagnen einsetzen können.
5. (Optional) Um Tags mit dem Fahrzeug zu verknüpfen, wählen Sie Neues Tag hinzufügen. Sie können Tags auch hinzufügen, nachdem das Fahrzeug erstellt wurde.
6. Wählen Sie Weiter aus.

## Schritt 2: Fahrzeugzertifikat konfigurieren

Um Ihr Fahrzeug als Objekt verwenden zu AWS IoT können, müssen Sie ein Fahrzeugzertifikat mit einer angehängten Richtlinie konfigurieren. Wenn Sie diesen Schritt überspringen, müssen Sie nach der Erstellung des Fahrzeugs ein Zertifikat konfigurieren, bevor Sie es für Kampagnen einsetzen können.

1. Wählen Sie Neues Zertifikat automatisch generieren (empfohlen).
2. Wählen Sie Weiter aus.

## Schritt 3: Richtlinien an das Zertifikat anhängen

Hängen Sie eine Richtlinie an das Zertifikat an, das Sie im vorherigen Schritt konfiguriert haben.

1. Geben Sie unter Richtlinien einen vorhandenen Richtliniennamen ein. Um eine neue Richtlinie zu erstellen, wählen Sie Richtlinie erstellen aus.
2. Wählen Sie Weiter aus.

## Schritt 4: Überprüfen und Erstellen

Überprüfen Sie die Konfigurationen für das Fahrzeug und wählen Sie dann Fahrzeug erstellen aus.

**⚠ Important**

Nachdem das Fahrzeug erstellt wurde, müssen Sie das Zertifikat und die Schlüssel herunterladen. Sie verwenden das Zertifikat und den privaten Schlüssel, um das Fahrzeug in der Edge Agent for AWS FleetWise IoT-Software zu verbinden.

## Erstellen Sie ein Fahrzeug (AWS CLI)

Wenn Sie ein Fahrzeug erstellen, müssen Sie ein Fahrzeugmodell verwenden, das einem Decoder-Manifest zugeordnet ist. Sie können den [CreateVehicle](#) API-Vorgang verwenden, um ein Fahrzeug zu erstellen. Das folgende Beispiel verwendet die AWS CLI.

**⚠ Important**

Bevor Sie beginnen, überprüfen Sie Folgendes:

- Sie müssen über ein Fahrzeugmodell verfügen und der Status des Fahrzeugmodells muss lauten ACTIVE. Weitere Informationen finden Sie unter [Fahrzeugmodelle erstellen und verwalten](#).
- Ihr Fahrzeugmodell muss mit einem Decoder-Manifest verknüpft sein, und der Status des Decoder-Manifests muss lauten ACTIVE. Weitere Informationen finden Sie unter [Decoder-Manifeste erstellen und verwalten](#).

Führen Sie den folgenden Befehl aus, um ein Fahrzeug zu erstellen.

Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Fahrzeugkonfiguration enthält.

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

### Example Fahrzeugkonfiguration

- (Optional) Der `associationBehavior` Wert kann einer der folgenden sein:
  - `CreateIotThing`— Wenn Ihr Fahrzeug erstellt wird, erstellt AWS IoT FleetWise automatisch eine AWS IoT Sache mit dem Namen Ihrer Fahrzeug-ID für Ihr Fahrzeug.
  - `ValidateIotThingExists`— Verwenden Sie ein vorhandenes AWS IoT Objekt, um ein Fahrzeug zu erstellen.

Um ein AWS IoT Ding zu erstellen, führen Sie den folgenden Befehl aus. Ersetzen Sie *thing-name* durch den Namen des Dings, das Sie erstellen möchten.

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

Wenn es nicht spezifiziert ist, erstellt AWS IoT FleetWise automatisch AWS IoT etwas für Ihr Fahrzeug.

**⚠ Important**

Stellen Sie sicher, dass das AWS IoT Ding nach der Erstellung des Fahrzeugs bereitgestellt wird. Weitere Informationen finden Sie unter [Fahrzeuge bereitstellen](#).

- Ersetzen Sie den *Fahrzeugnamen* durch einen der folgenden Werte.
  - Der Name Ihres Dings AWS IoT , falls er so konfiguriert `associationBehavior` ist `ValidateIotThingExists`
  - Die ID des Fahrzeugs, für das es erstellt werden soll, `associationBehavior` ist `konfiguriertCreateIotThing`.

Die Fahrzeug-ID kann 1—100 Zeichen lang sein. Gültige Zeichen: a—z, A—Z, 0—9, Bindestrich (-), Unterstrich (\_) und Doppelpunkt (:).

- Ersetzen Sie *Model-Manifest-ARN* durch den ARN Ihres Fahrzeugmodells (Modellmanifest).
- Ersetzen Sie *Decoder-Manifest-ARN* durch den ARN des Decoder-Manifests, das dem angegebenen Fahrzeugmodell zugeordnet ist.
- (Optional) Sie können zusätzliche Attribute hinzufügen, um dieses Fahrzeug von anderen Fahrzeugen zu unterscheiden, die mit demselben Fahrzeugmodell hergestellt wurden. Wenn Sie beispielsweise ein Elektroauto haben, können Sie den folgenden Wert für ein Attribut angeben: `{"fuelType": "electric"}`.

**⚠ Important**

Attribute müssen im zugehörigen Fahrzeugmodell definiert werden, bevor Sie sie einzelnen Fahrzeugen hinzufügen können.

```
{
```

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

## Erstellen Sie mehrere Fahrzeuge (AWS CLI)

Sie können den [BatchCreateVehicle](#) API-Vorgang verwenden, um mehrere Fahrzeuge gleichzeitig zu erstellen. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um mehrere Fahrzeuge zu erstellen.

Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Konfigurationen mehrerer Fahrzeuge enthält.

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

### Example Fahrzeugkonfigurationen

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

```
    }
  }
]
}
```

Sie können bis zu 10 Fahrzeuge für jeden Batch-Vorgang erstellen. Weitere Informationen zur Fahrzeugkonfiguration finden Sie unter [Erstellen Sie ein Fahrzeug \(AWS CLI\)](#).

## Ein Fahrzeug aktualisieren (AWS CLI)

Sie können den [UpdateVehicle](#) API-Vorgang verwenden, um ein vorhandenes Fahrzeug zu aktualisieren. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um ein Fahrzeug zu aktualisieren.

Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Konfiguration Ihres Fahrzeugs enthält.

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

### Example Fahrzeugkonfiguration

- Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs, das Sie aktualisieren möchten.
- (Optional) Ersetzen Sie *Model-Manifest-ARN* durch den ARN des Fahrzeugmodells (Modellmanifest), mit dem Sie das verwendete Fahrzeugmodell ersetzen.
- (Optional) Ersetzen Sie *Decoder-Manifest-ARN durch den ARN* Ihres Decoder-Manifests, das dem von Ihnen angegebenen neuen Fahrzeugmodell zugeordnet ist.
- (Optional) Ersetzen Sie es durch Fahrzeugattribute. *attribute-update-mode*
  - Merge— Führen Sie neue Attribute mit vorhandenen Attributen zusammen, indem Sie bestehende Attribute mit neuen Werten aktualisieren und neue Attribute hinzufügen, falls sie nicht existieren.

Wenn ein Fahrzeug beispielsweise die folgenden Attribute hat: {"color": "black", "fuelType": "electric"}, und Sie aktualisieren das Fahrzeug mit den folgenden Attributen: {"color": "", "fuelType": "gasoline", "model": "x"}, hat das aktualisierte Fahrzeug die folgenden Attribute: {"fuelType": "gasoline", "model": "x"}.

- **Overwrite**— Ersetzt bestehende Attribute durch neue Attribute.

Wenn ein Fahrzeug beispielsweise die folgenden Attribute hat:{"color": "black", "fuelType": "electric"}, und Sie das Fahrzeug mit dem {"model": "x"} Attribut aktualisieren, hat das aktualisierte Fahrzeug das {"model": "x"} Attribut.

Dies ist erforderlich, wenn Attribute in der Eingabe vorhanden sind.

- (Optional) Um neue Attribute hinzuzufügen oder bestehende mit neuen Werten zu aktualisieren, konfigurieren Sie `attributes`. Wenn Sie beispielsweise ein Elektroauto haben, können Sie den folgenden Wert für ein Attribut angeben:{"fuelType": "electric"}.

Um Attribute zu löschen, konfigurieren Sie `attributeUpdateMode` auf `merge`.

### Important

Attribute müssen im zugehörigen Fahrzeugmodell definiert werden, bevor Sie sie einzelnen Fahrzeugen hinzufügen können.

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

## Aktualisieren Sie mehrere Fahrzeuge (AWS CLI)

Sie können den [BatchUpdateVehicle](#) API-Vorgang verwenden, um mehrere vorhandene Fahrzeuge gleichzeitig zu aktualisieren. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um mehrere Fahrzeuge zu aktualisieren.

Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Konfigurationen mehrerer Fahrzeuge enthält.

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

## Example Fahrzeugkonfigurationen

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

Sie können bis zu 10 Fahrzeuge für jeden Batch-Vorgang aktualisieren. Weitere Informationen zur Konfiguration der einzelnen Fahrzeuge finden Sie unter [Ein Fahrzeug aktualisieren \(AWS CLI\)](#).

## Ein Fahrzeug löschen

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Fahrzeuge zu löschen.

### Important

Nachdem ein Fahrzeug gelöscht wurde, entfernt AWS IoT das Fahrzeug FleetWise automatisch aus den zugehörigen Flotten und Kampagnen. Weitere Informationen finden Sie unter [Flotten erstellen und verwalten](#) und [Sammeln und übertragen Sie Daten mit Kampagnen](#). Das Fahrzeug existiert jedoch immer noch als Ding oder ist immer noch mit einem Ding in AWS IoT Core Verbindung gebracht. Anweisungen zum Löschen eines Objekts finden Sie unter [Löschen eines Objekts](#) im AWS IoT Core Entwicklerhandbuch.

## Lösche ein Fahrzeug (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um ein Fahrzeug zu löschen.

Um ein Fahrzeug zu löschen

1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Fahrzeuge aus.
3. Wählen Sie auf der Seite Fahrzeuge die Schaltfläche neben dem Fahrzeug aus, das Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Geben Sie **vehicle-name** unter Löschen den Namen des Fahrzeugs ein und wählen Sie dann Löschen.

## Lösche ein Fahrzeug (AWS CLI)

Sie können den [DeleteVehicle](#)API-Vorgang verwenden, um ein Fahrzeug zu löschen. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um ein Fahrzeug zu löschen.

Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs, das Sie löschen möchten.

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

## Fahrzeuginformationen abrufen ()AWS CLI

Sie können den [ListVehicles](#)API-Vorgang verwenden, um zu überprüfen, ob ein Fahrzeug gelöscht wurde. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen aller Fahrzeuge abzurufen.

```
aws iotfleetwise list-vehicles
```

Sie können den [GetVehicle](#)API-Vorgang verwenden, um Fahrzeuginformationen abzurufen. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um die Metadaten eines Fahrzeugs abzurufen.

Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs, das Sie abrufen möchten.

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

 Note

Diese Operation ist letztlich konsistent. Mit anderen Worten, Änderungen am Fahrzeug werden möglicherweise nicht sofort übernommen.

# Flotten erstellen und verwalten

Eine Flotte stellt eine Gruppe von Fahrzeugen dar. Eine Flotte ohne zugehörige Fahrzeuge ist eine leere Einheit. Bevor Sie die Flotte verwenden können, um mehrere Fahrzeuge gleichzeitig zu verwalten, müssen Sie Fahrzeuge der Flotte zuordnen. Ein Fahrzeug darf mehreren Flotten. Sie können steuern, welche Daten von einer Fahrzeugflotte erfasst werden sollen und wann Daten gesammelt werden sollen, indem Sie eine Kampagne einsetzen. Weitere Informationen finden Sie unter [Sammeln und übertragen Sie Daten mit Kampagnen](#).

Eine Flotte enthält die folgenden Informationen.

`fleetId`

Die ID der Flotte.

(Optional) `description`

Eine Beschreibung, die Ihnen hilft, die Flotte zu finden.

`signalCatalogArn`

Der Amazon-Ressourcenname (ARN) des Signalkatalogs.

AWS IoT FleetWise bietet die folgenden API-Operationen, mit denen Sie Flotten erstellen und verwalten können.

- [CreateFleet](#)— Erzeugt eine Gruppe von Fahrzeugen, die dieselbe Gruppe von Signalen enthalten.
- [AssociateVehicleFleet](#)— Ordnet ein Fahrzeug einer Flotte zu.
- [DisassociateVehicleFleet](#)— Trennt ein Fahrzeug von einer Flotte.
- [UpdateFleet](#)— Aktualisiert die Beschreibung für eine bestehende Flotte.
- [DeleteFleet](#)— Löscht eine bestehende Flotte.
- [ListFleets](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Flotten ab.
- [ListFleetsForVehicle](#)— Ruft eine paginierte Liste mit IDs aller Flotten ab, zu denen das Fahrzeug gehört.
- [ListVehiclesInFleet](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Fahrzeuge in einer Flotte ab.
- [GetFleet](#)— Ruft Informationen über eine Flotte ab.

## Themen

- [Erstelle eine Flotte \(AWS CLI\)](#)
- [Ordnen Sie ein Fahrzeug einer Flotte zu \(AWS CLI\)](#)
- [Ein Fahrzeug von einer Flotte trennen \(AWS CLI\)](#)
- [Aktualisieren Sie eine Flotte \(AWS CLI\)](#)
- [Löschen Sie eine Flotte \(AWS CLI\)](#)
- [Flotteninformationen abrufen \(AWS CLI\)](#)

## Erstelle eine Flotte (AWS CLI)

Sie können den [CreateFleet](#)API-Vorgang verwenden, um eine Fahrzeugflotte zu erstellen. Im folgenden Beispiel wird verwendet AWS CLI.

### Important

Sie müssen über einen Signalkatalog verfügen, bevor Sie über einen Signalkatalog verfügen, bevor Sie eine Flotte erstellen. Weitere Informationen finden Sie unter [Erstellen AWS CLI Sie einen Signalkatalog \(\)](#).

Führen den folgenden Befehl aus, um eine Flotte zu erstellen, führen den folgenden Befehl aus, um eine Flotte

- Ersetzen Sie *fleet-id* durch die ID der Flotte, die Sie erstellen.

Die Flotten-ID muss eindeutig sein und 1-100 Zeichen lang sein. Gültige Zeichen: Buchstaben (A—Z) und a—z), Zahlen (0—9), Doppelpunkte (:), Bindestriche (-) und Unterstriche (\_).

- (Optional) Ersetzen Sie die *Beschreibung* durch eine Beschreibung.

Die Beschreibung kann 1-2048 Zeichen lang sein.

- *signal-catalog-arn* Ersetzen Sie den ARN des Signalkatalogs.

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

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

## Ordnen Sie ein Fahrzeug einer Flotte zu (AWS CLI)

Sie können den [AssociateVehicleFleet](#) API-Vorgang verwenden, um ein Fahrzeug einer Flotte zuzuordnen. Im folgenden Beispiel wird verwendet AWS CLI.

### Wichtig

- Sie müssen über ein Fahrzeug und eine Flotte verfügen, bevor Sie ein Fahrzeug einer Flotte zuordnen können. Weitere Informationen finden Sie unter [Fahrzeuge erstellen, bereitstellen und verwalten](#).
- Wenn Sie ein Fahrzeug einer Flotte zuordnen, auf die eine Kampagne abzielt, setzt AWS IoT die Kampagne FleetWise automatisch für das Fahrzeug ein.

Um ein Fahrzeug mit einer Flotte führen den folgenden Befehl aus, um ein Fahrzeug mit einer Flotte zu verknüpfen.

- Ersetzen Sie *fleet-id* durch die ID der Flotte.
- Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs.

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

## Ein Fahrzeug von einer Flotte trennen (AWS CLI)

Sie können den [DisassociateVehicleFleet](#) API-Vorgang verwenden, um ein Fahrzeug von einer Flotte zu trennen. Im folgenden Beispiel wird verwendet AWS CLI.

Führen den folgenden Befehl aus, um die Verknüpfung eines Fahrzeugs mit einer Flotte zu trennen, führen den folgenden Befehl aus, um die Verbindung zu einer Flotte

- Ersetzen Sie *fleet-id* durch die ID der Flotte.
- Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs.

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

## Aktualisieren Sie eine Flotte (AWS CLI)

Sie können den [UpdateFleet](#)API-Vorgang verwenden, um die Beschreibung für eine Flotte zu aktualisieren. Im folgenden Beispiel wird verwendet AWS CLI.

Führen den folgenden Befehl zur Aktualisierung einer Flotte den folgenden Befehl aus, um eine Flotte zu aktualisieren.

- Ersetzen Sie *fleet-id* durch die ID der Flotte, die Sie aktualisieren.
- Ersetzen Sie die *Beschreibung* durch eine neue Beschreibung.

Die Beschreibung kann 1-2048 Zeichen lang sein.

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

## Löschen Sie eine Flotte (AWS CLI)

Sie können den [DeleteFleet](#)API-Vorgang verwenden, um eine Flotte zu löschen. Im folgenden Beispiel wird verwendet AWS CLI.

### Important

Bevor Sie eine Flotte löschen, stellen Sie sicher, dass ihr keine Fahrzeuge zugeordnet sind. Anweisungen zum Trennen eines Fahrzeugs von einer Flotte finden Sie unter [Ein Fahrzeug von einer Flotte trennen \(AWS CLI\)](#).

Um eine Flotte zu löschen, führen den folgenden Befehl aus, um eine Flotte zu löschen.

Ersetzen Sie *fleet-id* durch die ID der Flotte, die Sie löschen möchten.

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

## Flotteninformationen abrufen (AWS CLI)

Sie können den [ListFleets](#)API-Vorgang verwenden, um zu überprüfen, ob eine Flotte gelöscht wurde. Das folgende Beispiel verwendet die AWS CLI.

Um eine paginierte Liste mit Zusammenfassungen aller Flotten führen den folgenden Befehl aus, um eine paginierte Liste von Zusammenfassungen aller Flotten abzurufen.

```
aws iotfleetwise list-fleets
```

Sie können den [ListFleetsForVehicle](#)API-Vorgang verwenden, um eine paginierte Liste von IDs aller Flotten abzurufen, zu denen das Fahrzeug gehört. Das folgende Beispiel verwendet die AWS CLI.

Um eine paginierte Liste von IDs abzurufen, die allen Flotten, führen den folgenden Befehl aus, um eine paginierte Liste der IDs abzurufen, zu denen das Fahrzeug den folgenden Befehl aus.

Ersetzen Sie den *Fahrzeugnamen* durch die ID des Fahrzeugs.

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

Sie können den [ListVehiclesInFleet](#)API-Vorgang verwenden, um eine paginierte Liste mit Zusammenfassungen aller Fahrzeuge in einer Flotte abzurufen. Das folgende Beispiel verwendet die AWS CLI.

Um eine paginierte Liste mit Zusammenfassungen aller Fahrzeuge in einer Flotte führen den folgenden Befehl aus, um eine paginierte Liste von Zusammenfassungen aller Fahrzeuge in einer Flotte abzurufen.

Ersetzen Sie *fleet-id* durch die ID der Flotte.

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

Sie können den [GetFleet](#)API-Vorgang verwenden, um Flotteninformationen abzurufen. Das folgende Beispiel verwendet die AWS CLI.

Um die Metadaten einer Flotte abzurufen, führen den folgenden Befehl aus, um die Metadaten abzurufen.

Ersetzen Sie *fleet-id* durch die ID der Flotte.

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

 Note

Diese Operation ist [letztlich konsistent](#). Mit anderen Worten, Änderungen an der Flotte wirken sich möglicherweise nicht sofort aus.

# Sammeln und übertragen Sie Daten mit Kampagnen

Eine Kampagne ist eine Orchestrierung von Regeln für die Datenerfassung. Kampagnen geben der Edge Agent for AWS FleetWise IoT-Software Anweisungen zur Auswahl, Erfassung und Übertragung von Daten in die Cloud.

Sie erstellen Kampagnen in der Cloud. Nachdem Sie oder Ihr Team eine Kampagne genehmigt haben, stellt AWS IoT sie FleetWise automatisch in Fahrzeugen bereit. Sie können wählen, ob Sie eine Kampagne für ein Fahrzeug oder eine Fahrzeugflotte einsetzen möchten. Die Edge Agent-Software beginnt erst mit der Erfassung von Daten, wenn eine laufende Kampagne für das Fahrzeug bereitgestellt wird.

## Note

Kampagnen funktionieren erst, wenn Sie über Folgendes verfügen.

- Die Edge Agent-Software wird in Ihrem Fahrzeug ausgeführt. Gehen Sie wie folgt vor, um weitere Informationen zur Entwicklung, Installation und Arbeit mit der Edge Agent-Software zu erhalten.
  1. Navigieren Sie zur [AWS FleetWiseIoT-Konsole](#).
  2. Wählen Sie auf der Service-Startseite im FleetWise Abschnitt Erste Schritte mit AWS IoT die Option Explore Edge Agent aus.
- Sie haben die Einrichtung AWS IoT Core für die Bereitstellung Ihres Fahrzeugs eingerichtet. Weitere Informationen finden Sie unter [Fahrzeuge bereitstellen](#).

Jede Kampagne enthält die folgenden Informationen.

`signalCatalogArn`

Der Amazon-Ressourcenname (ARN) des Signalkatalogs, der mit der Kampagne verknüpft ist.

(Optional) `tags`

Tags sind Metadaten, die zur Verwaltung der Kampagne verwendet werden können. Sie können Ressourcen aus verschiedenen Diensten dasselbe Tag zuweisen, um anzuzeigen, dass die Ressourcen miteinander verknüpft sind.

## TargetArn

Der ARN eines Fahrzeugs oder einer Flotte, in der die Kampagne bereitgestellt wird.

## name

Ein eindeutiger Name, anhand dessen die Kampagne leichter identifiziert werden kann.

## collectionScheme

Die Datenerfassungsschemata geben der Edge Agent-Software Anweisungen darüber, welche Daten gesammelt werden sollen oder wann sie gesammelt werden sollen. AWS IoT unterstützt FleetWise derzeit das zustandsbasierte Sammelschema und das zeitbasierte Sammelschema.

## conditionBasedCollectionScheme

Das zustandsabhängige Erfassungsschema verwendet einen logischen Ausdruck, um zu erkennen, welche Daten gesammelt werden sollen. Die Edge Agent-Software sammelt Daten, wenn die Bedingung erfüllt ist.

## expression

Der logische Ausdruck, mit dem erkannt wird, welche Daten gesammelt werden sollen. Wenn der `$variable.`myVehicle.InVehicleTemperature` > 50.0` Ausdruck beispielsweise angegeben ist, erfasst die Edge Agent-Software Temperaturwerte, die über 50,0 liegen. Anweisungen zum Schreiben von Ausdrücken finden Sie unter [Logische Ausdrücke für Kampagnen](#).

(Optional) `triggerMode` kann einer der folgenden Werte sein.

- `RISING_EDGE`— Die Edge Agent-Software sammelt Daten nur, wenn die Bedingung zum ersten Mal erfüllt ist. Zum Beispiel `$variable.`myVehicle.AirBagDeployed` == true`.
- `ALWAYS`— Die Edge Agent-Software sammelt Daten, wenn die Bedingung erfüllt ist.

(Optional) `minimumTriggerIntervalMs`

Die Mindestdauer zwischen zwei Datenerfassungsereignissen in Millisekunden. Wenn sich ein Signal häufig ändert, erfassen Sie Daten möglicherweise langsamer.

(Optional) `conditionLanguageVersion`

Die Version der Sprache für bedingte Ausdrücke.

## timeBasedCollectionScheme

Wenn Sie ein zeitbasiertes Erfassungsschema definieren, geben Sie einen Zeitraum in Millisekunden an. Die Edge Agent-Software entscheidet anhand des Zeitraums, wie oft Daten

erfasst werden sollen. Wenn der Zeitraum beispielsweise 120.000 Millisekunden beträgt, erfasst die Edge Agent-Software alle zwei Minuten Daten.

(Optional) `compression`

Um drahtlose Bandbreite zu sparen und den Netzwerkverkehr zu reduzieren, können Sie [SNAPPY angeben, um Daten in Fahrzeugen zu komprimieren](#).

Standardmäßig (OFF) komprimiert die Edge Agent-Software keine Daten.

`dataDestinationConfigs`

Wählen Sie das Ziel aus, an das die Kampagne Fahrzeugdaten übertragen soll. Sie können wählen, ob Sie Daten in Amazon S3 oder Amazon Timestream speichern möchten.

S3 ist ein kostengünstiger Datenspeichermechanismus, der dauerhafte Datenverwaltungsfunktionen und nachgelagerte Datendienste bietet. Sie können S3 für Daten zum Fahrverhalten oder zur Analyse langfristiger Wartungsarbeiten verwenden.

Timestream ist ein Mechanismus zur Datenpersistenz, mit dem Sie Trends und Muster nahezu in Echtzeit erkennen können. Sie können Timestream für Zeitreihendaten verwenden, z. B. um historische Trends bei der Fahrzeuggeschwindigkeit oder beim Bremsen zu analysieren.

(Optional) `dataExtraDimensions`

Sie können ein oder mehrere Attribute hinzufügen, um zusätzliche Informationen für ein Signal bereitzustellen.

(Optional) `description`

Sie können eine Beschreibung hinzufügen, um den Zweck der Kampagne besser zu identifizieren.

(Optional) `diagnosticsMode`

Wenn der Diagnosemodus so konfiguriert ist `SEND_ACTIVE_DTCS`, sendet die Kampagne gespeicherte Standarddiagnosefehlercodes (DTCs), anhand derer Sie feststellen können, was mit Ihrem Fahrzeug nicht stimmt. Beispielsweise bedeutet P0097, dass das Motorsteuergerät (ECM) festgestellt hat, dass der Eingang des Ansauglufttemperatursensors 2 (IAT2) unter dem normalen Sensorbereich liegt.

Standardmäßig (OFF) sendet die Edge Agent-Software keine Diagnosecodes.

(Optional) `expiryTime`

Sie können das Ablaufdatum für Ihre Kampagne definieren. Wenn die Kampagne abläuft, beendet die Edge Agent-Software die Erfassung von Daten, wie in dieser Kampagne angegeben. Wenn

mehrere Kampagnen für das Fahrzeug bereitgestellt werden, verwendet die Edge Agent-Software andere Kampagnen, um Daten zu sammeln.

Standardwert: 253402243200 (31. Dezember 9999, 00:00:00 UTC)

#### (Optional) `postTriggerCollectionDuration`

Sie können eine Dauer für die Erfassung nach dem Auslösen definieren, sodass die Edge Agent-Software nach dem Aufrufen eines Schemas für einen bestimmten Zeitraum weiterhin Daten sammelt. Wenn beispielsweise ein zustandsbasiertes Erfassungsschema mit dem folgenden Ausdruck aufgerufen wird: `$variable.`myVehicle.Engine.RPM` > 7000.0`, erfasst die Edge Agent-Software weiterhin Werte für Umdrehungen pro Minute (U/min) für den Motor. Selbst wenn die Drehzahl nur einmal höher als 7000 ist, kann dies auf ein mechanisches Problem hinweisen. In diesem Fall möchten Sie möglicherweise, dass die Edge Agent-Software weiterhin Daten sammelt, um den Zustand zu überwachen.

Standardwert: 0

#### (Optional) `priority`

Sie können eine Ganzzahl angeben, um die Prioritätsstufe der Kampagne anzugeben. Kampagnen mit einer kleineren Anzahl haben eine höhere Priorität. Wenn Sie mehrere Kampagnen für ein Fahrzeug bereitstellen, werden die Kampagnen mit höherer Priorität zuerst initiiert.

Standardwert: 0

#### (Optional) `signalsToCollect`

Eine Liste von Signalen, aus denen Daten gesammelt werden, wenn das Datenerfassungsschema aufgerufen wird.

#### Important

In diesem Feld müssen die Signale angegeben werden, die im Ausdruck für das bedingungs-basierte Erfassungsschema verwendet werden.

#### `name`

Der Name des Signals, aus dem Daten gesammelt werden, wenn das Datenerfassungsschema aufgerufen wird.

### (Optional) `maxSampleCount`

Die maximale Anzahl von Datenproben, die die Edge Agent-Software sammelt und in die Cloud überträgt, wenn das Datenerfassungsschema aufgerufen wird.

### (Optional) `minimumSamplingIntervalMs`

Die Mindestdauer zwischen zwei Datenprobenerfassungsereignissen in Millisekunden. Wenn sich ein Signal häufig ändert, können Sie diesen Parameter verwenden, um Daten langsamer zu sammeln.

Gültiger Bereich: 0-4294967295

### (Optional) `spoolingMode`

Wenn dies konfiguriert `spoolingMode` ist `T0_DISK`, speichert die Edge Agent-Software vorübergehend Daten lokal, wenn ein Fahrzeug nicht mit der Cloud verbunden ist. Nachdem die Verbindung wiederhergestellt wurde, werden die lokal gespeicherten Daten automatisch in die Cloud übertragen.

Standardwert: `OFF`

### (Optional) `startTime`

Eine genehmigte Kampagne wird zum Startzeitpunkt aktiviert.

Standardwert: `0`

Der Status einer Kampagne kann einer der folgenden Werte sein.

- `CREATING`— AWS IoT FleetWise bearbeitet Ihre Anfrage zur Erstellung der Kampagne.
- `WAITING_FOR_APPROVAL`— Nachdem eine Kampagne erstellt wurde, geht sie in den `WAITING_FOR_APPROVAL` Status über. Verwenden Sie den `UpdateCampaign` API-Vorgang, um die Kampagne zu genehmigen. Nachdem die Kampagne genehmigt wurde, stellt AWS IoT die Kampagne FleetWise automatisch für das Zielfahrzeug oder die Zielflotte bereit. Weitere Informationen finden Sie unter [Aktualisieren Sie eine Kampagne \(AWS CLI\)](#).
- `RUNNING` — Die Kampagne ist aktiv.
- `SUSPENDED`— Die Kampagne ist ausgesetzt. Verwenden Sie den `UpdateCampaign` API-Vorgang, um die Kampagne fortzusetzen.

AWS IoT FleetWise bietet die folgenden API-Operationen, mit denen Sie Kampagnen erstellen und verwalten können.

- [CreateCampaign](#)— Erstellt eine neue Kampagne.
- [UpdateCampaign](#)— Aktualisiert eine bestehende Kampagne. Nachdem eine Kampagne erstellt wurde, müssen Sie diese API-Operation verwenden, um die Kampagne zu genehmigen.
- [DeleteCampaign](#)— Löscht eine bestehende Kampagne.
- [ListCampaigns](#)— Ruft eine paginierte Liste mit Zusammenfassungen für alle Kampagnen ab.
- [GetCampaign](#)— Ruft Informationen über eine Kampagne ab.

## Tutorials

- [Erstellen einer Kampagne](#)
- [Aktualisieren Sie eine Kampagne \(AWS CLI\)](#)
- [Löscht eine Kampagne](#)
- [Kampagneninformationen abrufen \(\) AWS CLI](#)

## Erstellen einer Kampagne

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Kampagnen zur Erfassung von Fahrzeugdaten zu erstellen.

### Important

Damit Ihre Kampagne funktioniert, müssen Sie über Folgendes verfügen:

- Die Edge Agent-Software wird in Ihrem Fahrzeug ausgeführt. Gehen Sie wie folgt vor, um weitere Informationen zur Entwicklung, Installation und Arbeit mit der Edge Agent-Software zu erhalten:
  1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
  2. Wählen Sie auf der Service-Startseite im FleetWise Abschnitt Erste Schritte mit AWS IoT die Option Explore Edge Agent aus.
- Sie haben die Einrichtung AWS IoT Core für die Bereitstellung Ihres Fahrzeugs eingerichtet. Weitere Informationen finden Sie unter [Fahrzeuge bereitstellen](#).

## Themen

- [Erstellen Sie eine Kampagne \(Konsole\)](#)
- [Erstelle eine Kampagne \(AWS CLI\)](#)
- [Logische Ausdrücke für Kampagnen](#)

## Erstellen Sie eine Kampagne (Konsole)

Sie können die AWS FleetWise IoT-Konsole verwenden, um eine Kampagne zur Auswahl, Erfassung und Übertragung von Fahrzeugdaten in die Cloud zu erstellen.

So erstellen Sie eine Kampagne

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im Navigationsbereich Kampagnen aus.
3. Wählen Sie auf der Seite Kampagnen die Option Kampagne erstellen aus, und führen Sie dann die Schritte in den folgenden Themen aus.

## Themen

- [Schritt 1: Kampagne konfigurieren](#)
- [Schritt 2: Definieren Sie das Speicherziel](#)
- [Schritt 3: Fahrzeuge hinzufügen](#)
- [Schritt 4: Überprüfen und Erstellen](#)
- [Schritt 5: Implementieren Sie eine Kampagne](#)

### Important

- Sie benötigen einen Signalkatalog und ein Fahrzeug, bevor Sie eine Kampagne erstellen können. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#) und [Fahrzeuge erstellen, bereitstellen und verwalten](#).
- Nachdem eine Kampagne erstellt wurde, müssen Sie die Kampagne genehmigen. Weitere Informationen finden Sie unter [Schritt 5: Implementieren Sie eine Kampagne](#).

## Schritt 1: Kampagne konfigurieren

Gehen Sie unter Allgemeine Informationen wie folgt vor:

1. Geben Sie einen Namen für die Kampagne ein.
2. (Optional) Geben Sie eine Beschreibung ein.

Konfigurieren Sie das Datenerfassungsschema der Kampagne. Ein Datenerfassungsschema gibt der Edge Agent-Software Anweisungen darüber, welche Daten gesammelt werden sollen oder wann sie gesammelt werden sollen. In der AWS FleetWise IoT-Konsole können Sie ein Datenerfassungsschema auf folgende Weise konfigurieren:

- Definieren Sie das Datenerfassungsschema manuell.
- Laden Sie eine Datei hoch, um das Datenerfassungsschema automatisch zu definieren.

Wählen Sie unter Konfigurationsoption eine der folgenden Optionen aus:

- Um den Typ des Datenerfassungsschemas manuell anzugeben und Optionen zur Anpassung des Schemas zu definieren, wählen Sie Datenerfassungsschema definieren aus.

Geben Sie manuell den Typ des Datenerfassungsschemas an und definieren Sie Optionen zur Anpassung des Schemas.

1. Wählen Sie im Abschnitt Details zum Datenerfassungsschema die Art des Datenerfassungsschemas aus, das für diese Kampagne verwendet werden soll. Wenn Sie anhand eines logischen Ausdrucks erkennen möchten, welche Fahrzeugdaten erfasst werden sollen, wählen Sie „Zustandsbasiert“. Wenn Sie anhand eines bestimmten Zeitraums entscheiden möchten, wie oft Fahrzeugdaten erfasst werden sollen, wählen Sie Zeitbasiert.
2. Definieren Sie die Dauer, für die die Kampagne Daten sammelt.

### Note

Standardmäßig wird eine genehmigte Kampagne sofort aktiviert und hat keine festgelegte Endzeit. Um zusätzliche Gebühren zu vermeiden, müssen Sie einen Zeitraum angeben.

3. Wenn Sie ein auf Bedingungen basierendes Datenerfassungsschema angegeben haben, müssen Sie einen logischen Ausdruck definieren, um zu erkennen, welche Daten gesammelt

werden sollen. AWS IoT FleetWise verwendet einen logischen Ausdruck, um zu erkennen, welche Daten für ein zustandsorientiertes Schema gesammelt werden müssen. Der Ausdruck muss den vollständig qualifizierten Namen eines Signals als Variable, einen Vergleichsoperator und einen Vergleichswert angeben.

Wenn Sie beispielsweise den `$variable.`myVehicle.InVehicleTemperature` > 50.0` Ausdruck angeben, FleetWise erfasst AWS IoT Temperaturwerte, die größer als 50,0 sind. Anweisungen zum Schreiben von Ausdrücken finden Sie unter [Logische Ausdrücke für Kampagnen](#).

Geben Sie den logischen Ausdruck ein, anhand dessen erkannt wird, welche Daten erfasst werden sollen.

4. (Optional) Sie können die Sprachversion des bedingten Ausdrucks angeben. Der Standardwert lautet 1.
5. (Optional) Sie können das minimale Triggerintervall angeben, das die kleinste Zeitspanne zwischen zwei Datenerfassungsereignissen darstellt. Wenn sich ein Signal beispielsweise häufig ändert, möchten Sie möglicherweise Daten langsamer erfassen.
6. Geben Sie die Bedingung für den Triggermodus an, unter der die Edge Agent-Software Daten sammelt. Standardmäßig sammelt die Edge Agent for AWS FleetWise IoT-Software immer dann Daten, wenn die Bedingung erfüllt ist. Oder er kann nur Daten sammeln, wenn die Bedingung zum ersten Mal erfüllt ist (beim ersten Trigger).
7. Wenn Sie ein zeitbasiertes Datenerfassungsschema angegeben haben, müssen Sie einen Zeitraum in Millisekunden zwischen 10.000 und 60.000 Millisekunden angeben. Die Edge Agent-Software entscheidet anhand des Zeitraums, wie oft Daten erfasst werden sollen.
8. (Optional) Sie können die erweiterten Schemaoptionen des Schemas bearbeiten.
  - a. Um drahtlose Bandbreite zu sparen und den Netzwerkverkehr durch Komprimieren von Daten zu reduzieren, wählen Sie Snappy.
  - b. (Optional) Um zu definieren, wie lange (in Millisekunden) die Datenerfassung nach einem Datenerfassungsereignis fortgesetzt werden soll, können Sie die Dauer der Erfassung nach dem Trigger angeben.
  - c. (Optional) Um die Prioritätsstufe der Kampagne anzugeben, können Sie die Kampagnenpriorität angeben. Kampagnen mit einer kleineren Prioritätsnummer werden zuerst bereitgestellt und haben dann eine höhere Priorität.
  - d. Die Edge Agent-Software kann Daten vorübergehend lokal speichern, wenn ein Fahrzeug nicht mit der Cloud verbunden ist. Nachdem die Verbindung wiederhergestellt wurde,

werden die lokal gespeicherten Daten automatisch in die Cloud übertragen. Geben Sie an, ob der Edge-Agent Daten während einer unterbrochenen Verbindung lokal speichern soll.

- e. (Optional) Um zusätzliche Informationen für ein Signal bereitzustellen, fügen Sie bis zu fünf Attribute als Zusätzliche Datendimensionen hinzu.
- Um eine Datei zur Definition des Datenerfassungsschemas hochzuladen, wählen Sie „json-Datei von Ihrem lokalen Gerät hochladen“. AWS IoT definiert FleetWise automatisch, welche Optionen Sie in der Datei definieren können. Sie können die ausgewählten Optionen überprüfen und aktualisieren.

Laden Sie eine JSON-Datei mit Details zum Datenerfassungsschema hoch.

1. Um Informationen über das Datenerfassungsschema zu importieren, wählen Sie Dateien auszuwählen. Weitere Informationen zum erforderlichen Dateiformat finden Sie in der [CreateCampaignAPI-Dokumentation](#).

 Note

AWSIoT unterstützt FleetWise derzeit die Dateiformaterweiterung „.json“.

2. AWSIoT definiert das Datenerfassungsschema FleetWise automatisch auf der Grundlage der Informationen in Ihrer Datei. Sehen Sie sich die Optionen an, die AWS IoT für Sie FleetWise ausgewählt hat. Sie können die Optionen bei Bedarf aktualisieren.

Geben Sie Signale an

Sie können die Signale angeben, aus denen Daten gesammelt werden sollen, wenn das Datenerfassungsschema aufgerufen wird.

 Important

Signale, die im Ausdruck für das bedingungs-basierte Erfassungsschema verwendet werden, müssen in diesem Feld angegeben werden.

Um die Signale anzugeben, von denen Daten gesammelt werden sollen

1. Suchen Sie nach dem vollqualifizierten Namen des Signals.

**Note**

Der vollqualifizierte Name des Signals besteht aus dem Pfad zum Signal plus dem Namen des Signals. Verwenden Sie einen Punkt (.), um auf ein untergeordnetes Signal zu verweisen.

Dies `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` ist beispielsweise der vollqualifizierte Name für den `HandsOffSteeringState` Aktuator. `Vehicle.Chassis.SteeringWheel.HandsOff` ist der Pfad zu diesem Aktuator.

2. (Optional) Geben Sie für Max. Stichprobenanzahl die maximale Anzahl von Datenproben ein, die die Edge Agent-Software sammelt und in die Cloud überträgt, wenn das Datenerfassungsschema aufgerufen wird.
3. (Optional) Geben Sie für Min. Stichprobenintervall die Mindestdauer zwischen zwei Datenprobenerfassungsereignissen in Millisekunden ein. Wenn sich ein Signal häufig ändert, können Sie diesen Parameter verwenden, um Daten langsamer zu sammeln.
4. Um ein weiteres Signal hinzuzufügen, wählen Sie Weitere Signale hinzufügen. Sie können bis zu 999 Signale hinzufügen.
5. Wählen Sie Weiter.

## Schritt 2: Definieren Sie das Speicherziel

**Note**

Sie können Fahrzeugdaten nur an Amazon S3 übertragen, wenn die Kampagne Datensignale des Bildverarbeitungssystems enthält.

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

Wählen Sie das Ziel aus, an dem Sie die im Rahmen der Kampagne gesammelten Daten speichern möchten. Sie können Fahrzeugdaten an Amazon S3 oder Amazon Timestream übertragen.

Gehen Sie in den Zieleinstellungen wie folgt vor:

- Wählen Sie S3 oder Timestream aus der Dropdownliste aus.

Um Fahrzeugdaten in einem S3-Bucket zu speichern, wählen Sie Amazon S3. S3 ist ein Objektspeicherdienst, der Daten als Objekte in Buckets speichert. Weitere Informationen finden Sie unter [Erstellen, Konfigurieren und Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.

S3 optimiert die Kosten für die Datenspeicherung und bietet zusätzliche Mechanismen zur Nutzung von Fahrzeugdaten, wie z. B. Data Lakes, zentrale Datenspeicherung, Datenverarbeitungspipelines und Analysen. Sie können S3 verwenden, um Daten für die Stapelverarbeitung und Analyse zu speichern. Sie können beispielsweise Berichte über Ereignisse mit starker Bremsung für Ihr Modell für maschinelles Lernen (ML) erstellen. Eingehende Fahrzeugdaten werden vor der Auslieferung für 10 Minuten zwischengespeichert.

## Amazon S3

### Important

Sie können Daten nur an S3 übertragen, wenn AWS IoT FleetWise über Schreibberechtigungen in den S3-Bucket verfügt. Weitere Informationen zur Zugriffsgewährung finden Sie unter [Zugriffskontrolle mit AWS IoT FleetWise](#).

Gehen Sie in den S3-Zieleinstellungen wie folgt vor:

1. Wählen Sie für den S3-Bucket einen Bucket aus, der AWS IoT FleetWise über Berechtigungen für verfügt.
2. (Optional) Geben Sie ein benutzerdefiniertes Präfix ein, mit dem Sie die im S3-Bucket gespeicherten Daten organisieren können.
3. Wählen Sie das Ausgabeformat aus. Dabei handelt es sich um die Formatdateien, die wie im S3-Bucket gespeichert werden.
4. Wählen Sie aus, ob Sie die im S3-Bucket gespeicherten Daten als .gzip-Datei komprimieren möchten. Wir empfehlen die Komprimierung von Daten, da dadurch die Speicherkosten minimiert werden.
5. Die Optionen, die Sie in den S3-Zieleinstellungen auswählen, ändern den Beispiel-S3-Objekt-URI. Dies ist ein Beispiel dafür, wie Dateien in S3 gespeichert werden.

Um Fahrzeugdaten in einer Timestream-Tabelle zu speichern, wählen Sie Amazon Timestream. Sie können Timestream verwenden, um Fahrzeugdaten abzufragen, um Trends und Muster zu

erkennen. Sie können Timestream beispielsweise verwenden, um einen Alarm für den Kraftstoffstand des Fahrzeugs zu erstellen. Eingehende Fahrzeugdaten werden nahezu in Echtzeit an Timestream übertragen. Weitere Informationen finden Sie unter [Was ist Amazon Timestream?](#) im Amazon Timestream Developer Guide.

## Amazon Timestream

### Important

Sie können Daten nur in eine Tabelle übertragen, wenn AWS IoT berechtigt FleetWise ist, Daten in Timestream zu schreiben. Weitere Informationen zur Zugriffsgewährung finden Sie unter [Zugriffskontrolle mit AWS IoT FleetWise](#).

Gehen Sie in den Einstellungen der Timestream-Tabelle wie folgt vor:

1. Wählen Sie als Timestream-Datenbankname den Namen Ihrer Timestream-Datenbank aus der Dropdownliste aus.
2. Wählen Sie als Timestream-Tabellenname den Namen Ihrer Timestream-Tabelle aus der Dropdown-Liste aus.

Gehen Sie unter Servicezugriff für Timestream wie folgt vor:

- Wählen Sie eine IAM-Rolle aus der Dropdownliste aus.
- Wählen Sie Weiter.

## Schritt 3: Fahrzeuge hinzufügen

Um auszuwählen, für welche Fahrzeuge deine Kampagne eingesetzt werden soll, wähle sie in der Fahrzeugliste aus. Filtern Sie Fahrzeuge, indem Sie nach den Attributen und ihren Werten suchen, die Sie bei der Erstellung der Fahrzeuge hinzugefügt haben, oder nach Fahrzeugnamen.

Gehen Sie unter Fahrzeuge filtern wie folgt vor:

1. Suchen Sie im Suchfeld das Attribut oder den Fahrzeugnamen und wählen Sie es aus der Liste aus.

**Note**

Jedes Attribut kann nur einmal verwendet werden.

2. Geben Sie den Wert des Attributs oder den Fahrzeugnamen ein, für das Sie die Kampagne bereitstellen möchten. Wenn der vollqualifizierte Name des Attributs beispielsweise lautet `fuelType`, geben Sie `gasoline` als Wert ein.
3. Um nach einem anderen Fahrzeugattribut zu suchen, wiederholen Sie die vorherigen Schritte. Sie können nach bis zu fünf Fahrzeugattributen und einer unbegrenzten Anzahl von Fahrzeugnamen suchen.
4. Fahrzeuge, die Ihrer Suche entsprechen, werden unter Fahrzeugname aufgeführt. Wählen Sie die Fahrzeuge aus, für die Sie die Kampagne einsetzen möchten.

**Note**

In den Suchergebnissen werden bis zu 100 Fahrzeuge angezeigt. Wählen Sie Alle auswählen, um alle Fahrzeuge zur Kampagne hinzuzufügen.

5. Wählen Sie Weiter.

## Schritt 4: Überprüfen und Erstellen

Überprüfen Sie die Konfigurationen für die Kampagne und wählen Sie dann Kampagne erstellen aus.

**Note**

Nachdem eine Kampagne erstellt wurde, müssen Sie oder Ihr Team die Kampagne für Fahrzeuge bereitstellen.

## Schritt 5: Implementieren Sie eine Kampagne

Nachdem Sie eine Kampagne erstellt haben, müssen Sie oder Ihr Team die Kampagne in Fahrzeugen einsetzen.

Um eine Kampagne zu implementieren

1. Wählen Sie auf der Seite mit der Kampagnenübersicht die Option Bereitstellen aus.

- Überprüfe und bestätige, dass du mit der Bereitstellung beginnen und mit der Erfassung von Daten von Fahrzeugen beginnen möchtest, die mit der Kampagne verbunden sind.
- Wählen Sie Bereitstellen aus.

Wenn Sie die Erfassung von Daten von Fahrzeugen, die mit der Kampagne verbunden sind, unterbrechen möchten, wählen Sie auf der Seite mit der Kampagnenübersicht die Option Sperren aus. Wenn du die Erfassung von Daten von Fahrzeugen fortsetzen möchtest, die mit der Kampagne verbunden sind, wähle „Fortfahren“.

## Erstelle eine Kampagne (AWS CLI)

Sie können den [CreateCampaign](#) API-Vorgang verwenden, um eine Kampagne zu erstellen. Das folgende Beispiel verwendet die AWS CLI.

Wenn Sie eine Kampagne erstellen, können von Fahrzeugen gesammelte Daten entweder in Amazon S3 (S3) oder Amazon Timestream gespeichert werden. Wählen Sie Timestream für eine schnelle, skalierbare und serverlose Zeitreihendatenbank, z. B. zum Speichern von Daten, die nahezu in Echtzeit verarbeitet werden müssen. Wählen Sie S3 als Objektspeicher mit branchenführender Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung.

### Important

Sie können Fahrzeugdaten nur übertragen, wenn AWS IoT berechtigt FleetWise ist, Daten in S3 oder Timestream zu schreiben. Weitere Informationen zur Zugriffsgewährung finden Sie unter [Zugriffskontrolle mit AWS IoT FleetWise](#).

## Kampagne erstellen

### Important

- Sie benötigen einen Signalkatalog und ein Fahrzeug oder eine Flotte, bevor Sie eine Kampagne erstellen können. Weitere Informationen finden Sie unter [Signalkataloge erstellen und verwalten](#), [Fahrzeuge erstellen, bereitstellen und verwalten](#) und [Flotten erstellen und verwalten](#).

- Nachdem eine Kampagne erstellt wurde, müssen Sie den UpdateCampaign API-Vorgang verwenden, um die Kampagne zu genehmigen. Weitere Informationen finden Sie unter [Aktualisieren Sie eine Kampagne \(AWS CLI\)](#)

Führen Sie den folgenden Befehl aus, um eine Kampagne zu erstellen.

Ersetzen Sie *file-name* durch den Namen der JSON-Datei, die die Kampagnenkonfiguration enthält.

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

- Ersetzen Sie *kampagnenname* durch den Namen der Kampagne, die Sie erstellen.
- *signal-catalog-arn* Ersetzen Sie durch den Amazon-Ressourcennamen (ARN) des Signalkatalogs.
- Ersetzen Sie *target-arn* durch den ARN einer Flotte oder eines Fahrzeugs, das Sie erstellt haben.
- Ersetzen Sie *bucket-arn* durch den ARN des S3-Buckets.

```
{
  "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"
        }
    }
]
}

```

- Ersetzen Sie *kampagnenname* durch den Namen der Kampagne, die Sie erstellen.
- *signal-catalog-arn* Ersetzen Sie durch den Amazon-Ressourcennamen (ARN) des Signalkatalogs.
- Ersetzen Sie *target-arn* durch den ARN einer Flotte oder eines Fahrzeugs, das Sie erstellt haben.
- Ersetzen Sie *role-arn* durch den ARN der Aufgabenausführungsrolle, die AWS FleetWise IoT-Berechtigungen zur Übermittlung von Daten an die Timestream-Tabelle erteilt.
- Ersetzen Sie *table-arn* durch den ARN der Timestream-Tabelle.

```

{
  "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"
    }
  }
]
}
```

## Logische Ausdrücke für Kampagnen

AWS IoT FleetWise verwendet einen logischen Ausdruck, um zu erkennen, welche Daten im Rahmen einer Kampagne gesammelt werden sollen. Weitere Informationen zu Ausdrücken finden Sie unter [Ausdrücke](#) im AWS IoT Events Entwicklerhandbuch.

Die Ausdrucksvariable sollte so konstruiert sein, dass sie den Regeln für den Typ der gesammelten Daten entspricht. Bei Telemetriesystemdaten sollte die Ausdrucksvariable der vollständig qualifizierte Name des Signals sein. Bei Bildverarbeitungssystemdaten kombiniert der Ausdruck den vollständig qualifizierten Namen des Signals mit dem Pfad, der vom Datentyp des Signals zu einer seiner Eigenschaften führt.

Wenn der Signalkatalog beispielsweise die folgenden Knoten enthält:

```
{
  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"
}
```

Wenn die Knoten der ROS 2-Definition folgen:

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

Im Folgenden sind alle möglichen Variablen für Ereignisausdrücke aufgeführt:

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

## Aktualisieren Sie eine Kampagne (AWS CLI)

Sie können den [UpdateCampaign](#) API-Vorgang verwenden, um eine bestehende Kampagne zu aktualisieren. Der folgende Befehl verwendet AWS CLI.

- Ersetzen Sie *kampagnenname* durch den Namen der Kampagne, die Sie aktualisieren.
- Ersetzen Sie *action* durch eine der folgenden Optionen:
  - APPROVE— Genehmigt die Kampagne, damit das AWS IoT der Dinge FleetWise in einem Fahrzeug oder einer Flotte eingesetzt werden kann.
  - SUSPEND— Setzt die Kampagne aus. Die Kampagne wird aus den Fahrzeugen gelöscht und alle Fahrzeuge der ausgesetzten Kampagne senden keine Daten mehr.
  - RESUME— Reaktiviert die SUSPEND Kampagne. Die Kampagne wird auf alle Fahrzeuge übertragen und die Fahrzeuge werden wieder Daten senden.
  - UPDATE— Aktualisiert die Kampagne, indem Attribute definiert und sie mit einem Signal verknüpft werden.

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

## Löscht eine Kampagne

Sie können die AWS FleetWise IoT-Konsole oder API verwenden, um Kampagnen zu löschen.

### Löschen Sie eine Kampagne (Konsole)

Verwenden Sie die AWS FleetWise IoT-Konsole, um eine Kampagne zu löschen.

Um eine Kampagne zu löschen

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).

2. Wählen Sie im Navigationsbereich Kampagnen aus.
3. Wählen Sie auf der Seite Kampagnen die Zielkampagne aus.
4. Wählen Sie Löschen aus.
5. Unter Löschen **campaign-name?** , geben Sie den Namen der Kampagne ein, die gelöscht werden soll, und wählen Sie dann Bestätigen.

## Löschen Sie eine Kampagne (AWS CLI)

Sie können den [DeleteCampaign](#) API-Vorgang verwenden, um eine Kampagne zu löschen. Im folgenden Beispiel wird verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um eine Kampagne zu löschen.

Ersetzen Sie *kampagnenname* durch den Namen des Fahrzeugs, das Sie löschen möchten.

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

## Kampagneninformationen abrufen () AWS CLI

Sie können den [ListCampaigns](#) API-Vorgang verwenden, um zu überprüfen, ob eine Kampagne gelöscht wurde. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um eine paginierte Liste mit Zusammenfassungen für alle Kampagnen abzurufen.

```
aws iotfleetwise list-campaigns
```

Sie können den [GetCampaign](#) API-Vorgang verwenden, um Fahrzeuginformationen abzurufen. Das folgende Beispiel verwendet die AWS CLI.

Führen Sie den folgenden Befehl aus, um die Metadaten einer Kampagne abzurufen.

Ersetzen Sie *kampagnenname* durch den Namen der Kampagne, die Sie abrufen möchten.

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

 Note

Diese Operation ist letztlich konsistent. Mit anderen Worten, Änderungen an der Kampagne werden möglicherweise nicht sofort übernommen.

# Verarbeitung und Visualisierung von Fahrzeugdaten

Die Edge Agent for AWS FleetWise IoT-Software überträgt ausgewählte Fahrzeugdaten an Amazon Timestream oder Amazon Simple Storage Service (Amazon S3). Nachdem Ihre Daten am Datenziel angekommen sind, können Sie sie mit anderen AWS Diensten visualisieren und teilen.

## Verarbeitung von Fahrzeugdaten in Timestream

Timestream ist eine vollständig verwaltete Zeitreihendatenbank, die Billionen von Zeitreihendatenpunkten pro Tag speichern und analysieren kann. Ihre Daten werden in einer vom Kunden verwalteten Timestream-Tabelle gespeichert. Sie können Timestream verwenden, um Fahrzeugdaten abzufragen, um Einblicke in Ihre Fahrzeuge zu gewinnen. Weitere Informationen finden Sie unter [Was ist Amazon Timestream?](#)

Das Standardschema für Daten, die an Timestream übertragen werden, enthält die folgenden Felder.

Feldname	Datentyp	Beschreibung
eventId	varchar	Die ID des Datenerfassungsereignisses.
vehicleName	varchar	Die ID des Fahrzeugs, von dem die Daten gesammelt wurden.
name	varchar	Der Name der Kampagne, mit der die Edge Agent-Software Daten sammelt.
time	Zeitstempel	Der Zeitstempel des Datenpunkts.
measure_name	varchar	Der Name des Signals.
measure_value::bigint	bigint	Signalwerte vom Typ Integer.

Feldname	Datentyp	Beschreibung
measure_value::double	double	Signalwerte vom Typ Double.
measure_value::boolean	boolesch	Signalwerte des Typs Boolean.

## Visualisierung der in Timestream gespeicherten Fahrzeugdaten

Nachdem Ihre Fahrzeugdaten an Timestream übertragen wurden, können Sie die folgenden AWS Dienste verwenden, um Ihre Daten zu visualisieren, zu überwachen, zu analysieren und zu teilen.

- Visualisieren und überwachen Sie Daten in Dashboards mithilfe von [Grafana oder Amazon Managed Grafana](#). Sie können Daten aus mehreren AWS Quellen (wie Amazon CloudWatch und Timestream) und anderen Datenquellen mit einem einzigen Grafana-Dashboard visualisieren.
- Analysieren und visualisieren Sie Daten in Dashboards mithilfe von [Amazon QuickSight](#).

## Verarbeitung von Fahrzeugdaten in S3

Amazon S3 ist ein Objektspeicherservice, der beliebige Datenmengen speichert und schützt. Sie können S3 für eine Vielzahl von Anwendungsfällen verwenden, z. B. für Datensichten, Sicherung und Wiederherstellung, Archivierung, Unternehmensanwendungen, AWS IoT Geräte und Big-Data-Analysen. Ihre Daten werden in S3 als Objekte in Buckets gespeichert. Weitere Informationen finden Sie unter [Was ist Amazon S3?](#)

Das Standardschema von Daten, die an Amazon S3 übertragen werden, enthält die folgenden Felder.

Feldname	Datentyp	Beschreibung
eventId	varchar	Die ID des Datenerfassungsereignisses.

Feldname	Datentyp	Beschreibung
vehicleName	varchar	Die ID des Fahrzeugs , von dem die Daten gesammelt wurden.
name	varchar	Der Name der Kampagne, mit der die Edge Agent-Software Daten sammelt.
time	Zeitstempel	Der Zeitstempel des Datenpunkts.
measure_name	varchar	Der Name des Signals.
measure_value_BIGINT	bigint	Signalwerte vom Typ Integer.
measure_value_DOUBLE	double	Signalwerte vom Typ Double.
measure_value_BOOLEAN	boolesch	Signalwerte des Typs Boolean.
measure_value_STRUCT	struct	Signalwerte des Typs Struct.

## S3-Objektformat

AWS IoT FleetWise überträgt Fahrzeugdaten an S3, wo sie als Objekt gespeichert werden. Sie können den Objekt-URI verwenden, der die Daten eindeutig identifiziert, um Daten aus der Kampagne zu finden. Das S3-Objekt-URI-Format hängt davon ab, ob es sich bei den gesammelten Daten um unstrukturierte oder verarbeitete Daten handelt.

## Unstrukturierte Daten

Unstrukturierte Daten werden in S3 auf nicht vordefinierte Weise gespeichert. Es kann in verschiedenen Formaten vorliegen, z. B. in Bildern oder Videos.

Fahrzeugnachrichten, die FleetWise mit Signaldaten aus Amazon Ion-Dateien an das AWS IoT übergeben werden, werden dekodiert und als Objekte an S3 übertragen. Die S3-Objekte repräsentieren jedes Signal und sind binär codiert.

Der S3-Objekt-URI für unstrukturierte Daten verwendet das folgende Format:

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

## Verarbeitete Daten

Verarbeitete Daten werden in S3 gespeichert und durchlaufen Verarbeitungsschritte, die Nachrichten validieren, anreichern und transformieren. Objektlisten und Geschwindigkeit sind Beispiele für verarbeitete Daten.

An S3 übertragene Daten werden als Objekte gespeichert, die Datensätze darstellen, die für einen Zeitraum von etwa 10 Minuten zwischengespeichert wurden. Standardmäßig FleetWise fügt AWS IoT dem Format ein UTC-Zeitpräfix hinzu, `year=YYYY/month=MM/date=DD/hour=HH` bevor Objekte in S3 geschrieben werden. Dieses Präfix erstellt eine logische Hierarchie im Bucket, in der jeder Schrägstrich (/) eine Ebene in der Hierarchie erzeugt. Die verarbeiteten Daten enthalten auch den S3-Objekt-URI für unstrukturierte Daten.

Der S3-Objekt-URI für verarbeitete Daten verwendet das folgende Format:

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

## Rohdaten

Rohdaten, auch Primärdaten genannt, sind Daten, die aus Amazon Ion-Dateien gesammelt wurden. Sie können Rohdaten verwenden, um Probleme zu beheben oder Fehler zu beheben.

Der S3-Objekt-URI für Rohdaten verwendet das folgende Format:

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

# Analyse der in S3 gespeicherten Fahrzeugdaten

Nachdem Ihre Fahrzeugdaten an S3 übertragen wurden, können Sie die folgenden AWS Dienste verwenden, um Ihre Daten zu überwachen, zu analysieren und zu teilen.

Extrahieren und analysieren Sie Daten mit Amazon SageMaker für nachgelagerte Labeling- und Machine Learning-Workflows (ML).

Weitere Informationen finden Sie in den folgenden Themen im Amazon SageMaker Developer Guide:

- [Daten verarbeiten](#)
- [Trainieren Sie Modelle für maschinelles Lernen](#)
- [Bilder beschriften](#)

Katalogisieren Sie Ihre Daten mithilfe AWS-Glue-Crawler und analysieren Sie sie in Amazon Athena. Standardmäßig verfügen in S3 geschriebene Objekte über Zeitpartitionen im Apache Hive-Stil mit Datenpfaden, die Schlüssel-Wert-Paare enthalten, die durch Gleichheitszeichen verbunden sind.

Weitere Informationen finden Sie in den folgenden Themen im Amazon Athena Athena-Benutzerhandbuch:

- [Partitionierung von Daten in Athena](#)
- [Wird verwendet AWS Glue, um eine Verbindung zu Datenquellen in Amazon S3 herzustellen](#)
- [Bewährte Methoden bei der Verwendung von Athena mit AWS Glue](#)

Visualisieren Sie Daten mit Amazon QuickSight indem Sie entweder Ihre Athena-Tabelle oder Ihren S3-Bucket direkt lesen.

## Tip

Wenn Sie direkt aus S3 lesen, vergewissern Sie sich, dass Ihre Fahrzeugdaten im JSON-Format vorliegen, da Amazon QuickSight das Apache Parquet-Format nicht unterstützt.

Weitere Informationen finden Sie in den folgenden Themen im QuickSight Amazon-Benutzerhandbuch:

- [Unterstützte Datenquellen](#)

- [Eine Datenquelle erstellen](#)

## AWS CLI und AWS-SDKs

Dieser Abschnitt enthält Informationen zur Herstellung von AWS IoT FleetWise API-Anfragen. Weitere Informationen zu folgenden Themen: AWS IoT FleetWise [Operationen und Datentypen](#) finden Sie unter [AWS IoT FleetWise API-Referenz](#).

Zu verwenden AWS IoT FleetWise mit einer Vielzahl von Programmiersprachen verwenden Sie [AWS-SDKs](#), die die folgenden automatischen Funktionen enthalten:

- Kryptographisches Signieren Ihrer Serviceanfragen
- Wiederholen von Anfragen
- Umgang mit Fehlerreaktionen

Für den Zugriff auf die Befehlszeile verwenden Sie AWS IoT FleetWise mit dem [AWS CLI](#). Du kannst kontrollieren AWS IoT FleetWise und automatisiert diese mithilfe von Skripten.

# AWSIoT-Problembhebung FleetWise

Verwenden Sie die Informationen und Lösungen zur Fehlerbehebung in diesem Abschnitt, um Probleme mit AWS IoT zu lösen FleetWise.

Die folgenden Informationen können Ihnen helfen, häufig auftretende Probleme mit AWS IoT zu beheben FleetWise.

## Themen

- [Probleme mit dem Decoder-Manifest](#)
- [FleetWise Softwareprobleme mit dem Edge-Agent für AWS IoT](#)

## Probleme mit dem Decoder-Manifest

Beheben Sie Probleme mit dem Decoder-Manifest.

### Diagnose von API-Aufrufen des Decoder-Manifests

Fehler	Richtlinien für die Fehlerbehebung
<code>UpdateOperationFailure.ConflictingDecoderUpdate</code>	Das gleiche Decoder-Manifest enthält mehrere Aktualisierungsanforderungen. Warten Sie und versuchen Sie es erneut.
<code>UpdateOperationFailure.InternalFailure</code>	InternalFailure wird als gekapselte Ausnahme gestartet. Das Problem selbst hängt von der gekapselten Ausnahme ab.
<code>UpdateOperationFailure.ActiveDecoderUpdate</code>	Das Decoder-Manifest befindet sich in einem Active Zustand und kann nicht aktualisiert werden. Ändern Sie den Status des Decoder-Manifests auf DRAFT, und versuchen Sie es erneut.
<code>UpdateOperationFailure.ConflictingModelUpdate</code>	AWSIoT FleetWise versucht, anhand eines Fahrzeugmodells (Modellmanifest) zu validieren, das von einer anderen Person geändert wird. Warten Sie und versuchen Sie es erneut.

Fehler	Richtlinien für die Fehlerbehebung
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND</pre>	<p>Dem Fahrzeugmodell sind keine Signale zugeordnet. Fügen Sie dem Fahrzeugmodell Signale hinzu und überprüfen Sie, ob die Signale im zugehörigen Signalkatalog zu finden sind.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE</pre>	<p>Aktualisieren Sie das Fahrzeugmodell, sodass es den aktuellen ACTIVE Status aufweist, und versuchen Sie es erneut.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_FOUND</pre>	<p>AWS IoT FleetWise kann das mit dem Decoder-Manifest verknüpfte Fahrzeugmodell nicht finden. Überprüfen Sie den Amazon Resource Name (ARN) des Fahrzeugmodells und versuchen Sie es erneut.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE)</pre>	<p>Die Validierung des Fahrzeugmodells ist fehlgeschlagen, weil die Signalnamen des Fahrzeugmodells nicht im Signalkatalog gefunden wurden. Stellen Sie sicher, dass die Signale im Fahrzeugmodell alle im zugehörigen Signalkatalog enthalten sind.</p>
<pre>UpdateOperationFailure.ValidationFailure</pre>	<p>In der Anforderung zur Aktualisierung des Decoder-Manifests wurden ungültige Signale oder Netzwerkschnittstellen gefunden. Stellen Sie sicher, dass alle von der Ausnahme zurückgegebenen Signale und Netzwerkschnittstellen vorhanden sind, dass alle verwendeten Signale einer verfügbaren Schnittstelle zugeordnet sind und dass Sie keine Schnittstelle entfernen, der Signale zugeordnet sind.</p>

Fehler	Richtlinien für die Fehlerbehebung
<code>UpdateOperationFailure.KmsKeyAccessDenied</code>	Bei der für den Vorgang verwendeten Taste AWS Key Management Service (AWS KMS) ist ein Berechtigungsproblem aufgetreten. Stellen Sie sicher, dass Sie eine Rolle verwenden, die Zugriff auf den Schlüssel hat, und versuchen Sie es erneut.
<code>UpdateOperationFailure.DecoderDoesNotExist</code>	Das Decoder-Manifest ist nicht vorhanden. Überprüfen Sie den Namen des Decoder-Manifests und versuchen Sie es erneut.

Fehlermeldungen in den Daten des Vision-Systems mit Angabe des `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` Grundes enthalten in der Antwort einen Hinweis, der Aufschluss darüber gibt, warum die Anfrage fehlgeschlagen ist. Anhand des Hinweises können Sie festlegen, welche Richtlinien zur Fehlerbehebung befolgt werden müssen.

#### Note

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

### Diagnose, Decoder, Validierung der Daten des Bildverarbeitungssystems

Fehler	Richtlinien für die Fehlerbehebung
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</code>	AWS IoT FleetWise hat die im Signaldecoder verwendete Wurzelsignalstruktur mithilfe des Signalkatalogs nicht gefunden. Stellen Sie sicher, dass das Wurzelsignal der Struktur im Signalkatalog korrekt definiert ist.
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_IN</code>	Eine primitive Nachricht im Signalkatalog wurde nicht mit demselben Datentyp in der Aktualisierungsanforderung des Decoder-Manifests definiert. Stellen Sie sicher, dass die in der

Fehler	Richtlinien für die Fehlerbehebung
<p><code>COMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</code></p> <p><code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</code></p>	<p>Anfrage definierten primitiven Nachrichten mit der entsprechenden Signalkatalogdefinition übereinstimmen.</p> <p>Die Anzahl der in einer Struktur im Signalkatalog definierten Eigenschaften entspricht nicht der Anzahl der Eigenschaften, die Sie im Decoder-Manifest zu dekodieren versuchen. Stellen Sie sicher, dass Sie die richtige Anzahl von Signalen zum Dekodieren haben, indem Sie diese mit den im Signalkatalog definierten Signalen vergleichen.</p>
<p><code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code></p>	<p>AWSIoT FleetWise hat ein Signal gefunden, das im Signalkatalog als STRUCT definiert ist, ohne dass eine im Decoder structure dMessageDefinition definierte Manifest-Anfrage vorhanden war. Stellen Sie sicher, dass jede Struktur als Aktualisierungsanforderung structuredMessageDefinition im Decoder-Manifest definiert ist.</p>
<p><code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code></p>	<p>Das Wurzelsignal der Struktur, die im Decoder-Manifest verwendet wird, ist nicht korrekt als Struktur im Signalkatalog definiert. Für die im Decoder-Manifest verwendete Wurzelstruktur muss das Feld structFullyQualifiedName definiert sein. Dazu benötigen sie auch einen STRUCT-Knoten. fullyQualifiedName</p>
<p><code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code></p>	<p>Eine der in der Decoder-Manifest-Anfrage verwendeten Blattnachrichten ist nicht als primitive Nachricht definiert. Stellen Sie sicher, dass alle Blattobjekte in der Anfrage als primitive Nachrichten definiert sind.</p>

Fehler	Richtlinien für die Fehlerbehebung
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	Ein Array-Objekt im Signalkatalog wurde in der Aktualisierungsanforderung des Decoder-Manifests nicht als <code>structuredMessageList</code> Definition definiert. Stellen Sie sicher, dass alle Array-Eigenschaften in der Aktualisierungsanforderung für das Decoder-Manifest als <code>structuredMessageList</code> Definition definiert sind.

## FleetWise Softwareprobleme mit dem Edge-Agent für AWS IoT

Beheben Sie Probleme mit der Edge Agent-Software.

### Problembereiche

- [Problem: Die Edge Agent-Software startet nicht.](#)
- [Problem: \[FEHLER\] \[IoTFleetWiseEngine: :connect\]: \[Persistenzbibliothek konnte nicht initialisiert werden\]](#)
- [Problem: Die Edge Agent-Software erfasst keine PIDs und Diagnose-Fehlercodes \(DTCs\) für die integrierte Diagnose \(OBD\) II.](#)
- [Problem: Die Edge Agent for AWS FleetWise IoT-Software sammelt keine Daten aus dem Netzwerk oder kann keine Dateninspektionsregeln anwenden.](#)
- [Problem: \[FEHLER\] \[AwsIotConnectivityModule: :connect\]: \[Verbindung mit Fehler fehlgeschlagen\] oder \[WARN\] \[AwsIotChannel: :send\]: \[Keine aktive MQTT-Verbindung.\]](#)

### Problem: Die Edge Agent-Software startet nicht.

Möglicherweise werden die folgenden Fehler angezeigt, wenn die Edge Agent-Software nicht gestartet wird.

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

Lösung: Stellen Sie sicher, dass die FleetWise Softwarekonfigurationsdatei für den Edge-Agenten für AWS IoT ein gültiges JSON-Format verwendet. Stellen Sie z. B. sicher, dass Kommas korrekt verwendet werden. Gehen Sie wie folgt vor, um weitere Informationen zur Konfigurationsdatei zu erhalten, um das Edge Agent for AWS FleetWise IoT-Softwareentwicklerhandbuch herunterzuladen.

1. Navigieren Sie zur [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie auf der Service-Startseite im FleetWise Abschnitt Erste Schritte mit AWS IoT die Option Explore Edge Agent aus.

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

Lösung: Dieser Fehler wird möglicherweise angezeigt, wenn die Edge Agent-Software keine Socket-Kommunikation mit den in der Konfigurationsdatei definierten Netzwerkschnittstellen herstellen kann.

Führen Sie den folgenden Befehl aus, um zu überprüfen, ob jede in der Konfiguration definierte Netzwerkschnittstelle verfügbar ist.

```
ip link show
```

Führen Sie den folgenden Befehl aus, um eine Netzwerkschnittstelle online zu schalten. *network-interface-id* Ersetzen Sie es durch die ID der Netzwerkschnittstelle.

```
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]
```

Lösung: Dieser Fehler wird möglicherweise angezeigt, wenn die Edge Agent-Software keine MQTT-Verbindung zu AWS IoT Core herstellen kann. Überprüfen Sie, ob die folgenden Komponenten korrekt konfiguriert sind, und starten Sie die Edge Agent-Software neu.

- `mqtConnection::endpointUrl`— Endpunkt des IoT-Geräts des AWS Kontos.

- `mqttnConnection::clientId`— Die ID des Fahrzeugs, in dem die Edge Agent-Software ausgeführt wird.
- `mqttnConnection::certificateFilename`— Der Pfad zur Fahrzeug-Zertifikatsdatei.
- `mqttnConnection::privateKeyFilename`— Der Pfad zur Datei mit dem privaten Fahrzeugschlüssel.
- Sie haben es AWS IoT Core zur Bereitstellung des Fahrzeugs verwendet. Weitere Informationen finden Sie unter [Fahrzeuge bereitstellen](#).

Weitere Informationen zur Fehlerbehebung finden Sie unter [AWS IoT Device SDK for C++Häufig gestellte Fragen](#).

## Problem: [FEHLER] [IoTFleetWiseEngine: :connect]: [Persistenzbibliothek konnte nicht initialisiert werden]

Lösung: Dieser Fehler wird möglicherweise angezeigt, wenn die Edge Agent-Software den Persistenzspeicher nicht finden kann. Vergewissern Sie sich, dass Folgendes korrekt konfiguriert ist, und starten Sie die Edge Agent-Software neu.

`persistency::persistencyPath`— Ein lokaler Pfad, der zur Beibehaltung von Sammlungsschemas, Decodermanifesten und Datenschnappschüssen verwendet wird.

## Problem: Die Edge Agent-Software erfasst keine PIDs und Diagnose-Fehlercodes (DTCs) für die integrierte Diagnose (OBD) II.

Lösung: Dieser Fehler wird möglicherweise angezeigt, wenn `obdInterface::pidRequestIntervalSeconds` oder auf 0 konfiguriert `obdInterface::dtcRequestIntervalSeconds` ist.

Wenn die Edge Agent-Software in einem Fahrzeug mit Automatikgetriebe ausgeführt wird, stellen Sie sicher, dass sie auf konfiguriert `obdInterface::hasTransmissionEcu` ist `true`.

Wenn Ihr Fahrzeug erweiterte Controller Area Network (CAN-Bus) -Arbitrierungs-IDs unterstützt, stellen Sie sicher, dass dies konfiguriert `obdInterface::useExtendedIds` ist `true`.

**Problem:** Die Edge Agent for AWS FleetWise IoT-Software sammelt keine Daten aus dem Netzwerk oder kann keine Dateninspektionsregeln anwenden.

**Lösung:** Dieser Fehler wird möglicherweise angezeigt, wenn die Standardkontingente überschritten werden.

| Ressource                                              | Kontingent                                         | Einstellbar | Hinweis                                                                                                                                                                                                        |
|--------------------------------------------------------|----------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Wert der Signal-ID                                     | Die Signal-ID muss kleiner oder gleich 50.000 sein | Ja          | Die Edge Agent-Software sammelt keine Daten von Signalen, deren ID größer als 50.000 ist. Wir empfehlen Ihnen, zu überprüfen, wie viele Signale der Signalkatalog enthält, bevor Sie dieses Kontingent ändern. |
| Anzahl der aktiven Datenerfassungssysteme pro Fahrzeug | 256                                                | Ja          | Wir empfehlen Ihnen, zu überprüfen, wie viele Kampagnen Sie in der Cloud erstellt haben und wie viele Schemas jede Kampagne enthält, bevor Sie dieses Kontingent ändern.                                       |
| Größe des Puffers für die Signalhistorie               | 20 MB                                              | Ja          | Wenn das Kontingent überschritten wird, hört die Edge Agent-Software auf, neue Daten zu sammeln.                                                                                                               |

Problem: [FEHLER] [AwsIotConnectivityModule: :connect]: [Verbindung mit Fehler fehlgeschlagen] oder [WARN] [AwsIotChannel: :send]: [Keine aktive MQTT-Verbindung.]

Lösung: Dieser Fehler wird möglicherweise angezeigt, wenn die Edge Agent-Software nicht mit der Cloud verbunden ist. Standardmäßig sendet die Edge Agent-Software AWS IoT Core jede Minute eine Ping-Anfrage und wartet drei Minuten. Wenn keine Antwort erfolgt, stellt die Edge Agent-Software die Verbindung zur Cloud automatisch wieder her.

# Sicherheit bei AWS IoT FleetWise

Cloud-Sicherheit bei AWS hat höchste Priorität. Als AWS Als Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt ist, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung zwischen AWS und du. Das [Modell der übergreifenden Verantwortlichkeit](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- **Sicherheit der Cloud** — AWS ist verantwortlich für den Schutz der Infrastruktur, die läuft AWS Dienstleistungen in der AWS Wolke. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheit im Rahmen der [AWS Compliance-Programme](#) . Um mehr über die Compliance-Programme zu erfahren, die gelten für AWS IoT FleetWise, siehe [AWS Services im Umfang nach Compliance-Programm AWS](#) .
- **Sicherheit in der Cloud** — Ihre Verantwortung wird bestimmt durch AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung anwenden können, wenn Sie AWS IoT FleetWise. Es zeigt Ihnen, wie Sie konfigurieren AWS IoT FleetWise , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie lernen auch, wie Sie andere verwenden AWS Dienste, die Ihnen helfen, Ihre AWS FleetWise IoT-Ressourcen.

## Inhalt

- [Datenschutz in AWS IoT FleetWise](#)
- [Steuerung des Zugriffs mit AWS IoT FleetWise](#)
- [Identity and Access Management für AWS IoT FleetWise](#)
- [Konformitätsprüfung für AWS IoT FleetWise](#)
- [Resilienz in AWS IoT FleetWise](#)
- [Sicherheit der Infrastruktur in AWS IoT FleetWise](#)
- [Konfiguration und Schwachstellenanalyse in AWS IoT FleetWise](#)
- [Bewährte Sicherheitsmethoden für AWS IoT FleetWise](#)

# Datenschutz in AWS IoT FleetWise

Das Tool AWS [Das Modell](#) AWS IoT FleetWise. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre Inhalte zu behalten, die auf dieser Infrastruktur gehostet werden. Sie sind auch verantwortlich für die Sicherheitskonfiguration und die Verwaltungsaufgaben für AWS-Services die Sie verwenden. Weitere Informationen zum Datenschutz finden Sie in der [Datenschutzerklärung FAQ](#). Informationen zum Datenschutz in Europa finden Sie auf der [AWS Modell der geteilten Verantwortung und GDPR](#) Blogbeitrag auf der AWS Blog zum Thema Sicherheit.

Aus Datenschutzgründen empfehlen wir Ihnen, AWS-Konto Anmeldeinformationen und richten Sie einzelne Benutzer ein mit AWS IAM Identity Center or AWS Identity and Access Management (IAM). So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit zu kommunizieren AWS Ressourcen schützen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail. Für Informationen zur Verwendung von CloudTrail Spuren zum Erfassen AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) in der AWS CloudTrail Benutzeranleitung.
- Verwenden Sie AWS Verschlüsselungslösungen, zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff FIPS 140-3 validierte kryptografische Module benötigen AWS über eine Befehlszeilenschnittstelle oder einen API, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies beinhaltet, wenn Sie mit arbeiten AWS IoT FleetWise oder anderes AWS-Services mit der Konsole API, AWS CLI, oder AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie

einem externen Server eine URL zur Verfügung stellen, empfehlen wir dringend, dass Sie keine Anmeldeinformationen angeben, URL um Ihre Anfrage an diesen Server zu validieren.

AWS IoT FleetWise ist für die Verwendung mit einem Edge-Agenten vorgesehen, den Sie entwickeln und auf unterstützter Fahrzeughardware installieren, um Fahrzeugdaten an die AWS Wolke. Das Extrahieren von Daten aus Fahrzeugen kann in bestimmten Ländern den Datenschutzbestimmungen unterliegen. Vor der Verwendung AWS IoT FleetWise und die Installation Ihres Edge Agents. Wir empfehlen Ihnen dringend, Ihre Compliance-Verpflichtungen nach geltendem Recht zu überprüfen. Dazu gehören alle geltenden gesetzlichen Anforderungen zur Bereitstellung rechtlich angemessener Datenschutzhinweise und zur Einholung aller erforderlichen Einwilligungen für die Extraktion von Fahrzeugdaten.

## Verschlüsselung im Ruhezustand

Die von einem Fahrzeug gesammelten Daten werden über eine in die Cloud übertragen AWS IoT Core Nachricht mit dem MQTT Nachrichtenprotokoll. AWS IoT FleetWise liefert die Daten an Ihre Amazon Timestream Timestream-Datenbank. In Timestream sind Ihre Daten verschlüsselt. Alle AWS-Services verschlüsselt standardmäßig Daten im Ruhezustand.

Verschlüsselung im Ruhezustand ist integriert mit AWS Key Management Service (AWS KMS), um den Verschlüsselungsschlüssel zu verwalten, der zur Verschlüsselung Ihrer Daten verwendet wird. Sie können wählen, ob Sie einen vom Kunden verwalteten Schlüssel verwenden möchten, um die gesammelten Daten zu verschlüsseln AWS IoT FleetWise. Sie können Ihren Verschlüsselungsschlüssel erstellen, verwalten und einsehen über AWS KMS. Weitere Informationen finden Sie unter [Was ist AWS Key Management Service?](#) in der AWS Key Management Service Leitfaden für Entwickler.

## Verschlüsselung während der Übertragung

Alle Daten wurden ausgetauscht mit AWS IoT Dienste werden bei der Übertragung mithilfe von Transport Layer Security (TLS) verschlüsselt. Weitere Informationen finden Sie unter [Transportsicherheit](#) im AWS IoT Entwicklerhandbuch.

Außerdem AWS IoT Core unterstützt [Authentifizierung](#) und [Autorisierung](#), um den Zugriff auf sicher zu kontrollieren AWS FleetWise IoT-Ressourcen. Fahrzeuge können X.509-Zertifikate verwenden, um sich für die Nutzung zu authentifizieren (angemeldet) zu lassen AWS IoT FleetWise und Nutzung AWS IoT Core Richtlinien, um autorisiert zu werden (über Berechtigungen zu verfügen), bestimmte

Aktionen auszuführen. Weitere Informationen finden Sie unter [the section called “Fahrzeuge bereitstellen”](#).

## Datenverschlüsselung

Datenverschlüsselung bezieht sich auf den Schutz von Daten während der Übertragung (beim Hin- und Rücktransport). AWS IoT FleetWise und zwischen Gateways und Servern) und im Ruhezustand (während es auf lokalen Geräten gespeichert ist oder in AWS-Services). Sie können Daten im Ruhezustand mit clientseitiger Verschlüsselung schützen.

### Note

AWS FleetWise IoT-Edge-Processing-ExposésAPIs, die innerhalb gehostet werden AWS FleetWise IoT-Gateways und sind über das lokale Netzwerk zugänglich. Diese APIs werden über eine TLS Verbindung verfügbar gemacht, die durch ein Serverzertifikat gestützt wird, das Eigentum von AWS IoT FleetWise Edge-Anschluss. Für die Client-Authentifizierung APIs verwenden diese ein Passwort für die Zugriffskontrolle. Der private Schlüssel des Serverzertifikats und das Passwort für die Zugriffskontrolle werden beide auf der Festplatte gespeichert. AWS Die FleetWise IoT-Edge-Verarbeitung stützt sich auf die Dateisystemverschlüsselung, um die Sicherheit dieser gespeicherten Anmeldeinformationen zu gewährleisten.

Weitere Informationen zur serverseitigen Verschlüsselung und zur clientseitigen Verschlüsselung finden Sie in den unten aufgeführten Themen.

### Inhalt

- [Verschlüsselung im Ruhezustand](#)
- [Schlüsselverwaltung](#)

## Verschlüsselung im Ruhezustand

AWS IoT FleetWise speichert Ihre Daten im AWS In der Cloud und auf Gateways.

### Daten im Ruhezustand im AWS Cloud

AWS IoT FleetWise speichert Daten in anderen AWS-Services die Daten im Ruhezustand standardmäßig verschlüsseln. Verschlüsselung im Ruhezustand ist integriert mit [AWS Key](#)

[Management Service \(AWS KMS\)](#) zur Verwaltung des Verschlüsselungsschlüssels, der zur Verschlüsselung Ihrer Immobilienwerte und aggregierten Werte in verwendet wird AWS IoT FleetWise. Sie können sich dafür entscheiden, einen vom Kunden verwalteten Schlüssel zur Verschlüsselung von Vermögenswerten und Aggregatwerten in AWS IoT FleetWise. Sie können Ihren Verschlüsselungsschlüssel erstellen, verwalten und einsehen über AWS KMS.

Sie können eine wählen AWS-eigener Schlüssel oder einen vom Kunden verwalteten Schlüssel zur Verschlüsselung Ihrer Daten.

### Funktionsweise

Encryption at Rest integriert sich in AWS KMS zur Verwaltung des Verschlüsselungsschlüssels, der zur Verschlüsselung Ihrer Daten verwendet wird.

- **AWS-eigener Schlüssel** — Standard-Verschlüsselungsschlüssel. AWS IoT FleetWise besitzt diesen Schlüssel. Sie können diesen Schlüssel nicht in Ihrem anzeigen, verwalten oder verwenden AWS-Konto. Sie können auch keine Operationen auf der Tastatur sehen AWS CloudTrail Logs. Sie können diesen Schlüssel ohne zusätzliche Kosten verwenden.
- **Vom Kunden verwalteter Schlüssel** — Der Schlüssel wird in Ihrem Konto gespeichert, das Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über den KMS Schlüssel. Zusätzlich AWS KMS Es fallen Gebühren an.

### AWS-eigene Schlüssel

AWS-eigene Schlüssel sind nicht in Ihrem Konto gespeichert. Sie sind Teil einer Sammlung von KMS Schlüsseln AWS besitzt und verwaltet zur Verwendung in mehreren AWS-Konten. AWS-Services kann verwenden AWS-eigene Schlüssel um Ihre Daten zu schützen.

Sie können es nicht ansehen, verwalten oder verwenden AWS-eigene Schlüssel, oder ihre Verwendung überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme ändern, um Schlüssel zu schützen, die Ihre Daten verschlüsseln.

Ihnen wird keine Gebühr berechnet, wenn Sie AWS-eigene Schlüssel, und sie zählen nicht dagegen AWS KMS Kontingente für Ihr Konto.

### Kundenverwaltete Schlüssel

Von Kunden verwaltete KMS Schlüssel sind Schlüssel in Ihrem Konto, die Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über diese KMS Schlüssel, z. B. über die folgenden:

- Festlegung und Aufrechterhaltung ihrer wichtigsten Richtlinien, IAM Richtlinien und Zuschüsse
- Sie aktivieren und deaktivieren
- Rotation ihres kryptographischen Materials
- Hinzufügen von Tags
- Aliase erstellen, die auf sie verweisen
- Sie für das Löschen planen

Sie können auch Amazon CloudTrail CloudWatch Logs verwenden, um die Anfragen zu verfolgen AWS IoT FleetWise sendet an AWS KMS in Ihrem Namen.

Wenn Sie vom Kunden verwaltete Schlüssel verwenden, müssen Sie Folgendes gewähren AWS FleetWise IoT-Zugriff auf den in Ihrem Konto gespeicherten KMS Schlüssel. AWS IoT FleetWise verwendet Envelope-Verschlüsselung und Schlüsselhierarchie, um Daten zu verschlüsseln. Ihre AWS KMS Der Verschlüsselungsschlüssel wird verwendet, um den Stammschlüssel dieser Schlüsselhierarchie zu verschlüsseln. Weitere Informationen finden Sie unter [Envelope-Verschlüsselung](#) im AWS Key Management Service Entwicklerhandbuch.

Das folgende Beispiel für Policy-Grants AWS FleetWise IoT-Berechtigungen zur Erstellung eines vom Kunden verwalteten Schlüssels in Ihrem Namen.

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

**⚠ Important**

Wenn Sie die neuen Abschnitte zu Ihrer KMS wichtigsten Richtlinie hinzufügen, ändern Sie keine vorhandenen Abschnitte in der Richtlinie. AWS IoT FleetWise kann keine Operationen mit Ihren Daten ausführen, wenn die Verschlüsselung aktiviert ist für AWS IoT FleetWise und alle der folgenden Punkte sind wahr:

- Der KMS Schlüssel ist deaktiviert oder gelöscht.
- Die KMS Schlüsselrichtlinie ist für den Dienst nicht richtig konfiguriert.

**Verwendung von Bildverarbeitungssystemdaten mit Verschlüsselung im Ruhezustand****📘 Note**

Die Daten des Bildverarbeitungssystems befinden sich in der Vorschauversion und können sich ändern.

Wenn Sie die vom Kunden verwaltete Verschlüsselung mit AWS KMS Schlüssel sind auf Ihrem aktiviert AWS Wenn Sie FleetWise ein IoT-Konto haben und Bildverarbeitungssystemdaten verwenden möchten, setzen Sie Ihre Verschlüsselungseinstellungen zurück, damit sie mit komplexen Datentypen kompatibel sind. Das ermöglicht AWS IoT FleetWise zur Einrichtung zusätzlicher Genehmigungen, die für Bildverarbeitungssystemdaten erforderlich sind.

**📘 Note**

Ihr Decoder-Manifest befindet sich möglicherweise in einem Validierungsstatus, wenn Sie Ihre Verschlüsselungseinstellungen für Bildverarbeitungssystemdaten nicht zurückgesetzt haben.

1. Verwenden Sie den [GetEncryptionConfiguration](#) APIVorgang, um zu überprüfen, ob AWS KMS Verschlüsselung ist aktiviert. Wenn der Verschlüsselungstyp ist, sind keine weiteren Maßnahmen erforderlich `FLEETWISE_DEFAULT_ENCRYPTION`.
2. Wenn der Verschlüsselungstyp ist `KMS_BASED_ENCRYPTION`, verwenden Sie den [PutEncryptionConfiguration](#) APIVorgang, um den Verschlüsselungstyp auf zurückzusetzen `FLEETWISE_DEFAULT_ENCRYPTION`.

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

3. Verwenden Sie den [PutEncryptionConfiguration](#) API-Vorgang, um den Verschlüsselungstyp wieder zu `KMS_BASED_ENCRYPTION` aktivieren.

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

Weitere Informationen zum Aktivieren der Verschlüsselung finden Sie unter [Schlüsselverwaltung](#).

## Schlüsselverwaltung

### AWS FleetWise IoT-Cloud-Schlüsselverwaltung

Standardmäßig AWS FleetWise IoT-Anwendungen von AWS verwaltete Schlüssel zum Schutz Ihrer Daten in der AWS Cloud. Sie können Ihre Einstellungen aktualisieren, um Daten mit einem vom Kunden verwalteten Schlüssel zu verschlüsseln AWS IoT FleetWise. Sie können Ihren Verschlüsselungsschlüssel erstellen, verwalten und einsehen über AWS Key Management Service (AWS KMS).

AWS IoT FleetWise unterstützt serverseitige Verschlüsselung mit vom Kunden verwalteten Schlüsseln, die gespeichert sind in AWS KMS um Daten für die folgenden Ressourcen zu verschlüsseln.

| AWS FleetWise IoT-Ressource | Datentyp | Felder, die im Ruhezustand mit vom Kunden verwalteten Schlüsseln verschlüsselt sind |
|-----------------------------|----------|-------------------------------------------------------------------------------------|
| Signalkatalog               |          | description                                                                         |
|                             | Attribut | Beschreibung, allowedValues, defaultValue, min, max                                 |

|                              |                                |                                                                                                                           |
|------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| AWS FleetWise IoT-Ressource  | Datentyp                       | Felder, die im Ruhezustand mit vom Kunden verwalteten Schlüsseln verschlüsselt sind                                       |
|                              | Aktuator                       | BeschreibungallowedValues, min, max                                                                                       |
|                              | Sensor                         | BeschreibungallowedValues, min, max                                                                                       |
| Fahrzeugmodell (Modellliste) |                                | description                                                                                                               |
| Decoder-Manifest             |                                | description                                                                                                               |
|                              | CanInterface                   | protocolName, protocolVersion                                                                                             |
|                              | ObdInterface                   | requestMessageId, dtcRequestInterval Sekunden, hasTransmissionEcu, pidRequestInterval SekundenobdStandard, useExtendedIds |
|                              | CanSignal                      | Faktor, isBigEndianSigned, LängemessageId, Offset, startBit                                                               |
|                              | ObdSignal                      | byteLength, Offset, PID pidResponseLength, Skalierung, serviceMode, startByte, bitMaskLength, bitRightShift               |
| Fahrzeug                     |                                | Attribute                                                                                                                 |
| Kampagne                     |                                | description                                                                                                               |
|                              | conditionBasedCollectionSchema | Ausdruck conditionLanguageVersion, minimumTriggerInterval Frau, triggerMode                                               |

|                             |                           |                                                                                     |
|-----------------------------|---------------------------|-------------------------------------------------------------------------------------|
| AWS FleetWise IoT-Ressource | Datentyp                  | Felder, die im Ruhezustand mit vom Kunden verwalteten Schlüsseln verschlüsselt sind |
|                             | TimeBasedCollectionScheme | periodMs                                                                            |

 Note

Andere Daten und Ressourcen werden mit der Standardverschlüsselung verschlüsselt, wobei die Schlüssel verwaltet werden von AWS IoT FleetWise. Dieser Schlüssel wird erstellt und gespeichert im AWS FleetWise IoT-Konto.

Weitere Informationen finden Sie unter [Was ist AWS Key Management Service?](#) in der AWS Key Management Service Leitfaden für Entwickler.

Aktivieren Sie die Verschlüsselung mit KMS Schlüsseln (Konsole)

Um vom Kunden verwaltete Schlüssel zu verwenden mit AWS IoT FleetWise, Sie müssen Ihre aktualisieren AWS FleetWise IoT-Einstellungen.

Um die Verschlüsselung mit KMS Schlüsseln zu aktivieren (Konsole)

1. Öffnen Sie [AWS FleetWise IoT-Konsole](#).
2. Navigieren Sie zu Einstellungen.
3. Wählen Sie unter Verschlüsselung die Option Bearbeiten aus, um die Seite Verschlüsselung bearbeiten zu öffnen.
4. Wählen Sie als Typ des Verschlüsselungsschlüssels die Option Anderes auswählen AWS KMS Schlüssel. Dies ermöglicht die Verschlüsselung mit vom Kunden verwalteten Schlüsseln, die in gespeichert sind AWS KMS.

 Note

Sie können die vom Kunden verwaltete Schlüsselverschlüsselung nur verwenden für AWS FleetWise IoT-Ressourcen. Dazu gehören der Signalkatalog, das Fahrzeugmodell (Modellmanifest), das Decoder-Manifest, das Fahrzeug, die Flotte und die Kampagne.

## 5. Wählen Sie Ihren KMS Schlüssel mit einer der folgenden Optionen:

- Um einen vorhandenen KMS Schlüssel zu verwenden — Wählen Sie Ihren KMS Schlüsselalias aus der Liste aus.
- Um einen neuen KMS Schlüssel zu erstellen — Wählen Sie `Create an AWS KMS Schlüssel`.

### Note

Dies öffnet die AWS KMS console. Weitere Informationen zum Erstellen eines KMS Schlüssels finden Sie unter [Schlüssel erstellen](#) im AWS Key Management Service Leitfaden für Entwickler.

## 6. Wählen Sie Speichern, um Ihre Einstellungen zu aktualisieren.

Aktivieren Sie die Verschlüsselung mit KMS Schlüsseln (AWS CLI)

Sie können den [PutEncryptionConfiguration](#) API-Vorgang verwenden, um die Verschlüsselung für Ihre zu aktivieren AWS FleetWise IoT-Konto. Das folgende Beispiel verwendet AWS CLI.

Führen Sie den folgenden Befehl aus, um die Verschlüsselung zu aktivieren.

- Ersetzen *KMS key id* mit der ID des KMS Schlüssels.

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

### Example response

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

### KMSwichtige Richtlinie

Nachdem Sie einen KMS Schlüssel erstellt haben, müssen Sie Ihrer KMS Schlüsselrichtlinie mindestens die folgende Anweisung hinzufügen, damit er funktioniert AWS IoT FleetWise.

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

Weitere Informationen zum Bearbeiten einer KMS wichtigen Richtlinie zur Verwendung mit AWS IoT FleetWise, siehe [Änderung einer wichtigen Richtlinie](#) in der AWS Key Management Service Leitfadens für Entwickler.

#### Important

Wenn Sie die neuen Abschnitte zu Ihrer KMS wichtigsten Richtlinie hinzufügen, ändern Sie keine vorhandenen Abschnitte in der Richtlinie. AWS IoT FleetWise kann keine Operationen mit Ihren Daten ausführen, wenn die Verschlüsselung aktiviert ist für AWS IoT FleetWise und alle der folgenden Punkte sind wahr:

- Der KMS Schlüssel ist deaktiviert oder gelöscht.
- Die KMS Schlüsselrichtlinie ist für den Dienst nicht richtig konfiguriert.

## Steuerung des Zugriffs mit AWS IoT FleetWise

In den folgenden Abschnitten wird beschrieben, wie Sie den Zugriff auf und von Ihrem steuern können AWS IoT FleetWise Ressourcen schätzen. Zu den Informationen, die sie behandeln, gehört, wie Sie Ihrer Anwendung Zugriff gewähren können AWS IoT FleetWise kann Fahrzeugdaten während Kampagnen übertragen. Sie beschreiben auch, wie Sie gewähren können AWS IoT FleetWise Zugriff

auf Ihren Amazon S3 (S3) -Bucket oder Ihre Amazon Timestream Timestream-Datenbank und -Tabelle zum Speichern von Daten.

Die Technologie für die Verwaltung all dieser Zugriffsformen ist AWS Identity and Access Management (IAM). Weitere Informationen zu IAM finden Sie unter [Was ist IAM?](#) .

## Inhalt

- [Gewährung AWS IoT FleetWise Zugriff auf ein Amazon S3 S3-Ziel](#)
- [Gewährung AWS IoT FleetWise Zugriff auf ein Amazon Timestream Timestream-Ziel](#)

## Gewährung AWS IoT FleetWise Zugriff auf ein Amazon S3 S3-Ziel

Wenn Sie ein Amazon S3 S3-Ziel verwenden, AWS IoT FleetWise liefert Fahrzeugdaten an Ihren S3-Bucket und kann optional eine verwenden AWS KMS Schlüssel, den Sie für die Datenverschlüsselung besitzen. Wenn die Fehlerprotokollierung aktiviert ist, AWS IoT FleetWise sendet auch Fehler bei der Datenübermittlung an Ihre CloudWatch Protokollgruppe und Ihre Streams. Sie müssen eine IAM Rolle spielen, wenn Sie einen Lieferstream erstellen.

AWS IoT FleetWise verwendet eine Bucket-Richtlinie mit dem Service Principal für das S3-Ziel. Weitere Informationen zum Hinzufügen von Bucket-Richtlinien finden Sie unter [Hinzufügen einer Bucket-Richtlinie mithilfe der Amazon S3 S3-Konsole](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Verwenden Sie zur Aktivierung die folgende Zugriffsrichtlinie AWS IoT FleetWise um auf Ihren S3-Bucket zuzugreifen. Wenn Sie nicht Eigentümer des S3-Buckets sind, fügen Sie `s3:PutObjectACL` der Liste der Amazon-S3-Aktionen hinzu. Dies gewährt dem Bucket-Besitzer vollen Zugriff auf die Objekte, die bereitgestellt werden von AWS IoT FleetWise. Weitere Informationen darüber, wie Sie den Zugriff auf Objekte in Ihren Buckets sichern können, finden Sie unter [Beispiele für Bucket-Richtlinien](#) im Amazon Simple Storage Service-Benutzerhandbuch.

```
{
  "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"
    }
  }
}
]
}

```

Die folgende Bucket-Richtlinie gilt für alle Kampagnen in einem Konto in einem AWS Region.

```

{
  "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"
      }
    }
  }
]
}

```

Wenn Sie einen KMS Schlüssel an Ihren S3-Bucket angehängt haben, benötigt der Schlüssel die folgende Richtlinie. Informationen zur Schlüsselverwaltung finden Sie unter [Schutz von Daten mithilfe serverseitiger Verschlüsselung mit AWS Key Management Service Schlüssel \(SSE-KMS\)](#) im Amazon Simple Storage Service-Benutzerhandbuch.

```

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

```

**⚠ Important**

Wenn Sie einen Bucket erstellen, erstellt S3 eine Standard-Zugriffskontrollliste (ACL), die dem Eigentümer der Ressource die volle Kontrolle über die Ressource gewährt. Wenn AWS IoT FleetWise kann keine Daten an S3 liefern. Stellen Sie sicher, dass Sie das ACL auf dem S3-Bucket deaktivieren. Weitere Informationen finden Sie unter [Deaktivierung ACLs für alle neuen Buckets und Durchsetzung des Objektbesitzes](#) im Amazon Simple Storage Service-Benutzerhandbuch.

## Gewährung AWS IoT FleetWise Zugriff auf ein Amazon Timestream Timestream-Ziel

Wenn Sie ein Timestream-Ziel verwenden, AWS IoT FleetWise liefert Fahrzeugdaten an eine Timestream-Tabelle. Sie müssen die Richtlinien an die IAM Rolle anhängen, um dies zuzulassen AWS IoT FleetWise um Daten an Timestream zu senden.

Wenn Sie die Konsole verwenden, um [eine Kampagne zu erstellen](#), AWS IoT fügt der Rolle FleetWise automatisch die erforderliche Richtlinie hinzu.

Bevor Sie beginnen, überprüfen Sie Folgendes:

**⚠ Important**

- Sie müssen dasselbe verwenden AWS Region, für die Sie Timestream-Ressourcen erstellen AWS IoT FleetWise. Wenn du wechselst AWS Regionen, möglicherweise haben Sie Probleme beim Zugriff auf die Timestream-Ressourcen.
  - AWS IoT FleetWise ist in den USA Ost (Nord-Virginia) und Europa (Frankfurt) verfügbar.
  - Eine Liste der unterstützten Regionen finden Sie unter [Timestream-Endpunkte und Kontingente](#) im Allgemeine AWS-Referenz.
- 
- Sie benötigen eine Timestream-Datenbank. Ein Tutorial finden Sie unter [Create a database](#) im Amazon Timestream Developer Guide.
  - Sie müssen eine Tabelle in der angegebenen Timestream-Datenbank erstellt haben. Ein Tutorial finden Sie unter [Erstellen einer Tabelle](#) im Amazon Timestream Developer Guide.

Sie können das AWS CLI um eine IAM Rolle mit einer Vertrauensrichtlinie für Timestream zu erstellen. Führen Sie den folgenden Befehl aus, um eine IAM Rolle zu erstellen.

Um eine IAM Rolle mit einer Vertrauensrichtlinie zu erstellen

- Ersetzen *TimestreamExecutionRole* mit dem Namen der Rolle, die Sie erstellen.
- Ersetzen *trust-policy* mit der JSON Datei, die die Vertrauensrichtlinie enthält.

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

Erstellen Sie eine Berechtigungsrichtlinie, die Sie weitergeben möchten AWS FleetWise IoT-Berechtigungen zum Schreiben von Daten in Timestream. Führen Sie den folgenden Befehl aus, um eine Berechtigungsrichtlinie zu erstellen.

## Um eine Berechtigungsrichtlinie zu erstellen

- Ersetzen *AWSIoT FleetwiseAccessTimestreamPermissionsPolicy* mit dem Namen der Richtlinie, die Sie erstellen.
- Ersetzen *permissions-policy* mit dem Namen der JSON Datei, die die Berechtigungsrichtlinie enthält.

```
aws iam create-policy --policy-name AWSIoT FleetwiseAccessTimestreamPermissionsPolicy --  
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": "*"   
    }  
  ]  
}
```

## Um die Berechtigungsrichtlinie an Ihre IAM Rolle anzuhängen

1. Kopieren Sie aus der Ausgabe den Amazon-Ressourcennamen (ARN) der Berechtigungsrichtlinie.

2. Führen Sie den folgenden Befehl aus, um die IAM Berechtigungsrichtlinie an Ihre IAM Rolle anzuhängen.
  - Ersetzen *permissions-policy-arn* mit demARN, den Sie im vorherigen Schritt kopiert haben.
  - Ersetzen *TimestreamExecutionRole* mit dem Namen der IAM Rolle, die Sie erstellt haben.

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

Weitere Informationen finden Sie unter [Zugriffsverwaltung für AWS Ressourcen](#) im IAMBenutzerhandbuch.

## Identity and Access Management für AWS IoT FleetWise

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft einem Administrator, den Zugriff auf sicher zu kontrollieren AWS Ressourcen schützen. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und zur Nutzung autorisiert werden kann (über Berechtigungen verfügt) AWS FleetWise IoT-Ressourcen. IAMist ein AWS-Service das Sie ohne zusätzliche Kosten nutzen können.

### Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie AWS IoT FleetWise funktioniert mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)
- [Fehlerbehebung AWS FleetWise IoT-Identität und -Zugriff](#)

### Zielgruppe

Wie benutzt du AWS Identity and Access Management (IAM) unterscheidet sich je nach der Arbeit, die Sie in AWS IoT FleetWise.

**Servicebenutzer** — Wenn Sie den AWS FleetWise IoT-Service, um Ihre Arbeit zu erledigen. Anschließend stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Je mehr Sie verwenden AWS FleetWise IoT-Funktionen Um Ihre Arbeit zu erledigen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in nicht auf eine Funktion zugreifen können AWS IoT FleetWise, siehe [Fehlerbehebung AWS FleetWise IoT-Identität und -Zugriff](#).

**Serviceadministrator** — Wenn Sie verantwortlich sind für AWS FleetWise IoT-Ressourcen in Ihrem Unternehmen, auf die Sie wahrscheinlich vollen Zugriff haben AWS IoT FleetWise. Es ist Ihre Aufgabe, herauszufinden, welche AWS FleetWise IoT-Funktionen und Ressourcen, auf die Ihre Servicebenutzer zugreifen sollten. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Dienstbenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehen IAM. Um mehr darüber zu erfahren, wie Ihr Unternehmen Folgendes nutzen IAM kann AWS IoT FleetWise, siehe [Wie AWS IoT FleetWise funktioniert mit IAM](#).

**IAM Administrator** — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf AWS IoT FleetWise. Um ein Beispiel anzusehen AWS FleetWise Identitätsbasierte IoT-Richtlinien, die Sie verwenden können IAM, finden Sie unter. [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich anmelden AWS mit Ihren Identitätsdaten. Sie müssen authentifiziert (angemeldet) sein AWS) als Root-Benutzer des AWS-Kontos, als IAM Benutzer oder indem Sie eine IAM Rolle übernehmen.

Sie können sich anmelden bei AWS als föderierte Identität mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) - Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie darauf zugreifen AWS Wenn Sie den Verbund verwenden, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der anmelden AWS Management Console oder das AWS Zugangportal. Weitere Informationen zur Anmeldung bei AWS, siehe [So melden Sie sich bei Ihrem an AWS-Konto](#) in der AWS-Anmeldung Benutzerleitfaden.

Wenn Sie darauf zugreifen AWS programmatisch AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie es nicht verwenden AWS Tools, Sie müssen Anfragen selbst unterschreiben. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie unter [Signieren AWS APIAnfragen](#) im IAMBenutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. Zum Beispiel AWS empfiehlt, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center Benutzerhandbuch und [Verwendung der Multi-Faktor-Authentifizierung \(\) MFA in AWS](#) im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie eine erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle hat AWS-Services und Ressourcen im Konto. Diese Identität wird als AWS-Konto Root-Benutzer. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

## Verbundidentität

Es hat sich bewährt, menschlichen Benutzern, einschließlich Benutzern, die Administratorzugriff benötigen, vorzuschreiben, für den Zugriff den Verbund mit einem Identitätsanbieter zu verwenden AWS-Services mithilfe temporärer Anmeldeinformationen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, ein Web-Identitätsanbieter, der AWS Directory Service, das Identity Center-Verzeichnis oder ein beliebiger Benutzer, der zugreift AWS-Services mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für eine zentralisierte Zugriffsverwaltung empfehlen wir die Verwendung AWS IAM Identity Center. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und

diese synchronisieren, um sie in all Ihren AWS-Konten und Anwendungen. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) in der AWS IAM Identity Center Benutzerleitfaden.

## IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres AWS-Konto das über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind](#).

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Wann sollte ein IAM Benutzer \(statt einer Rolle\) erstellt werden?](#) im IAMBenutzerhandbuch.

## IAMRollen

Eine [IAMRolle](#) ist eine Identität in deinem AWS-Konto das hat spezifische Berechtigungen. Es ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend eine IAM Rolle in der übernehmen AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie einen anrufen AWS CLI or AWS API Operation oder mithilfe eines benutzerdefinierten URL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie [unter Verwenden von IAM Rollen](#) im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert,

so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center Benutzerleitfaden.

- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Kontoübergreifender Zugriff — Sie können eine IAM Rolle verwenden, um jemandem (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Allerdings mit einigen AWS-Services, Sie können eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- Serviceübergreifender Zugriff — Einige AWS-Services Funktionen in anderen verwenden AWS-Services. Wenn Sie beispielsweise einen Service aufrufen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, Sie gelten als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- Servicerolle — Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.

- **Dienstverknüpfte Rolle** — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon laufen EC2** — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS CLI or AWS APIAnfragen. Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instanz vorzuziehen. Um eine zuzuweisen AWS Sie erstellen ein EC2 Instanzprofil, das an die Instanz angehängt ist. Sie müssen einer Instanz eine Rolle zuweisen und sie allen ihren Anwendungen zur Verfügung stellen. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden [Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle \(anstelle eines IAM Benutzers\) erstellt](#) werden.

## Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff in AWS indem Sie Richtlinien erstellen und diese anhängen AWS Identitäten oder Ressourcen. Eine Richtlinie ist ein Objekt in AWS das, wenn es einer Identität oder Ressource zugeordnet ist, ihre Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien sind gespeichert in AWS als JSON Dokumente. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können Folgendes verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM

Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAM Richtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der abrufen AWS Management Console, der AWS CLI, oder der AWS API.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAM Benutzerhandbuch unter [Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können nicht verwenden AWS verwaltete Richtlinien aus IAM einer ressourcenbasierten Richtlinie.

## Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3, AWS WAF, und Amazon VPC sind Beispiele für Dienste, die unterstützen ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAM Benutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAM Benutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten den Ihr Unternehmen besitzt. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und finden Sie SCPs unter [Richtlinien zur Servicesteuerung](#) in der AWS Organizations Benutzerleitfaden.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und

der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Um zu erfahren, wie AWS bestimmt, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, siehe [Bewertungslogik für Richtlinien](#) im IAMBenutzerhandbuch.

## Wie AWS IoT FleetWise funktioniert mit IAM

Vor der Verwendung IAM zur Verwaltung des Zugriffs auf AWS IoT FleetWise, erfahren Sie, welche IAM Funktionen zur Verwendung mit verfügbar sind AWS IoT FleetWise.

IAMFunktionen, mit denen Sie arbeiten können AWS IoT FleetWise

| IAMMerkmal                                             | AWS FleetWise IoT-Unterstützung |
|--------------------------------------------------------|---------------------------------|
| <a href="#">Identitätsbasierte Richtlinien</a>         | Ja                              |
| <a href="#">Ressourcenbasierte Richtlinien</a>         | Nein                            |
| <a href="#">Richtlinienaktionen</a>                    | Ja                              |
| <a href="#">Richtlinienressourcen</a>                  | Ja                              |
| <a href="#">Bedingungsschlüssel für die Richtlinie</a> | Ja                              |
| <a href="#">ACLs</a>                                   | Nein                            |
| <a href="#">ABAC(Tags in Richtlinien)</a>              | Teilweise                       |
| <a href="#">Temporäre Anmeldeinformationen</a>         | Ja                              |
| <a href="#">Hauptberechtigungen</a>                    | Ja                              |
| <a href="#">Servicerollen</a>                          | Nein                            |

| IAMMerkmal                               | AWS FleetWise IoT-Unterstützung |
|------------------------------------------|---------------------------------|
| <a href="#">Serviceverknüpfte Rollen</a> | Nein                            |

Um einen allgemeinen Überblick darüber zu erhalten, wie AWS IoT FleetWise und andere AWS Dienste funktionieren mit den meisten IAM Funktionen, siehe [AWS Dienste, mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

## Identitätsbasierte Richtlinien für AWS IoT FleetWise

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAMRichtlinien erstellen im Benutzerhandbuch](#). IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAMBenutzerhandbuch unter [Referenz zu IAM JSON Richtlinienelementen](#).

Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise

Um Beispiele zu sehen AWS FleetWise Identitätsbasierte IoT-Richtlinien, siehe. [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)

## Ressourcenbasierte Richtlinien innerhalb AWS IoT FleetWise

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu

steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Erlaubnis erteilen, auf die Ressource zuzugreifen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie [IAMim IAMBenutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#).

## Politische Maßnahmen für AWS IoT FleetWise

Unterstützt Richtlinienaktionen: Ja

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörigen AWS API-Betrieb. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur Berechtigungen erforderlich sind und für die es keine entsprechende Operation gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Um eine Liste von zu sehen AWS FleetWise IoT-Aktionen, siehe [Aktionen definiert von AWS IoT FleetWise](#) in der Serviceautorisierungsreferenz.

Politische Maßnahmen in AWS IoT FleetWise verwendet vor der Aktion das folgende Präfix:

```
iotfleetwise
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

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

Um Beispiele zu sehen AWS FleetWise Identitätsbasierte IoT-Richtlinien, siehe. [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)

## Politische Ressourcen für AWS IoT FleetWise

Unterstützt Richtlinienressourcen: Ja

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem [Amazon-Ressourcennamen \(ARN\)](#) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Um eine Liste von zu sehen AWS FleetWise IoT-Ressourcentypen und ihre ARNs, siehe [Ressourcen definiert von AWS IoT FleetWise](#) in der Serviceautorisierungsreferenz. Informationen zu den Aktionen, mit denen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Aktionen definiert durch AWS IoT FleetWise](#).

Um Beispiele zu sehen AWS FleetWise Identitätsbasierte IoT-Richtlinien, siehe. [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)

## Schlüssel für die Bedingungen der Richtlinien für AWS IoT FleetWise

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können verwenden AWS JSONRichtlinien, um festzulegen, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition` Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition` Element angeben, AWS wertet sie mithilfe einer logischen AND Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem IAM Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinienelemente: Variablen und Tags](#).

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Um alle zu sehen AWS globale Bedingungsschlüssel finden Sie unter [AWS Kontexttasten für globale Bedingungen](#) im IAMBenutzerhandbuch.

Um eine Liste von zu sehen AWS FleetWise IoT-Bedingungsschlüssel, siehe [Bedingungsschlüssel für AWS IoT FleetWise](#) in der Serviceautorisierungsreferenz. Informationen zu den Aktionen und

Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen, die definiert sind von AWS IoT FleetWise](#).

Um Beispiele zu sehen AWS FleetWise Identitätsbasierte IoT-Richtlinien, siehe. [Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise](#)

## Zugriffskontrolllisten (ACLs) in AWS IoT FleetWise

UnterstütztACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

## Attributbasierte Zugriffskontrolle (ABAC) mit ABAC AWS IoT FleetWise

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen auf der Grundlage von Attributen definiert werden. In AWS, diese Attribute werden Tags genannt. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele andere anhängen AWS Ressourcen schätzen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABAC ist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu finden Sie ABAC unter [Was ist? ABAC](#) im IAM Benutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter [Verwenden der attributbasierten Zugriffskontrolle \(ABAC\)](#). IAM

**Note**

AWS IoT unterstützt FleetWise `nuriam:PassRole`, was für den `CreateCampaign` API Betrieb erforderlich ist.

## Temporäre Anmeldeinformationen verwenden mit AWS IoT FleetWise

Unterstützt temporäre Anmeldeinformationen: Ja

Etwas AWS-Services funktioniert nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Für zusätzliche Informationen, einschließlich welcher AWS-Services mit temporären Anmeldeinformationen arbeiten, finden Sie unter [AWS-Services mit denen IAM](#) im IAMBenutzerhandbuch gearbeitet werden kann.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich bei der AWS Management Console mit einer beliebigen Methode außer einem Benutzernamen und einem Passwort. Zum Beispiel, wenn Sie darauf zugreifen AWS Wenn Sie den Single Sign-On-Link (SSO) Ihres Unternehmens verwenden, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAMBenutzerhandbuch.

Sie können temporäre Anmeldeinformationen manuell erstellen, indem Sie AWS CLI or AWS API. Sie können dann diese temporären Anmeldeinformationen für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen unter IAM](#).

## Serviceübergreifende Prinzipalberechtigungen für AWS IoT FleetWise

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen in AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anfrage AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, die Interaktionen mit anderen erfordert AWS-Services oder zu vervollständigende Ressourcen. In diesem Fall müssen Sie über

Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

## Servicerollen für AWS IoT FleetWise

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschen IAM. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an ein AWS-Service](#) im IAM-Benutzerhandbuch.

### Warning

Das Ändern der Berechtigungen für eine Servicerolle kann fehlerhaft sein AWS FleetWise IoT-Funktionalität. Bearbeiten Sie Servicerollen nur, wenn AWS IoT FleetWise bietet dazu Anleitungen.

## Mit Diensten verknüpfte Rollen für AWS IoT FleetWise

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen durchzuführen. Mit Diensten verknüpfte Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie unter [AWS Dienste, die mit IAM](#) funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

## Verwenden von serviceverknüpften Rollen für das IoT AWS FleetWise

AWS FleetWise IoT-Anwendungen AWS Identity and Access Management (IAM) [Rollen, die mit Diensten verknüpft sind](#). Eine dienstbezogene Rolle ist ein einzigartiger IAM Rollentyp, der direkt mit dem AWS IoT FleetWise verknüpft ist. Servicebezogene Rollen sind von AWS IoT vordefiniert FleetWise und beinhalten die Berechtigungen, die AWS IoT FleetWise benötigt, um Metriken an

Amazon CloudWatch zu senden. Weitere Informationen finden Sie unter [Überwachung des AWS IoT FleetWise mit Amazon CloudWatch](#).

Eine serviceverknüpfte Rolle FleetWise beschleunigt die Einrichtung von AWS IoT, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS IoT FleetWise definiert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, FleetWise kann nur AWS IoT seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keiner anderen IAM Entität zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre AWS FleetWise IoT-Ressourcen, da Sie die Zugriffsberechtigung für die Ressourcen nicht versehentlich entfernen können.

Informationen zu anderen Diensten, die dienstbezogene Rollen unterstützen, finden Sie unter [AWS Dienste, die mit Diensten funktionieren IAM](#), und suchen Sie nach den Diensten, für die in der Spalte mit dienstverknüpften Rollen Ja angegeben ist. Um die serviceverknüpfte Rollendokumentation für diesen Service anzuzeigen, wählen Sie ein Ja mit einem Link.

Dienstbezogene Rollenberechtigungen für IoT AWS FleetWise

AWS IoT FleetWise verwendet die serviceverknüpfte Rolle mit dem Namen `AWSServiceRoleForIoT FleetWise`— Eine AWS verwaltete Richtlinie, die für alle out-of-the-box Berechtigungen für AWS IoT FleetWise verwendet wird.

Die `AWSServiceRoleForIoT FleetWise` serviceverknüpfte Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `IoT FleetWise`

Die genannte Rollenberechtigungsrichtlinie `AWS IoT FleetWise ServiceRolePolicy` ermöglicht es AWS IoT FleetWise, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `cloudwatch:PutMetricData` auf Ressource: \*

Sie müssen Berechtigungen konfigurieren, damit eine IAM Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) eine dienstbezogene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [Berechtigungen für dienstbezogene Rollen](#) im IAM Benutzerhandbuch.

## Schaffung einer serviceverknüpften Rolle für das IoT AWS FleetWise

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie ein Konto in der AWS FleetWise IoT-Konsole registrieren, AWS CLI, oder das AWS API, AWS IoT FleetWise erstellt die serviceverknüpfte Rolle für Sie. Weitere Informationen finden Sie unter [Einstellungen konfigurieren](#).

## Erstellen einer serviceverknüpften Rolle in AWS IoT FleetWise (Konsole)

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie ein Konto in der AWS FleetWise IoT-Konsole registrieren, AWS CLI, oder das AWS API, AWS IoT FleetWise erstellt die serviceverknüpfte Rolle für Sie.

## Bearbeiten einer serviceverknüpften Rolle für IoT AWS FleetWise

Sie können die `AWSServiceRoleForIoT FleetWise` serviceverknüpfte Rolle in AWS IoT FleetWise nicht bearbeiten. Da verschiedene Entitäten möglicherweise auf jede dienstverknüpfte Rolle verweisen, die Sie erstellen, können Sie den Namen der Rolle nicht ändern. Sie können die Beschreibung der Rolle jedoch bearbeiten, indem Sie IAM Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Bearbeiten einer dienstbezogenen Rolle](#).

## Bereinigen einer serviceverknüpften Rolle

Bevor Sie eine dienstverknüpfte Rolle löschen können, müssen Sie zunächst alle Ressourcen löschen, die von der Rolle verwendet werden. IAM

### Note

Wenn AWS IoT FleetWise die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut. Um zu erfahren, wie Sie die service-linked-role über die Konsole löschen, AWS CLI, oder AWS API, siehe [Verwenden von serviceverknüpften Rollen](#) im IAMBenutzerhandbuch.

Wenn Sie diese dienstverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie ein Konto bei AWS IoT FleetWise registrieren. AWSIoT erstellt FleetWise dann erneut die serviceverknüpfte Rolle für Sie.

## Beispiele für identitätsbasierte Richtlinien für AWS IoT FleetWise

Standardmäßig sind Benutzer und Rollen nicht berechtigt, etwas zu erstellen oder zu ändern AWS FleetWise IoT-Ressourcen. Sie können auch keine Aufgaben mit dem ausführen AWS Management Console, AWS Command Line Interface (AWS CLI), oder AWS API. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie anhand dieser JSON Beispieldokumente finden Sie unter [IAM Richtlinien erstellen](#) im IAM Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, definiert von AWS IoT FleetWise, einschließlich des Formats der ARNs für jeden Ressourcentyp, siehe [Aktionen, Ressourcen und Bedingungsschlüssel für AWS IoT FleetWise](#) in der Serviceautorisierungsreferenz.

### Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwendung der AWS FleetWise IoT-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Greifen Sie auf Ressourcen in Amazon Timestream zu](#)

### Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand etwas erstellen, darauf zugreifen oder löschen kann AWS FleetWise IoT-Ressourcen in Ihrem Konto. Diese Aktionen können Kosten für Sie verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Fangen Sie an mit AWS verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Um zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie AWS verwaltete Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie Folgendes definieren AWS vom Kunden verwaltete Richtlinien, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS verwaltete Richtlinien](#) oder [AWS verwaltete Richtlinien für Jobfunktionen](#) im IAM Benutzerhandbuch.

- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung IAM zum Anwenden von Berechtigungen finden Sie [IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen](#). IAM
- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über eine bestimmte AWS-Service, wie beispielsweise AWS CloudFormation. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente: Zustand](#) im IAMBenutzerhandbuch.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtliniensprache (JSON) und den IAM bewährten Methoden entsprechen. IAMAccess Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie unter [IAMAccess Analyzer-Richtlinienvvalidierung](#) im IAMBenutzerhandbuch.
- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, das IAM Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, schalten Sie MFA für zusätzliche Sicherheit ein. Wenn Sie festlegen möchten, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter [Konfiguration des MFA -geschützten API Zugriffs](#) im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch. IAM

## Verwendung der AWS FleetWiseIoT-Konsole

Um auf die zuzugreifen AWS FleetWise IoT-Konsole, Sie müssen über einen Mindestsatz an Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den aufzulisten und anzuzeigen AWS FleetWise IoT-Ressourcen in Ihrem AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver ist als die erforderlichen Mindestberechtigungen,

funktioniert die Konsole für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie nicht wie vorgesehen.

Sie müssen Benutzern, die nur Anrufe tätigen, keine Mindestberechtigungen für die Konsole gewähren AWS CLI oder das AWS API. Erlauben Sie stattdessen nur den Zugriff auf die Aktionen, die dem API Vorgang entsprechen, den sie ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die weiterhin verwenden können AWS FleetWise IoT-Konsole, schließen Sie auch die AWS IoT FleetWise ConsoleAccess oder ReadOnly AWS verwaltete Richtlinie für die Entitäten. Weitere Informationen finden Sie im [Benutzerhandbuch unter Hinzufügen von Berechtigungen für einen IAM Benutzer](#).

## Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen könnten, die es IAM Benutzern ermöglicht, die internen und verwalteten Richtlinien einzusehen, die mit ihrer Benutzeridentität verknüpft sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe von AWS CLI or AWS API.

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

```

```
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

## Greifen Sie auf Ressourcen in Amazon Timestream zu

Vor der Verwendung AWS IoT FleetWise, Sie müssen Ihre registrieren AWS Konto- und Amazon Timestream Timestream-RessourcenIAM, die gewährt werden sollen AWS FleetWise IoT-Erlaubnis zum Senden von Fahrzeugdaten an AWS Cloud in Ihrem Namen. Um sich zu registrieren, benötigen Sie:

- Eine Amazon Timestream Timestream-Datenbank.
- Eine Tabelle, die in der angegebenen Amazon Timestream Timestream-Datenbank erstellt wurde.
- Eine IAM Rolle, die es ermöglicht AWS IoT FleetWise zum Senden von Daten an Amazon Timestream.

Weitere Informationen, einschließlich Verfahren und Beispielrichtlinien, finden [Sie unter Einstellungen konfigurieren](#).

## Fehlerbehebung AWS FleetWise IoT-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS IoT FleetWise undIAM.

### Themen

- [Ich bin nicht berechtigt, eine Aktion auszuführen in AWS IoT FleetWise](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Leute außerhalb meines AWS-Konto um auf meine zuzugreifen AWS FleetWise IoT-Ressourcen](#)

## Ich bin nicht berechtigt, eine Aktion auszuführen in AWS IoT FleetWise

Wenn das Symbol AWS Management Console teilt Ihnen mit, dass Sie nicht berechtigt sind, eine Aktion auszuführen. Dann müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven `myVehicle` Ressource anzuzeigen, aber nicht über die `iotfleetwise:GetVehicleStatus` entsprechenden Berechtigungen verfügt.

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

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `myVehicle` auf die Ressource `iotfleetwise:GetVehicleStatus` zugreifen zu können.

## Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht autorisiert sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an übergeben können AWS IoT FleetWise.

Etwas AWS-Services ermöglicht es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion auszuführen in AWS IoT FleetWise. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

## Ich möchte Leute außerhalb meines AWS-Konto um auf meine zuzugreifen AWS FleetWise IoT-Ressourcen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Um zu erfahren, ob AWS IoT FleetWise unterstützt diese Funktionen, siehe [Wie AWS IoT FleetWise funktioniert mit IAM](#).
- Um zu erfahren, wie Sie Zugriff auf Ihre Ressourcen in allen Bereichen gewähren können AWS-Konten die Ihnen gehören, finden Sie unter [Gewähren des Zugriffs für einen IAM Benutzer in einem anderen AWS-Konto die Sie besitzen, finden Sie](#) im IAM Benutzerhandbuch.
- Um zu erfahren, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, siehe Zugriff [gewähren auf AWS-Konten Eigentum Dritter](#) im IAM Benutzerhandbuch.
- Informationen zur [Bereitstellung des Zugriffs über einen Identitätsverbund finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAM im Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM

## Konformitätsprüfung für AWS IoT FleetWise

### Note

AWS IoT FleetWise fällt nicht in den Geltungsbereich von AWS Compliance-Programme.

Um zu erfahren, ob ein AWS-Service fällt in den Geltungsbereich bestimmter Compliance-Programme, siehe [AWS-Services im Geltungsbereich nach Compliance-Programm](#) Compliance-Programmen das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme](#) .

Sie können Prüfberichte von Drittanbietern herunterladen unter AWS Artifact. Weitere Informationen finden Sie unter Berichte [herunterladen in AWS Artifact](#).

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Verwendung AWS-Services hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS bietet die folgenden Ressourcen zur Unterstützung bei der Einhaltung von Vorschriften:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen auf AWS die sich auf Sicherheit und Compliance konzentrieren.
- [Architektur für HIPAA Sicherheit und Compliance auf Amazon Web Services](#) — Dieses Whitepaper beschreibt, wie Unternehmen Folgendes nutzen können AWS um in Frage kommende Anwendungen zu erstellenHIPAA.

 Note

Nicht alle AWS-Services sind HIPAA berechtigt. Weitere Informationen finden Sie in der [Referenz für HIPAA qualifizierte Dienste](#).

- [AWS Ressourcen zur Einhaltung von Vorschriften](#) — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. Die Leitfäden fassen die bewährten Methoden zur Sicherung zusammen AWS-Services und geben einen Überblick über die Leitlinien für Sicherheitskontrollen in verschiedenen Regelwerken (darunter National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI) und International Organization for Standardization (ISO)).
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Das AWS Config Der Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Das AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS Ressourcen und um Ihre Einhaltung der Sicherheitsstandards und Best Practices der Sicherheitsbranche zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).

- [Amazon GuardDuty](#) — Das AWS-Service erkennt potenzielle Bedrohungen für Ihr AWS-Konten, Workloads, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.
- [AWS Audit Manager](#)— Das AWS-Service hilft Ihnen dabei, Ihre kontinuierlich zu überprüfen AWS Nutzung, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

## Resilienz in AWS IoT FleetWise

Das Tool AWS Die globale Infrastruktur basiert auf AWS Regionen und Verfügbarkeitszonen. Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zur AWS Regionen und Verfügbarkeitszonen, siehe [AWS Globale Infrastruktur](#).

### Note

Daten verarbeitet von AWS IoT FleetWise wird in einer Amazon Timestream Timestream-Datenbank gespeichert. Timestream unterstützt Backups auf andere AWS Verfügbarkeitszonen oder Regionen. Sie können jedoch mithilfe des Timestreams Ihre eigene Anwendung schreiben, SDK um Daten abzufragen und sie an einem Ziel Ihrer Wahl zu speichern.

Weitere Informationen zu Amazon Timestream finden Sie [im Amazon Timestream Developer Guide](#).

## Sicherheit der Infrastruktur in AWS IoT FleetWise

Als verwalteter Dienst AWS IoT FleetWise ist geschützt durch AWS globale Netzwerksicherheit. Für Informationen über AWS Sicherheitsdienste und wie AWS schützt die Infrastruktur, siehe

[AWS Cloud-Sicherheit](#). Um Ihre zu entwerfen AWS Umgebung, in der die besten Methoden für die Infrastruktursicherheit verwendet werden, finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Gut durchdachtes Framework.

Du verwendest AWS veröffentlichte API Aufrufe zum Zugriff AWS IoT FleetWise über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Sicherheit auf Transportebene (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit perfekter Vorwärtsgeheimhaltung (PFS) wie (Ephemeral Diffie-Hellman) oder DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. IAM Oder Sie können das verwenden [AWS Security Token Service](#) (AWS STS), um temporäre Sicherheitsanmeldedaten zum Signieren von Anfragen zu generieren.

Sie können diese API Operationen von jedem Netzwerkstandort aus aufrufen, aber AWS IoT FleetWise unterstützt ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf der Grundlage der Quell-IP-Adresse beinhalten können. Sie können auch Folgendes verwenden AWS FleetWise IoT-Richtlinien zur Steuerung des Zugriffs von bestimmten Amazon Virtual Private Cloud (AmazonVPC) -Endpunkten oder bestimmtenVPCs. Dadurch wird der Netzwerkzugriff auf ein bestimmtes Objekt effektiv isoliert AWS FleetWise IoT-Ressource nur aus dem Spezifischen VPC innerhalb der AWS Netzwerk.

Themen

- [Verbindung herstellen zu AWS IoT FleetWise über einen VPC Schnittstellenendpunkt](#)

## Verbindung herstellen zu AWS IoT FleetWise über einen VPC Schnittstellenendpunkt

Sie können eine direkte Verbindung herstellen zu AWS IoT FleetWise mithilfe eines [VPC Schnittstellenendpunkts \(AWS PrivateLink\)](#) in Ihrer Virtual Private Cloud (VPC), anstatt eine Verbindung über das Internet herzustellen. Wenn Sie einen VPC Schnittstellenendpunkt verwenden, erfolgt die Kommunikation zwischen Ihrem VPC und AWS IoT FleetWise wird vollständig innerhalb der AWS Netzwerk. Jeder VPC Endpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) (ENIs) mit privaten IP-Adressen in Ihren VPC Subnetzen repräsentiert.

Der VPC Schnittstellen-Endpunkt verbindet Sie VPC direkt mit AWS IoT FleetWise ohne Internet-Gateway, NAT Gerät, VPN Verbindung oder AWS Direct Connect Verbindung. Die Instanzen in Ihrem benötigen VPC keine öffentlichen IP-Adressen, um mit dem zu kommunizieren AWS IoT FleetWiseAPI.

Zur Verwendung AWS IoT FleetWise über IhrVPC, Sie müssen eine Verbindung von einer Instanz herstellen, die sich innerhalb des befindet, VPC oder Ihr privates Netzwerk mit Ihrem verbinden, VPC indem Sie eine AWS Virtual Private Network (VPN) oder AWS Direct Connect. Informationen zu Amazon VPN finden Sie unter [VPNVerbindungen](#) im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch. Für Informationen über AWS Direct Connect, siehe [Eine Verbindung herstellen](#) in der AWS Direct Connect Benutzerleitfaden.

Sie können einen VPC Schnittstellen-Endpunkt erstellen, mit dem Sie eine Verbindung herstellen können AWS IoT FleetWise mithilfe der AWS Konsole oder AWS Command Line Interface (AWS CLI) Befehle. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#).

Wenn Sie nach dem Erstellen eines VPC Schnittstellenendpunkts private DNS Hostnamen für den Endpunkt aktivieren, ist der Standard AWS Der FleetWise IoT-Endpunkt wird zu Ihrem VPC Endpunkt aufgelöst. Der standardmäßige Endpunkt für den Dienstenamen für AWS IoT FleetWise hat das folgende Format.

```
iotfleetwise.Region.amazonaws.com
```

Wenn Sie private DNS Hostnamen nicht aktivieren, VPC stellt Amazon einen DNS Endpunktnamen bereit, den Sie im folgenden Format verwenden können.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

Weitere Informationen finden Sie unter [VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) im VPCAmazon-Benutzerhandbuch.

AWS IoT FleetWise unterstützt das Ausführen von Aufrufen all seiner [APIAktionen](#) in IhremVPC.

Sie können VPC Endpunktrichtlinien an einen VPC Endpunkt anhängen, um den Zugriff für IAM Prinzipale zu kontrollieren. Sie können einem VPC Endpunkt auch Sicherheitsgruppen zuordnen, um den eingehenden und ausgehenden Zugriff auf Grundlage des Ursprungs und Ziels des Netzwerkverkehrs zu steuern, z. B. anhand eines Bereichs von IP-Adressen. Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC Endpunkten](#).

## Erstellen einer VPC Endpunktrichtlinie für AWS IoT FleetWise

Sie können eine Richtlinie für VPC Amazon-Endpunkte erstellen für AWS IoT FleetWise , um Folgendes zu spezifizieren:

- Prinzipal, der Aktionen ausführen bzw. nicht ausführen kann
- Die Aktionen, die ausgeführt werden können oder nicht

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC Endpunkten](#) im VPCAmazon-Benutzerhandbuch.

Example — VPC Endpunktrichtlinie zur Verweigerung jeglichen Zugriffs von einem bestimmten AWS Konto

Die folgende VPC Endpunktrichtlinie verweigert AWS Konto **123456789012** alle API Anrufe, die den Endpunkt verwenden.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Example — VPC Endpunktrichtlinie, um VPC den Zugriff nur einem bestimmten IAM Prinzipal (Benutzer) zu ermöglichen

Die folgende VPC Endpunktrichtlinie erlaubt vollen Zugriff nur einem Benutzer *lijuan* in AWS Konto *123456789012*. Es verweigert allen anderen IAM Principals den Zugriff auf den Endpunkt.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/lijuan"
        ]
      }
    }
  ]
}
```

Example — VPC Endpunktrichtlinie für AWS FleetWise IoT-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für AWS IoT FleetWise. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie Zugriff auf die aufgelisteten AWS FleetWise IoT-Aktionen für den IAM Benutzer *fleetWise* in der AWS-Konto *123456789012*.

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/fleetWise"
        ]
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:ListFleets",
        "iotfleetwise:ListCampaigns",
        "iotfleetwise:CreateVehicle",
      ]
    }
  ]
}
```

}

## Konfiguration und Schwachstellenanalyse in AWS IoT FleetWise

IoT-Umgebungen können aus einer großen Anzahl von Geräten mit unterschiedlichsten Funktionen bestehen und sind langlebig und geografisch verteilt. Aufgrund dieser Merkmale ist die Geräteeinrichtung komplex und fehleranfällig. Da Geräte häufig in Bezug auf Rechenleistung, Arbeitsspeicher und Speicherkapazitäten eingeschränkt sind, ist auch der Einsatz von Verschlüsselung und anderen Sicherheitsformen auf den Geräten begrenzt. Geräte verwenden häufig Software mit bekannten Schwachstellen. Diese Faktoren sorgen dafür, dass IoT-Geräte, einschließlich Fahrzeuge, Daten sammeln für AWS IoT FleetWise ist ein attraktives Ziel für Hacker und macht es schwierig, sie dauerhaft zu sichern.

Konfiguration und IT-Steuerung fallen in die gemeinsame Verantwortung von AWS und Sie, unser Kunde. Weitere Informationen finden Sie auf der AWS [Modell der geteilten Verantwortung](#).

## Bewährte Sicherheitsmethoden für AWS IoT FleetWise

AWS IoT FleetWise bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Um mehr über Sicherheit zu erfahren in AWS IoT Weitere Informationen finden Sie unter [Bewährte Sicherheitsmethoden AWS IoT Core](#) in der AWS IoT Leitfaden für Entwickler

## Erteilen von Mindestberechtigungen

Folgen Sie dem Prinzip der geringsten Rechte, indem Sie die Mindestanzahl an Berechtigungen in IAM Rollen verwenden. Beschränken Sie die Verwendung des \* Platzhalters für die Resource Eigenschaften Action und in Ihren IAM Richtlinien. Deklarieren Sie stattdessen, wenn möglich, eine endliche Menge von Aktionen und Ressourcen. Weitere Informationen zu den geringsten Berechtigungen und anderen bewährten Methoden für Richtlinien finden Sie unter [the section called "Bewährte Methoden für Richtlinien"](#).

## Keine Protokollierung sensibler Informationen

Sie sollten die Protokollierung von Anmeldeinformationen und anderen personenbezogenen Daten verhindern (PII). Wir empfehlen Ihnen, die folgenden Sicherheitsvorkehrungen zu treffen:

- Verwenden Sie keine vertraulichen Informationen in Gerätenamen.
- Verwenden Sie keine vertraulichen Informationen in den Namen und IDs von AWS FleetWise IoT-Ressourcen, zum Beispiel im Namen von Kampagnen, Decoder-Manifesten, Fahrzeugmodellen und Signalkatalogen oder im Namen IDs von Fahrzeugen und Flotten.

## Verwenden Sie AWS CloudTrail um die Anrufliste einzusehen API

Sie können eine Historie von anzeigen AWS FleetWise API IoT-Aufrufe, die auf Ihrem Konto für Sicherheitsanalysen und betriebliche Problembehebungen getätigt wurden. Um eine Historie zu erhalten von AWS FleetWise API IoT-Anrufe, die auf Ihrem Konto getätigt wurden, schalten Sie einfach CloudTrail in der AWS Management Console. Weitere Informationen finden Sie unter [the section called "CloudTrail-Protokolle"](#).

## Synchronisieren der internen Uhr Ihres Geräts

Es ist wichtig, dass Sie eine genaue Uhrzeit auf Ihrem Gerät haben. X.509-Zertifikate haben ein Ablaufdatum und eine Ablaufzeit. Die Uhr auf Ihrem Gerät wird verwendet, um sicherzustellen, dass ein Serverzertifikat noch gültig ist. Geräteuhren können im Laufe der Zeit unpräzise werden, oder die Batterien werden entladen.

Weitere Informationen finden Sie unter [Bewährtes Verfahren zur Synchronisation der Geräteuhr im AWS IoT Core Leitfadens für Entwickler](#).

# Überwachung des AWS IoT FleetWise

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS IoT FleetWise und Ihren anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, um das AWS IoT zu beobachten FleetWise, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Metriken erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarmer festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer Amazon EC2 EC2-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Amazon CloudWatch Logs kann verwendet werden, um Ihre Protokolldateien von Amazon EC2 EC2-Instances und anderen Quellen zu überwachen CloudTrail, zu speichern und darauf zuzugreifen. CloudWatch Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihrer AWS-Konto. Anschließend werden die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket übermittelt. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

## Überwachung des AWS IoT FleetWise mit Amazon CloudWatch

Amazon CloudWatch Amazon-Metriken können Sie Ihre AWS Ressourcen und deren Leistung überwachen. AWS IoT FleetWise sendet Metriken an CloudWatch. Sie können die AWS Management Console, oder eine API verwenden AWS CLI, um die Metriken aufzulisten, an die AWS IoT FleetWise sendet CloudWatch. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

**⚠ Important**

Sie müssen die Einstellungen so konfigurieren, dass AWS IoT Metriken an senden FleetWise kann CloudWatch. Weitere Informationen finden Sie unter [Einstellungen konfigurieren](#).

Der AWS/IoTFleetWise-Namespaces enthält die folgenden Metriken.

## Metriken signalisieren

| Kennzahl               | Beschreibung                                                                                                                                                                                                                                                                                                 |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IllegalMessageFromEdge | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht entsprach FleetWise nicht dem erforderlichen Format.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: VehicleName</p> <p>Gültige Statistiken: Summe</p>                                                                                      |
| MessageThrottled       | <p>Eine vom Fahrzeug an das AWS IoT gesendete Nachricht FleetWise wurde gedrosselt. Dies liegt daran, dass Sie die <a href="#">Dienstlimits</a> für dieses Konto in der aktuellen Region überschritten haben.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: VehicleName</p> <p>Gültige Statistiken: Summe</p> |
| ModelingError          | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht FleetWise enthält Signale, die nicht mit dem Fahrzeugmodell verglichen werden können.</p> <p>Einheiten: Anzahl</p>                                                                                                                       |

| Kennzahl      | Beschreibung                                                                                                                                                                                                                                                                |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>Abmessungen: ModelManifestName</p> <p>Gültige Statistiken: Summe</p>                                                                                                                                                                                                     |
| DecodingError | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht FleetWise enthält Signale, die nicht anhand des Decoder-Manifests des Fahrzeugs dekodiert werden können.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: DecoderName</p> <p>Gültige Statistiken: Summe</p> |

## Kampagnenmetriken

| Kennzahl        | Beschreibung                                                                                                                                                                                                            |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VehicleNotFound | <p>Eine Nachricht, die vom Fahrzeug gesendet und vom AWS IoT empfangen wird FleetWise, wobei das Fahrzeug unbekannt ist.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: VehicleName</p> <p>Gültige Statistiken: Summe</p> |
| CampaignInvalid | <p>Eine Nachricht, die vom Fahrzeug gesendet und vom AWS IoT empfangen wurde FleetWise, wenn die Kampagne nicht gültig ist.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: CampaignName</p>                               |

| Kennzahl         | Beschreibung                                                                                                                                                                                                                   |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | Gültige Statistiken: Summe                                                                                                                                                                                                     |
| CampaignNotFound | <p>Eine Nachricht, die vom Fahrzeug gesendet und vom AWS IoT empfangen wurde FleetWise , wobei die Kampagne nicht bekannt ist.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: CampaignName</p> <p>Gültige Statistiken: Summe</p> |

## Kampagnendaten, Zielkennzahlen

| Kennzahl             | Beschreibung                                                                                                                                                                                                              |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TimestreamWriteError | <p>AWSIoT FleetWise konnte keine Nachricht vom Fahrzeug in die Amazon Timestream-Tabelle schreiben.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: DatabaseName, TableName</p> <p>Gültige Statistiken: Summe</p>            |
| S3 WriteError        | <p>AWSIoT FleetWise konnte keine Nachricht vom Fahrzeug in den Amazon Simple Storage Service (Amazon S3) -Bucket schreiben.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: BucketName</p> <p>Gültige Statistiken: Summe</p> |
| S3 ReadError         | <p>AWSIoT FleetWise konnte keinen Objektschlüssel aus dem Fahrzeug im Amazon Simple Storage Service (Amazon S3) -Bucket lesen.</p>                                                                                        |

| Kennzahl | Beschreibung               |
|----------|----------------------------|
|          | Einheiten: Anzahl          |
|          | Abmessungen: BucketName    |
|          | Gültige Statistiken: Summe |

### Vom Kunden verwaltete AWS KMS Schlüsselkennzahlen

| Kennzahl            | Beschreibung                                                                                                                                                                                                                                                                       |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMS KeyAccessDenied | <p>AWSIoT FleetWise konnte aufgrund eines Fehlers „AWS KMSSchlüsselzugriff verweigert“ keine Nachricht vom Fahrzeug in die Timestream-Tabelle oder den Amazon S3 S3-Bucket schreiben.</p> <p>Einheiten: Anzahl</p> <p>Abmessungen: KMS KeyId</p> <p>Gültige Statistiken: Summe</p> |

## Überwachung des AWS IoT FleetWise mit Amazon CloudWatch Logs

Amazon CloudWatch Logs überwacht die Ereignisse, die in Ihren Ressourcen auftreten, und benachrichtigt Sie, wenn Probleme auftreten. Wenn Sie eine Warnung erhalten, können Sie auf die Protokolldateien zugreifen, um Informationen über das jeweilige Ereignis zu erhalten. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).

### AWS FleetWise IoT-Protokolle in der CloudWatch Konsole anzeigen

#### Important

Bevor Sie die AWS FleetWise IoT-Protokollgruppe in der CloudWatch Konsole sehen können, stellen Sie sicher, dass Folgendes zutrifft:

- Sie haben die Protokollierung im AWS IoT aktiviert FleetWise. Weitere Informationen zur Protokollierung finden Sie unter [AWS FleetWise IoT-Protokollierung konfigurieren](#).
- Es gibt bereits Protokolleinträge, die von AWS IoT Vorgängen geschrieben wurden.

So zeigen Sie Ihre AWS FleetWise IoT-Logs in der CloudWatch Konsole an

1. Öffnen Sie die [CloudWatch -Konsole](#).
2. Wählen Sie im Navigationsbereich Logs, Log-Gruppen aus.
3. Wählen Sie die Protokollgruppe aus.
4. Wählen Sie Protokollgruppe suchen aus. Sie sehen eine vollständige Liste der Protokollereignisse, die für Ihr Konto generiert wurden.
5. Wählen Sie das Erweiterungssymbol, um sich einen einzelnen Stream anzusehen und alle Logs mit einem Log-Level von zu finden ERROR.

Sie können auch eine Abfrage in das Suchfeld Ereignisse filtern eingeben. Sie können beispielsweise die folgende Abfrage ausprobieren:

```
{ $.logLevel = "ERROR" }
```

Weitere Informationen zum Erstellen von Filterausdrücken finden Sie unter [Filter- und Mustersyntax](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

### Example Protokolleintrag

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

## Arten von Signalereignissen

| Ereignistyp               | Beschreibung                                                                                                                                                                                                                                                                                                      |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MODELING_ERROR            | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht FleetWise enthält Signale, die nicht mit dem Fahrzeugmodell verglichen werden können.</p> <p>Attribute: VehicleName, CampaignName, SignalID signalCatalogName, SignalValue, Min, Max, signalValueRange signalValueRange modelManifestName</p> |
| ILLEGAL_MESSAGE_FROM_EDGE | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht entsprach FleetWise nicht dem erforderlichen Format.</p> <p>Attribute: VehicleName, CampaignName, signalCatalogName</p>                                                                                                                       |
| DECODIERUNGSFEHLER        | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht FleetWise enthält Signale, die nicht anhand des Decoder-Manifests des Fahrzeugs dekodiert werden können.</p> <p>Attribute: CampaignName,, (optional) SignalName signalCatalogName decoderManifestName, (optional) S3URI</p>                   |

## Ereignistypen der Kampagne

| Ereignistyp             | Beschreibung                                                                                                                   |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| FAHRZEUG NICHT GEFUNDEN | <p>Eine Nachricht, die vom Fahrzeug gesendet und vom AWS IoT empfangen wurde FleetWise , wobei das Fahrzeug unbekannt war.</p> |

| Ereignistyp        | Beschreibung                                                                                                                                                                         |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | Attribute: VehicleName, CampaignName                                                                                                                                                 |
| CAMPAIGN_NOT_FOUND | <p>Eine Nachricht, die vom Fahrzeug gesendet und von AWS IoT empfangen wurde FleetWise, wobei die Kampagne unbekannt war.</p> <p>Attribute: VehicleName (optional), CampaignName</p> |
| CAMPAIGN_INVALID   | <p>Eine vom Fahrzeug gesendete und vom AWS IoT empfangene Nachricht FleetWise, bei der die Kampagne nicht gültig war.</p> <p>Attribute: VehicleName (optional), CampaignName</p>     |

#### Kampagnendaten, Zielereignistypen

| Ereignistyp            | Beschreibung                                                                                                                                                                                      |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIMESTREAM_WRITE_ERROR | <p>AWSIoT FleetWise konnte keine Nachricht vom Fahrzeug in die Amazon Timestream-Tabelle schreiben.</p> <p>Attribute: Fahrzeugname, Kampagnenname, timestreamDatabaseName timestreamTableName</p> |
| S3_WRITE_ERROR         | <p>AWSIoT FleetWise konnte keine Nachricht vom Fahrzeug in den Amazon Simple Storage Service (Amazon S3) -Bucket schreiben.</p> <p>Attribute: CampaignName, DestinationName</p>                   |
| S3_READ_ERROR          | <p>AWSIoT FleetWise konnte keinen Objektschlüssel aus dem Fahrzeug im Amazon Simple Storage Service (Amazon S3) -Bucket lesen.</p>                                                                |

| Ereignistyp | Beschreibung                             |
|-------------|------------------------------------------|
|             | Attribute: CampaignName, DestinationName |

Vom Kunden verwaltete wichtige Ereignistypen AWS KMS

| Ereignistyp           | Beschreibung                                                                                                                                                                       |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMS_KEY_ACCESS_DENIED | AWSIoT FleetWise konnte aufgrund eines Fehlers „AWS KMSSchlüsselzugriff verweigert“ keine Nachricht vom Fahrzeug in die Timestream-Tabelle oder den Amazon S3 S3-Bucket schreiben. |

## Attribute

Alle CloudWatch Log-Einträge enthalten die folgenden Attribute:

accountId

Deine AWS-Konto ID.

eventType

Der Ereignistyp, für den das Protokoll generiert wurde. Der Wert des Ereignistyps hängt vom Ereignis ab, das den Protokolleintrag generiert hat. Jede Beschreibung des Protokolleintrags enthält den Wert von eventType für diesen Protokolleintrag.

logLevel

Die Protokollebene, die verwendet wird. Weitere Informationen finden Sie unter [Protokollebenen](#) im AWS IoT CoreEntwicklerhandbuch.

Nachricht

Enthält spezifische Details zum Protokoll.

Zeitstempel

Der Epochen-Millisekunden-Zeitstempel, zu dem das IoT das Protokoll verarbeitet hat. AWS FleetWise

## Optionale Attribute

CloudWatch Protokolleinträge enthalten optional diese Attribute, abhängig von: `eventType`

### `decoderManifestName`

Der Name des Decoder-Manifests, das das Signal enthält.

### Name des Ziels

Der Name des Ziels für Fahrzeugdaten. Zum Beispiel der Amazon S3 S3-Bucket-Name.

### Name der Kampagne

Der Name der Kampagne.

### `signalCatalogName`

Der Name des Signalkatalogs, der das Signal enthält.

### SignalID

Die ID des Fehlersignals.

### Signal-IDs

Eine Liste von Fehlersignal-IDs.

### Signalname

Der Name des Signals.

### `signalTimestampEpochFrau`

Der Zeitstempel des Fehlersignals.

### SignalWert

Der Wert des Fehlersignals.

### `signalValueRangeMax`

Die maximale Reichweite des Fehlersignals.

### `signalValueRangeMin.`

Der Mindestbereich des Fehlersignals.

## S3URI

Die eindeutige Amazon S3 S3-ID einer Amazon Ion-Datei aus einer Fahrzeugnachricht.

## timestreamDatabaseName

Der Name der Timestream-Datenbank.

## timestreamTableName

Der Name der Timestream-Tabelle.

## Name des Fahrzeugs

Der Name des Fahrzeugs.

## AWS FleetWise IoT-Protokollierung konfigurieren

Sie können Ihre AWS FleetWise IoT-Protokolldaten an eine CloudWatch Protokollgruppe senden. CloudWatch Protokolle bieten Transparenz für den Fall, dass FleetWise das AWS IoT Nachrichten von Fahrzeugen nicht verarbeiten kann. Dies kann beispielsweise aufgrund einer fehlerhaften Konfiguration oder anderer Client-Fehler passieren. Sie werden über alle Fehler informiert, sodass Sie Probleme identifizieren und beheben können.

Bevor Sie Protokolle an senden können CloudWatch, müssen Sie eine CloudWatch Protokollgruppe erstellen. Konfigurieren Sie die Protokollgruppe mit demselben Konto und in derselben Region, die Sie mit AWS IoT verwendet haben FleetWise. Wenn Sie die Protokollierung in AWS IoT aktivieren FleetWise, geben Sie den Namen der Protokollgruppe an. Nachdem die Protokollierung aktiviert wurde, FleetWise übermittelt AWS IoT Protokolle in CloudWatch Protokollstreams an die Protokollgruppe.

Sie können die vom AWS IoT gesendeten Protokolldaten FleetWise in der CloudWatch Konsole anzeigen. Weitere Informationen zur Konfiguration einer CloudWatch Protokollgruppe und zum Anzeigen von Protokolldaten finden Sie unter [Arbeiten mit Protokollgruppen](#).

## Berechtigungen zum Veröffentlichen von Protokollen in CloudWatch

Für die Konfiguration der Protokollierung für eine CloudWatch Protokollgruppe sind die in diesem Abschnitt beschriebenen Berechtigungseinstellungen erforderlich. Informationen zur Verwaltung von Berechtigungen finden Sie unter [Zugriffsverwaltung für AWS Ressourcen](#) im IAM-Benutzerhandbuch.

Mit diesen Berechtigungen können Sie die Protokollierungskonfiguration ändern, die Protokollzustellung für CloudWatch Ihre Protokollgruppe konfigurieren und Informationen zu dieser Gruppe abrufen.

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

Wenn Aktionen für alle AWS-Ressourcen zulässig sind, wird dies in der Richtlinie mit einer "Resource"-Einstellung von "\*" angezeigt. Dies bedeutet, dass die Aktionen für alle AWS-Ressourcen zulässig sind, die diese Aktionen unterstützen.

## Protokollierung im AWS IoT konfigurieren FleetWise (Konsole)

In diesem Abschnitt wird beschrieben, wie die AWS FleetWise IoT-Konsole zur Konfiguration der Protokollierung verwendet wird.

So konfigurieren Sie die Protokollierung mit der AWS FleetWise IoT-Konsole

1. Öffnen Sie die [AWS FleetWise IoT-Konsole](#).
2. Wählen Sie im linken Bereich Settings (Einstellungen) aus.
3. Wählen Sie auf der Einstellungsseite im Abschnitt Protokollierung die Option Bearbeiten aus.
4. Geben Sie im Abschnitt CloudWatch Protokollierung die Protokollgruppe ein.
5. Um Ihre Änderungen zu speichern, wählen Sie Submit.

Nachdem Sie die Protokollierung aktiviert haben, können Sie Ihre Protokolldaten in der [CloudWatch Konsole](#) anzeigen.

## Standardprotokollierung in AWS IoT FleetWise (CLI) konfigurieren

In diesem Abschnitt wird beschrieben, wie die Protokollierung für AWS IoT FleetWise mithilfe der CLI konfiguriert wird.

Sie können dieses Verfahren auch mit der API durchführen, indem Sie die Methoden der AWS-API verwenden, die den hier gezeigten CLI-Befehlen entsprechen. Sie können den [GetLoggingOptions](#) API-Vorgang verwenden, um die aktuelle Konfiguration abzurufen, und den [PutLoggingOptions](#) API-Vorgang, um die Konfiguration zu ändern.

So konfigurieren Sie die Protokollierung für AWS IoT mit der CLI FleetWise

1. Verwenden Sie den get-logging-options Befehl, um die Protokollierungsoptionen für Ihr Konto abzurufen.

```
aws iotfleetwise get-logging-options
```

2. Verwenden Sie den put-logging-options Befehl, um die Protokollierung zu aktivieren.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery  
logType=ERROR,logGroupName=MyLogGroup
```

Wobei:

## logType

Der Protokolltyp, um Daten an CloudWatch Logs zu senden. Um die Protokollierung zu deaktivieren, ändern Sie den Wert auf OFF.

## logGroupName

Die Gruppe CloudWatch Logs, an die der Vorgang Daten sendet. Stellen Sie sicher, dass Sie den Namen der Protokollgruppe erstellen, bevor Sie die Protokollierung für AWS IoT aktivieren FleetWise.

Nachdem Sie die Protokollierung aktiviert haben, finden Sie weitere Informationen unter [Suchen nach Protokolleinträgen mit der AWS CLI](#).

## Protokollierung AWS IoT FleetWise API-Aufrufe mit AWS CloudTrail

AWS IoT FleetWise ist integriert mit AWS CloudTrail, ein Dienst, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service in AWS IoT FleetWise. CloudTrail erfasst alle API-Aufrufe für AWS IoT FleetWise als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS IoT FleetWise Konsolen- und Codeaufrufe an die AWS IoT FleetWise API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von aktivierten CloudTrail Ereignissen für einen Amazon S3-Bucket, einschließlich Ereignissen für AWS IoT FleetWise. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole in Event history (Ereignisverlauf) anzeigen. Verwendung der gesammelten Informationen von CloudTrail, können Sie die Anfrage ermitteln, die gestellt wurde an AWS IoT FleetWise, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

## AWS IoT FleetWise Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Wenn Aktivität stattfindet in AWS IoT FleetWise, diese Aktivität wird in einem CloudTrail Veranstaltung zusammen mit anderen AWS Serviceveranstaltungen in Historie des Ereignisses. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail -API-Ereignisverlauf](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Veranstaltungen für AWS IoT FleetWise, erstellen Sie eine Spur. Ein Wanderweg aktiviert CloudTrail um Protokolldateien an einen Amazon S3-Bucket zu liefern. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [In CloudTrail unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen](#)
- [Empfangen von CloudTrail-Protokolldateien von mehreren Konten](#)

Alle AWS IoT FleetWise-Aktionen werden protokolliert von CloudTrail und sind dokumentiert in der [AWS IoT FleetWise API-Referenz](#). Zum Beispiel werden durch Aufrufe der `CreateCampaign`-, `AssociateVehicleFleet`- und `GetModelManifest`-Aktionen Einträge in den CloudTrail-Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anfrage mit Root- oder IAM-Benutzer-Anmeldeinformationen von ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail-Element `userIdentity`](#).

## Verständnis AWS IoT FleetWise Logdateieinträge

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit

der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die *AssociateVehicleFleet*-Operation demonstriert:

```
{
  "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"
}
```

# Dokumentenverlauf für den AWS IoT FleetWise Developer Guide

In der folgenden Tabelle werden die Dokumentationsversionen für AWS IoT beschrieben FleetWise.

| Änderung                                                        | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                      | Datum             |
|-----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Vorschau der Daten des Bildverarbeitungssystems</a> | Sie können die Vorschau von Bildverarbeitungssystemdateien aus dem AWS IoT der Dinge verwenden FleetWise , um Daten aus Fahrzeugen, einschließlich Kameras, Radaren und Lidars, zu sammeln und zu organisieren. Dabei werden sowohl strukturierte als auch unstrukturierte Bilddaten, Metadaten (Event-ID, Kampagne, Fahrzeug) und Standardsensordaten (Telemetriedaten) automatisch in der Cloud synchronisiert. | 26. November 2023 |
| <a href="#">AWS KMS vom Kunden verwaltete Schlüssel</a>         | AWS IoT unterstützt FleetWise jetzt vom AWS KMS Kunden verwaltete Schlüssel. Sie können den KMS-Schlüssel verwenden, um serverseitige Daten zu AWS FleetWise IoT-Ressourcen (Signalkatalog, Fahrzeugmodell, Decodermanifest, Fahrzeuge und Konfigurationen für Datenerfassungskampagnen)                                                                                                                          | 16. Oktober 2023  |

zu verschlüsseln, die in gespeichert sind. AWS Cloud

### [Objektspeicher in Amazon S3](#)

AWSIoT unterstützt FleetWise jetzt das Speichern von Daten mit Amazon Simple Storage Service (Amazon S3). Sie können Daten, die während Kampagnen gesammelt wurden, zusätzlich zu Amazon Timestream in Amazon S3 speichern.

01. Juni 2023

### [Allgemeine Verfügbarkeit](#)

Dies ist die öffentliche Version von AWS IoT FleetWise.

27. September 2022

### [Erstversion](#)

Dies ist die Vorabversion des AWS IoT FleetWise Developer Guide.

30. November 2021

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.