



Entwicklerhandbuch

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: 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 Kinesis Video Streams?	1
Verfügbarkeit in Regionen	3
Verwenden Sie Kinesis Video Streams zum ersten Mal?	4
Systemanforderungen	5
Anforderungen an die Kamera	5
Getestete Betriebssysteme	6
SDK-Speicheranforderungen	6
So funktioniert es	7
API und Producer-Bibliotheken	8
Kinesis Video Streams	8
Produzentenbibliotheken	11
Video Playback (Videowiedergabe)	11
Anforderungen für die Wiedergabe	12
Videowiedergabe mit HLS	15
Videowiedergabe mit MPEG-DASH	25
Verwenden von Streaming-Metadaten	29
Hinzufügen von Metadaten zu einem Kinesis-Videostream	30
Nutzung von Metadaten, die in einen Kinesis-Videostream eingebettet sind	32
Beschränkungen für Streaming-Metadaten	34
Datenmodell	34
Stream-Header-Elemente	34
Streamen Sie Track-Daten	41
Frame-Header-Elemente	43
MKV-Frame-Daten	43
Erste Schritte	44
Einrichten eines -Kontos	44
Melden Sie sich an für ein AWS-Konto	45
Erstellen eines Administratorbenutzers	45
Erstellen Sie einen Schlüssel AWS-Konto	46
Einen Kinesis-Videostream erstellen	47
Erstellen Sie mit der Konsole einen Videostream	47
Erstellen Sie einen Videostream mit dem AWS CLI	47
Daten an einen Amazon Kinesis Kinesis-Videostream senden	48
Erstellen Sie das SDK und die Beispiele	48

Führen Sie die Beispiele aus, um Medien auf Kinesis Video Streams hochzuladen	51
Bestätigungsobjekte überprüfen	53
Mediendaten konsumieren	53
Medien in der Konsole anzeigen	53
Konsumieren Sie Mediendaten mit HLS	54
Edge-Agent	55
API-Operationen von Amazon Kinesis Video Streams Edge Agent	56
Überwachen von Amazon Kinesis Video Streams Edge Agent	56
Im Nicht-AWS IoT Greengrass-Modus bereitstellen	56
1. Abhängigkeiten installieren	57
2. Erstellen von Ressourcen für Ihre IP-Kamera-RTSP-URLs	59
3. Erstellen einer IAM-Berechtigungsrichtlinie	61
4. Erstellen einer IAM-Rolle	63
5. Erstellen des AWS IoT Rollenalias	64
6. Erstellen der AWS IoT-Richtlinie	65
7. Erstellen eines AWS IoT Objekts und Abrufen von AWS IoT Core Anmeldeinformationen	66
8. Erstellen und Ausführen des Edge-Agenten von Amazon Kinesis Video Streams	69
9. (Optional) Installieren des CloudWatch Agenten	81
10. (Optional) Führen Sie Amazon Kinesis Video Streams Edge Agent aus	84
Bereitstellen in AWS IoT Greengrass	87
1. Erstellen einer Ubuntu-Instance	87
2. Einrichten des AWS IoT Greengrass Core-Geräts	89
3. Erstellen von Ressourcen für Ihre IP-Kamera-RTSP-URLs	90
4. Hinzufügen von Berechtigungen zur TES-Rolle	92
5. Installieren der Secret Manager-Komponente	95
6. Bereitstellen von Amazon Kinesis Video Streams Edge Agent auf dem Gerät	98
7. (Optional) Installieren der AWS IoT Greengrass Log Manager-Komponente	107
Häufig gestellte Fragen	111
Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent?	111
Unterstützt der Edge Agent von Amazon Kinesis Video Streams H.265-Medien?	111
Funktioniert der Edge-Agent von Amazon Kinesis Video Streams in AL2?	111
Wie kann ich mehrere Streams innerhalb des AWS IoT Objekts oder Geräts ausführen?	111
Wie kann ich eine bearbeiten, StartEdgeConfigurationUpdate nachdem sie gesendet wurde?	112
Gibt es Beispiele für gängige ScheduleConfigs?	112

Gibt es ein maximales Stream-Limit?	113
Wie starte ich einen Auftrag neu, bei dem ein Fehler aufgetreten ist?	113
Wie überwache ich den Zustand meines Amazon Kinesis Video Streams Edge Agent?	113
Video über eine VPC streamen	114
Zusätzliche Informationen	114
VPC-Endpunktverfahren	115
Images	117
Erste Schritte mit GetImages	117
Erste Schritte mit der Amazon S3-Lieferung	118
UpdateImageGenerationConfiguration	118
DescribeImageGenerationConfiguration	120
MKV-Tags für Produzenten	121
Hinzufügen von Metadaten-Tags im Producer SDK mithilfe vonPutEventMetaData	122
Einschränkungen	122
S3-Objektmetadaten	123
S3-Objektpfad (Bild)	123
Amazon S3-URI-Empfehlungen zum Schutz vor Drosselung	124
Benachrichtigungen	125
UpdateNotificationConfiguration	125
DescribeNotificationConfiguration	125
Hersteller MKV-Tags	121
Syntax für Producer-MKV-Tags	121
Beschränkungen für MKV-Tags	126
.....	126
.....	126
Nutzlast für Amazon SNS-Themen	127
Ihre Amazon SNS SNS-Nachrichten anzeigen	128
Sicherheit	129
Datenschutz	130
Was ist serverseitige Verschlüsselung für Kinesis Video Streams?	130
Überlegungen zu Kosten, Regionen und Leistung	130
Wie fange ich mit der serverseitigen Verschlüsselung an?	131
Benutzergenerierten KMS-Schlüssel erstellen und verwenden	132
Berechtigungen zur Verwendung von benutzergenerierten KMS-Schlüsseln	133
Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von IAM	135
Richtliniensyntax	135

Aktionen für Kinesis Video Streams	136
Amazon-Ressourcennamen (ARNs) für Kinesis Video Streams	137
Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren	137
Beispielrichtlinien	138
Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von AWS IoT	140
AWS IoT ThingName als Streamname	141
AWS IoT CertificateId als Streamname	148
Verwenden Sie AWS IoT Anmeldeinformationen, um zu einem fest codierten Stream-Namen zu streamen	149
Überwachen	150
Compliance-Validierung	150
Ausfallsicherheit	152
Sicherheit der Infrastruktur	152
Bewährte Methoden für die Sicherheit	152
Implementieren des Zugriffs mit geringsten Berechtigungen	153
Verwenden von IAM-Rollen	153
Wird CloudTrail zur Überwachung von API-Aufrufen verwendet	154
Producer-Bibliotheken	155
Produzent und Kunde von Kinesis Video Streams	155
Produzentenbibliothek von Kinesis Video Streams	156
Verwandte Themen	157
Java Producer-Bibliothek	157
Vorgehensweise: Verwenden des Java Producer-SDK	158
Schritt 1: Herunterladen und Konfigurieren des Codes	159
Schritt 2: Schreiben und Untersuchen des Codes	160
Schritt 3: Ausführen und Überprüfen des Codes	162
Android Producer-Bibliothek	163
Vorgehensweise: Verwenden des Android Producer-SDK	163
Voraussetzungen	163
Schritt 1: Herunterladen und Konfigurieren des Codes	167
Schritt 2: Untersuchen des Codes	169
Schritt 3: Ausführen und Überprüfen des Codes	171
C++ Producer-Bibliothek	173
Objektmodell	173
Medien in den Stream bringen	173
Callback-Schnittstellen	173

Vorgehensweise: Verwenden des C++ Producer-SDK	174
Schritt 1: Herunterladen und Konfigurieren des Codes	177
Schritt 2: Schreiben und untersuchen Sie den Code	177
Schritt 3: Ausführen und verifizieren Sie den Code	184
Das C++ Producer SDK als GStreamer-Plugin verwenden	185
Verwendung des C++ Producer SDK als GStreamer-Plugin in einem Docker-Container	185
Verwenden der Protokollierung	185
C-Producer-Bibliothek	186
Objektmodell	186
Medien in den Stream bringen	188
Vorgehensweise: Verwenden des C-Producer-SDK	188
Schritt 1: Herunterladen des Codes	191
Schritt 2: Schreiben und untersuchen Sie den Code	191
Schritt 3: Führen Sie den Code aus und überprüfen Sie ihn	194
C++ Producer SDK auf Raspberry Pi	196
Voraussetzungen	197
Erstellen eines IAM-Benutzers mit der Berechtigung zum Schreiben in Kinesis Video Streams	198
Verbinden Ihres Raspberry Pi mit Ihrem WLAN-Netzwerk	199
Remote-Verbindung zu Ihrem Raspberry Pi herstellen	200
Konfigurieren der Raspberry Pi-Kamera	201
Installieren von Softwarevoraussetzungen	202
Herunterladen und Erstellen des Kinesis Video Streams C++ Producer SDK	202
Streamen Sie Videos in Ihren Kinesis-Videostream und sehen Sie sich den Livestream an .	203
Referenz	204
Beschränkungen des Producer-SDK	205
Fehlercodereferenz	209
NAL Adaptation Flags	269
Producer-Strukturen	270
Streamstrukturen	273
Rückrufe	295
"Stream Parser"-Bibliothek	304
Verfahren: Verwenden der Kinesis-Videodatenstrom-Parserbibliothek	304
Voraussetzungen	304
Schritt 1: Herunterladen und Konfigurieren des Codes	305
Nächster Schritt	305

Schritt 2: Schreiben und Untersuchen des Datenstroms	305
StreamingMkvReader	306
FragmentMetadataVisitor	306
OutputSegmentMerger	308
KinesisVideoExample	309
Nächster Schritt	313
Schritt 3: Ausführen und Überprüfen des Codes	313
Beispiele	314
Beispiele: Senden von Daten an Kinesis Video Streams	314
Beispiele: Abrufen von Daten aus Kinesis Video Streams	314
Beispiele: Wiedergabe von Videodaten	314
Voraussetzungen	315
GStreamer	315
Herunterladen, Erstellen und Konfigurieren des GStreamer-Elements	316
Ausführen des GStreamer-Elements	317
Startbefehle	318
Ausführen des GStreamer-Elements in einem Docker-Container	320
Parameterreferenz	323
PutMedia API	337
Schritt 1: Herunterladen und Konfigurieren des Codes	338
Schritt 2: Schreiben und Untersuchen des Codes	339
Schritt 3: Ausführen und Überprüfen des Codes	341
RTSP und Docker	342
Voraussetzungen	343
Erstellen des Docker-Images	343
Ausführen der RTSP-Beispielanwendung	344
Renderer	345
Voraussetzungen	346
Ausführen des Renderer-Beispiels	346
So funktioniert's	347
SageMaker	348
Voraussetzungen	349
Erstellen der Anwendung	351
Überwachung der Anwendung	353
Erweitern der Anwendung	355
Bereinigen der Anwendung	356

Überwachen	358
Metriken überwachen mit CloudWatch	359
CloudWatch Anleitung zu Metriken	374
Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch	379
CloudWatch Leitfaden für Kennzahlen für Amazon Kinesis Video Streams Edge Agent	382
Protokollieren von CloudTrail-API-Aufrufen mit	384
Amazon Kinesis Video Streams und CloudTrail	384
Beispiel: Einträge in der Amazon Kinesis Video Streams Streams-Protokolldatei	386
Kontingente	390
API-Servicekontingente der Steuerebene	390
API-Servicekontingente für Medien und archivierte Medien	396
Kontingente für Fragment-Metadaten und Fragment-Medien	402
Fragment-Metadatenkontingente	405
Stream-Tags	405
Fehlerbehebung	406
Fehlerbehebung bei allgemeinen Problemen	406
Latenz zu hoch	406
Behebung von API-Problemen	407
Fehler: „Unbekannte Optionen“	407
Fehler: „Unable to determine service/operation name to be authorized“ (Zu autorisierender Service-/Operationsname kann nicht ermittelt werden)	407
Fehler: „Failed to put a frame in the stream“ (Es konnte kein Frame in den Stream eingefügt werden)	408
Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“	408
Fehler: "STATUS_STORE_OUT_OF_MEMORY"	409
Behebung von HLS-Problemen	409
Behebung von Java-Problemen	409
Aktivieren von Java-Protokollen	409
Behebung von Problemen mit der Producer Library	410
Das Producer SDK kann nicht kompiliert werden	411
Der Videostream wird nicht in der Konsole angezeigt	411
Fehler: „Security token included in the request is invalid“ (Das Sicherheits-Token der Anfrage ist ungültig) beim Streamen von Daten mit der GStreamer-Demo-Anwendung	412
Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“	412

GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt”	412
Fehler: „Failed to allocate heap” (Heap konnte nicht zugeordnet werden) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi	413
Fehler: „Illegal Instruction” (Ungültige Anweisung) beim Ausführen der GStreamer-Demo auf Raspberry Pi	413
Die Kamera kann nicht auf Raspberry Pi geladen werden.	413
Kamera kann auf macOS High Sierra nicht gefunden werden.	414
Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden	414
Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung	414
Zeitstempel-/Zeitraum-Assertion während der Laufzeit auf Raspberry Pi	415
Assertion auf <code>gst_value_set_fraction_range_full</code> auf Raspberry Pi	415
STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d)-Fehler unter Android	415
Fehler: Die maximale Fragmentdauer wurde erreicht	415
"Ungültiger Objektname abgelaufen", Fehler bei der Verwendung der IoT-Autorisierung	416
Behebung von Problemen mit der Stream Parser-Bibliothek	416
Es kann nicht auf einen einzelnen Frame aus dem Stream zugegriffen werden	417
Fragmentdekodierungsfehler	417
Dokumentverlauf	418
API-Referenz	423
Aktionen	423
Amazon Kinesis Video Streams	425
Medien für Amazon Kinesis Video Streams	549
Amazon Kinesis Video Streams	566
Amazon Kinesis Video Video Streams	615
Amazon Kinesis Video WebRTC Storage	624
Datentypen	628
Amazon Kinesis Video Streams	629
Medien Amazon Kinesis Video Streams	674
Amazon Kinesis Video Streams Streams-Archivierte Medien	677
Amazon Kinesis Video Video Kinesis Video Streams	696
Amazon Kinesis Video WebRTC Speicher	698
Häufige Fehler	698
Geläufige Parameter	700
.....	dciii

Was ist Kinesis Video Streams?

Sie können Amazon Kinesis Video Streams, ein vollständig verwaltetes Programm, verwenden AWS-Service, um Live-Video von Geräten auf das AWS Cloud zu streamen oder Anwendungen für Echtzeit-Videoverarbeitung oder stapelorientierte Videoanalyse zu entwickeln.

Kinesis Video Streams ist nicht nur ein Speicher für Videodaten. Sie können sie verwenden, um Ihre Videostreams in Echtzeit anzusehen, während sie in der Cloud empfangen werden. Sie können entweder Ihre Live-Streams in der AWS Management Console überwachen oder Ihre eigene Überwachungsanwendung entwickeln, die die Kinesis Video Streams API-Bibliothek zur Anzeige von Live-Videos verwendet.

Sie können Kinesis Video Streams verwenden, um riesige Mengen an Live-Videodaten aus Millionen von Quellen zu erfassen, darunter Smartphones, Sicherheitskameras, Webcams, in Autos eingebettete Kameras, Drohnen und andere Quellen. Sie können auch zeitserialisierte Daten wie Audiodaten, Wärmebilder, Tiefendaten und RADAR-Daten senden, die keine Videodaten sind. Da Live-Videostreams von diesen Quellen in einen Kinesis-Videostream übertragen werden, können Sie Anwendungen erstellen, frame-by-frame die in Echtzeit auf die Daten zugreifen und so eine Verarbeitung mit geringer Latenz ermöglichen. Kinesis Video Streams ist quellenunabhängig. Sie können Videos von der Webcam eines Computers mithilfe der [GStreamer](#) Bibliothek oder von einer Kamera in Ihrem Netzwerk mithilfe des Echtzeit-Streaming-Protokolls (RTSP) streamen.

Sie können Ihren Kinesis-Videostream auch so konfigurieren, dass Mediendaten für den angegebenen Aufbewahrungszeitraum dauerhaft gespeichert werden. Kinesis Video Streams speichert diese Daten automatisch und verschlüsselt sie im Ruhezustand. Darüber hinaus indiziert Kinesis Video Streams gespeicherte Daten auf der Grundlage von Produktionszeitstempeln und Aufnahmezeitstempeln. Sie können Anwendungen erstellen, die die Videodaten regelmäßig stapelweise verarbeiten, oder Sie können Anwendungen erstellen, die für verschiedene Anwendungsfälle einmaligen Zugriff auf historische Daten erfordern.

Ihre benutzerdefinierten Anwendungen, ob Echtzeit- oder Batch-orientiert, können auf Amazon EC2 EC2-Instances ausgeführt werden. Diese Anwendungen können Daten mithilfe von Open-Source-Deep-Learning-Algorithmen verarbeiten oder Drittanbieteranwendungen verwenden, die in Kinesis Video Streams integriert sind.

Die Verwendung von Kinesis Video Streams bietet unter anderem folgende Vorteile:

- **Connect und von Millionen von Geräten streamen** — Mit Kinesis Video Streams können Sie Video-, Audio- und andere Daten von Millionen von Geräten verbinden und streamen, darunter Verbraucher-Smartphones, Drohnen und Dashcams. Sie können die Kinesis Video Streams Streams-Producer-Bibliotheken verwenden, um Ihre Geräte zu konfigurieren und zuverlässig in Echtzeit oder als after-the-fact Medien-Uploads zu streamen.
- **Dauerhaftes Speichern, Verschlüsseln und Indexieren von Daten** — Sie können Ihren Kinesis-Videostream so konfigurieren, dass Mediendaten für benutzerdefinierte Aufbewahrungsfristen dauerhaft gespeichert werden. Kinesis Video Streams generiert außerdem einen Index über die gespeicherten Daten auf der Grundlage von vom Hersteller oder vom Service generierten Zeitstempeln. Ihre Anwendungen können mithilfe des Zeitindex bestimmte Daten in einem Stream abrufen.
- **Konzentrieren Sie sich auf die Verwaltung von Anwendungen statt auf die Infrastruktur** — Kinesis Video Streams ist serverlos, sodass keine Infrastruktur eingerichtet oder verwaltet werden muss. Sie müssen sich keine Gedanken über die Bereitstellung, Konfiguration oder elastische Skalierung der zugrunde liegenden Infrastruktur machen, da Ihre Datenströme und die Anzahl der verbrauchenden Anwendungen wachsen und schrumpfen. Kinesis Video Streams erledigt automatisch die gesamte Verwaltung und Wartung, die für die Verwaltung von Streams erforderlich ist, sodass Sie sich auf die Anwendungen konzentrieren können, nicht auf die Infrastruktur.
- **Echtzeit- und Batch-Anwendungen auf Datenströmen erstellen** — Mit Kinesis Video Streams können Sie benutzerdefinierte Echtzeitanwendungen erstellen, die mit Live-Datenströmen arbeiten, sowie Batch- oder Einmalanwendungen erstellen, die dauerhaft persistente Daten ohne strenge Latenzanforderungen verarbeiten. Sie können benutzerdefinierte Anwendungen erstellen, bereitstellen und verwalten: Open-Source-Anwendungen (Apache MXNet, OpenCV), selbst entwickelte Lösungen oder Lösungen von Drittanbietern, mit denen Sie Ihre Streams verarbeiten und AWS Marketplace analysieren. Sie können Kinesis Video Streams Get Streams-APIs verwenden, um mehrere gleichzeitige Anwendungen zu erstellen, die Daten in Echtzeit oder stapelweise verarbeiten.
- **Sicheres Streamen von Daten** — Kinesis Video Streams verschlüsselt alle Daten, während sie durch den Service fließen und wenn die Daten gespeichert werden. Kinesis Video Streams erzwingt eine auf Transport Layer Security (TLS) basierende Verschlüsselung beim Datenstreaming von Geräten und verschlüsselt alle Daten im Ruhezustand mit AWS Key Management Service (KMS). Darüber hinaus können Sie den Zugriff auf Ihre Daten mithilfe von AWS Identity and Access Management (IAM) verwalten.
- **Nutzungsabhängige Bezahlung** — Weitere Informationen finden Sie unter [AWS Pricing Calculator](#).

Verfügbarkeit in Regionen

Amazon Kinesis Video Streams ist in den folgenden Regionen verfügbar:

Name der Region	AWSRegionalcode
USA Ost (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
USA West (Oregon)	us-west-2
Afrika (Kapstadt)	af-south-1
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
Europa (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Südamerika (São Paulo)	sa-east-1

Verwenden Sie Kinesis Video Streams zum ersten Mal?

Wenn Sie Kinesis Video Streams zum ersten Mal verwenden, empfehlen wir Ihnen, die folgenden Abschnitte der Reihe nach zu lesen:

1. [Kinesis Video Streams: So funktioniert's](#)— Um mehr über die Konzepte von Kinesis Video Streams zu erfahren.
2. [Erste Schritte mit Amazon Kinesis Video Streams](#)— Um Ihr Konto einzurichten und Kinesis Video Streams zu testen.
3. [Produzentenbibliotheken von Kinesis Video Streams](#)— Um mehr über die Erstellung einer Kinesis Video Streams-Producer-Anwendung zu erfahren.
4. [Parser-Bibliothek für Kinesis-Videostreams](#)— Um mehr über die Verarbeitung eingehender Datenframes in einer Kinesis Video Streams Streams-Verbraucheranwendung zu erfahren.
5. [Beispiele für Amazon Kinesis Video Streams](#) — Um weitere Beispiele dafür zu sehen, was Sie mit Kinesis Video Streams machen können.

Kinesis Video Streams Systemanforderungen

Die folgenden Abschnitte enthalten Hardware-, Software- und Speicheranforderungen für Amazon Kinesis Video Streams.

Themen

- [Anforderungen an die Kamera](#)
- [Getestete Betriebssysteme](#)
- [SDK-Speicheranforderungen](#)

Anforderungen an die Kamera

Für Kameras, die für die Ausführung des Kinesis Video Streams Producer SDK und der Beispiele verwendet werden, gelten die folgenden Speicheranforderungen:

- Die SDK-Inhaltsansicht benötigt 16 MB Arbeitsspeicher.
- Die Standardkonfiguration der Beispielanwendung ist 128 MiB Arbeitsspeicher. Dieser Wert eignet sich für Producer, die eine gute Netzwerkverbindung besitzen und keine zusätzliche Pufferung benötigen. Wenn die Netzwerkverbindung schlecht ist und mehr Pufferung erforderlich ist, können Sie den Speicherbedarf pro Sekunde Pufferung berechnen, indem Sie die Bildrate pro Sekunde mit der Größe des Bildspeichers multiplizieren. Weitere Informationen über die Speicherzuordnung finden Sie unter [StorageInfo](#).

Wir empfehlen die Verwendung von USB- oder RTSP-Kameras (Real Time Streaming Protocol), die Daten mit H.264 codieren, da dadurch die CPU entlastet wird.

Derzeit unterstützt die Demo-Anwendung das User Datagram Protocol (UDP) für RTSP-Streaming nicht. Diese Funktionen wird zu einem späteren Zeitpunkt hinzugefügt.

Das Producer-SDK unterstützt die folgenden Kameratypen:

- Web-Kameras.
- USB-Kameras.
- Kameras mit H.264-Codierung (bevorzugt).
- Kameras ohne H.264-Kodierung.

- Raspberry Pi-Kameramodul. Dies wird für Raspberry Pi-Geräte bevorzugt, da es für die Videodatenübertragung an die GPU angeschlossen wird, sodass kein Zusatzaufwand für die CPU-Verarbeitung entsteht.
- RTSP (Netzwerk)-Kameras. Diese Kameras werden bevorzugt, da die Videostreams bereits mit H.264 kodiert sind.

Getestete Betriebssysteme

Wir haben Web-Kameras und RTSP-Kameras mit den folgenden Geräten und Betriebssystemen getestet:

- Mac mini
 - High Sierra
- MacBook Profi-Laptops
 - Sierra (10.12)
 - El Capitan (10.11)
- HP-Notebooks unter Ubuntu 16.04
- Ubuntu 17.10 (Docker-Container)
- Raspberry Pi 3

SDK-Speicheranforderungen

Für die Installation des [Produzentenbibliotheken von Kinesis Video Streams](#) besteht ein Mindestspeicherbedarf von 170 MB und eine empfohlener Speicherbedarf von 512 MB.

Kinesis Video Streams: So funktioniert's

Themen

- [Kinesis Video Streams](#)
- [Wiedergabe von Kinesis Video Streams](#)
- [Verwenden von Streaming-Metadaten mit Kinesis Video Streams](#)
- [Kinesis Video Streams Streams-Datenmodell](#)

Sie können Amazon Kinesis Video Streams, ein vollständig verwaltetes Programm, verwenden AWS-Service, um Live-Videos von Geräten auf das Gerät zu streamen AWS Cloud und dauerhaft zu speichern. Anschließend können Sie eigene Anwendungen für die Echtzeitverarbeitung von Videodaten oder die Analyse von Videodaten mittels Stapelverarbeitung erstellen.

Das folgende Diagramm gibt einen Überblick über die Funktionsweise von Kinesis Video Streams.

Die Abbildung illustriert die Interaktion zwischen den folgenden Komponenten:

- **Produzent** — Jede Quelle, die Daten in einen Kinesis-Videostream einspeist. Produzent kann jedes Gerät sein, das Videodaten generiert, also z. B. eine Überwachungskamera, eine am Körper getragene Kamera, eine Smartphone-Kamera oder eine Dashcam. Ein Produzent kann zudem andere Daten wie Audiofeeds, Bilder oder RADAR-Daten senden.

Ein einzelner Produzent kann einzelne, aber auch mehrere Videostreams generieren.

Beispielsweise kann eine Videokamera Videodaten an einen Kinesis-Videostream und Audiodaten an einen anderen übertragen.

- **Kinesis Video Streams Producer-Bibliotheken** — Eine Reihe von Software und Bibliotheken, die Sie auf Ihren Geräten installieren und konfigurieren können. Sie können diese Bibliotheken verwenden, um Videos auf verschiedene Weise sicher zu verbinden und zuverlässig zu streamen, unter anderem in Echtzeit, nachdem sie einige Sekunden lang gepuffert wurden, oder als after-the-fact Medien-Upload.
- **Kinesis-Videostream** — Eine Ressource, mit der Sie Live-Videodaten transportieren, optional speichern und die Daten sowohl in Echtzeit als auch stapelweise oder einmalig zur Nutzung zur Verfügung stellen können. In einer typischen Konfiguration hat ein Kinesis-Videostream nur einen Produzenten, der Daten darin veröffentlicht.

Der Stream kann Audio-, Video- und ähnliche zeitcodierte Datenstreams transportieren, z. B. Tiefenmessungsfeeds, RADAR-Feeds und mehr. Sie erstellen einen Kinesis-Videostream mithilfe der AWS Management Console oder programmgesteuert mithilfe der AWS SDKs.

Mehrere unabhängige Anwendungen können einen Kinesis-Videostream parallel nutzen.

- **Consumer** — Ruft Daten wie Fragmente und Frames aus einem Kinesis-Videostream ab, um sie anzusehen, zu verarbeiten oder zu analysieren. Im Allgemeinen werden diese Verbraucher als Kinesis Video Streams Streams-Anwendungen bezeichnet. Sie können Anwendungen schreiben, die Daten in Kinesis Video Streams in Echtzeit nutzen und verarbeiten, oder nachdem die Daten gespeichert und zeitindexiert wurden, wenn keine Verarbeitung mit geringer Latenz erforderlich ist. Sie können diese Anwendungen für die Ausführung auf Amazon EC2 EC2-Instances ausgeführt werden.
- [Parser-Bibliothek für Kinesis-Videostreams](#)— Ermöglicht Kinesis Video Streams Streams-Anwendungen, Medien zuverlässig und mit geringer Latenz aus dem Kinesis-Videostream abzurufen. Zudem werden die Frame-Grenzen in den Medien analysiert, damit Anwendungen sich auf die Verarbeitung und Analyse der Frames konzentrieren können.

Kinesis Video Streams

Kinesis Video Streams bietet APIs, mit denen Sie Streams erstellen und verwalten sowie Mediendaten in und aus einem Stream lesen oder schreiben können. Die Kinesis Video Streams Streams-Konsole unterstützt neben den Verwaltungsfunktionen auch Live- und video-on-demand Playback-Funktionen. Kinesis Video Streams bietet auch eine Reihe von Produzentenbibliotheken, die Sie in Ihrem Anwendungscode verwenden können, um Daten aus Ihren Medienquellen zu extrahieren und in Ihren Kinesis-Videostream hochzuladen.

Themen

- [Kinesis Video Streams](#)
- [Produzentenbibliotheken](#)

Kinesis Video Streams

Kinesis Video Streams bietet APIs für die Erstellung und Verwaltung von Kinesis Video Streams. Außerdem sind APIs zum Lesen und Schreiben von Daten in Streams verfügbar:

- **Producer-API** — Kinesis Video Streams bietet eine `PutMedia` API zum Schreiben von Mediendaten in einen Kinesis-Videostream. In einer `PutMedia`-Anforderung sendet der Produzent einen Stream von Medienfragmenten. Ein Fragment ist eine abgeschlossene Folge von Frames. Zwischen den Frames eines Fragments und den Frames anderer Fragmente dürfen keine Abhängigkeiten bestehen. Weitere Informationen finden Sie unter [PutMedia](#).

Sobald Fragmente ankommen, weist Kinesis Video Streams eine eindeutige Fragmentnummer zu, und zwar in aufsteigender Reihenfolge. Es speichert auch herstellerseitige und serverseitige Zeitstempel für jedes Fragment als Kinesis Video Streams-spezifische Metadaten.

- **Verbraucher-APIs** — Verbraucher können die folgenden APIs verwenden, um Daten aus einem Stream abzurufen:
 - **GetMedia**: Wenn Konsumenten diese API verwenden, müssen sie das Startfragment identifizieren. Die API gibt anschließend Fragmente in der Reihenfolge zurück, in der sie dem Stream hinzugefügt wurden (in aufsteigender Reihenfolge der Fragmentnummern). Die Mediendaten in den Fragmenten liegen in einem strukturierten Format wie [Matroska \(MKV\)](#) vor. Weitere Informationen finden Sie unter [GetMedia](#).

Note

`GetMedia` weiß, wo sich die Fragmente befinden (archiviert im Datenspeicher oder verfügbar in Echtzeit). Wenn `GetMedia` beispielsweise feststellt, dass das Startfragment archiviert ist, wird das Zurückgeben von Fragmenten aus dem Datenspeicher gestartet. Wenn es neuere Fragmente zurückgeben muss, die noch nicht archiviert sind, wechselt `GetMedia` es zum Lesen von Fragmenten aus einem In-Memory-Stream-Puffer.

Dies ist ein Beispiel für einen kontinuierlichen Konsumenten, der Fragmente in der Reihenfolge verarbeitet, in der sie vom Stream übernommen werden.

Bei Verwendung von `GetMedia` können Anwendungen aus- oder zurückfallen und dann ohne zusätzlichen Aufwand wieder auf den aktuellen Stand gelangen. Mit `GetMedia` können Anwendungen Daten verarbeiten, die im Datenspeicher archiviert sind. Sobald die Anwendung wieder auf dem aktuellen Stand ist, setzt `GetMedia` das Einspeisen von Mediendaten in Echtzeit unmittelbar bei deren Ankunft fort.

- `GetMediaFromFragmentList` (und `ListFragments`): Stapelverarbeitungsanwendungen gelten als Offline-Konsumenten. Offline-Konsumenten können entscheiden, bestimmte

Medienfragmente oder Bereiche von Videodaten explizit abzurufen, indem die APIs `ListFragments` und `GetMediaFromFragmentList` kombiniert werden. Mit `ListFragments` und `GetMediaFromFragmentList` kann eine Anwendung Videosegmente für einen bestimmten Zeitraum oder Fragmentbereich identifizieren und die betreffenden Fragmente dann sequenziell oder parallel zur Verarbeitung abrufen. Diese Herangehensweise eignet sich für MapReduce-Anwendungssuites, die große Datenmengen schnell parallel verarbeiten müssen.

Nehmen wir an, dass ein Konsument die Videofragmente für einen Tag verarbeiten möchte. Der Konsument würde folgendermaßen vorgehen:

1. Liste der Fragmente durch Aufrufen der `ListFragments`-API abrufen und einen Zeitraum angeben, um die gewünschten Fragmente auszuwählen.

Die API gibt die Metadaten aller Fragmente im angegebenen Zeitraum zurück. Die Metadaten enthalten Informationen wie die Fragmentnummer, herstellerseitige und serverseitige Zeitstempel usw.

2. Fragment-Metadatenliste verwenden und Fragmente in beliebiger Reihenfolge abrufen. Um beispielsweise alle Fragmente des Tages zu verarbeiten, könnte sich der Verbraucher dafür entscheiden, die Liste in Unterlisten aufzuteilen und Mitarbeiter (z. B. mehrere Amazon EC2 EC2-Instances) die Fragmente parallel mit der `GetMediaFromFragmentList` abrufen und parallel verarbeiten zu lassen.

Die folgende Abbildung zeigt den Datenfluss für Fragmente und Datenblöcke im Rahmen dieser API-Aufrufe.

Wenn ein Produzent eine `PutMedia`-Anforderung sendet, sendet er die Medienmetadaten in der Nutzlast und dann eine Folge von Mediendatenfragmenten. Nach dem Empfang der Daten speichert Kinesis Video Streams eingehende Mediendaten als Kinesis Video Streams Streams-Chunks. Jeder Datenblock umfasst Folgendes:

- Eine Kopie der Medienmetadaten
- Ein Fragment
- Kinesis Video Streams-spezifische Metadaten, z. B. die Fragmentnummer und serverseitige und herstellerseitige Zeitstempel

Wenn ein Verbraucher Medienmetadaten anfordert, gibt Kinesis Video Streams einen Stream mit Blöcken zurück, beginnend mit der Fragmentnummer, die Sie in der Anfrage angeben.

Wenn Sie die Datenpersistenz für den Stream aktivieren, speichert Kinesis Video Streams nach dem Empfang eines Fragments im Stream auch eine Kopie des Fragments im Datenspeicher.

Produzentenbibliotheken

Nachdem Sie einen Kinesis-Videostream erstellt haben, können Sie damit beginnen, Daten an den Stream zu senden. In Ihrem Anwendungscode können Sie diese Bibliotheken verwenden, um Daten aus Ihren Medienquellen zu extrahieren und in Ihren Kinesis-Videostream hochzuladen. Weitere Informationen zu den verfügbaren Producer-Bibliotheken finden Sie unter [Produzentenbibliotheken von Kinesis Video Streams](#).

Wiedergabe von Kinesis Video Streams

Sie können einen Kinesis-Videostream mit den folgenden Methoden ansehen:

- **GetMedia**— Sie können die `GetMedia` API verwenden, um Ihre eigenen Anwendungen zur Verarbeitung von Kinesis Video Streams zu erstellen. `GetMedia` ist eine Echtzeit-API mit geringer Latenz. Um einen Player zu erstellen, der verwendet `GetMedia`, müssen Sie ihn selbst erstellen. Informationen zur Entwicklung einer Anwendung, mit der ein Kinesis-Videostream angezeigt wird `GetMedia`, finden Sie unter "[Stream Parser](#)"-Bibliothek.
- **HLS** — [HTTP Live Streaming \(HLS\)](#) ist ein dem Industriestandard entsprechendes HTTP-basiertes Kommunikationsprotokoll für Medienstreaming. Sie können HLS verwenden, um einen Kinesis-Videostream anzusehen, entweder für die Live-Wiedergabe oder um archivierte Videos anzusehen.

Sie können HLS zur Live-Wiedergabe verwenden. Die Latenz liegt in der Regel zwischen 3 und 5 Sekunden, kann aber je nach Anwendungsfall, Player und Netzwerkbedingungen auch zwischen 1 und 10 Sekunden liegen. Sie können einen Drittanbieter-Player (z. B. [Video.js](#) oder [Google Shaka Player](#)) zur Ansicht des Videostreams verwenden, indem Sie die URL der HLS-Streaming-Sitzung entweder programmgesteuert oder manuell angeben. Sie können Videos auch abspielen, indem Sie die URL der HLS-Streamingsitzung in die Adressleiste der Browser [Apple Safari](#) oder [Microsoft Edge](#) eingeben.

- **MPEG-DASH** — [Dynamic Adaptive Streaming over HTTP \(DASH\)](#), auch bekannt als MPEG-DASH, ist ein Streaming-Protokoll mit adaptiver Bitrate, das qualitativ hochwertige Streaming

von Medieninhalten über das Internet ermöglicht, die von herkömmlichen HTTP-Webservern bereitgestellt werden.

Sie können MPEG-DASH für die Live-Wiedergabe verwenden. Die Latenz liegt in der Regel zwischen 3 und 5 Sekunden, kann aber je nach Anwendungsfall, Player und Netzwerkbedingungen auch zwischen 1 und 10 Sekunden liegen. Sie können einen Player eines Drittanbieters (wie [dash.js](#) oder [Google Shaka Player](#)) verwenden, um den Videostream anzuzeigen, indem Sie die URL der MPEG-DASH-Streamingsitzung entweder programmgesteuert oder manuell angeben.

- **GetClip**— Sie können die `GetClip` API verwenden, um einen Clip (in einer MP4-Datei) herunterzuladen, der die archivierten On-Demand-Medien aus dem angegebenen Videostream über den angegebenen Zeitraum enthält. Weitere Informationen finden Sie in der [GetClip-API-Referenz](#).

Themen

- [Anforderungen an den Titel für die Videowiedergabe](#)
- [Videowiedergabe mit HLS](#)
- [Videowiedergabe mit MPEG-DASH](#)

Anforderungen an den Titel für die Videowiedergabe

Amazon Kinesis Video Streams unterstützt Medien, die in mehreren Formaten codiert sind. Wenn Ihr Kinesis-Videostream ein Format verwendet, das von einer der vier unten aufgeführten APIs nicht unterstützt wird, verwenden Sie [GetMedia](#) oder [GetMediaForFragmentList](#), da es für diese keine Einschränkungen hinsichtlich des Tracktyps gibt.

Themen

- [GetClip Anforderungen](#)
- [Anforderungen an die GetDash-URL StreamingSession](#)
- [Anforderungen an die URL von StreamingSession GetHLS](#)
- [GetImages Anforderungen](#)

GetClip Anforderungen

Weitere Informationen über diese API finden Sie unter [GetClip](#).

Beschreibung von Track 1	Codec-ID von Track 1	Beschreibung von Track 2	Codec-ID für Track 2
H.264-Video	V_MPEG/ISO/AVC	N/A	N/A
H.264-Video	V_MPEG/ISO/AVC	AAC-Audio	A_AAC
H.264-Video	V_MPEG/ISO/AVC	G.711-Audio (nur A-Law)	A_MS/ACM
H.265-Video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265-Video	V_MPEGH/ISO/HEVC	AAC-Audio	A_AAC

Anforderungen an die GetDash-URL StreamingSession

Weitere Informationen über diese API finden Sie unter [GetDASHStreamingSessionURL](#).

Beschreibung von Track 1	Codec-ID von Track 1	Beschreibung von Track 2	Codec-ID für Track 2
H.264-Video	V_MPEG/ISO/AVC	N/A	N/A
H.264-Video	V_MPEG/ISO/AVC	AAC-Audio	A_AAC
H.264-Video	V_MPEG/ISO/AVC	G.711-Audio (nur A-Law)	A_MS/ACM
H.264-Video	V_MPEG/ISO/AVC	G.711-Audio (nur U-Law)	A_MS/ACM
AAC-Audio	A_AAC	N/A	N/A
H.265-Video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265-Video	V_MPEGH/ISO/HEVC	AAC-Audio	A_AAC

Anforderungen an die URL von StreamingSession GetHLS

Weitere Informationen über diese API finden Sie unter [GetHLSStreamingSessionURL](#).

HLS MP4

Beschreibung von Track 1	Codec-ID von Track 1	Beschreibung von Track 2	Codec-ID für Track 2
H.264-Video	V_MPEG/ISO/AVC	N/A	N/A
H.264-Video	V_MPEG/ISO/AVC	AAC-Audio	A_AAC
AAC-Audio	A_AAC	N/A	N/A
H.265-Video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265-Video	V_MPEGH/ISO/HEVC	AAC-Audio	A_AAC

HLS TS

Beschreibung von Track 1	Codec-ID von Track 1	Beschreibung von Track 2	Codec-ID für Track 2
H.264-Video	V_MPEG/ISO/AVC	N/A	N/A
H.264-Video	V_MPEG/ISO/AVC	AAC-Audio	A_AAC
AAC-Audio	A_AAC	N/A	N/A

GetImages Anforderungen

Weitere Informationen über diese API finden Sie unter [GetImages](#).

Note

Das GetImages Medium sollte in Track 1 eine Videospur enthalten.

Videowiedergabe mit HLS

[HTTP Live Streaming \(HLS\)](#) ist ein dem Industriestandard entsprechendes HTTP-basiertes Kommunikationsprotokoll für Medienstreaming. Sie können HLS verwenden, um einen Kinesis-Videostream anzusehen, entweder für die Live-Wiedergabe oder um archivierte Videos anzusehen.

Sie können HLS zur Live-Wiedergabe verwenden. Die Latenz liegt in der Regel zwischen 3 und 5 Sekunden, kann aber je nach Anwendungsfall, Player und Netzwerkbedingungen auch zwischen 1 und 10 Sekunden liegen. Sie können einen Drittanbieter-Player (z. B. [Video.js](#) oder [Google Shaka Player](#)) zur Ansicht des Videostreams verwenden, indem Sie die URL der HLS-Streaming-Sitzung entweder programmgesteuert oder manuell angeben. Sie können Videos auch abspielen, indem Sie die URL der HLS-Streamingsitzung in die Adressleiste der Browser [Apple Safari](#) oder [Microsoft Edge](#) eingeben.

Um einen Kinesis-Videostream mit HLS anzusehen, erstellen Sie zunächst eine Streaming-Sitzung mit der [StreamingSessionGetHLS-URL](#). Diese Aktion gibt eine URL (mit einem Sitzungs-Token) für den Zugriff auf die HLS-Sitzung zurück. Anschließend können Sie die URL in einen Media Player oder einer eigenständigen Anwendung zum Anzeigen des Streams verwenden.

Important

Nicht alle an Kinesis Video Streams gesendeten Medien können über HLS wiedergegeben werden. Spezifische [the section called "GetHLSStreamingSessionURL"](#) Upload-Anforderungen finden Sie unter.

Themen

- [Verwenden Sie die AWS CLI , um die URL einer HLS-Streaming-Sitzung abzurufen](#)
- [Beispiel: Verwenden Sie HLS in HTML und JavaScript](#)
- [Behebung von HLS-Problemen](#)

Verwenden Sie die AWS CLI , um die URL einer HLS-Streaming-Sitzung abzurufen

Gehen Sie wie folgt vor, um mithilfe der AWS CLI eine HLS-Streaming-Sitzungs-URL für einen Kinesis-Videostream zu generieren.

Installationsanweisungen finden Sie im [AWS Command Line Interface Benutzerhandbuch](#). Nach der Installation [konfigurieren Sie das AWS CLI mit den](#) Anmeldeinformationen und der Region.

Öffnen Sie alternativ das AWS CloudShell Terminal, auf dem das AWS CLI installiert und konfiguriert ist. Weitere Informationen finden Sie im [AWS CloudShell -Benutzerhandbuch](#).

Rufen Sie den HLS-URL-Endpunkt für Ihren Kinesis-Videostream ab.

1. Geben Sie Folgendes in das Terminal ein:

```
aws kinesisisvideo get-data-endpoint \  
  --api-name GET_HLS_STREAMING_SESSION_URL \  
  --stream-name YourStreamName
```

Sie erhalten eine Antwort, die wie folgt aussieht:

```
{  
  "DataEndpoint": "https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com"  
}
```

2. Stellen Sie die URL-Anfrage für die HLS-Streaming-Sitzung an den zurückgegebenen Endpunkt.

Live

Bei der Live-Wiedergabe wird die HLS-Medien-Playlist kontinuierlich mit den neuesten Medien aktualisiert, sobald diese verfügbar sind. Wenn Sie diese Art von Sitzung in einem Media Player abspielen, zeigt die Benutzeroberfläche in der Regel eine Live-Benachrichtigung an, ohne dass Sie die anzuzeigende Position im Wiedergabefenster auswählen können.

Stellen Sie sicher, dass Sie Medien in diesen Stream hochladen, wenn Sie diesen Befehl ausführen.

```
aws kinesisisvideo get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode LIVE
```

Live replay

Bei der Live-Wiedergabe beginnt die Wiedergabe ab einer bestimmten Startzeit. Die HLS-Medien-Playlist wird außerdem ständig mit den neuesten Medien aktualisiert, sobald diese verfügbar sind. Die Sitzung enthält weiterhin neu aufgenommene Medien, bis die Sitzung

abläuft oder bis zur angegebenen Endzeit, je nachdem, was zuerst eintritt. Dieser Modus ist nützlich, um die Wiedergabe ab dem Zeitpunkt starten zu können, an dem ein Ereignis erkannt wird, und das Live-Streaming von Medien fortsetzen zu können, die zum Zeitpunkt der Sitzungserstellung noch nicht aufgenommen wurden.

Ermitteln Sie einen Startzeitstempel.

In diesem Beispiel verwenden wir das Format Unix Epoch Time in Sekunden. Weitere Informationen zur [Formatierung von Zeitstempeln](#) finden Sie im Abschnitt Zeitstempel im AWS Command Line Interface Benutzerhandbuch.

Auf [UnixTime.org](https://unixtime.org) finden Sie ein Konvertierungstool.

- 1708471800 entspricht dem 20. Februar 2024, 15:30:00 Uhr GMT- 08:00

In diesem Beispiel geben wir keinen Endzeitstempel an, was bedeutet, dass die Sitzung weiterhin neu aufgenommene Medien enthält, bis die Sitzung abläuft.

Rufen Sie die `GetHLSStreamingSessionURL` API mit dem angegebenen `LIVE_REPLAY` Wiedergabemodus und einem angegebenen [HLS](#) Fragment Selector auf.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode LIVE_REPLAY \  
  --hls-fragment-selector \  
  
"FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800}"
```

On-demand

Für die On-Demand-Wiedergabe enthält die HLS-Medien-Playlist die Medien, die mit der HLS-Fragmentauswahl angegeben wurden. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel ein Steuerelement an, mit dem Sie die anzuzeigende Position im Wiedergabefenster auswählen können.

Um eine URL für einen bestimmten Abschnitt des Streams zu erstellen, legen Sie zunächst die Start- und Endzeitstempel fest.


In diesem Beispiel verwenden wir das Format Unix-Epochenzeit im Sekundenformat. Weitere Informationen zur [Formatierung von Zeitstempeln](#) finden Sie im Abschnitt Zeitstempel im AWS Command Line Interface Benutzerhandbuch.

Auf [UnixTime.org](https://unixtime.org) finden Sie ein Konvertierungstool.

- 1708471800 entspricht dem 20. Februar 2024, 15:30:00 Uhr GMT- 08:00
- 1708471860 entspricht 20. Februar 2024 15:31:00 Uhr GMT- 08:00

[Rufen Sie die GetHLSStreamingSessionURL API mit dem angegebenen ON_DEMAND Wiedergabemodus und einem angegebenen HLS Fragment Selector auf.](#)

```
aws kinesis-video-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode ON_DEMAND \  
  --hls-fragment-selector \  
  
"FragmentSelectorType=SERVER_TIMESTAMP,TimeStampRange={StartTimestamp=1708471800,EndTim
```

 Note

Die Zeitstempel müssen nicht weiter als 24 Stunden voneinander entfernt sein, wie in der Dokumentation beschrieben. [the section called "HLSTimestampRange"](#)

Sie erhalten eine Antwort, die wie folgt aussieht:

```
{  
  "HLSStreamingSessionURL": "https://b-1234abcd.kinesisvideo.aws-  
region.amazonaws.com/hls/v1/getHLSMasterPlaylist.m3u8?SessionToken=CiAz...DkRE6M~"  
}
```

 Important

Teilen oder speichern Sie dieses Token nicht an einem Ort, an dem eine nicht autorisierte Entität darauf zugreifen könnte. Das Token ermöglicht den Zugriff auf den

Inhalt des Streams. Schützen Sie das Token mit denselben Maßnahmen, die Sie für Ihre AWS Anmeldeinformationen verwenden würden.

Sie können diese URL und jeden HLS-Player verwenden, um den HLS-Stream anzusehen.

Verwenden Sie beispielsweise den VLC Media Player.

Sie können den HLS-Stream auch abspielen, indem Sie die URL der HLS-Streaming-Sitzung in die Adressleiste der Apple Safari- oder Microsoft Edge-Browser eingeben.

Beispiel: Verwenden Sie HLS in HTML und JavaScript

Das folgende Beispiel zeigt, wie Sie mit dem AWS SDK für JavaScript v2 eine HLS-Streaming-Sitzung für einen Kinesis-Videostream abrufen und auf einer Webseite wiedergeben können. Das Beispiel veranschaulicht die Wiedergabe von Videos in den folgenden Playern:

- [Video.js](#)
- [Google Shaka Player](#)
- [hls.js](#)

Den [vollständigen Beispielcode](#) und die [gehostete Webseite finden](#) Sie unter. GitHub

Erläuterung der folgenden Themen durch den Code:

- [Importieren Sie das AWS SDK JavaScript für Browser](#)
- [Den Kinesis Video Streams Streams-Client einrichten](#)
- [Rufen Sie den Endpunkt für die HLS-Wiedergabe ab](#)
- [Den Kinesis Video Streams Streams-Client für archivierte Medien einrichten](#)
- [Rufen Sie die URL der HLS-Streaming-Sitzung ab](#)
- [Zeigen Sie den HLS-Stream auf der Webseite an](#)

Importieren Sie das AWS SDK JavaScript für Browser

Fügen Sie auf der Webseite das folgende Skript-Tag hinzu, um das AWS SDK für JavaScript Version 2 in das Projekt zu importieren.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/aws-sdk/2.490.0/aws-sdk.min.js"></script>
```

Weitere Informationen finden Sie in der JavaScript Dokumentation [zum AWS SDK](#).

Den Kinesis Video Streams Streams-Client einrichten

Um mit HLS auf Streaming-Video zuzugreifen, müssen Sie zunächst den Kinesis Video Streams Streams-Client erstellen und konfigurieren. Weitere Authentifizierungsmethoden finden Sie unter [Anmeldeinformationen in einem Webbrowser einrichten](#).

```
const clientConfig = {
  accessKeyId: 'YourAccessKey',
  secretAccessKey: 'YourSecretKey',
  region: 'us-west-2'
};
const kinesisVideoClient = new AWS.KinesisVideo(clientConfig);
```

Die Anwendung ruft die erforderlichen Werte aus Eingabefeldern auf der HTML-Seite ab.

Rufen Sie den Endpunkt für die HLS-Wiedergabe ab

Verwenden Sie den Kinesis Video Streams Streams-Client, um die [the section called "GetDataEndpoint"](#) API aufzurufen, um den Endpunkt abzurufen.

```
const getDataEndpointOptions = {
  StreamName: 'YourStreamName',
  APIName: 'GET_HLS_STREAMING_SESSION_URL'
};
const getDataEndpointResponse = await kinesisVideoClient
  .getDataEndpoint(getDataEndpointOptions)
  .promise();
const hlsDataEndpoint = getDataEndpointResponse.DataEndpoint;
```

Dieser Code speichert den Endpunkt in der `hlsDataEndpoint` Variablen.

Den Kinesis Video Streams Streams-Client für archivierte Medien einrichten

Geben Sie in der Client-Konfiguration für den Kinesis Video Streams Streams-Client für archivierte Medien den Endpunkt an, den Sie im vorherigen Schritt abgerufen haben.

```
const archivedMediaClientConfig = {
```

```
    accessKeyId: 'YourAccessKey',
    secretAccessKey: 'YourSecretKey',
    region: 'us-west-2',
    endpoint: hlsDataEndpoint
  }
const kinesisVideoArchivedMediaClient = new
  AWS.KinesisVideoArchivedMedia(archivedMediaClientConfig);
```

Rufen Sie die URL der HLS-Streaming-Sitzung ab

Verwenden Sie den Kinesis Video Streams Streams-Client für archivierte Medien, um die [the section called "GetHLSStreamingSessionURL"](#) API zum Abrufen der HLS-Wiedergabe-URL aufzurufen.

```
const getHLSStreamingSessionURLOptions = {
  StreamName: 'YourStreamName',
  PlaybackMode: 'LIVE'
};
const getHLSStreamingSessionURLResponse = await
  kinesisVideoArchivedMediaClient.getHLSStreamingSessionURL(getHLSStreamingSessionURLOptions);
const hlsUrl = getHLSStreamingSessionURLResponse.HLSStreamingSessionURL;
```

Zeigen Sie den HLS-Stream auf der Webseite an

Wenn Sie die HLS-Streaming-Sitzungs-URL erhalten haben, stellen Sie sie für den Video-Player bereit. Die Methode, mit der die URL für den Video-Player bereitgestellt wird, ist vom jeweils verwendeten Player abhängig.

Video.js

Gehen Sie wie folgt vor, um [Video.js](#) und seine CSS-Klassen in unser Browser-Skript zu importieren:

```
<link rel="stylesheet" href="https://vjs.zencdn.net/6.6.3/video-js.css">
<script src="https://vjs.zencdn.net/6.6.3/video.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/videojs-contrib-hls/5.14.1/
videojs-contrib-hls.js"></script>
```

Erstellen Sie ein video HTML-Element, um das Video anzuzeigen:

```
<video id="videojs" class="player video-js vjs-default-skin" controls autoplay></
video>
```

Stellen Sie die HLS-URL als Quelle für das HTML-Videoelement ein:

```
const playerElement = document.getElementById('videojs');
const player = videojs(playerElement);
player.src({
  src: hlsUrl,
  type: 'application/x-mpegURL'
});
player.play();
```

Shaka

Gehen Sie wie folgt vor, um den [Google Shaka-Player](#) in unser Browserskript zu importieren:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js"></script>
```

Erstellen Sie ein video HTML-Element, um das Video anzuzeigen:

```
<video id="shaka" class="player" controls autoplay></video>
```

Erstellen Sie einen Shaka-Player, der das Videoelement angibt, und rufen Sie die Lademethode auf:

```
const playerElement = document.getElementById('shaka');
const player = new shaka.Player(playerElement);
player.load(hlsUrl);
```

hls.js

Gehen Sie wie folgt vor, um [hls.js](#) in unser Browser-Skript zu importieren:

```
<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
```

Erstellen Sie ein video HTML-Element, um das Video anzuzeigen:

```
<video id="hlsjs" class="player" controls autoplay></video>
```

Erstellen Sie einen Player mit der Datei hls.js, geben Sie ihm die HLS-URL und weisen Sie ihn an, dass er abgespielt werden soll:


```
var playerElement = document.getElementById('hlsjs');
var player = new Hls();
player.loadSource(hlsUrl);
player.attachMedia(playerElement);
player.on(Hls.Events.MANIFEST_PARSED, function() {
    video.play();
});
```

Behebung von HLS-Problemen

In diesem Abschnitt werden Probleme beschrieben, die bei der Verwendung von HTTP Live Streaming (HLS) mit Kinesis Video Streams auftreten können.

Problembereiche

- [Abrufen der HLS-Streaming-Sitzungs-URL erfolgreich, aber Wiedergabe im Video-Player nicht möglich](#)
- [Latenz zwischen Produzent und Player zu hoch](#)

Abrufen der HLS-Streaming-Sitzungs-URL erfolgreich, aber Wiedergabe im Video-Player nicht möglich

Diese Situation tritt auf, wenn Sie eine HLS-Streaming-Sitzungs-URL mit `GetHLSStreamingSessionURL` erfolgreich abrufen können, das Video aber nicht abgespielt wird, wenn die URL einem Video-Player bereitgestellt wird.

Um das Problem zu beheben, versuchen Sie Folgendes:

- Stellen Sie fest, ob der Videostream in der Kinesis Video Streams Streams-Konsole wiedergegeben wird. Ziehen Sie alle in der Konsole angezeigten Fehler in Betracht.
- Wenn die Fragmentdauer weniger als eine Sekunde beträgt, erhöhen Sie sie auf eine Sekunde. Wenn die Fragmentdauer zu kurz ist, drosselt der Dienst möglicherweise den Player, weil er zu häufig Videofragmente anfordert.
- Stellen Sie sicher, dass jede HLS-Streaming-Sitzungs-URL von nur einem Player verwendet wird. Wenn eine einzelne HLS-Streaming-Sitzungs-URL von mehr als einem Player verwendet wird, empfängt der Service möglicherweise zu viele Videoanforderungen und drosselt sie.
- Stellen Sie sicher, dass Ihr Player alle Optionen unterstützt, die Sie für die HLS-Streaming-Sitzung angeben. Probieren Sie verschiedene Kombinationen von Werten für die folgenden Parameter aus:

- `ContainerFormat`
- `PlaybackMode`
- `FragmentSelectorType`
- `DiscontinuityMode`
- `MaxMediaPlaylistFragmentResults`

Einige Media Player (z. B. HTML5 und mobile Media Player) unterstützen in der Regel nur HLS mit dem fMP4-Containerformat. Andere Media Player (wie Flash und benutzerdefinierte Player) unterstützen HLS möglicherweise nur mit dem MPEG TS-Containerformat. Wir empfehlen, mit dem `ContainerFormat` Parameter zu experimentieren, um mit der Problembehandlung zu beginnen.

- Stellen Sie sicher, dass jedes Fragment über eine konsistente Anzahl von Tracks verfügt. Stellen Sie sicher, dass sich Fragmente im Stream nicht ändern, wenn Sie sowohl eine Audio- und Videospur als auch nur eine Videospur haben. Stellen Sie außerdem sicher, dass sich die Encoder-Einstellungen (Auflösung und Bildrate) nicht zwischen den einzelnen Fragmenten in den einzelnen Spuren ändern.

Latenz zwischen Produzent und Player zu hoch

Diese Situation tritt auf, wenn die Latenz ab dem Zeitpunkt, an dem das Video erfasst wurde, bis zu dem Zeitpunkt, an dem es im Video-Player abgespielt wird, zu hoch ist.

Das Video wird über HLS fragmentweise wiedergegeben. Daher darf die Latenz nicht kürzer als die Fragmentdauer sein. Die Latenz umfasst auch die für die Pufferung und Übertragung von Daten benötigte Zeit. Wenn Ihre Lösung weniger als eine Sekunde Latenz erfordert, empfehlen wir, stattdessen die `GetMedia-API` zu verwenden.

Sie können die folgenden Parameter können so anpassen, dass die Latenz insgesamt verringert wird. Eine Anpassung dieser Parameter kann jedoch zur Folge haben, dass sich die Videoqualität verringert oder die Neupufferungsrate erhöht.

- **Fragmentdauer** — Die Fragmentdauer ist die Menge an Video zwischen den einzelnen Abschnitten im Stream, die durch die Häufigkeit der vom Video-Encoder generierten Keyframes gesteuert wird. Der empfohlene Wert ist eine Sekunde. Eine kürzere Fragmentdauer bedeutet, dass weniger lange auf den Abschluss des Fragments gewartet wird, bevor die Videodaten zum Service übertragen werden. Außerdem kann der Service kürzere Fragmente schneller verarbeiten. Wenn die Fragmentdauer zu kurz ist, nimmt jedoch die Wahrscheinlichkeit zu, dass dem Player die Inhalte ausgehen und er anhalten und Inhalte puffern muss. Wenn die Fragmentdauer weniger als

500 Millisekunden beträgt, erstellt der Produzent möglicherweise zu viele Anforderungen, die dann vom Service gedrosselt werden.

- **Bitrate** — Ein Videostream mit einer niedrigeren Bitrate benötigt weniger Zeit zum Lesen, Schreiben und Übertragen. Ein Videostream mit einer niedrigeren Bitrate verfügt in der Regel jedoch über eine geringere Videoqualität.
- **Anzahl der Fragmente in Medien-Playlisten** — Ein Player, der latenzempfindlich ist, sollte nur die neuesten Fragmente in eine Medien-Playlist laden. Die meisten Spieler beginnen stattdessen mit dem frühesten Fragment. Indem Sie die Anzahl der Fragmente in der Playlist reduzieren, verringern Sie den Zeitabstand zwischen dem vorherigen und dem neuen Fragment. Bei einer kleineren Playlist kann es sein, dass ein Fragment bei der Wiedergabe übersprungen wird, wenn es zu Verzögerungen beim Hinzufügen neuer Fragmente zur Playlist kommt oder wenn der Player verzögert eine aktualisierte Playlist erhält. Wir empfehlen, 3—5 Fragmente zu verwenden und einen Player zu verwenden, der so konfiguriert ist, dass er nur die neuesten Fragmente aus einer Playlist lädt.
- **Puffergröße des Players** — Die meisten Videoplayer haben eine konfigurierbare Mindestpufferdauer, normalerweise mit einer Standardeinstellung von 10 Sekunden. Für die niedrigste Latenz können Sie diesen Wert auf 0 Sekunden setzen. Dies bedeutet jedoch, dass der Player zurückweist, wenn es zu Verzögerungen kommt, wodurch Fragmente entstehen, da der Player dann keinen Puffer hat, um die Verzögerung zu absorbieren.
- **Spieler-Aufholjagd** — Videoplayer catch die Wiedergabe normalerweise nicht automatisch bis zum Anfang des Videopuffers auf, wenn der Puffer voll wird, z. B. wenn ein verzögertes Fragment zu einem Rückstau von Fragmenten führt. Um dies zu vermeiden, kann ein benutzerdefinierter Player Frames verwerfen oder die Wiedergabegeschwindigkeit erhöhen (z. B. auf 1,1-fach), um bis zum Anfang des Puffers aufzuholen. Dies führt zu einer abgehackten Wiedergabe oder einer Zunahme der Geschwindigkeit, während der Player aufholt, und die erneute Pufferung kann häufiger erfolgen, da die Puffergröße gering gehalten wird.

Videowiedergabe mit MPEG-DASH

[Um einen Kinesis-Videostream mit MPEG-DASH anzusehen, erstellen Sie zunächst eine Streaming-Sitzung mit der GetDash-URL. `StreamingSession`](#) Diese Aktion gibt eine URL (mit einem Sitzungstoken) für den Zugriff auf die MPEG-DASH-Sitzung zurück. Anschließend können Sie die URL in einen Media Player oder einer eigenständigen Anwendung zum Anzeigen des Streams verwenden.

Für einen Amazon Kinesis Kinesis-Videostream gelten die folgenden Anforderungen für die Bereitstellung von Video über MPEG-DASH:

- Informationen zu den Anforderungen für die Wiedergabe von Streaming-Videotracks finden Sie unter [the section called “GetDash-URL StreamingSession”](#)
- Der Datenerhaltzeitraum muss größer als 0 sein.
- Die Videospur jedes Fragments muss private Codec-Daten in Advanced Video Coding (AVC) für H.264 und HEVC für H.265 enthalten. Weitere Informationen finden Sie unter [MPEG-4-Spezifikation ISO/IEC 14496-15](#). Informationen zur Adaptierung von Stream-Daten für ein bestimmtes Format finden Sie unter [NAL Adaptation Flags](#).
- Die Audiospur (falls vorhanden) jedes Fragments muss private Codec-Daten im AAC-Format ([AAC-Spezifikation ISO/IEC 13818-7](#)) oder dem [MS Wave-Format](#) enthalten.

Beispiel: Verwendung von MPEG-DASH in HTML und JavaScript

Das folgende Beispiel zeigt, wie eine MPEG-DASH-Streaming-Sitzung für einen Kinesis-Videostream abgerufen und auf einer Webseite wiedergegeben wird. Das Beispiel veranschaulicht die Wiedergabe von Videos in den folgenden Playern:

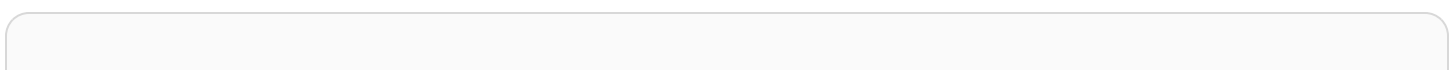
- [Google Shaka Player](#)
- [dash.js](#)

Themen

- [Den Kinesis Video Streams Client für die MPEG-DASH-Wiedergabe einrichten](#)
- [Rufen Sie den Kinesis Video Streams Streams-Endpunkt für archivierte Inhalte für die MPEG-DASH-Wiedergabe ab](#)
- [Abrufen der MPEG-DASH-Streaming-Sitzungs-URL](#)
- [Anzeigen des Streaming-Videos mit MPEG-DASH-Wiedergabe](#)
- [Ausgeführtes Beispiel](#)

Den Kinesis Video Streams Client für die MPEG-DASH-Wiedergabe einrichten

Um mit MPEG-DASH auf Streaming-Video zuzugreifen, erstellen und konfigurieren Sie zunächst den Kinesis Video Streams Streams-Client (um den Service-Endpunkt abzurufen) und den archivierten Media-Client (um die MPEG-DASH-Streaming-Sitzung abzurufen). Die Anwendung ruft die erforderlichen Werte aus Eingabefeldern auf der HTML-Seite ab.



```
var streamName = $('#streamName').val();

// Step 1: Configure SDK Clients
var options = {
  accessKeyId: $('#accessKeyId').val(),
  secretAccessKey: $('#secretAccessKey').val(),
  sessionToken: $('#sessionToken').val() || undefined,
  region: $('#region').val(),
  endpoint: $('#endpoint').val() || undefined
}
var kinesisVideo = new AWS.KinesisVideo(options);
var kinesisVideoArchivedContent = new AWS.KinesisVideoArchivedMedia(options);
```

Rufen Sie den Kinesis Video Streams Streams-Endpunkt für archivierte Inhalte für die MPEG-DASH-Wiedergabe ab

Rufen Sie nach der Initiierung der Clients den Endpunkt für archivierte Inhalte von Kinesis Video Streams ab, sodass Sie die URL der MPEG-DASH-Streaming-Sitzung wie folgt abrufen können:

```
// Step 2: Get a data endpoint for the stream
console.log('Fetching data endpoint');
kinesisVideo.getDataEndpoint({
  StreamName: streamName,
  APIName: "GET_DASH_STREAMING_SESSION_URL"
}, function(err, response) {
  if (err) { return console.error(err); }
  console.log('Data endpoint: ' + response.DataEndpoint);
  kinesisVideoArchivedContent.endpoint = new AWS.Endpoint(response.DataEndpoint);
```

Abrufen der MPEG-DASH-Streaming-Sitzungs-URL

Wenn Sie den Endpunkt für archivierte Inhalte haben, rufen Sie die [StreamingSessionGetDash-URL-API](#) auf, um die URL der MPEG-DASH-Streaming-Sitzung wie folgt abzurufen:

```
// Step 3: Get a Streaming Session URL
var consoleInfo = 'Fetching ' + protocol + ' Streaming Session URL';
console.log(consoleInfo);

if (protocol === 'DASH') {
  kinesisVideoArchivedContent.getDASHStreamingSessionURL({
```

```

StreamName: streamName,
PlaybackMode: $('#playbackMode').val(),
DASHFragmentSelector: {
  FragmentSelectorType: $('#fragmentSelectorType').val(),
  TimestampRange: $('#playbackMode').val() === "LIVE" ? undefined : {
    StartTimestamp: new Date($('#startTimestamp').val()),
    EndTimestamp: new Date($('#endTimestamp').val())
  }
},
DisplayFragmentTimestamp: $('#displayFragmentTimestamp').val(),
DisplayFragmentNumber: $('#displayFragmentNumber').val(),
MaxManifestFragmentResults: parseInt($('#maxResults').val()),
Expires: parseInt($('#expires').val())
}, function(err, response) {
  if (err) { return console.error(err); }
  console.log('DASH Streaming Session URL: ' + response.DASHStreamingSessionURL);
}

```

Anzeigen des Streaming-Videos mit MPEG-DASH-Wiedergabe

Wenn Sie die MPEG-DASH-Streaming-Sitzungs-URL erhalten haben, stellen Sie sie für den Video-Player bereit. Die Methode, mit der die URL für den Video-Player bereitgestellt wird, ist vom jeweils verwendeten Player abhängig.

Das folgende Codebeispiel veranschaulicht, wie die Streaming-Sitzungs-URL für einen [Google Shaka](#)-Player bereitgestellt wird:

```

// Step 4: Give the URL to the video player.

//Shaka Player elements
<video id="shaka" class="player" controls autoplay></video>
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js">
</script>
...

var playerName = $('#player').val();

if (playerName === 'Shaka Player') {
  var playerElement = $('#shaka');
  playerElement.show();
}

```

```
var player = new shaka.Player(playerElement[0]);
console.log('Created Shaka Player');

player.load(response.DASHStreamingSessionURL).then(function() {
    console.log('Starting playback');
});
console.log('Set player source');
}
```

Das folgende Codebeispiel veranschaulicht, wie die Streaming-Sitzungs-URL für einen [dash.js](#)-Player bereitgestellt wird:

```
<!-- dash.js Player elements -->
<video id="dashjs" class="player" controls autoplay=""></video>
<script src="https://cdn.dashjs.org/latest/dash.all.min.js"></script>

...

var playerElement = $('#dashjs');
playerElement.show();

var player = dashjs.MediaPlayer().create();
console.log('Created DASH.js Player');

player.initialize(document.querySelector('#dashjs'), response.DASHStreamingSessionURL,
    true);
console.log('Starting playback');
console.log('Set player source');
}
```

Ausgeführtes Beispiel

Sie können den [fertigen Beispielcode unter herunterladen oder ansehen](#). GitHub

Verwenden von Streaming-Metadaten mit Kinesis Video Streams

Sie können das Amazon Kinesis Video Streams Producer SDK verwenden, um Metadaten auf der Ebene einzelner Fragmente in einen Kinesis-Videostream einzubetten. Metadaten in Kinesis Video Streams sind ein veränderbares Schlüssel-Wert-Paar. Sie können es verwenden, um den Inhalt des Fragments zu beschreiben, zugehörige Sensormesswerte einzubetten, die zusammen

mit dem eigentlichen Fragment übertragen werden müssen, oder um andere benutzerdefinierte Anforderungen zu erfüllen. Die Metadaten werden als Teil der API-Operationen [the section called “GetMedia”](#) oder [the section called “GetMediaForFragmentList”](#) zur Verfügung gestellt. Es wird zusammen mit den Fragmenten für die gesamte Dauer der Aufbewahrungszeit des Streams gespeichert. Ihre nutzenden Anwendungen können anhand der Metadaten lesen, verarbeiten und darauf reagieren [Parser-Bibliothek für Kinesis-Videostreams](#).

Es gibt zwei Modi, in denen die Metadaten in Fragmente in einem Stream eingebettet werden können:

- Nicht persistent — Sie können Metadaten einmalig oder ad hoc an Fragmente in einem Stream anhängen, je nachdem, welche geschäftsspezifischen Kriterien erfüllt wurden. Ein Beispiel ist eine Smart-Kamera, die Bewegungen erkennt und Metadaten zu den entsprechenden Fragmenten hinzufügt, die die Bewegung enthalten, bevor sie die Fragmente an ihren Kinesis-Videostream sendet. Sie können Metadaten im folgenden Format auf das Fragment anwenden: `Motion = true`.
- Persistent — Sie können je nach Bedarf Metadaten an aufeinanderfolgende Fragmente in einem Stream anhängen. Ein Beispiel ist eine Smart-Kamera, die die aktuellen Breiten- und Längengradkoordinaten aller Fragmente, die sie sendet, an ihren Kinesis-Videostream sendet. Sie können Metadaten im folgenden Format auf alle Fragmente anwenden: `Lat = 47.608013N , Long = -122.335167W`.

Sie können Metadaten je nach den Anforderungen Ihrer Anwendung in beiden Modi gleichzeitig an dasselbe Fragment binden. Die eingebetteten Metadaten können erkannte Objekte, nachverfolgte Aktivitäten, GPS-Koordinaten oder beliebige andere benutzerdefinierte Daten enthalten, die Sie an den Fragmenten im Stream zuordnen möchten. Metadaten sind als Schlüssel-Werte-Paare codiert.

Themen

- [Hinzufügen von Metadaten zu einem Kinesis-Videostream](#)
- [Nutzung von Metadaten, die in einen Kinesis-Videostream eingebettet sind](#)
- [Beschränkungen für Streaming-Metadaten](#)

Hinzufügen von Metadaten zu einem Kinesis-Videostream

Metadaten, die Sie einem Kinesis-Videostream hinzufügen, werden als MKV-Tags modelliert, die als Schlüssel-Wert-Paare implementiert werden.

Metadaten können entweder transient sein, wie z. B. zur Markierung eines Elements innerhalb des Streams, oder persistent, wie z. B. zum Identifizieren von Fragmenten, bei denen ein bestimmtes Ereignis stattfindet. Ein persistentes Metadatenelement bleibt bestehen und wird auf jedes aufeinanderfolgende Fragment angewendet, bis es gelöscht wird.

Note

Die über das [Producer-Bibliotheken](#) hinzugefügten Metadatenelemente unterscheiden sich von den markierenden APIs auf Stream-Ebene, die mit [the section called "TagStream"](#), [the section called "UntagStream"](#) und [the section called "ListTagsForStream"](#) implementiert werden.

API für Streaming-Metadaten

Sie können die folgenden Operationen im Producer SDK zur Implementierung von Streaming-Metadaten verwenden.

PIC

```
PUBLIC_API STATUS putKinesisVideoFragmentMetadata(STREAM_HANDLE streamHandle,
    PCHAR name,
    PCHAR value,
    BOOL persistent);
```

C++ Producer SDK

```
/**
 * Appends a "tag" or metadata - a key/value string pair into the stream.
 */
bool putFragmentMetadata(const std::string& name, const std::string& value, bool
    persistent = true);
```

Java Producer SDK

Sie können das Java Producer SDK verwenden, um Metadaten zu einer MediaSource Anwendung hinzuzufügen `MediaSourceSink.onCodecPrivateData`:

```
void onFragmentMetadata(final @NonNull String metadataName, final @NonNull String
    metadataValue, final boolean persistent)
```

```
throws KinesisVideoException;
```

Persistente und nicht persistente Metadaten

Für nicht persistente Metadaten können Sie mehrere Metadatenelemente mit demselben Namen hinzufügen. Das Producer SDK sammelt die Metadatenelemente in der Metadaten-Warteschlange, bis sie dem nächsten Fragment vorangestellt werden. Die Metadaten-Warteschlange wird gelöscht, während Metadatenelemente auf den Stream angewendet werden. Um die Metadaten zu wiederholen, rufen Sie `putKinesisVideoFragmentMetadata` oder `putFragmentMetadata` erneut auf.

Das Producer SDK geht zur Sammlung der Metadatenelemente in der Metadaten-Warteschlange bei persistenten Metadaten auf die gleiche Weise vor wie bei nicht persistenten Metadaten. Die Metadatenelemente werden jedoch nicht aus der Warteschlange entfernt, wenn sie dem nächsten Fragment vorangestellt werden.

Wenn `putKinesisVideoFragmentMetadata` oder `putFragmentMetadata` mit `persistent` auf `true` eingestellt aufgerufen wird, bewirkt dies das folgende Verhalten:

- Durch Aufrufen der API wird das Metadatenelement in die Warteschlange eingefügt. Die Metadaten werden jedem Fragment als MKV-Tag hinzugefügt, während sich das Element in der Warteschlange befindet.
- Wenn die API mit demselben Namen und einem anderen Wert als ein zuvor hinzugefügtes Metadatenelement aufgerufen wird, wird das Element überschrieben.
- Wenn die API mit einem leeren Wert aufgerufen wird, wird das Metadatenelement aus der Metadaten-Warteschlange entfernt (storniert).

Nutzung von Metadaten, die in einen Kinesis-Videostream eingebettet sind

Um die Metadaten in einem Kinesis-Videostream zu nutzen, verwenden Sie eine Implementierung von `MkvTagProcessor`:

```
public interface MkvTagProcessor {
    default void process(MkvTag mkvTag, Optional<FragmentMetadata>
        currentFragmentMetadata) {
        throw new NotImplementedException("Default
        FragmentMetadataVisitor.MkvTagProcessor");
    }
}
```

```

    }
    default void clear() {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
}
}
}

```

Diese Schnittstelle ist in der Klasse [FragmentMetadataVisitor](#) in der [Parser-Bibliothek für Kinesis-Videostreams](#) zu finden.

Die Klasse `FragmentMetadataVisitor` enthält eine Implementierung von `MkvTagProcessor`:

```

public static final class BasicMkvTagProcessor implements
FragmentMetadataVisitor.MkvTagProcessor {
    @Getter
    private List<MkvTag> tags = new ArrayList<>();

    @Override
    public void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        tags.add(mkvTag);
    }

    @Override
    public void clear() {
        tags.clear();
    }
}
}

```

Die Klasse `KinesisVideoRendererExample` enthält ein Beispiel zur Verwendung eines `BasicMkvTagProcessor`. Im folgenden Beispiel wird ein `BasicMkvTagProcessor` zu den `MediaProcessingArguments` einer Anwendung hinzugefügt:

```

if (renderFragmentMetadata) {
    getMediaProcessingArguments =
KinesisVideoRendererExample.GetMediaProcessingArguments.create(
        Optional.of(new FragmentMetadataVisitor.BasicMkvTagProcessor()));
}

```

Die Methode `BasicMkvTagProcessor.process` wird aufgerufen, wenn Fragment-Metadaten eintreffen. Sie können die angesammelten Metadaten mit `GetTags` abrufen. Um ein einzelnes

Metadatenelement abzurufen, rufen Sie zuerst auf, `clear` um die gesammelten Metadaten zu löschen, und rufen Sie dann die Metadatenelemente erneut ab.

Beschränkungen für Streaming-Metadaten

Weitere Informationen zu den Beschränkungen, die [the section called “Fragment-Metadatenkontingente”](#) für das Hinzufügen von Streaming-Metadaten zu einem Kinesis-Videostream gelten, finden Sie unter

Kinesis Video Streams Streams-Datenmodell

Das [Producer-Bibliotheken](#) und die ["Stream Parser"-Bibliothek](#) senden und empfangen Videodaten in einem Format, das das Einbetten von Informationen neben den eigentlichen Videodaten unterstützt. Dieses Format basiert auf der Matroska (MKV)-Spezifikation.

Das [MKV-Format](#) ist eine offene Spezifikation für Mediendaten. Alle Bibliotheken und Codebeispiele im Amazon Kinesis Video Streams Developer Guide senden oder empfangen Daten im MKV-Format.

The [Produzentenbibliotheken von Kinesis Video Streams](#) verwendet die `Frame` Typen `StreamDefinition` und, um MKV-Stream-Header, Frame-Header und Frame-Daten zu erzeugen.

Weitere Hinweise zur vollständigen MKV-Spezifikation finden Sie unter [Matroska-Spezifikationen](#).

Die folgenden Abschnitte beschreiben die Komponenten von vom [C++ Producer-Bibliothek](#) erzeugten Daten im MKV-Format.

Themen

- [Stream-Header-Elemente](#)
- [Streamen Sie Track-Daten](#)
- [Frame-Header-Elemente](#)
- [MKV-Frame-Daten](#)

Stream-Header-Elemente

Die folgenden MKV-Header-Elemente werden von `StreamDefinition` (definiert in `StreamDefinition.h`) verwendet.

Element	Beschreibung	Typische Werte
stream_name	Entspricht dem Namen des Kinesis-Videostreams.	my-stream
retention_period	Die Dauer (in Stunden), für die Stream-Daten von Kinesis Video Streams gespeichert werden. Geben Sie dies 0 für einen Stream an, der keine Daten speichert.	24
Tags	Als Schlüssel-Wert-Paare angegebene Benutzerdaten. Diese Daten werden in der AWS Management Console angezeigt und können von Clientanwendungen gelesen werden, um eine Filterung durchzuführen oder Informationen zu einem Stream abzurufen.	
kms_key_id	Falls vorhanden, wird der benutzerdefinierte AWS KMS Schlüssel verwendet, um Daten im Stream zu verschlüsseln. Falls nicht, werden die Daten mit dem von Kinesis bereitgestellten Schlüssel () verschlüsselt. aws/kinesis-video	01234567-89ab-cdef-0123-456789ab
streaming_type	Derzeit ist der einzige gültige Streaming-Typ STREAMING_TYPE_REALTIME .	STREAMING_TYPE_REALTIME

Element	Beschreibung	Typische Werte
content_type	Der benutzerdefinierte Inhaltstyp. Damit gestreamte Videodaten in der Konsole abgespielt werden können, muss der Typ <code>video/h264</code> lauten.	video/h264
max_latency	Dieser Wert wird derzeit nicht verwendet und sollte auf 0 gesetzt werden.	0
fragment_duration	Dies ist eine Schätzung der Fragmentdauer und wird zur Optimierung verwendet. Die tatsächliche Fragmentdauer wird durch die Streaming-Daten bestimmt.	2
timecode_scale	<p>Gibt die Skala an, die von Frame-Timestamps verwendet wird. Der Standardwert beträgt 1 Millisekunde.</p> <p>Wenn Sie 0 festlegen, wird ebenfalls der Standardwert von 1 Millisekunde festgelegt. Dieser Wert kann zwischen 100 Nanosekunden und einer Sekunde liegen.</p> <p>Weitere Informationen finden Sie TimecodeScale in der Matroska-Dokumentation.</p>	

Element	Beschreibung	Typische Werte
<code>key_frame_fragmentation</code>	Bei der Einstellung <code>true</code> startet der Stream einen neuen Cluster, wenn ein Keyframe empfangen wird.	<code>true</code>
<code>frame_timecodes</code>	Falls <code>true</code> , verwendet Kinesis Video Streams die Werte für den Präsentationszeitstempel (<code>pts</code>) und den Dekodierungszeitstempel (<code>dts</code>) der empfangenen Frames. Falls <code>false</code> , stempelt Kinesis Video Streams die Frames beim Empfang mit vom System generierten Zeitwerten.	<code>true</code>
<code>absolute_fragment_time</code>	Mit dem Wert <code>true</code> werden die Cluster-Timecodes als absolute Zeiten (z. B. anhand der Systemuhr des Produzenten) interpretiert. Mit dem Wert <code>false</code> werden die Cluster-Timecodes als relativ zur Startzeit des Streams interpretiert.	<code>true</code>

Element	Beschreibung	Typische Werte
<code>fragment_acks</code>	Fallst <code>true</code> , werden Bestätigungen (ACKs) gesendet, wenn Kinesis Video Streams die Daten empfängt. Die ACKs können mit der Callback-Funktion <code>KinesisVideoStreamFragmentAck</code> oder <code>KinesisVideoStreamParseFragmentAck</code> angenommen werden.	<code>true</code>
<code>restart_on_error</code>	Gibt an, ob der Stream die Übertragung nach einem Fehler fortsetzen soll.	<code>true</code>
<code>nal_adaptation_flags</code>	Gibt an, ob der Inhalt NAL (Network Abstraction Layer)-Adaptions- oder Codec-interne Daten enthält. Gültige Flags sind <code>NAL_ADAPTATION_ANNEXB_NALS</code> und <code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code> .	<code>NAL_ADAPTATION_ANNEXB_NALS</code>
<code>frame_rate</code>	Schätzwert der Frame-Rate des Inhalts. Dieser Wert wird zur Optimierung verwendet. Die tatsächliche Frame-Rate wird durch die Rate der eingehenden Daten bestimmt. Durch Festlegen von <code>0</code> wird der Standardwert <code>24</code> zugewiesen.	<code>24</code>

Element	Beschreibung	Typische Werte
avg_bandwidth_bps	Eine Schätzung der Inhaltsbandbreite in Mbit/s. Dieser Wert wird zur Optimierung verwendet. Der tatsächliche Wert wird durch die Bandbreite der eingehenden Daten bestimmt. Die ungefähre Bandbreite für einen Videostream mit einer Auflösung von 720p bei 25 FPS beträgt beispielsweise etwa 5 Mbit/s.	5
buffer_duration	Die Dauer der Pufferung des Inhalts auf Konsumentenseite. Bei geringer Netzwerklatenz kann dieser Wert reduziert werden. Wenn die Netzwerklatenz hoch ist, wird durch eine Erhöhung dieses Werts verhindert, dass Frames verworfen werden, bevor sie gesendet werden können, da die Zuordnung die Frames nicht in den kleineren Puffer legt.	

Element	Beschreibung	Typische Werte
replay_duration	Die Zeitspanne, für die der Videodatenstrom „zurückgespult“ wird, wenn die Verbindung unterbrochen wird. Dieser Wert kann Null sein, wenn Bildverluste aufgrund eines Verbindungsverlusts kein Problem darstellen. Der Wert kann erhöht werden, wenn die verbrauchende Anwendung redundante Frames entfernen kann. Dieser Wert sollte kleiner als die Pufferdauer sein, andernfalls wird die Pufferdauer verwendet.	
connection_staleness	Die Zeitspanne, die eine Verbindung aufrechterhalten wird, wenn keine Daten eingehen.	
codec_id	Der von dem Inhalt verwendete Codec. Weitere Informationen finden Sie unter CodeclD in der Matroska-Spezifikation.	V_MPEG2
track_name	Der benutzerdefinierte Name des Tracks	my_track

Element	Beschreibung	Typische Werte
codecPrivateData	Vom Encoder zur Decodierung der Frame-Daten bereitgestellte Informationen wie Frame-Breite und -Höhe in Pixeln, die von vielen nachgelagerten Konsumenten benötigt werden. In der C++ Producer-Bibliothek enthält das Array <code>gMkvTrackVideoBits</code> in der Datei <code>MkvStatics.cpp</code> Elemente für Frame-Breite und -Höhe in Pixeln.	
codecPrivateDataGröße	Die Größe der Daten im Parameter <code>codecPrivateData</code>	
track_type	Der Typ des Tracks für den Stream.	MKV_TRACK_INFO_TYPE_AUDIO oder MKV_TRACK_INFO_TYPE_VIDEO
segment_uuid	Benutzerdefinierte Segment-UUID (16 Bytes).	
default_track_id	Eindeutige Zahl ungleich Null für den Track.	1

Streamen Sie Track-Daten

Die folgenden MKV-Track-Elemente werden von `StreamDefinition` (definiert in `StreamDefinition.h`) verwendet.

Element	Beschreibung	Typische Werte
track_name	Benutzerdefinierter Track-Name. Beispiel: "Audio" für die Audiospur.	audio
codec_id	Codec-ID für den Track. Beispiel: "A_AAC" für eine Audiospur.	A_AAC
cpd	Die vom Encoder bereitgestellten Daten, die zum Decodieren der Frame-Daten verwendet werden. Diese Daten können Frame-Breite und -Höhe in Pixel umfassen, die von vielen nachgelagerten Konsumenten benötigt werden. In der C++ Producer Library enthält das gMkvTrack VideoBits Array in MkvStatics.cpp die Pixelbreite und -höhe für den Frame.	
cpd_size	Die Größe der Daten im codecPrivateData Parameter.	
track_type	Der Typ des Tracks. Beispielsweise können Sie den Aufzählungswert MKV_TRACK_INFO_TYPE_AUDIO für die Audioverarbeitung auswählen.	MKV_TRACK_INFO_TYPE_AUDIO

Frame-Header-Elemente

Die folgenden MKV-Header-Elemente werden von `Frame` (definiert im `KinesisVideoPic`-Paket in der Datei `mkvgen/Include.h`) verwendet:

- **Frame Index:** Ein monoton ansteigender Wert.
- **Flags:** Der Typ des Frames. Gültige Werte sind unter anderem:
 - `FRAME_FLAGS_NONE`
 - `FRAME_FLAG_KEY_FRAME`: Wenn `key_frame_fragmentation` für den Stream aktiviert ist, wird bei jedem Keyframe ein neues Fragment begonnen.
 - `FRAME_FLAG_DISCARDABLE_FRAME`: Gibt an, dass der Decoder diesen Frame verwerfen kann, wenn die Decodierung langsam erfolgt.
 - `FRAME_FLAG_INVISIBLE_FRAME`: Die Dauer dieses Blocks beträgt 0.
- **Zeitstempel für die Dekodierung:** Der Zeitstempel, zu dem dieser Frame dekodiert wurde. Wenn frühere Frames für die Dekodierung von diesem Frame abhängen, liegt dieser Zeitstempel möglicherweise vor dem früherer Frames. Dieser Wert gilt relativ zum Anfang des Fragments.
- **Zeitstempel der Präsentation:** Der Zeitstempel, zu dem dieser Frame angezeigt wird. Dieser Wert gilt relativ zum Anfang des Fragments.
- **Duration:** Die Wiedergabedauer des Frames.
- **Size:** Die Größe der Framedaten in Byte.

MKV-Frame-Daten

Die Daten in `frame.frameData` können je nach verwendetem Codierungsschema entweder nur die Mediendaten des Frames enthalten oder zusätzlich auch eingebettete Header-Informationen. Um in der [AWS Management Console](#) angezeigt zu werden, müssen die Daten im [H.264-Codec](#) codiert sein. Kinesis Video Streams kann jedoch zeitserialisierte Datenströme in jedem Format empfangen.

Erste Schritte mit Amazon Kinesis Video Streams

In diesem Abschnitt wird beschrieben, wie Sie die folgenden Aufgaben in Amazon Kinesis Video Streams ausführen:

- Richten Sie Ihren Administrator ein AWS-Konto und erstellen Sie einen, falls Sie dies noch nicht getan haben.
- Kinesis-Video-Stream erstellen
- Senden Sie Daten von Ihrer Kamera an den Kinesis-Videostream und sehen Sie sich die Medien in der Konsole an.

Wenn Sie Amazon Kinesis Video Streams noch nicht kennen, empfehlen wir Ihnen, [Kinesis Video Streams: So funktioniert's](#) zuerst zu lesen.

Note

Wenn Sie dem Beispiel „Erste Schritte“ folgen, fallen für Sie keine Gebühren an. AWS-Konto Informationen zu den Datenkosten in Ihrer Region finden Sie unter [Amazon Kinesis Video Streams Streams-Preise](#).

Themen

- [Einrichten eines -Kontos](#)
- [Einen Kinesis-Videostream erstellen](#)
- [Daten an einen Amazon Kinesis Kinesis-Videostream senden](#)
- [Mediendaten konsumieren](#)

Einrichten eines -Kontos

Bevor Sie Amazon Kinesis Video Streams zum ersten Mal verwenden, führen Sie die folgenden Aufgaben aus.

Themen

- [Melden Sie sich an für ein AWS-Konto](#)
- [Erstellen eines Administratorbenutzers](#)

- [Erstellen Sie einen Schlüssel AWS-Konto](#)

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine 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/signup>.
2. Folgen Sie den Online-Anweisungen.

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

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. 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. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen eines Administratorbenutzers

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

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

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

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Administratorbenutzers

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter [Benutzerzugriff mit der Standardeinstellung konfigurieren](#).AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-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 [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal](#).

Erstellen Sie einen Schlüssel AWS-Konto

Sie benötigen einen AWS-Konto Schlüssel, um programmgesteuert auf Amazon Kinesis Video Streams zuzugreifen.

Gehen Sie wie folgt vor, um einen AWS-Konto Schlüssel zu erstellen:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie in der Navigationsleiste Benutzer und dann den Administratorbenutzer aus.
3. Wählen Sie die Registerkarte Security credentials (Sicherheitsanmeldeinformationen) und anschließend Create access key (Zugriffsschlüssel erstellen) aus.

4. Notieren Sie sich die Zugriffsschlüssel-ID. Wählen Sie unter Geheimer Zugriffsschlüssel die Option Anzeigen aus. Notieren Sie sich den unter geheimer Zugriffsschlüssel angezeigten geheimen Zugriffsschlüssel.

Einen Kinesis-Videostream erstellen

In diesem Abschnitt wird beschrieben, wie Sie einen Kinesis-Videostream erstellen.

In diesem Abschnitt werden folgende Verfahren beschrieben:

- [the section called “Erstellen Sie mit der Konsole einen Videostream”](#)
- [the section called “Erstellen Sie einen Videostream mit dem AWS CLI”](#)

Erstellen Sie mit der Konsole einen Videostream

1. Öffnen Sie die Konsole unter <https://console.aws.amazon.com/kinesisvideo/home>.
2. Wählen Sie auf der Seite Video streams (Videostreams) die Option Create video stream (Videostream erstellen) aus.
3. Geben Sie auf der Seite Neuen Videostream erstellen ***YourStreamName*** den Namen des Streams ein. Lassen Sie die Schaltfläche Standardkonfiguration ausgewählt.
4. Wählen Sie Create video stream (Videostream erstellen).
5. Nachdem Amazon Kinesis Video Streams den Stream erstellt hat, überprüfen Sie die Details auf der YourStreamNameSeite.

Erstellen Sie einen Videostream mit dem AWS CLI

1. Stellen Sie sicher, dass Sie das AWS CLI installiert und konfiguriert haben. Weitere Informationen finden Sie in der Dokumentation zu [AWS Command Line Interface](#).
2. Führen Sie in der AWS CLI den folgenden Create-Stream-Befehl aus:

```
aws kinesisvideo create-stream --stream-name "YourStreamName" --data-retention-in-hours 24
```

Die Befehlsausgabe sieht etwa folgendermaßen aus:

```
{
  "StreamARN": "arn:aws:kinesisvideo:us-
west-2:123456789012:stream/YourStreamName/123456789012"
}
```

Daten an einen Amazon Kinesis Kinesis-Videostream senden

In diesem Abschnitt wird beschrieben, wie Sie Mediendaten von einer Kamera an den Kinesis-Videostream senden, den Sie im vorherigen Abschnitt erstellt haben. In diesem Abschnitt wird die [C++ Producer-Bibliothek](#) als [GStreamer](#) Plug-in verwendet.

Um Medien von einer Vielzahl von Geräten auf verschiedenen Betriebssystemen zu senden, verwendet dieses Tutorial die Kinesis Video Streams C++ Producer Library und [GStreamer](#), ein Open-Source-Medienframework, das den Zugriff auf Kameras und andere Medienquellen standardisiert.

Themen

- [Erstellen Sie das SDK und die Beispiele](#)
- [Führen Sie die Beispiele aus, um Medien auf Kinesis Video Streams hochzuladen](#)
- [Bestätigungsobjekte überprüfen](#)

Erstellen Sie das SDK und die Beispiele

Sie können das SDK und die Beispiele auf Ihrem Computer oder in erstellen AWS Cloud9. Folgen Sie den unten angegebenen Verfahren.

Build on your computer

Verwenden Sie die Anweisungen in der [Readme-Datei](#), um die Producer-Bibliothek und die Beispielanwendung zu erstellen.

Dies umfasst:

- Installieren von Abhängigkeiten
- Das Repository klonen
- Verwenden von CMake zum Generieren von Makefiles

- Die Binärdateien mit Make erstellen

Build in AWS Cloud9

Gehen Sie wie folgt vor, um in AWS Cloud9 Kinesis Video Streams hochzuladen. Sie müssen nichts auf Ihren Computer herunterladen.

1. In der AWS Management Console, öffnen [AWS Cloud9](#).

Wählen Sie Umgebung erstellen aus.

2. Gehen Sie auf dem Bildschirm Umgebung erstellen wie folgt vor:

- Name — Geben Sie einen Namen für Ihre neue Umgebung ein.
- Plattform — Wählen Sie Ubuntu Server 22.04 LTS.

Sie können die Standardauswahl für die anderen Felder beibehalten.

3. Wenn die Umgebung erstellt wurde, wählen Sie in der Spalte Cloud9 IDE die Option Öffnen aus.

Im unteren mittleren Bereich des Bildschirms sehen Sie. `Admin:~/environment $` Dies ist das AWS Cloud9 (Amazon EC2) -Terminal.

Note

Wenn Sie das Terminal versehentlich schließen, wählen Sie Fenster, Neues Terminal.

Führen Sie die folgenden Befehle im Terminal aus, um die Lautstärke auf 20 GiB zu ändern.

- a. Laden Sie das -Skript herunter.

```
wget https://awsj-iot-handson.s3-ap-northeast-1.amazonaws.com/kvs-workshop/resize_volume.sh
```

- b. Erteilen Sie dem Skript Ausführungsberechtigungen.

```
chmod +x resize_volume.sh
```

- c. Führen Sie das Skript aus.

```
./resize_volume.sh
```

4. Rufen Sie die neuesten Informationen über die gesamte Software ab, die Sie mit dem Advanced Packaging Tool (APT) installieren oder aktualisieren können.

Dieser Befehl aktualisiert nicht die Software selbst, sondern stellt sicher, dass Ihr System über die neuesten verfügbaren Versionen informiert ist.

```
sudo apt-get update
```

5. Installieren Sie die Abhängigkeiten des C++-Producer-SDK.

```
sudo apt-get install -y cmake m4 git build-essential pkg-config libssl-dev  
libcurl4-openssl-dev \  
liblog4cplus-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \  
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad gstreamer1.0-plugins-  
good \  
gstreamer1.0-plugins-ugly gstreamer1.0-tools
```

6. Verwenden Sie Git, um das C++-Producer-SDK zu klonen.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-  
cpp.git
```

7. Bereiten Sie ein Build-Verzeichnis vor.

```
cd amazon-kinesis-video-streams-producer-sdk-cpp  
mkdir build  
cd build
```

8. Verwenden Sie CMake, um Makefiles zu generieren.

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=TRUE -DBUILD_DEPENDENCIES=OFF
```

Das Ende der erwarteten Ausgabe sieht wie folgt aus:

```
-- Build files have been written to: /home/ubuntu/environment/amazon-kinesis-  
video-streams-producer-sdk-cpp/build
```

9. Verwenden Sie `make`, um das SDK und die Beispielanwendungen zu kompilieren und die endgültigen ausführbaren Dateien zu erstellen.

```
make
```

Das Ende der erwarteten Ausgabe sieht wie folgt aus:

```
[100%] Linking CXX executable kvs_gstreamer_file_uploader_sample  
[100%] Built target kvs_gstreamer_file_uploader_sample
```

10. Vergewissern Sie sich, dass die Beispieldateien erstellt wurden. Listet die Dateien im aktuellen Verzeichnis auf:

```
ls
```

Vergewissern Sie sich, dass die folgenden Dateien vorhanden sind:

- `kvs_gstreamer_sample`
- `libgstkvssink.so`

Führen Sie die Beispiele aus, um Medien auf Kinesis Video Streams hochzuladen

Die Beispielanwendung unterstützt keine IMDS-Anmeldeinformationen. Exportieren Sie in Ihrem Terminal die AWS Anmeldeinformationen für Ihren IAM-Benutzer oder Ihre IAM-Rolle und die Region, in der sich Ihr Stream befindet.

```
export AWS_ACCESS_KEY_ID=YourAccessKey  
export AWS_SECRET_KEY=YourSecretKey  
export AWS_DEFAULT_REGION=us-west-2
```

Wenn du temporäre AWS Anmeldeinformationen verwendest, exportiere auch dein Sitzungstoken:

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Laden Sie ein MP4-Beispielvideo herunter, um es auf Kinesis Video Streams hochzuladen.

```
wget https://awsj-iot-handson.s3-ap-northeast-1.amazonaws.com/kvs-workshop/sample.mp4
```

Führen Sie den folgenden Befehl mit dem Namen des Streams aus, den Sie zuvor erstellt haben. Wenn Sie noch keinen Stream erstellt haben, finden Sie weitere Informationen unter [the section called "Einen Kinesis-Videostream erstellen"](#).

```
./kvs_gstreamer_sample YourStreamName ./sample.mp4
```

Sample video from GStreamer

Verwenden Sie den folgenden Befehl, um mit GStreamer ein Video zu generieren.

Teilen Sie GStreamer mit, wo sich das GStreamer-Plugin befinden soll. kvssink Geben Sie in Ihrem Build-Verzeichnis den Pfad zu dem Ordner an, der die Datei enthält. `libgstkvssink.so`

Führen Sie in Ihrem Build-Verzeichnis den folgenden Befehl aus:

```
export GST_PLUGIN_PATH=`pwd`
```

Diese GStreamer-Pipeline generiert einen Live-Testvideostream mit einem Standard-Testmuster, das mit 10 Bildern pro Sekunde und einer Auflösung von 640x480 Pixeln läuft. Es wird ein Overlay hinzugefügt, das die aktuelle Systemzeit und das aktuelle Systemdatum anzeigt. Das Video wird dann in das H.264-Format codiert, und es werden höchstens alle 10 Bilder Keyframes generiert, was zu einer Fragmentdauer (auch bekannt als Größe einer Bildgruppe (GoP)) von 1 Sekunde führt. kvssinknimmt den H.264-codierten Videostream, packt ihn in das Matroska (MKV) - Containerformat und lädt ihn in Ihren Kinesis-Videostream hoch.

Führen Sie den folgenden Befehl aus:

```
gst-launch-1.0 -v videotestsrc is-live=true \  
! video/x-raw,framerate=10/1,width=640,height=480 \  
! clockoverlay time-format="%a %B %d, %Y %I:%M:%S %p" \  
! x264enc bframes=0 key-int-max=10 \  
! h264parse \  
! kvssink stream-name="YourStreamName"
```

Um die GStreamer-Pipeline zu stoppen, wählen Sie das Terminalfenster aus und drücken Sie STRG+C.

Note

Weitere Informationen zur Verwendung des GStreamer-Plug-ins zum Streamen von Videos von einem RTSP-Stream von einer Kamera oder von einer USB-Kamera finden Sie unter.

[Beispiel: Kinesis Video Streams Producer SDK GStreamer-Plugin](#)

Bestätigungsobjekte überprüfen

Während des Uploads sendet Kinesis Video Streams Bestätigungsobjekte zurück an den Client, der den Upload durchführt. Sie sollten diese in der Befehlsausgabe gedruckt sehen. Ein Beispiel sieht wie folgt aus:

```
{"EventType": "PERSISTED", "FragmentTimecode": 1711124585823, "FragmentNumber": "1234567890123456789"}
```

Wenn die Bestätigung „Ja“ Event Type lautet PERSISTED, bedeutet dies, dass Kinesis Video Streams diesen Medienblock dauerhaft gespeichert und verschlüsselt hat, um ihn abzurufen, zu analysieren und langfristig zu speichern.

Weitere Informationen zu Bestätigungen finden Sie unter [the section called “PutMedia”](#)

Mediendaten konsumieren

Sie können Mediendaten konsumieren, indem Sie sie entweder in der Konsole anzeigen oder indem Sie eine Anwendung erstellen, die Mediendaten mithilfe von Hypertext Live Streaming (HLS) aus einem Stream liest.

Medien in der Konsole anzeigen

Öffnen Sie in einem anderen Browser-Tab den AWS Management Console. Wählen Sie im Kinesis Video Streams Streams-Dashboard die Option [Videostreams](#) aus.

Wählen Sie den Namen Ihres Streams in der Liste der Streams aus. Verwenden Sie bei Bedarf die Suchleiste.

Erweitern Sie den Bereich Medienwiedergabe. Wenn das Video noch hochgeladen wird, wird es angezeigt. Wenn der Upload abgeschlossen ist, wählen Sie den Doppelpfeil nach links.

Konsumieren Sie Mediendaten mit HLS

Sie können mithilfe von HLS eine Client-Anwendung erstellen, die Daten aus einem Kinesis-Videostream verwendet. Weitere Informationen zum Erstellen einer Anwendung, die Mediendaten mithilfe von HLS verarbeitet, finden Sie unter [the section called “Video Playback \(Videowiedergabe\)”](#).

Edge-Agent von Amazon Kinesis Video Streams

Amazon Kinesis Video Streams bietet eine effiziente, kostengünstige Möglichkeit, eine Verbindung zu IP-Kameras vor Ort für Kunden herzustellen. Mit dem Edge Agent von Amazon Kinesis Video Streams können Sie Videos von den Kameras lokal aufzeichnen und speichern und Videos nach einem vom Kunden definierten Zeitplan für die langfristige Speicherung, Wiedergabe und analytische Verarbeitung in die Cloud streamen.

Note

Füllen Sie dieses [Kurzformular](#) aus, um auf den Edge-Agenten von Amazon Kinesis Video Streams zuzugreifen.

Sie können den Amazon Kinesis Video Streams Edge Agent herunterladen und auf Ihren On-Premises-Edge-Rechengeräten bereitstellen. Sie können sie auch einfach in Docker-Containern bereitstellen, die auf Amazon EC2 ausgeführt werden. Nach der Bereitstellung können Sie die Amazon Kinesis Video Streams API verwenden, um Konfigurationen für Videoaufzeichnungen und Cloud-Uploads zu aktualisieren. Die Funktion funktioniert mit jeder IP-Kamera, die über das RTSP-Protokoll streamen kann. Es ist keine zusätzliche Firmware-Bereitstellung für die Kameras erforderlich.

Wir bieten die folgenden Installationen für den Amazon Kinesis Video Streams Edge Agent an:

- Als AWS IoT Greengrass V2 Komponente: Sie können den Edge-Agenten von Amazon Kinesis Video Streams als AWS IoT Greengrass Komponente auf jedem AWS IoT Greengrass zertifizierten Gerät installieren. Weitere Informationen zu AWS IoT Greengrass finden Sie im [AWS IoT Greengrass Version 2-Benutzerhandbuch](#).
- Auf AWS Snowball Edge: Sie können den Edge-Agenten von Amazon Kinesis Video Streams auf Snowball-Edge-Geräten ausführen. Weitere Informationen finden Sie im [AWS Snowball -Edge-Entwicklerhandbuch](#).
- Bei einer nativen AWS IoT Bereitstellung: Sie können den Edge-Agenten von Amazon Kinesis Video Streams nativ auf jeder Rechen-Instance installieren. Edge SDK verwendet [AWS IoT Core](#) für die Verwaltung von Edges über die [the section called "Amazon Kinesis Video Streams"](#).

Um mit Amazon Kinesis Video Streams Edge Agent zu beginnen, fahren Sie mit den entsprechenden Verfahren unten fort.

Themen

- [API-Operationen von Amazon Kinesis Video Streams Edge Agent](#)
- [Überwachen von Amazon Kinesis Video Streams Edge Agent](#)
- [Ausführen von Amazon Kinesis Video Streams Edge Agent im Nicht-AWS IoT Greengrass-Modus](#)
- [Bereitstellen des Edge-Agenten von Amazon Kinesis Video Streams in AWS IoT Greengrass](#)
- [Häufig gestellte Fragen zum Edge-Agenten von Amazon Kinesis Video Streams](#)

API-Operationen von Amazon Kinesis Video Streams Edge Agent

Verwenden Sie die folgenden API-Operationen, um den Edge-Agenten von Amazon Kinesis Video Streams zu konfigurieren:

- [the section called “StartEdgeConfigurationUpdate”](#)
- [the section called “DescribeEdgeConfiguration”](#)
- [the section called “DeleteEdgeConfiguration”](#)
- [the section called “ListEdgeAgentConfigurations”](#)

Überwachen von Amazon Kinesis Video Streams Edge Agent

Informationen zur Überwachung Ihres Amazon Kinesis Video Streams Edge Agent finden Sie unter [the section called “Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch”](#).

Ausführen von Amazon Kinesis Video Streams Edge Agent im Nicht-AWS IoT Greengrass-Modus

Gehen Sie wie folgt vor, um den Edge-Agenten von Amazon Kinesis Video Streams mit AWS IoT MQTT als eigenständige Bereitstellung auszuführen.

Themen

- [Schritt 1: Installieren der erforderlichen Abhängigkeiten auf dem Gerät](#)

- [Schritt 2: Erstellen der Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera-RTSP-URLs](#)
- [Schritt 3: Erstellen einer IAM-Berechtigungsrichtlinie](#)
- [Schritt 4: Erstellen einer IAM-Rolle](#)
- [Schritt 5: Erstellen des AWS IoT Rollenalias](#)
- [Schritt 6: Erstellen der AWS IoT Richtlinie](#)
- [Schritt 7: Erstellen eines -AWS IoT Objekts und Abrufen von Anmeldeinformationen für AWS IoT Core](#)
- [Schritt 8: Erstellen und Ausführen des Edge-Agenten von Amazon Kinesis Video Streams](#)
- [Schritt 9: \(Optional\) Installieren des CloudWatch Agenten auf dem Gerät](#)
- [Schritt 10: \(Optional\) Führen Sie den Edge-Agenten von Amazon Kinesis Video Streams als nativen Prozess aus](#)

Schritt 1: Installieren der erforderlichen Abhängigkeiten auf dem Gerät

Note

Eine Liste der unterstützten Betriebssysteme finden Sie unter [the section called “Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent?”](#).

Installieren von Abhängigkeiten auf dem Gerät

1. Um den Edge-Agenten von Amazon Kinesis Video Streams auszuführen, installieren Sie die folgenden entsprechenden Bibliotheken auf Ihrem Gerät:

Ubuntu

Type:

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -  
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'  
sudo apt-get update  
  
sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \  
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \  
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \  

```

```
gstreamer1.0-plugins-good gstreamer1.0-tools \  
unzip java-11-amazon-corretto-jdk maven
```

Amazon Linux 2

Type:

```
sudo yum update -y && sudo yum upgrade -y && sudo yum clean all -y  
sudo yum install -y gcc-c++ openssl-devel libcurl-devel gstreamer1* wget \  
java-11-amazon-corretto tar
```

Installieren Sie `log4cplus-2.1.0` aus der Quelle.

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/  
log4cplus-2.1.0.tar.gz  
tar -xzvf log4cplus-2.1.0.tar.gz  
cd log4cplus-2.1.0 && \  
mkdir build && \  
cd build && \  
cmake .. && \  
sudo make && \  
sudo make install
```

Installieren Sie `apache-maven-3.9.2` aus der Quelle.

```
wget https://dlcdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-  
bin.tar.gz  
RUN tar -xzvf apache-maven-3.9.2-bin.tar.gz -C /opt
```

Important

Wenn ein Bildschirm angezeigt wird, in dem Sie darüber informiert werden, dass einige Services neu gestartet werden müssen, drücken Sie die Eingabetaste, um OK auszuwählen.

Weitere Informationen finden Sie im [Amazon Corretto 11-Benutzerhandbuch](#).

2. Installieren Sie die AWS Command Line Interface. Weitere Informationen finden Sie unter [Installieren oder Aktualisieren der neuesten Version der AWS CLI](#) Verfahren im AWS Command Line Interface -Benutzerhandbuch.

Schritt 2: Erstellen der Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera-RTSP-URLs

Gehen Sie wie folgt vor, um die in benötigten Streams und Geheimnisse zu erstellenAWS Secrets Manager. Führen Sie diesen Schritt zuerst aus, da Sie die ARNs der erstelltenRessourcen in den Richtlinien benötigen.

Erstellen von Amazon Kinesis Video Streams

Erstellen Sie Amazon Kinesis Video Streams mithilfe der AWS CLI, oder AWS Management ConsoleAPI.

AWS Management ConsoleÖffnen Sie in der die [Amazon Kinesis Video Streams-Konsole](#). Wählen Sie im linken Navigationsbereich Videostreams aus.

Weitere Informationen finden Sie unter [the section called "Einen Kinesis-Videostream erstellen"](#).

Erstellen von Geheimnissen in AWS Secrets Manager

AWS Management ConsoleÖffnen Sie in der die [AWS Secrets ManagerKonsole](#). Wählen Sie in der linken Navigation Secrets aus.

Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.

1. Wählen Sie Store a new secret (Ein neues Secret speichern).
 - a. Schritt 1: Auswählen des Secret-Typs
 - Wählen Sie Anderer Geheimnistyp aus.
 - Fügen Sie im Abschnitt Schlüssel/Wert-Paare ein Schlüssel-Wert-Paar hinzu.

Schlüssel: MediaURI

Note

Der Schlüssel muss seinMediaURI. Dabei wird zwischen Groß- und Kleinschreibung unterschieden. Wenn Sie es falsch eingeben, funktioniert die Anwendung nicht.

Wert : *Your MediaURI*.

Example

Beispiel: `rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI`.

- b. Schritt 2: Konfigurieren von Secret . Geben Sie diesem Secret einen Namen. Benennen Sie sie nach Belieben.
 - c. Schritt 3: Rotation konfigurieren – optional . Wählen Sie Weiter aus.
 - d. Schritt 4: Überprüfen von . Wählen Sie Store (Speichern) aus.
2. Wenn Ihr Secret nicht sofort angezeigt wird, wählen Sie die Schaltfläche Aktualisieren aus.

Wählen Sie den Namen Ihres Secrets aus. Notieren Sie sich den geheimen ARN .

3. Wiederholen Sie diesen Vorgang für jeden MediaURI, von dem Sie streamen möchten.

Note

Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der Amazon EC2-Instance aus auf diese zugreifen oder wenn Sie nicht verwaltet werden, während Sie mit dem VPN verbunden sind.

Important

Ihre Kamera-RTSP-URL sollte Videos im h.264-Format streamen. Die Fragmentdauer darf das in angegebene Limit nicht überschreiten [the section called “Beschränkungen des Producer-SDK”](#).

Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.

Führen Sie aus, `gst-discoverer-1.0 Your RtspUrl` um sicherzustellen, dass Ihre Kamera von Ihrem Gerät aus erreichbar ist.

Speichern Sie die ARNs für alle Streams und Secrets, die Sie erstellt haben. Sie benötigen diese für den nächsten Schritt.

Schritt 3: Erstellen einer IAM-Berechtigungsrichtlinie

Gehen Sie wie folgt vor, um eine IAM-Richtlinie zu erstellen. Diese Berechtigungsrichtlinie ermöglicht eine selektive Zugriffskontrolle (eine Teilmenge der unterstützten Operationen) für eine - AWS-Ressource. In diesem Fall sind die AWS Ressourcen die Videostreams, zu denen der Amazon Kinesis Video Streams Edge Agent streamen soll. Die Ressourcen enthalten auch die AWS Secrets Manager Secrets, die der Amazon Kinesis Video Streams Edge Agent abrufen kann. Weitere Informationen finden Sie unter [IAM-Richtlinien](#).

Erstellen einer Richtlinie mithilfe des JSON-Richtlinienditors

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.

Wenn Sie zum ersten Mal Policies (Richtlinien) auswählen, erscheint die Seite Welcome to Managed Policies (Willkommen bei verwalteten Richtlinien). Wählen Sie Get Started.

3. Wählen Sie oben auf der Seite Create policy (Richtlinie erstellen) aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie folgendes JSON-Richtliniendokument ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData",
        "kinesisvideo:ListStreams",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
```

```

    "Action": [
      "kinesisvideo:DescribeStream",
      "kinesisvideo:PutMedia",
      "kinesisvideo:TagStream",
      "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": [
      "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
      "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": [
      "arn:aws:secretsmanager:*:*:secret:*",
      "arn:aws:secretsmanager:*:*:secret:*"
    ]
  }
]
}

```

Note

Ersetzen Sie `arn:aws:kinesisvideo:*:*:stream/streamName1/*` und `arn:aws:kinesisvideo:*:*:stream/streamName2/*` durch die ARNs für die Videostreams und ersetzen Sie durch `arn:aws:secretsmanager:*:*:secret:*` die ARNs, die die MediaURI-Secrets enthalten, die Sie in [erstellt haben](#) [the section called "2. Erstellen von Ressourcen für Ihre IP-Kamera-RTSP-URLs"](#). Verwenden Sie die ARNs für die Secrets, auf die der Amazon Kinesis Video Streams Edge Agent zugreifen soll.

6. Wählen Sie Weiter aus.

Note

Sie können jederzeit zwischen den Editoroptionen Visual und JSON wechseln. Wenn Sie jedoch Änderungen vornehmen oder im Visual-Editor Weiter wählen, strukturiert IAM Ihre Richtlinie möglicherweise um, um sie für den visuellen Editor zu optimieren. Weitere Informationen finden Sie unter [Richtlinienrestrukturierung](#) im IAM-Benutzerhandbuch.

7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen und eine optionale Beschreibung für die Richtlinie ein, die Sie erstellen. Überprüfen Sie Permissions defined in this policy (In dieser Richtlinie definierte Berechtigungen), um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden.
8. Wählen Sie Create policy (Richtlinie erstellen) aus, um Ihre neue Richtlinie zu speichern.

Schritt 4: Erstellen einer IAM-Rolle

Die Rolle, die Sie in diesem Schritt erstellen, kann von übernommen werden, AWS IoT um temporäre Anmeldeinformationen von AWS Security Token Service () zu erhalten AWS STS. Dies geschieht, wenn Autorisierungsanforderungen für Anmeldeinformationen vom Amazon Kinesis Video Streams Edge Agent ausgeführt werden.

Erstellen der Servicerolle für Amazon Kinesis Video Streams (IAM-Konsole)

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie den Rollentyp Benutzerdefinierte Vertrauensrichtlinie und fügen Sie die folgende Richtlinie ein:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "credentials.iot.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
}
```

4. Aktivieren Sie das Kontrollkästchen neben der IAM-Richtlinie, die Sie in erstellt haben [the section called "3. Erstellen einer IAM-Berechtigungsrichtlinie"](#).
5. Wählen Sie Weiter aus.
6. Geben Sie einen Rollennamen oder ein Suffix für den Rollennamen ein, um den Zweck dieser Rolle zu identifizieren.

Example

Beispiel: KvsEdgeAgentRole

7. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.
8. (Optional) Fügen Sie der Rolle Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen.

Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter [Markieren von IAM-Ressourcen](#) im IAM-Benutzerhandbuch.

9. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Schritt 5: Erstellen des AWS IoT Rollenalias

Gehen Sie wie folgt vor, um einen -AWS IoT Rollenalias für die IAM-Rolle zu erstellen, die Sie in [the section called "4. Erstellen einer IAM-Rolle"](#) erstellt haben. Ein Rollenalias ist ein alternatives Datenmodell, das auf die IAM-Rolle verweist. Eine Anforderung an den AWS IoT Anmeldeinformationsanbieter muss einen Rollenalias enthalten, der angibt, welche IAM-Rolle angenommen werden soll, um temporäre Anmeldeinformationen von AWS Security Token Service (AWS STS) zu erhalten. Weitere Informationen finden Sie unter [Verwenden eines Zertifikats zum Abrufen eines Sicherheitstokens](#).

Erstellen des AWS IoT Rollenalias

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die -AWS IoT Core-Konsole unter <https://console.aws.amazon.com/iot/>.
2. Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.
3. Wählen Sie in der linken Navigation Sicherheit und dann Rollen-Aliasnamen aus.
4. Wählen Sie Rollenalias erstellen aus.
5. Geben Sie einen Namen für Ihren Rollenalias ein.

Example

Beispiel: KvsEdgeAgentRoleAlias

6. Wählen Sie in der Dropdownliste Rolle die IAM-Rolle aus, die Sie in [the section called "4. Erstellen einer IAM-Rolle"](#) erstellt haben.

7. Wählen Sie Erstellen. Auf der nächsten Seite sehen Sie eine Notiz, dass Ihr Rollenalias erfolgreich erstellt wurde.
8. Suchen Sie nach dem neu erstellten Rollenalias und wählen Sie ihn aus. Notieren Sie sich den Rollen-Alias-ARN . Sie benötigen dies für die AWS IoT Richtlinie im nächsten Schritt.

Schritt 6: Erstellen der AWS IoT Richtlinie

Gehen Sie wie folgt vor, um eine -AWS IoT Richtlinie zu erstellen, die an das Gerätezertifikat angehängt wird. Dies erteilt Berechtigungen für -AWS IoT Funktionen und ermöglicht die Annahme des Rollenalias mithilfe des Zertifikats.

Mit -AWS IoT Core Richtlinien können Sie den Zugriff auf die AWS IoT Core Datenebene steuern. Die AWS IoT Core Datenebene besteht aus Operationen, mit denen Sie Folgendes tun können:

- Herstellen einer Verbindung mit dem AWS IoT Core Message Broker
- Senden und Empfangen von MQTT-Nachrichten
- Geräteschatten eines Objekts abrufen oder aktualisieren

Weitere Informationen finden Sie unter [AWS IoT Core Richtlinien](#).

Verwenden des AWS IoT Richtlinieneditors zum Erstellen einer -AWS IoT Richtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>.
2. Wählen Sie in der linken Navigation Sicherheit und dann Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Geben Sie einen Namen für Ihre Richtlinie ein.

Example

Ein Beispiel für einen Richtliniennamen ist KvsEdgeAccessIoTPolicy .

5. (Optional) Fügen Sie der Richtlinie Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen.

Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter [Markieren Ihrer -AWS IoT Ressourcen](#) im AWS IoT Core -Entwicklerhandbuch.

6. Wählen Sie den Tab JSON.

7. Fügen Sie das folgende JSON-Richtliniendokument ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "your-role-alias-arn"
    }
  ]
}
```

Note

Ersetzen Sie durch `your-role-alias-arn` den ARN des Rollenalias, den Sie in [erstellt haben](#) [the section called "5. Erstellen des AWS IoT Rollenalias"](#).

8. Wählen Sie Erstellen, um Ihre Arbeit zu speichern.

Schritt 7: Erstellen eines -AWS IoT Objekts und Abrufen von Anmeldeinformationen für AWS IoT Core

An dieser Stelle haben Sie Folgendes erstellt:

- Eine IAM-Berechtigungsrichtlinie. Siehe [the section called “3. Erstellen einer IAM-Berechtigungsrichtlinie”](#).
- Eine IAM-Rolle, an die die Berechtigungsrichtlinie angehängt ist. Siehe [the section called “4. Erstellen einer IAM-Rolle”](#).
- Ein -AWS IoT Rollenalias für die IAM-Rolle. Siehe [the section called “5. Erstellen des AWS IoT Rollenalias”](#).
- Eine -AWS IoT Richtlinie, die derzeit keiner AWS Ressource zugeordnet ist. Siehe [the section called “6. Erstellen der AWS IoT-Richtlinie”](#).

So erstellen und registrieren Sie ein -AWS IoT Objekt und erhalten AWS IoT Core Anmeldeinformationen

1. Registrieren Sie das Gerät als -AWS IoT Objekt und generieren Sie das X.509-Zertifikat für das Gerät.
 - a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>.
 - b. Wählen Sie die entsprechende Region aus.
 - c. Wählen Sie in der linken Navigation Alle Geräte und dann Objekte aus.
 - d. Wählen Sie Objekte erstellen aus.
 - e. Wählen Sie Einzelobjekt erstellen und dann Weiter aus.
 1. Schritt 1. Objekteigenschaften angeben
Geben Sie einen Namen für Ihr Objekt ein und wählen Sie dann Weiter aus.
 2. Schritt 2. Konfigurieren des Gerätezertifikats
Wählen Sie Neues Zertifikat automatisch generieren (empfohlen) und dann Weiter aus.
 3. Schritt 3. Anfügen von Richtlinien an das Zertifikat
Suchen Sie nach der Berechtigungsrichtlinie, die Sie in erstellt haben [the section called “6. Erstellen der AWS IoT-Richtlinie”](#).
Aktivieren Sie das Kontrollkästchen neben Ihrer Richtlinie und wählen Sie Objekt erstellen aus.
 - f. Laden Sie im daraufhin angezeigten Fenster die folgenden Dateien herunter:

- Gerätezertifikat. Dies ist das X.509-Zertifikat.
- Datei mit öffentlichem Schlüssel
- Datei mit privatem Schlüssel
- Amazon-Trust-Services-Endpunkt (RSA-2048-Bit-Schlüssel: Amazon Root CA 1)

Notieren Sie sich den Speicherort jeder dieser Dateien für einen späteren Schritt.

- g. Wählen Sie Erledigt aus. Auf der nächsten Seite sehen Sie eine Notiz, dass Ihr Objekt erfolgreich erstellt wurde.
 - h. Übertragen Sie die oben heruntergeladenen Dateien auf Ihr AWS IoT Objekt, falls noch nicht vorhanden.
2. Rufen Sie den Anmeldeinformationsanbieter-Endpunkt für Ihr AWS Konto ab.

AWS CLI

Führen Sie den folgenden Befehl aus:

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

AWS Management Console

Führen Sie [AWS CloudShell](#) in den folgenden Befehl aus:

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

Notieren Sie sich diese Informationen für einen späteren Schritt.

3. Rufen Sie den Gerätedatenendpunkt für Ihr AWS Konto ab.

AWS CLI

Führen Sie den folgenden Befehl aus:

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

AWS Management Console

Gehen Sie wie folgt vor:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS IoT CoreKonsole unter <https://console.aws.amazon.com/iot/>.
2. Wählen Sie in der linken Navigation Einstellungen aus.
3. Suchen Sie den Gerätedatenendpunkt .

Notieren Sie sich diese Informationen für einen späteren Schritt.

4. (Optional) Überprüfen Sie, ob Ihre Zertifikate korrekt generiert wurden.

Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Ihre Elemente korrekt generiert wurden.

```
curl --header "x-amzn-iot-thingname:your-thing-name" \  
  --cert /path/to/certificateID-certificate.pem.crt \  
  --key /path/to/certificateID-private.pem.key \  
  --cacert /path/to/AmazonRootCA1.pem \  
  https://your-credential-provider-endpoint/role-aliases/your-role-alias-name/  
  credentials
```

Weitere Informationen finden Sie unter [Verwenden eines Zertifikats zum Abrufen eines Sicherheitstokens](#).

Schritt 8: Erstellen und Ausführen des Edge-Agenten von Amazon Kinesis Video Streams

Erstellen und Ausführen des Edge-Agenten von Amazon Kinesis Video Streams

1. Laden Sie die tar Datei über den Link herunter, der Ihnen zur Verfügung gestellt wurde.

Wenn Sie das Interessenformular für Amazon Kinesis Video Streams Edge Agent ausgefüllt haben, überprüfen Sie Ihre E-Mail auf den Download-Link. Wenn Sie das Formular noch nicht ausgefüllt haben, füllen Sie es [hier aus](#).

2. Überprüfen Sie die Prüfsumme.
3. Extrahieren Sie die Binärdateien und das JAR auf Ihrem Gerät.

Typ:`tar -xvf kvs-edge-agent.tar.gz`.

Nach der Extraktion sieht Ihre Ordnerstruktur wie folgt aus:

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

Note

Der Name des Release-Ordners sollte so eingerichtet werden, dass er die neueste binäre Versionsnummer widerspiegelt. Bei einer Version 1.0.0 ist der Ordnername beispielsweise auf 1.0.0 festgelegt.

4. Erstellen Sie das Abhängigkeits-JAR.

Note

Das im enthaltene JAR `kvs-edge-agent.tar.gz` hat nicht die Abhängigkeiten. Gehen Sie wie folgt vor, um diese Bibliotheken zu erstellen.

Navigieren Sie zu dem `kvs-edge-agent` Ordner, der enthält `pom.xml`.

Typ `mvn clean package`.

Dadurch wird eine JAR-Datei generiert, die die Abhängigkeiten enthält, die der Amazon Kinesis Video Streams Edge Agent unter `benötigt kvs-edge-agent/target/libs.jar`.

- Platzieren Sie den `libs.jar` in dem Ordner, der die Artefakte der Komponente enthält.

Typ `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/`.

- Legen Sie Umgebungsvariablen anhand der Werte aus den vorherigen Schritten fest. Die folgende Tabelle enthält Beschreibungen für die Variablen.

Name der Umgebungsvariablen	Erforderlich	Beschreibung
<code>AWS_REGION</code>	Ja	Die Region, die verwendet wird. Beispiel: <code>us-west-2</code>
<code>AWS_IOT_CA_CERT</code>	Ja	Dateipfad zum CA-Zertifikat, das verwendet wird, um eine Vertrauensstellung mit dem Backend-Service über TLS herzustellen. Beispiel: <code>/file/path/to/AmazonRootCA1.pem</code>
<code>AWS_IOT_CORE_CERT</code>	Ja	Dateipfad zum X.509-Zertifikat. Beispiel: <code>/file/path/to/certificateID-certificate.pem.crt</code>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	Ja	<p>Der Endpunkt AWS IoT Core des Anmeldeinformationen für Ihr AWS Konto.</p> <p>Beispiel: <i>credential-account-specific-prefix</i>.credentials.iot.<i>aws-region</i>.amazonaws.com</p>
AWS_IOT_CORE_DATA_ATS_ENDPOINT	Ja	<p>Der AWS IoT Core Datenebenen-Endpunkt für Ihr AWS Konto.</p> <p>Beispiel: <i>data-account-specific-prefix</i>.iot.<i>aws-region</i>.amazonaws.com</p>
AWS_IOT_CORE_PRIVATE_KEY	Ja	<p>Dateipfad zum privaten Schlüssel, der im öffentlichen/privaten Schlüsselpaar verwendet wird. Weitere Informationen finden Sie unter Schlüsselverwaltung in AWS IoT.</p> <p>Beispiel: <i>/file/path/to/certificateID-private</i>.pem.key</p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
<code>AWS_IOT_CORE_ROLE_ALIAS</code>	Ja	<p>Der Name des Rollenalias, der auf die AWS IAM-Rolle verweist, die beim Herstellen einer Verbindung mit verwendet werden soll AWS IoT Core.</p> <p>Beispiel: <code>kvs-edge-role-alias</code></p>
<code>AWS_IOT_CORE_THING_NAME</code>	Ja	<p>Der Name des AWS IoT Objekts, auf dem die Anwendung ausgeführt wird.</p> <p>Beispiel: <code>my-edge-device-thing</code></p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
GST_PLUGIN_PATH	Ja	<p>Dateipfad, der auf den Ordner verweist, der die IngestorPipelineJNI plattformabhängigen Bibliotheken gstkvsink und enthält. Ermöglicht es GStreamer, diese Plugins zu laden. Weitere Informationen finden Sie unter Herunterladen, Erstellen und Konfigurieren des GStreamer-Elements.</p> <p>Beispiel: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
LD_LIBRARY_PATH	Ja	<p>Dateipfad, der auf das Verzeichnis verweist, das die KinesisVideoProducer plattformabhängigen Bibliotheken <code>cproducer</code> und enthält.</p> <p>Beispiel: <code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Nein	<p>Bestimmt, ob der Edge-Agent von Amazon Kinesis Video Streams Zustandsmetriken auf veröffentlichtAmazon CloudWatch.</p> <p>Akzeptierte Werte: TRUE/FALSE (Groß- und Kleinschreibung wird nicht berücksichtigt). Standardmäßig ist , FALSE wenn nicht angegeben.</p> <p>Beispiel: FALSE</p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_LEVEL	Nein	<p>Die Stufe der Protokollierung der Edge-Agent-Ausgaben von Amazon Kinesis Video Streams.</p> <p>Zulässige Werte:</p> <ul style="list-style-type: none">• OFF• ALL• FATAL• ERROR• WARN• INFO, Standard, falls nicht angegeben• DEBUG• TRACE <p>Beispiel: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	Nein	<p>Sobald die Protokolldatei diese Größe erreicht hat, wird ein Rollover durchgeführt.</p> <ul style="list-style-type: none">• Min.: 0• Max.: 10 000• Standard: 20, falls nicht angegeben• Einheiten: Megabyte (MB) <p>Beispiel: 5</p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	Nein	<p>Der Dateipfad, der auf das Verzeichnis verweist, in dem die Amazon Kinesis Video Streams Edge Agent-Protokolle ausgegeben werden. Der Standardwert ist <code>./log</code>, wenn nicht angegeben.</p> <p>Beispiel: <code>./file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Nein	<p>Die Anzahl der Rollover-Protokolle, die vor dem Löschen aufbewahrt werden sollen.</p> <ul style="list-style-type: none">• Min.: 1• Max.: 100• Standard: 10, falls nicht angegeben <p>Beispiel: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Nein	<p>Dateipfad, der auf die aufgezeichneten Verzeichnisse verweist, wird in geschrieben. Ist standardmäßig das aktuelle Verzeichnis, falls nicht angegeben.</p> <p>Beispiel: <code>./file/path/</code></p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
GST_DEBUG	Nein	Gibt die Ebene der auszugebenden GStreamer-Protokolle an. Weitere Informationen finden Sie in der GStreamer-Dokumentation . Beispiel: 0
GST_DEBUG_FILE	Nein	Gibt die Ausgabedatei der GStreamer-Debug-Protokolle an. Wenn diese Option nicht festgelegt ist, erhalten Debug-Protokolle die Ausgabe des Standardfehlers. Weitere Informationen finden Sie in der GStreamer-Dokumentation . Beispiel: <code>/tmp/gstreamer-logging.log</code>

7. Löschen Sie den GStreamer-Cache. Type:

```
rm ~/.cache/gstreamer-1.0/registry.your-os-architecture.bin
```

Weitere Informationen finden Sie in der [GStreamer-Registrierungsdokumentation](#).

8. Bereiten Sie den Java-Befehl vor und führen Sie ihn aus. Der Amazon Kinesis Video Streams Edge Agent akzeptiert die folgenden Argumente:


Java-Eigenschaftsname	Erforderlich	Beschreibung
<code>java.library.path</code>	Nein	Dateipfad, der auf den Ordner verweist, der die IngestorPipelineJN

Java-Eigenschaftsname

Erforderlich

Beschreibung

I abhängigen Bibliotheken `gstkvsink` und enthält. Falls nicht angegeben, sucht der Amazon Kinesis Video Streams Edge Agent im aktuellen Verzeichnis nach ihnen.

 **Important**

Der Amazon Kinesis Video Streams Edge Agent funktioniert nicht richtig, wenn er diese Dateien nicht finden kann.

Beispiel: `/file/path/`

Um diese festzulegen, fügen Sie `-Djava-property-name=value` dem Java-Befehl hinzu, der zum Ausführen der JAR verwendet wird.

Beispielsweise:

```
java -Djava.library.path=download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion \
--add-opens java.base/jdk.internal.misc=ALL-UNNAMED \
-Dio.netty.tryReflectionSetAccessible=true \
-cp kvs-edge-agent.jar:libs.jar \
com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

⚠ Important

Führen Sie den obigen Java-Befehl aus demselben Verzeichnis wie aus `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion`.

9. Senden Sie Konfigurationen mit der an die Anwendung AWS CLI.

a. Erstellen Sie eine neue Datei, `example-edge-configuration.json`.

Fügen Sie folgenden Code in die Datei ein. Dies ist eine Beispielkonfiguration, die täglich von 9:00:00 bis 16:59:59 Uhr aufzeichnet (abhängig von der Systemzeit auf Ihrem AWS IoT Gerät). Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 bis 21:59:59 Uhr hochgeladen.

Weitere Informationen finden Sie unter [the section called "StartEdgeConfigurationUpdate"](#).

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
      },
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "DeletionConfig": {
```

```
        "EdgeRetentionInHours": 15,
        "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": 2800,
            "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
        },
        "DeleteAfterUpload": true
    }
}
```

- b. Um die Datei an den Amazon Kinesis Video Streams Edge Agent zu senden, geben Sie Folgendes in ein AWS CLI:

```
aws kinesishvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"
```

10. Wiederholen Sie den vorherigen Schritt für jeden Stream für den Amazon Kinesis Video Streams Edge Agent.

Schritt 9: (Optional) Installieren des CloudWatch Agenten auf dem Gerät

Note

Beachten Sie die [CloudWatch Kontingente](#).

Gehen Sie wie folgt vor, um den CloudWatch Agenten so zu installieren und zu konfigurieren, dass die vom Amazon Kinesis Video Streams Edge Agent generierten Protokolle automatisch in hochgeladen werden CloudWatch.

[Verfahren](#) zum Installieren des CloudWatch Agenten auf Ihrem Gerät finden Sie im Amazon CloudWatch -Benutzerhandbuch.

Wenn Sie zur Konfiguration aufgefordert werden, wählen Sie eine der folgenden Konfigurationen aus.

Important

`file_path` In den folgenden Konfigurationen wird davon ausgegangen, dass der Standardspeicherort für die Protokollierung verwendet wird.

Der verwendete Dateipfad geht davon aus, dass Sie den Edge-Agenten von Amazon Kinesis Video Streams vom Speicherort aus ausführen: *download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*version*.

- Um den CloudWatch Agenten für das Hochladen von Protokollen und die RAM- und CPU-Metriken nach dem Gerät zu konfigurieren, fügen Sie Folgendes in die Konfigurationsdatei ein.

```
{
  "agent": {
    "run_as_user": "ubuntu",
    "metrics_collection_interval": 60
  },
  "metrics": {
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      },
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true,
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
```



```

    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_edge.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvs_edge.log"
    },
    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_streams.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvs_streams.log"
    },
    {
      "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvssink.log*",
      "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
      "log_stream_name": "YourIoTThingName-cpp_kvssink.log"
    }
  ]
}
}
}
}
}

```

Schritt 10: (Optional) Führen Sie den Edge-Agenten von Amazon Kinesis Video Streams als nativen Prozess aus

Richten Sie den Edge-Agenten von Amazon Kinesis Video Streams als systemd-Service ein.

systemd ist ein System- und Servicemanager auf Linux-Geräten. systemd ist die empfohlene Methode zur Verwaltung des Prozesses, da der Amazon Kinesis Video Streams Edge Agent neu gestartet wird, falls die Anwendung auf einen Fehler stößt oder das Gerät, auf dem die Anwendung ausgeführt wird, die Stromversorgung verliert.

Gehen Sie wie folgt vor:

Führen Sie den Edge-Agenten von Amazon Kinesis Video Streams als nativen Prozess aus

1. Erstellen Sie eine neue Datei in `/etc/systemd/system` und benennen Sie sie `aws.kinesisvideo.edge-runtime-agent.service`.

Fügen Sie Folgendes ein:

```
[Unit]
Description=AWS Kinesis Video Streams edge agent
After=network.target
StartLimitBurst=3
StartLimitInterval=30

[Service]
Type=simple
Restart=on-failure
RestartSec=10
WorkingDirectory=/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
Environment="GST_PLUGIN_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion"
Environment="LD_LIBRARY_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib"
...
Environment="AWS_IOT_CORE_DATA_ATS_ENDPOINT=data-account-specific-prefix.iot.aws-
region.amazonaws.com"
ExecStart=/usr/lib/jvm/java-11-amazon-corretto/bin/java --add-opens java.base/
jdk.internal.misc=ALL-UNNAMED -Dio.netty.tryReflectionSetAccessible=true -cp kvs-
edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp

[Install]
WantedBy=multi-user.target
```

Weitere Informationen zu den von der `systemd` Servicekonfigurationsdatei akzeptierten Parametern finden Sie in der [Dokumentation](#).

Note

Fügen Sie die erforderlichen Umgebungsvariablen am ... Speicherort hinzu, wie unter angegeben [the section called “8. Erstellen und Ausführen des Edge-Agenten von Amazon Kinesis Video Streams”](#).

2. Laden Sie die Servicedateien neu, um den neuen Service einzuschließen.

Typ `sudo systemctl daemon-reload`.

3. Starten Sie den Service.

Typ `sudo systemctl start aws.kinesisvideo.edge-runtime-agent.service`.

4. Überprüfen Sie den Status des Amazon Kinesis Video Streams Edge Agent-Services, um zu überprüfen, ob er ausgeführt wird.

Typ `sudo systemctl status aws.kinesisvideo.edge-runtime-agent.service`.

Im Folgenden finden Sie ein Beispiel für die Ausgabe, die Sie sehen werden.

```
aws.kinesisvideo.edge-runtime-agent.service - AWS Kinesis Video Streams edge agent
  Loaded: loaded (/etc/systemd/system/aws.kinesisvideo.edge-runtime-agent.service; disabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-06-08 19:15:02 UTC; 6s ago
  Main PID: 506483 (java)
  Tasks: 23 (limit: 9518)
  Memory: 77.5M
  CPU: 4.214s
  CGroup: /system.slice/aws.kinesisvideo.edge-runtime-agent.service
          ##506483 /usr/lib/jvm/java-11-amazon-corretto/bin/java -cp kvs-edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

5. Überprüfen Sie die Protokolle auf Fehler.

Typ `journalctl -e -u aws.kinesisvideo.edge-runtime-agent.service`.

6. Geben Sie `systemctl --help` für die vollständige Liste der Optionen ein, um den Prozess mit zu verwalten `systemctl`.

Im Folgenden finden Sie einige gängige Befehle zur Verwaltung des Amazon Kinesis Video Streams Edge Agent:

- Geben Sie zum Neustart ein `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`.
- Geben Sie zum Anhalten ein `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Um bei jedem Geräteneustart automatisch zu starten, geben Sie ein `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Bereitstellen des Edge-Agenten von Amazon Kinesis Video Streams in AWS IoT Greengrass

Gehen Sie wie folgt vor, um den Edge-Agenten von Amazon Kinesis Video Streams in bereitzustellen, AWS IoT Greengrass um Medien von IP-Kameras aufzuzeichnen und hochzuladen.

Themen

- [Schritt 1: Erstellen einer UbuntuAmazon EC2Instance](#)
- [Schritt 2: Einrichten des AWS IoT Greengrass V2 Core-Geräts auf dem Gerät](#)
- [Schritt 3: Erstellen der Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera-RTSP-URLs](#)
- [Schritt 4: Hinzufügen von Berechtigungen zur Rolle des Token Exchange Service \(TES\)](#)
- [Schritt 5: Installieren der AWS IoT Greengrass Secret-Manager-Komponente auf dem Gerät](#)
- [Schritt 6: Bereitstellen der Edge-Agent-AWS IoT GreengrassKomponente von Amazon Kinesis Video Streams auf dem Gerät](#)
- [Schritt 7: \(Optional\) Installieren der AWS IoT Greengrass Log Manager-Komponente auf dem Gerät](#)

Schritt 1: Erstellen einer UbuntuAmazon EC2Instance

Gehen Sie wie folgt vor, um eine Ubuntu-Amazon EC2Instance zu erstellen.

Erstellen einer UbuntuAmazon EC2-Instance

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-EC2-Konsole unter <https://console.aws.amazon.com/EC2/>.

Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.

2. Wählen Sie Launch Instance aus.

Füllen Sie die folgenden Felder aus:

- Name – Geben Sie einen Namen für die Instance ein.
- Anwendungs- und Betriebssystemabbilder (Amazon Machine Image) – Wählen Sie Ubuntu aus.
- Instance-Typ – Wählen Sie t2.large aus.

- Anmeldung bei Schlüsselpaaren – Erstellen Sie Ihr eigenes Schlüsselpaar.
 - Netzwerkeinstellungen – Behalten Sie die Standardeinstellung bei.
 - Speicher konfigurieren – Erhöhen Sie das Volume auf 256 GiB.
 - Erweiterte Einstellungen – Behalten Sie die Standardeinstellung bei.
3. Starten Sie die Instance und gehen Sie ein SSH ein.

Gehen Sie wie folgt vor:

1. Wählen Sie im linken Navigationsbereich Instances und dann die Instance-ID aus.
 2. Wählen Sie oben rechts Verbinden aus.
 3. Wählen Sie SSH-Client und folgen Sie den Anweisungen auf dem Bildschirm.
 4. Öffnen Sie ein Terminal und navigieren Sie zur heruntergeladenen `.pem` Datei (wahrscheinlich in `~/Downloads`).
 5. Wenn Sie diese Verfahren zum ersten Mal befolgen, erhalten Sie die Meldung „Die Authentizität des Hosts (...) kann nicht hergestellt werden“. Geben Sie Ja ein.
4. Installieren Sie Systembibliotheken, um den Edge-Agenten von Amazon Kinesis Video Streams auf der Instance zu erstellen.

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -  
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'  
  
sudo apt-get update  
  
sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \  
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \  
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \  
gstreamer1.0-plugins-good gstreamer1.0-tools \  
unzip java-11-amazon-corretto-jdk maven
```

Important

Wenn ein Bildschirm angezeigt wird, in dem Sie darüber informiert werden, dass einige Services neu gestartet werden müssen, drücken Sie die Eingabetaste, um OK auszuwählen.

Weitere Informationen finden Sie im [Amazon Corretto 11-Benutzerhandbuch](#).

Schritt 2: Einrichten des AWS IoT Greengrass V2 Core-Geräts auf dem Gerät

Gehen Sie wie folgt vor, um die AWS IoT Greengrass Core-Kernkern-Software auf der Amazon EC2 zu installieren.

Einrichten des AWS IoT Greengrass Core-Geräts

1. Melden Sie sich bei der anAWS Management Console, <https://console.aws.amazon.com/iot/>.

Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.
2. Wählen Sie in der linken Navigation Greengrass-Geräte , Core-Geräte aus.
3. Wählen Sie Ein Core-Gerät einrichten aus.
4. Führen Sie die Schritte auf dem Bildschirm aus.
 - Schritt 1: Registrieren eines Greengrass-Core-Geräts. Geben Sie einen Namen für das Gerät ein.
 - Schritt 2: Hinzufügen zu einer Objektgruppe, um eine kontinuierliche Bereitstellung anzuwenden. Wählen Sie Keine Gruppe aus.
 - Schritt 3: Installieren der Greengrass Core-Software . Wählen Sie Linux aus.
 - Schritt 3.1: Installieren von Java auf dem Gerät


Java wird als Teil von installiert [the section called "1. Erstellen einer Ubuntu-Instance"](#). Kehren Sie zu diesem Schritt zurück, wenn Java noch nicht installiert ist.
 - Schritt 3.2: Kopieren von AWS Anmeldeinformationen auf das Gerät

Öffnen Sie die bash/zsh Option und fügen Sie die Exportbefehle in die Amazon EC2 ein.
 - Schritt 3.3: Ausführen des Installationsprogramms
 1. Kopieren und führen Sie die Befehle Installationsprogramm herunterladen und Installationsprogramm ausführen in der Ubuntu-Amazon EC2-Instance aus.

Note

Der Befehl Installationsprogramm ausführen wird automatisch basierend auf dem Namen aktualisiert, den Sie in einem vorherigen Schritt ausgewählt haben.

2. Notieren Sie sich die erstellte Token Exchange Service (TES)-Rolle. Sie benötigen sie später.

 Note

Standardmäßig heißt die erstellte Rolle GreengrassV2TokenExchangeRole.

Schritt 3: Erstellen der Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera-RTSP-URLs

Gehen Sie wie folgt vor, um die in benötigten Streams und Secrets zu erstellen AWS Secrets Manager. Führen Sie diesen Schritt zuerst aus, da Sie die ARNs der erstellten Ressourcen in den Richtlinien benötigen.

Erstellen von Amazon Kinesis Video Streams

Erstellen Sie Amazon Kinesis Video Streams mithilfe der AWS CLI, oder AWS Management Console API.

AWS Management Console Öffnen Sie in der die [Amazon Kinesis Video Streams-Konsole](#) . Wählen Sie im linken Navigationsbereich Videostreams aus.

Weitere Informationen finden Sie unter [the section called "Einen Kinesis-Videostream erstellen"](#).

Erstellen von Geheimnissen in AWS Secrets Manager

AWS Management Console Öffnen Sie in der die [-AWS Secrets Manager Konsole](#) . Wählen Sie in der linken Navigation Secrets aus.


Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.

1. Wählen Sie Store a new secret (Ein neues Secret speichern).

- a. Schritt 1: Auswählen des Secret-Typs

- Wählen Sie Anderer Geheimnistyp aus.
- Fügen Sie im Abschnitt Schlüssel/Wert-Paare ein Schlüssel-Wert-Paar hinzu.

Schlüssel: MediaURI

 Note

Der Schlüssel muss sein `MediaURI`. Dabei wird zwischen Groß- und Kleinschreibung unterschieden. Wenn Sie sie falsch eingeben, funktioniert die Anwendung nicht.

Wert : *Your MediaURI*.


Example

Beispiel: `rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI`.

- b. Schritt 2: Konfigurieren von Secret . Geben Sie diesem Secret einen Namen. Benennen Sie sie nach Belieben.
 - c. Schritt 3: Konfigurieren der Rotation – optional . Wählen Sie Weiter aus.
 - d. Schritt 4: Überprüfen von . Wählen Sie Store (Speichern) aus.
2. Wenn Ihr Secret nicht sofort angezeigt wird, wählen Sie die Schaltfläche Aktualisieren aus.

Wählen Sie den Namen Ihres Secrets aus. Notieren Sie sich den geheimen ARN .

3. Wiederholen Sie diesen Vorgang für jeden MediaURI, von dem Sie streamen möchten.

 Note

Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der Amazon EC2-Instance aus auf diese zugreifen oder wenn Sie nicht verwaltet werden, während Sie mit dem VPN verbunden sind.

 Important

Ihre Kamera-RTSP-URL sollte Videos im h.264-Format streamen. Die Fragmentdauer darf das in angegebene Limit nicht überschreiten [the section called “Beschränkungen des Producer-SDK”](#).

Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.

Führen Sie aus, `gst-discoverer-1.0` *Your RtspUrl* um sicherzustellen, dass Ihre Kamera von Ihrem Gerät aus erreichbar ist.

Speichern Sie die ARNs für alle Streams und Secrets, die Sie erstellt haben. Sie benötigen diese für den nächsten Schritt.

Schritt 4: Hinzufügen von Berechtigungen zur Rolle des Token Exchange Service (TES)

Erteilen Sie der Token Exchange Service (TES)-Rolle dem Gerät, das Berechtigungen zum Betrachten der Secrets übernimmt. Dies ist erforderlich, damit die AWS Secrets Manager AWS IoT Greengrass Komponente ordnungsgemäß funktioniert.

Hinzufügen von Berechtigungen zur TES-Rolle

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich Rollen und suchen Sie nach der TES-Rolle, die Sie zuvor im Prozess erstellt haben.
3. Wählen Sie in der Dropdownliste Berechtigungen hinzufügen die Option Richtlinien anfügen aus.
4. Wählen Sie Richtlinie erstellen aus.
5. Scrollen Sie nach unten und wählen Sie Bearbeiten aus.
6. Wählen Sie im Richtlinieneditor JSON aus und bearbeiten Sie die Richtlinie.

Ersetzen Sie die Richtlinie durch Folgendes:

Note

Ersetzen Sie `arn:aws:kinesisvideo:*:*:stream/streamName1/*` und `arn:aws:kinesisvideo:*:*:stream/streamName2/*` durch die ARNs für die Streams, die Sie in einem vorherigen Schritt erstellt haben.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "kinesisvideo:ListStreams"
        ],
        "Resource": "*"
      },
      {
        "Effect": "Allow",
        "Action": [
          "kinesisvideo:DescribeStream",
          "kinesisvideo:PutMedia",
          "kinesisvideo:TagStream",
          "kinesisvideo:GetDataEndpoint"
        ],
        "Resource": [
          "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
          "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
        ]
      }
    ]
  }

```

7. Wählen Sie auf der Seite Add tags (Tags hinzufügen) die Option Next: Review (Weiter: Prüfen) aus.
8. Benennen Sie Ihre Richtlinie und wählen Sie dann Richtlinie erstellen aus.

Ein Beispiel für einen Richtliniennamen ist KvsEdgeAccessPolicy.

9. Schließen Sie die Registerkarte und kehren Sie zu der Registerkarte zurück, auf der Sie der TES-Rolle eine Richtlinie angefügt haben.

Wählen Sie die Schaltfläche Aktualisieren und suchen Sie dann nach der neu erstellten Richtlinie.

Aktivieren Sie das Kontrollkästchen und wählen Sie Richtlinien anfügen aus.

Auf dem nächsten Bildschirm sehen Sie eine Notiz, die besagt, dass die Richtlinie erfolgreich an die Rolle angefügt wurde.

10. Erstellen und fügen Sie dieses Mal eine weitere Richtlinie für Ihre Secrets an.

Ersetzen Sie die Richtlinie durch Folgendes:

Note

Ersetzen Sie durch `arn:aws:secretsmanager:*:*:secret:*` die ARNs, die die MediaURI-Secrets enthalten, die Sie in erstellt haben [the section called “3. Erstellen von Ressourcen für Ihre IP-Kamera-RTSP-URLs”](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
      ]
    }
  ]
}
```

11. Erstellen und fügen Sie dieses Mal eine andere Richtlinie für -Amazon CloudWatchMetriken an. Ersetzen Sie die Richtlinie durch Folgendes:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```


Schritt 5: Installieren der AWS IoT Greengrass Secret-Manager-Komponente auf dem Gerät

Der Edge-Agent von Amazon Kinesis Video Streams erfordert, dass die AWS IoT Greengrass Secret-Manager-Komponente zuerst auf dem Gerät installiert wird.

Installieren der Secret Manager-Komponente

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS IoT Core-Konsole unter <https://console.aws.amazon.com/iot/>. Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.
2. Wählen Sie in der linken Navigation Greengrass-Geräte, Bereitstellungen aus.

Wählen Sie die Bereitstellung mit demselben Ziel wie das Objekt aus, das wir in erstellt haben [the section called "2. Einrichten des AWS IoT Greengrass Core-Geräts"](#).


3. Wählen Sie in der Dropdownliste Aktionen in der oberen rechten Ecke die Option Überarbeiten aus.

Wählen Sie im daraufhin angezeigten Popup-Fenster Bereitstellung überarbeiten aus.

4. Füllen Sie die folgenden Abschnitte aus:
 - Schritt 1: Geben Sie das Ziel an. Wählen Sie Weiter aus.
 - Schritt 2: Komponenten auswählen.
 - Stellen Sie sicher, dass die Komponente `aws.greengrass.Cli` ausgewählt ist. Deinstallieren Sie diese Komponente nicht.
 - Aktivieren Sie den Schalter Nur ausgewählte Komponenten anzeigen und suchen Sie nach `aws.greengrass.SecretManager`
 - Aktivieren Sie das Kontrollkästchen neben `aws.greengrass` und `SecretManager` wählen Sie dann Weiter aus.
 - Schritt 3: Konfigurieren von Komponenten . Konfigurieren Sie die AWS IoT Greengrass Secret Manager-Komponente, um die Secrets aus der AWS IoT Greengrass Umgebung herunterzuladen.


Wählen Sie die Komponente `aws.greengrass.SecretManager` und dann Komponente konfigurieren aus.

Aktualisieren Sie auf dem daraufhin angezeigten Bildschirm die AWS Secrets Manager ARNs im Feld Konfiguration für Zusammenführung.

 Note

Ersetzen Sie durch `arn:aws:secretsmanager:*:*:secret:*` die ARNs der Secrets, die Sie in erstellt haben [the section called “3. Erstellen von Ressourcen für Ihre IP-Kamera-RTSP-URLs”](#).

```
{
  "cloudSecrets": [
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    },
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    }
  ]
}
```

 Note

`cloudSecrets` ist eine Liste von Objekten mit dem Schlüssel `arn`. Weitere Informationen finden Sie im Abschnitt [Secret-Manager-Konfiguration](#) im *-AWS IoT Greengrass Version 2* Entwicklerhandbuch.

Wenn Sie fertig sind, wählen Sie **Bestätigen** und dann **Weiter** aus.

- Schritt 4: Konfigurieren der erweiterten Einstellungen . Klicken Sie auf **Weiter**.
 - Schritt 5: Überprüfen von . Wählen Sie **Bereitstellen** aus.
5. Vergewissern Sie sich, dass die AWS Secrets Manager Komponente und die Berechtigungen korrekt installiert wurden.

Geben Sie auf der Ubuntu Amazon EC2-Instance ein, `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` um zu überprüfen, ob die Komponente die aktualisierte Konfiguration erhalten hat.

6. Überprüfen Sie die AWS IoT Greengrass Core-Protokolle.

```
Typ sudo less /greengrass/v2/logs/greengrass.log.
```

Überprüfen Sie auf Bereitstellungsfehler.

Wenn ein Fehler aufgetreten ist, überarbeiten Sie die Bereitstellung, um die `aws.greengrass.SecretManager` Komponente zu entfernen.

Geben Sie ein `sudo service greengrass restart`, um den AWS IoT Greengrass Core-Service neu zu starten.

Wenn der Bereitstellungsfehler mit fehlenden Berechtigungen zusammenhängt, überprüfen Sie den [the section called "4. Hinzufügen von Berechtigungen zur TES-Rolle"](#) Abschnitt, um sicherzustellen, dass die TES-Rolle über die richtigen Berechtigungen verfügt. Wiederholen Sie diesen Abschnitt.

Aktualisieren der Secrets auf der AWS IoT Greengrass Secret-Manager-Komponente

Important

Die AWS IoT Greengrass Secret-Manager-Komponente ruft Secrets nur ab und speichert sie im Cache, wenn die Bereitstellung aktualisiert wird.

Um die Secrets auf der AWS IoT Greengrass Secret Manager-Komponente zu aktualisieren, führen Sie die vorherigen Schritte 1–6 mit der folgenden Änderung aus.

Schritt 3: Konfigurieren von Komponenten . Konfigurieren Sie die AWS IoT Greengrass Secret Manager-Komponente, um die Secrets aus der AWS IoT Greengrass Umgebung herunterzuladen.

Wählen Sie die Komponente `aws.greengrass.SecretManager` und dann Komponente konfigurieren aus.

Fügen Sie auf dem daraufhin angezeigten Bildschirm [""] das Feld Pfade zurücksetzen ein und aktualisieren Sie die AWS Secrets Manager ARNs im Feld Konfiguration zum Zusammenführen.

Weitere Informationen finden Sie unter [Updates zurücksetzen](#).

Schritt 6: Bereitstellen der Edge-Agent-AWS IoT GreengrassKomponente von Amazon Kinesis Video Streams auf dem Gerät

Bereitstellen der Edge-Agent-AWS IoT GreengrassKomponente von Amazon Kinesis Video Streams auf dem Gerät

1. Laden Sie die tar Datei über den bereitgestellten Link herunter.

Wenn Sie das Interessenformular für Amazon Kinesis Video Streams Edge Agent ausgefüllt haben, überprüfen Sie Ihre E-Mail auf den Download-Link. Wenn Sie das Formular noch nicht ausgefüllt haben, füllen Sie es [hier aus](#).

2. Überprüfen Sie die Prüfsumme.
3. Extrahieren Sie die Binärdateien und das JAR auf Ihrem Gerät.

```
Typ:tar -xvf kvs-edge-agent.tar.gz.
```

Nach der Extraktion sieht Ihre Ordnerstruktur wie folgt aus:

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config

kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
```

```
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

Note

Der Name des Release-Ordners sollte so eingerichtet werden, dass er die neueste binäre Versionsnummer widerspiegelt. Bei einer Version 1.0.0 ist der Ordnername beispielsweise auf 1.0.0 festgelegt.

4. Erstellen Sie die Abhängigkeits-jar.

Note

Das in der `kvs-edge-agent.tar.gz` enthaltene JAR hat nicht die Abhängigkeiten. Gehen Sie wie folgt vor, um diese Bibliotheken zu erstellen.

Navigieren Sie zu dem `kvs-edge-agent` Ordner, der enthält `pom.xml`.

Typ `mvn clean package`.

Dadurch wird eine JAR-Datei generiert, die die Abhängigkeiten enthält, die der Amazon Kinesis Video Streams Edge Agent unter `benötigtkvs-edge-agent/target/libs.jar`.

5. Platzieren Sie `libs.jar` in dem Ordner, der die Artefakte der Komponente enthält.

Typ `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/`
`aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/`.

6. Optional. Konfigurieren Sie Eigenschaften. Der Amazon Kinesis Video Streams Edge Agent akzeptiert die folgenden Umgebungsvariablen im -AWS IoT GreengrassModus:

Name der Umgebungsvariablen	Erforderlich	Beschreibung
<code>AWS_REGION</code>	Ja	Die verwendete Region.

Name der Umgebungsvariablen	Erforderlich	Beschreibung
GST_PLUGIN_PATH	Ja	<p data-bbox="1084 254 1360 289">Beispiel: us-west-2</p> <p data-bbox="1084 331 1507 751">AWS IoT Greengrass Die - Core-Software legt diesen Wert automatisch für Sie fest. Weitere Informationen finden Sie im Referenzthema Komponentenumgebungsvariablen im -AWS IoT Greengrass Version 2Entwicklerhandbuch.</p> <p data-bbox="1084 793 1507 1360">Dateipfad, der auf den Ordner verweist, der die IngestorPipelineJNI plattformabhängigen Bibliotheken gstkvsink und enthält. Auf diese Weise kann GStreamer diese Plugins laden. Weitere Informationen finden Sie unter Herunterladen, Erstellen und Konfigurieren des GStreamer-Elements .</p> <p data-bbox="1084 1402 1507 1772">Beispiel: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
LD_LIBRARY_PATH	Ja	<p>Dateipfad, der auf das Verzeichnis verweist, das die KinesisVideoProducer plattformabhängigen Bibliotheken <code>cproducer</code> und enthält.</p> <p>Beispiel: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Nein	<p>Bestimmt, ob der Edge-Agent von Amazon Kinesis Video Streams Zustandsmetriken auf veröffentlichtAmazon CloudWatch.</p> <p>Zulässige Werte: TRUE/FALSE (Groß- und Kleinschreibung wird nicht berücksichtigt). Der Standardwert ist FALSE, wenn nicht angegeben.</p> <p>Beispiel: FALSE</p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_LEVEL	Nein	<p>Die Stufe der Protokollierung der Edge-Agent-Ausgaben von Amazon Kinesis Video Streams.</p> <p>Zulässige Werte:</p> <ul style="list-style-type: none">• OFF• ALL• FATAL• ERROR• WARN• INFO, Standard, falls nicht angegeben• DEBUG• TRACE <p>Beispiel: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	Nein	<p>Sobald die Protokolldatei diese Größe erreicht hat, wird ein Rollover durchgeführt.</p> <ul style="list-style-type: none">• Min.: 1• Max.: 100• Standard: 20, falls nicht angegeben• Einheiten: Megabyte (MB) <p>Beispiel: 5</p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	Nein	<p>Der Dateipfad, der auf das Verzeichnis verweist, in dem die Amazon Kinesis Video Streams Edge Agent-Protokolle ausgegeben werden. Der Standardwert ist <code>./log</code>, wenn nicht angegeben.</p> <p>Beispiel: <code>./file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Nein	<p>Die Anzahl der Rollover-Protokolle, die vor dem Löschen aufbewahrt werden sollen.</p> <ul style="list-style-type: none">• Min.: 1• Max.: 100• Standard: 10, falls nicht angegeben <p>Beispiel: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Nein	<p>Dateipfad, der auf die aufgezeichneten Verzeichnisse verweist, wird in geschrieben. Ist standardmäßig das aktuelle Verzeichnis, falls nicht angegeben.</p> <p>Beispiel: <code>./file/path/</code></p>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
GREENGRASS_ROOT_DIRECTORY	Nein	<p>Der Dateipfad zum AWS IoT Greengrass Stammverzeichnis.</p> <p>Dies ist standardmäßig , /greengrass/v2/ wenn nicht angegeben.</p> <p>Beispiel: <i>/file/path/</i></p>
GST_DEBUG	Nein	<p>Gibt die Ebene der auszugebenden GStreamer-Protokolle an. Weitere Informationen finden Sie in der GStreamer-Dokumentation.</p> <p>Beispiel: 0</p>
GST_DEBUG_FILE	Nein	<p>Gibt die Ausgabedatei der GStreamer-Debug-Protokolle an. Wenn diese Option nicht festgelegt ist, erhalten Debug-Protokolle die Ausgabe zum Standardfehler. Weitere Informationen finden Sie in der GStreamer-Dokumentation.</p> <p>Beispiel: <i>/tmp/gstreamer-logging .log</i></p>

Öffnen Sie das Ausführungsskript `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` und ändern Sie es, um eine der vorherigen Umgebungsvariablen hinzuzufügen.

⚠ Important

Stellen Sie sicher, dass das geänderte Ausführungsskript keine Tabulatorzeichen enthält. Die AWS IoT Greengrass Kernsoftware kann das Rezept nicht lesen.

7. Stellen Sie die Edge-Agent-AWS IoT GreengrassKomponente von Amazon Kinesis Video Streams bereit.

Type:

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \  
  --recipeDir <download location>/kvs-edge-agent/KvsEdgeComponent/recipes/ \  
  --artifactDir <download location>/kvs-edge-agent/KvsEdgeComponent/artifacts/ \  
  --merge "aws.kinesisvideo.KvsEdgeComponent=EdgeAgentVersion"
```

Weitere Informationen finden Sie in den folgenden Abschnitten im AWS IoT Greengrass Version 2 -Entwicklerhandbuch:

- [AWS IoT Greengrass CLI-Befehle](#)
- [Bereitstellen von AWS IoT Greengrass Komponenten auf Geräten](#)

8. Senden Sie Konfigurationen mit der an die AnwendungAWS CLI.
 - a. Erstellen Sie eine neue Datei, *example-edge-configuration.json*.

Fügen Sie folgenden Code in die Datei ein. Dies ist eine Beispielkonfiguration, die täglich von 9:00:00 bis 16:59:59 Uhr aufzeichnet (abhängig von der Systemzeit auf Ihrem AWS IoT Gerät). Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 bis 21:59:59 Uhr hochgeladen.

Weitere Informationen finden Sie unter [the section called "StartEdgeConfigurationUpdate"](#).

```
{  
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",  
  "EdgeConfig": {  
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",  
    "RecorderConfig": {  
      "MediaSourceConfig": {
```

```
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
    },
    "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
    }
},
"UploaderConfig": {
    "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
    }
},
"DeletionConfig": {
    "EdgeRetentionInHours": 15,
    "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": 2800,
        "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
    },
    "DeleteAfterUpload": true
}
}
```

- b. Geben Sie Folgendes in die einAWS CLI, um die Datei an den Amazon Kinesis Video Streams Edge Agent zu senden:

```
aws kinesisvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"
```

9. Wiederholen Sie den vorherigen Schritt für jeden Stream für den Amazon Kinesis Video Streams Edge Agent.

Schritt 7: (Optional) Installieren der AWS IoT Greengrass Log Manager-Komponente auf dem Gerät

Note

Beachten Sie die [CloudWatch Kontingente](#).

Gehen Sie wie folgt vor, um die Edge-Agent-Protokolle von Amazon Kinesis Video Streams so zu konfigurieren, dass sie CloudWatch mithilfe der AWS IoT Greengrass Log-Manager-Komponente automatisch in hochgeladen werden.

Installieren der AWS IoT Greengrass Log Manager-Komponente

1. Vergewissern Sie sich, dass die AWS IoT Greengrass Geräterolle über die [entsprechenden Berechtigungen](#) verfügt.
 - a. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
 - b. Klicken Sie im linken Navigationsbereich auf Rollen.
 - c. Wählen Sie den Namen der TES-Rolle aus, die in erstellt wurde [the section called "2. Einrichten des AWS IoT Greengrass Core-Geräts"](#). Verwenden Sie bei Bedarf die Suchleiste.
 - d. Wählen Sie die `GreengrassV2TokenExchangeRoleAccess` Richtlinie aus.
 - e. Wählen Sie die Registerkarte JSON aus und überprüfen Sie, ob die Richtlinie wie folgt aussieht:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "s3:GetBucketLocation"
      ]
    }
  ],
}
```

```
        "Resource": "*"
    }
  ]
}
```

- f. Wenn die `GreengrassV2TokenExchangeRoleAccess` Richtlinie nicht vorhanden ist oder einige erforderliche Berechtigungen fehlen, erstellen Sie eine neue IAM-Richtlinie mit diesen Berechtigungen und fügen Sie sie der TES-Rolle an, die in erstellt wurde [the section called "2. Einrichten des AWS IoT Greengrass Core-Geräts"](#).
2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die -AWS IoT Core-Konsole unter <https://console.aws.amazon.com/iot/>. Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.
 3. Wählen Sie in der linken Navigation Greengrass-Geräte, Bereitstellungen aus.

Wählen Sie die Bereitstellung mit demselben Ziel wie das Objekt aus, das Sie in erstellt haben [the section called "2. Einrichten des AWS IoT Greengrass Core-Geräts"](#).
 4. Wählen Sie in der oberen rechten Ecke Aktionen und dann Überarbeiten aus.

Wählen Sie im daraufhin angezeigten Popup-Fenster Bereitstellung überarbeiten aus.
 5. Füllen Sie die folgenden Abschnitte aus:
 - a. Schritt 1: Geben Sie das Ziel an. Wählen Sie Weiter aus.
 - b. Schritt 2: Komponenten auswählen.
 - i. Stellen Sie sicher, dass die Komponenten `aws.greengrass.Cli` und `aws.greengrass.SecretManager` weiterhin ausgewählt sind.

 **Important**

Deinstallieren Sie diese Komponenten nicht.

- ii. Aktivieren Sie den Schalter Nur ausgewählte Komponenten anzeigen und suchen Sie nach `aws.greengrass.LogManager`
 - iii. Aktivieren Sie das Feld neben `aws.greengrass` und `LogManager` wählen Sie dann Weiter aus.
- c. Schritt 3: Komponenten konfigurieren. Konfigurieren Sie die AWS IoT Greengrass Log-Manager-Komponente, um die vom Amazon Kinesis Video Streams Edge Agent generierten Protokolle hochzuladen.

Wählen Sie die Komponente `aws.greengrass.LogManager` und dann Komponente konfigurieren aus.

Fügen Sie auf dem daraufhin angezeigten Bildschirm die folgende Protokollmanager-Konfiguration in das Feld Konfiguration zum Zusammenführen ein.

```
{
  "logsUploaderConfiguration": {
    "componentLogsConfigurationMap": {
      "aws.kinesisvideo.KvsEdgeComponent/java_kvs.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "java_kvs.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_edge.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_edge.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvssink.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvssink.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_streams.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_streams.log\\w*"
      }
    }
  },
  "periodicUploadIntervalSec": "1"
}
```

⚠ Important

Der `logfileDirectoryPath` in der vorherigen Konfiguration geht davon aus, dass der Standardspeicherort für die Protokollierung verwendet wird.

ℹ Note

Weitere Informationen zu den einzelnen Parametern für die Protokollmanager-Konfiguration finden Sie im Abschnitt [Protokollmanager](#) des -AWS IoT Greengrass Version 2Entwicklerhandbuchs.

Wenn Sie fertig sind, wählen Sie Bestätigen und dann Weiter aus.

- d. Schritt 4: Konfigurieren erweiterter Einstellungen. Klicken Sie auf Weiter.
 - e. Schritt 5: Überprüfen. Wählen Sie Bereitstellen aus.
6. Vergewissern Sie sich, dass die AWS Log Manager-Komponente und die Berechtigungen korrekt installiert wurden.
 7. Geben Sie auf der UbuntuAmazon EC2-Instance ein, `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` um zu überprüfen, ob die Komponente die aktualisierte Konfiguration erhalten hat.
 8. Überprüfen Sie die AWS IoT Greengrass Core-Protokolle.

Typ `sudo less /greengrass/v2/logs/greengrass.log`.

Überprüfen Sie auf Bereitstellungsfehler.

Wenn ein Fehler aufgetreten ist, ändern Sie die Bereitstellung, um die `aws.greengrass.LogManager` Komponente zu entfernen.

Geben Sie ein `sudo service greengrass restart`, um den AWS IoT Greengrass Core-Service neu zu starten.

Wenn der Bereitstellungsfehler mit fehlenden Berechtigungen zusammenhängt, überprüfen Sie [the section called “4. Hinzufügen von Berechtigungen zur TES-Rolle”](#) um sicherzustellen, dass die TES-Rolle über die richtigen Berechtigungen verfügt. Wiederholen Sie diesen Abschnitt.

Häufig gestellte Fragen zum Edge-Agenten von Amazon Kinesis Video Streams

Im Folgenden finden Sie einige häufig gestellte Fragen zum Amazon Kinesis Video Streams Edge Agent-Service.

Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent?

Amazon Kinesis Video Streams Edge Agent unterstützt derzeit die folgenden Betriebssysteme:

Ubuntu

- 22.x
 - AMD64
- 18.x
 - ARM

AL2

- amzn2
 - AMD64 amazonlinux:2.0.20210219.0-amd64 (Snowball)

Unterstützt der Edge Agent von Amazon Kinesis Video Streams H.265-Medien?

Amazon Kinesis Video Streams Edge Agent unterstützt nur H.264-Elementarstreams.

Funktioniert der Edge-Agent von Amazon Kinesis Video Streams in AL2?

Ja.

Wie kann ich mehrere Streams innerhalb des AWS IoT Objekts oder Geräts ausführen?

Senden Sie einen anderen [the section called "StartEdgeConfigurationUpdate"](#) an denselben `HubDeviceArn`, aber an verschiedene Amazon Kinesis Video Streams AWS Secrets Manager/ARNs.

Wie kann ich eine bearbeiten, **StartEdgeConfigurationUpdate** nachdem sie gesendet wurde?

Senden Sie ein aktualisiertes `HubDeviceArn` mit demselben Amazon Kinesis Video Streams-ARN [the section called "StartEdgeConfigurationUpdate"](#) an denselben . Wenn die Anwendung die Nachricht von Amazon Kinesis Video Streams erhält, überschreibt sie die vorherige Konfiguration für diesen Stream. Anschließend werden Änderungen vorgenommen.

Gibt es Beispiele für gängige **ScheduleConfigs**?

Der Amazon Kinesis Video Streams Edge Agent verwendet die Systemzeit des Geräts, auf dem er ausgeführt wird.

Beschreibung	ScheduleExpression	DurationInSeconds
Aufzeichnung rund um die Uhr, stündliches Hochladen	(Null ScheduleConfig)	
9:00:00 Uhr bis 16:59:59 Uhr jeden Tag	0 0 9-16 * * ? *	3599
9:00:00 AM - 4:59:59 PM weekdays	0 0 9-16 ? * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599
9:00:00 Uhr bis 16:59:59 Uhr an Wochenenden	0 0 9-16 ? * SAT,SUN *	3599
22:00:00 Uhr bis 23:59:59 Uhr an Wochentagen	0 0 22,23 ? * MON-FRI *	3599
9:00:00 Uhr bis 10:00:00 Uhr jeden Tag	0 0 9 * * ? *	3600

Beschreibung	ScheduleExpression	DurationInSeconds
16:00:00 Uhr bis 17:59:59 Uhr jeden Tag	0 0 16-17 * * ? *	3599

Weitere Beispiele finden Sie in der [microSDtz-Dokumentation](#).

Gibt es ein maximales Stream-Limit?

Der Amazon Kinesis Video Streams Edge Agent hat derzeit ein festes Limit von 16 Streams pro Gerät. Verwenden Sie die [the section called "DeleteEdgeConfiguration"](#) -API, um Streams von einem Gerät zu löschen. Durch das Aktualisieren einer Konfiguration für denselben Stream mithilfe der [the section called "StartEdgeConfigurationUpdate"](#) wird die Stream-Anzahl des Geräts nicht erhöht.

Wie starte ich einen Auftrag neu, bei dem ein Fehler aufgetreten ist?

Wenn ein Fehler auftritt, versucht der Amazon Kinesis Video Streams Edge Agent, den Auftrag neu zu starten. Bei einigen Fehlern (z. B. Konfigurationsfehlern) müssen Sie den Auftrag jedoch manuell neu starten.

Informationen dazu, welche Aufträge manuell neu gestartet werden müssen, finden Sie in der FatalError Metrik unter [the section called "Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch"](#).

Senden Sie die erneut, [the section called "StartEdgeConfigurationUpdate"](#) um den Auftrag für den Stream neu zu starten.

Wie überwache ich den Zustand meines Amazon Kinesis Video Streams Edge Agent?

Weitere Informationen finden Sie unter [the section called "Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch"](#).

Video über eine VPC streamen

Diese Betaversion ist in der Region Europa (Paris), eu-west-3, als Vorschauversion verfügbar. Wenn Sie auf diese Komponenten und unseren Leitfaden für die ersten Schritte zugreifen möchten, [senden Sie uns eine E-Mail](#).

Mit dem VPC-Endpunktservice Amazon Kinesis Video Streams können Sie Videos über das Amazon-Netzwerk streamen und konsumieren, ohne dass Daten über das öffentliche Internet übertragen werden.

Um Zugriff zu beantragen, [senden Sie uns die folgenden Informationen per E-Mail](#):

- Konto-ID
- ARNs streamen
- VPC-ID

Note

Es kann bis zu einer Woche dauern, bis wir Sie zum Service hinzufügen.

Wenn Sie in der Vergangenheit noch nicht mit VPC-Endpunkten gearbeitet haben, lesen Sie die folgenden Informationen, um sich mit dem Konzept vertraut zu machen:

- [AWS PrivateLinkHintergrund](#)
- [Leitfaden für die ersten Schritte mit VPC](#)

Zusätzliche Informationen

Sobald Sie zur Betaversion hinzugefügt wurden, senden wir Ihnen per E-Mail einen Link zu weiteren Informationen zu dieser Funktion.

VPC-Endpunktverfahren

Kontingente

Die wichtigsten Unterschiede bei den Quoten sind:

- Niedrigeres Kontingent für alle Bandbreiten-APIs (2 Mbit/s):
 - PutMedia
 - GetMedia
 - GetMediaForFragmentList
- Pro Kunde sind 10 Streams zulässig

Endpunkt herstellen

Sobald Sie die Zulassung in der Liste haben, erhalten Sie den VPC-Endpunkt-Servicenamen für Amazon Kinesis Video Streams. Es wird so aussehen. `com.amazonaws.region.kinesisvideo`

Erstellen Sie einen [VPC-Schnittstellen-Endpunkt](#) für Amazon Kinesis Video Streams entweder mit der Amazon VPC-Konsole oder mit (). AWS Command Line Interface AWS CLI

Geben Sie in der AWS CLI Folgendes ein:

```
aws ec2 create-vpc-endpoint \  
--vpc-id customer-provided-vpc-id \  
--service-name com.amazonaws.eu-west-2.kinesisvideo \  
--private-dns-enabled
```

Important

Der Datenverkehr innerhalb Ihrer VPC verwendet privates DNS, um über den Endpunkt zu leiten. Wenn Sie dies nicht aktivieren, müssen Sie Ihre eigene DNS-Logik implementieren. Weitere Informationen zu privatem DNS finden Sie in der [AWS PrivateLinkDokumentation](#).

Weitere Informationen zu dieser AWS CLI Option finden Sie unter [create-vpc-endpoint](#).

Steuern Sie den Zugriff auf Endgeräte

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf Amazon Kinesis Video Streams steuert. Die Richtlinie gibt die folgenden Informationen an:

- der Principal, der Aktionen ausführen kann,
- die Aktionen, die ausgeführt werden können, und
- die Ressourcen, auf denen Aktionen ausgeführt werden können.

Weitere Informationen finden Sie im Handbuch unter [Steuern des Zugriffs auf Dienste mit VPC-Endpunkten mithilfe von Endpunktrichtlinien](#). [AWS PrivateLink](#)

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für Amazon Kinesis Video Streams. Wenn diese Richtlinie an einen Endpunkt angehängt ist, verweigert sie allen Principals auf allen Ressourcen den Zugriff auf die aufgelisteten PutMedia Aktionen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Deny",
      "Action": [
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Bilder in Kinesis-Videostreams

Sie können die APIs und SDKs von Amazon Kinesis Video Streams verwenden, um Bilder aus Ihren Videostreams zu extrahieren. Sie können diese Bilder für erweiterte Wiedergabeanwendungen wie Miniaturansichten oder erweitertes Scrubbing oder für die Verwendung in Pipelines für maschinelles Lernen verwenden. Kinesis Video Streams bietet eine On-Demand-Bildextraktion über eine API oder eine automatische Bildextraktion aus Metadaten-Tags in einem aufgenommenen Video.

Informationen zur Verwendung der von Kinesis Video Streams verwalteten Unterstützung für Bilder finden Sie unter:

- [Bildgenerierung auf Abruf \(GetImages\)](#)- Mit dieser API können Kunden ein einzelnes Bild oder mehrere Bilder aus Videos extrahieren, die in Kinesis Video Streams gespeichert sind.
- [Automatisierte Imagegenerierung \(S3-Bereitstellung\)](#) - Konfigurieren Sie Kinesis Video Streams so, dass anhand von Tags in einem hochgeladenen Video automatisch Bilder aus Videodaten in Echtzeit extrahiert und die Bilder an den vom Kunden angegebenen S3-Bucket übermittelt werden.

Themen

- [Erste Schritte mit GetImages](#)
- [Erste Schritte mit der Amazon S3-Lieferung](#)

Erste Schritte mit GetImages

Die verwaltete Unterstützung für Bilder bietet eine vollständig verwaltete Möglichkeit, Bilder aus den Videodaten abzurufen, die in Kinesis Video Streams gestreamt und gespeichert werden. Sie können Bilder verwenden, um Workloads für maschinelles Lernen (ML) wie die Erkennung von Personen, Haustieren oder Fahrzeugen auszuführen. Bilder können auch verwendet werden, um interaktive Elemente zur Wiedergabe hinzuzufügen, z. B. Bildvorschauen für Bewegungsereignisse und Scrubbing für einen Videoclip.

Für weitere Informationen über die `GetImages` Funktion, siehe [GetImages](#) in der Amazon Kinesis Video Streams Archivierte MedienAPI-Referenzleitfaden.

Erste Schritte mit der Amazon S3-Lieferung

Derzeit betreiben und verwalten Kunden ihre eigene Bildtranskodierungspipeline, um Bilder für verschiedene Zwecke wie Scrubbing, Bildvorschau, Ausführen von ML-Modellen auf Bildern und mehr zu erstellen. Kinesis Video Streams bietet die Möglichkeit, die Bilder zu transkodieren und zu übertragen. Kinesis Video Streams extrahiert anhand eines Tags automatisch Bilder aus Videodaten in Echtzeit und übermittelt die Bilder an einen vom Kunden angegebenen S3-Bucket.

UpdateImageGenerationConfiguration

So richten Sie einen Kinesis-Videostream ein, um die Bildgenerierung in Amazon S3 zu ermöglichen:

1. Erstelle eine S3-Eimer für die Bildgenerierung auf der Grundlage der im SDK mithilfe der neuen API hinzugefügten Tags. Beachten Sie die S3-URI, was im nächsten Schritt erforderlich ist, wenn die Konfigurationen zur Image-Generierung für die Streams aktualisiert werden.
2. Erstellen Sie eine JSON-Datei mit dem Namen `update-image-generation-input.json` mit dem folgenden Inhalt als Eingabe.

```
{
  "StreamName": "TestStream",
  "ImageGenerationConfiguration":
  {
    "Status": "ENABLED",
    "DestinationConfig":
    {
      "DestinationRegion": "us-east-1",
      "Uri": "s3://bucket-name"
    },
    "SamplingInterval": 200,
    "ImageSelectorType": "PRODUCER_TIMESTAMP",
    "Format": "JPEG",
    "FormatConfig": {
      "JPEGQuality": "80"
    },
    "WidthPixels": 320,
    "HeightPixels": 240
  }
}
```


Sie können das verwenden AWS CLI um die `UpdateImageGenerationConfiguration` API-Vorgang, um den zuvor erstellten Amazon S3-ARN hinzuzufügen und den Status zu ändern `ENABLED`.

```
aws kinesishvideo update-image-generation-configuration \
--cli-input-json file://./update-image-generation-input.json \
```

Anfrage:

UpdateImageGenerationConfiguration HTTP/1.1

```
Method: 'POST'
Path: '/updateImageGenerationConfiguration'
Body: {
  StreamName: 'String', // Optional. Either stream name or arn should be passed
  StreamArn: 'String', // Optional. Either stream name or arn should be passed
  ImageGenerationConfiguration : {
    // required
    Status: 'Enum', // ENABLED | DISABLED,
    ImageSelectorType: 'Enum', // SERVER_TIMESTAMP | PRODUCER_TIMESTAMP..
    DestinationConfig: {
      DestinationRegion: 'String',
      Uri: string,
    },
    SamplingInterval: 'Number'//
    Format: 'Enum', // JPEG | PNG
    // Optional parameters
    FormatConfig: {
      'String': 'String',
    },
    WidthPixels: 'Number', // 1 - 3840 (4k).
    HeightPixels: 'Number' // 1 - 2160 (4k).
  }
}
```

Antwort:

```
HTTP/1.1 200
Content-type: application/json
Body: {
```

```
}
```

Note

Nach der Aktualisierung der Konfiguration zur Imagegenerierung dauert es mindestens 1 Minute, bis der Workflow zur Imagegenerierung eingeleitet wird. Warten Sie mindestens 1 Minute, bevor Sie es aufrufen `PutMedia` nach dem Update-Anruf.

DescribeImageGenerationConfiguration

Um sich Konfigurationen zur Imagegenerierung anzusehen, die bereits für einen Stream festgelegt sind, können Kunden eine `DescribeImageGenerationConfiguration`Anfrage, wie folgt.

Anfrage:

```
DescribeImageGenerationConfiguration HTTP/1.1
```

```
Method: 'POST'
Path: '/describeImageGenerationConfiguration'
Body: {
  StreamName: 'String', // Optional. Either stream name or arn should be passed
  StreamArn: 'String', // Optional. Either stream name or arn should be passed
}
```

Antwort:

```
HTTP/1.1 200
Content-type: application/json
Body: {
  ImageGenerationConfiguration : {
    Status: 'Enum',
    ImageSelectorType: 'Enum', // SERVER_TIMESTAMP | PRODUCER_TIMESTAMP
    DestinationConfig: {
      DestinationRegion: 'String'
    }
  }
}
```

```

        Uri: 'string',
    },
    SamplingInterval: 'Number',
    Format: 'Enum',
    FormatConfig: {
        'String': 'String',
    },
    WidthPixels: 'Number',
    HeightPixels: 'Number'
}
}

```

Um mehr über die zu erfahren `DescribeImageGenerationConfiguration` Funktion, siehe [DescribeImageGenerationConfiguration](#) in der Amazon Kinesis Video Streams — Entwicklerleitfaden.

MKV-Tags für Produzenten

Sie können das Kinesis Video Streams Producer SDK verwenden, um bestimmte Fragmente von Interesse zu taggen, indem Sie eine API-Operation im SDK verfügbar machen. Ein Beispiel für ein Tag finden Sie unter [dieser Code](#). Beim Aufrufen dieser API fügt das SDK zusammen mit den Fragmentdaten eine Reihe vordefinierter MKV-Tags hinzu. Kinesis Video Streams erkennt diese speziellen MKV-Tags und initiiert den Workflow zur Bildgenerierung auf der Grundlage der Bildverarbeitungskonfiguration dieses Streams.

Alle Fragmentmetadaten, die zusammen mit den Amazon S3-Bildgenerierungs-Tags bereitgestellt werden, werden als Amazon S3-Metadaten gespeichert.

Syntax für Producer-MKV-Tags

```

|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger image generation for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION

| // OPTIONAL: S3 prefix which will be set as prefix for generated image.
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_PREFIX

```

```

| + String: image_prefix_in_s3 // 256 bytes max m

| // OPTIONAL: Key value pairs that will be persisted as S3 Image object metadata.
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String:CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes

```

Hinzufügen von Metadaten-Tags im Producer SDK mithilfe vonPutEventMetaData

Die Funktion `PutEventMetadata` hängt eine MKV-Datei an, die mit einem Ereignis verknüpft ist. `PutEventMetadata` nimmt zwei Parameter an. Der erste Parameter ist ein Ereignis, dessen Wert von `STREAM_EVENT_TYPE` aufzählung. Der zweite Parameter, [pStreamEventMetadata](#), ist optional und kann verwendet werden, um zusätzliche Metadaten als Schlüssel-Wert-Paare einzuschließen. Es gibt ein Limit von fünf Schlüssel-Wert-Metadatenpaaren, die hinzugefügt werden können.

Einschränkungen

In der folgenden Tabelle sind die Einschränkungen aufgeführt, die mit den Metadaten-Tags verbunden sind. Wenn das Limit für Metadaten-Tags anpassbar ist, können Sie über Ihren Account Manager eine Erhöhung beantragen.

Limit	Maximaler Wert	Anpassbar
Länge des Bildpräfixes	256	Nein
Optionale Schlüssellänge für Metadaten	128	Nein
Länge des optionalen Metadatenwerts	256	Nein
Maximale Anzahl optionaler Metadaten	10	Ja

S3-Objektmetadaten

Standardmäßig stellt Kinesis Video Streams die Fragmentnummer, Produzent, und Serverzeitstempel des Bildes, das als Amazon S3-Objektmetadaten generiert wurde. Wenn zusätzliche Fragmentdaten in den MKV-Tags angegeben werden, werden diese Tags auch zu den Amazon S3-Objektmetadaten hinzugefügt. Das folgende Beispiel zeigt die korrekte Syntax für die Amazon S3-Objektmetadaten.

```
{
  // KVS S3 object metadata
  x-amz-meta-aws_kinesisvideo_fragment_number : 'string',
  x-amz-meta-aws_kinesisvideo_producer_timestamp: 'number',
  x-amz-meta-aws_kinesisvideo_server_timestamp: 'number',

  // Optional key value pair sent as part of the MKV tags
  custom_key_1: custom_value_1,
  custom_key_2: custom_value_2,
}
```

S3-Objektpfad (Bild)

Die folgende Liste zeigt das richtige Format des Objektpfads und beschreibt jedes Element innerhalb des Pfads.

Format:

ImagePrefix_Konto-ID_StreamName_ImageTimecode_Zufällige ID.datei-Erweiterung

1. ImagePrefix - Wert von `AWS_KINESISVIDEO_IMAGE_PREFIX`.
2. AccountID - Konto-ID, unter der der Stream erstellt wird.
3. StreamName - Name des Streams, für den das Bild generiert wird.
4. ImageTimecode - Epochen-Timecode in dem Fragment, in dem das Bild generiert wird.
5. RandomID - Zufällige GUID.
6. file-extension - JPG oder PNG basierend auf dem angeforderten Bildformat.

Amazon S3-URI-Empfehlungen zum Schutz vor Drosselung

Wenn Sie Tausende von Bildern in Amazon S3 schreiben, besteht die Gefahr einer Drosselung. Weitere Informationen finden Sie unter [S3-Präfix-Put-Anforderungslimits](#).

Ein Amazon S3-Präfix beginnt mit einem PUT-Limit von 3.500 PUT-Anforderungen pro Sekunde und wird im Laufe der Zeit für eindeutige Präfixe schrittweise erhöht. Vermeiden Sie die Verwendung von Datum und Uhrzeit als Amazon S3-Präfixe. Zeitcodierte Daten wirken sich jeweils auf ein Präfix aus und ändern sich auch regelmäßig, wodurch frühere Präfix-Skalierungen ungültig werden. Um eine schnellere und konsistente Amazon S3-Skalierung zu ermöglichen, empfehlen wir, der Amazon S3-Ziel-URI ein zufälliges Präfix hinzuzufügen, z. B. einen Hex-Code oder eine UUID. Beispielsweise teilen Hex-Code-Präfixe Ihre Anfragen auf natürliche Weise nach dem Zufallsprinzip auf 16 verschiedene Präfixe auf (ein Präfix für jedes eindeutige Hex-Zeichen), sodass nach der automatischen Skalierung von Amazon S3 56.000 PUT-Anfragen pro Sekunde möglich sind.

Benachrichtigungen in Kinesis Video Streams

Wenn ein Medienfragment zur Nutzung verfügbar ist, benachrichtigt Kinesis Video Streams die Kunden mithilfe von Amazon Simple Notification Service (Amazon SNS) -Benachrichtigungen. Im folgenden Thema wird erklärt, wie Sie mit Benachrichtigungen beginnen können.

UpdateNotificationConfiguration

Verwenden Sie diesen API-Vorgang, um die Benachrichtigungsinformationen für einen Stream zu aktualisieren. Weitere Informationen zu dieser `UpdateNotificationConfiguration` Funktion finden Sie [UpdateNotificationConfiguration](#) im Amazon Kinesis Video Streams Developer Guide.

Note

Nach der Aktualisierung der Benachrichtigungskonfiguration dauert es mindestens eine Minute, bis die Benachrichtigung initiiert wird. Warten Sie mindestens eine Minute, bevor Sie `PutMedia` nach dem Aktualisierungsaufruf aufrufen.

DescribeNotificationConfiguration

Verwenden Sie diese API, um eine Benachrichtigungskonfiguration zu beschreiben, die an einen Stream angehängt ist. Weitere Informationen zu dieser `DescribeNotificationConfiguration` Funktion finden Sie [DescribeNotificationConfiguration](#) im Amazon Kinesis Video Streams Developer Guide.

Hersteller MKV-Tags

Sie können das Kinesis Video Streams Producer SDK verwenden, um bestimmte Fragmente von Interesse zu taggen, indem Sie eine API-Operation im SDK verfügbar machen. [In diesem Codeabschnitt finden Sie ein Beispiel dafür, wie das](#) funktioniert. Beim Aufrufen dieser API fügt das SDK zusammen mit den Fragmentdaten eine Reihe vordefinierter MKV-Tags hinzu. Kinesis Video Streams erkennt diese speziellen MKV-Tags und leitet Benachrichtigungen für die markierten Fragmente ein.

Alle Fragment-Metadaten, die zusammen mit den Notification MKV-Tags bereitgestellt werden, werden als Teil der Amazon SNS SNS-Themen-Payload veröffentlicht.

Syntax für Producer-MKV-Tags

```

|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger the notification for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_NOTIFICATION
| + String
| // OPTIONAL: Key value pairs that will be sent as part of the Notification payload
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String:CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes

```

Beschränkungen für MKV-Tags

In der folgenden Tabelle sind die Einschränkungen aufgeführt, die mit den Metadaten-Tags verbunden sind. Wenn das Limit für Metadaten-Tags anpassbar ist, können Sie über Ihren Account Manager eine Erhöhung beantragen.

Limit	Maximaler Wert	Einstellbar
Optionale Länge des Metadatenschlüssels	128	Nein
Optionale Länge des Metadatenwerts	256	Nein
Maximale Anzahl optionaler Metadaten	10	Ja

Nutzlast für Amazon SNS-Themen

Jede Benachrichtigung, die durch den vorherigen Workflow initiiert wurde, liefert die Amazon SNS SNS-Themennutzlast, wie im folgenden Beispiel gezeigt. Dieses Beispiel ist eine Amazon SNS SNS-Nachricht, die nach dem Verzehr von Benachrichtigungsdaten aus einer Amazon Simple Queue Service (Amazon SQS) -Warteschlange auftritt.

```
{
  "Type" : "Notification",
  "MessageId" : Message ID,
  "TopicArn" : SNS ARN,
  "Subject" : "Kinesis Video Streams Notification",
  "Message" : "{\"StreamArn\":Stream Arn,\"FragmentNumber\":Fragment Number,
  \"FragmentStartProducerTimestamp\":FragmentStartProducerTimestamp,
    \"FragmentStartServerTimestamp\":FragmentStartServerTimestamp,
  \"NotificationType\": \"PERSISTED\", \"NotificationPayload\":{CUSTOM_KEY_1:
  CUSTOM_VALUE_1,
    CUSTOM_KEY_2:CUSTOM_VALUE_2}}",
  "Timestamp" : "2022-04-25T18:36:29.194Z",
  "SignatureVersion" : Signature Version,
  "Signature" : Signature,
  "SigningCertURL" : Signing Cert URL,
  "UnsubscribeURL" : Unsubscribe URL
}
```

```
Subject: "Kinesis Video Streams Notification"
Message:
{
  "StreamArn":Stream Arn,
  "FragmentNumber":Fragment Number,
  "FragmentStartProducerTimestamp":Fragment Start Producer Timestamp,
  "FragmentStartServerTimestamp":Fragment Start Server Timestamp,
  "NotificationType":"PERSISTED",
  "NotificationPayload":{
    CUSTOM_KEY_1:CUSTOM_VALUE_1,
    CUSTOM_KEY_2:CUSTOM_VALUE_2
  }
}
```

Ihre Amazon SNS SNS-Nachrichten anzeigen

Sie können Nachrichten nicht direkt aus einem Amazon SNS SNS-Thema lesen, da es dafür keine API gibt. Um die Nachrichten anzusehen, abonnieren Sie eine SQS-Warteschlange für das SNS-Thema oder wählen Sie ein anderes von [Amazon SNS](#) unterstütztes Ziel. Die effizienteste Option zum Anzeigen von Nachrichten ist jedoch die Verwendung von Amazon SQS.

So zeigen Sie Ihre Amazon SNS SNS-Nachrichten mit Amazon SQS an

1. Erstellen Sie eine [Amazon SQS SQS-Warteschlange](#).
2. Öffnen Sie von der AWS Management Console aus das Amazon SNS SNS-Thema, das als Ziel festgelegt wurde, unter `NotificationConfiguration`.
3. Wählen Sie `Create Subscription` und dann die Amazon SQS SQS-Warteschlange aus, die im ersten Schritt erstellt wurde.
4. Führen Sie eine `PutMedia` Sitzung mit aktivierter Benachrichtigungskonfiguration aus und fügen Sie den Fragmenten die MKV-Tags für Benachrichtigungen hinzu.
5. Wählen Sie in der Amazon SQS SQS-Konsole die Amazon SQS SQS-Warteschlange aus und wählen Sie dann `Nachrichten senden und empfangen` für die Amazon SQS SQS-Warteschlange aus.
6. Nach Nachrichten abfragen. Dieser Befehl sollte alle von der `PutMedia` Sitzung generierten Benachrichtigungen anzeigen. Informationen zu Umfragen finden Sie unter [Amazon SQS Short and Long Polling](#).

Sicherheit in Amazon Kinesis Video Streams

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitssensibelsten Unternehmen zugeschnitten sind.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Die Wirksamkeit unserer Sicherheitsfunktionen wird regelmäßig von externen Prüfern im Rahmen des [AWS -Compliance-Programms getestet und überprüft](#). Weitere Informationen zu den Compliance-Programmen, die für Kinesis Video Streams gelten, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen](#).
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Kinesis Video Streams anwenden können. In den folgenden Themen erfahren Sie, wie Sie Kinesis Video Streams konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Kinesis Video Streams Streams-Ressourcen überwachen und sichern können.

Themen

- [Datenschutz in Kinesis Video Streams](#)
- [Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von IAM](#)
- [Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von AWS IoT](#)
- [Überwachung von Amazon Kinesis Video Streams](#)
- [Konformitätsprüfung für Amazon Kinesis Video Streams](#)
- [Resilienz in Amazon Kinesis Video Streams](#)
- [Infrastruktursicherheit in Kinesis Video Streams](#)
- [Bewährte Sicherheitsmethoden für Kinesis Video Streams](#)

Datenschutz in Kinesis Video Streams

Sie können serverseitige Verschlüsselung (SSE) mit AWS Key Management Service (AWS KMS) -Schlüsseln verwenden, um strenge Datenverwaltungsanforderungen zu erfüllen, indem Sie Ihre ruhenden Daten in Amazon Kinesis Video Streams verschlüsseln.

Themen

- [Was ist serverseitige Verschlüsselung für Kinesis Video Streams?](#)
- [Überlegungen zu Kosten, Regionen und Leistung](#)
- [Wie fange ich mit der serverseitigen Verschlüsselung an?](#)
- [Benutzergenerierten KMS-Schlüssel erstellen und verwenden](#)
- [Berechtigungen zur Verwendung von benutzergenerierten KMS-Schlüsseln](#)

Was ist serverseitige Verschlüsselung für Kinesis Video Streams?

Die serverseitige Verschlüsselung ist eine Funktion in Kinesis Video Streams, die Daten automatisch verschlüsselt, bevor sie sich im Ruhezustand befinden, und zwar mithilfe einer AWS KMS, die Sie angeben. Daten werden verschlüsselt, bevor sie in die Kinesis Video Streams Streams-Stream-Speicherschicht geschrieben werden, und sie werden entschlüsselt, nachdem sie aus dem Speicher abgerufen werden. Daher werden Ihre Daten im Ruhezustand innerhalb des Kinesis Video Streams Streams-Dienstes immer verschlüsselt.

Dank serverseitiger Verschlüsselung müssen Ihre Kinesis-Videostream-Produzenten und -Nutzer keine KMS-Schlüssel oder kryptografische Operationen verwalten. Wenn die Datenspeicherung aktiviert ist, werden Ihre Daten beim Ein- und Austritt aus Kinesis Video Streams automatisch verschlüsselt, sodass Ihre Daten im Ruhezustand verschlüsselt sind. AWS KMS stellt alle Schlüssel bereit, die von der serverseitigen Verschlüsselungsfunktion verwendet werden. AWS KMS optimiert die Verwendung eines KMS-Schlüssels für Kinesis Video Streams, der von einem benutzerdefinierten AWS KMS Schlüssel verwaltet wird AWS, der in den Service importiert wurde. AWS KMS

Überlegungen zu Kosten, Regionen und Leistung

Wenn Sie serverseitige Verschlüsselung anwenden, fallen Kosten für AWS KMS API-Nutzung und Schlüssel an. Im Gegensatz zu benutzerdefinierten AWS KMS Schlüsseln wird der (Default) `aws/kinesis-video` KMS-Schlüssel kostenlos angeboten. Sie müssen jedoch weiterhin die API-Nutzungskosten bezahlen, die Kinesis Video Streams in Ihrem Namen entstehen.

API-Nutzungskosten fallen für jeden KMS-Schlüssel an, auch für benutzerdefinierte. Die AWS KMS Kosten hängen von der Anzahl der Benutzeranmeldedaten ab, die Sie bei Ihren Datenproduzenten und -verbrauchern verwenden, da für jeden Benutzeranmeldenachweis ein eindeutiger API-Aufruf erforderlich ist. AWS KMS

Im Folgenden werden die Kosten pro Ressource aufgeführt:

Schlüssel

- Der KMS-Schlüssel für Kinesis Video Streams, der von AWS (alias =aws/kinesis-video) verwaltet wird, ist kostenlos.
- Benutzergenerierte KMS-Schlüssel sind kostenpflichtig AWS KMS key . Weitere Informationen finden Sie unter [AWS Key Management Service -Preisgestaltung](#).

AWS KMS API-Nutzung

API-Anfragen zur Generierung neuer Datenverschlüsselungsschlüssel oder zum Abrufen vorhandener Verschlüsselungsschlüssel nehmen mit zunehmendem Datenverkehr zu und sind mit AWS KMS Nutzungskosten verbunden. Weitere Informationen finden Sie unter [AWS Key Management Service Preise: Nutzung](#).

Kinesis Video Streams generiert wichtige Anfragen, auch wenn die Aufbewahrung auf 0 gesetzt ist (keine Aufbewahrung).

Verfügbarkeit serverseitiger Verschlüsselung nach Regionen

Die serverseitige Verschlüsselung von Kinesis Video Streams ist überall verfügbar, AWS-Regionen wo Kinesis Video Streams verfügbar ist.

Wie fange ich mit der serverseitigen Verschlüsselung an?

Die serverseitige Verschlüsselung ist bei Kinesis Video Streams immer aktiviert. Wenn bei der Erstellung des Streams kein vom Benutzer bereitgestellter Schlüssel angegeben wird, wird der Standardschlüssel (bereitgestellt von Kinesis Video Streams) verwendet.

Ein vom Benutzer bereitgestellter KMS-Schlüssel muss einem Kinesis-Videostream zugewiesen werden, wenn er erstellt wird. Sie können einem Stream später mithilfe der [UpdateStream](#)API keinen anderen Schlüssel zuweisen.

Sie können einem Kinesis-Videostream auf zwei Arten einen vom Benutzer bereitgestellten KMS-Schlüssel zuweisen:

- Geben Sie beim Erstellen eines Kinesis-Videostreams in den AWS Management Console den den KMS-Schlüssel auf der Registerkarte Verschlüsselung auf der Seite Neuen Videostream erstellen an.
- Geben Sie beim Erstellen eines Kinesis-Videostreams mithilfe der [CreateStream](#) API die Schlüssel-ID im `KmsKeyId` Parameter an.

Benutzergenerierten KMS-Schlüssel erstellen und verwenden

In diesem Abschnitt wird beschrieben, wie Sie Ihre eigenen KMS-Schlüssel erstellen und verwenden können, anstatt den von Amazon Kinesis Video Streams verwalteten Schlüssel zu verwenden.

Benutzergenerierte KMS-Schlüssel erstellen

Informationen zum Erstellen eigener Schlüssel finden Sie unter [Creating Keys](#) im AWS Key Management Service Developer Guide. Nachdem Sie Schlüssel für Ihr Konto erstellt haben, gibt der Kinesis Video Streams Streams-Dienst diese Schlüssel in der KMS-Masterschlüsselliste zurück.

Verwenden von benutzergenerierten KMS-Schlüsseln

Nachdem Ihren Verbrauchern, Produzenten und Administratoren die richtigen Berechtigungen zugewiesen wurden, können Sie benutzerdefinierte KMS-Schlüssel in Ihrem eigenen AWS-Konto oder einem anderen AWS-Konto System verwenden. Alle KMS-Schlüssel in Ihrem Konto werden in der KMS-Hauptschlüsselliste auf der Konsole angezeigt.

Um benutzerdefinierte KMS-Schlüssel verwenden zu können, die sich in einem anderen Konto befinden, müssen Sie über die entsprechenden Berechtigungen verfügen. Sie müssen außerdem den Stream mithilfe der `CreateStream`-API erstellen. Sie können keine KMS-Schlüssel von verschiedenen Konten in Streams verwenden, die in der Konsole erstellt wurden.

Note

Auf den KMS-Schlüssel wird erst zugegriffen, wenn der `GetMedia` Vorgang `PutMedia` oder ausgeführt wurde. Dies führt zu folgendem Ergebnis:

- Wenn der von Ihnen angegebene Schlüssel nicht existiert, ist der `CreateStream` Vorgang erfolgreich, aber `PutMedia` die `GetMedia` Operationen im Stream schlagen fehl.

- Wenn Sie den bereitgestellten Schlüssel (`aws/kinesis-video`) verwenden, ist der Schlüssel erst in Ihrem Konto vorhanden, wenn der erste PutMedia GetMedia OR-Vorgang ausgeführt wird.

Berechtigungen zur Verwendung von benutzergenerierten KMS-Schlüsseln

Bevor Sie die serverseitige Verschlüsselung mit einem benutzergenerierten KMS-Schlüssel verwenden können, müssen Sie KMS-Schlüsselrichtlinien konfigurieren, um die Verschlüsselung von Streams sowie die Verschlüsselung und Entschlüsselung von Stream-Datensätzen zu ermöglichen. Beispiele und weitere Informationen zu AWS KMS Berechtigungen finden Sie unter [AWS KMS API-Berechtigungen: Referenz zu Aktionen und Ressourcen](#).

Note

Die Verwendung des Standard-Serviceschlüssels für die Verschlüsselung erfordert keine Anwendung benutzerdefinierter IAM-Berechtigungen.

Bevor Sie benutzergenerierte KMS-Schlüssel verwenden, stellen Sie sicher, dass Ihre Kinesis-Videostream-Produzenten und -Verbraucher (IAM-Prinzipale) Benutzer der AWS KMS Master-Key-Richtlinie sind. Andernfalls schlägt das Schreiben in und das Lesen aus dem Stream fehl. Dies kann zu einem Datenverlust, einer verspäteten Verarbeitung oder abgestürzten Anwendungen führen. Sie können Berechtigungen für KMS-Schlüssel mit IAM-Richtlinien verwalten. Weitere Informationen finden Sie unter [Verwenden von IAM-Richtlinien mit AWS KMS](#).

Beispiel für Produzentenberechtigungen

Ihre Kinesis-Videostream-Produzenten müssen über die folgenden `kms:GenerateDataKey` Berechtigungen verfügen:

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

```

    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis-video:PutMedia",
    ],
    "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
  }
]
}

```

Beispiel für Benutzerberechtigungen

Ihre Kinesis-Videostream-Nutzer müssen über die folgenden `kms:Decrypt` Berechtigungen verfügen:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesis-video:GetMedia",
      ],
      "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
    }
  ]
}

```


Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von IAM

Sie können AWS Identity and Access Management (IAM) mit Amazon Kinesis Video Streams verwenden, um zu kontrollieren, ob Benutzer in Ihrer Organisation eine Aufgabe mithilfe bestimmter Kinesis Video Streams-API-Operationen ausführen können und ob sie bestimmte Ressourcen verwenden können. AWS

Weitere Informationen zu IAM finden Sie unter:

- [AWS Identity and Access Management \(IAM\)](#)
- [Erste Schritte](#)
- [IAM Benutzerhandbuch](#)

Inhalt

- [Richtliniensyntax](#)
- [Aktionen für Kinesis Video Streams](#)
- [Amazon-Ressourcennamen \(ARNs\) für Kinesis Video Streams](#)
- [Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren](#)
- [Beispielrichtlinien für Kinesis Video Streams](#)

Richtliniensyntax

Eine IAM-Richtlinie ist ein JSON-Dokument, das eine oder mehrere Anweisungen enthält. Jede Anweisung ist folgendermaßen strukturiert:

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  ]
}
```

```
]
}
```

Eine Anweisung kann aus verschiedenen Elementen bestehen:

- **Effekt** — Der Effekt kann `Allow` oder `Deny` sein. -Benutzer verfügen standardmäßig nicht über die Berechtigung zur Verwendung von Ressourcen und API-Aktionen. Daher werden alle Anfragen abgelehnt. Dieser Standardwert kann durch eine explizite Zugriffserlaubnis überschrieben werden. Eine explizite Zugriffsverweigerung überschreibt jedwede Zugriffserlaubnis.
- **Aktion** — Die Aktion ist die spezifische API-Aktion, für die Sie die Erlaubnis erteilen oder verweigern.
- **Ressource** — Die Ressource, die von der Aktion betroffen ist. Um eine Ressource in der Anweisung anzugeben, müssen Sie deren Amazon-Ressourcennamen (ARN) verwenden.
- **Bedingung** — Bedingungen sind optional. Mit ihrer Hilfe können Sie bestimmen, wann Ihre Richtlinie wirksam ist.

Wir empfehlen Ihnen, bei der Erstellung und Verwaltung von IAM-Richtlinien den [IAM-Richtliniengenerator und den IAM-Richtliniensimulator](#) zu verwenden.

Aktionen für Kinesis Video Streams

In einer IAM-Richtlinienanweisung können Sie jede API-Aktion von jedem Service, der IAM unterstützt, angeben. Verwenden Sie für Kinesis Video Streams das folgende Präfix mit dem Namen der API-Aktion: `kinesisvideo:`. Beispiel: `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams` und `kinesisvideo:DescribeStream`.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": ["kinesisvideo:action1", "kinesisvideo:action2"]
```

Sie können auch mehrere Aktionen mittels Platzhaltern angeben. Beispielsweise können Sie alle Aktionen festlegen, deren Name mit dem Wort "Get" beginnt:

```
"Action": "kinesisvideo:Get*"
```

Um alle Kinesis Video Streams Streams-Operationen anzugeben, verwenden Sie das Sternchen (*) als Platzhalter wie folgt:

```
"Action": "kinesisvideo:*"
```

Eine vollständige Liste der Kinesis Video Streams Streams-API-Aktionen finden Sie in der [Kinesis Video Streams Streams-API-Referenz](#).

Amazon-Ressourcennamen (ARNs) für Kinesis Video Streams

Jede IAM-Richtlinienanweisung gilt für die Ressourcen, die Sie mithilfe ihrer ARNs angegeben haben.

Verwenden Sie das folgende ARN-Ressourcenformat für Kinesis Video Streams:

```
arn:aws:kinesisvideo:region:account-id:stream/stream-name/code
```

Beispielsweise:

```
"Resource": arn:aws:kinesisvideo:*:111122223333:stream/my-stream/0123456789012
```

Sie können den ARN eines Streams mit abrufen [DescribeStream](#).

Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren

Möglicherweise müssen Sie anderen IAM-Konten die Erlaubnis erteilen, Operationen mit Kinesis-Videostreams durchzuführen. Die folgende Übersicht beschreibt die allgemeinen Schritte, um den kontenübergreifenden Zugriff auf Videostreams zu gewähren:

1. Ermitteln Sie die 12-stellige Konto-ID des Kontos, dem Sie Berechtigungen zum Durchführen von Operationen mit dem Stream gewähren möchten (z. B. 111111111111).
2. Erstellen Sie für das Konto, dem der Stream gehört, eine verwaltete Richtlinie, die die gewünschte Zugriffsstufe gewährt. [Beispielrichtlinien](#) Im nächsten Abschnitt finden Sie beispielsweise Richtlinien für Kinesis Video Streams Streams-Ressourcen.
3. Erstellen Sie eine Rolle und geben Sie das Konto an, für das Sie Berechtigungen erteilen möchten. Fügen Sie dann die Richtlinie hinzu, die Sie im vorherigen Schritt erstellt haben.
4. Erstellen Sie eine verwaltete Richtlinie, die die AssumeRole Aktion für die Rolle ermöglicht, die Sie im vorherigen Schritt erstellt haben. Die Rolle kann beispielsweise wie folgt aussehen:

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

```
"Action": "sts:AssumeRole",
"Resource": "arn:aws:iam::123456789012:role/CustomRole"
}
}
```

step-by-step Anweisungen zur Gewährung von kontenübergreifendem Zugriff finden Sie unter [Delegieren des Zugriffs für alle Benutzer AWS-Konten mithilfe von IAM-Rollen](#).

Beispielrichtlinien für Kinesis Video Streams

Die folgenden Beispielrichtlinien zeigen, wie Sie den Benutzerzugriff auf Ihre Kinesis Video Streams kontrollieren können.

Example 1: Erlauben Sie Benutzern, Daten aus jedem Kinesis-Videostream abzurufen

Diese Richtlinie ermöglicht es einem Benutzer oder einer Gruppe, die `DescribeStream`, `GetDataEndpoint`, `GetMediaListStreams`, und `ListTagsForStream` -Operationen für jeden Kinesis-Videostream auszuführen. Diese Richtlinie ist für Benutzer geeignet, die Daten aus beliebigen Videostreams abrufen können.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:Describe*",
        "kinesisvideo:Get*",
        "kinesisvideo:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 2: Erlauben Sie einem Benutzer, einen Kinesis-Videostream zu erstellen und Daten darauf zu schreiben

Diese Richtlinie erlaubt es einem Benutzer oder einer Gruppe, die Operationen `CreateStream` und `PutMedia` auszuführen. Diese Richtlinie ist für eine Überwachungskamera geeignet, die einen Videostream erstellen und Daten an diesen Stream senden kann.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:CreateStream",
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 3: Erlauben Sie einem Benutzer vollen Zugriff auf alle Kinesis Video Streams Streams-Ressourcen

Diese Richtlinie ermöglicht es einem Benutzer oder einer Gruppe, jeden Kinesis Video Streams Streams-Vorgang auf einer beliebigen Ressource durchzuführen. Diese Richtlinie ist für Administratoren geeignet.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:*",
      "Resource": "*"
    }
  ]
}
```

Example 4: Erlauben Sie einem Benutzer, Daten in einen bestimmten Kinesis-Videostream zu schreiben

Diese Richtlinie erlaubt es einem Benutzer oder einer Gruppe, Daten in einen bestimmten Videostream zu schreiben. Diese Richtlinie ist für ein Gerät geeignet, das Daten an einen einzelnen Stream senden kann.

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

```
{
  "Effect": "Allow",
  "Action": "kinesisvideo:PutMedia",
  "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
your_stream/0123456789012"
}
```

Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von AWS IoT

In diesem Abschnitt wird beschrieben, wie Sie einem Gerät (z. B. einer Kamera) ermöglichen, Audio- und Videodaten nur an einen bestimmten Kinesis-Videostream zu senden. Sie können dies tun, indem Sie den Anbieter für AWS IoT Anmeldeinformationen und eine AWS Identity and Access Management (IAM-) Rolle verwenden.

Geräte können X.509-Zertifikate verwenden, um AWS IoT mithilfe von Protokollen für die gegenseitige TLS-Authentifizierung eine Verbindung herzustellen. Andere AWS-Services (z. B. Kinesis Video Streams) unterstützen keine zertifikatsbasierte Authentifizierung, können jedoch mithilfe von AWS Anmeldeinformationen im AWS Signature Version 4-Format aufgerufen werden. Der Signature Version 4-Algorithmus erfordert in der Regel, dass der Anrufer über eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel verfügt. AWS IoT verfügt über einen Anbieter für Anmeldeinformationen, mit dem Sie das integrierte X.509-Zertifikat als eindeutige Geräteidentität für die Authentifizierung von AWS Anfragen (z. B. Anfragen an Kinesis Video Streams) verwenden können. Dadurch müssen Sie keine Zugriffsschlüssel-ID und keinen geheimen Zugriffsschlüssel auf Ihrem Gerät speichern.

Der Anbieter für Anmeldeinformationen authentifiziert einen Client (in diesem Fall ein Kinesis Video Streams Streams-SDK, das auf der Kamera läuft, die Sie Daten an einen Videostream senden möchten) mithilfe eines X.509-Zertifikats und stellt ein temporäres Sicherheitstoken mit eingeschränkten Rechten aus. Sie können das Token verwenden, um jede AWS Anfrage zu signieren und zu authentifizieren (in diesem Fall ein Aufruf der Kinesis Video Streams). Weitere Informationen finden Sie unter [Autorisieren von direkten Aufrufen](#) von Diensten. AWS

Für diese Art der Authentifizierung der Anfragen Ihrer Kamera an Kinesis Video Streams müssen Sie eine IAM-Rolle erstellen und konfigurieren und der Rolle entsprechende IAM-Richtlinien zuordnen, damit der Anbieter der AWS IoT Anmeldeinformationen die Rolle in Ihrem Namen übernehmen kann.

Weitere Informationen AWS IoT dazu finden Sie in der [Dokumentation.AWS IoT Core](#) Weitere Informationen über IAM finden Sie unter [AWS Identity and Access Management \(IAM\)](#).

Themen

- [AWS IoT ThingName als Streamname](#)
- [AWS IoT Certificateld als Streamname](#)
- [Verwenden Sie AWS IoT Anmeldeinformationen, um zu einem fest codierten Stream-Namen zu streamen](#)

AWS IoT ThingName als Streamname

Themen

- [Schritt 1: Erstelle einen AWS IoT Ding-Typ und ein AWS IoT Ding](#)
- [Schritt 2: Erstellen Sie eine IAM-Rolle, die übernommen werden soll von AWS IoT](#)
- [Schritt 3: Erstellen und konfigurieren Sie das X.509-Zertifikat](#)
- [Schritt 4: Testen Sie die AWS IoT Anmeldeinformationen mit Ihrem Kinesis-Videostream](#)
- [Schritt 5: Bereitstellen von AWS IoT Zertifikaten und Anmeldeinformationen im Dateisystem Ihrer Kamera und Streamen von Daten in Ihren Videostream](#)

Schritt 1: Erstelle einen AWS IoT Ding-Typ und ein AWS IoT Ding

AWS IoT In ist ein Ding eine Repräsentation eines bestimmten Geräts oder einer logischen Entität. In diesem Fall steht ein AWS IoT Ding für Ihren Kinesis-Videostream, für den Sie die Zugriffskontrolle auf Ressourcenebene konfigurieren möchten. Um ein Ding zu erstellen, müssen Sie zunächst einen AWS IoT Dingtyp erstellen. Sie können Dingtypen verwenden AWS IoT , um Beschreibungs- und Konfigurationsinformationen zu speichern, die allen Dingen gemeinsam sind, die demselben Dingtyp zugeordnet sind.

1. Der folgende Beispielbefehl erstellt einen Objekttyp `kvs_example_camera`:

```
aws --profile default iot create-thing-type --thing-type-name kvs_example_camera >
iot-thing-type.json
```

2. Dieser Beispielbefehl erstellt das `kvs_example_camera_stream` Ding vom Typ `kvs_example_camera` Ding:

```
aws --profile default iot create-thing --thing-name kvs_example_camera_stream --
thing-type-name kvs_example_camera > iot-thing.json
```

Schritt 2: Erstellen Sie eine IAM-Rolle, die übernommen werden soll von AWS IoT

IAM-Rollen ähneln Benutzern, da es sich bei einer Rolle um eine AWS Identität mit Berechtigungsrichtlinien handelt, die festlegen, wofür die Identität zuständig ist und was nicht. AWS Eine Rolle kann von jedem übernommen werden, der sie benötigt. Wenn Sie eine Rolle annehmen, erhalten Sie temporäre Sicherheitsanmeldeinformationen für Ihre Rollensitzung.

Die Rolle, die Sie in diesem Schritt erstellen, kann verwendet werden, AWS IoT um temporäre Anmeldeinformationen vom Security Token Service (STS) abzurufen, wenn Autorisierungsanfragen von einem Client ausgeführt werden. In diesem Fall ist der Client das Kinesis Video Streams SDK, das auf Ihrer Kamera läuft.

Führen Sie die folgenden Schritte aus, um diese IAM-Rolle zu erstellen und zu konfigurieren:

1. Erstellen Sie eine IAM-Rolle.

Der folgende Beispielbefehl erstellt eine IAM-Rolle mit dem Namen `KVSCameraCertificateBasedIAMRole`:

```
aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
--assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
```

Sie können die folgende JSON-Datei für die Vertrauensrichtlinie `iam-policy-document.json` verwenden:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "credentials.iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



```
}

```

2. Fügen Sie als Nächstes der IAM-Rolle, die Sie zuvor erstellt haben, eine Berechtigungsrichtlinie hinzu. Diese Berechtigungsrichtlinie ermöglicht die selektive Zugriffskontrolle (eine Teilmenge der unterstützten Operationen) für eine AWS Ressource. In diesem Fall ist die AWS Ressource der Videostream, den Ihre Kamera Daten senden soll. Mit anderen Worten, sobald alle Konfigurationsschritte abgeschlossen sind, kann diese Kamera Daten nur an diesen Videostream senden.

```
aws --profile default iam put-role-policy --role-name
  KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
  document 'file://iam-permission-document.json'
```

Sie können den folgenden IAM-Richtlinien-JSON für die Datei `iam-permission-document.json` verwenden:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
        iot:ThingName}/*"
    }
  ]
}
```

Beachten Sie, dass diese Richtlinie die angegebenen Aktionen nur für einen Videostream (AWS Ressource) autorisiert, der durch den Platzhalter (`${credentials iot:}`) angegeben ist. `ThingName` Dieser Platzhalter nimmt den Wert des AWS IoT Thing-Attributs an, `ThingName` wenn der Anbieter der AWS IoT Anmeldeinformationen den Namen des Videostreams in der Anfrage sendet.

3. Erstellen Sie als Nächstes einen Rollenalias für Ihre IAM-Rolle. Rollenalias ist ein alternatives Datenmodell, das auf die IAM-Rolle verweist. Eine Anfrage eines AWS IoT

Anmeldeinformationsanbieters muss einen Rollenalias enthalten, der angibt, welche IAM-Rolle übernommen werden soll, um die temporären Anmeldeinformationen vom STS zu erhalten.

Mit dem folgenden Beispielbefehl wird ein Rollenalias namens `KvsCameraIoTRoleAlias` erstellt:

```
aws --profile default iot create-role-alias --role-alias KvsCameraIoTRoleAlias --
role-arn $(jq --raw-output '.Role.Arn' iam-role.json) --credential-duration-seconds
3600 > iot-role-alias.json
```

4. Jetzt können Sie die Richtlinie erstellen, die es ermöglicht, mithilfe des Rollenalias die Rolle mit dem Zertifikat AWS IoT zu übernehmen (sobald es angehängt ist).

Mit dem folgenden Beispielbefehl wird eine Richtlinie für AWS IoT aufgerufen `KvsCameraIoTPolicy`.

```
aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-
document 'file://iot-policy-document.json'
```

Sie können den folgenden Befehl verwenden, um das `iot-policy-document` JSON-Dokument „.json“ zu erstellen:

```
cat > iot-policy-document.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "$(jq --raw-output '.roleAliasArn' iot-role-alias.json)"
    }
  ]
}
EOF
```

Schritt 3: Erstellen und konfigurieren Sie das X.509-Zertifikat

Die Kommunikation zwischen einem Gerät (Ihrem Videostream) und AWS IoT wird durch die Verwendung von X.509-Zertifikaten geschützt.

1. Erstellen Sie das Zertifikat, an das Sie die zuvor erstellte Richtlinie anhängen müssen. AWS IoT

```
aws --profile default iot create-keys-and-certificate --set-as-active --
certificate-pem-outfile certificate.pem --public-key-outfile public.pem.key --
private-key-outfile private.pem.key > certificate
```

2. Hängen Sie die Richtlinie für AWS IoT (zuvor KvsCameraIoTPolicy erstellt) an dieses Zertifikat an.

```
aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
$(jq --raw-output '.certificateArn' certificate)
```

3. Hängen Sie Ihr AWS IoT Ding (kvs_example_camera_stream) an das Zertifikat an, das Sie gerade erstellt haben:

```
aws --profile default iot attach-thing-principal --thing-name
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
certificate)
```

4. Um Anfragen über den Anbieter für AWS IoT Anmeldeinformationen zu autorisieren, benötigen Sie den Endpunkt für AWS IoT Anmeldeinformationen, der für Ihre AWS-Konto ID eindeutig ist. Sie können den folgenden Befehl verwenden, um den Endpunkt der AWS IoT Anmeldeinformationen abzurufen.

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
--output text > iot-credential-provider.txt
```

5. Zusätzlich zu dem zuvor erstellten X.509-Zertifikat benötigen Sie auch ein CA-Zertifikat, um über TLS eine Vertrauensstellung mit dem Back-End-Dienst aufzubauen. Sie können das CA-Zertifikat mit dem folgenden Befehl abrufen:

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output
cacert.pem
```

Schritt 4: Testen Sie die AWS IoT Anmeldeinformationen mit Ihrem Kinesis-Videostream

Jetzt können Sie die AWS IoT Anmeldedaten testen, die Sie bisher eingerichtet haben.

1. Erstellen Sie zunächst einen Kinesis-Videostream, mit dem Sie diese Konfiguration testen möchten.

Important

Erstellen Sie einen Videostream mit einem Namen, der mit dem Namen des AWS IoT Dings identisch ist, den Sie im vorherigen Schritt erstellt haben (`kvs_example_camera_stream`).

```
aws kinesisvideo create-stream --data-retention-in-hours 24 --stream-name
kvs_example_camera_stream
```

2. Rufen Sie als Nächstes den Anbieter AWS IoT für Anmeldeinformationen an, um die temporären Anmeldeinformationen abzurufen:

```
curl --silent -H "x-amzn-iot-thingname:kvs_example_camera_stream" --cert
certificate.pem --key private.pem.key https://IOT_GET_CREDENTIAL_ENDPOINT/role-
aliases/KvsCameraIoTRoleAlias/credentials --cacert ./cacert.pem > token.json
```

Note

Sie können den folgenden Befehl verwenden, um Folgendes abzurufen `IOT_GET_CREDENTIAL_ENDPOINT`:

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

Das Ausgabe-JSON enthält den `AccessKey`, den `secretKey` und das `SessionToken`, mit denen Sie auf die Kinesis Video Streams zugreifen können.

3. Für Ihren Test können Sie diese Anmeldeinformationen verwenden, um die Kinesis Video Streams DescribeStream Streams-API für den `kvs_example_camera_stream` Beispielvideostream aufzurufen.

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json)
aws kinesisvideo describe-stream --stream-name kvs_example_camera_stream
```

Schritt 5: Bereitstellen von AWS IoT Zertifikaten und Anmeldeinformationen im Dateisystem Ihrer Kamera und Streamen von Daten in Ihren Videostream

Note

Die Schritte in diesem Abschnitt beschreiben das Senden von Medien an einen Kinesis-Videostream von einer Kamera, die den [the section called “C++ Producer-Bibliothek”](#) verwendet.

1. Kopieren Sie das X.509-Zertifikat, den privaten Schlüssel und das CA-Zertifikat, das in den vorherigen Schritten generiert wurde, in das Dateisystem Ihrer Kamera. Geben Sie die Pfade an, in denen diese Dateien gespeichert werden, den Rollenaliasnamen und den Endpunkt der AWS IoT Anmeldeinformationen für die Ausführung des `gst-launch-1.0` Befehls oder Ihrer Beispielanwendung.
2. Der folgende Beispielbefehl verwendet die AWS IoT Zertifikatsautorisierung, um Video an Kinesis Video Streams zu senden:

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink stream-
name="kvs_example_camera_stream" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem.key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias"
```

AWS IoT CertificateId als Streamname

Um Ihr Gerät (z. B. Ihre Kamera) durch ein AWS IoT Ding darzustellen, aber einen anderen Streamnamen zu autorisieren, können Sie das AWS IoT `certificateId` Attribut als Ihren Streamnamen verwenden und Kinesis Video Streams Streams-Berechtigungen für den Stream mit erteilen. AWS IoT Die Schritte, um dies zu erreichen, ähneln den zuvor beschriebenen, mit einigen Änderungen.

- Ändern Sie die Berechtigungsrichtlinie wie folgt auf Ihre IAM-Rolle (`iam-permission-document.json`):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:AwsCertificateId}/*"
    }
  ]
}
```

Note

Der Ressourcen-ARN verwendet die Zertifikat-ID als Platzhalter für den Streamnamen. Die IAM-Berechtigung funktioniert, wenn Sie die Zertifikat-ID als Streamnamen verwenden. Rufen Sie die Zertifikat-ID aus dem Zertifikat ab, sodass Sie diese als Streamnamen im folgenden beschreibenden Stream-API-Aufruf verwenden können.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Überprüfen Sie diese Änderung mit dem `describe-stream-CLI`-Befehl für Kinesis Video Streams:

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
```

```
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name ${CERTIFICATE_ID}
```

- Übergeben Sie die `certificateId` an den Anbieter für AWS IoT Anmeldeinformationen in der [Beispielanwendung](#) im Kinesis Video Streams C++ SDK:

```
credential_provider =
    make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
        cert_path,
        private_key_path,
        role_alias,
        ca_cert_path,
        certificateId);
```

Note

Beachten Sie, dass Sie den Thingname an den Anbieter für Anmeldeinformationen weitergeben. AWS IoT Sie können `getenv` verwenden, um den Dingnamen an die Demo-Anwendung zu übergeben, ähnlich wie bei der Übergabe der anderen Attribute. AWS IoT Verwenden Sie die Zertifikat-ID als Streamnamen in den Befehlszeilenparametern, wenn Sie die Beispielanwendung ausführen.

Verwenden Sie AWS IoT Anmeldeinformationen, um zu einem fest codierten Stream-Namen zu streamen

Um Ihr Gerät (z. B. Ihre Kamera) durch ein AWS IoT Ding darzustellen, aber das Streaming zu einem bestimmten Amazon Kinesis-Videostream zu autorisieren, geben Sie Amazon Kinesis Video Streams-Berechtigungen für den Stream, der verwendet. AWS IoT Der Vorgang ähnelt den vorherigen Abschnitten, mit einigen Änderungen.

Ändern Sie die Berechtigungsrichtlinie wie folgt auf Ihre IAM-Rolle (`iam-permission-document.json`):

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

```

    "Action": [
      "kinesisvideo:DescribeStream",
      "kinesisvideo:PutMedia",
      "kinesisvideo:TagStream",
      "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
  }
]
}

```

Kopieren Sie das X.509-Zertifikat, den privaten Schlüssel und das CA-Zertifikat, die in den vorherigen Schritten generiert wurden, in das Dateisystem Ihrer Kamera.

Geben Sie die Pfade an, in denen diese Dateien gespeichert werden, den Rollenaliasnamen, den AWS IoT Namen des Dings und den Endpunkt der AWS IoT Anmeldeinformationen für die Ausführung des `gst-launch-1.0` Befehls oder Ihrer Beispielanwendung.

Der folgende Beispielbefehl verwendet die AWS IoT Zertifikatsautorisierung, um Videos an Amazon Kinesis Video Streams zu senden:

```

gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink
stream-name="YourStreamName" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem.key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias,iot-
thing-name=YourThingName"

```

Überwachung von Amazon Kinesis Video Streams

Kinesis Video Streams bietet Überwachungsfunktionen für Ihre Lieferstreams. Weitere Informationen finden Sie unter [Überwachen](#).


Konformitätsprüfung für Amazon Kinesis Video Streams

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter heruntergeladenen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

 Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmapen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).

- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in Amazon Kinesis Video Streams

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Infrastruktursicherheit in Kinesis Video Streams

Als verwalteter Service ist Amazon Kinesis Video Streams durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper [Amazon Web Services: Sicherheitsprozesse im Überblick](#) beschrieben werden.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Kinesis Video Streams zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Clients müssen außerdem Cipher-Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman (DHE) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Die meisten modernen Systeme, z. B. Java 7 und höher, unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem IAM-Prinzipal zugeordnet ist, signiert werden. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Bewährte Sicherheitsmethoden für Kinesis Video Streams

Amazon Kinesis Video Streams bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die

folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Bewährte Sicherheits-Methoden für Ihre Remote-Geräte finden Sie unter [Bewährte Sicherheitsmethoden für Geräteagenten](#).

Implementieren des Zugriffs mit geringsten Berechtigungen

Bei der Erteilung von Berechtigungen entscheiden Sie, wer welche Berechtigungen für welche Kinesis Video Streams Streams-Ressourcen erhält. Sie aktivieren die spezifischen Aktionen, die daraufhin für die betreffenden Ressourcen erlaubt sein sollen. Aus diesem Grund sollten Sie nur Berechtigungen gewähren, die zum Ausführen einer Aufgabe erforderlich sind. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Ein Producer, der Daten an Kinesis Video Streams sendet, benötigt beispielsweise nur `PutMediaGetStreamingEndpoint`, und `DescribeStream`. Vergeben Sie keine Berechtigungen für Producer-Anwendungen für alle Aktionen (*) oder für andere Aktionen, wie z. B. `GetMedia`.

Weitere Informationen finden Sie unter [Was bedeutet das Konzept der geringsten Rechte und warum ist es erforderlich?](#)

Verwenden von IAM-Rollen

Produzenten- und Client-Anwendungen müssen über gültige Anmeldeinformationen verfügen, um auf Kinesis Video Streams zugreifen zu können. AWS -Anmeldeinformationen sollten Sie nicht direkt in einer Client-Anwendung oder in einem Amazon-S3-Bucket speichern. Dabei handelt es sich um langfristige Zugangsdaten, die nicht automatisch ausgetauscht werden und erhebliche Auswirkungen auf das Geschäft haben könnten, wenn sie kompromittiert werden.

Stattdessen sollten Sie eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Ihre Produzenten- und Client-Anwendungen für den Zugriff auf Kinesis Video Streams zu verwalten. Wenn Sie eine Rolle verwenden, müssen Sie keine langfristigen Anmeldeinformationen (wie einen Benutzernamen und ein Passwort oder Zugriffsschlüssel) verwenden, um auf andere Ressourcen zuzugreifen.

Weitere Informationen finden Sie unter folgenden Themen im IAM-Benutzerhandbuch:

- [IAM-Rollen](#)
- [Gängige Szenarien für Rollen: Benutzer, Anwendungen und Services](#)

Wird CloudTrail zur Überwachung von API-Aufrufen verwendet

Kinesis Video Streams funktioniert mit einem Dienst AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS-Service in Kinesis Video Streams bereitstellt.

Sie können die von gesammelten Informationen verwenden, CloudTrail um die Anfrage an Kinesis Video Streams, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details zu ermitteln.

Weitere Informationen finden Sie unter [the section called "Protokollieren von CloudTrail-API-Aufrufen mit"](#).

Produzentenbibliotheken von Kinesis Video Streams

Die Amazon Kinesis Video Streams Producer-Bibliotheken sind eine Reihe von Bibliotheken im Kinesis Video Streams Producer SDK. Der Client verwendet die Bibliotheken und das SDK, um die geräteinterne Anwendung für die sichere Verbindung mit Kinesis Video Streams zu erstellen und Mediendaten zu streamen, die in Echtzeit in der Konsole oder in den Client-Anwendungen angezeigt werden können.

Mediendaten können auf folgende Arten gestreamt werden:

- In Echtzeit
- Nach dem Puffern für ein paar Sekunden
- Nach den Medien-Uploads

Nachdem Sie einen Kinesis Video Streams-Stream erstellt haben, können Sie damit beginnen, Daten an ihn zu senden. Sie können das SDK verwenden, um Anwendungscode zu erstellen, der die Videodaten, sogenannte Frames, aus der Medienquelle extrahiert und in Kinesis Video Streams hochlädt. Diese Anwendungen werden auch als Produzentenanwendungen bezeichnet.

Die Producer-Bibliotheken enthalten folgende Komponenten:

- [Produzent und Kunde von Kinesis Video Streams](#)
- [Produzentenbibliothek von Kinesis Video Streams](#)

Produzent und Kunde von Kinesis Video Streams

Der Kinesis Video Streams Producer Client beinhaltet eine `KinesisVideoClient`-Klasse. Diese Klasse verwaltet Medienquellen, empfängt Daten aus den Quellen und verwaltet den Stream-Lebenszyklus, während Daten von einer Medienquelle zu Kinesis Video Streams fließen. Es bietet auch eine `MediaSource`-Schnittstelle zur Definition der Interaktion zwischen Kinesis Video Streams und Ihrer proprietären Hard- und Software.

Eine Medienquelle kann nahezu alles sein. Sie können beispielsweise eine Kamera oder ein Mikrofon als Medienquelle verwenden. Medienquellen sind aber nicht auf Audio- und Videoquellen beschränkt. So können beispielsweise auch als Textdateien vorliegende Protokolldaten als Datenstream gesendet werden. Sie können außerdem die Videodaten mehrerer Kameras gleichzeitig über Ihr Smartphone streamen.

Um die Daten dieser Quellen abzurufen, können Sie die Schnittstelle `MediaSource` implementieren. Diese Schnittstelle ermöglicht weitere Einsatzgebiete, für die wir keine integrierte Unterstützung bieten. Sie könnten sich beispielsweise dafür entscheiden, Folgendes an Kinesis Video Streams zu senden:

- Diagnosedatenstream (z. B. Anwendungsprotokolle und -ereignisse)
- Daten von Infrarotkameras, Radargeräten oder Tiefenkameras

Kinesis Video Streams bietet keine integrierten Implementierungen für Medienproduktionsgeräte wie Kameras. Um Daten von diesen Geräten zu extrahieren, müssen Sie den entsprechenden Code selbst implementieren, d. h. eine benutzerdefinierte Medienquelle erstellen. Sie können dann Ihre benutzerdefinierten Medienquellen explizit registrieren bei `KinesisVideoClient`, das die Daten auf Kinesis Video Streams hochlädt.

Der Kinesis Video Streams Producer Client ist für Java- und Android-Anwendungen verfügbar. Weitere Informationen erhalten Sie unter [Verwenden der Java Producer-Bibliothek](#) und [Verwenden der Android Producer-Bibliothek](#).

Produzentenbibliothek von Kinesis Video Streams

Die Kinesis Video Streams Producer Library ist im Kinesis Video Streams Producer Client enthalten. Die Bibliothek kann auch direkt für diejenigen verwendet werden, die eine tiefere Integration mit Kinesis Video Streams wünschen. Sie ermöglicht die Integration von Geräten mit proprietären Betriebssystemen, Netzwerk-Stacks oder eingeschränkten Gerätere Ressourcen.

Die Kinesis Video Streams Producer Library implementiert die State Machine für das Streaming zu Kinesis Video Streams. Sie stellt Callback-Haken bereit, die voraussetzen, dass Sie eine eigene Transportimplementierung bereitstellen und jede ein- und ausgehende Nachricht des Services explizit verarbeiten.

Sie könnten sich aus den folgenden Gründen dafür entscheiden, die Kinesis Video Streams Producer Library direkt zu verwenden:

- Das Gerät, auf dem die Anwendung ausgeführt werden soll, verfügt nicht über eine Java Virtual Machine.
- Sie möchten die Anwendung in einer anderen Sprache als Java programmieren.
- Sie möchten den Overhead in Ihrem Code reduzieren und ihn aufgrund von Einschränkungen wie Speicher und Rechenleistung auf das absolute Minimum an Abstraktion beschränken.

Derzeit ist die Kinesis Video Streams Producer Library für Android-, C-, C++- und Java-Anwendungen verfügbar. Weitere Informationen finden Sie in den folgenden unterstützten Sprachen Verwandte Themen.

Verwandte Themen

[Verwenden der Java Producer-Bibliothek](#)

[Verwenden der Android Producer-Bibliothek](#)

[Verwenden der C++ Producer-Bibliothek](#)

[Verwenden der C-Producer-Bibliothek](#)

[Verwenden des C++ Producer SDK auf Raspberry Pi](#)

Verwenden der Java Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte Java Producer Library verwenden, um Anwendungscode mit minimaler Konfiguration zu schreiben und Mediendaten von einem Gerät an einen Kinesis Video Stream zu senden.

Führen Sie die folgenden Schritte aus, um Ihren Code in Kinesis Video Streams zu integrieren, damit Ihre Anwendung mit dem Streamen von Daten in Ihren Kinesis-Videostream beginnen kann:

1. Erstellen Sie eine Instanz des Objekts `KinesisVideoClient`.
2. Erstellen Sie ein `MediaSource`-Objekt, indem Sie die Medienquelleninformationen bereitstellen. Wenn Sie beispielsweise eine Kamera-Medienquelle erstellen, geben Sie Informationen an, die die Kamera identifizieren und die von der Kamera verwendete Codierung festlegen.

Wenn Sie mit dem Streamen beginnen möchten, müssen Sie eine benutzerdefinierte Medienquelle erstellen.

3. Registrieren Sie die Medienquelle bei `KinesisVideoClient`.

Nachdem Sie die Medienquelle beim `KinesisVideoClient`-Objekt registriert haben, ruft die Medienquelle jedes Mal, wenn Daten verfügbar werden, das `KinesisVideoClient`-Objekt mit den Daten auf.

Vorgehensweise: Verwenden des Java Producer-SDK

Dieses Verfahren zeigt, wie Sie den Java Producer Client von Kinesis Video Streams in Ihrer Java-Anwendung verwenden, um Daten an Ihren Kinesis Video Stream zu senden.

Für diese Schritte benötigen Sie keine Medienquelle wie eine Kamera oder ein Mikrofon. Der Code erzeugt stattdessen zu Testzwecken Beispiel-Frames, die aus einer Reihe von Bytes besteht. Sie können das gleiche Codierungsmuster zum Senden von Mediendaten aus echten Quellen wie Kameras und Mikrofonen verwenden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- [Herunterladen und Konfigurieren des Codes](#)
- [Schreiben und Untersuchen des Codes](#)
- [Ausführen und Prüfen des Codes](#)

Voraussetzungen

- Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei eingerichtet haben. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein. Weitere Informationen finden Sie unter [Einrichten von AWS Anmeldeinformationen und Region für die Entwicklung](#) im AWS SDK for Java.

Note

Das Java-Beispiel verwendet ein `-SystemPropertiesCredentialsProvider` Objekt, um Ihre Anmeldeinformationen abzurufen. Das Provider-Objekt liest diese Anmeldeinformationen aus den Java-Systemeigenschaften `aws.accessKeyId` und `aws.secretKey` aus. Diese Systemeigenschaften werden in der Java-Entwicklungsumgebung konfiguriert. Informationen zum Festlegen von Java-Systemeigenschaften finden Sie in der Dokumentation der integrierten Entwicklungsumgebung (IDE).

- Ihr `NativeLibraryPath` muss Ihre `-KinesisVideoProducerJNI` Datei enthalten, verfügbar unter <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>. Die Namensweiterung für diese Datei hängt von Ihrem Betriebssystem ab:
 - `KinesisVideoProducerJNI.so` für Linux

- KinesisVideoProducerJNI.dylib für macOS
- KinesisVideoProducerJNI.dll für Windows

Note

Vorgefertigte Bibliotheken für macOS , Ubuntu, Windows und Raspbian sind in `src/main/resources/lib` unter [https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java.git](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java) verfügbar. Für andere Umgebungen kompilieren Sie die Datei [C++ Producer-Bibliothek](#).

Schritt 1: Herunterladen und Konfigurieren des Java Producer Library-Codes

In diesem Abschnitt der Vorgehensweise zur Java Producer-Bibliothek laden Sie den Java-Beispielcode herunter, importieren das Projekt in die Java-IDE und konfigurieren die Bibliotheksverzeichnisse.

Informationen zu den Voraussetzungen und weitere Details zu diesem Beispiel finden Sie unter [Verwenden der Java Producer-Bibliothek](#).

1. Erstellen Sie ein Verzeichnis und klonen Sie dann den Beispielquellcode aus dem GitHub Repository.

```
$ git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. Öffnen Sie die von Ihnen verwendete integrierte Entwicklungsumgebung (IDE) von Java (z. B. [Eclipse](#) oder [JetBrains IntelliJ IDEA](#)) und importieren Sie das heruntergeladene Apache-Maven-Projekt:
 - In IntelliJ IDEA: Klicken Sie auf Import. Navigieren Sie zur Datei `pom.xml` im Stammverzeichnis des heruntergeladenen Pakets.
 - In Eclipse: Klicken Sie auf File, Import, Maven, Existing Maven Projects. Navigieren Sie anschließend zum Verzeichnis `kinesis-video-java-demo`.

Weitere Informationen finden Sie in der Dokumentation zur IDE.

3. Der Java-Beispielcode verwendet die aktuellen AWS Anmeldeinformationen. Wenn Sie ein anderes Anmeldeinformationsprofil verwenden möchten, suchen Sie in der Datei `DemoAppMain.java` nach folgendem Code:

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Ändern Sie den Code wie folgt:

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        new ProfileCredentialsProvider("credentials-profile-name"));
```

Weitere Informationen finden Sie unter [ProfileCredentialsProvider](#) in der AWS SDK for Java-Referenz.

Nächster Schritt

[the section called “Schritt 2: Schreiben und Untersuchen des Codes”](#)

Schritt 2: Schreiben und Untersuchen des Codes

In diesem Abschnitt des [Java Producer Library-Verfahrens](#) schreiben und untersuchen Sie den Java-Beispielcode, den Sie im vorherigen Abschnitt heruntergeladen haben.

Die Java-Anwendung ([DemoAppMain](#)) führt folgende Aufgaben durch:

- Erstellen Sie eine Instance von `KinesisVideoClient`.
- Erstellen Sie eine Instance von `MediaSource`.
- Die `MediaSource`-Instanz wird beim Client registriert.
- Das Streamen wird gestartet. Starten Sie und `MediaSource` es beginnt, Daten an den Client zu senden.

Die folgenden Abschnitte stellen Details bereit.

Erstellen einer Instance von KinesisVideoClient

Das `KinesisVideoClient`-Objekt wird durch einen Aufruf der Operation `createKinesisVideoClient` erstellt.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Damit `KinesisVideoClient` auf das Netzwerk zugreifen kann, benötigt das Objekt Anmeldeinformationen zur Authentifizierung. Sie übergeben dazu eine Instanz von `SystemPropertiesCredentialsProvider`, die den `AWSCredentials`-Wert des Standardprofils in der Anmeldeinformationsdatei ausliest:

```
[default]
aws_access_key_id = ABCDEFGHIJKLMNOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

Erstellen einer Instance von MediaSource

Um Bytes an Ihren Kinesis-Videostream zu senden, müssen Sie die Daten erzeugen. Amazon Kinesis Video Streams stellt die `MediaSource` Schnittstelle bereit, die die Datenquelle darstellt.

Die Java-Bibliothek von Kinesis Video Streams bietet beispielsweise die `ImageFileMediaSource` Implementierung der `MediaSource`-Schnittstelle. Diese Klasse liest nur Daten aus einer Reihe von Mediendateien und nicht aus einem Kinesis-Videostream, aber Sie können sie zum Testen des Codes verwenden.

```
final MediaSource bytesMediaSource = createImageFileMediaSource();
```

Registrieren des MediaSource beim Client

Registrieren Sie die erstellte Medienquelle beim `KinesisVideoClient`-Objekt, damit sie den Client kennt (und diesem Daten senden kann).

```
kinesisVideoClient.registerMediaSource(mediaSource);
```

Starten der Medienquelle

Starten Sie die Medienquelle, damit sie mit der Generierung von Daten beginnen und an den Client senden kann.

```
bytesMediaSource.start();
```

Nächster Schritt

[the section called "Schritt 3: Ausführen und Überprüfen des Codes"](#)

Schritt 3: Ausführen und Überprüfen des Codes

Gehen Sie wie folgt vor, um die Java-Testumgebung für die [Java Producer Library](#) auszuführen.

1. Wählen Sie DemoAppMain.
2. Wählen Sie Ausführen , Ausführen 'DemoAppMain'.
3. Fügen Sie den JVM-Argumenten für die Anwendung Ihre Anmeldeinformationen hinzu:
 - Für nicht-temporäre AWS Anmeldeinformationen: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
 - Für temporäre AWS Anmeldeinformationen: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"
4. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Kinesis-Video-Streams-Konsole](#).

Wählen Sie auf der Seite Manage Streams den entsprechenden Stream aus.

5. Das Beispielvideo wird im eingebetteten Player wiedergegeben. Sie müssen möglicherweise kurz warten (bis zu zehn Sekunden unter normalen Bandbreiten- und Prozessorbedingungen), während die Frames angesammelt werden, bevor das Video angezeigt wird.

Das Codebeispiel erstellt einen Stream. Sobald das `MediaSource`-Objekt im Code gestartet wird, beginnt es mit dem Senden von Beispiel-Frames an das `KinesisVideoClient`-Objekt. Der Client sendet die Daten dann an Ihren Kinesis-Videostream.

Verwenden der Android Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte Android Producer Library verwenden, um Anwendungscode mit minimaler Konfiguration zu schreiben und Mediendaten von einem Android-Gerät an einen Kinesis-Videostream zu senden.

Führen Sie die folgenden Schritte aus, um Ihren Code in Kinesis Video Streams zu integrieren, damit Ihre Anwendung mit dem Streamen von Daten in Ihren Kinesis-Videostream beginnen kann:

1. Erstellen Sie eine Instanz des Objekts `KinesisVideoClient`.
2. Erstellen Sie ein `MediaSource`-Objekt, indem Sie die Medienquelleninformationen bereitstellen. Wenn Sie beispielsweise eine Kamera-Medienquelle erstellen, geben Sie Informationen an, die die Kamera identifizieren und die von der Kamera verwendete Codierung festlegen.

Wenn Sie mit dem Streamen beginnen möchten, müssen Sie eine benutzerdefinierte Medienquelle erstellen.

Vorgehensweise: Verwenden des Android Producer-SDK

Dieses Verfahren zeigt, wie Sie den Android Producer Client von Kinesis Video Streams in Ihrer Android-Anwendung verwenden, um Daten an Ihren Kinesis Video Stream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- [the section called “Voraussetzungen”](#)
- [the section called “Schritt 1: Herunterladen und Konfigurieren des Codes”](#)
- [the section called “Schritt 2: Untersuchen des Codes”](#)
- [the section called “Schritt 3: Ausführen und Überprüfen des Codes”](#)

Voraussetzungen

- Wir empfehlen [Android Studio](#) für das Überprüfen, Bearbeiten und Ausführen des Anwendungscodes. Wir empfehlen, die neueste stabile Version zu verwenden.

- Im Beispielcode geben Sie Amazon Cognito-Anmeldeinformationen an.

Gehen Sie wie folgt vor, um einen Amazon Cognito-Benutzerpool und einen Identitäten-Pool einzurichten.

- [Einrichten eines Benutzerpools](#)
- [Einrichten eines Identitäten-Pools](#)

Einrichten eines Benutzerpools

Einrichten eines Benutzerpools

1. Melden Sie sich bei der [Amazon Cognito-Konsole](#) an und überprüfen Sie, ob die Region korrekt ist.
2. Wählen Sie in der Navigation auf der linken Seite Benutzerpools aus.
3. Wählen Sie im Abschnitt Benutzerpools die Option Benutzerpool erstellen aus.
4. Füllen Sie die folgenden Abschnitte aus:
 - a. Schritt 1: Konfigurieren der Anmeldeerfahrung – Wählen Sie im Abschnitt Cognito-Benutzerpool-Anmeldeoptionen die entsprechenden Optionen aus.

Klicken Sie auf Weiter.
 - b. Schritt 2: Konfigurieren der Sicherheitsanforderungen – Wählen Sie die entsprechenden Optionen aus.

Klicken Sie auf Weiter.
 - c. Schritt 3: Konfigurieren der Anmeldeerfahrung – Wählen Sie die entsprechenden Optionen aus.

Klicken Sie auf Weiter.
 - d. Schritt 4: Konfigurieren der Nachrichtenzustellung – Wählen Sie die entsprechenden Optionen aus.

Wählen Sie im Auswahlfeld IAM-Rolle eine vorhandene Rolle aus oder erstellen Sie eine neue Rolle.

Klicken Sie auf Weiter.
 - e. Schritt 5: Integrieren Ihrer App – Wählen Sie die entsprechenden Optionen aus.

Wählen Sie im Feld Anfänglicher App-Client die Option vertraulicher Client aus.

Klicken Sie auf Weiter.

- f. Schritt 6: Überprüfen und erstellen – Überprüfen Sie Ihre Auswahl aus den vorherigen Abschnitten und wählen Sie dann Benutzerpool erstellen aus.

5. Wählen Sie auf der Seite Benutzerpools den Pool aus, den Sie gerade erstellt haben.

Kopieren Sie die Benutzerpool-ID und notieren Sie sich dies für später. In der `awsconfiguration.json` Datei ist dies `CognitoUserPool.Default.PoolId`.

6. Wählen Sie die Registerkarte App-Integration und gehen Sie unten auf der Seite.
7. Wählen Sie im Abschnitt App-Clientliste den App-Clientnamen aus, den Sie gerade erstellt haben.

Kopieren Sie die Client-ID und notieren Sie sich diese für später. In der `awsconfiguration.json` Datei ist dies `CognitoUserPool.Default.AppClientId`.

8. Zeigen Sie das Client-Secret an und notieren Sie es sich für später. In der `awsconfiguration.json` Datei ist dies `CognitoUserPool.Default.AppClientSecret`.

Einrichten eines Identitäten-Pools

Einrichten eines Identitäten-Pools

1. Melden Sie sich bei der [Amazon Cognito-Konsole](#) an und überprüfen Sie, ob die Region korrekt ist.
2. Wählen Sie in der Navigation auf der linken Seite Identitätspools aus.
3. Wählen Sie Identitätspool erstellen.
4. Konfigurieren Sie den Identitäten-Pool.
 - a. Schritt 1: Konfigurieren der Identitätspool-Vertrauensstellung – Füllen Sie die folgenden Abschnitte aus:
 - Benutzerzugriff – Authentifizierten Zugriff auswählen
 - Authentifizierte Identitätsquellen – Amazon Cognito-Benutzerpool auswählen

Klicken Sie auf Weiter.

- b. Schritt 2: Konfigurieren von Berechtigungen – Füllen Sie im Abschnitt Authentifizierte Rolle die folgenden Felder aus:

- IAM-Rolle – Wählen Sie Neue IAM-Rolle erstellen aus
- IAM-Rollenname – Geben Sie einen Namen ein und notieren Sie ihn für einen späteren Schritt.

Klicken Sie auf Weiter.

- c. Schritt 3: Identitätsanbieter verbinden – Füllen Sie im Abschnitt Benutzerpool-Details die folgenden Felder aus:

- Benutzerpool-ID – Wählen Sie den Benutzerpool aus, den Sie zuvor erstellt haben.
- App-Client-ID – Wählen Sie die App-Client-ID aus, die Sie zuvor erstellt haben.

Klicken Sie auf Weiter.

- d. Schritt 4: Konfigurieren von Eigenschaften – Geben Sie einen Namen in das Feld Name des Identitätspools ein.

Klicken Sie auf Weiter.

- e. Schritt 5: Überprüfen und Erstellen – Überprüfen Sie Ihre Auswahl in jedem der Abschnitte und wählen Sie dann Identitäten-Pool erstellen aus.

5. Wählen Sie auf der Seite Identitätspools Ihren neuen Identitätspool aus.

Kopieren Sie die Identitätspool-ID und notieren Sie sich dies für später. In der `awsconfiguration.json` Datei ist dies `CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Aktualisieren Sie die Berechtigungen für die IAM-Rolle.

- Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- Wählen Sie in der Navigation auf der linken Seite Rollen aus.
- Suchen und wählen Sie die Rolle aus, die Sie oben erstellt haben.

Note

Verwenden Sie bei Bedarf die Suchleiste.

- d. Wählen Sie die angehängte Berechtigungsrichtlinie aus.

Wählen Sie Bearbeiten aus.

- e. Wählen Sie die Registerkarte JSON aus und ersetzen Sie die Richtlinie durch Folgendes:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cognito-identity:*",
        "kinesisvideo:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Klicken Sie auf Weiter.

- f. Aktivieren Sie das Kontrollkästchen neben Diese neue Version als Standard festlegen, falls sie noch nicht ausgewählt ist.

Wählen Sie Änderungen speichern aus.

Schritt 1: Herunterladen und Konfigurieren des Android Producer Library-Codes

In diesem Abschnitt des Android-Producer Library-Verfahrens laden Sie den Android-Beispielcode herunter und öffnen das Projekt in Android Studio.

Informationen zu den Voraussetzungen und weitere Details zu diesem Beispiel finden Sie unter [Verwenden der Android Producer-Bibliothek](#).

1. Erstellen Sie ein Verzeichnis und klonen Sie dann die AWS Mobile SDK for Android aus dem GitHub Repository.

```
$ git clone https://github.com/aws-labs/aws-sdk-android-samples
```

2. Öffnen Sie [Android Studio](#).
3. Wählen Sie auf dem Öffnungsbildschirm Open an existing Android Studio project.
4. Navigieren Sie zum Verzeichnis `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` und wählen Sie OK.
5. Öffnen Sie die `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json` Datei.

Geben Sie auf dem `CredentialsProvider` Knoten die Identitätspool-ID aus dem Abschnitt So richten Sie eine Identitätspool-[Prozedur im Abschnitt Voraussetzungen](#) ein und geben Sie Ihre an AWS-Region (z. B. **us-west-2**).

Geben Sie im `CognitoUserPool` Knoten das App-Client-Secret, die App-Client-ID und die Pool-ID aus dem Verfahren So richten Sie eine Benutzerpool-Prozedur im Abschnitt [Voraussetzungen](#) ein und geben Sie Ihre an AWS-Region (z. B. **us-west-2**).

6. Ihre `awsconfiguration.json`-Datei sollte wie folgt aussehen:

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "us-west-2:01234567-89ab-cdef-0123-456789abcdef",
        "Region": "us-west-2"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "CognitoUserPool": {
    "Default": {
```

```
"AppClientSecret": "abcdefghijklmnopqrstuvwxy0123456789abcdefghijklmnop",
"AppClientId": "0123456789abcdefghijklmnop",
"PoolId": "us-west-2_qRsTuVwXy",
"Region": "us-west-2"
}
}
}
```

7. Aktualisieren Sie die `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` mit Ihrer Region (im folgenden Beispiel ist sie auf `US_WEST_2` festgelegt):

```
public class KinesisVideoDemoApp extends Application {
    public static final String TAG = KinesisVideoDemoApp.class.getSimpleName();
    public static Regions KINESIS_VIDEO_REGION = Regions.US_WEST_2;
```

Weitere Informationen zu AWS-Region Konstanten finden Sie unter [Regionen](#).

Nächster Schritt

[the section called "Schritt 2: Untersuchen des Codes"](#)

Schritt 2: Untersuchen des Codes

In diesem Abschnitt der [Vorgehensweise zur Android Producer-Bibliothek](#) untersuchen Sie den Beispielcode.

Die Android-Testanwendung (`AmazonKinesisVideoDemoApp`) zeigt das folgende Codiermuster:

- Erstellen Sie eine Instance von `KinesisVideoClient`.
- Erstellen Sie eine Instance von `MediaSource`.
- Das Streamen wird gestartet. Starten Sie die und sie beginnt `MediaSource`, Daten an den Client zu senden.

Die folgenden Abschnitte stellen Details bereit.

Erstellen einer Instance von KinesisVideoClient

Das [KinesisVideoClient](#)-Objekt wird durch einen Aufruf der Operation [createKinesisVideoClient](#) erstellt.

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(
    getActivity(),
    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,
    KinesisVideoDemoApp.getCredentialsProvider());
```

Damit `KinesisVideoClient` auf das Netzwerk zugreifen kann, benötigt das Objekt Anmeldeinformationen zur Authentifizierung. Sie übergeben eine Instance von `AWSCredentialsProvider`, die Ihre Amazon Cognito-Anmeldeinformationen aus der `awsconfiguration.json` Datei liest, die Sie im vorherigen Abschnitt geändert haben.

Erstellen einer Instance von MediaSource

Um Bytes an Ihren Kinesis-VideoStream zu senden, müssen Sie die Daten erzeugen. Amazon Kinesis Video Streams stellt die [MediaSource](#) Schnittstelle bereit, die die Datenquelle darstellt.

Die Android-Bibliothek von Kinesis Video Streams bietet beispielsweise die [AndroidCameraMediaSource](#) Implementierung der `MediaSource` Schnittstelle. Dieser Kurs liest Daten aus einer der Kameras des Speichergeräts.

Im folgenden Codebeispiel (aus der [fragment/StreamConfigurationFragment.java](#)-Datei) wird die Konfiguration für die Medien-Quelle erstellt:

```
private AndroidCameraMediaSourceConfiguration getCurrentConfiguration() {
    return new AndroidCameraMediaSourceConfiguration(
        AndroidCameraMediaSourceConfiguration.builder()
            .withCameraId(mCamerasDropdown.getSelectedItem().getCameraId())

            .withEncodingMimeType(mMimeTypeDropdown.getSelectedItem().getMimeType())

            .withHorizontalResolution(mResolutionDropdown.getSelectedItem().getWidth())

            .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
                .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
                .withIsEncoderHardwareAccelerated(
                    mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
```

```
        .withFrameRate(FRAMERATE_20)
        .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
        .withEncodingBitRate(BITRATE_384_KBPS)
        .withCameraOrientation(-
mCamerasDropdown.getSelectedItem().getCameraOrientation())

        .withNalAdaptationFlags(StreamInfo.NalAdaptationFlags.NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS
            .withIsAbsoluteTimecode(false));
    }
```

Im folgenden Codebeispiel (aus der [fragment/StreamingFragment.java](#)-Datei) wird die Konfiguration für die Medien-Quelle erstellt:

```
mCameraMediaSource = (AndroidCameraMediaSource) mKinesisVideoClient
    .createMediaSource(mStreamName, mConfiguration);
```

Starten der Medienquelle

Starten Sie die Medienquelle, damit diese mit dem Erzeugen von Daten und Senden an den Client beginnen kann. Das folgende Codebeispiele stammt aus der Datei [fragment/StreamingFragment.java](#).

```
mCameraMediaSource.start();
```

Nächster Schritt


[the section called “Schritt 3: Ausführen und Überprüfen des Codes”](#)

Schritt 3: Ausführen und Überprüfen des Codes

Gehen Sie wie folgt vor, um die Android-Beispielanwendung für die [Android Producer-Bibliothek](#) auszuführen.

1. Verbindung mit einem Android-Gerät.
2. Wählen Sie Run, Run... und dann Edit configurations....
3. Wählen Sie das Plusymbol (+) und die Android-App aus. Geben Sie im Feld Name **AmazonKinesisVideoDemoApp** ein. Wählen Sie im Pulldown Modul die Option ausAmazonKinesisVideoDemoApp. Wählen Sie OK aus.

4. Wählen Sie Run, Run.
5. Wählen Sie auf dem Bildschirm Select a Deployment Target Ihr verbundenes Gerät aus und klicken Sie auf OK.
6. Wählen Sie in der AWSKinesisVideoDemoApp Anwendung auf dem Gerät die Option Neues Konto erstellen aus.
7. Geben Sie Werte für USERNAME, Password, Given name, Email address und Phone number ein und wählen Sie Sign up.

 Note


Für diese Werte gelten die folgenden Einschränkungen:

- Passwort: Muss eine Kombination aus Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen enthalten. Sie können diese Einschränkungen auf der Seite Benutzerpool in der [Amazon Cognito-Konsole](#) ändern.
- E-Mail-Adresse: Muss eine gültige Adresse sein, damit Sie einen Bestätigungscode erhalten können.
- Telefonnummer: Muss das folgende Format haben: **+<Country code><Number>**, z. B. **+12065551212** .

8. Geben Sie den Code ein, den Sie per E-Mail erhalten, und wählen Sie Bestätigen aus. Wählen Sie Ok aus.
9. Behalten Sie auf der nächsten Seite die Standardwerte bei und wählen Sie Streamen aus.
10. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Kinesis-Video-Streams-Konsole](#) in der Region USA West (Oregon).

Wählen Sie auf der Seite Manage Streams den Eintrag demo-stream.

11. Das Streaming-Video wird im eingebetteten Player wiedergegeben. Sie müssen möglicherweise kurz warten (bis zu zehn Sekunden unter normalen Bandbreiten- und Prozessorbedingungen), während die Frames angesammelt werden, bevor das Video angezeigt wird.

 Note

Wenn sich der Bildschirm des Geräts dreht (z. B. von Hochformat zu Querformat), stoppt die Anwendung das Streamen von Videos.

Das Codebeispiel erstellt einen Stream. Sobald das `MediaSource`-Objekt im Code gestartet wird, beginnt es mit dem Senden von Beispiel-Frames von der Kamera an das `KinesisVideoClient`-Objekt. Der Client sendet die Daten dann an einen Kinesis-Videostream mit dem Namen `demo-stream`.

Verwenden der C++ Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte C++ Producer Library verwenden, um Anwendungscode zu schreiben, um Mediendaten von einem Gerät an einen Kinesis-Videostream zu senden.

Objektmodell

Die C++-Bibliothek stellt die folgenden Objekte zur Verwaltung des Sendens von Daten an einen Kinesis-Videostream bereit:

- `KinesisVideoProducer`: Enthält Informationen zu Ihrer Medienquelle und Ihren AWS Anmeldeinformationen und verwaltet Rückrufe, um über Kinesis Video Streams Streams-Ereignisse zu berichten.
- `KinesisVideoStream`: Stellt den Kinesis-Videostream dar. Enthält Informationen zu den Parametern des Videostreams wie Name, Datenaufbewahrungsdauer und Art des Medieninhalts.

Medien in den Stream bringen

Sie können die von der C++-Bibliothek bereitgestellten Methoden verwenden (z. B. `PutFrame`), um Daten in das `KinesisVideoStream` Objekt einzufügen. Die Bibliothek verwaltet dann den internen Status der Daten. Das umfasst beispielsweise folgende Aufgaben:

- Durchführen der Authentifizierung.
- Überwachen auf Netzwerklatenz. Wenn die Latenz zu hoch ist, verwirft die Bibliothek gegebenenfalls Frames.
- Überwachen des Status des laufenden Streamings.

Callback-Schnittstellen

Ein Layer stellt eine Reihe von Callback-Schnittstellen für die Kommunikation mit dem Anwendungslayer bereit. Verfügbare Callback-Schnittstellen:

- Schnittstelle für Service-Callbacks (`CallbackProvider`): Die Bibliothek ruft Ereignisse auf, die über diese Schnittstelle abgerufen werden, wenn sie einen Stream erstellt, eine Stream-Beschreibung abrufen und einen Stream löscht.
- Schnittstelle für "Client-ready"-Status oder "Wenig Speicher"-Ereignisse (`ClientCallbackProvider`): Die Bibliothek ruft Ereignisse über diese Schnittstelle auf, wenn der Client bereit ist oder festgestellt wird, dass der Arbeits- oder Festplattenspeicher knapp wird.
- Callback-Schnittstelle für Stream-Ereignisse (`StreamCallbackProvider`): Die Bibliothek ruft Ereignisse über diese Schnittstelle auf, wenn Stream-Ereignisse auftreten, der Stream also beispielsweise den Status "Bereit" erhält, Frames verworfen werden oder Stream-Fehler auftreten.

Kinesis Video Streams bietet Standardimplementierungen für diese Schnittstellen. Sie können auch Ihre eigene benutzerdefinierte Implementierung bereitstellen, zum Beispiel, wenn Sie eine benutzerdefinierte Netzwerklogik benötigen oder wenn Sie der Benutzeroberfläche einen geringen Speicherplatz zur Verfügung stellen möchten.

Weitere Informationen zu Callbacks in den Producer-Bibliotheken finden Sie unter [SDK-Rückrufe von Produzenten](#).

Vorgehensweise: Verwenden des C++ Producer-SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Streams-Client und die Medienquellen in einer C++-Anwendung verwenden, um Daten an Ihren Kinesis-Videostream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- [Schritt 1: Herunterladen und Konfigurieren des Codes](#)
- [Schritt 2: Schreiben und Untersuchen des Codes](#)
- [Schritt 3: Ausführen und Prüfen des Codes](#)

Voraussetzungen

- Anmeldeinformationen: Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei eingerichtet haben. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein.

Weitere Informationen finden Sie unter [AWSAnmeldeinformationen einrichten und Region für Entwicklung](#).

- Integration des Zertifikatsspeichers: Die Kinesis Video Streams Producer Library muss den aufgerufenen Dienst als vertrauenswürdig betrachten. Dies erfolgt durch die Validierung der Zertifizierungsstellen (CAs) im öffentlichen Zertifikatsspeicher. Bei Linux-basierten Modellen befindet sich dieser Speicher im Verzeichnis „/etc/ssl“.

Laden Sie das Zertifikat aus dem folgenden Speicherort in den Zertifikatsspeicher:

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installieren Sie die folgenden Build-Abhängigkeiten für macOS:
 - [Autoconf 2.69](#) (Lizenz GPLv3+/Autoconf: GNU GPL Version 3 oder höher)
 - [CMake 3.7 oder 3.8](#)
 - [Pkg-Config](#)
 - [Flex 2.5.35 Apple \(flex-31\) oder höher](#)
 - [Bison 2.4](#) (GNU-Lizenz)
 - [Automake 1.15.1](#) (GNU-Lizenz)
 - GNU Libtool (Apple Inc. Version cctools-898)
 - xCode (macOS)/clang/gcc (xcode-select Version 2347)
 - Java Development Kit (JDK) (für Java-JNI-Kompilierung)
 - [Lib-Pkg](#)
- Installieren Sie die folgenden Build-Abhängigkeiten für Ubuntu (Antworten auf Versionsbefehle werden abgeschnitten):
 - Installieren Sie Git: `sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- Installieren Sie [CMake](#): `sudo apt-get install cmake`

```
$ cmake --version
cmake version 3.9.1
```

- Installieren Sie Libtool: `sudo apt-get install libtool`

```
2.4.6-2
```

```
$ libtool --version
libtool (GNU libtool) 2.4.6
Written by Gordon Matzigkeit, 1996
```

- Installieren Sie GNU Automake: `sudo apt-get install automake`

```
$ automake --version
automake (GNU automake) 1.15
```

- Installieren Sie GNU Bison: `sudo apt-get install bison`

```
$ bison -V
bison (GNU Bison) 3.0.4
```

- Installieren Sie G++: `sudo apt-get install g++`

```
g++ --version
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- Installieren Sie curl: `sudo apt-get install curl`

```
$ curl --version
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- Installieren Sie pkg-config: `sudo apt-get install pkg-config`

```
$ pkg-config --version
0.29.1
```

- Installieren Sie Flex: `sudo apt-get install flex`

```
$ flex --version
flex 2.6.1
```

- Installieren Sie OpenJDK: `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- Legen Sie die JAVA_HOME-Umgebungsvariable fest: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
- Führen Sie das Build-Skript aus: `./install-script`

Nächster Schritt

[Schritt 1: Herunterladen und Konfigurieren des C++ Producer-Bibliothekscodes](#)

Schritt 1: Downloaden und konfigurieren Sie den Code der C++ Producer Library

Informationen zum Herunterladen und Konfigurieren der C++ Producer Library finden Sie unter [Amazon Kinesis Video Streams CPP Producer, GStreamer-Plugin](#) und JNI.

Voraussetzungen und weitere Informationen zu diesem Beispiel finden Sie unter [Verwenden der C++ Producer Library](#).

Nächster Schritt

[Schritt 2: Schreiben und untersuchen Sie den Code](#)

Schritt 2: Schreiben und untersuchen Sie den Code

In diesem Abschnitt der [Vorgehensweise zur C++ Producer-Bibliothek](#) untersuchen Sie den Code im C++-Testeinsatz (`tst/ProducerTestFixture.h` und andere Dateien). Sie haben diesen Code im vorherigen Abschnitt heruntergeladen.

Das plattformunabhängige C++-Beispiel zeigt das folgende Codierungsmuster:

- Erstellen Sie eine Instanz von `KinesisVideoProducer`, um auf Kinesis Video Streams zuzugreifen.
- Erstellen Sie eine Instance von `KinesisVideoStream`. Dadurch wird ein Kinesis-Videostream in Ihrem erstellt, AWS-Konto falls ein Stream mit dem gleichen Namen noch nicht existiert.
- Rufen Sie `putFrame` im `KinesisVideoStream` für jeden Daten-Frame auf, sobald er verfügbar wird, um den Frame an den Stream zu senden.

Die folgenden Abschnitte enthalten weitere Informationen zu diesem Codierungsmuster.

Erstellen einer Instanz von KinesisVideoProducer

Das KinesisVideoProducer-Objekt wird durch einen Aufruf der Methode `KinesisVideoProducer::createSync` erstellt. Im folgenden Beispiel wird `KinesisVideoProducer` in der Datei `ProducerTestFixture.h` erstellt:

```
kinesis_video_producer_ = KinesisVideoProducer::createSync(move(device_provider_),
    move(client_callback_provider_),
    move(stream_callback_provider_),
    move(credential_provider_),
    defaultRegion_);
```

Die Methode `createSync` verwendet die folgenden Parameter:

- Ein `DeviceInfoProvider`-Objekt, das ein `DeviceInfo`-Objekt mit den Daten zu einem Gerät oder einer Speicherkonfiguration zurückgibt.

Note

Sie konfigurieren die Größe Ihres Inhaltsspeichers unter Verwendung des Parameters `deviceInfo.storageInfo.storageSize`. Ihre Inhalts-Streams verwenden denselben Inhaltsspeicher. Um Ihren Speicherbedarf zu ermitteln, multiplizieren Sie die durchschnittliche Frame-Größe mit der Anzahl der für die maximale Dauer aller Streams gespeicherten Frames. Multiplizieren Sie das Ergebnis dann mit 1,2, um die Defragmentierung zu berücksichtigen. Angenommen, Ihre Anwendung verwendet die folgende Konfiguration:

- Drei Streams
- 3 Minuten maximale Dauer
- Jeder Stream ist 30 Frames pro Sekunde (FPS)
- Jeder Frame ist 10.000 KB groß

Die Inhaltsspeicheranforderung für diese Anwendung beträgt $3 \text{ (Streams)} * 3 \text{ (Minuten)} * 60 \text{ (Sekunden in einer Minute)} * 10000 \text{ (KB)} * 1,2 \text{ (Defragmentierungszulage)} = 194,4 \text{ Mb} \sim 200 \text{ Mb}$.

- Ein `ClientCallbackProvider`-Objekt, das Funktionszeiger zurückgibt, die Client-spezifische Ereignisse melden.

- Ein `StreamCallbackProvider`-Objekt, das Funktionszeiger zurückgibt, die per Callback aufgerufen werden, wenn Stream-spezifische Ereignisse auftreten.
- Ein `CredentialProvider` Objekt, das Zugriff auf Umgebungsvariablen mit AWS Anmeldeinformationen bietet.
- Die AWS-Region („us-west-2“). Der Service-Endpoint wird anhand der Region bestimmt.

Erstellen einer Instanz von `KinesisVideoStream`

Sie erstellen das `KinesisVideoStream`-Objekt, indem Sie die Methode `KinesisVideoProducer::CreateStream` mit einem `StreamDefinition`-Parameter aufrufen. Das Beispiel erstellt den `KinesisVideoStream` in der Datei `ProducerTestFixture.h` mit dem Track-Typ als Video und der ID 1:

```
auto stream_definition = make_unique<StreamDefinition>(stream_name,
                                                    hours(2),
                                                    tags,
                                                    "",
                                                    STREAMING_TYPE_REALTIME,
                                                    "video/h264",
                                                    milliseconds::zero(),
                                                    seconds(2),
                                                    milliseconds(1),
                                                    true,
                                                    true,
                                                    true);
return kinesis_video_producer_->createStream(move(stream_definition));
```

Das `StreamDefinition`-Objekt enthält folgende Felder:

- Stream-Name.
- Aufbewahrungszeitraum.
- Tags für den Stream. Diese Tags können von Konsumenten Anwendungen verwendet werden, um den richtigen Stream zu finden oder weitere Informationen über den Stream abzurufen. Die Tags können auch in der AWS Management Console angezeigt werden.
- AWS KMS-Verschlüsselungsschlüssel für den Stream. Weitere Informationen finden Sie unter [Verwenden serverseitiger Verschlüsselung mit Kinesis Video Streams](#).
- Streaming-Typ. Der einzige gültige Wert ist derzeit `STREAMING_TYPE_REALTIME`.

- Medieninhaltstyp.
- Medienlatenz. Dieser Wert wird derzeit nicht verwendet und sollte auf 0 gesetzt werden.
- Wiedergabedauer jedes Fragments.
- Medienzeitcode-Skala.
- Gibt an, ob die Medien Keyframe-Fragmentierung verwenden.
- Gibt an, ob die Medien Zeitcodes verwenden.
- Gibt an, ob die Medien absolute Fragmentzeiten verwenden.

Hinzufügen eines Audiodatenstroms zum Kinesis-Videostream

Sie können einer Videospur-Stream-Definition Audiospurdetails hinzufügen, indem Sie die `AddTrack`-Methode der `StreamDefinition` folgenden Methode verwenden:

```
stream_definition->addTrack(DEFAULT_AUDIO_TRACKID, DEFAULT_AUDIO_TRACK_NAME,  
    DEFAULT_AUDIO_CODEC_ID, MKV_TRACK_INFO_TYPE_AUDIO);
```

Für die `addTrack`-Methode sind folgende Parameter erforderlich:

- Track-ID (wie eine für Audio). Dies sollte ein eindeutiger Wert ungleich Null sein.
- Benutzerdefinierter Titelname (z. B. „Audio“ für die Audiospur).
- Codec-ID für diesen Track (z. B. für die Audiospur „A_AAC“).
- Spurtyp (verwenden Sie beispielsweise den Enum-Wert von `MKV_TRACK_INFO_TYPE_AUDIO` für Audio).

Wenn Sie private Codec-Daten für die Audiospur haben, können Sie sie beim Aufrufen der `addTrack`-Funktion übergeben. Sie können dem Codec auch private Daten senden, nachdem Sie das `KinesisVideoStream` Objekt erstellt haben und dabei die Startmethode aufgerufen haben.

`KinesisVideoStream`

Einfügen eines Frames in den Kinesis-Videostream

Sie fügen Medien in den Kinesis-Videostream ein `KinesisVideoStream::putFrame`, indem Sie ein `Frame` Objekt übergeben, das den Header und die Mediendaten enthält. Das Beispiel ruft `putFrame` in der Datei `ProducerApiTest.cpp` auf:

```
frame.duration = FRAME_DURATION_IN_MICROS * HUNDREDS_OF_NANOS_IN_A_MICROSECOND;
frame.size = sizeof(frameBuffer_);
frame.frameData = frameBuffer_;
memset(frame.frameData, 0x55, frame.size);

while (!stop_producer_) {
    // Produce frames
    timestamp = std::chrono::duration_cast<std::chrono::nanoseconds>(
        std::chrono::system_clock::now().time_since_epoch()).count() /
    DEFAULT_TIME_UNIT_IN_NANOS;
    frame.index = index++;
    frame.decodingTs = timestamp;
    frame.presentationTs = timestamp;

    // Key frame every 50th
    frame.flags = (frame.index % 50 == 0) ? FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    ...

    EXPECT_TRUE(kinesis_video_stream->putFrame(frame));
}
```

Note

Das vorstehende C++ Producer-Beispiel sendet einen Puffer mit Testdaten. In einer echten Anwendung sollten Sie Frame-Puffer und -Größe aus den Frame-Daten von einer Medienquelle (z. B. einer Kamera) abrufen.

Das Frame-Objekt enthält folgende Felder:

- Frame-Index. Dies sollte ein Wert sein, der gleichmäßig erhöht wird.
- Dem Frame zugeordnete Flags. Wenn der Encoder beispielsweise zum Generieren eines Keyframes konfiguriert wurde, würde diesem Frame das Flag `FRAME_FLAG_KEY_FRAME` zugewiesen.
- Zeitstempel dekodieren.
- Präsentationszeitstempel.
- Dauer des Frames (in 100-ns-Einheiten).
- Größe des Frames in Byte.
- Framedaten.

Weitere Informationen zum Format des Frames finden Sie unter [Kinesis Video Streams Data Model](#).

Einbinden eines KinesisVideoFrame in einen bestimmten Track von KinesisVideoStream

Sie können die PutFrameHelper Klasse verwenden, um Rahmendaten in eine bestimmte Spur einzufügen. Rufen Sie zunächst den getFrameData Buffer auf, um einen Zeiger auf einen der vorab zugewiesenen Puffer zu erhalten, um die KinesisVideoFrame Daten auszufüllen. Anschließend können Sie den putFrameMulti Track aufrufen, um den Wert KinesisVideoFrame zusammen mit dem booleschen Wert zu senden, der den Typ der Frame-Daten angibt. Verwenden Sie "true", wenn es sich um Videodaten handelt, oder "false", wenn der Frame Audiodaten enthält. Die putFrameMulti Track-Methode verwendet einen Warteschlangenmechanismus, um sicherzustellen, dass die MKV-Fragmente monoton steigende Frame-Zeitstempel beibehalten und sich zwei beliebige Fragmente nicht überlappen. Beispielsweise sollte der MKV-Zeitstempel des ersten Frames eines Fragments immer größer sein als der MKV-Zeitstempel des letzten Frames des vorherigen Fragments.

Der PutFrameHelper hat die folgenden Felder:

- Maximale Anzahl von AudioFrames in der Warteschlange.
- Maximale Anzahl von VideoFrames in der Warteschlange.
- Größe, die einem einzelnen Audioframe zugewiesen werden soll.
- Größe, die einem einzelnen Videobild zugewiesen werden soll.

Metriken und metrische Protokollierung

Das C++ Producer SDK umfasst Funktionen für Metriken und Metrikprotokollierung.

Sie können die getKinesisVideoMetrics und getKinesisVideoStreamMetrics API-Operationen verwenden, um Informationen über Kinesis Video Streams und Ihre aktiven Streams abzurufen.

Der folgende Code stammt aus der Datei kinesis-video-pic/src/client/include/com/amazonaws/kinesis/video/client/Include.h.

```
/**
 * Gets information about the storage availability.
 *
 * @param 1 CLIENT_HANDLE - the client object handle.
 * @param 2 PKinesisVideoMetrics - OUT - Kinesis Video metrics to be filled.
```



```
*
* @return Status of the function call.
*/
PUBLIC_API STATUS getKinesisVideoMetrics(CLIENT_HANDLE, PKinesisVideoMetrics);

/**
* Gets information about the stream content view.
*
* @param 1 STREAM_HANDLE - the stream object handle.
* @param 2 PStreamMetrics - Stream metrics to fill.
*
* @return Status of the function call.
*/
PUBLIC_API STATUS getKinesisVideoStreamMetrics(STREAM_HANDLE, PStreamMetrics);
```

Das von `getKinesisVideoMetrics` gefüllte `PClientMetrics`-Objekt folgende Informationen enthalten:

- `contentStoreSize`: Die Gesamtgröße des Inhaltsspeichers (des Speichers, der zum Speichern von Streaming-Daten verwendet wird) in Byte.
- `contentStoreAvailableGröße`: Der verfügbare Speicher im Inhaltsspeicher in Byte.
- `contentStoreAllocatedGröße`: Der zugewiesene Speicher im Inhaltsspeicher.
- `totalContentViewsGröße`: Der gesamte Speicher, der für die Inhaltsansicht verwendet wird. Die Inhaltsansicht besteht aus einer Reihe von Informationsindizes im Content Store.
- `totalFrameRate`: Die Gesamtzahl der Frames pro Sekunde in allen aktiven Streams.
- `totalTransferRate`: Die Gesamtzahl der Bits pro Sekunde (bps), die in allen Streams gesendet werden.

Das von `getKinesisVideoStreamMetrics` gefüllte `PStreamMetrics`-Objekt folgende Informationen enthalten:

- `currentViewDuration`: Der Unterschied in 100-ns-Einheiten zwischen dem Kopf der Inhaltsansicht (wenn Frames codiert sind) und der aktuellen Position (wenn Bilddaten an Kinesis Video Streams gesendet werden).
- `overallViewDuration`: Der Unterschied in 100-ns-Einheiten zwischen dem Anfang der Inhaltsansicht (wenn Frames codiert sind) und dem Ende (wenn Frames aus dem Speicher gelöscht werden, entweder weil der insgesamt zugewiesene Speicherplatz für die Inhaltsansicht überschritten wird

oder weil eine `PersistedAck` Nachricht von Kinesis Video Streams empfangen wird und Frames, von denen bekannt ist, dass sie persistent sind, geleert werden).

- `currentViewSize`: Die Größe der Inhaltsansicht in Byte vom Kopf (wenn Frames codiert sind) bis zur aktuellen Position (wenn Frames an Kinesis Video Streams gesendet werden).
- `overallViewSize`: Die Gesamtgröße der Inhaltsansicht in Byte.
- `currentFrameRate`: Die letzte gemessene Geschwindigkeit des Streams in Bildern pro Sekunde.
- `currentTransferRate`: Die letzte gemessene Rate des Streams in Byte pro Sekunde.

Abbruch

Wenn Sie die verbleibenden Bytes in einem Puffer senden und auf das ACK warten möchten, können Sie `stopSync` verwenden:

```
kinesis_video_stream->stopSync();
```

Oder Sie können `stop` aufrufen, um das Streamen zu beenden:

```
kinesis_video_stream->stop();
```

Nachdem Sie den Stream gestoppt haben, können Sie den Stream durch Aufrufen der folgenden API freigeben:

```
kinesis_video_producer_->freeStream(kinesis_video_stream);
```

Nächster Schritt

[the section called “Schritt 3: Ausführen und verifizieren Sie den Code”](#)

Schritt 3: Ausführen und verifizieren Sie den Code

Informationen zum Ausführen und Überprüfen des Codes für die [C++ Producer Library-Prozedur](#) finden Sie in den folgenden betriebssystemspezifischen Anweisungen:

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [Raspberry Pi OS](#)

Sie können den Traffic in Ihrem Stream überwachen, indem Sie sich die mit Ihrem Stream verknüpften Metriken in der CloudWatch Amazon-Konsole ansehen, z. `PutMedia.IncomingBytes` B.

Das C++ Producer SDK als GStreamer-Plugin verwenden

[GStreamer](#) ist ein beliebtes Medienframework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medien-Pipelines durch die Kombination modularer Plugins zu erstellen. Das Kinesis Video Streams GStreamer-Plugin optimiert die Integration Ihrer bestehenden GStreamer-Medienpipeline mit Kinesis Video Streams.

Weitere Informationen zur Verwendung des C++ Producer SDK als GStreamer-Plugin finden Sie unter [Beispiel: Kinesis Video Streams Producer SDK GStreamer-Plugin](#).

Verwendung des C++ Producer SDK als GStreamer-Plugin in einem Docker-Container

[GStreamer](#) ist ein beliebtes Medienframework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medien-Pipelines durch die Kombination modularer Plugins zu erstellen. Das Kinesis Video Streams GStreamer-Plugin optimiert die Integration Ihrer bestehenden GStreamer-Medienpipeline mit Kinesis Video Streams.

Darüber hinaus standardisiert die Verwendung von [Docker](#) zur Erstellung der GStreamer-Pipeline die Betriebsumgebung für Kinesis Video Streams, was die Erstellung und Ausführung der Anwendung optimiert.

Weitere Informationen zur Verwendung des C++ Producer SDK als GStreamer-Plugin in einem Docker-Container finden Sie unter [Ausführen des GStreamer-Elements in einem Docker-Container](#).

Verwenden der Protokollierung mit dem C++ Producer SDK

Sie konfigurieren die Protokollierung für C++ Producer SDK-Anwendungen in der Datei `kvs_log_configuration` im Ordner `kinesis-video-native-build`.

Das folgende Beispiel zeigt die erste Zeile der Standard-Konfigurationsdatei, mit der die Anwendung zum Schreiben von DEBUG-Level-Protokolleinträgen an die AWS Management Console konfiguriert wird:

```
log4cplus.rootLogger=DEBUG, KvsConsoleAppender
```

Für eine weniger ausführliche Protokollierung können Sie als Protokollierungsebene INFO festlegen.

Um die Anwendung so zu konfigurieren, dass sie Protokolleinträge in eine Protokolldatei schreibt, aktualisieren Sie die erste Zeile der Datei wie folgt:

```
log4cplus.rootLogger=DEBUG, KvsConsoleAppender, KvsFileAppender
```

Damit wird die Anwendung zum Schreiben von Protokolleinträgen in `kvs.log` im Ordner `kinesis-video-native-build/log` konfiguriert.

Um den Speicherort der Protokolldatei zu ändern, aktualisieren Sie die folgende Zeile mit dem neuen Pfad:

```
log4cplus.appender.KvsFileAppender.File=../log/kvs.log
```

Note

Wenn DEBUG-Level-Protokollierung in eine Datei geschrieben wird, kann die Protokolldatei den auf dem Gerät verfügbaren Speicherplatz schnell aufbrauchen.

Verwenden der C-Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte C Producer Library verwenden, um Anwendungscode zu schreiben, um Mediendaten von einem Gerät an einen Kinesis-Videostream zu senden.

Objektmodell

Die Kinesis Video Streams C Producer Library basiert auf einer gemeinsamen Komponente namens Platform Independent Codebase (PIC), die GitHub unter <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/> verfügbar ist. Das PIC enthält plattformunabhängige Geschäftslogik für

die grundlegenden Komponenten. Die Kinesis Video Streams C Producer Library umschließt PIC mit einer zusätzlichen API-Ebene, die szenario- und plattformsspezifische Rückrufe und Ereignisse ermöglicht. Die Kinesis Video Streams C Producer Library enthält die folgenden Komponenten, die auf PIC aufbauen:

- **Anbieter von Geräteinformationen** — Stellt die `DeviceInfo` Struktur bereit, die direkt an die PIC-API geliefert werden kann. Sie können eine Reihe von Anbietern konfigurieren, einschließlich eines für Anwendungsszenarien optimierten Anbieters, der den Inhaltsspeicher auf der Grundlage der Anzahl und Art der Streams, die Ihre Anwendung verarbeitet, und der Menge der erforderlichen Pufferung, die auf der Grundlage der verfügbaren RAM-Größe konfiguriert wird, optimieren kann.
- **Stream-Informationsanbieter** — Macht die `StreamInfo` Struktur verfügbar, die direkt an die PIC-API geliefert werden kann. Es gibt eine Reihe von Anbietern, die spezifisch für die Anwendungstypen und die gängigen Arten von Streaming-Szenarien sind. Dazu gehören Anbieter wie Video, Audio sowie Audio- und Video-Multitrack. Jedes dieser Szenarien hat Standardeinstellungen, die Sie an die Anforderungen Ihrer Anwendung anpassen können.
- **Callback-Anbieter** — Macht die `ClientCallbacks` Struktur verfügbar, die direkt an die PIC-API übergeben werden kann. Dazu gehören eine Reihe von Callback-Anbietern für Netzwerke (CURL-basierte API-Callbacks), Autorisierung (AWSAnmeldeinformationen-API) und Wiederholungsstreaming-on-Error-Callbacks. Die Callback Provider-API benötigt zur Konfiguration eine Reihe von Argumenten, wie z. B. die AWS-Region und die Autorisierungsinformationen. Dies erfolgt durch die Verwendung von IoT-Zertifikaten oder durch die Verwendung von `AWSAccessKeyIdSecretKey`, `SessionToken`. Sie können Callback Provider um benutzerdefinierte Callbacks erweitern, wenn Ihre Anwendung eine Weiterverarbeitung eines bestimmten Callback benötigt, um eine anwendungsspezifische Logik zu erreichen.
- **FrameOrderCoordinator** — Hilft bei der Audio- und Videosynchronisierung für mehrspurige Szenarien. Es hat ein Standardverhalten, das Sie an die spezifische Logik Ihrer Anwendung anpassen können. Es optimiert auch die Paketierung der Frame-Metadaten in der PIC-Frame-Struktur, bevor sie an die PIC-API der unteren Ebene gesendet werden. Für Nicht-Multitrack-Szenarien ist diese Komponente ein Übergang zur PIC-`putFrame`-API.

Die C-Bibliothek enthält die folgenden Objekte zum Senden von Daten an einen Kinesis-Videostream:

- **KinesisVideoClient** — Enthält Informationen über Ihr Gerät und unterhält Rückrufe, um über Kinesis Video Streams Streams-Ereignisse zu berichten.
- **KinesisVideoStream** — Stellt Informationen über die Parameter des Videostreams dar, wie z. B. Name, Datenaufbewahrungsdauer und Art des Medieninhalts.

Medien in den Stream bringen

Sie können die von der C-Bibliothek bereitgestellten Methoden verwenden (z. B. `PutKinesisVideoFrame`), um Daten in das `KinesisVideoStream` Objekt einzufügen. Die Bibliothek verwaltet dann den internen Status der Daten. Das umfasst beispielsweise folgende Aufgaben:

- Durchführen der Authentifizierung.
- Überwachen auf Netzwerklatenz. Wenn die Latenz zu hoch ist, verwirft die Bibliothek gegebenenfalls Frames.
- Überwachen des Status des laufenden Streamings.

Vorgehensweise: Verwenden des C-Producer-SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Streams-Client und die Medienquellen in einer C-Anwendung verwenden, um H.264-kodierte Videobilder an Ihren Kinesis-Videostream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- [Schritt 1: Herunterladen des C Producer-Library-Codes](#)
- [Schritt 2: Schreiben und untersuchen Sie den Code](#)
- [Schritt 3: Führen Sie den Code aus und überprüfen Sie ihn](#)

Voraussetzungen

- Anmeldeinformationen — Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei eingerichtet haben. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein.

Weitere Informationen finden Sie unter [AWSAnmeldeinformationen einrichten und Region für Entwicklung](#).

- Integration des Zertifikatsspeichers — Die Kinesis Video Streams Producer Library muss Vertrauen zu dem von ihr aufgerufenen Dienst aufbauen. Dies erfolgt durch die Validierung der Zertifizierungsstellen (CAs) im öffentlichen Zertifikatsspeicher. Bei Linux-basierten Modellen befindet sich dieser Speicher im Verzeichnis „`/etc/ssl`“.

Laden Sie das Zertifikat aus dem folgenden Speicherort in den Zertifikatspeicher:

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installieren Sie die folgenden Build-Abhängigkeiten für macOS:
 - [Autoconf 2.69](#) (Lizenz GPLv3+/Autoconf: GNU GPL Version 3 oder höher)
 - [CMake 3.7 oder 3.8](#)
 - [Pkg-Config](#)
 - [Flex 2.5.35 Apple \(flex-31\) oder höher](#)
 - [Bison 2.4](#) (GNU-Lizenz)
 - [Automake 1.15.1](#) (GNU-Lizenz)
 - GNU Libtool (Apple Inc. Version cctools-898)
 - xCode (macOS)/clang/gcc (xcode-select Version 2347)
 - Java Development Kit (JDK) (für Java-JNI-Kompilierung)
 - [Lib-Pkg](#)
- Installieren Sie die folgenden Build-Abhängigkeiten für Ubuntu (Antworten auf Versionsbefehle werden abgeschnitten):
 - Installieren Sie Git: `sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- Installieren Sie [CMake](#): `sudo apt-get install cmake`

```
$ cmake --version
cmake version 3.9.1
```

- Installieren Sie Libtool: `sudo apt-get install libtool`

```
2.4.6-2
```

- Installieren Sie libtool-bin: `sudo apt-get install libtool-bin`

```
$ libtool --version
libtool (GNU libtool) 2.4.6
Written by Gordon Matzigkeit, 1996
```

- Installieren Sie GNU Automake: `sudo apt-get install automake`

```
$ automake --version
automake (GNU automake) 1.15
```

- Installieren Sie GNU Bison: `sudo apt-get install bison`

```
$ bison -V
bison (GNU Bison) 3.0.4
```

- Installieren Sie G++: `sudo apt-get install g++`

```
g++ --version
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- Installieren Sie curl: `sudo apt-get install curl`

```
$ curl --version
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- Installieren Sie pkg-config: `sudo apt-get install pkg-config`

```
$ pkg-config --version
0.29.1
```

- Installieren Sie Flex: `sudo apt-get install flex`

```
$ flex --version
flex 2.6.1
```

- Installieren Sie OpenJDK: `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- Legen Sie die JAVA_HOME-Umgebungsvariable fest: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
- Führen Sie das Build-Skript aus: `./install-script`

Nächster Schritt

[Schritt 1: Herunterladen des C Producer-Library-Codes](#)

Schritt 1: Herunterladen des C Producer-Library-Codes

In diesem Abschnitt laden Sie die Low-Level-Bibliotheken herunter. Informationen zu den Voraussetzungen und weitere Details zu diesem Beispiel finden Sie unter [Verwenden der C-Producer-Bibliothek](#).

1. Erstellen Sie ein Verzeichnis und klonen Sie dann den Beispielquellcode aus dem GitHub Repository.

```
git clone --recursive https://github.com/awslabs/amazon-kinesis-video-streams-producer-c.git
```

Note

Wenn Sie es verabsäumen, `git clone` mit `--recursive` auszuführen, führen Sie `git submodule update --init` im Verzeichnis `amazon-kinesis-video-streams-producer-c/open-source` aus. Sie müssen auch `pkg-config`, `automake`, `CMake` und eine Build-Umgebung installieren.

Weitere Informationen finden Sie `README.md` in <https://github.com/awslabs/amazon-kinesis-video-streams-producer-c.git>.

2. Öffnen Sie den Code in einer integrierten Entwicklungsumgebung (Integrated Development Environment, IDE) Ihrer Wahl (z. B. [Eclipse](#)).

Nächster Schritt

[Schritt 2: Schreiben und untersuchen Sie den Code](#)

Schritt 2: Schreiben und untersuchen Sie den Code

In diesem Abschnitt untersuchen Sie den Code der Beispielanwendung `KvsVideoOnlyStreamingSample.c` im `samples` Ordner des <https://github.com/awslabs/amazon-kinesis-video-streams-producer-c-Repository>s auf GitHub. Sie haben diesen Code im

vorherigen Schritt heruntergeladen. Dieses Beispiel zeigt, wie Sie die C Producer Library verwenden, um H.264-codierte Videobilder innerhalb des Ordners an Ihren Kinesis-Videostream `samples/h264SampleFrames` zu senden.

Diese Beispielanwendung hat drei Abschnitte:

- Initialisierung und Konfiguration:
 - Initialisieren und Konfigurieren der plattformspezifischen Medien-Pipeline.
 - Initialisierung und Konfiguration von `KinesisVideoClient` und `KinesisVideoStream` für die Pipeline, Einstellung der Callbacks, Integration der szenariospezifischen Authentifizierung, Extrahieren und Senden privater Codec-Daten und Versetzen des Streams in den READY-Status.
- Hauptschleife:
 - Abrufen des Frame von der Medien-Pipeline mit dem Zeitstempel und Flags.
 - Senden des Frames an die `KinesisVideoStream`.
- Abbau:
 - Stoppen (Synchronisieren) `KinesisVideoStream`, Freigeben `KinesisVideoStream`, Befreien `KinesisVideoClient`.

Diese Beispielanwendung erfüllt die folgenden Aufgaben:

- Aufruf der `createDefaultDeviceInfo`-API zur Erstellung des `deviceInfo`-Objekts mit den Daten zum Gerät bzw. zur Speicherkonfiguration.

```
// default storage size is 128MB. Use setDeviceInfoStorageSize after create to change
// storage size.
CHK_STATUS(createDefaultDeviceInfo(&pDeviceInfo));
// adjust members of pDeviceInfo here if needed
pDeviceInfo->clientInfo.loggerLogLevel = LOG_LEVEL_DEBUG;
```

- Aufruf der `createRealtimeVideoStreamInfoProvider`-API zur Erstellung des `StreamInfo`-Objekts.

```
CHK_STATUS(createRealtimeVideoStreamInfoProvider(streamName,
DEFAULT_RETENTION_PERIOD, DEFAULT_BUFFER_DURATION, &pStreamInfo));
// adjust members of pStreamInfo here if needed
```

- Rufen Sie die `createDefaultCallbacksProviderWithAwsCredentials` API auf, um den Standard-Callback-Anbieter auf der Grundlage statischer AWS Anmeldeinformationen zu erstellen.

```
CHK_STATUS(createDefaultCallbacksProviderWithAwsCredentials(accessKey,
                                                            secretKey,
                                                            sessionToken,
                                                            MAX_UINT64,
                                                            region,
                                                            cacertPath,
                                                            NULL,
                                                            NULL,
                                                            FALSE,
                                                            &pClientCallbacks));
```

- Rufen Sie die `createKinesisVideoClient` API auf, um das `KinesisVideoClient` Objekt zu erstellen, das Informationen über den Speicher Ihres Geräts enthält und Callbacks verwaltet, um über Kinesis Video Streams Streams-Ereignisse zu berichten.

```
CHK_STATUS(createKinesisVideoClient(pDeviceInfo, pClientCallbacks, &clientHandle));
```

- Aufruf der `createKinesisVideoStreamSync`-API zur Erstellung des `KinesisVideoStream`-Objekts.

```
CHK_STATUS(createKinesisVideoStreamSync(clientHandle, pStreamInfo, &streamHandle));
```

- Einrichten eines Beispiel-Frame und Aufruf der `PutKinesisVideoFrame`-API zur Übermittlung dieses Frame an das `KinesisVideoStream`-Objekt.

```
// setup sample frame
MEMSET(frameBuffer, 0x00, frameSize);
frame.frameData = frameBuffer;
frame.version = FRAME_CURRENT_VERSION;
frame.trackId = DEFAULT_VIDEO_TRACK_ID;
```

```
frame.duration = HUNDREDS_OF_NANOS_IN_A_SECOND / DEFAULT_FPS_VALUE;
frame.decodingTs = defaultGetTime(); // current time
frame.presentationTs = frame.decodingTs;

while(defaultGetTime() > streamStopTime) {
    frame.index = frameIndex;
    frame.flags = fileIndex % DEFAULT_KEY_FRAME_INTERVAL == 0 ?
FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    frame.size = sizeof(frameBuffer);

    CHK_STATUS(readFrameData(&frame, frameFilePath));

    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));
    defaultThreadSleep(frame.duration);

    frame.decodingTs += frame.duration;
    frame.presentationTs = frame.decodingTs;
    frameIndex++;
    fileIndex++;
    fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
}
```

- Abbau:

```
CHK_STATUS(stopKinesisVideoStreamSync(streamHandle));
CHK_STATUS(freeKinesisVideoStream(&streamHandle));
CHK_STATUS(freeKinesisVideoClient(&clientHandle));
```

Nächster Schritt

[Schritt 3: Führen Sie den Code aus und überprüfen Sie ihn](#)

Schritt 3: Führen Sie den Code aus und überprüfen Sie ihn

Gehen Sie folgendermaßen vor, um den Code für die [Vorgehensweise zur C-Producer-Bibliothek](#) auszuführen und zu verifizieren:

1. Führen Sie die folgenden Befehle aus, um ein `build` Verzeichnis in Ihrem [heruntergeladenen C-SDK](#) zu erstellen, und starten Sie es `cmake` von dort aus:

```
mkdir -p amazon-kinesis-video-streams-producer-c/build;
cd amazon-kinesis-video-streams-producer-c/build;
cmake ..
```

Sie können die folgenden Optionen an `cmake ..` übergeben:

- `-DBUILD_DEPENDENCIES`- ob abhängige Bibliotheken aus dem Quellcode erstellt werden sollen.
- `-DBUILD_TEST=TRUE`- Einheiten- und Integrationstests erstellen. Könnte nützlich sein, um die Unterstützung für Ihr Gerät zu bestätigen.

```
./tst/webrtc_client_test
```

- `-DCODE_COVERAGE`- Berichterstattung zur Berichterstattung aktivieren.
 - `-DCOMPILER_WARNINGS`- aktiviert alle Compiler-Warnungen.
 - `-DADDRESS_SANITIZER`- baue mitAddressSanitizer.
 - `-DMEMORY_SANITIZER`- baue mitMemorySanitizer.
 - `-DTHREAD_SANITIZER`- baue mitThreadSanitizer.
 - `-DUNDEFINED_BEHAVIOR_SANITIZER`- baue mitUndefinedBehaviorSanitizer.
 - `-DALIGNED_MEMORY_MODEL` – Build für Geräte mit ausgerichtetem Memory Model Der Standardwert ist OFF.
2. Navigieren Sie zu dem `build` Verzeichnis, das Sie gerade mit dem vorherigen Schritt erstellt haben, und führen Sie den Vorgang aus, `make` um das WebRTC C SDK und die bereitgestellten Beispiele zu erstellen.

```
make
```

3. Die Beispielanwendung `kinesis_video_cproducer_video_only_sample` sendet h.264-codierte Videobilder innerhalb des Ordners `samples/h264SampleFrames` an Kinesis Video Streams. Mit dem folgenden Befehl werden die Videobilder zehn Sekunden lang in einer Schleife an Kinesis Video Streams gesendet:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Wenn Sie H.264-codierte Frames aus einem anderen Ordner senden möchten (z. B. `MyH264FramesFolder`), führen Sie das Beispiel mit den folgenden Argumenten aus:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10 MyH264FramesFolder
```

4. Definieren Sie die C-Definitionen `HEAP_DEBUG` und `LOG_STREAMING`, indem Sie die entsprechenden Zeilen in `CMakeList.txt` auskommentieren, um ausführliche Protokolle zu aktivieren.

Sie können den Fortschritt der Testsuite in der Debug-Ausgabe in der IDE überwachen. Sie können den Traffic in Ihrem Stream auch überwachen, indem Sie sich die mit Ihrem Stream verknüpften Metriken in der CloudWatch Amazon-Konsole ansehen, z. `PutMedia.IncomingBytes` B.

Note

Da der Testeinsatz nur Frames mit leeren Bytes sendet, zeigt die Konsole die Daten nicht als Videostream an.

Verwenden des C++ Producer SDK auf Raspberry Pi

Raspberry Pi ist ein kleiner, kostengünstiger Computer, mit dem Computergrundkenntnisse in der Programmierung beigebracht und erlernt werden können. In diesem Tutorial wird beschrieben, wie Sie das Amazon Kinesis Video Streams C++ Producer SDK auf einem Raspberry Pi-Gerät einrichten und verwenden können. Die Verfahrensschritte umfassen auch, wie Sie die Installation mithilfe der GStreamer-Demoanwendung überprüfen.

Themen

- [Voraussetzungen](#)
- [Erstellen eines IAM-Benutzers mit der Berechtigung zum Schreiben in Kinesis Video Streams](#)
- [Verbinden Ihres Raspberry Pi mit Ihrem WLAN-Netzwerk](#)

- [Remote-Verbindung zu Ihrem Raspberry Pi herstellen](#)
- [Konfigurieren der Raspberry Pi-Kamera](#)
- [Installieren von Softwarevoraussetzungen](#)
- [Herunterladen und Erstellen des Kinesis Video Streams C++ Producer SDK](#)
- [Streamen Sie Videos in Ihren Kinesis-Videostream und sehen Sie sich den Livestream an](#)

Voraussetzungen

Stellen Sie vor dem Einrichten des C++ Producer SDK auf Ihrem Raspberry Pi sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Ein Raspberry Pi-Gerät mit der folgenden Konfiguration:
 - Board-Version: 3 Modell B oder höher.
 - Angeschlossenes Kamera-Modul.
 - SD-Karte mit einer Kapazität von mindestens 8 GB.
 - Das Betriebssystem Raspbian (Kernelversion 4.9 oder höher) ist installiert. Sie können das neueste Raspberry Pi OS-Image (früher Raspbian genannt) von der [Raspberry Pi-Website](#) herunterladen. Befolgen Sie die Raspberry Pi-Anweisungen zur [Installation des heruntergeladenen Abbildes auf einer SD-Karte](#).
- Ein AWS-Konto mit einem Kinesis-Videostream. Weitere Informationen finden Sie unter [Erste Schritte mit Kinesis-Videostreams](#).

Note

Das C++ Producer SDK verwendet standardmäßig die `us-west-2` Region USA West (Oregon) (). Um den Standard zu verwenden, AWS-Region erstellen Sie Ihren Kinesis-Videostream in der Region USA West (Oregon).

Führen Sie einen der folgenden Schritte aus, um eine andere Region für Ihren Kinesis-Videostream zu verwenden:

- Legen Sie für Ihre Region (z. B. `us-east-1`) die folgende Umgebungsvariable fest:

```
export AWS_DEFAULT_REGION=us-east-1
```

Erstellen eines IAM-Benutzers mit der Berechtigung zum Schreiben in Kinesis Video Streams

Falls noch nicht geschehen, richten Sie einen AWS Identity and Access Management (IAM)-Benutzer mit Berechtigungen zum Schreiben in einen Kinesis-Videostream ein.

Diese Verfahren sollen Ihnen helfen, schnell mit der Verwendung eines AWS Zugriffsschlüsselpaars zu beginnen. Geräte können X.509-Zertifikate verwenden, um eine Verbindung zu herzustellen AWS IoT. [the section called “Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von AWS IoT”](#) Weitere Informationen zur Konfiguration Ihres Geräts für die Verwendung der zertifikatsbasierten Authentifizierung finden Sie unter .

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Klicken Sie im Navigationsmenü links auf Users (Benutzer).
3. Um einen neuen Benutzer zu erstellen, wählen Sie Add user (Benutzer hinzufügen).
4. Geben Sie einen beschreibenden Benutzernamen für den Benutzer an, wie etwa **kinesis-video-raspberry-pi-producer**.
5. Wählen Sie unter Access type (Zugriffstyp) Programmatic access (Programmgesteuerte Zugriff).
6. Wählen Sie Weiter: Berechtigungen aus.
7. Wählen Sie unter Berechtigungen für kinesis-video-raspberry-pi-Produzent festlegen die Option Vorhandene Richtlinien direkt anfügen aus.
8. Wählen Sie Richtlinie erstellen aus. Die Seite Create policy (Richtlinie erstellen) wird in einer neuen Webbrowser-Registerkarte geöffnet.
9. Wählen Sie den Tab JSON.
10. Kopieren Sie die folgende JSON-Richtlinie und fügen Sie sie in den Textbereich ein. Diese Richtlinie erteilt Ihrem Benutzer die Berechtigung zum Erstellen und Schreiben von Daten in Kinesis-Videostreams.

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



```
    "kinesisvideo:GetDataEndpoint",
    "kinesisvideo:PutMedia"
  ],
  "Resource": [
    "*"
  ]
}]
}
```

11. Wählen Sie Richtlinie prüfen.
12. Geben Sie einen Namen für die Richtlinie ein, z. B. **kinesis-video-stream-write-policy**.
13. Wählen Sie Richtlinie erstellen aus.
14. Kehren Sie zur Registerkarte Add user (Benutzer hinzufügen) im Browser zurück. Wählen Sie Refresh (Aktualisieren).
15. Geben Sie in das Suchfeld den Namen der von Ihnen erstellten Richtlinie ein.
16. Aktivieren Sie das Kontrollkästchen neben der neuen Richtlinie in der Liste.
17. Klicken Sie auf Next: Review (Weiter: Prüfen).
18. Wählen Sie Create user aus.
19. Die Konsole zeigt die Access key ID (Zugriffsschlüssel-ID) für den neuen Benutzer an. Wählen Sie Anzeigen, um einen Wert unter Secret access key (Geheimer Zugriffsschlüssel) anzuzeigen. Notieren Sie diese Werte. Sie sind erforderlich, wenn Sie die Anwendung konfigurieren.

Verbinden Ihres Raspberry Pi mit Ihrem WLAN-Netzwerk

Sie können den Raspberry Pi im Headless--Modus, d. h. ohne Anschluss von Tastatur, Monitor oder Netzwerkkabel verwenden. Fahren Sie bei Anschluss eines Monitors und einer Tastatur mit [Konfigurieren der Raspberry Pi-Kamera](#).

1. Erstellen Sie auf dem Computer ein Datei mit dem Namen `wpa_supplicant.conf`.
2. Kopieren Sie den folgenden Text und fügen Sie ihn in die `wpa_supplicant.conf` Datei ein:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="Your Wi-Fi SSID"
  scan_ssid=1
```

```
key_mgmt=WPA-PSK
psk="Your Wi-Fi Password"
}
```

Ersetzen Sie die Werte `ssid` und `psk` durch die Angaben für Ihr WLAN-Netzwerk.

3. Kopieren Sie die Datei `wpa_supplicant.conf` auf die SD-Karte. Sie muss in das Stammverzeichnis des Volumes `boot` kopiert werden.
4. Legen Sie die SD-Karte in den Raspberry Pi ein und schalten Sie das Gerät ein. Es wird mit Ihrem WLAN-Netzwerk verbunden, und SSH wird aktiviert.

Remote-Verbindung zu Ihrem Raspberry Pi herstellen

Sie können im Headless-Modus eine Fernverbindung mit dem Raspberry Pi herstellen. Wenn Sie Ihren Raspberry Pi mit angeschlossenem Monitor und angeschlossener Tastatur verwenden, fahren Sie mit [Konfigurieren der Raspberry Pi-Kamera](#).

1. Führen Sie vor dem Herstellen einer Fernverbindung mit dem Raspberry Pi-Gerät einen der folgenden Schritte durch, um seine IP-Adresse zu bestimmen:
 - Wenn Sie Zugriff auf den WLAN-Router des Netzwerks haben, zeigen Sie die angeschlossenen WLAN-Geräte an. Suchen Sie nach dem Gerät mit dem Namen `Raspberry Pi`, um die IP-Adresse des Geräts ausfindig zu machen.
 - Wenn Sie keinen Zugriff auf den WLAN-Router des Netzwerks haben, können Sie mit anderer Software nach Geräten in Ihrem Netzwerk suchen. [Fing](#) ist eine beliebte Anwendung, die sowohl für Android- als auch iOS-Geräte verfügbar ist. Sie können mit der kostenlosen Version dieser Anwendung nach den IP-Adressen von Geräten in Ihrem Netzwerk suchen.
2. Wenn Sie die IP-Adresse des Raspberry Pi-Geräts kennen, können Sie zum Herstellen einer Verbindung jede Terminal-Anwendung verwenden.
 - Verwenden Sie unter macOS oder Linux `ssh`:

```
$ ssh pi@<IP address>
```

- Verwenden Sie unter Windows [PuTTY](#), einen kostenlosen SSH-Client für Windows.

Der Benutzername und das Passwort für eine neue Installation von Raspbian lauten **pi** und **raspberry**. Wir empfehlen Ihnen, [das Standardpasswort zu ändern](#).

Konfigurieren der Raspberry Pi-Kamera

Gehen Sie wie folgt vor, um die Raspberry Pi-Kamera so zu konfigurieren, dass Videos vom Gerät an einen Kinesis-Videostrom gesendet werden.

1. Öffnen Sie einen Editor, um die Datei `modules` mit dem folgenden Befehl zu aktualisieren:

```
$ sudo nano /etc/modules
```

2. Fügen Sie die folgende Zeile an das Ende der Datei an, sofern dort noch nicht vorhanden:

```
bcm2835-v4l2
```

3. Speichern Sie die Datei und beenden Sie den Texteditor (Strg+X).
4. Starten Sie den Raspberry Pi neu:

```
$ sudo reboot
```

5. Wenn das Gerät neu gestartet wird, verbinden Sie es erneut über die Terminalanwendung, sofern Sie eine Fernverbindung herstellen.
6. Öffnen `raspi-config`:

```
$ sudo raspi-config
```

7. Wählen Sie Verbindungsoptionen, Legacy-Kamera aus. In älteren Builds des Raspbian-Betriebssystems befindet sich diese Menüoption möglicherweise unter Verbindungsoptionen, Kamera .

Aktivieren Sie die Kamera, sofern noch nicht geschehen, und führen Sie einen Neustart durch, wenn Sie dazu aufgefordert werden.

8. Vergewissern Sie sich durch Eingabe des folgenden Befehls, dass die Kamera funktioniert:

```
$ raspistill -v -o test.jpg
```

Wenn Ihre Kamera korrekt konfiguriert ist, erfasst dieser Befehl ein Bild von der Kamera, speichert es in einer Datei mit dem Namen `test.jpg` und zeigt Informationsmeldungen an.

Installieren von Softwarevoraussetzungen

Das C++ Producer SDK erfordert die Installation der folgenden Softwarevoraussetzungen auf Raspberry Pi.

1. Aktualisieren Sie die Paketliste und installieren Sie die Bibliotheken, die zum Erstellen des SDK erforderlich sind. Geben Sie die folgenden Befehle ein:

```
$ sudo apt update
$ sudo apt install -y \
  automake \
  build-essential \
  cmake \
  git \
  gstreamer1.0-plugins-base-apps \
  gstreamer1.0-plugins-bad \
  gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-ugly \
  gstreamer1.0-tools \
  gstreamer1.0-omx-generic \
  libcurl4-openssl-dev \
  libgstreamer1.0-dev \
  libgstreamer-plugins-base1.0-dev \
  liblog4cplus-dev \
  libssl-dev \
  pkg-config
```

2. Kopieren Sie die folgende PEM-Datei zu `/etc/ssl/cert.pem`:

```
$ sudo curl https://www.amazontrust.com/repository/AmazonRootCA1.pem -o /etc/ssl/
AmazonRootCA1.pem
$ sudo chmod 644 /etc/ssl/AmazonRootCA1.pem
```

Herunterladen und Erstellen des Kinesis Video Streams C++ Producer SDK

Sie können das Kinesis Video Streams C++ Producer SDK mit dem folgenden Verfahren herunterladen und erstellen. Mit diesem Ansatz dauert das Bauen, je nach Netzwerkverbindung und Prozessorgeschwindigkeit, länger.

1. Laden Sie das SDK herunter. Type:

```
$ git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp.git
```

2. Bereiten Sie ein Build-Verzeichnis vor. Type:

```
$ mkdir -p amazon-kinesis-video-streams-producer-sdk-cpp/build  
$ cd amazon-kinesis-video-streams-producer-sdk-cpp/build
```

3. Erstellen Sie das SDK und Beispielanwendungen. Je nachdem, auf welchem Modell von Raspberry Pi Sie bauen, kann es einige Stunden dauern, bis es zum ersten Mal ausgeführt wird:

```
$ cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=FALSE  
$ make
```

Streamen Sie Videos in Ihren Kinesis-Videostream und sehen Sie sich den Livestream an

1. Zur Ausführung der Beispielanwendung benötigen Sie die folgenden Informationen:
 - Name des Streams, der von Ihnen im Abschnitt [Voraussetzungen](#) erstellt wurde.
 - Die Konto-Anmeldeinformationen (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel), die von Ihnen unter [Erstellen eines IAM-Benutzers mit der Berechtigung zum Schreiben in Kinesis Video Streams](#) erstellt wurden
2. Führen Sie die Beispielanwendung mit den folgenden Befehlen aus. Ersetzen Sie die Platzhalter durch Werte für Ihre Umgebung.

```
$ export GST_PLUGIN_PATH=Directory Where You Cloned the SDK/amazon-kinesis-video-streams-producer-sdk-cpp/build  
$ export AWS_DEFAULT_REGION=AWS Region i.e. us-east-1  
$ export AWS_ACCESS_KEY_ID=Access Key ID  
$ export AWS_SECRET_ACCESS_KEY=Secret Access Key  
$ ./kvs_gstreamer_sample Your Stream Name
```

3. Wenn die Beispielanwendung mit einem `library not found` Fehler beendet wird, geben Sie den folgenden Befehl ein, um zu überprüfen, ob das Projekt korrekt mit seinen Open-Source-Abhängigkeiten verknüpft ist:

```
$ gst-inspect-1.0 kvssink
```

4. Öffnen Sie die [Kinesis-Video-Streams-Konsole](#).
5. Wählen Sie unter Stream name (Stream-Name) den Namen des von Ihnen erstellten Streams aus.

Der vom Raspberry Pi gesendete Videostream erscheint in der Konsole.

Wenn der Stream abgespielt wird, können Sie mit den folgenden Funktionen der Kinesis-Video-Streams-Konsole experimentieren:

- Führen Sie mit den Navigationssteuerelementen im Bereich Video preview (Video-Vorschau) einen Rück- oder Vorlauf des Streams durch.
- Beachten Sie im Bereich Stream info (Stream-Info) den Codec, die Auflösung und die Bitrate des Streams. Die Werte für Auflösung und Bitrate werden auf dem Raspberry Pi absichtlich niedrig eingestellt, um die Bandbreitennutzung für dieses Tutorial zu minimieren. Um die Amazon-CloudWatch Metriken anzuzeigen, die für Ihren Stream erstellt werden, wählen Sie Stream-Metriken in anzeigen. CloudWatch
- Beachten Sie unter Data retention period (Aufbewahrungszeitraum), dass der Videostream einen Tag lang aufbewahrt wird. Sie können diesen Wert bearbeiten und auf No data retention (Keine Datenaufbewahrung) einstellen oder einen Wert von einem Tag bis mehrere Jahre festlegen.

Beachten Sie unter serverseitiger Verschlüsselung, dass Ihre Daten im Ruhezustand mit einem Schlüssel verschlüsselt werden, der von AWS Key Management Service (AWS KMS) verwaltet wird.

Producer SDK-Referenz

Dieser Abschnitt enthält Limits, Fehlercodes und andere Referenzinformationen für die [Produzentenbibliotheken von Kinesis Video Streams](#).

Themen

- [Beschränkungen des Producer-SDK](#)
- [Fehlercodereferenz](#)
- [Network Abstraction Layer \(NAL\) Adaptation Flag-Referenz](#)

- [SDK-Strukturen für Hersteller](#)
- [Kinesis-Videostream-Strukturen](#)
- [SDK-Rückrufe von Produzenten](#)

Beschränkungen des Producer-SDK

Die folgende Tabelle enthält die aktuellen Limits für Werte in den [Producer-Bibliotheken](#).

Note

Bevor Sie diese Werte festlegen, müssen Sie Ihre Eingabe überprüfen. Das SDK prüft diese Limits nicht. Wenn sie überschritten werden, treten Laufzeitfehler auf.

Wert	Limit	Hinweise
Max stream count (Max. Stream-Anzahl)	128	Die maximale Anzahl von Streams, die ein Producer-Objekt erstellen kann. Dies ist ein weiches Limits (Sie können eine Erhöhung beantragen). Es garantiert, dass der Producer nicht versehentlich rekursiv Streams erstellt.
Max device name length (Max. Länge des Gerätenamens)	128 Zeichen	
Max tag count (Max. Tag-Anzahl)	50 pro Stream	
Max stream name length (Max. Länge des Stream-Namens)	256 Zeichen	

Wert	Limit	Hinweise
Min storage size (Min. Speichergröße)	10 MiB = 10 x 1024 x 1024 Byte	
Max storage size (Max. Speichergröße)	10 GiB = 10 x 1024 x 1024 x 1024 Byte	
Max root directory path length (Max. Länge des Stammverzeichnispfades)	4,096 Zeichen	
Max auth info length (Max. Länge der Authentifizierungsinformationen)	10,000 Bytes	
Max URI string length (Max. Länge der URI-Zeichenfolge)	10,000 Zeichen	
Max tag name length (Max. Länge des Tag-Namens)	128 Zeichen	
Max tag value length (Max. Länge des Tag-Wertes)	1,024 Zeichen	
Min security token period (Min. Zeitraum des Sicherheitstokens)	30 Sekunden	
Security token grace period (Übergangsfrist des Sicherheitstokens)	40 Minuten	Wenn die angegebene Dauer länger ist, ist sie auf diesen Wert begrenzt.
Aufbewahrungszeitraum	0 oder mehr als eine Stunde	0 bedeutet keine Aufbewahrung.

Wert	Limit	Hinweise
Min cluster duration (Min. Cluster-Dauer)	1 Sekunde	Der Wert wird in Einheiten von 100 ns angegeben. Dabei handelt es sich um den SDK-Standard.
Max cluster duration (Max. Cluster-Dauer)	30 Sekunden	Der Wert wird in Einheiten von 100 ns angegeben. Dabei handelt es sich um den SDK-Standard. Die Backend-API kann eine kürzere Cluster-Dauer erzwingen.
Max fragment size (Max. Fragmentgröße)	50 MB	Weitere Informationen finden Sie unter Service Quotas für Kinesis Video Streams .
Max fragment duration (Max. Fragmentdauer)	20 Sekunden	Weitere Informationen finden Sie unter Service Quotas für Kinesis Video Streams .
Max connection duration (Max. Verbindungsdauer)	45 Minuten	Die Backend schließt die Verbindung nach Verstreichen dieser Frist. Das SDK rotiert das Token und baut innerhalb dieses Zeitraums eine neue Verbindung auf.
Max ACK segment length (Max. ACK-Segmentlänge)	1,024 Zeichen	Maximale Segmentlänge der an die ACK-Parser-Funktionen gesendeten Empfangsbestätigung.
Max content type string length (Max. Zeichenfolgenlänge des Inhaltstyps)	128 Zeichen	

Wert	Limit	Hinweise
Max codec ID string length (Max. Zeichenfolgenlänge der Codec-ID)	32 Zeichen	
Max track name string length (Max. Zeichenfolgenlänge des Tracknamens)	32 Zeichen	
Max codec private data length (Max. Länge privater Codec-Daten)	1 MiB = 1 x 1024 x 1024 Byte	
Min timecode scale value length (Min. Länge des Timecode-Skalierungswertes)	100 ns	Der minimale Timecode-Skalierungswert zur Darstellung der Frame-Zeitstempel im resultierenden MKV-Cluster. Der Wert wird in Inkrementen von 100 ns angegeben. Dabei handelt es sich um den SDK-Standard.
Max timecode scale value length (Max. Länge des Timecode-Skalierungswertes)	1 Sekunde	Der maximale Timecode-Skalierungswert zur Repräsentation der Frame-Zeitstempel im resultierenden MKV-Cluster. Der Wert wird in Inkrementen von 100 ns angegeben. Dabei handelt es sich um den SDK-Standard.
Min content view item count (Min. Elementanzahl der Inhaltsansicht)	10	

Wert	Limit	Hinweise
Min buffer duration (Min. Dauer der Pufferung)	20 Sekunden	Der Wert wird in Inkrementen von 100 ns angegeben. Dabei handelt es sich um den SDK-Standard.
Max update version length (Max. Länge der Aktualisierungsversion)	128 Zeichen	
Max ARN length (Max. ARN-Länge)	1024 Zeichen	
Max fragment sequence length (Max. Länge der Fragmentsequenz)	128 Zeichen	
Maximaler Aufbewahrungszeitraum	10 Jahre	

Fehlercodereferenz

Dieser Abschnitt enthält Fehler- und Statuscodeinformationen für das [Producer-Bibliotheken](#).

Weitere Informationen zu Lösungen für gängige Probleme finden Sie unter [Problembhebung bei Kinesis Video Streams](#).

Themen

- [Von PutFrame Callbacks — Platform Independent Code \(PIC\) zurückgegebene Fehler und Statuscodes](#)
- [Von PutFrame Callbacks zurückgegebene Fehler und Statuscodes — C-Producer-Bibliothek](#)

Von PutFrame Callbacks — Platform Independent Code (PIC) zurückgegebene Fehler und Statuscodes

Die folgenden Abschnitte enthalten Fehler- und Statusinformationen, die von Callbacks für den PutFrame Vorgang innerhalb des Platform Independent Code (PIC) zurückgegeben werden.

Themen

- [Von der Client-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Duration-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der gemeinsamen Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Heap-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der MKVGen-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Trace-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Utils-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der View-Bibliothek zurückgegebene Fehler- und Statuscodes](#)

Von der Client-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Kinesis Video Streams Client Streams-Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000001	STATUS_MAX_STREAM_COUNT	Die maximale Anzahl von Streams wurde erreicht.	Geben Sie in DeviceInfo o wie unter Beschränkungen des Producer-SDK angegeben eine größere maximale Anzahl von Streams ein.
0x52000002	STATUS_MIN_STREAM_COUNT	Fehler bei minimaler Anzahl von Streams.	Geben Sie die maximale Anzahl von Streams an, die größer als Null sind. DeviceInfo

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	Ungültige Länge des Gerätenamens.	Beziehen Sie sich auf die maximale Länge von Gerätenamen in Zeichen, die in angegeben ist Beschränkungen des Producer-SDK .
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	Ungültige Version der DeviceInfo - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x52000005	STATUS_MAXIMUM_TAG_COUNT	Die maximale Tag-Anzahl wurde erreicht.	Beziehen Sie sich auf die aktuelle maximale Anzahl von Tags, die in angegeben ist Beschränkungen des Producer-SDK .
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	Ungültige Version der Callbacks - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	Ungültige Version der StreamInfo - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	Ungültige Länge des Stream-Namens.	Beziehen Sie sich auf die maximale Länge des Stream-Namens in Zeichen, die in angegeben ist Beschränkungen des Producer-SDK .

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200000a	STATUS_INVALID_STORAGE_SIZE	Es wurde eine ungültige Speichergröße angegeben.	Die Speichergröße in Byte muss innerhalb der unter Beschränkungen des Producer-SDK angegebenen Limits liegen.
0x5200000b	STATUS_INVALID_DIRECTORY_LENGTH	Ungültige Länge der Stammverzeichnis-Zeichenfolge.	Beziehen Sie sich auf die maximale Pfadlänge des Stammverzeichnisses, die in angegeben ist Beschränkungen des Producer-SDK .
0x5200000c	STATUS_INVALID_SPILL_RATIO	Ungültiges Überlaufverhältnis.	Geben Sie die Überlaufquote als Prozentsatz zwischen 0 und 100 an.
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Ungültige Version der StorageInfo-Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200000e	STATUS_INVALID_STREAM_STATE	Der Stream befindet sich in einem Zustand, in dem die aktuelle Operation nicht zulässig ist.	In den meisten Fällen tritt dieser Fehler auf, wenn das SDK nicht den Status erreicht, den es für die Ausführung des angeforderten Vorgangs benötigt. Dies tritt z. B. auf, wenn der <code>GetStream ingEndpoint -API-</code> Aufruf fehlschlägt und die Client-Anwendung ihn ignoriert und weiterhin Frames in den Stream leitet.
0x5200000f	STATUS_SERVICE_CALLBACKS_MISSING	In der <code>Callbacks -Struktur</code> fehlen Funktions-Einstiegs punkte für einige obligatorische Funktionen.	Stellen Sie sicher, dass die obligatorischen Callbacks in der Client-Anwendung implementiert sind. Dieser Fehler tritt nur bei PIC-Clients (Platform Independent Code) auf. C++ und andere High-Level-Wrapper erfüllen diese Aufrufe.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000010	STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR	Nicht autorisiert.	Überprüfen Sie das Sicherheitstoken, das Zertifikat, die Integration und den Ablauf des Sicherheitstokens. Stellen Sie sicher, dass dem Token die richtigen Rechte zugewiesen sind. Stellen Sie bei den Kinesis Video Streams Streams-Beispielanwendungen sicher, dass die Umgebungsvariable richtig eingestellt ist.
0x52000011	STATUS_DESCRIBE_STREAM_CALL_FAILED	DescribeStream -API-Fehler.	Dieser Fehler wird nach dem fehlgeschlagenen DescribeStream -API-Wiederholungsversuch zurückgegeben. Der PIC-Client gibt diesen Fehler zurück, nachdem er den erneuten Versuch beendet hat.
0x52000012	STATUS_INVALID_DESCRIPTOR_RESPONSE	Ungültige DescribeStreamResponse -Struktur.	Die an DescribeStreamResultEvent übergebene Struktur ist entweder null oder enthält ungültige Elemente wie einen fehlenden Amazon-Ressourcennamen (ARN).

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	Es wird gerade versucht, den Stream zu löschen.	Durch den Versuch, den Stream zu löschen, wurde ein API-Fehler verursacht. Stellen Sie sicher, dass keine anderen Prozesse versuchen, den Stream zu löschen, während der Stream verwendet wird.
0x52000014	STATUS_SERVICE_CALL_INVALID_ARGUMENT_ERROR	Für den Dienstauf ruf wurden ungültige Argumente angegeben.	Das Backend gibt diesen Fehler zurück, wenn ein Serviceaufrufargument nicht gültig ist oder wenn das SDK auf einen Fehler stößt, den es nicht interpretieren kann.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	Die Gerät wurde nicht gefunden.	Stellen Sie sicher, dass das Gerät nicht gelöscht wird, während es verwendet wird.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	Die Gerät wurde nicht bereitgestellt.	Stellen Sie sicher, dass das Gerät bereitgestellt wurde.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	Der Dienst gibt zurück, dass eine generische Ressource nicht gefunden wurde.	Dieser Fehler tritt auf, wenn der Dienst die Ressource (z. B. einen Stream) nicht finden kann. Dies kann in verschiedenen Kontexten unterschiedliche Bedeutungen haben. Die wahrscheinlichste Ursache ist aber die Nutzung von APIs, bevor der Stream erstellt wurde. Durch die Verwendung des SDK wird bestätigt, dass der Stream zuerst erstellt wurde.
0x52000018	STATUS_INVALID_AUTH_LEN	Ungültige Länge der Authentifizierungsinformationen.	Die aktuellen Werte werden unter Beschränkungen des Producer-SDK angegeben.
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	Der CreateStream-API-Aufruf ist fehlgeschlagen.	Weitere Informationen zum Grund für das Fehlschlagen der Operation finden Sie in der Fehlerzeichenfolge.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	Der GetStreamingToken-API-Aufruf ist fehlgeschlagen.	Weitere Informationen zum Grund für das Fehlschlagen der Operation finden Sie in der Fehlerzeichenfolge.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	Der GetStreamingEndpoint-API-Aufruf ist fehlgeschlagen.	Weitere Informationen zum Grund für das Fehlschlagen der Operation finden Sie in der Fehlerzeichenfolge.
0x5200002c	STATUS_INVALID_URI_LEN	Von der GetStreamingEndpoint-API wurde eine URI-Zeichenfolge ungültiger Länge zurückgegeben.	Die aktuellen maximalen Werte werden unter Beschränkungen des Producer-SDK angegeben.
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	Der PutMedia-API-Aufruf ist fehlgeschlagen.	Weitere Informationen zum Grund für das Fehlschlagen der Operation finden Sie in der Fehlerzeichenfolge.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200002e	STATUS_STREAM_OUT_OF_MEMORY	Im Inhaltsspeicher ist nicht mehr genügend Speicherplatz vorhanden.	Der Inhaltsspeicher wird von den Streams gemeinsam genutzt und sollte über genügend Kapazität zum Speichern der maximalen Dauer aller Streams + ~20 % (einschließlich Defragmentierung) verfügen. Es ist wichtig, dass der Speicher nicht überläuft. Wählen Sie Werte für die maximale Dauer pro Stream, die der kumulativen Speicherröße und den Latenztoleranzen entsprechen. Wir empfehlen, die Frames zu löschen, sobald sie aus der Inhaltsansicht herausfallen, anstatt sie einfach nur zu platzieren (Speicherdruck im Inhaltsspeicher). Das liegt daran, dass durch das Löschen der Frames die Rückrufe von Benachrichtigungen über den Stream-Druck ausgelöst werden. Die Anwendung kann dann die vorgelagerten Medien-Komponenten (wie den Encoder) so anpassen, dass die Bitrate gesenkt, Frames

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
			verworfen oder entsprechend gehandelt wird.
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	Für einen Stream sind derzeit keine Daten mehr verfügbar.	Dies ist potenziell ein gültiges Ergebnis, wenn die Medien-Pipeline langsamer produziert, als der Netzwerk-Thread die an den Dienst zu sendenden Frames konsumiert. Clients auf höherer Ebene (z. B. C++, Java oder Android) sehen diese Warnung nicht, da sie intern verarbeitet wird.
0x52000030	STATUS_INVALID_TAG_VERSION	Ungültige Version der Tag-Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x52000031	STATUS_SERVICE_UNKNOWN_ERROR	Von dem Netzwerk-Stack wurde ein unbekannter oder generischer Fehler zurückgegeben.	Weitere detaillierte Informationen finden Sie in den Protokollen.
0x52000032	STATUS_SERVICE_RESOURCE_IN_USE_ERROR	Die Ressource ist in Gebrauch.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000033	STATUS_SERVICE_CLIENT_LIMIT_ERROR	Client-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000034	STATUS_SERVICE_DEVICE_LIMIT_ERROR	Geräte-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000035	STATUS_SERVICE_STREAM_LIMIT_ERROR	Stream-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000036	STATUS_SERVICE_RESOURCE_DELETED_ERROR	Die Ressource wurde gelöscht oder wird gerade gelöscht wird.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000037	STATUS_SERVICE_TIMEOUT_ERROR	Zeitüberschreitung des Dienstaufrufs.	Der Aufruf einer bestimmten Dienst-API führte zu einem Timeout. Stellen Sie sicher, dass Sie über eine gültige Netzwerkverbindung verfügen. Der PIC wiederholt die Operation automatisch.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000038	STATUS_STREAM_READY_CALLBACK_FAILED	Stream bereit-Benachrichtigung.	Diese Benachrichtigung wird vom PIC an den Client gesendet und bestätigt, dass der asynchrone Stream erstellt wurde.
0x52000039	STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	Es wurden ungültige Tags angegeben.	Die Tag-Anzahl ist nicht Null, aber die Tags sind leer. Stellen Sie sicher, dass die Tags angegeben sind oder dass die Anzahl Null ist.
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	Ungültige Version der StreamDescription - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	Ungültige Länge des Tag-Namens.	Nehmen Sie auf die Limits des Tag-Namens Bezug, die unter Beschränkungen des Producer-SDK angegeben werden.
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	Ungültige Länge des Tag-Wertes.	Nehmen Sie auf die Limits des Tag-Wertes Bezug, die unter Beschränkungen des Producer-SDK angegeben werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	Die TagResource -API ist fehlgeschlagen.	Der TagResource -API-Aufruf ist fehlgeschlagen. Prüfen Sie auf eine gültige Netzwerkverbindung. Weitere Informationen zum Fehler finden Sie in den Protokollen.
0x5200003e	STATUS_INVALID_CUSTOM_DATA	PIC-APIs werden mit ungültigen benutzerdefinierten Daten aufgerufen.	In einem Aufruf der PIC-APIs wurden ungültige benutzerdefinierte Daten angegeben. Dies kann nur bei Clients vorkommen, die PIC direkt nutzen.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	Ungültige CreateStreamResponse -Struktur.	Die Struktur oder die zugehörigen Felder sind ungültig (d. h. ARN ist Null oder größer als der unter Beschränkungen des Producer-SDK angegebene Wert).

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	Client-Authentifizierung fehlgeschlagen.	Der PIC konnte nach einer Reihe von Wiederholungsversuchen nicht die richtigen Authentifizierungsinformationen (AccessKeyId oderSecretAccessKey) abrufen. Überprüfen Sie die Authentifizierungsintegration. Die Beispielanwendungen verwenden Umgebungsvariablen, um Anmeldeinformationen an die C++ Producer-Bibliothek zu übergeben.
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	Das Anfordern des Sicherheits-Token ist fehlgeschlagen.	Diese Situation kann bei Clients auftreten, die PIC direkt nutzen. Der Aufruf schlägt nach einigen Wiederholungsversuchen mit diesem Fehler fehl.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	Bereitstellungsfehler.	Die Bereitstellung ist nicht implementiert.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	Der Producer Client konnte nicht erstellt werden.	Vom PIC wurde nach einigen Wiederholungsversuchen ein generischer Fehler zurückgegeben, wenn die Client-Erstellung fehlschlägt.
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	Der Producer Client konnte nicht in den Status READY (BEREIT) gebracht werden.	Wird vom PIC-Zustandsautomaten zurückgegeben, wenn der PIC nicht in den Status READY (BEREIT) gebracht werden kann. Weitere Informationen zur Ursache finden Sie in den Protokollen.
0x52000045	STATUS_TAG_RESOURCE_CLIENT_CALL_FAILED	TagResource für den Producer Client ist fehlgeschlagen.	Der API-Aufruf TagResource für den Producer Client ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Protokollen.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	Gerät/Producer konnte nicht erstellt werden.	Die SDKs auf höherer Ebene (z. B. C++ oder Java) implementieren die API zur Erstellung von Geräten oder Herstellern noch nicht. Clients, die PIC direkt nutzen, können über die Ergebnis-Benachrichtigung auf einen Fehler hinweisen.
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	Der Zeitstempel der empfangenen ACK wird nicht angezeigt .	Dieser Fehler tritt auf, wenn der Frame, der der empfangenen ACK entspricht, aus dem Inhaltsansichtsfenster fällt. Dies geschieht in der Regel bei langsamer ACK-Übermittlung. Dies kann als eine Warnung und ein Hinweis auf einen langsamen Downlink gedeutet werden.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	Ungültige Version der FragmentAck -Struktur.	Geben Sie die korrekte aktuelle Version der FragmentAck -Struktur an.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	Ungültige Angabe des Sicherheits-Token-Ablaufs.	Der Ablauf des Sicherheitstokens sollte einen absoluten Zeitstempel in der future haben, der größer ist als der aktuelle Zeitstempel, mit einer Nachfrist. Informationen zu den Limits der Übergangsfrist finden Sie unter Beschränkungen des Producer-SDK .
0x5200004a	STATUS_END_OF_STREAM	Ende des Streams-(EOS)-Indikator.	Gibt im API-Aufruf <code>GetStreamData</code> an, dass die aktuelle Upload-Sitzung beendet ist. Dies tritt auf, wenn die Sitzung endet oder fehlerhaft ist, das Sitzungs-Token abgelaufen ist und die Sitzung rotiert wird.
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	Doppelter Stream-Name.	Mehrere Datenströme können nicht denselben Stream-Namen besitzen. Wählen Sie eine eindeutigen Namen für den Stream.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	Ungültiger Aufbewahrungszeitraum.	In der <code>StreamInfo</code> - Struktur wird ein ungültiger Aufbewahrungszeitraum angegeben. Weitere Informationen zum gültigen Wertebereich für den Aufbewahrungszeitraum finden Sie unter Beschränkungen des Producer-SDK .
0x5200004d	STATUS_INVALID_ACK_KEY_START	Ungültig <code>FragmentAck</code> .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge. Ungültiger Schlüssel-Startindikator. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x5200004e	STATUS_INVALID_ACK_DUPLICATE_KEY_NAME	Ungültig <code>FragmentAck</code> .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge. Mehrere Schlüssel besitzen den gleichen Namen. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200004f	STATUS_INVALID_ACK_VALUE_START	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüsselwert-Startindikators. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000050	STATUS_INVALID_ACK_VALUE_END	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüsselwert-Endindikators. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000051	STATUS_INVALID_ACK_TYPE	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge, da ein ungültiges ACK-Typ angegeben wurde.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	Der Stream wurde gestoppt.	Obwohl der Stream gestoppt wurde, wird dennoch ein Frame in den Stream geleitet.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Ungültige Version der <code>StreamMetrics</code> -Struktur.	Geben Sie die korrekte aktuelle Version der <code>StreamMetrics</code> -Struktur an.
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Ungültige Version der <code>ClientMetrics</code> -Struktur.	Geben Sie die korrekte aktuelle Version der <code>ClientMetrics</code> -Struktur an.
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	Bei der Initialisierung des Producers wurde nicht der Status READY (BEREIT) erreicht.	Während der Producer Client-Initialisierung wurde nicht der Status READY (BEREIT) erreicht. Weitere Informationen finden Sie in den Protokollen.
0x52000056	STATUS_STATE_MACHINE_STATE_NOT_FOUND	Interner Fehler des Zustandsautomaten.	Kein öffentlich sichtbarer Fehler.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	In der <code>FragmentAck</code> -Struktur wird ein ungültiger ACK-Typ angegeben.	Die in der <code>FragmentAck</code> -Struktur enthaltenen ACK-Typen sollten im öffentlichen Header definiert sein.
0x52000058	STATUS_INVALID_STREAM_READY_STATE	Interner Übergangsfehler des Zustandsautomaten.	Kein öffentlich sichtbarer Fehler.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	Das Stream-Objekt wurde freigegeben, nachdem der Producer freigegeben wurde.	Es wurde versucht, ein Stream Objekt freizugeben, nachdem das Producer-Objekt freigegeben wurde. Dies kann nur bei Clients vorkommen, die PIC direkt nutzen.
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	Interner Speicherfehler.	Ein interner Fehler, der darauf hinweist, dass die tatsächliche Zuweisung sgröße aus dem Inhaltsspeicher kleiner ist als die Größe des verpackten Frames und Fragments.
0x5200005b	STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION	Interner Speicherfehler.	Die gespeicherte Größe der Zuweisung in der Inhaltsansicht ist größer als die Zuweisungsgröße im Inhaltsspeicher.
0x5200005c	STATUS_ACK_STREAM_READ_ERROR	ACK-Streamlesefehler.	Ein Fehler, den das ACK vom Backend zurückgegeben hat, was auf einen Fehler beim Lesen oder Analysieren des Streams hinweist. Dies tritt in der Regel auf, wenn das Backend den Stream nicht abrufen kann. Durch automatisches erneutes Streamen kann der Fehler gewöhnlich behoben werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200005d	STATUS_AK_ERR_FRAGMENT_SIZE_REACHED	Die maximale Fragmentgröße wurde erreicht.	Die maximale Fragmentgröße in Byte ist unter Beschränkungen des Producer-SDK definiert. Dieser Fehler zeigt an, dass entweder sehr große Frames oder keine Schlüssel-Frames zum Erstellen von Fragmenten einer überschaubaren Größe vorhanden sind. Überprüfen Sie die Encoder-Einstellungen und stellen Sie sicher, dass die Keyframes korrekt erzeugt werden. Konfigurieren Sie den Encoder bei Streams mit sehr hoher Dichte so, dass zur Bewältigung der maximalen Größe Fragmente einer kleineren Dauer erstellt werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200005e	STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED	Die maximale Fragmentdauer wurde erreicht.	Die maximale Fragmentdauer ist unter Beschränkungen des Producer-SDK definiert. Dieser Fehler weist entweder auf eine sehr niedrige Framerate pro Sekunde oder auf fehlende Schlüssel-Frames zum Erstellen von Fragmenten einer überschaubaren Dauer hin. Überprüfen Sie die Encoder-Einstellungen und stellen Sie sicher, dass die Keyframes in regelmäßigen Abständen ordnungsgemäß erzeugt werden.
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	Die maximale Verbindungsdauer wurde erreicht.	Kinesis Video Streams erzwingt die maximale Verbindungsdauer, wie in der angegeben . Beschränkungen des Producer-SDK Das Producer SDK rotiert den Stream oder das Token automatisch, bevor das Maximum erreicht ist. Clients, die das SDK verwenden, sollten diesen Fehler nicht erhalten.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000060	STATUS_AK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC	Timecodes steigen nicht monoton an.	Das Producer-SDK erzwingt Zeitstempel, sodass Clients, die das SDK verwenden, diesen Fehler nicht erhalten sollten.
0x52000061	STATUS_AK_ERR_MULTI_TRACK_MKV	Mehrere Tracks im MKV.	Das Producer-SDK erzwingt Single-Track-Streams, sodass Clients, die das SDK verwenden, diesen Fehler nicht erhalten sollten.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000062	STATUS_AK_ERR_INVALID_MKV_DATA	Ungültige MKV-Daten.	Am Backend-MKV-Parser trat ein Fehler beim Parsen des Streams auf. Bei Clients, die das SDK verwenden, kann dieser Fehler auftreten, wenn der Stream während des Übergangs beschädigt wird. Dies kann auch passieren, wenn der Pufferdruck das SDK dazu zwingt, Tail-Frames zu verwerfen, die teilweise übertragen wurden. In letzterem Fall empfehlen wir, entweder die FPS und die Auflösung zu reduzieren, die Kompressionsrate zu erhöhen oder (falls es sich um ein „Burst-Netzwerk“ handelt) eine größere Speicher- und Pufferdauer einzuplanen, um dem temporären Druck Rechnung zu tragen.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000063	STATUS_AC K_ERR_INV ALID_PROD UCER_TIMESTAMP	Ungültiger Producer-Zeitstempel.	Der Dienst gibt diesen ACK-Fehler zurück, wenn die Uhr des Producers wesentlich vorgeht. Übergeordnete SDKs (z. B. Java oder C++) erfüllen mittels einer bestimmten Version der Systemuhr das aktuelle Zeit-Callback des PIC. Stellen Sie sicher, dass die Systemuhr richtig eingestellt ist. Clients, die den PIC direkt verwenden, sollten überprüfen, ob ihre Callback-Funktionen den richtigen Zeitstempel zurückgeben.
0x52000064	STATUS_AC K_ERR_STR EAM_NOT_ACTIVE	Inaktiver Stream.	Es wurde ein Aufruf einer Backend-API vorgenommen, während sich der Stream nicht im Status „Active“ (Aktiv) befand. Dies tritt auf, wenn der Client sofort nach dem Erstellen des Streams Frames in ihn einleitet. Das SDK bewältigt dieses Szenario mittels der Zustandsautomat- und Wiederherstellungsvorgänge.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000065	STATUS_AK_ERR_KMS_KEY_ACCESS_DENIED	Fehler: AWS KMS-Zugriff verweigert.	Wird zurückgegeben, wenn das Konto nicht auf den angegebenen Schlüssel zugreifen kann.
0x52000066	STATUS_AK_ERR_KMS_KEY_DISABLED	AWS KMS Der Schlüssel ist deaktiviert.	Der angegebene Schlüssel wurde deaktiviert.
0x52000067	STATUS_AK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS-Schlüssel-Validierungsfehler.	Allgemeiner Validierungsfehler. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz .
0x52000068	STATUS_AK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key nicht verfügbar.	Der Schlüssel ist nicht verfügbar. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz .
0x52000069	STATUS_AK_ERR_KMS_KEY_INVALID_USAGE	Ungültige Verwendung des KMS-Schlüssels.	Der AWS KMS key ist nicht für die Verwendung in diesem Kontext konfiguriert. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz .
0x5200006a	STATUS_AK_ERR_KMS_KEY_INVALID_STATE	Ungültiger AWS KMS-Status.	Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz .

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200006b	STATUS_ACK_ERR_KMS_KEY_NOT_FOUND	Der KMS-Schlüssel wurde nicht gefunden.	Die Schlüssel wurde nicht gefunden. Weitere Informationen finden Sie in der AWS Key Management Service-API-Referenz .
0x5200006c	STATUS_ACK_ERR_STREAM_DELETED	Der Stream wurde oder wird derzeit gelöscht.	Der Stream wird derzeit von einer anderen Anwendung oder über die AWS Management Console gelöscht.
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	Interner Fehler	Interner Fehler eines generischen Dienstes.
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR	Fragment-Archivierungsfehler.	Wird zurückgegeben, wenn der Dienst das Fragment nicht dauerhaft speichern und indizieren kann. Dies kann, wenn auch selten, aus verschiedenen Gründen auftreten. Standardmäßig wiederholt das SDK das Senden des Fragments.
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	Unbekannter Fehler.	Der Service gab einen unbekanntes Fehler zurück.
0x52000070	STATUS_MISSING_ERR_ACK_ID	Fehlende ACK-Informationen.	Die ACK-Parser hat das Analysieren abgeschlossen, die FragmentAck - Informationen fehlen aber.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	Ungültige ACK-Segmentlänge.	Für den ACK-Parser wurde eine ACK-Segment-Zeichenfolge ungültiger Länge angegeben. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x52000074	STATUS_MAXIMUM_FRAGMENT_METADATA_COUNT	Die maximale Anzahl von Metadaten elementen wurde zu einem Fragment hinzugefügt.	Ein Kinesis-Videostream kann einem Fragment bis zu 10 Metadaten elemente hinzufügen, indem entweder ein nicht persistentes Element zu einem Fragment hinzugefügt wird oder indem ein persistentes Element zur Metadaten-Warteschlange hinzugefügt wird. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	Ein Limit (maximale Metadatenanzahl, Länge des Metadatennamens oder Länge des Metadatenwerts) wurde erreicht.	Das Producer-SDK begrenzt die Anzahl und Größe von Metadaten elementen. Dieser Fehler tritt nur auf, wenn die Grenzwerte im Producer-SDK-Code geändert werden. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .
0x52000076	STATUS_BLOCKING_INTERRUPTED_STREAM_TERMINATED	Nicht implementiert.	
0x52000077	STATUS_INVALID_METADATA_NAME	Der Metadatenname ist ungültig.	Der Metadatenname darf nicht mit der Zeichenfolge "AWS" beginnen. Wenn dieser Fehler auftritt, wird das Metadatenelement nicht zur Fragment- oder Metadatenwarteschlange hinzugefügt. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	Das Ende eines Fragment-Frames weist einen ungültigen Status auf.	Das Ende eines Fragments sollte nicht in einem non-key-frame fragmentierten Stream gesendet werden.
0x52000079	STATUS_TRACK_INFO_MISSING	Track-Informationen fehlen.	Die Titelnummer muss größer als Null sein und mit der Track-ID übereinstimmen.
0x5200007a	STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED	Maximale Track-Anzahl wurde überschritten.	Sie können maximal drei Titel pro Stream haben.
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	Der Offline-Streaming-Modus-Aufbewahrungszeitraum ist auf Null gesetzt.	Die Aufbewahrungszeit für den Offline-Streaming-Modus sollte nicht auf Null gesetzt werden.
0x5200007c	STATUS_ACK_TRACK_NUMBER_MISMATCH	Die Track-Nummer des ACK-Fehlers ist falsch zugeordnet.	
0x5200007d	STATUS_ACK_FRAMES_MISSING_FOR_TRACK	Frames fehlen für einen Track.	
0x5200007e	STATUS_ACK_MORE_THAN_ALLOWED_TRACKS_FOUND	Die maximal zulässige Anzahl von Tracks wurde überschritten.	

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200007f	STATUS_UPLOAD_HANDLE_ABORTED	Die Upload-Verarbeitung wurde abgebrochen.	
0x52000080	STATUS_INVALID_CERT_PATH_LENGTH	Ungültige Zertifikatpfadlänge.	
0x52000081	STATUS_DUPLICATE_TRACK_ID_FOUND	Doppelte Track-ID gefunden.	
0x52000082	STATUS_INVALID_CLIENT_INFO_VERSION		
0x52000083	STATUS_INVALID_CLIENT_ID_STRING_LENGTH		
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAXIMUM_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUT_MEDIA_LAST_PERSIST_ACK_NOT_RECEIVED		
0x52000089	STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS		
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200008e	STATUS_INVALID_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_METADATA_VALUE_LENGTH		

Von der Duration-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Duration Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

Von der gemeinsamen Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Common Bibliothek zurückgegeben werden.

Note

Diese Codes für Fehler- und Statusinformationen sind vielen APIs gemeinsam.

Code	Fehlermeldung	Beschreibung
0x00000001	STATUS_NULL_ARG	Für ein obligatorisches Argument wurde NULL übergeben.

Code	Fehlermeldung	Beschreibung
0x00000002	STATUS_INVALID_ARG	Für ein Argument wurde ein ungültiger Wert angegeben.
0x00000003	STATUS_INVALID_ARG_LEN	Es wurde eine ungültige Länge für ein Argument angegeben.
0x00000004	STATUS_NOT_ENOUGH_MEMORY	Es konnte nicht genug Speicher zugewiesen werden.
0x00000005	STATUS_BUFFER_TOO_SMALL	Die angegebene Puffergröße ist zu klein.
0x00000006	STATUS_UNEXPECTED_EOF	Es wurde ein unerwartetes Dateende erreicht.
0x00000007	STATUS_FORMAT_ERROR	Es ist ein ungültiges Format aufgetreten.
0x00000008	STATUS_INVALID_HANDLE_ERROR	Ungültiger Handle-Wert.
0x00000009	STATUS_OPEN_FILE_FAILED	Fehler beim Öffnen einer Datei.
0x0000000a	STATUS_READ_FILE_FAILED	Fehler beim Lesen aus einer Datei.
0x0000000b	STATUS_WRITE_TO_FILE_FAILED	Fehler beim Schreiben in eine Datei.
0x0000000c	STATUS_INTERNAL_ERROR	Es ist ein ungewöhnlicher interner Fehler aufgetreten, der möglicherweise auf einen SDK- oder Service-API-Fehler hinweist.

Code	Fehlermeldung	Beschreibung
0x0000000d	STATUS_INVALID_OPERATION	Eine Operation ist ungültig oder nicht zulässig.
0x0000000e	STATUS_NOT_IMPLEMENTED	Die Funktion ist nicht implementiert.
0x0000000f	STATUS_OPERATION_TIMED_OUT	Bei der Operation ist eine Zeitüberschreitung aufgetreten.
0x00000010	STATUS_NOT_FOUND	Eine erforderliche Ressource wurde nicht gefunden.
0x00000011	STATUS_CREATE_THREAD_FAILED	Fehler beim Erstellen eines Threads.
0x00000012	STATUS_THREAD_NOT_ENOUGH_RESOURCES	Es sind nicht genügend Ressourcen vorhanden, um einen weiteren Thread zu erstellen, oder es wurde eine vom System festgelegte Beschränkung der Anzahl von Threads festgestellt.
0x00000013	STATUS_THREAD_INVALID_ARG	Es wurden ungültige Thread-Attribute angegeben, oder ein anderer Thread wartet bereits darauf, diesem Thread beizutreten.
0x00000014	STATUS_THREAD_PERMISSIONS	Keine Berechtigung zum Festlegen der in Thread-Attributen angegebenen Zeitplanungsrichtlinie und der in Thread-Attributen angegebenen Parameter.

Code	Fehlermeldung	Beschreibung
0x00000015	STATUS_THREAD_DEAD LOCKED	Ein Deadlock wurde erkannt oder der beitretende Thread spezifiziert den aufrufenden Thread.
0x00000016	STATUS_THREAD_DOES _NOT_EXIST	Der Thread mit der angegebenen Thread-ID konnte nicht gefunden werden.
0x00000017	STATUS_JOIN_THREAD _FAILED	Beim Thread-Join-Vorgang wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x00000018	STATUS_WAIT_FAILED	Die maximale Wartezeit für die Bedingungsvariable wurde überschritten.
0x00000019	STATUS_CANCEL_THRE AD_FAILED	Beim Vorgang zum Abbrechen des Threads wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x0000001a	STATUS_THREAD_IS_N OT_JOINABLE	Der Thread-Join-Vorgang wird für einen Thread angefordert, dem keine Verbindung hergestellt werden kann.
0x0000001b	STATUS_DETACH_THRE AD_FAILED	Beim Vorgang zum Trennen des Threads wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x0000001c	STATUS_THREAD_ATTR _INIT_FAILED	Das Thread-Attribute-Objekt konnte nicht initialisiert werden.

Code	Fehlermeldung	Beschreibung
0x0000001d	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	Die Stackgröße für das Thread-Attribute-Objekt konnte nicht festgelegt werden.
0x0000001e	STATUS_MEMORY_NOT_FREED	Wird nur in den Tests verwendet. Zeigt an, dass nicht der gesamte angeforderte Speicher freigegeben wurde.
0x10000015	STATUS_INVALID_ALLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

Von der Heap-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Heap Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung
0x01000001	STATUS_HEAP_FLAGS_ERROR	Es wurde eine ungültige Kombination von Flags angegeben.
0x01000002	STATUS_HEAP_NOT_INITIALIZED	Eine Operation wurde vor der Heap-Initialisierung versucht.
0x01000003	STATUS_HEAP_CORRUPTED	Der Heap wurde beschädigt oder das Sicherheits-Freque

Code	Fehlermeldung	Beschreibung
		nzband (im Debug-Modus) wurde überschrieben. Ein Pufferüberlauf im Client-Code könnte eine Heap-Beschädigung verursachen.
0x01000004	STATUS_HEAP_VRAM_LIB_MISSING	Die VRAM-Benutzer- oder Kernelmodusbibliothek (Video-RAM) kann nicht geladen werden oder fehlt. Überprüfen Sie, ob die zugrunde liegende Plattform VRAM-Zuweisungen unterstützt.
0x01000005	STATUS_HEAP_VRAM_LIB_REOPEN	Fehler beim Öffnen der VRAM-Bibliothek.
0x01000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	Fehler beim Laden des INIT-Funktionsexports.
0x01000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	Fehler beim Laden des ALLOC-Funktionsexports.
0x01000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	Fehler beim Laden des FREE-Funktionsexports.
0x01000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	Fehler beim Laden des LOCK-Funktionsexports.
0x0100000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	Fehler beim Laden des UNLOCK-Funktionsexports.
0x0100000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	Fehler beim Laden des UNINIT-Funktionsexports.
0x0100000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	Fehler beim Laden des GETMAX-Funktionsexports.

Code	Fehlermeldung	Beschreibung
0x0100000d	STATUS_HEAP_DIRECT_MEM_INIT	Fehler bei der Initialisierung des Haupt-Heap-Pools im Hybrid-Heap.
0x0100000e	STATUS_HEAP_VRAM_INIT_FAILED	Die dynamische VRAM-Initialisierung ist fehlgeschlagen.
0x0100000f	STATUS_HEAP_LIBRARY_FREE_FAILED	Fehler beim Aufheben der Zuweisung und Freisetzen der VRAM-Bibliothek.
0x01000010	STATUS_HEAP_VRAM_ALLOC_FAILED	Die VRAM-Zuweisung ist fehlgeschlagen.
0x01000011	STATUS_HEAP_VRAM_FREE_FAILED	Die VRAM-Freisetzung ist fehlgeschlagen.
0x01000012	STATUS_HEAP_VRAM_MAP_FAILED	Die VRAM-Zuordnung ist fehlgeschlagen.
0x01000013	STATUS_HEAP_VRAM_UNMAP_FAILED	Das Aufheben der VRAM-Zuordnung ist fehlgeschlagen.
0x01000014	STATUS_HEAP_VRAM_UNINIT_FAILED	Das Aufheben der VRAM-Initialisierung ist fehlgeschlagen.

Von der MKVGen-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der MKVGen Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Ungültige Mitglieder der Frame-Datenstruktur. Stellen

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
		<p>Sie sicher, dass Dauer, Größe und Frame-Daten gültig sind und innerhalb der unter angegebenen Grenzen liegen Beschränkungen des Producer-SDK.</p>
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	<p>Ungültiger Frame-Zeitstempel. Der errechnete PTS (Präsentationszeitstempel) und der DTS (Decodierungszeitstempel) sind größer oder gleich dem Zeitstempel des Start-Frames des Fragments. Dies ist ein Hinweis auf einen potenziellen Neustart der Medien-Pipeline oder ein Stabilitätsproblem des Encoders. Informationen zur Problembeseitigung finden Sie unter Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“</p>
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	<p>Es wurde eine ungültige Fragment-Dauer angegeben wurde. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK.</p>
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	<p>Ungültige Zeichenfolgenlänge des Inhaltstyps. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK.</p>

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	Es wurde versucht, eine Zahl zu codieren, die zur Darstellung im EBML (Extensible Binary Meta Language)-Format zu groß ist. Die SDK-Clients sollten damit nicht konfrontiert werden.
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	Ungültige Zeichenfolgenlänge der Codec-ID. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	Ungültige Zeichenfolgenlänge des Tracknamens. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	Ungültige Länge privater Codec-Daten. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	Der Codec Private Data (CPD) ist NULL, wohingegen die CPD-Größe größer als Null ist.
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	Ungültiger Timecode-Skalierungswert. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	Der Frame-Timecode ist größer als das maximale Limit. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	Der maximale Frame-Timecode wurde erreicht. Das MKV-Format verwendet signierte 16 Bits zur Darstellung des Timecodes des Frames relativ zum Beginn des Clusters. Der Fehler wird generiert, wenn der Frame-Timecode nicht dargestellt werden kann. Dieser Fehler weist entweder auf eine ungültige Timecode-Skalierungsauswahl oder eine zu lange Cluster-Dauer hin, die bei der Darstellung des Frame-Timecodes zu einem Überlauf des signierten 16-Bit-Bereichs führt.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x3200000d	STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA	Es ist ein ungültiger Annex-B-Startcode aufgetreten. Beispielsweise tritt bei Angabe des Annex-B-Anpassungsflags während der Auswertung eine ungültige Startsequenz von mehr als drei Nullen auf. Ein gültiges Annex-B-Format sollte über eine Emulationsverhütungsssequenz verfügen, mit der eine Abfolge von drei oder mehr Nullen im Bytestream ausgeklammert wird. Weitere Informationen finden Sie in der MPEG-Spezifikation. Weitere Informationen über diesen Fehler bei Android finden Sie unter STATUS_MKV_INVALID_ANNEXB_NALU_IN_FR_AME_DATA (0x3200000d)-Fehler unter Android .
0x3200000e	STATUS_MKV_INVALID _AVCC_NALU_IN_FRAM E_DATA	Ungültiges AVCC-NALU-Paketieren, wenn das anpassende AVCC-Flag angegeben ist. Stellen Sie sicher, dass der Bytestream ein gültiges AVCC-Format hat. Weitere Informationen finden Sie in der MPEG-Spezifikation .

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x3200000f	STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED	Es wurden sowohl AVCC- als auch Annex-B-NALUs zur Anpassung spezifiziert. Machen Sie nur eine oder keine Angabe.
0x32000010	STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD	Ungültiges Annex-B-Format der CPD bei Angabe des Annex-B-Anpassungsflags. Stellen Sie sicher, dass die CPD ein gültiges Anhang-B-Format hat. Ist dies nicht der Fall, entfernen Sie das CPD-Anhang-B-Anpassungskennzeichen.
0x32000011	STATUS_MKV_PTS_DTS_ARE_NOT_SAME	Kinesis Video Streams erzwingt, dass PTS (Presentation Timestamp) und DTS (Decoding Timestamp) für die Fragment-Startframes identisch sind. Dies sind die Schlüssel-Frames, mit denen das Fragment beginnt.
0x32000012	STATUS_MKV_INVALID_H264_H265_CPD	Fehler beim Analysieren privater H264/H265 Codec-Daten.
0x32000013	STATUS_MKV_INVALID_H264_H265_SPS_WIDTH	Fehler beim Extrahieren der Breite privater Codec-Daten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000014	STATUS_MKV_INVALID_H264_H265_SPS_HEIGHT	Fehler beim Extrahieren der Höhe privater Codec-Daten.
0x32000015	STATUS_MKV_INVALID_H264_H265_SPS_NALU	Ungültiges H264/H265 SPS NALU.
0x32000016	STATUS_MKV_INVALID_BIH_CPD	Ungültiges Bitmap-Info-Header-Format in den privaten Codec-Daten.
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	Ungültige Anzahl an Invalid High Efficiency Video Coding (HEVC)-Network Abstraction Layer (NALU)-Einheiten.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	Ungültiges HEVC-Format.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	Fehlende HEVC-NALUs im Sequence Parameter Set (SPS).
0x3200001a	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	Ungültige HEVC-SPS-NALU-Größe.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	Ungültige Chroma-Format-IDC.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	Ungültiges HEVC-reserviertes SPS.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	Mindestgröße AnnexBb des privaten Betawerts für den Codec. Für H264 muss dieser Wert größer oder gleich 11 sein. Für H265 muss dieser Wert größer oder gleich 15 sein.
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Fehlende private Codec-Daten in Annex-B-NALUs.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Ungültige private Codec-Daten in Annex-B-NALUs.
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	Ungültige Länge des Tag-Namens. Zulässiger Wert ist größer als Null und niedriger als 128.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	Ungültige Länge des Tag-Wertes. Der gültige Wert ist größer als Null und kleiner als 256.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	Ungültige Generator-Zustands-Tags.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	Ungültiger Sampling-Häufigkeitsindex für private AAC-Codec-Daten.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	Ungültige Kanalkonfiguration für private AAC-Codec-Daten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000025	STATUS_MKV_INVALID_AAC_CPD	Ungültige private AAC-Codec-Daten.
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	Track-Informationen nicht gefunden.
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	Ungültige Segment-UUID.
0x32000028	STATUS_MKV_INVALID_TRACK_UUID	Ungültige Track-UUID.
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	
0x3200002a	STATUS_MKV_INVALID_AMS_ACM_CPD	
0x3200002b	STATUS_MKV_MISSING_SPS_FROM_H264_CPD	
0x3200002c	STATUS_MKV_MISSING_PPS_FROM_H264_CPD	
0x3200002d	STATUS_MKV_INVALID_PARENT_TYPE	

Von der Trace-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Trace Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0x10100001	STATUS_MIN_PROFILER_BUFFER

Von der Utils-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der `Utils` Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED

Code	Fehlermeldung
0x40000011	STATUS_UTIL_MAX_TAG_COUNT
0x40000012	STATUS_UTIL_INVALID_TAG_VERSION
0x40000013	STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL
0x40000014	STATUS_UTIL_INVALID_TAG_NAME_LEN
0x40000015	STATUS_UTIL_INVALID_TAG_VALUE_LEN
0x4000002a	STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE
0x4000002b	STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED
0x4000002c	STATUS_THREADPOOL_MAX_COUNT
0x4000002d	STATUS_THREADPOOL_INTERNAL_ERROR
0x40100001	STATUS_HASH_KEY_NOT_PRESENT
0x40100002	STATUS_HASH_KEY_ALREADY_PRESENT
0x40100003	STATUS_HASH_ENTRY_ITERATION_ABORT
0x41000001	STATUS_BIT_READER_OUT_OF_RANGE
0x41000002	STATUS_BIT_READER_INVALID_SIZE
0x41100001	STATUS_TIMER_QUEUE_STOP_SCHEDULING
0x41100002	STATUS_INVALID_TIMER_COUNT_VALUE

Code	Fehlermeldung
0x41100003	STATUS_INVALID_TIMER_PERIOD_VALUE
0x41100004	STATUS_MAX_TIMER_COUNT_REACHED
0x41100005	STATUS_TIMER_QUEUE_SHUTDOWN
0x41200001	STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN
0x41200002	STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

Von der View-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der View Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	Es wurde eine ungültige Anzahl von Inhaltsansichts-Elementen angegeben. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	Es wurde eine ungültige Dauer der Inhaltsansicht angegeben. Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .

Code	Fehlermeldung	Beschreibung
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	Es wurde versucht, über die Kopfposition hinauszugehen.
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	Es wurde ein ungültiger Index angegeben.

Code	Fehlermeldung	Beschreibung
0x30000005	STATUS_CONTENT_VI EW_INVALID_TIMESTAMP	<p>Es liegt ein ungültiger Zeitstempel oder eine Zeitstempel-Überlappung vor. Der Zeitstempel für die Frame-Dekodierung sollte größer oder gleich dem vorherigen Frame-Zeitstempel sein, zuzüglich der vorherigen Frame-Dauer: $\text{DTS}(n) \geq \text{DTS}(n-1) + \text{Duration}(n-1)$. Dieser Fehler weist oft auf einen instabilen Encoder hin. Der Encoder produziert einen Burst codierter Frames, deren Zeitstempel kleiner als die Intra-Frame-Dauer ist. Oder der Stream ist zur Verwendung von SDK-Zeitstempeln konfiguriert, und die Frames werden schneller als ihre Dauer gesendet. Um dem Jitter im Encoder etwas abzuhelpfen, geben Sie in der <code>StreamInfo.StreamCaps</code>-Struktur eine kleinere Frame-Dauer an. Wenn der Stream beispielsweise 25 FPS hat, beträgt die Dauer jedes Frames 40 ms. Um den „Jitter“ des Encoders zu handhaben, empfehlen wir jedoch, die Hälfte dieser Frame-Dauer (20 ms) zu verwenden. Bei einigen</p>

Code	Fehlermeldung	Beschreibung
		Streams muss die Dauer für die Fehlererkennung genauer kontrolliert werden.
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	Es wurde eine ungültige Länge für Inhaltsansichts-Elementdaten angegeben.

Von PutFrame Callbacks zurückgegebene Fehler und Statuscodes — C-Producer-Bibliothek

Der folgende Abschnitt enthält Fehler- und Statusinformationen, die von Callbacks für den PutFrame Vorgang innerhalb der C-Producer-Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x15000001	STATUS_STOP_CALLBACK_CHAIN	Die Callback-Kette wurde gestoppt.	
0x15000002	STATUS_MAX_CALLBACK_CHAIN	Die maximale Callback-Kette wurde erreicht.	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	Ungültige Version der PlatformCallbacks - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Ungültige Version der ProducerCallbacks - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Ungültige Version der <code>StreamCallbacks</code> -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Ungültige Version der <code>AuthCallbacks</code> -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	Ungültige Version der <code>ApiCallbacks</code> -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Ungültige Version der <code>AwsCredentials</code> -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000009	STATUS_MAX_REQUEST_HEADER_COUNT	Die maximale Anzahl der Anforderungs-Header wurde erreicht.	
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	Die maximale Länge des Namens des Anforderungs-Headers wurde erreicht.	

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	Die maximale Länge des Werts des Anforderungs-Headers wurde erreicht.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	Ungültiges JSON-Objekt für einen API-Aufruf zurückgegeben.	
0x1500000d	STATUS_CURL_INIT_FAILED	Curl-Initialisierung fehlgeschlagen.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Curl-lib-Initialisierung fehlgeschlagen.	
0x1500000f	STATUS_INVALID_DESCRIPTOR_RETURN_JSON	Ungültiges Rückgabe-JSON für DescribeStream.	
0x15000010	STATUS_HMAC_GENERATION_ERROR	Fehler bei HMAC-Generation.	
0x15000011	STATUS_IOT_FAILED	Die IoT-Autorisierung ist fehlgeschlagen.	
0x15000012	STATUS_MAX_ROLE_ALIAS_LENGTH_EXCEEDED	Die maximale Länge des Rollenalias wurde erreicht.	Geben Sie einen kürzeren Alias an.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x15000013	STATUS_MAX_USER_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED	Die maximale Länge der Namensendung des Agenten wurde erreicht.	
0x15000014	STATUS_MAX_CUSTOM_USER_AGENT_LENGTH_EXCEEDED	Die maximale Länge des Kundenbenutzeragenten wurde erreicht.	
0x15000015	STATUS_INVALID_USAGE_AGENT_LENGTH	Ungültige Benutzeragentenlänge.	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	Ungültiger Endpunkt-Caching-Zeitraum.	Geben Sie einen Caching-Zeitraum von weniger als 24 Stunden an.
0x15000017	STATUS_IOT_EXPIRATION_OCCURS_IN_PAST	Der IoT-Ablaufzeitstempel liegt in der Vergangenheit.	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	Die IoT-Ablaufanalyse ist fehlgeschlagen.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNCTION		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAXIMUM_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEGINNING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEGINNING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RESET_FAILED		
0x16000001	STATUS_CURL_PERFORMANCE_FAILED		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH		
0x16000003	STATUS_IOT_NULL_AWS_CREDS		
0x16000004	STATUS_IOT_INVALID_URI_LEN		
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT		

Network Abstraction Layer (NAL) Adaptation Flag-Referenz

Dieser Abschnitt enthält Informationen zu verfügbaren Flags für die `StreamInfo.NalAdaptationFlags`-Aufzählung.

Der [Elementarstrom](#) in einer Anwendung kann das Format Annex-B oder AVCC haben:

- Das Annex-B-Format begrenzt [NALUs \(Network Abstraction Layer Units\)](#) mit zwei Bytes Nullen, gefolgt von einem oder drei Bytes Nullen, gefolgt von der Zahl 1 (Startcode genannt, z. B. 000000001).
- Das Format AVCC hüllt auch NALUs ein, aber jeder NALU geht ein Wert voraus, der die Größe der NALU angibt (normalerweise vier Bytes).

Viele Encoder produzieren das Annex-B-Bitstream-Format. Einige höhere Bitstream-Prozessoren (z. B. die Wiedergabe-Engine oder der [Media Source Extensions \(MSE\)](#)-Player in der AWS Management Console) verwenden das AVCC-Format für ihre Frames.

Die privaten Codec-Daten (CPD), bei denen es sich um SPS/PPS (Sequence Parameter Set/ Picture Parameter Set) für den H.264-Codec handelt, können auch im Annex-B- oder AVCC-Format vorliegen. Bei der CPD unterscheiden sich die Formate jedoch von den zuvor beschriebenen.

Die Flags teilen dem SDK mit, die NALUs an AVCC oder Annex-B für Frame-Daten und CPD wie folgt anzupassen:

Flag	Anpassung
NAL_ADAPTATION_FLAG_NONE	Keine Anpassung.
NAL_ADAPTATION_ANNEXB_NALS	Passen Sie die NALUs des Anhangs B an die AVCC-NALUs an.
NAL_ADAPTATION_AVCC_NALS	Passen Sie die AVCC-NALUs an die Anhang-B-NALUs an.
NAL_ADAPTATION_ANNEXB_CPD_NALS	Passen Sie die NALUs von Anhang B für die privaten Codec-Daten an die NALUs im AVCC-Format an.
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	Passen Sie die NALUs von Anhang B für den Codec an und stellen Sie private Daten in die NALUs im AVCC-Format ein.

Weitere Informationen zu NALU-Typen finden Sie in Abschnitt 1.3: Einheitentypen der Netzwerkabstraktionsschicht in [RFC 3984](#).

SDK-Strukturen für Hersteller

Dieser Abschnitt enthält Informationen zu Strukturen, die Sie verwenden können, um Daten für das Kinesis Video Streams Producer-Objekt bereitzustellen.

Themen

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

Die `DefaultDeviceInfoProvider` Objekte `DeviceInfo` steuern das Verhalten des Kinesis Video Streams Producer-Objekts.

Felder für Mitglieder

- `Version` — Ein ganzzahliger Wert, mit dem sichergestellt wird, dass die richtige Version der Struktur mit der aktuellen Version der Codebasis verwendet wird. Die aktuelle Version wird mit dem Makro `DEVICE_INFO_CURRENT_VERSION` angegeben.
- `name` — Der für Menschen lesbare Name für das Gerät.
- `tagCount/tags` — Wird derzeit nicht verwendet.
- `streamCount` — Die maximale Anzahl von Streams, die das Gerät verarbeiten kann. Dadurch wird anfänglich Speicher für Zeiger auf die Streamobjekte zugewiesen, die eigentlichen Streamobjekte werden aber erst später erstellt. Standardmäßig handelt es sich um 16 Streams, dies kann jedoch in der Datei `DefaultDeviceInfoProvider.cpp` geändert werden.
- `storageInfo`: Ein Objekt, das die Hauptspeicherkonfiguration beschreibt. Weitere Informationen finden Sie unter [StorageInfo](#).

StorageInfo

Gibt die Konfiguration des Hauptspeichers für Kinesis Video Streams an.

Die Standardimplementierung basiert auf einem schnellen Heap mit geringer Fragmentierung, optimiert für das Streamen. Sie verwendet den Allocator `MEMALLOC`, der auf einer gegebenen Plattform überschrieben werden kann. Einige Plattformen sehen eine virtuelle Speicherzuweisung vor, ohne diese Zuweisung mit physischen Seiten zu hinterlegen. Erst bei Nutzung des Speichers werden den virtuellen Seiten physische Seiten hinterlegt. Dies führt zu wenig verfügbarem Speicher im Gesamtsystem, während der Speicher tatsächlich nicht ausgelastet wird.

Berechnen Sie die Standardspeichergröße mit folgender Formel. Der Parameter `DefragmentationFactor` muss auf 1,2 (20 Prozent) festgelegt werden.

$$\text{Size} = \text{NumberOfStreams} * \text{AverageFrameSize} * \text{FramesPerSecond} * \text{BufferDurationInSeconds} * \text{DefragmentationFactor}$$

Im folgenden Beispiel verfügt ein Gerät über Audio- und Videostreams. Der Audiostream besteht aus 512 Samples pro Sekunde, ein durchschnittliches Sample aus 100 Byte. Der Videostream besteht aus 25 Frames pro Sekunde, mit einem Durchschnitt von 10.000 Bytes. Jeder Stream verfügt über einen Puffer mit 3 Minuten Laufzeit.

$$\text{Size} = (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) * 1.2 = 65059200 = \sim 66\text{MB}.$$

Wenn das Gerät über mehr verfügbaren Speicher verfügt, empfehlen wir, dem Speicher mehr Speicher hinzuzufügen, um eine starke Fragmentierung zu vermeiden.

Stellen Sie sicher, dass die Speichergröße ausreichend ist, um die vollen Puffer für alle Streams mit hoher Kodierungskomplexität (wenn die Framegröße aufgrund von viel Bewegung größer ist) oder wenn die Bandbreite gering ist, aufzunehmen. Wenn der Producer Speicherauslastung erreicht, gibt er Speicherüberlauf-Druck-Callbacks () aus. `StorageOverflowPressureFunc` Wenn jedoch kein Speicher im Inhaltsspeicher verfügbar ist, wird der Frame, der in Kinesis Video Streams übertragen wird, mit einem Fehler (`STATUS_STORE_OUT_OF_MEMORY = 0x5200002e`) gelöscht. Weitere Informationen finden Sie unter [Von der Client-Bibliothek zurückgegebene Fehler- und Statuscodes](#). Das kann auch der Fall sein, wenn die ACKs nicht verfügbar sind oder persistente ACKs verzögert sind. In diesem Fall werden die Puffer bis zur Kapazität „Pufferdauer“ gefüllt, bevor die vorherigen Frames ausfallen.

Felder für Mitglieder

- **Version** — Ein ganzzahliger Wert, mit dem sichergestellt wird, dass die richtige Version der Struktur mit der aktuellen Version der Codebasis verwendet wird.
- **StorageType** — Eine `DEVICE_STORAGE_TYPE` Aufzählung, die die zugrunde liegende Unterstützung und Implementierung des Speichers angibt. Derzeit wird nur der Wert `DEVICE_STORAGE_TYPE_IN_MEM` unterstützt. `DEVICE_STORAGE_TYPE_HYBRID_FILE` wird in einer künftigen Implementierung unterstützt werden und angeben, dass der Speicher in den dateigestützten Inhaltsspeicher zurückfällt.
- **StorageSize** — Die Speichergröße in Byte, die vorab zugewiesen werden soll. Der kleinste zu reservierende Wert ist 10 MB, der größte 10 GB. (Dies wird sich in einer zukünftigen Implementierung des dateigestützten Inhaltsspeichers ändern.)

- `SpillRatio` — Ein ganzzahliger Wert, der den Prozentsatz des Speichers darstellt, der vom direkten Speichertyp (RAM) zugewiesen werden soll, im Gegensatz zum sekundären Überlaufspeicher (Dateispeicher). Derzeit nicht verwendet.
- `rootDirectory`: Der Pfad zu dem Verzeichnis, in dem sich der dateigestützte Inhaltsspeicher befindet. Derzeit nicht verwendet.

Kinesis-Videostream-Strukturen

Sie können die folgenden Strukturen verwenden, um Daten für eine Instanz eines Kinesis-Videostreams bereitzustellen.

Themen

- [StreamDefinition/StreamInfo](#)
- [ClientMetrics](#)
- [StreamMetrics](#)

StreamDefinition/StreamInfo

Das Objekt `StreamDefinition` auf C++-Ebene kapselt das Objekt `StreamInfo` im plattformunabhängigen Code und stellt im Konstruktor einige Standardwerte bereit.

Felder für Mitglieder

Feld	Datentyp	Beschreibung	Standardwert
<code>stream_name</code>	<code>string</code>	Ein optionaler Streamname. Weitere Informationen zur Länge des Streamnamens finden Sie unter Beschränkungen des Producer-SDK . Jeder Stream sollte einen eindeutigen Namen haben.	Wenn kein Name angegeben wird, wird ein zufälliger Name generiert.

Feld	Datentyp	Beschreibung	Standardwert
retention_period	duration<uint64_t, ratio<3600>>	Der Aufbewahrungszeitraum für den Stream in Sekunden. 0 bedeutet keine Aufbewahrung.	3600 (eine Stunde)
Tags	const map<string, string>*	Eine Map mit Schlüssel-Wert-Paaren, die Benutzerdaten enthalten. Wenn der Stream bereits einen Satz Tags aufweist, werden die neuen Tags an die vorhandenen Tags angehängt.	Keine Tags
kms_key_id	string	Die AWS KMS-Schlüssel-ID für die Verschlüsselung des Streams. Weitere Informationen finden Sie unter Datenschutz in Kinesis Video Streams .	Der KMS-Standard Schlüssel (aws/kinesis-video).
streaming_type	STREAMING_TYPE - Aufzählung	Der einzige unterstützte Wert ist STREAMING_TYPE_REALTIME.	

Feld	Datentyp	Beschreibung	Standardwert
<code>content_type</code>	<code>string</code>	Inhaltsformat des Streams. Die Kinesis Video Streams Streams-Konsole kann Inhalte in diesem <code>video/h264</code> Format wiedergeben.	<code>video/h264</code>
<code>max_latency</code>	<code>duration<uint64_t, milli></code>	Die maximale Latenz für den Stream in Millisekunden. Der Callback für Stream-Latenzdruck (sofern angegeben) wird aufgerufen, wenn die Pufferlaufzeit diesen Zeitraum übersteigt. Mit der Angabe <code>0</code> wird festgelegt, dass der Callback für Stream-Latenzdruck aufgerufen wird.	<code>milliseconds::zero()</code>

Feld	Datentyp	Beschreibung	Standardwert
<code>fragment_duration</code>	<code>duration<uint64_t></code>	<p>Die gewünschte Fragmentdauer in Sekunden. Dieser Wert wird in Kombination mit dem <code>key_frame_fragmentation</code> - Wert verwendet. Wenn dieser Wert lautet <code>false</code>, generiert Kinesis Video Streams nach Ablauf dieser Dauer Fragmente auf einem Schlüssel bild. Beispiel: In einem AAC-Audio stream ist jeder Frame ein Keyframe. Mit <code>key_frame_fragmentation = false</code> kommt es nach Ablauf dieser Dauer zur Fragmentierung eines Keyframes mit dem Ergebnis zweisekündiger Fragmente.</p>	2

Feld	Datentyp	Beschreibung	Standardwert
timecode_scale	<code>duration<uint64_t, milli></code>	Die MKV-Timecodeskala in Millisekunden, die die Granularität der Timecodes für die Frames im MKV-Cluster angibt. Der MKV-Frame-Timecode gilt immer relativ zum Beginn des Clusters. MKV verwendet einen signierten 16-Bit-Wert (0 bis 32 767) zur Darstellung des Timecodes innerhalb des Clusters (Fragment). Stellen Sie sicher, dass der Frame-Timecode mit der angegebenen Timecode-Skala dargestellt werden kann. Mit dem Standard-Timecode-Skalawert von 1 ms wird sichergestellt, dass der größte darstellbare Frame 32 767 ms \approx 32 Sekunden lang ist. Dieser Wert liegt über der maximalen Fragmentdauer von 10 Sekunden, die	1

Feld	Datentyp	Beschreibung	Standardwert
		in Service Quotas für Kinesis Video Streams angegeben ist.	
<code>key_frame_fragmentation</code>	<code>bool</code>	Gibt an, ob Fragmente für einen Keyframe erstellt werden. Wenn der Wert <code>true</code> ist, produziert das SDK jedes Mal einen Start des Fragments , wenn ein Keyframe auftritt. Wenn <code>false</code> , wartet Kinesis Video Streams mindestens darauf <code>fragment_duration</code> und erzeugt ein neues Fragment auf dem darauffolgenden Schlüsselbild.	<code>true</code>

Feld	Datentyp	Beschreibung	Standardwert
<code>frame_timecodes</code>	<code>bool</code>	Verwendung von Frame-Timecodes oder Generieren von Zeitstempeln mit dem aktuellen Zeit-Callback: Viele Encoder erzeugen keine Zeitstempel mit den Frames. Wenn Sie also diesen Parameter angeben <code>false</code> , wird sichergestellt, dass die Frames bei der Eingabe in Kinesis Video Streams mit einem Zeitstempel versehen werden.	<code>true</code>

Feld	Datentyp	Beschreibung	Standardwert
<code>absolute_fragment_times</code>	<code>bool</code>	<p>Kinesis Video Streams verwendet MKV als zugrunde liegenden Verpackungsmechanismus. In der MKV-Spezifikation müssen Frame-Timecodes relativ zum Anfang des Clusters (Fragment s) sein. Die Cluster-Timecodes können jedoch entweder absolut oder relativ zum Startzeitpunkt des Streams sein. Wenn die Zeitstempel relativ sind, verwendet der Aufruf von <code>PutMedia</code> in der Service-API den optionalen Zeitstempel für den Streamanfang und passt die Cluster-Zeitstempel entsprechend an. Der Service speichert die Fragmente immer mit den absoluten Zeitstempeln.</p>	<code>true</code>

Feld	Datentyp	Beschreibung	Standardwert
<code>fragment_acks</code>	<code>bool</code>	Ob Fragment-ACKs (Bestätigungen) auf Anwendungsebene empfangen werden sollen.	<code>true</code> bedeutet, dass das SDK ACKs empfängt und entsprechend verfährt.
<code>restart_on_error</code>	<code>bool</code>	Wird bei bestimmten Fehlern neu gestartet ?	<code>true</code> bedeutet, dass das SDK beim Auftreten eines Fehlers versucht, das Streamen neu zu starten.

Feld	Datentyp	Beschreibung	Standardwert
recalculate_metrics	bool	Gibt an, ob die Metriken neu berechnet werden. Jeder Aufruf zum Abrufen der Kennzahlen kann diese neu berechnen, um sicherzustellen, dass der aktuelle "laufende" Wert abgerufen wird. Dafür wird – in geringem Umfang – CPU-Leistung benötigt. Diese Einstellung muss auf extrem leistungs schwachen/kleinen Geräten möglicherweise auf false eingestellt werden, um CPU-Takte zu sparen. Andernfalls wird die Verwendung false für diesen Wert nicht empfohlen.	true

Feld	Datentyp	Beschreibung	Standardwert
<code>nal_adaptation_flags</code>	<code>uint32_t</code>	<p>Gibt die Anpassungs-Flags für die Netzwerk-Abstraktionsschicht (Network Abstraction Layer Unit, NALU) an. Wenn der Bitstream mit H.264 codiert ist, kann er in NALUs als Rohdaten oder Paket verarbeitet werden. Diese liegen entweder im Annex-B- oder AVCC-Format vor. Die meisten elementaren Stream-Produzenten und -Verbraucher (Lese-Encoder und -Decoder) verwenden das Annex-B-Format, da es Vorteile bietet, wie z. B. die Fehlerbehebung. Systeme auf höherer Ebene verwenden das AVCC-Format. Dieses Format ist Standard für MPEG, HLS, DASH usw. Die Konsolenausgabe verwendet die Browser-MSE (Media Source</p>	<p>Standardmäßig wird das Format Annex-B für das Format AVCC sowohl für die Framedaten als auch für die CPD (Codec Private Data) adaptiert.</p>

Feld	Datentyp	Beschreibung	Standardwert
		<p>Extensions), um den Stream im Format AVCC zu decodieren und wiederzugeben. Für H.264 (und für M-JPEG und H.265), stellt das SDK Adaptionfunktionen bereit.</p> <p>Viele einfache Streams liegen in folgendem Format vor. In diesem Beispiel ist Ab der Annex-B-Startcode (001 oder 0001).</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)... Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)</pre> </div> <p>Im Fall von H.264 ist der Codec Private Data (CPD) in den Parametern SPS (Sequence Parameter Set) und PPS (Picture Parameter Set) enthalten und kann</p>	

Feld	Datentyp	Beschreibung	Standardwert
		<p>an das AVCC-Format angepasst werden. Sofern die Medien-Pipeline die CPD nicht separat ausgibt, kann die Anwendung die CPD aus dem Frame extrahieren. Dazu sucht es nach dem ersten IDR-Frame (der SPS und PPS enthalten sollte), extrahiert die beiden NALUs (die es sind <code>Ab(Sps)</code> <code>Ab(Pps)</code>) und fügt sie in die <code>CPD</code> ein. <code>StreamDefinition</code></p> <p>Weitere Informationen finden Sie unter NAL Adaptation Flags.</p>	
frame_rate	uint32_t	Die erwartete Bildrate. Dieser Wert wird verwendet, um den Pufferbedarf exakter zu berechnen.	25

Feld	Datentyp	Beschreibung	Standardwert
avg_bandwidth_bps	uint32_t	Die erwartete durchschnittliche Bandbreite für den Stream. Dieser Wert wird verwendet, um den Pufferbedarf exakter zu berechnen.	$4 * 1\,024 * 1\,024$

Feld	Datentyp	Beschreibung	Standardwert
buffer_duration	duration<uint64_t>	Laufzeit des Streampuffers in Sekunden. Das SDK speichert die Frames bis zu einem Tag im Inhaltsspeicher. Danach werden die vorherigen Frames gelöscht. <code>buffer_duration</code> , wenn sich das Fenster weiterbewegt. Wenn der Frame, der gelöscht wird, nicht an das Backend gesendet wurde, wird der Drop-Frame-Callback aufgerufen. Wenn die aktuelle Pufferlaufzeit größer als <code>max_latency</code> ist, wird der Callback für Stream-Latenzdruck aufgerufen. Der Puffer wird bis zum nächsten Fragmentstart gekürzt, wenn das persistente Fragment-ACK empfangen wird. Dies gibt an, dass der Inhalt dauerhaft persistent in der Cloud liegt und nicht	120

Feld	Datentyp	Beschreibung	Standardwert
		mehr auf dem lokalen Gerät gespeichert werden muss.	

Feld	Datentyp	Beschreibung	Standardwert
replay_duration	duration< uint64_t>	Die Dauer in Sekunden, bis das aktuelle Lesegerät während eines Fehlers zur Wiedergabe zurückgeschaltet wird, wenn der Neustart aktiviert ist. Der Rollback stoppt am Pufferanfang (falls das Streamen gerade erst begonnen hat oder eine persistente ACK aufgetreten ist). Das Rollback versucht, auf einem Keyframe zu landen, der einen Fragments tart bezeichnet. Wenn der Fehler, der den Neustart verursacht hat, nicht auf einen ausgefallenen Host hinweist (der Host ist noch aktiv und enthält die Frame-Daten in seinen internen Puffern), wird der Rollback beim letzten empfangenen ACK-Frame beendet. Dann wird zum nächsten Keyframe	40

Feld	Datentyp	Beschreibung	Standardwert
		vor gesprungen, da das gesamte Fragment bereits im Arbeitsspeicher des Host gespeichert ist.	
connection_staleness	duration<uint64_t>	Die Zeit in Sekunden, nach der der Stream Staleness Callback aufgerufen wird, wenn das SDK das Puffer-ACK nicht empfängt. Es zeigt an, dass die Frames vom Gerät gesendet werden, das Backend sie jedoch nicht bestätigt. Dies weist auf eine getrennte Verbindung am Zwischenpunkt oder Load Balancer hin.	30
codec_id	string	Die Codec-ID für den MKV-Track.	V_MPEG4/ISO/AVC
track_name	string	Der Name des MKV-Tracks.	kinesis_video

Feld	Datentyp	Beschreibung	Standardwert
codecPrivateData	unsigned char*	Der CPD-Puffer (Codec Private Data). Wenn die Medienpipeline über die Informationen zu den CPD verfügt, bevor der Stream startet, kann dies in <code>StreamDefinition.codecPrivateData</code> festgelegt werden. Die Bits werden kopiert und der Puffer kann nach dem Aufruf zum Erstellen des Streams wiederverwendet oder freigegeben werden. Wenn die Daten jedoch bei der Erstellung des Streams nicht verfügbar sind, können sie in einer der Überladungen der Funktion festgelegt werden. <code>KinesisVideoStream.start(cpd)</code>	null
codecPrivateDataGröße	uint32_t	Die CPD-Puffergröße	0

ClientMetrics

Das ClientMetricsObjekt wird durch Aufrufen gefülltgetKinesisVideoMetrics.

Felder für Mitglieder

Feld	Datentyp	Beschreibung
Version	UINT32	Die Version der Struktur, definiert im Makro CLIENT_METRICS_CURRENT_VERSION .
contentStoreSize	UINT64	Die Inhaltsspeicher-Gesamtgröße in Byte. Dies ist der in DeviceInfo.StorageInfo.storageSize angegebene Wert.
contentStoreAvailableGröße	UINT64	Aktuell verfügbare Speichergöße in Byte.
contentStoreAllocatedGröße	UINT64	Aktuell zugewiesene Größe. Die zugewiesene plus die verfügbaren Größen sollten etwas kleiner als die Speichergesamtgröße sein, da für die interne Speicherverwaltung und die Implementierung des Inhaltsspeichers ebenfalls Speicher benötigt wird.
totalContentViewsGröße	UINT64	Die Größe des für alle Inhaltsansichten für alle Streams zugewiesenen Speichers. Dies wird nicht auf die Speichergröße angerechnet. Dieser Speicher wird

Feld	Datentyp	Beschreibung
		mit dem Makro MEMALLOC zugeordnet, das überschrieben werden kann, um einen benutzerdefinierten Allocator bereitzustellen.
totalFrameRate	UINT64	Die insgesamt über alle Streams beobachtete Bildrate.
totalTransferRate	UINT64	Die insgesamt über alle Streams beobachtete Streamrate in Byte pro Sekunde.

StreamMetrics

Das StreamMetricsObjekt wird durch Aufrufen `gefülltgetKinesisVideoMetrics`.

Felder für Mitglieder

Feld	Datentyp	Beschreibung
Version	UINT32	Die Version der Struktur, definiert im Makro <code>STREAM_METRICS_CURRENT_VERSION</code> .
currentViewDuration	UINT64	Die Dauer der akkumulierten Frames. Im Fall eines schnellen Netzwerks ist diese Dauer entweder Null oder die Framedauer (während der Frame übertragen wird). Wenn die Dauer länger als die in der <code>max_latency</code> angegebene Dauer <code>wirdStreamDef</code>

Feld	Datentyp	Beschreibung
		<p><code>initiation</code> , wird der Stream-Latenz-Callback aufgerufen, sofern er angegeben ist. Die Dauer wird in Einheiten von 100 ns – die Standardzeiteinheit für den PIC-Layer – angegeben.</p>
<p><code>overallViewDuration</code></p>	<p>UINT64</p>	<p>Die Gesamtanzeigedauer. Wenn der Stream ohne ACKs oder Persistenz konfiguriert ist, steigt dieser Wert, wenn die Frames in den Kinesis-Videostream eingefügt werden, und entspricht dem <code>buffer_duration</code> in der <code>StreamDefinition</code>. Wenn ACKs aktiviert sind und das persistente ACK empfangen wird, wird der Puffer auf das nächste Schlüsselbild gekürzt. Das liegt daran, dass der ACK-Zeitstempel den Anfang des gesamten Fragments angibt. Die Dauer wird in Einheiten von 100 ns – die Standardzeiteinheit für den PIC-Layer – angegeben.</p>
<p><code>currentViewSize</code></p>	<p>UINT64</p>	<p>Die Größe des aktuellen Puffers in Byte.</p>
<p><code>overallViewSize</code></p>	<p>UINT64</p>	<p>Die Gesamtanzeigegröße in Byte.</p>

Feld	Datentyp	Beschreibung
currentFrameRate	UINT64	Die beobachtete Bildrate für den aktuellen Stream.
currentTransferRate	UINT64	Die beobachtete Übertragungsratesrate in Byte pro Sekunde für den aktuellen Stream.

SDK-Rückrufe von Produzenten

Die Klassen und Methoden im Amazon Kinesis Video Streams Producer SDK verwalten keine eigenen Prozesse. Sie verwenden stattdessen eingehende Funktionsaufrufe und Ereignisse, um Callbacks für die Kommunikation mit der Anwendung zu planen.

Es gibt zwei Callback-Mustern, die die Anwendung für die Interaktion mit dem SDK verwenden kann:

- [CallbackProvider](#)— Dieses Objekt macht jeden Callback von der PIC-Komponente (Platform-Independent Code) für die Anwendung verfügbar. Dieses Muster ermöglicht eine umfassende Funktionalität, bedeutet jedoch auch, dass die Implementierung alle öffentlichen API-Methoden und -Signaturen auf C++-Ebene verarbeiten muss.
- [StreamCallbackProvider](#) und [ClientCallbackProvider](#) — Diese Objekte machen die stream- und clientspezifischen Callbacks verfügbar, und die C++-Schicht des SDK macht die restlichen Callbacks verfügbar. Dies ist das empfohlene Callback-Muster für die Interaktion mit dem Producer SDK.

In der folgenden Abbildung ist das Objektmodell der Callback-Objekte dargestellt:

In der vorherigen Abbildung ist `DefaultCallbackProvider` von `CallbackProvider` abgeleitet (der alle Callbacks im PIC bereitstellt) und enthält `StreamCallbackProvider` und `ClientCallbackProvider`.

Dieses Thema enthält die folgenden Abschnitte:

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks Struktur](#)

- [Callback-Implementierungen, um das Streaming erneut zu versuchen](#)

ClientCallbackProvider

Das `ClientCallbackProvider` -Objekt stellt Callback-Funktionen auf Client-Ebene bereit. Die Details der Funktionen sind im Abschnitt [ClientCallbacks](#) beschrieben.

Callback-Methoden:

- `getClientReadyCallback`— Meldet einen Bereitschaftsstatus für den Client.
- `getStorageOverflowPressureCallback`— Meldet einen Speicherüberlauf oder -druck. Dieser Callback wird aufgerufen, wenn die Speichernutzung den Wert `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, also 5 Prozent der Gesamtspeichergröße, unterschreitet. Weitere Informationen finden Sie unter [StorageInfo](#).

StreamCallbackProvider

Das `StreamCallbackProvider` -Objekt stellt Callback-Funktionen auf Stream-Ebene bereit.

Callback-Methoden:

- `getDroppedFragmentReportCallback`: Meldet ein verworfenes Fragment.
- `getDroppedFrameReportCallback`— Meldet einen verlorenen Frame.
- `getFragmentAckReceivedCallback`— Meldet, dass ein Fragment-ACK für den Stream empfangen wurde.
- `getStreamClosedCallback`— Meldet einen Zustand, in dem der Stream geschlossen ist.
- `getStreamConnectionStaleCallback`— Meldet einen veralteten Verbindungszustand. In diesem Zustand sendet der Hersteller Daten an den Dienst, erhält jedoch keine Bestätigungen.
- `getStreamDataAvailableCallback`— Meldet, dass Daten im Stream verfügbar sind.
- `getStreamErrorReportCallback`— Meldet einen Stream-Fehler.
- `getStreamLatencyPressureCallback`— Meldet einen Zustand der Stream-Latenz, wenn die kumulierte Puffergröße größer als der `max_latency` Wert ist. Weitere Informationen finden Sie unter [StreamDefinition/StreamInfo](#).
- `getStreamReadyCallback`: — Meldet einen Zustand, in dem der Stream bereit ist.
- `getStreamUnderflowReportCallback`— Meldet einen Zustand, in dem der Stream unterläuft. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.

Den Quellcode für finden Sie `StreamCallbackProvider` unter [StreamCallbackProvider.h](#).

ClientCallbacks Struktur

Die Struktur `ClientCallbacks` enthält die Callback-Funktionseingangspunkte, die der PIC aufruft, wenn spezielle Ereignisse auftreten. Die Struktur enthält auch Versionsinformationen im Feld `CALLBACKS_CURRENT_VERSION` sowie das Feld `customData` mit benutzerdefinierten Daten, die für einzelne Callback-Funktionen zurückgegeben werden.

Die Client-Anwendung kann einen `this`-Zeiger für das Feld `custom_data` verwenden, um Mitgliedsfunktionen den statischen `ClientCallback`-Funktionen während der Laufzeit zuzuweisen, wie im folgenden Codebeispiel gezeigt:

```
STATUS TestStreamCallbackProvider::streamClosedHandler(UINT64 custom_data,
    STREAM_HANDLE stream_handle, UINT64 stream_upload_handle) {
    LOG_INFO("Reporting stream stopped.");

    TestStreamCallbackProvider* streamCallbackProvider =
        reinterpret_cast<TestStreamCallbackProvider*> (custom_data);
    streamCallbackProvider->streamClosedHandler(...);
```

Ereignisse

Funktion	Beschreibung	Typ
<code>CreateDeviceFunc</code>	Derzeit nicht im Backend implementiert. Dieser Aufruf schlägt fehl, wenn er mit Java oder C++ ausgeführt wird. Andere Clients führen eine plattformspezifische Initialisierung durch.	Backend API
<code>CreateStreamFunc</code>	Wird beim Erstellen des Streams aufgerufen	Backend API
<code>DescribeStreamFunc</code>	Wird aufgerufen, wenn <code>DescribeStream</code> aufgerufen wird	Backend API

Funktion	Beschreibung	Typ
<code>GetStreamingEndpointFunc</code>	Wird aufgerufen, wenn <code>GetStreamingEndpoint</code> aufgerufen wird	Backend API
<code>GetStreamingTokenFunc</code>	Wird aufgerufen, wenn <code>GetStreamingToken</code> aufgerufen wird	Backend API
<code>PutStreamFunc</code>	Wird aufgerufen, wenn <code>PutStream</code> aufgerufen wird	Backend API
<code>TagResourceFunc</code>	Wird aufgerufen, wenn <code>TagResource</code> aufgerufen wird	Backend API
<code>CreateMutexFunc</code>	Erstellt einen Synchronisierungs-Mutex.	Synchronisierung
<code>FreeMutexFunc</code>	Gibt den Mutex frei.	Synchronisierung
<code>LockMutexFunc</code>	Sperrt den Synchronisierungs-Mutex.	Synchronisierung
<code>TryLockMutexFunc</code>	Versucht den Mutex zu sperren. Derzeit nicht implementiert.	Synchronisierung
<code>UnlockMutexFunc</code>	Hebt die Sperrung des Mutex auf.	Synchronisierung
<code>ClientReadyFunc</code>	Wird aufgerufen, wenn der Client bereit ist.	Benachrichtigung

Funktion	Beschreibung	Typ
<code>DroppedFrameReportFunc</code>	Meldet, wenn ein Frame verworfen wird.	Benachrichtigung
<code>DroppedFragmentReportFunc</code>	Meldet, wenn ein Fragment verworfen wird. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Benachrichtigung
<code>FragmentAckReceivedFunc</code>	Wird aufgerufen, wenn eine Fragment-ACK (Puffern, Empfang, persistent und Fehler) empfangen wird.	Benachrichtigung
<code>StorageOverflowPressureFunc</code>	Wird aufgerufen, wenn die Speichernutzung den Wert <code>STORAGE_PRESSURE_NOTIFICATION_THRESHOLD</code> , also 5 Prozent der Gesamtspeichergröße, unterschreitet.	Benachrichtigung
<code>StreamClosedFunc</code>	Wird aufgerufen, wenn die letzten Bits der verbleibenden Frames gestreamt werden.	Benachrichtigung
<code>StreamConnectionStaleFunc</code>	Wird aufgerufen, wenn der Stream veraltet ist. In diesem Fall sendet der Produzent Daten an den Service, empfängt jedoch keine Bestätigungen.	Benachrichtigung
<code>StreamDataAvailableFunc</code>	Wird aufgerufen, wenn Stream-Daten verfügbar sind.	Benachrichtigung

Funktion	Beschreibung	Typ
StreamErrorReportFunc	Wird aufgerufen, wenn ein Stream-Fehler auftritt. Der PIC schließt den Stream in diesem Fall automatisch.	Benachrichtigung
StreamLatencyPressureFunc	Wird aufgerufen, wenn der Stream latent ist, d. h. die kumulierte Puffergröße den Wert <code>max_latency</code> überschreitet. Weitere Informationen finden Sie unter StreamDefinition/StreamInfo .	Benachrichtigung
StreamReadyFunc	Wird aufgerufen, wenn der Stream bereit ist.	Benachrichtigung
StreamUnderflowReportFunc	Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Benachrichtigung
DeviceCertToTokenFunc	Gibt das Verbindungszertifikat als Token zurück.	Plattformintegration
GetCurrentTimeFunc	Gibt die aktuelle Zeit zurück.	Plattformintegration
GetDeviceCertificateFunc	Gibt das Gerätezertifikat zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration

Funktion	Beschreibung	Typ
<code>GetDeviceFingerprintFunc</code>	Gibt die Geräteidentifikation zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration
<code>GetRandomNumberFunc</code>	Gibt eine zufällige Zahl zwischen 0 und <code>RAND_MAX</code> zurück.	Plattformintegration
<code>GetSecurityTokenFunc</code>	Gibt das Sicherheitstoken zurück, das an die Funktionen übergeben wurde, die mit der Backend-API kommunizieren. Die Implementierung kann die serialisierten Werte für <code>AccessKeyId</code> , <code>SecretKeyId</code> und das Sitzungstoken angeben.	Plattformintegration
<code>LogPrintFunc</code>	Protokolliert eine Textzeile mit dem Tag und der Protokoll ebene. Weitere Informationen finden Sie unter <code>PlatformUtils.h</code> .	Plattformintegration

Für die Plattformintegrationsfunktionen aus der vorherigen Tabelle ist der letzte Parameter eine `ServiceCallContext`-Struktur mit den folgenden Feldern:

- `version`: Die Version der Struktur
- `callAfter`: Eine absolute Zeit, nach der die Funktion aufgerufen wird
- `timeout`: Die Zeitüberschreitung der Operation in Einheiten von 100 Nanosekunden
- `customData`: Ein benutzerdefinierter Wert, der an den Client übergeben wird

- `pAuthInfo`: Die Anmeldeinformationen für den Aufruf. Weitere Informationen finden Sie in der folgenden (`__AuthInfo`)-Struktur.

Die Autorisierungsinformationen werden mit der Struktur `__AuthInfo` bereitgestellt. Diese kann entweder serialisierte Anmeldeinformationen oder ein anbieterspezifisches Authentifizierungstoken enthalten. Diese Struktur enthält die folgenden Felder:

- `version`: Die Version der `__AuthInfo`-Struktur
- `type`: Ein `AUTH_INFO_TYPE`-Wert, über den der Typ der Anmeldeinformationen (Zertifikat oder Sicherheitstoken) definiert wird
- `data`: Ein Byte-Array mit den Authentifizierungsinformationen
- `size`: Die Größe des `data`-Parameters
- `expiration`: Die Ablaufdauer der Anmeldeinformationen in Einheiten von 100 Nanosekunden

Callback-Implementierungen, um das Streaming erneut zu versuchen

Das Kinesis Video Producer-SDK gibt über Callback-Funktionen den Streaming-Status an. Es wird empfohlen, die folgenden Callback-Mechanismen zu implementieren, um bei vorübergehenden Netzwerkproblemen, die während des Streamings aufgetreten sind, die Wiederherstellung zu gewährleisten.

- **Stream-Latenzdruck-Callback** — Dieser Callback-Mechanismus wird ausgelöst, wenn das SDK auf eine Stream-Latenz stößt. Dies geschieht, wenn die kumulierte Puffergröße größer ist als der `MAX_LATENCY`-Wert. Wenn der Stream erstellt wurde, legt die Streaming-Anwendung `MAX_LATENCY` auf den Standardwert von 60 Sekunden fest. Die typische Implementierung für diesen Callback besteht darin, die Verbindung zurückzusetzen. Sie können die Beispielimplementierung unter <https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/.c> nach Bedarf verwenden. `StreamLatencyStateMachine` Beachten Sie, dass es keine Möglichkeit gibt, Frames, die aufgrund eines Netzerkausfalls nicht zugestellt wurden, in einem sekundären Speicher zu speichern, um sie wieder aufzufüllen.
- **Stream-Staleness-Callback** — Dieser Callback wird ausgelöst, wenn der Producer Daten an den Amazon Kinesis Data Streams Streams-Service (Uplink) senden kann, aber die Bestätigungen (gepufferte ACK) nicht rechtzeitig zurückholen kann (Standard ist 60 Sekunden). Abhängig von den Netzwerkeinstellungen kann entweder der Stream-Latenzdruck-Callback oder der Stream-Staleness-Callback oder beide initiiert werden. Ähnlich wie bei der

Wiederholungsversuchimplementierung des Callbacks für Stream-Latenzdruck besteht die typische Implementierung darin, die Verbindung zurückzusetzen und eine neue Verbindung für das Streaming zu starten. Sie können die Beispielimplementierung unter <https://github.com/aws-labs/amazon-kinesis-video-streams-connection-state-state-machine-producer-c/blob/master/src/source/.c> nach Bedarf verwenden.

- Stream-Fehler-Callback — Dieser Callback wird ausgelöst, wenn das SDK beim Aufruf der KVS-API-Serviceaufrufen auf einen Timeout bei der Netzwerkverbindung oder auf andere Fehler stößt.
- Abgebrochener Frame-Callback — dieser Callback wird ausgelöst, wenn der Speicherplatz voll ist, entweder aufgrund einer langsamen Netzwerkgeschwindigkeit oder eines Streamfehlers. Wenn die Netzwerkgeschwindigkeit dazu führt, dass Frames verloren gehen, können Sie entweder die Speichergröße erhöhen, die Videobildgröße reduzieren oder die Bildrate an die Netzwerkgeschwindigkeit anpassen.

Parser-Bibliothek für Kinesis-Videostreams

Die Kinesis-Videostream-Parser-Bibliothek besteht aus einer Reihe von Tools, die Sie in Java-Anwendungen verwenden können, um die MKV-Daten in einem Kinesis-Videostream zu verarbeiten.

Die Bibliothek enthält die folgenden Klassen:

- [StreamingMkvReader](#): Diese Klasse liest bestimmte MKV-Elemente aus Videostreams aus.
- [FragmentMetadataVisitor](#): Diese Klasse ruft die Metadaten von Fragmenten (Medienelementen) und Tracks (einzelnen Datenstreams mit Medieninformationen wie Audiodaten oder Untertiteln) ab.
- [OutputSegmentMerger](#): Diese Klasse fügt aufeinanderfolgende Fragmente oder Datenblöcke in einem Videostream zusammen.
- [KinesisVideoExample](#): Dies ist eine Beispielanwendung, die zeigt, wie Sie die Kinesis-Videostream-Parser-Bibliothek verwenden.

Die Bibliothek umfasst auch Tests, die die Verwendung der Tools verdeutlichen.

Verfahren: Verwenden der Kinesis-Videodatenstrom-Parserbibliothek

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- [the section called “Schritt 1: Herunterladen und Konfigurieren des Codes”](#).
- [the section called “Schritt 2: Schreiben und Untersuchen des Datenstroms”](#).
- [the section called “Schritt 3: Ausführen und Überprüfen des Codes”](#).

Voraussetzungen

Sie müssen über Folgendes verfügen, um die Kinesis-Videostream-Parser-Bibliothek untersuchen und verwenden zu können:

- Ein Amazon Web Services (AWS) -Konto. Wenn Sie noch kein -Konto haben, schauen Sie nach [the section called “Melden Sie sich an für ein AWS-Konto”](#).
- Eine integrierte Java-Entwicklungsumgebung (IDE) wie [Eclipse Java Neon](#) oder [JetBrains IntelliJ Idea](#).

Schritt 1: Herunterladen und Konfigurieren des Codes

In diesem Abschnitt laden Sie die Java-Bibliothek und den Testcode herunter und importieren das Projekt in die Java-IDE.

Die Voraussetzungen und andere Details zu diesem Verfahren finden Sie unter ["Stream Parser"-Bibliothek](#).

1. Erstellen Sie ein Verzeichnis und klonen Sie den Quellcode der Bibliothek aus dem GitHub Repository (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>).

```
$ git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Öffnen Sie die Java-IDE, die Sie verwenden (z. B. [Eclipse](#) oder [IntelliJ IDEA](#)) und importieren Sie das heruntergeladene Apache Maven-Projekt:
 - In Eclipse: Klicken Sie auf File, Import..., Maven, Existing Maven Projects und navigieren Sie zum Ordner `kinesis-video-streams-parser-lib`.
 - In IntelliJ Idea: Klicken Sie auf Import. Navigieren Sie zur Datei `pom.xml` im Stammverzeichnis des heruntergeladenen Pakets.

Weitere Informationen finden Sie in der Dokumentation zur IDE.

Nächster Schritt

[the section called "Schritt 2: Schreiben und Untersuchen des Datenstroms"](#).

Schritt 2: Schreiben und Untersuchen des Datenstroms

In diesem Abschnitt prüfen Sie die Java-Bibliothek sowie den Testcode und lernen, wie Sie die Klassen der Bibliothek in eigenem Code verwenden können.

Die Kinesis-Videostream-Parser-Bibliothek enthält die folgenden Tools:

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)

- [KinesisVideoExample](#)

StreamingMkvReader

Diese Klasse liest bestimmte MKV-Elemente aus einem Stream blockierungsfrei.

Das folgende Beispiel (aus der Datei `FragmentMetadataVisitorTest`) zeigt, wie Sie ein `StreamingMkvReader`-Objekt erstellen und verwenden können, um `MkvElement`-Objekte aus dem Eingabe-Stream `inputStream` zu extrahieren:

```
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new
InputStreamParserByteSource(inputStream));
while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        ...
    }
}
```

FragmentMetadataVisitor

Diese Klasse ruft Metadaten für Fragmente (Medienelemente) ab und verfolgt einzelne Datenströme, die Medieninformationen enthalten, wie z. B. private Codec-Daten, Pixelbreite oder Pixelhöhe.

Das folgende Codebeispiel (aus der Datei `FragmentMetadataVisitorTest`) zeigt, wie Sie mit `FragmentMetadataVisitor` Daten aus einem `MkvElement`-Objekt auslesen können:

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));
int segmentCount = 0;
while(mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
{
```


- ID des zur Codierung des Frames verwendeten Codecs
- Richtige Position des Frames im Stream. Stellen Sie sicher, dass die Titelnummer des vorherigen Frames, falls vorhanden, niedriger ist als die des aktuellen Frames.

Um `FragmentMetadataVisitor` im Projekt zu verwenden, übergeben Sie `MkvElement`-Objekte mit deren `accept`-Methode an das Visitor-Objekt:

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

Diese Klasse führt die Metadaten verschiedener Tracks im Stream zu einem Stream mit einem einzigen Segment zusammen.

Das folgende Codebeispiel (aus der Datei `FragmentMetadataVisitorTest`) zeigt, wie Sie mit einem `OutputSegmentMerger`-Objekt die Track-Metadaten im Byte-Array `inputBytes` zusammenführen:

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

OutputSegmentMerger outputSegmentMerger =
    OutputSegmentMerger.createDefault(outputStream);

CompositeMkvElementVisitor compositeVisitor =
    new TestCompositeVisitor(fragmentVisitor, outputSegmentMerger);

final InputStream in = TestResourceUtil.getTestInputStream("output_get_media.mkv");

StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));

while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(compositeVisitor);
        if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
    {
```

```
MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
Assert.assertTrue(frame.getFrameData().limit() > 0);
MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
    assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
}
}
```

Das vorhergehende Beispiel weist folgendes Codierungsmuster auf:

- Erstellen Sie [FragmentMetadataVisitor](#), um die Metadaten aus dem Stream abzurufen.
- Ein Ausgabe-Stream für die zusammengeführten Metadaten wird erstellt.
- Erstellen Sie `OutputSegmentMerger`, übergeben Sie dabei `ByteArrayOutputStream`.
- Erstellen Sie das `CompositeMkvElementVisitor`-Objekt, das die beiden Besucher enthält.
- Erstellen Sie einen `InputStream`, der auf die angegebene Datei zeigt.
- Die einzelnen Elemente in den Eingabedaten werden im Ausgabe-Stream zusammengeführt.

KinesisVideoExample

Dies ist eine Beispielanwendung, die zeigt, wie Sie die Kinesis-Videostream-Parser-Bibliothek verwenden.

Diese Klasse führt die folgenden Operationen aus:

- Erstellt einen Kinesis-Videostream. Wenn ein Stream mit dem angegebenen Namen bereits vorhanden ist, wird der Stream gelöscht und neu erstellt.
- Ruft [PutMedia](#) auf, um Videofragmente in den Kinesis-Videostream zu streamen.
- Aufrufe [GetMedia](#), um Videofragmente aus dem Kinesis-Videostream zu streamen.
- Verwendet einen [StreamingMkvReader](#) zum Analysieren der zurückgegebenen Fragmente auf dem Stream und verwendet [FragmentMetadataVisitor](#) für die Protokollierung der Fragmente.

Löschen und Neuerstellen des Streams

Das folgende Codebeispiel (aus der `StreamOps.java` Datei) löscht einen bestimmten Kinesis-Videostream:

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

Das folgende Codebeispiel (aus der `StreamOps.java` Datei) erstellt einen Kinesis-Videostream mit dem angegebenen Namen:

```
amazonKinesisVideo.createStream(new CreateStreamRequest().withStreamName(streamName)
    .withDataRetentionInHours(DATA_RETENTION_IN_HOURS)
    .withMediaType("video/h264"));
```

Ruf PutMedia

Das folgende Codebeispiel (aus der `PutMediaWorker.java` Datei) ruft den Stream [PutMedia](#) auf:

```
putMedia.putMedia(new PutMediaRequest().withStreamName(streamName)
    .withFragmentTimecodeType(FragmentTimecodeType.RELATIVE)
    .withProducerStartTimestamp(new Date())
    .withPayload(inputStream), new PutMediaAckResponseHandler() {
    ...
});
```

Ruf GetMedia

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) ruft den Stream [GetMedia](#) auf:

```
GetMediaResult result = videoMedia.getMedia(new
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

Analysieren Sie das Ergebnis GetMedia

Dieser Abschnitt beschreibt, wie [StreamingMkvReader](#), [FragmentMetadataVisitor](#) und `CompositeMkvElementVisitor` zur Analyse, zum Speichern in die Datei und zur Protokollierung der von `GetMedia` zurückgegebenen Daten verwendet wird.

Lesen Sie die Ausgabe von `GetMedia` mit `StreamingMkvReader`

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) erstellt eine [StreamingMkvReader](#) und verwendet sie, um das Ergebnis der [GetMedia](#) Operation zu analysieren:


```

StreamingMkvReader mkvStreamReader = StreamingMkvReader.createDefault(new
    InputStreamParserByteSource(result.getPayload()));
log.info("StreamingMkvReader created for stream {} ", streamName);
try {
    mkvStreamReader.apply(this.elementVisitor);
} catch (MkvElementVisitException e) {
    log.error("Exception while accepting visitor {}", e);
}

```

Im vorigen Codebeispiel ruft der [StreamingMkvReader](#) MKVElement-Objekte aus der Nutzlast des Ergebnisses von `GetMedia` ab. Im nächsten Abschnitt werden die Elemente einem [FragmentMetadataVisitor](#) übergeben.

Rufen Sie Fragmente ab mit `FragmentMetadataVisitor`

Die folgenden Codebeispiele (aus den Dateien `KinesisVideoExample.java` und `StreamingMkvReader.java`) erstellen einen [FragmentMetadataVisitor](#). Die `MkvElement`-Objekte, die von dem [StreamingMkvReader](#) durchlaufen werden, werden dann über die Methode `accept` dem Besucher übergeben.

von `KinesisVideoExample.java`:

```

FragmentMetadataVisitor fragmentMetadataVisitor = FragmentMetadataVisitor.create();

```

von `StreamingMkvReader.java`:

```

if (mkvElementOptional.isPresent()) {
    //Apply the MkvElement to the visitor
    mkvElementOptional.get().accept(elementVisitor);
}

```

Die Elemente schreiben und in eine Datei schreiben

Das folgende Code-Beispiel (aus der `KinesisVideoExample.java`-Datei) erstellt die folgenden Objekte und gibt sie als Teil des Rückgabewerts der `GetMediaProcessingArguments`-Funktion zurück:

- Einen `LogVisitor` (eine Erweiterung von `MkvElementVisitor`), die in das Systemprotokoll schreibt.
- Einen `OutputStream`, der die eingehenden Daten in eine MKV-Datei schreibt.
- Einen `BufferedOutputStream`, der die für `OutputStream` gebundenen Daten puffert.
- Einen [the section called “OutputSegmentMerger”](#), der aufeinanderfolgende Elemente in das `GetMedia`-Ergebnis einfügt, mit denselben Track- und EBML-Daten.
- Ein `CompositeMkvElementVisitor`, das den [FragmentMetadataVisitor](#) [the section called “OutputSegmentMerger”](#), und zu einem einzigen Element `LogVisitor` zusammensetzt: Besucher.

```
//A visitor used to log as the GetMedia stream is processed.
    LogVisitor logVisitor = new LogVisitor(fragmentMetadataVisitor);

    //An OutputSegmentMerger to combine multiple segments that share track and ebml
    metadata into one
    //mkv segment.
    OutputStream fileOutputStream =
Files.newOutputStream(Paths.get("kinesis_video_example_merged_output2.mkv"),
        StandardOpenOption.WRITE, StandardOpenOption.CREATE);
    BufferedOutputStream outputStream = new BufferedOutputStream(fileOutputStream);
    OutputSegmentMerger outputSegmentMerger =
OutputSegmentMerger.createDefault(outputStream);

    //A composite visitor to encapsulate the three visitors.
    CompositeMkvElementVisitor mkvElementVisitor =
        new CompositeMkvElementVisitor(fragmentMetadataVisitor,
outputSegmentMerger, logVisitor);

    return new GetMediaProcessingArguments(outputStream, logVisitor,
mkvElementVisitor);
```

Die Argumente für die Medienverarbeitung werden dann an die übergeben `GetMediaWorker`, die wiederum an die übergeben wird `ExecutorService`, die den Worker in einem separaten Thread ausführt:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
```

```
getMediaProcessingArgumentsLocal.getMkvElementVisitor());  
executorService.submit(getMediaWorker);
```

Nächster Schritt

[the section called “Schritt 3: Ausführen und Überprüfen des Codes”](#)

Schritt 3: Ausführen und Überprüfen des Codes

Die Kinesis-Videostream-Parser-Bibliothek enthält Tools, die Sie in Ihren eigenen Projekten verwenden können. Das Projekt enthält Komponententests für die Tools, die Sie ausführen können, um die Installation zu überprüfen.

Die folgenden Komponententests sind in der Bibliothek enthalten:

- mkv
 - ElementSizeAndOffsetVisitorTest
 - MkvValueTest
 - StreamingMkvReaderTest
- Dienstprogramme
 - FragmentMetadataVisitorTest
 - OutputSegmentMergerTest

Beispiele für Amazon Kinesis Video Streams

Die folgenden Codebeispiele veranschaulichen, wie Sie mit der API von Kinesis Video Streams arbeiten:

Beispiele: Senden von Daten an Kinesis Video Streams

- [Beispiel: Kinesis Video Streams Producer SDK GStreamer-Plugin](#): Zeigt, wie das Kinesis Video Streams Producer SDK erstellt wird, das als GStreamer-Ziel verwendet werden soll.
- [Ausführen des GStreamer-Elements in einem Docker-Container](#): Zeigt, wie Sie ein vorgefertigtes Docker-Image verwenden, um RTSP-Video von einer IP-Kamera an Kinesis Video Streams zu senden.
- [Beispiel: Streaming aus einer RTSP-Quelle](#): Zeigt, wie Sie Ihr eigenes Docker-Image erstellen und RTSP-Video von einer IP-Kamera an Kinesis Video Streams senden.
- [Beispiel: Senden von Daten an Kinesis Video Streams mithilfe der PutMedia API](#): Zeigt, wie Sie mithilfe der Daten [Verwenden der Java Producer-Bibliothek](#) an Kinesis Video Streams senden, die sich bereits in einem Container-Format (MKV) mit der [PutMedia-API](#) befinden.

Beispiele: Abrufen von Daten aus Kinesis Video Streams

- [KinesisVideoExample](#): Zeigt, wie Videofragmente mit der Kinesis Video Streams Parser Library analysiert und protokolliert werden.
- [Beispiel: Analysieren und Rendern von Kinesis-Video-Streams-Fragmenten](#): Zeigt, wie Kinesis-Videostreamfragmente mit [JCodec](#) und [JFrame](#) analysiert und gerendert werden.
- [the section called "SageMaker"](#): Zeigt eine Lösung an, die Amazon verwendet, SageMaker um zu bestimmen, wann bestimmte Objekte in einem Videostream angezeigt werden.

Beispiele: Wiedergabe von Videodaten

- [Beispiel: Verwenden Sie HLS in HTML und JavaScript](#): Zeigt, wie Sie eine HLS-Streaming-Sitzung für einen Kinesis-Videostream abrufen und auf einer Webseite wiedergeben.

Voraussetzungen

- Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei festgelegt haben, oder indem Sie Anmeldeinformationen in den Java-Systemeigenschaften Ihrer integrierten Entwicklungsumgebung (IDE) bereitstellen. Wenn Sie dies noch nicht getan haben, richten Sie zunächst Ihre -Anmeldeinformationen ein. Weitere Informationen finden Sie unter [Einrichten von AWS Anmeldeinformationen und Region für die Entwicklung](#).
- Wir empfehlen, eine der folgenden Java-IDEs zum Bearbeiten und Ausführen des Codes zu verwenden:
 - [Eclipse Java Neon](#)
 - [JetBrains IntelliJ IDEA](#)

Beispiel: Kinesis Video Streams Producer SDK GStreamer-Plugin

In diesem Thema wird beschrieben, wie Sie das Amazon Kinesis Video Streams Producer SDK erstellen, das als GStreamer-Plugin verwendet werden soll.

Themen

- [Herunterladen, Erstellen und Konfigurieren des GStreamer-Elements](#)
- [Ausführen des GStreamer-Elements](#)
- [Beispiel für GStreamer-Startbefehle](#)
- [Ausführen des GStreamer-Elements in einem Docker-Container](#)
- [Referenz zu GStreamer-Elementparametern](#)

[GStreamer](#) ist ein beliebtes Medien-Framework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medienpipelines durch die Kombination modularer Plugins zu erstellen. Das GStreamer-Plugin von Kinesis Video Streams optimiert die Integration Ihrer vorhandenen GStreamer-Medienpipeline mit Kinesis Video Streams. Nach der Integration von GStreamer können Sie Videos von einer Webcam oder Real Time Streaming Protocol (RTSP)-Kamera an Kinesis Video Streams streamen, um Wiedergabe, Speicherung und weitere Analysen in Echtzeit oder später zu ermöglichen.

Das GStreamer-Plugin verwaltet automatisch die Übertragung Ihres Videostreams an Kinesis Video Streams, indem es die Funktionalität des Kinesis Video Streams Producer SDK in einem

GStreamer-Sink-Element, , kapseltkvssink. Das GStreamer-Framework stellt eine verwaltete Standardumgebung für das Erstellen des Medienflusses von einem Gerät wie einer Kamera auf eine andere Videoquelle bereit, wo der Stream dann weiter verarbeitet, gerendert oder gespeichert werden kann.

Die GStreamer-Pipeline besteht in der Regel aus einem Link zwischen einer Quelle (Videokamera) und dem Sink-Element (entweder einem Player zum Rendern des Videos oder einem Speicher für den Offline-Abruf). In diesem Beispiel verwenden Sie das Producer SDK-Element als Sink bzw. Medienziel für die Videoquelle (Webcam oder IP-Kamera). Das Plugin-Element, das das SDK kapselt, sendet dann den Videostream an Kinesis Video Streams.

In diesem Thema wird beschrieben, wie Sie eine GStreamer-Medienpipeline erstellen, die Videos aus einer Videoquelle streamen kann, z. B. einer Webkamera oder einem RTSP-Stream, die in der Regel über Zwischenkodierungsphasen (mit H.264-Codierung) mit Kinesis Video Streams verbunden ist. Wenn Ihr Videostream als Kinesis-Videostream verfügbar ist, können Sie die Kinesis Video Stream Parser Library für die weitere Verarbeitung, Wiedergabe, Speicherung oder Analyse Ihres Videostreams verwenden.

Herunterladen, Erstellen und Konfigurieren des GStreamer-Elements

Das GStreamer-Plugin-Beispiel ist im Kinesis Video Streams C++ Producer SDK enthalten. Weitere Informationen zu den Voraussetzungen und Downloadoptionen für das SDK finden Sie unter [Schritt 1: Downloaden und konfigurieren Sie den Code der C++ Producer Library](#).

Sie können den GStreamer-Sink des Producer SDKs als dynamische Bibliothek auf macOS, Ubuntu, Raspberry Pi oder Windows erstellen. Das GStreamer-Plugin befindet sich in Ihrem `build`-Verzeichnis. Um dieses Plugin zu laden, muss es sich in Ihrem `GST_PLUGIN_PATH` befinden. Führen Sie den folgenden Befehl aus:

```
export GST_PLUGIN_PATH=`pwd`/build
```

Note

Unter macOS können Sie Videos nur von einer Netzwerkkamera streamen, wenn Sie GStreamer in einem Docker-Container ausführen. Video-Streaming von einer USB-Kamera auf macOS in einem Docker-Container wird nicht unterstützt.

Ausführen des GStreamer-Elements

Um GStreamer mit dem Kinesis Video Streams Producer SDK-Element als Senke auszuführen, verwenden Sie den `gst-launch-1.0` Befehl. Verwenden Sie Einstellungen, die für das GStreamer Plugin geeignet sind. Zum Beispiel [v4l2src](#) für v4l2-Geräte auf Linux-Systemen oder [rtspsrc](#) für RTSP-Geräte. Geben Sie `kvssink` als Sink (Endziel der Pipeline) an, um Videos an das Producer SDK zu senden.

Das Element `kvssink` hat die folgenden erforderlichen Parameter:

- **stream-name** – Der Name des Ziel-Kinesis-Videostreams.
- **storage-size** – Die Speichergröße des Geräts in Kilobyte. Weitere Informationen zum Konfigurieren von Speichergeräten finden Sie unter [StorageInfo](#).
- **access-key** – Der AWS Zugriffsschlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird. Sie müssen entweder diesen Parameter oder `credential-path` angeben.
- **secret-key** – Der AWS geheime Schlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird. Sie müssen entweder diesen Parameter oder `credential-path` angeben.
- **credential-path** – Ein Pfad zu einer Datei, die Ihre Anmeldeinformationen für den Zugriff auf Kinesis Video Streams enthält. Weitere Informationen zum Rotieren von Anmeldeinformationen finden Sie unter [Verwalten von Zugriffsschlüsseln für IAM-Benutzer](#). Sie müssen entweder diesen Parameter oder `access-key` und `secret-key` angeben.

Weitere Informationen zu optionalen `kvssink`-Parametern finden Sie unter [Referenz zu GStreamer-Elementparametern](#).

Die neuesten Informationen zu GStreamer-Plugins und -Parametern finden Sie unter [GStreamer-Plugins](#) oder verwenden Sie den folgenden Befehl, um Optionen aufzulisten:

```
gst-inspect-1.0 kvssink
```

Wenn der Build fehlgeschlagen ist oder `GST_PLUGIN_PATH` nicht richtig festgelegt ist, sieht Ihre Ausgabe wie folgt aus:

```
No such element or plugin 'kvssink'
```

Beispiel für GStreamer-Startbefehle

Die folgenden Beispiele zeigen, wie Sie ein GStreamer-Plugin verwenden, um Videos von verschiedenen Arten von Geräten zu streamen.

Beispiel 1: Video von einer RTSP-Kamera auf Ubuntu streamen

Mit dem folgenden Befehl wird eine GStreamer-Pipeline auf Ubuntu erstellt, die von einer RTSP-Netzwerkkamera mithilfe des [rtspsrc](#)-GStreamer-Plugins streamt:

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Beispiel 2: Kodieren und Streamen von Videos von einer USB-Kamera auf Ubuntu

Der folgende Befehl erstellt eine GStreamer-Pipeline auf Ubuntu, die den Stream von einer USB-Kamera im H.264-Format kodiert und an Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer-Plugin [v4l2src](#) verwendet.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-
raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc bframes=0 key-int-max=45
  bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink
  stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
  key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 3: Streamen von vorkodiertem Video von einer USB-Kamera auf Ubuntu

Der folgende Befehl erstellt eine GStreamer-Pipeline auf Ubuntu, die Videos streamt, die die Kamera bereits im H.264-Format an Kinesis Video Streams codiert hat. In diesem Beispiel wird das GStreamer-Plugin [v4l2src](#) verwendet.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-
h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512
  access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 4: Video von einer Netzwerkkamera unter macOS streamen

Der folgende Befehl erstellt eine GStreamer-Pipeline unter macOS, die Videos von einer Netzwerkkamera an Kinesis Video Streams streamt. In diesem Beispiel wird das [rtspsrc](#)-GStreamer-Plugin verwendet.


```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-
name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 5: Streamen von Videos von einer Netzwerkkamera unter Windows

Der folgende Befehl erstellt eine GStreamer-Pipeline unter Windows, die Videos von einer Netzwerkkamera an Kinesis Video Streams streamt. In diesem Beispiel wird das [rtspsrc](#)-GStreamer-Plugin verwendet.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-
name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 6: Video von einer Kamera auf Raspberry Pi streamen

Der folgende Befehl erstellt eine GStreamer-Pipeline auf Raspberry Pi, die Videos an Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer-Plugin [v4l2src](#) verwendet.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert !
  video/x-raw,format=I420,width=640,height=480,framerate=30/1 !
  omxh264enc control-rate=1 target-bitrate=5120000 periodicity-
idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-
format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink
  stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"
  aws-region="YourAWSRegion"
```

Beispiel 7: Video von einer Kamera auf Raspberry Pi streamen und Region angeben

Der folgende Befehl erstellt eine GStreamer-Pipeline auf Raspberry Pi, die Videos an Kinesis Video Streams in der Region USA Ost (Nord-Virginia) streamt. In diesem Beispiel wird das GStreamer-Plugin [v4l2src](#) verwendet.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert !
  video/x-raw,format=I420,width=640,height=480,framerate=30/1 !
  omxh264enc control-rate=1 target-bitrate=5120000 periodicity-
idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-
format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink
```

```
stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"  
aws-region="YourAWSRegion"
```

Beispiel 8: Streamen von Audio und Video in Raspberry Pi und Ubuntu

Erfahren Sie, wie Sie den [gst-launch-1.0-Befehl ausführen, um das Streamen von Audio und Video in Raspberry Pi und Ubuntu zu starten](#).

Beispiel 9: Streamen von Audio und Video in macOS

Erfahren Sie, wie Sie den [gst-launch-1.0-Befehl ausführen, um das Streamen von Audio und Video in MacOS zu starten](#).

Beispiel 10: Hochladen einer MKV-Datei, die sowohl Audio als auch Video enthält

Erfahren Sie, wie Sie den [Befehl gst-launch-1.0 ausführen, um eine MKV-Datei hochzuladen, die sowohl Audio als auch Video enthält](#).

Ausführen des GStreamer-Elements in einem Docker-Container

Docker ist eine Plattform für die Entwicklung, Bereitstellung und das Ausführen von Anwendungen mithilfe von Containern. Die Verwendung von Docker zum Erstellen der GStreamer-Pipeline standardisiert die Betriebsumgebung für Kinesis Video Streams, wodurch die Erstellung und Verwendung der Anwendung optimiert wird.

So installieren und konfigurieren Sie Docker:

- [Anleitung zum Herunterladen von Docker](#)
- [Erste Schritte mit Docker](#)

Nach der Installation von Docker können Sie das Kinesis Video Streams C++ Producer SDK (und das GStreamer-Plugin) mit dem `docker pull` Befehl aus der Amazon Elastic Container Registry herunterladen.

Gehen Sie wie folgt vor, um GStreamer mit dem Kinesis Video Streams Producer SDK-Element als Senke in einem Docker-Container auszuführen:

Themen

- [Authentifizieren Ihres Docker-Clients](#)
- [Herunterladen des Docker-Images für Ubuntu, macOS, Windows oder Raspberry Pi](#)

- [Ausführen des Docker-Images](#)

Authentifizieren Ihres Docker-Clients

Authentifizieren Sie Ihren Docker-Client bei der Amazon-ECR-Registrierung, aus der Sie Ihr Image abrufen möchten. Sie müssen Authentifizierungstoken für jede verwendete Registrierung erhalten. Token sind 12 Stunden gültig. Weitere Informationen finden Sie unter [Registry-Authentifizierung](#) im Benutzerhandbuch zu Amazon-Elastic-Container-Registry.

Example : Authentifizieren mit Amazon ECR

```
aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin
https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

Bei erfolgreicher Ausführung wird als Ausgabe `Login Succeeded` gedruckt.

Herunterladen des Docker-Images für Ubuntu, macOS, Windows oder Raspberry Pi

Verwenden Sie zum Herunterladen des Docker-Images in Ihre Docker-Umgebung abhängig vom verwendeten Betriebssystem einen der folgenden Befehle:

Herunterladen des Docker-Images für Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-amazon-linux:latest
```

Herunterladen des Docker-Images für macOS

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-amazon-linux:latest
```

Herunterladen des Docker-Images für Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-
cpp-amazon-windows:latest
```

Herunterladen des Docker-Images für Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-raspberry-pi:latest
```

Verwenden Sie den folgenden Befehl, um zu überprüfen, dass das Image erfolgreich hinzugefügt wurde:

```
docker images
```

Ausführen des Docker-Images

Führen Sie abhängig vom verwendeten Betriebssystem einen der folgenden Befehle aus, um das Docker-Image auszuführen:

Ausführen des Docker-Images auf Ubuntu

```
sudo docker run -it --network="host" --device=/dev/video0 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Ausführen des Docker-Images unter macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Ausführen des Docker-Images unter Windows

```
docker run -it 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-windows <AWS_ACCESS_KEY_ID> <AWS_SECRET_ACCESS_KEY> <RTSP_URL> <STREAM_NAME>
```

Ausführen des Docker-Images auf Raspberry Pi

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi /bin/bash
```

Docker startet den Container und zeigt Ihnen eine Eingabeaufforderung zur Verwendung von Befehlen innerhalb des Containers.

Legen Sie im Container die Umgebungsvariablen mit dem folgenden Befehl fest:

```
export LD_LIBRARY_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$LD_LIBRARY_PATH
export PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/bin:$PATH
```

```
export GST_PLUGIN_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/  
kinesis-video-native-build/downloads/local/lib:$GST_PLUGIN_PATH
```

Starten Sie das Streaming von der Kamera mit dem `gst-launch-1.0` Befehl, der für Ihr Gerät geeignet ist.

Beispiele für die Verwendung des Befehls `gst-launch-1.0` zum Herstellen einer Verbindung mit einer lokalen Webcam oder einer RTSP-Netzwerkkamera finden Sie unter [Startbefehle](#).

Referenz zu GStreamer-Elementparametern

Um Videos an das Amazon Kinesis Video Streams Producer SDK zu senden, geben Sie `kvssink` als Senke oder endgültiges Ziel der Pipeline an. Diese Referenz enthält Informationen zu den erforderlichen und optionalen `kvssink`-Parametern. Weitere Informationen finden Sie unter [the section called “GStreamer”](#).

Topics

- [the section called “Bereitstellen von Anmeldeinformationen für kvssink”](#)
- [the section called “Bereitstellen einer Region für kvssink”](#)
- [the section called “kvssink Erforderliche Parameter”](#)
- [the section called “kvssink optionale Parameter”](#)

Bereitstellen von Anmeldeinformationen für **kvssink**

Damit das `kvssink` GStreamer-Element Anfragen an stellen kann AWS, geben Sie AWS Anmeldeinformationen für die Verwendung beim Aufrufen des Amazon Kinesis Video Streams-Services an. Die Kette der Anbieter von Anmeldeinformationen sucht in der folgenden Reihenfolge nach Anmeldeinformationen:

1. AWS IoT Anmeldeinformationen

Informationen zum Einrichten von AWS IoT Anmeldeinformationen finden Sie unter [the section called “Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von AWS IoT”](#).

Der `iot-credentials` Parameterwert muss mit `beginnen iot-certificate`, und gefolgt von einer durch Komma getrennten Liste der folgenden *Schlüssel-Wert*-Paare sein.

Schlüssel	Erforderlich	Beschreibung
ca-path	Ja	<p>Dateipfad zum CA-Zertifikat, das verwendet wird, um eine Vertrauensstellung mit dem Backend-Service über TLS herzustellen.</p> <p>Example</p> <p>Beispiel: <code>/file/path/to/certificate.pem</code></p>
cert-path	Ja	<p>Dateipfad zum X.509-Zertifikat.</p> <p>Example</p> <p>Beispiel: <code>/file/path/to/certificateID-certificate.pem.crt</code></p>
endpoint	Ja	<p>Der Endpunktanbieter-Endpoint für AWS IoT Core Anmeldeinformationen für Ihr AWS Konto. Weitere Informationen finden Sie im AWS IoT - Entwicklerhandbuch.</p> <p>Example</p> <p>Beispiel: <code>credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com</code></p>

Schlüssel	Erforderlich	Beschreibung
key-path	Ja	<p>Dateipfad zum privaten Schlüssel, der im öffentlichen/privaten Schlüsselpaar verwendet wird.</p> <p>Example</p> <p>Beispiel: <code>/file/path/to/certificateID-private.pem.key</code></p>
role-aliases	Ja	<p>Der Name des Rollenalias, der auf die AWS IAM-Rolle verweist, die beim Herstellen einer Verbindung mit verwendet werden soll AWS IoT Core.</p> <p>Example</p> <p>Beispiel: <code>KvsCameraIoTRoleAlias</code></p>
iot-thing-name	Nein	<p>Die <code>iot-thing-name</code> ist optional. Wenn nicht angegeben <code>iot-thing-name</code> wird, wird der <code>stream-name</code> Parameterwert verwendet.</p> <p>Example</p> <p>Beispiel: <code>kvs_example_camera</code></p>

Example

Beispiel:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  iot-certificate="iot-certificate,endpoint=credential-account-specific-
  prefix.credentials.iot.aws-region.amazonaws.com,cert-path=certificateID-
  certificate.pem.crt,key-path=certificateID-private.pem.key,ca-
  path=certificate.pem,role-aliases=YourRoleAlias,iot-thing-name=YourThingName"
```

2. Umgebungsvariablen

Um Anmeldeinformationen aus der Umgebung kvssink verwenden zu lassen, legen Sie die folgenden Umgebungsvariablen fest:

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_ACCESS_KEY_ID	Ja	Der AWS Zugriffsschlüssel, der für den Zugriff auf Amazon Kinesis Video Streams verwendet wird.
AWS_SECRET_ACCESS_KEY	Ja	Der AWS geheime Schlüssel, der dem Zugriffsschlüssel zugeordnet ist.
AWS_SESSION_TOKEN	Nein	Gibt den erforderlichen Sitzungstokenwert an, wenn Sie temporäre Sicherheit anmeldeinformationen direkt aus - AWS STS Operationen verwenden.

Durch die Festlegung der Umgebungsvariablen wird der verwendete Wert bis zum Ende der Shell-Sitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Um die Variablen in zukünftigen Sitzungen persistent zu machen, legen Sie sie im Startup-Skript Ihrer Shell fest.

3. access-key, secret-key Parameter

Um Anmeldeinformationen direkt als kvssink Parameter anzugeben, legen Sie die folgenden Parameter fest:

kvssink Parametername	Erforderlich	Beschreibung
access-key	Ja	Der AWS Zugriffsschlüssel, der für den Zugriff auf Amazon Kinesis Video Streams verwendet wird.
secret-key	Ja	Der AWS geheime Schlüssel, der dem Zugriffsschlüssel zugeordnet ist.
session-token	Nein	Gibt den erforderlichen Sitzungstokenwert an, wenn Sie temporäre Anmeldeinformationen direkt aus - AWS STS Operationen verwenden.

Example

Verwenden statischer Anmeldeinformationen:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

Example

Verwenden temporärer Anmeldeinformationen:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

4. Datei mit Anmeldeinformationen

Important

Wenn Sie eine der vorherigen Methoden ausgewählt haben, können Sie den `credential-file kvssink` Parameter nicht verwenden.

kvssink Parametername	Erforderlich	Beschreibung
<code>credential-file</code>	Ja	Pfad zu der Textdatei, die Anmeldeinformationen in einem bestimmten Format enthält.

Die Textdatei muss Anmeldeinformationen in einem der folgenden Formate enthalten:

- GUTschriften *YourAccessKey YourSecretKey*
- *YourAccessKey Ablauf* von KREDENTIALS *YourSecretKey SessionToken*

Example

Beispiel: Ihre `credentials.txt` Datei befindet sich unter `/home/ubuntu` und enthält Folgendes:

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

Um es in zu verwenden `kvssink`, geben Sie Folgendes ein:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
credential-file="/home/ubuntu/credentials.txt"
```

Note

Die Ablaufzeit sollte mindestens $5 + 30 + 3 = 38$ Sekunden in der Zukunft liegen. Die Übergangsfrist ist als `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` Variable in [definiert](#) [IotCredentialProvider.h](#). Wenn die Anmeldeinformationen beim Starten

von zu nah am Ablauf sind `kvssink`, erhalten Sie den Fehlercode `0x52000049` - `STATUS_INVALID_TOKEN_EXPIRATION`.

Important

`kvssink` ändert die Datei mit den Anmeldeinformationen nicht. Wenn Sie temporäre Anmeldeinformationen verwenden, muss die Datei mit den Anmeldeinformationen von einer externen Quelle vor der Ablaufzeit abzüglich der Übergangsfrist aktualisiert werden.

Bereitstellen einer Region für `kvssink`

Im Folgenden finden Sie die Reihenfolge der Regionssuche:

1. `AWS_DEFAULT_REGION` Die `-Umgebungsvariable` wird zuerst überprüft. Wenn sie festgelegt ist, wird diese Region zur Konfiguration des Clients verwendet.
2. `aws-region` -Parameter wird als Nächstes überprüft. Wenn sie festgelegt ist, wird diese Region zur Konfiguration des Clients verwendet.
3. Wenn keine der vorherigen Methoden verwendet wurde, ist `kvssink` standardmäßig `us-west-2`.

`kvssink` Erforderliche Parameter

Zusätzlich zur Bereitstellung von Anmeldeinformationen und einer Region verfügt das `-kvssink` Element über den folgenden erforderlichen Parameter:



`stream-name` – Der Name des Amazon Kinesis-Ziel-Videostreams.

`kvssink` optionale Parameter

Das Element `kvssink` hat die folgenden optionalen Parameter. Weitere Informationen zu diesen Parametern finden Sie unter [Kinesis-Videostream-Strukturen](#).

Parameter	Beschreibung	Einheit/Typ	Standard
<code>absolute-fragment-times</code>	Gibt an, ob absolute Fragmentzeiten verwendet werden.	Boolesch	<code>true</code>

Parameter	Beschreibung	Einheit/Typ	Standard
access-key	<p>Der AWS Zugriffsschlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird.</p> <p>Sie müssen entweder AWS Anmeldeinformationen festgelegt haben oder diesen Parameter angeben. Um diese Informationen bereitzustellen, geben Sie Folgendes ein:</p> <pre>\$ export AWS_ACCESS_KEY_ID=</pre>		
avg-bandwidth-bps	Die erwartete durchschnittliche Bandbreite für den Stream.	Bytes pro Sekunde	4194304


Parameter	Beschreibung	Einheit/Typ	Standard
<code>aws-region</code>	<p>Die AWS-Region zu verwendende .</p> <div data-bbox="472 352 792 1528"><p> Note</p><p>Sie können der Region auch die <code>AWS_DEFAULT_REGION</code> Umgebungsvariable zur Verfügung stellen. Die Umgebungsvariablen haben Vorrang, wenn sowohl die Umgebungsvariablen als auch die <code>kvssink-Parameter</code> festgelegt sind.</p></div> <div data-bbox="472 1591 792 1869"><p> Important</p><p>Die Region wird standardmäßig auf</p></div>	String	<code>us-west-2</code>

Parameter	Beschreibung	Einheit/Typ	Standard
	gesetzus-west-2, sofern nicht anders angegeben.		
buffer-duration	Laufzeit des Streampuffers	Sekunden	180
codec-id	Die Codec-ID des Streams	String	"V_MPEG4/ISO/AVC"
connection-staleness	Die Zeit danach, nach der der Stream-Aktualitäts-Callback aufgerufen wird.	Sekunden	60
content-type	Der Inhaltstyp des Streams	String	"video/h264"
fragment-acks	Gibt an, ob Fragment-ACKs verwendet werden.	Boolesch	true
fragment-duration	Die gewünschte Fragmentdauer	Millisekunden	2000
framerate	Die erwartete Bildrate.	Frames pro Sekunde	25
frame-timecodes	Verwendung von Frame-Timecodes oder Generieren von Zeitstempeln mit dem aktuellen Zeit-Callback:	Boolesch	true

Parameter	Beschreibung	Einheit/Typ	Standard
<code>key-frame-fragmentation</code>	Gibt an, ob Fragmente für einen Keyframe erstellt werden.	Boolesch	true
<code>log-config</code>	Der Protokollkonfigurationspfad	String	<code>"/kvs_log_configuration"</code>
<code>max-latency</code>	Die maximale Latenz für den Stream	Sekunden	60
<code>recalculate-metrics</code>	Gibt an, ob die Metriken neu berechnet werden.	Boolesch	true
<code>replay-duration</code>	Die Laufzeit, um die der aktuelle Reader bei Auftreten eines Fehlers "zurückgespult" werden muss, um die Wiedergabe zu wiederholen	Sekunden	40
<code>restart-on-error</code>	Gibt an, ob beim Auftreten von Fehlern ein Neustart ausgeführt wird.	Boolesch	true
<code>retention-period</code>	Die Zeitdauer, für die der Stream aufbewahrt wird	Stunden	2

Parameter	Beschreibung	Einheit/Typ	Standard
<code>rotation-period</code>	<p>Der Schlüssel rotationszeitraum. Weitere Informationen finden Sie unter Rotieren von Kundenmasterschlüsseln.</p>	Sekunden	2400
<code>secret-key</code>	<p>Der AWS geheime Schlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird.</p> <p>Sie müssen entweder AWS Anmeldeinformationen festgelegt haben oder diesen Parameter angeben.</p> <pre>\$ export AWS_SECRET_ACCESS_KEY_ID=</pre>		
<code>session-token</code>	<p>Gibt den erforderlichen Sitzungstokenwert an, wenn Sie temporäre Sicherheitsanmeldeinformationen direkt aus AWS STS -Operationen verwenden.</p>		

Parameter	Beschreibung	Einheit/Typ	Standard
<code>storage-size</code>	Die Speicherg röße des Geräts in MegaBytes (MB). Weitere Informati onen zum Konfiguri eren von Speicherg eräten finden Sie unter StorageInfo .	MegaBytes	128
<code>streaming-type</code>	Der Streaming-Typ. Gültige Werte sind: <ul style="list-style-type: none">• 0: Echtzeit• 1: nahezu in Echtzeit (derzeit nicht unterstützt)• 2: offline	Enum GstKvsSin kStreamingType	0: Echtzeit
<code>timecode-scale</code>	Die MKV-Timecode- Skala	Millisekunden	1
<code>track-name</code>	Der Name des MKV- Tracks.	String	"kinesis_video"

Parameter	Beschreibung	Einheit/Typ	Standard
<code>iot-certificate</code>	<p>AWS IoT -Anmeldeinformationen, die im <code>-kvssinkElement</code> verwendet werden sollen.</p> <p><code>iot-certificate</code> akzeptiert die folgenden Schlüssel und Werte:</p> <div data-bbox="472 716 792 1514" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Die <code>iot-thing-name</code> ist optional. Wenn nicht angegeben <code>iot-thing-name</code> wird, wird der <code>stream-name</code> Parameterwert verwendet.</p> </div> <ul style="list-style-type: none"> • <code>endpoint=iotcredentialsproviderendpoint</code> • <code>cert-path=/localdirectorypat</code> 	String	None

Parameter	Beschreibung	Einheit/Typ	Standard
	<pre>h /to/certificate</pre> <ul style="list-style-type: none"> • <code>key-path=/localdirectorypath /to/private/key</code> • <code>ca-path=/localdirectorypath/to/ca-cert</code> • <code>role-aliases =role-aliases</code> • <code>iot-thing-name=YourIotThingName</code> 		

Beispiel: Senden von Daten an Kinesis Video Streams mithilfe der PutMedia API

Dieses Beispiel zeigt, wie die [PutMedia](#)-API verwendet wird. Es zeigt, wie Daten gesendet werden, die sich bereits in einem Container-Format (MKV) befinden. Wenn Ihre Daten vor dem Senden in ein Containerformat zusammengestellt werden müssen (z. B. wenn Sie Kameravideodaten in Frames zusammenstellen), finden Sie weitere Informationen unter [Produzentenbibliotheken von Kinesis Video Streams](#).

Note

Die `PutMedia` Operation ist nur in den C++- und Java-SDKs verfügbar. Dies ist auf die Vollduplex-Verwaltung von Verbindungen, Datenfluss und Bestätigungen zurückzuführen. Es wird in anderen Sprachen nicht unterstützt.

In diesem Beispiel werden folgende Schritte beschrieben:

- [Schritt 1: Herunterladen und Konfigurieren des Codes](#)
- [Schritt 2: Schreiben und Untersuchen des Codes](#)
- [Schritt 3: Ausführen und Überprüfen des Codes](#)

Schritt 1: Herunterladen und Konfigurieren des Codes

Führen Sie die Schritte aus, um den Java-Beispielcode herunterzuladen, das Projekt in Ihre Java-IDE zu importieren, die Bibliotheksspeicherorte zu konfigurieren und den Code für die Verwendung Ihrer AWS Anmeldeinformationen zu konfigurieren.

1. Erstellen Sie ein Verzeichnis und klonen Sie den Beispiel Quellcode aus dem GitHub Repository. Das PutMedia-Beispiel ist Bestandteil der [Java Producer-Bibliothek](#).

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. Öffnen Sie die Java-IDE, die Sie verwenden (z. B. [Eclipse](#) oder [IntelliJ IDEA](#)), und importieren Sie das Apache-Maven-Projekt, das Sie heruntergeladen haben:
 - In Eclipse: Klicken Sie auf File (Datei), Import (Importieren), Maven, Existing Maven Projects (Vorhandene Maven-Projekte) und navigieren Sie zum Stammverzeichnis des heruntergeladenen Pakets. Wählen Sie die pom.xml-Datei aus.
 - In IntelliJ Idea: Klicken Sie auf Import. Navigieren Sie zur Datei pom.xml im Stammverzeichnis des heruntergeladenen Pakets.

Weitere Informationen finden Sie in der Dokumentation zur IDE.

3. Aktualisieren Sie das Projekt, damit die IDE die importierten Bibliotheken findet.
 - Gehen Sie bei IntelliJ IDEA wie folgt vor:
 - a. Klicken Sie mit der rechten Maustaste auf das Verzeichnis lib des Projekts und wählen Sie Add as library.
 - b. Wählen Sie Datei und dann Projektstruktur aus.
 - c. Wählen Sie unter Project Settings die Option Modules aus.
 - d. Stellen Sie auf der Registerkarte Sources (Quellen) die Option Language Level (Sprachebene) auf **7** oder einen höheren Wert ein.

- Gehen Sie bei Eclipse wie folgt vor:
 - a. Klicken Sie mit der rechten Maustaste auf das Projekt und wählen Sie Properties, Java Build Path, Source. Führen Sie dann die folgenden Schritte aus:
 1. Doppelklicken Sie auf der Registerkarte Source auf Native library location.
 2. Klicken Sie im Assistenten Native Library Folder Configuration auf Workspace.
 3. Wählen Sie im Auswahlbildschirm Native Library Folder das Verzeichnis lib des Projekts aus.
 - b. Klicken Sie mit der rechten Maustaste auf das Projekt und wählen Sie Properties. Führen Sie dann die folgenden Schritte aus:
 1. Klicken Sie auf der Registerkarte Libraries auf Add Jars.
 2. Wählen Sie im Assistenten JAR selection (JAR-Auswahl) alle JAR-Dateien im Verzeichnis lib des Projekts aus.

Schritt 2: Schreiben und Untersuchen des Codes

Das PutMedia-API-Beispiel () weist folgendes Programmiermuster auf:PutMediaDemo

Themen

- [Erstellen der PutMediaClient](#)
- [Streamen von Medien und Anhalten des Threads](#)

Die Codebeispiele in diesem Abschnitt stammen aus der Klasse PutMediaDemo.

Erstellen der PutMediaClient

Das Erstellen des PutMediaClient Objekts erfordert die folgenden Parameter:

- URI des PutMedia-Endpunkts
- Ein InputStream-Wert, der auf die streamende MKV-Datei zeigt.
- Name des Streams. In diesem Beispiel wird wieder der in [Verwenden der Java Producer-Bibliothek](#) (my-stream) erstellte Stream verwendet. Wenn Sie einen anderen Stream verwenden möchten, ändern Sie folgenden Parameter:

```
private static final String STREAM_NAME="my-stream";
```

Note

Das PutMedia API-Beispiel erstellt keinen Stream. Sie müssen einen Stream erstellen, indem Sie entweder die Testanwendung für die [Verwenden der Java Producer-Bibliothek](#), die Kinesis-Video-Streams-Konsole oder die verwenden AWS CLI.

- Aktueller Zeitstempel
- Zeitcodetyp. In diesem Beispiel wird RELATIVE verwendet, was bedeutet, dass der Zeitstempel relativ zum Beginn des Containers gilt.
- Ein AWSKinesisVideoV4Signer-Objekt, das prüft, ob die empfangenen Pakete von einem autorisierten Absender gesendet wurden
- Maximale Upstream-Bandbreite in Kbit/s
- Ein AckConsumer-Objekt, das die empfangenen Bestätigungspakete aufnimmt.

Im folgenden Code wird das Objekt PutMediaClient erstellt:

```
/* actually URI to send PutMedia request */
final URI uri = URI.create(KINESIS_VIDEO_DATA_ENDPOINT + PUT_MEDIA_API);

/* input stream for sample MKV file */
final InputStream inputStream = new FileInputStream(MKV_FILE_PATH);

/* use a latch for main thread to wait for response to complete */
final CountdownLatch latch = new CountdownLatch(1);

/* a consumer for PutMedia ACK events */
final AckConsumer ackConsumer = new AckConsumer(latch);

/* client configuration used for AWS SigV4 signer */
final ClientConfiguration configuration = getClientConfiguration(uri);

/* PutMedia client */
final PutMediaClient client = PutMediaClient.builder()
    .putMediaDestinationUri(uri)
    .mkvStream(inputStream)
    .streamName(STREAM_NAME)
```

```
.timestamp(System.currentTimeMillis())
.fragmentTimeCodeType("RELATIVE")
.signWith(getKinesisVideoSigner(configuration))
.upstreamKbps(MAX_BANDWIDTH_KBPS)
.receiveAcks(ackConsumer)
.build();
```

Streamen von Medien und Anhalten des Threads

Nach dem Erstellen des Clients wird das asynchrone Streaming mit `putMediaInBackground` gestartet. Danach wird der Haupt-Thread mit `latch.await` angehalten, bis `AckConsumer` zurückkehrt. Nun wird der Client geschlossen.

```
/* start streaming video in a background thread */
    client.putMediaInBackground();

    /* wait for request/response to complete */
    latch.await();

    /* close the client */
    client.close();
```

Schritt 3: Ausführen und Überprüfen des Codes

Um das PutMedia-API-Beispiel auszuführen, gehen Sie wie folgt vor:

1. Erstellen Sie einen Stream mit dem Namen `my-stream` in der Kinesis-Video-Streams-Konsole oder mithilfe der AWS CLI.
2. Ändern Sie Ihr Arbeitsverzeichnis auf das Java Producer SDK-Verzeichnis:

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Kompilieren Sie das Java SDK und die Demoanwendung:

```
mvn package
```

4. Erstellen Sie einen temporären Dateinamen im Verzeichnis `/tmp`:

```
jar_files=$(mktemp)
```

- Erstellen Sie eine classpath-Zeichenfolge mit den Abhängigkeiten zum lokalen Repository in einer Datei:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

- Legen Sie den Wert der Umgebungsvariablen LD_LIBRARY_PATH wie folgt fest:

```
export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-  
video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:  
$LD_LIBRARY_PATH  
$ classpath_values=$(cat $jar_files)
```

- Führen Sie die Demo über die Befehlszeile wie folgt aus und geben Sie Ihre AWS Anmeldeinformationen an:

```
java -classpath target/kinesisvideo-java-demo-1.0-SNAPSHOT.jar:$classpath_values -  
Daws.accessKeyId=${ACCESS_KEY} -Daws.secretKey=${SECRET_KEY} -Djava.library.path=/  
opt/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build  
com.amazonaws.kinesisvideo.demoapp.DemoAppMain
```

- Öffnen Sie die [Kinesis-Video-Streams-Konsole](#) und wählen Sie Ihren Stream auf der Seite Streams verwalten aus. Das Video wird im Bereich Video Preview wiedergegeben.

Beispiel: Streaming aus einer RTSP-Quelle

[C++ Producer-Bibliothek](#) enthält eine Definition für einen [Docker](#)-Container, der eine Verbindung zu einer RTSP-Netzwerkkamera herstellt. Die Verwendung von Docker standardisiert die Betriebsumgebung für Kinesis Video Streams, wodurch die Erstellung und Verwendung der Anwendung optimiert wird.

Im folgenden Verfahren wird gezeigt, wie Sie die RTSP-Demoanwendung einrichten und verwenden.

Themen

- [Voraussetzungen](#)
- [Erstellen des Docker-Images](#)
- [Ausführen der RTSP-Beispielanwendung](#)

Voraussetzungen

Um die RTSP-Beispielanwendung von Kinesis Video Streams auszuführen, benötigen Sie Folgendes:

- Docker: Weitere Informationen zur Installation und Verwendung von Docker finden Sie über folgende Links:
 - [Anleitung zum Herunterladen von Docker](#)
 - [Erste Schritte mit Docker](#)
- RTSP-Netzwerkcameraquelle: Weitere Informationen zu empfohlenen Kameras finden Sie unter [Systemanforderungen](#).

Erstellen des Docker-Images

Erstellen Sie zunächst das Docker-Image, in dem die Demo-Anwendung ausgeführt wird.

1. Klonen Sie das Demo-Repository von Amazon Kinesis Video Streams.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Wechseln Sie zu dem Verzeichnis, das die Dockerfile enthält. In diesem Fall ist es das [docker-rtsp](#)-Verzeichnis.

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. Verwenden Sie den folgenden Befehl, um das Docker-Image zu erstellen. Dieser Befehl erstellt das Image und markiert es als `rtspdockertest`.

```
docker build -t rtspdockertest .
```

4. Führen Sie aus `docker images` und suchen Sie nach der Image-ID, die mit `rtspdockertest` gekennzeichnet ist.

In der folgenden Beispielausgabe `IMAGE ID` ist beispielsweise `54f0d65f69b2`.

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB

Sie benötigen dies in einem späteren Schritt.

Ausführen der RTSP-Beispielanwendung

Sie können die RTSP-Beispielanwendung entweder innerhalb oder außerhalb des Docker-Containers ausführen. Folgen Sie den entsprechenden Anweisungen unten.

Themen

- [Innerhalb des Docker-Containers](#)
- [Außerhalb des Docker-Containers](#)

Innerhalb des Docker-Containers

Ausführen der RTSP-Beispielanwendung

1. Starten Sie den Docker-Container von Amazon Kinesis Video Streams mit dem folgenden Befehl:

```
docker run -it YourImageId /bin/bash
```

2. Um die Beispielanwendung zu starten, geben Sie Ihre AWS Anmeldeinformationen, den Namen des Amazon Kinesis Video Streams und die URL der RTSP-Netzwerkamera an.

Important

Wenn Sie temporäre Anmeldeinformationen verwenden, müssen Sie auch Ihre `AWS_SESSION_TOKEN` angeben. Siehe das zweite Beispiel unten.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId  
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId  
export AWS_DEFAULT_REGION=YourAWSRegion  
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Temporäre Anmeldeinformationen:

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
```

```
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId
export AWS_SESSION_TOKEN=YourSessionToken
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die [Kinesis-Video-Streams-Konsole](#).

Zeigen Sie den Stream an.

4. Um den Docker-Container zu verlassen, schließen Sie das Terminalfenster oder geben Sie `exit`.

Außerhalb des Docker-Containers

Verwenden Sie außerhalb des Docker-Containers den folgenden Befehl:

```
docker run -it YourImageId /bin/bash -c "export AWS_ACCESS_KEY_ID=YourAccessKeyId;
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId; export
AWS_SESSION_TOKEN=YourSessionToken; export AWS_DEFAULT_REGION=YourAWSRegion; ./
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

Beispiel: Analysieren und Rendern von Kinesis-Video-Streams-Fragmenten

Die "[Stream Parser](#)"-Bibliothek enthält eine Demo-Anwendung mit dem Namen `KinesisVideoRendererExample`, die das Parsen und Rendern von Amazon Kinesis-Videostreamfragmenten demonstriert. Im Beispiel werden mit [JCodec](#) die H.264-codierten Frames, die mittels der [Beispiel: Kinesis Video Streams Producer SDK GStreamer-Plugin](#)-Anwendung übernommen werden, decodiert. Nachdem der Frame mit JCodec decodiert wurde, wird das sichtbare Bild mit [JFrame](#) gerendert.

Dieses Beispiel veranschaulicht, wie Sie folgende Aufgaben ausführen:

- Rufen Sie Frames aus einem Kinesis-Videostream mithilfe der `GetMedia` API ab und rendern Sie den Stream zur Anzeige.
- Zeigen Sie den Videoinhalt von Streams in einer benutzerdefinierten Anwendung an, anstatt die Kinesis-Video-Streams-Konsole zu verwenden.

Sie können die Klassen in diesem Beispiel auch verwenden, um Kinesis-Videostream-Inhalte anzuzeigen, die nicht als H.264 codiert sind, z. B. einen Stream von JPEG-Dateien, für die keine Dekodierung erforderlich ist, bevor sie angezeigt werden.

Im folgenden Verfahren wird veranschaulicht, wie Sie die Renderer-Demoanwendung einrichten und verwenden.

Voraussetzungen

Zum Untersuchen und Verwenden der Renderer-Beispielbibliothek ist Folgendes erforderlich:

- Ein Amazon Web Services (AWS)-Konto. Wenn Sie noch kein - AWS Konto haben, finden Sie weitere Informationen unter [Erste Schritte mit Kinesis Video Streams](#).
- Eine integrierte Java-Entwicklungsumgebung (IDE), wie [Eclipse Java Neon](#) oder [JetBrains IntelliJ Bol](#).

Ausführen des Renderer-Beispiels

1. Erstellen Sie ein Verzeichnis und klonen Sie dann den Beispielquellcode aus dem GitHub Repository.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Öffnen Sie die Java-IDE (z. B. [Eclipse](#) oder [IntelliJ IDEA](#)) und importieren Sie das heruntergeladene Apache Maven-Projekt:
 - In Eclipse: Klicken Sie auf File, Import, Maven, Existing Maven Projects. Navigieren Sie zum `kinesis-video-streams-parser-lib` Verzeichnis .
 - In IntelliJ Idea: Klicken Sie auf Import. Navigieren Sie zur Datei `pom.xml` im Stammverzeichnis des heruntergeladenen Pakets.

Note

Wenn IntelliJ Ihre Abhängigkeiten nicht finden kann, müssen Sie möglicherweise Folgendes tun:

- Neuerstellung: Klicken Sie auf Datei, Einstellungen, Build, Execution, Deployment (Erstellung, Ausführung, Bereitstellung), Compiler. Stellen Sie sicher, dass

Ausgabeverzeichnis bei der Neuerstellung löschen ausgewählt ist, und wählen Sie dann Erstellen, Projekt erstellen aus.

- Projekt erneut importieren: Öffnen Sie das Kontextmenü (Rechtsklick) für das Projekt und wählen Sie dann Maven und Reimport (Neu importieren) aus.

Weitere Informationen finden Sie in der Dokumentation zur IDE.

3. Öffnen Sie über Ihre Java-IDE `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest`.
4. Entfernen Sie die Anweisung `@Ignore` aus der Datei.
5. Aktualisieren Sie den `.stream` Parameter mit dem Namen Ihres Kinesis-Videostreams.
6. Führen Sie den Test `KinesisVideoRendererExample` aus.

So funktioniert's

Die Beispielanwendung veranschaulicht die folgenden Schritte:

- [Senden von MKV-Daten](#)
- [Parsen von MVV-Fragmenten in Frames](#)
- [Dekodieren und Anzeigen des Frames](#)

Senden von MKV-Daten

Das Beispiel sendet Beispiel-HSV-Daten aus der `rendering_example_video.mkv` Datei mit `PutMedia` um Videodaten an einen Stream mit dem Namen `send-render-example-stream`.

Die Anwendung erstellt eine `PutMediaWorker`:

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

Informationen zur Klasse `PutMediaWorker` finden Sie unter [Ruf PutMedia](#) in der Dokumentation zu ["Stream Parser"-Bibliothek](#).

Parsen von MVV-Fragmenten in Frames

Im Beispiel werden die MKV-Fragmente dann mittels einer `GetMediaWorker` aus dem Stream abgerufen und analysiert:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getFrameVisitor());
executorService.submit(getMediaWorker);
```

Weitere Informationen zur Klasse `GetMediaWorker` finden Sie unter [Ruf GetMedia](#) in der Dokumentation zu ["Stream Parser"-Bibliothek](#).

Dekodieren und Anzeigen des Frames

Der Frame wird dann im Beispiel mittels [JFrame](#) decodiert und angezeigt.

Das folgende Code-Beispiel stammt aus der Klasse `KinesisVideoFrameViewer`, die `JFrame` erweitert:

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

Das Image wird als Instance von [java.awt.image angezeigter BufferedImage](#) Beispiele, die die Arbeit mit `BufferedImage` veranschaulichen, finden Sie unter [Lesen/Laden eines Image](#).

Beispiel: Identifizieren von Objekten in Videostreams mit SageMaker

Dieses Beispiel zeigt, wie Sie eine Lösung erstellen, die verwendet, [SageMaker](#) um zu identifizieren, wann bestimmte Objekte in einem Amazon Kinesis Video Stream angezeigt werden. SageMaker ist die verwaltete Plattform, mit der Entwickler und Datenwissenschaftler schnell und einfach Machine-Learning-Modelle erstellen, trainieren und bereitstellen können.

Das Beispiel besteht aus einem [Docker](#)-Container, der die Anwendungsfunktionalität enthält, und einer [-AWS CloudFormation](#)-Vorlage, die die Bereitstellung der AWS Ressourcen der Anwendung automatisiert.

Die AWS CloudFormation Vorlage erstellt die folgenden Ressourcen:

- Ein [Amazon Elastic Container Service \(Amazon ECS\)](#)-Cluster, der die [AWS Fargate](#)-Rechen-Engine verwendet, auf der die Bibliothekssoftware ausgeführt wird.
- Eine [Amazon-DynamoDB](#)-Tabelle, die Checkpoints und den zugehörigen Status für Worker verwaltet, die auf Fargate-Aufgaben ausgeführt werden.
- Ein [Kinesis-Datenstrom](#), der die von generierten Inferenzausgaben erfasst SageMaker.
- Eine [-AWS Lambda Funktion](#), die die Ausgabe von analysiert SageMaker.
- [AWS Identity and Access Management \(IAM\)](#)-Ressourcen für den Zugriff auf alle -Services.
- [Amazon CloudWatch](#)-Ressourcen für die Überwachung der Anwendung.

Die Anwendung ist mit jedem SageMaker Endpunkt kompatibel, der Daten verarbeitet. Dieses Beispiel enthält Anweisungen zum Erstellen eines SageMaker Endpunkts, der eine Beispielvorlage für den Objektidentifizierungsalgorithmus verwendet. Sie können den Algorithmus je nach Anwendungsfall und Anforderungen Ihrer Anwendung ändern oder ersetzen.

Themen

- [Voraussetzungen](#)
- [Erstellen der Anwendung](#)
- [Überwachung der Anwendung](#)
- [Erweitern der Anwendung](#)
- [Bereinigen der Anwendung](#)

Voraussetzungen

Für die Beispielanwendung gelten die folgenden Voraussetzungen:

- [SageMaker](#)
- [Kinesis-Videostream](#)
- [Serviceverknüpfte Rolle](#)

SageMaker

Für dieses Beispiel ist ein SageMaker Notebook erforderlich. Informationen zum Erstellen eines Notebooks finden Sie unter [Erstellen einer Notebook-Instance](#) im Amazon- SageMaker Entwicklerhandbuch. Beachten Sie bei der Erstellung Ihres Notebooks Folgendes:

- Fügen Sie das `Amazon_JumpStart_Object_Detection.ipynb` Beispiel (aus dem Abschnitt Einführung in Amazon-Algorithmen auf der Registerkarte SageMaker Beispiele der Jupyter-Konsole) zum Notebook hinzu.
- Erstellen Sie einen Amazon Simple Storage Service (Amazon S3)-Bucket und geben Sie seinen Namen im Schritt Voraussetzungen an, wenn Sie das Beispiel hinzufügen.
- Nachdem Sie das Notebook erstellt haben, wählen Sie Endpunktkonfiguration in der SageMaker Konsole und notieren Sie sich den Endpunktnamen .

Kinesis-Videostream

Dieses Beispiel erfordert einen oder mehrere Kinesis Video Streams mit Live-Videodaten. Informationen zum Erstellen eines Kinesis-Videostreams und zum Senden von Daten von einer Kamera an ihn finden Sie unter [GStreamer](#). Notieren Sie sich den Namen Ihres Kinesis-Videostreams.

Serviceverknüpfte Rolle

In diesem Beispiel muss Ihr Konto über eine serviceverknüpfte Rolle für den Fargate-Vorgang verfügen. Bei neuen AWS-Konten ist diese Rolle standardmäßig aktiviert. Wenn Sie beim Erstellen der Anwendung die folgende Fehlermeldung sehen, müssen Sie die serviceverknüpfte Rolle aktivieren:

```
Unable to assume the service linked role. Please verify that the ECS service linked role exists
```

Führen Sie zum Aktivieren der serviceverknüpften Rolle den folgenden Befehl aus:

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```


Erstellen der Anwendung

Um die Beispielanwendung zu erstellen, verwenden Sie AWS CloudFormation und die bereitgestellten Vorlagen.

So verwenden Sie AWS CloudFormation zum Erstellen der Anwendung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die - AWS CloudFormation Konsole mit einem der folgenden Links für Ihr AWS-Region. Der Link startet den richtigen Stack für Ihre Region:
 - [Start in der Region Asien-Pazifik \(Sydney\) \(ap-southeast-2\)](#)
 - [Starten in der Region Asien-Pazifik \(Tokio\) \(ap-northeast-1\)](#)
 - [Start in der Region Europa \(Frankfurt\) \(eu-central-1\)](#)
 - [Starten in der Region Europa \(Irland\) \(eu-west-1\)](#)
 - [Starten in der Region USA Ost \(Nord-Virginia\) \(us-east-1\)](#)
 - [Starten in der Region USA West \(Oregon\) \(us-west-2\)](#)
2. Geben Sie auf der Seite Create Stack (Stack erstellen) die folgenden Werte an:
 - Geben Sie dem Stack einen eindeutigen Namen (z. B. *Benutzername* -KVS-SageMaker).
 - Geben Sie den SageMaker Endpunktnamen (nicht den Endpunkt-ARN) an, den Sie im vorherigen Abschnitt erstellt haben.
 - Geben Sie den Namen Ihres Kinesis-Videostreams an. Wenn Sie mehr als einen Kinesis-Videostream haben, geben Sie die Streamnamen in Anführungszeichen an und trennen Sie sie durch Kommas.
 - Behalten Sie die restlichen Einstellungen bei.

Wählen Sie Weiter aus.

3. Lassen Sie die Einstellungen auf der Seite Options (Optionen) unverändert.
4. Aktivieren Sie das Kontrollkästchen Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt. Wählen Sie Weiter aus.

AWS CloudFormation erstellt die Anwendung.

In der folgenden Tabelle sind mehrere Parameter aufgeführt, die vom Docker-Container verwendet werden, wenn Sie einen Stack mit dieser AWS CloudFormation Vorlage erstellen. Die Werte sind in der SSM-Ressource in der Vorlage vordefiniert, aber Sie können sie nach Bedarf anpassen.

Ressourcenname	Standardwert	Beschreibung
<code>inferenceInterval</code>	6	The sampling ratio for video frames that are sent to the SageMaker endpoint to support inferencing on I-Frames. The default value of 6 means that 6 frames are sent to the SageMaker endpoint.
<code>sageMakerTaskQueueSize</code>	5000	The size of the queue that maintains the pending requests to the SageMaker endpoint. The size of the queue is affected by 'inferenceInterval' and 'sageMakerTaskThreadPoolSize'. If SageMaker inference takes longer, requests are buffered in this queue.
<code>sageMakerTaskThreadPoolSize</code>	20	Number of threads that's used to concurrently carry out SageMaker inference requests.
<code>sageMakerTaskTimeoutInMilli</code>	20000	The maximum duration accepted for a single request (or a retry request) to the SageMaker endpoint.
<code>sageMakerTaskThreadPoolName</code>	SageMakerThreadPool-%d	The name of the threadpool that's sending requests to the SageMaker endpoint.

Um die Werte dieser Parameter anzupassen, laden Sie die AWS CloudFormation Vorlage herunter, indem Sie die Vorlagen-URL auf der Seite Stack erstellen auswählen, und suchen Sie diese Parameter dann im Params Abschnitt der Vorlage, der wie folgt aussieht:

```
Params:
  Type: AWS::SSM::Parameter
  Properties:
    Name:
      Ref: AppName
    Description: "Configuration for SageMaker app"
    Type: String
    Value:
      Fn::Sub: |
```

```
    {"streamNames":[${StreamNames}], "tagFilters":
[${TagFilters}], "sageMakerEndpoint": "${SageMakerEndpoint}",
    "endPointAcceptContentType": "${EndPointAcceptContentType}",

    "kdsStreamName": "${Kds}", "inferenceInterval":6, "sageMakerTaskQueueSize":5000,
    "sageMakerTaskThreadPoolSize":20, "sageMakerTaskTimeoutInMilli":20000,
    "sageMakerTaskThreadPoolName": "SageMakerThreadPool-%d"}
```

Überwachung der Anwendung

Die von der AWS CloudFormation Vorlage erstellte Anwendung enthält ein Amazon CloudWatch - Dashboard und einen CloudWatch Protokollstream, der zur Überwachung von Anwendungsmetriken und -ereignissen verwendet wird.

Anwendungs-Dashboard

Die Anwendung enthält ein CloudWatch Dashboard zur Überwachung von Anwendungsmetriken. Um das Anwendungs-Dashboard anzuzeigen, öffnen Sie die - CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/> und wählen Sie Dashboards in der linken Navigationsleiste aus.

Wählen Sie das Dashboard KVS-SageMaker-Driver-KvsSageMakerIntegration-**aws-region** aus. Das Dashboard zeigt die folgenden Informationen an:

- **Frame-Metriken:** Die Metriken für die Verarbeitung des Videostreams, das Senden von Frames an den SageMaker Endpunkt und das Schreiben in den Kinesis-Datenstrom, der das SageMaker Notebook mit der AWS Lambda Funktion verbindet, die SageMaker Inferenzausgabeergebnisse verarbeitet.
- **IngestToProcessLatency:** Der Zeitunterschied zwischen dem Zeitpunkt, an dem ein Videoframe in den Kinesis-Video-Streams-Service aufgenommen wird, und dem Zeitpunkt, an dem die Anwendung den Frame empfängt.
- **Aktuelle Lease insgesamt:** Der Anwendung werden Berechtigungen zum Lesen aus dem Kinesis-Videostream mithilfe eines Leases gewährt. Diese Metrik zeigt die Anzahl der aktiven Leases an. Die Anwendung verwendet einen Lease pro Kinesis-Videostream und einen Lease für die Synchronisation zwischen Streams.
- **Lease Sync Metrics:** Die Häufigkeit und Dauer der Synchronisation von Leases.
- **LeaseCount pro Worker:** Die Verteilung der Leases auf die SageMaker Worker-Threads.

- **Number of Workers:** die Anzahl der SageMaker-Auftragnehmer, die Streams verarbeiten. Jede Aufgabe in einem Amazon-ECS-Cluster hat einen Worker, der ausgeführt wird. Ein Worker kann mehrere Streams verarbeiten.
- **ECS-Serviceauslastung:** Die Nutzungsmetriken für den Amazon-ECS-Cluster.
- **KinesisDataStream:** Die Nutzungsmetriken des Kinesis-Datenstroms.
- **SageMaker:** Die vom Notebook ausgeführten SageMaker Operationen.
- **Lambda:** Die Anzahl und Dauer der Lambda-Funktion, die die Ausgabe aus dem SageMaker Notebook verarbeitet.

Wenn bestimmte Informationen in diesen Diagrammen auf ein betriebliches Problem hinweisen (z. B. ein permanent ansteigender Wert, der nicht konstant bleibt), informieren Sie sich im folgenden Abschnitt darüber, wie das Problem anhand von Anwendungsprotokollen bestimmt werden kann.

CloudWatch Protokolle

Die Anwendung enthält zwei CloudWatch Protokolle:

Themen

- [Das Anwendungsprotokoll](#)
- [Das Lambda-Funktionsprotokoll](#)

Das Anwendungsprotokoll

Mit dem Anwendungsprotokoll können Sie Anwendungsereignisse und Fehlerbedingungen überwachen. Sie können dieses Protokoll auch verwenden, wenn Sie den -Produktsupport bei einem Problem kontaktieren müssen.

So lesen Sie das Anwendungsprotokoll:

1. Öffnen Sie die Amazon-ECS-Konsole unter <https://console.aws.amazon.com/ecs>.
2. Wählen Sie den Cluster KVS-Sagemaker-Driver aus.
3. Wählen Sie auf der Registerkarte Services den Service **stack-name**-SageMakerDriverService aus.
4. Wählen Sie die Registerkarte Protokolle aus.

Das Anwendungsprotokoll zeigt Ereignisse wie Initialisierung, Konfiguration und Lease-Aktivitäten an.

Das Lambda-Funktionsprotokoll

Sie können das Lambda-Funktionsprotokoll verwenden, um erfolgreiche Objektidentifizierungen zu verfolgen.

So lesen Sie das Lambda-Protokoll

1. Öffnen Sie die - AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda>.
2. Wählen Sie die Lambda-Funktion für Ihre Anwendung aus. Der Name der Lambda-Funktion hat das folgende Format:

```
stack-name-LambdaFunction-A1B2C3D4E5F6G
```

3. Wählen Sie den Bereich Überwachung aus.
4. Wählen Sie Protokolle anzeigen in CloudWatch.

Das CloudWatch Protokoll für die Anwendung zeigt erfolgreiche Identifizierungen von Objekten im Kinesis-Videostrom und andere Anwendungsereignisse.

Erweitern der Anwendung

Sie können Ihrer Anwendung benutzerdefinierte Funktionen hinzufügen, indem Sie die Werte, die Sie im AWS CloudFormation Vorlagenfenster angeben, wie folgt ändern:

- `EndPointAcceptContentType`: Sie können diesen Wert ändern, wenn Ihr SageMaker Endpunkt keine Frames im JPG-Format akzeptiert. Folgende Formate werden unterstützt:
 - `image/jpeg`
 - `image/png`
 - `image/bmp`
 - `image/gif`
 - `application/x-image`
- `LambdaFunctionBucket`, `LambdaFunctionKey`: Die bereitgestellten Einstellungen verwenden eine - AWS Lambda Funktion, die die SageMaker Ausgabe verarbeitet und in CloudWatch Protokolle schreibt. Wenn Sie die SageMaker Ausgabe an eine andere Stelle senden möchten, können Sie Ihre eigene Lambda-Funktion bereitstellen.
- `Tag-Filter`: Bei Streams, die über die [the section called "TagStream"](#)-Aktion mit einem Tag markiert wurden, können Sie die Tags der Streams angeben, die Sie verarbeiten möchten. Liegen z. B. zwei

Streams vor, die den Location-Schlüssel mit den Werten Front und Parking haben, würden Sie mit dem folgenden Eintrag filtern, um nur diese Streams zu verwenden:

```
{"key": "Location", "values": ["Front", "Parking"]}
```

Bereinigen der Anwendung

Nachdem Sie mit der Anwendung, die Sie für dieses Tutorial erstellt haben, fertig sind, empfehlen wir Ihnen, alle nicht mehr benötigten Ressourcen zu löschen, um unnötige Kosten zu vermeiden.

1. SageMaker Endpunkt: Wenn Sie den SageMaker Endpunkt für dieses Tutorial erstellt haben, anstatt einen vorhandenen Endpunkt zu verwenden, löschen Sie den Endpunkt. Wählen Sie in der SageMaker Systemsteuerung Endpunktkonfigurationen aus. Wählen Sie den Endpunkt aus, den Sie erstellt haben, und wählen Sie Aktionen, Löschen. Bestätigen Sie das Löschen.
2. SageMaker Notebook: Wählen Sie in der - SageMaker Konsole Notebook-Instances aus. Wählen Sie das erstellte Notebook und dann Aktionen und Stopp aus. Wenn für das Notebook der Status Angehalten angezeigt wird, wählen Sie Aktionen und dann Delete (Löschen) aus. Bestätigen Sie das Löschen.

Note

Weitere Informationen zum Bereinigen von SageMaker Ressourcen finden Sie unter [Bereinigen](#) im [SageMaker Entwicklerhandbuch für](#) .

3. SageMaker Ausführungsrichtlinie: Wählen Sie in der IAM-Konsole im Navigationsbereich Richtlinien aus. Wählen Sie die Richtlinie aus, die Sie für dieses Tutorial erstellt haben. Der Name der Richtlinie ähnelt dem folgenden: AmazonSageMaker - ExecutionPolicy-*timestamp*.

Wählen Sie Policy actions (Richtlinienaktionen) und anschließend Delete (Löschen) aus. Bestätigen Sie das Löschen.

4. SageMaker Ausführungsrolle: Wählen Sie in der IAM-Konsole im Navigationsbereich Rollen aus. Wählen Sie die Rolle aus, die Sie für dieses Tutorial erstellt haben. Der Name der Rolle ähnelt dem folgenden: AmazonSageMaker-ExecutionRole-*timestamp*.

Wählen Sie Delete role (Rolle löschen) aus. Bestätigen Sie das Löschen.

5. -AWS CloudFormation Stack: Wählen Sie in der - AWS CloudFormation Konsole den -Stack aus, den Sie für dieses Tutorial erstellt haben. Wählen Sie Actions (Aktionen), Delete Stack (Stack löschen). Bestätigen Sie das Löschen.
6. Amazon S3-Bucket: Wählen Sie in der Amazon S3-Konsole den Bucket aus, den Sie zum Speichern der SageMaker Komponenten erstellt haben. Wählen Sie Löschen aus. Geben Sie den Namen des Buckets ein und wählen Sie Bestätigen aus, um das Löschen zu bestätigen.
7. Kinesis-Videostream: Wählen Sie in der Kinesis-Videostreams-Konsole den Videostream aus, den Sie für die Anwendung erstellt haben. Wählen Sie Delete (Löschen). Bestätigen Sie das Löschen.

Überwachung von Amazon Kinesis Video Streams

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon Kinesis Video Streams und Ihrer AWS Lösungen. Wir empfehlen, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln, um Sie beim Debuggen eines etwaigen Fehlers an mehreren Stellen zu unterstützen. Bevor Sie mit der Überwachung von Amazon Kinesis Video Streams beginnen, empfehlen wir Ihnen, einen Überwachungsplan zu erstellen, der Antworten auf die folgenden Fragen enthält:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Nachdem Sie Ihre Überwachungsziele definiert und Ihren Überwachungsplan erstellt haben, besteht der nächste Schritt darin, eine Ausgangsbasis für die normale Leistung von Amazon Kinesis Video Streams in Ihrer Umgebung festzulegen. Sie sollten die Leistung von Amazon Kinesis Video Streams zu verschiedenen Zeiten und unter verschiedenen Lastbedingungen messen. Speichern Sie bei der Überwachung von Amazon Kinesis Video Streams einen Verlauf der von Ihnen gesammelten Überwachungsdaten. Sie können die aktuelle Leistung von Amazon Kinesis Video Streams mit diesen historischen Daten vergleichen, um normale Leistungsmuster und Leistungsanomalien zu identifizieren und Methoden zur Behebung eventuell auftretender Probleme zu entwickeln.

Themen

- [Überwachen Amazon Kinesis Video Streams Streams-Metriken mit CloudWatch](#)
- [Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch](#)
- [Protokollieren Amazon Kinesis Video Streams Streams-API-Aufrufen mit AWS CloudTrail](#)

Überwachen Amazon Kinesis Video Streams Streams-Metriken mit CloudWatch

Sie können einen Kinesis-Videostream mit Amazon überwachen. Amazon CloudWatch sammelt Rohdaten aus Amazon Kinesis Video Streams und verarbeitet sie zu lesbaren Metriken, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden über einen Zeitraum von 15 Monaten aufgezeichnet, sodass Sie auf historische Informationen zugreifen und sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder Ihres Dienstes verschaffen können.

In der [Amazon Kinesis Video Streams Streams-Konsole](#) können Sie CloudWatch Metriken für einen Amazon Kinesis Kinesis-Videostream auf zwei Arten anzeigen:

- Wählen Sie auf der Seite Dashboard die Registerkarte Video streams (Videostreams) im Bereich Account-level metrics for Current Region (Metriken auf Kontoebene für aktuelle Region) aus.
- Wählen Sie auf der Detailseite des Videostreams die Registerkarte Monitoring (Überwachung) aus.


Amazon Kinesis Video Streams bietet die folgenden Metriken:


Metrik	Beschreibung
<code>ArchivedFragmentsConsumed.Media</code>	Die Anzahl der Quotenpunkte für Fragmentmedien, die von allen APIs verbraucht wurden. Eine Erläuterung des Konzepts der Quotenpunkte finden Sie unter the section called “Kontingente für Fragment-Metadaten und Fragment-Medien” . Einheiten: Anzahl
<code>ArchivedFragmentsConsumed.Metadata</code>	Die Anzahl der Quotenpunkte für Fragmente, Metadaten, die von allen APIs genutzt wurden. Eine Erläuterung des Konzepts der Quotenpunkte finden Sie unter the section called “Kontingente für Fragment-Metadaten und Fragment-Medien” . Einheiten: Anzahl
<code>PutMedia.Requests</code>	Die Anzahl der PutMedia API-Anfragen für einen bestimmten Stream.


Metrik	Beschreibung
	Einheiten: Anzahl
<code>PutMedia.IncomingBytes</code>	Die Anzahl der Bytes, die als Teil von <code>PutMedia</code> für den Stream empfangen wurden. Einheiten: Byte
<code>PutMedia.IncomingFragments</code>	Die Anzahl der vollständigen Fragmente, die als Teil von <code>PutMedia</code> für den Stream empfangen wurden. Einheiten: Anzahl
<code>PutMedia.IncomingFrames</code>	Die Anzahl der vollständigen Frames, die als Teil von <code>PutMedia</code> für den Stream empfangen wurden. Einheiten: Anzahl
<code>PutMedia.ActiveConnections</code>	Die Gesamtzahl der Verbindungen zum Service-Host. Einheiten: Anzahl
<code>PutMedia.ConnectionErrors</code>	Die Fehler beim <code>PutMedia</code> Verbindungsaufbau für den Stream. Einheiten: Anzahl
<code>PutMedia.FragmentIngestionLatency</code>	Der Zeitunterschied zwischen dem Zeitpunkt, an dem das erste und das letzte Byte eines Fragments von Amazon Kinesis Video Streams empfangen werden. Einheiten: Millisekunden
<code>PutMedia.FragmentPersistLatency</code>	Die Zeit, die vergeht, bis die vollständigen Fragmentdaten empfangen und archiviert wurden. Einheiten: Anzahl


Metrik	Beschreibung
<code>PutMedia.Latency</code>	<p>Der Zeitunterschied zwischen der Anfrage und der HTTP-Antwort <code>InletService</code> beim Verbindungsaufbau.</p> <p>Einheiten: Anzahl</p>
<code>PutMedia.BufferingAckLatency</code>	<p>Der Zeitunterschied zwischen dem Zeitpunkt, an dem das erste Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, an dem das Buffering ACK für das Fragment gesendet wird.</p> <p>Einheiten: Millisekunden</p>
<code>PutMedia.ReceivedAckLatency</code>	<p>Der Zeitunterschied zwischen dem Zeitpunkt, an dem das letzte Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, an dem das empfangene ACK für das Fragment gesendet wird.</p> <p>Einheiten: Millisekunden</p>
<code>PutMedia.PersistedAckLatency</code>	<p>Der Zeitunterschied zwischen dem Zeitpunkt, an dem das letzte Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, an dem das persistente ACK für das Fragment gesendet wird.</p> <p>Einheiten: Millisekunden</p>
<code>PutMedia.ErrorAckCount</code>	<p>Die Anzahl der Fehler-ACKs, die während der Bearbeitung <code>PutMedia</code> für den Stream gesendet wurden.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
<code>PutMedia.Success</code>	<p>1 für jedes erfolgreich geschriebene Fragment; 0 für jede fehlerhafte Fragment. Der durchschnittliche Wert dieser Metrik zeigt an, wie viele vollständige, gültige Fragmente gesendet werden.</p> <p>Einheiten: Anzahl</p>
<code>GetMedia.Requests</code>	<p>Die Anzahl der <code>GetMedia</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>
<code>GetMedia.OutgoingBytes</code>	<p>Die Gesamtzahl der Byte, die vom Dienst als Teil der <code>GetMedia</code> API für einen bestimmten Stream gesendet wurden.</p> <p>Einheiten: Byte</p>
<code>GetMedia.OutgoingFragments</code>	<p>Die Anzahl der Fragmente, die während der Bearbeitung <code>GetMedia</code> für den Stream gesendet wurden.</p> <p>Einheiten: Anzahl</p>
<code>GetMedia.OutgoingFrames</code>	<p>Die Anzahl der Frames, die während des <code>GetMedia</code> angegebenen Streams gesendet wurden.</p> <p>Einheiten: Anzahl</p>
<code>GetMedia.MillisBehindNow</code>	<p>Der Zeitunterschied zwischen dem aktuellen Serverzeitstempel und dem Serverzeitstempel des zuletzt gesendeten Fragments.</p> <p>Einheiten: Millisekunden</p>


Metrik	Beschreibung
<code>GetMedia.ConnectionErrors</code>	Die Anzahl der Verbindungen, die nicht erfolgreich aufgebaut wurden. Einheiten: Anzahl
<code>GetMedia.Success</code>	1 für jedes erfolgreich gesendete Fragment; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. <div data-bbox="748 604 1510 1066" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfrage-IDs, finden Sie unter AWS Protokollierung der Zusammenfassung von Anfragen und Antworten.</p> </div> Einheiten: Anzahl
<code>GetMediaForFragmentList.OutgoingBytes</code>	Die Gesamtzahl der Byte, die vom Dienst als Teil der <code>GetMediaForFragmentList</code> API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
<code>GetMediaForFragmentList.OutgoingFragments</code>	Die Gesamtzahl der Fragmente, die vom Dienst als Teil der <code>GetMediaForFragmentList</code> API für einen bestimmten Stream gesendet wurden. Einheiten: Anzahl


Metrik	Beschreibung
<code>GetMediaForFragmentList.OutgoingFrames</code>	<p>Die Gesamtzahl der Frames, die vom Service als Teil der <code>GetMediaForFragmentList</code> API für einen bestimmten Stream gesendet wurden.</p> <p>Einheiten: Anzahl</p>
<code>GetMediaForFragmentList.Requests</code>	<p>Die Anzahl der <code>GetMediaForFragmentList</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>
<code>GetMediaForFragmentList.Success</code>	<p>1 für jedes erfolgreich gesendete Fragment; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 863 1507 1318" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>ListFragments.Latency</code>	<p>Die Latenz der <code>ListFragments</code> API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
<code>ListFragments.Requests</code>	<p>Die Anzahl der <code>ListFragments</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
ListFragments.Success	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="750 352 1507 806" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p> </div> <p>Einheiten: Anzahl</p>
GetHLSStreamingSessionURL.Latency	<p>Die Latenz der GetHLSStreamingSessionURL API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
GetHLSStreamingSessionURL.Requests	<p>Die Anzahl der GetHLSStreamingSessionURL API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
GetHLSStreamingSessionURL.Success	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="751 352 1507 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
GetHLSMasterPlaylist.Latency	<p>Die Latenz der GetHLSMasterPlaylist API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
GetHLSMasterPlaylist.Requests	<p>Die Anzahl der GetHLSMasterPlaylist API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
<code>GetHLSMasterPlaylist.Success</code>	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 352 1508 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>GetHLSMediaPlaylist.Latency</code>	<p>Die Latenz der <code>GetHLSMediaPlaylist</code> API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
<code>GetHLSMediaPlaylist.Requests</code>	<p>Die Anzahl der <code>GetHLSMediaPlaylist</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
GetHLSMediaPlaylist.Success	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="751 352 1507 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p> </div> <p>Einheiten: Anzahl</p>
GetMP4InitFragment.Latency	<p>Die Latenz der GetMP4InitFragment API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
GetMP4InitFragment.Requests	<p>Die Anzahl der GetMP4InitFragment API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>


Metrik	Beschreibung
<code>GetMP4InitFragment.Success</code>	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 352 1510 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>GetMP4MediaFragment.Latency</code>	<p>Die Latenz der <code>GetMP4MediaFragment</code> API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
<code>GetMP4MediaFragment.Requests</code>	<p>Die Anzahl der <code>GetMP4MediaFragment</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
<code>GetMP4MediaFragment.Success</code>	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 352 1507 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>GetMP4MediaFragment.OutgoingBytes</code>	<p>Die Gesamtzahl der Byte, die vom Dienst als Teil der <code>GetMP4MediaFragment</code> API für einen bestimmten Stream gesendet wurden.</p> <p>Einheiten: Byte</p>
<code>GetTSFragment.Latency</code>	<p>Die Latenz der <code>GetTSFragment</code> API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
<code>GetTSFragment.Requests</code>	<p>Die Anzahl der <code>GetTSFragment</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
<code>GetTSFfragment.Success</code>	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>GetTSFfragment.OutgoingBytes</code>	<p>Die Gesamtzahl der Byte, die vom Dienst als Teil der <code>GetTSFfragment</code> API für einen bestimmten Stream gesendet wurden.</p> <p>Einheiten: Byte</p>
<code>GetDASHStreamingSessionURL.Latency</code>	<p>Die Latenz der <code>GetDASHStreamingSessionURL</code> API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
<code>GetDASHStreamingSessionURL.Requests</code>	<p>Die Anzahl der <code>GetDASHStreamingSessionURL</code> API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
GetDASHStreamingSessionURL. Success	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="751 352 1507 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
GetDASHManifest.Latency	<p>Die Latenz der GetDASHManifest API-Aufrufe für den angegebenen Stream-Namen.</p> <p>Einheiten: Millisekunden</p>
GetDASHManifest.Requests	<p>Die Anzahl der GetDASHManifest API-Anfragen für einen bestimmten Stream.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
<code>GetDASHManifest.Success</code>	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 350 1510 808" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
<code>GetClip.Latency</code>	<p>Die Latenz der GetClip API-Aufrufe für den angegebenen Videostreamnamen.</p> <p>Einheiten: Millisekunden</p>
<code>GetClip.Requests</code>	<p>Die Anzahl der GetClip API-Anfragen für einen bestimmten Videostream.</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
GetClip.Success	<p>1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an.</p> <div data-bbox="748 352 1507 808"><p> Note</p><p>Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich AWS Anfrage-IDs, finden Sie unter Protokollierung der Zusammenfassung von Anfragen und Antworten.</p></div> <p>Einheiten: Anzahl</p>
GetClip.OutgoingBytes	<p>Die Gesamtzahl der Byte, die vom Dienst als Teil der GetClip API für einen bestimmten Videostream gesendet wurden.</p> <p>Einheiten: Byte</p>

CloudWatch Anleitung zu Metriken

CloudWatch Kennzahlen können dabei helfen, Antworten auf die folgenden Fragen zu finden:

Themen

- [Erreichen Daten den Amazon Kinesis Video Streams Streams-Service?](#)
- [Warum werden Daten vom Amazon Kinesis Video Streams Streams-Service nicht erfolgreich aufgenommen?](#)
- [Warum können die Daten vom Amazon Kinesis Video Streams Streams-Service nicht mit der gleichen Geschwindigkeit gelesen werden, mit der sie vom Produzenten gesendet werden?](#)
- [Warum befindet sich kein Video in der Konsole, oder warum wird das Video verzögert abgespielt?](#)

- [Worum handelt es sich bei der Verzögerung beim Lesen von Echtzeitdaten und warum bleibt der Client zeitlich hinter dem Datenstromkopf zurück?](#)
- [Liest der Client Daten aus dem Kinesis-Videostream und mit welcher Geschwindigkeit?](#)
- [Warum kann der Client keine Daten aus dem Kinesis-Videostream lesen?](#)

Erreichen Daten den Amazon Kinesis Video Streams Streams-Service?

Relevante Metriken:

- `PutMedia.IncomingBytes`
- `PutMedia.IncomingFragments`
- `PutMedia.IncomingFrames`

Aktionselemente:

- Wenn diese Messwerte sinken, überprüfen Sie, ob Ihre Anwendung immer noch Daten an den Service sendet.
- Prüfen Sie die Netzwerkbandbreite. Wenn Ihre Netzwerkbandbreite nicht ausreicht, könnte dies die Geschwindigkeit, mit der der Service die Daten empfängt, verlangsamen.

Warum werden Daten vom Amazon Kinesis Video Streams Streams-Service nicht erfolgreich aufgenommen?

Relevante Metriken:

- `PutMedia.Requests`
- `PutMedia.ConnectionErrors`
- `PutMedia.Success`
- `PutMedia.ErrorAckCount`

Aktionselemente:

- Wenn es eine Zunahme von `gibtPutMedia.ConnectionErrors`, schauen Sie sich die HTTP-Antwort und die Fehlercodes an, die der Producer-Client erhalten hat, um zu sehen, welche Fehler beim Verbindungsaufbau aufgetreten sind.

- Wenn es zu einem Rückgang `PutMedia.Success` oder Anstieg `PutMedia.ErrorAckCount`, sehen Sie sich den Ack-Fehlercode in den vom Dienst gesendeten Ack-Antworten an, um herauszufinden, warum die Datenaufnahme fehlschlägt. [Weitere Informationen finden Sie unter `.Values.AckErrorCode`](#)

Warum können die Daten vom Amazon Kinesis Video Streams Streams-Service nicht mit der gleichen Geschwindigkeit gelesen werden, mit der sie vom Produzenten gesendet werden?

Relevante Metriken:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Aktionselemente:

- Wenn diese Messwerte sinken, überprüfen Sie die Netzwerkbandbreite Ihrer Verbindungen. Verbindungen mit geringer Bandbreite können dazu führen, dass die Daten mit geringerer Geschwindigkeit zum Service gelangen.

Warum befindet sich kein Video in der Konsole, oder warum wird das Video verzögert abgespielt?

Relevante Metriken:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Aktionselemente:

- Wenn die Netzwerkbandbreite zunimmt `PutMedia.FragmentIngestionLatency` oder sinkt `PutMedia.IncomingFragments`, überprüfen Sie die Netzwerkbandbreite und ob die Daten weiterhin gesendet werden.
- Wenn es zu einem Rückgang kommt `PutMedia.Success`, überprüfen Sie die ACK-Fehlercodes. Weitere Informationen finden Sie unter [AckErrorCode.Values](#).
- Wenn es zu einer Zunahme von `PutMedia.FragmentPersistLatency` oder kommt `ListFragments.Latency`, liegt höchstwahrscheinlich ein Serviceproblem vor. Wenn der Zustand über einen längeren Zeitraum andauert, erkundigen Sie sich bei Ihrem Kundendienstkontakt, ob ein Problem mit Ihrem Service vorliegt.

Worum handelt es sich bei der Verzögerung beim Lesen von Echtzeitdaten und warum bleibt der Client zeitlich hinter dem Datenstromkopf zurück?

Relevante Metriken:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Aktionselemente:

- Wenn der Wert zunimmt, kann es sein `GetMedia.ConnectionErrors`, dass der Verbraucher beim Lesen des Streams ins Hintertreffen gerät, weil er häufig versucht, erneut eine Verbindung zum Stream herzustellen. Sehen Sie sich die HTTP-Antwort-/Fehlercodes an, die für die Anfrage `GetMedia` zurückgegeben wurden.
- Wenn es zu einem Rückgang kommt `GetMedia.Success`, liegt das wahrscheinlich daran, dass der Dienst die Daten nicht an den Verbraucher senden kann, was zu einem Verbindungsabbruch führen würde, und die Verbindung von Verbrauchern wieder herstellen würde, was dazu führen würde, dass der Verbraucher hinter der Spitze des Streams zurückbleibt.
- Wenn es zu einem Anstieg kommt `GetMedia.MillisBehindNow`, schauen Sie sich Ihre Bandbreitenbeschränkungen an, um festzustellen, ob Sie die Daten aufgrund der geringeren Bandbreite langsamer empfangen.

Liest der Client Daten aus dem Kinesis-Videostream und mit welcher Geschwindigkeit?

Relevante Metriken:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

Aktionselemente:

- Diese Kennzahlen geben die Geschwindigkeit an, mit der Echtzeit- und archivierte Daten gelesen werden.

Warum kann der Client keine Daten aus dem Kinesis-Videostream lesen?

Relevante Metriken:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Aktionselemente:

- Wenn es eine Zunahme von `gibtGetMedia.ConnectionErrors`, sehen Sie sich die HTTP-Antwort und die Fehlercodes an, die von der `GetMedia` Anfrage zurückgegeben wurden. Weitere Informationen finden Sie unter [AckErrorCode.Values](#).
- Wenn Sie versuchen, die neuesten Daten oder Live-Daten `PutMedia.IncomingBytes` zu lesen, überprüfen Sie, ob Daten in den Stream gelangen, die der Dienst an die Verbraucher senden kann.
- Wenn es zu einem Rückgang bei `GetMedia.Success` oder `kommtGetMediaForFragmentList.Success`, liegt das wahrscheinlich daran, dass der Dienst die Daten nicht an den Verbraucher senden kann. Wenn der Zustand über einen längeren Zeitraum

andauert, erkundigen Sie sich bei Ihrem Kundendienstkontakt, ob ein Problem mit Ihrem Service vorliegt.

Überwachung des Amazon Kinesis Video Streams Edge Agent mit CloudWatch



Sie können den Amazon Kinesis Video Streams Edge Agent mithilfe von Amazon überwachen CloudWatch, der Rohdaten sammelt und zu lesbaren, nahezu in Echtzeit verfügbaren Metriken verarbeitet. Diese Statistiken werden über einen Zeitraum von 15 Monaten aufgezeichnet. Mit diesen historischen Informationen können Sie sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder des Amazon Kinesis Video Streams Edge Agent-Service verschaffen.

Gehen Sie wie folgt vor, um die Metriken einzusehen:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie in der linken Navigationsleiste unter Metriken die Option Alle Metriken aus.
3. Wählen Sie die Registerkarte Durchsuchen und dann den EdgeRuntimeAgentbenutzerdefinierten Namespace aus.

Amazon Kinesis Video Streams Edge Agent veröffentlicht die folgenden Metriken unter dem Namespace: `EdgeRuntimeAgent`

Dimensionen	Status	Beschreibung
Name des Streams, RecordJob	In Ausführung	Veröffentlicht kontinuierlich, wenn der RecordJob läuft. Einheiten: Keine. „1“ wird veröffentlicht, solange sie RecordJob sich in diesem Status befindet.
	FatalError	Publiziert, wenn ein RecordJob fataler Fehler auftritt. Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.

Dimensionen	Status	Beschreibung
		<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note Weitere Informationen finden Sie in den Protokollen.</p> </div>
	Completed	<p>Veröffentlicht, wenn a abgeschlossen RecordJob ist.</p> <p>Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.</p>
Name des Streams, UploadJob	In Ausführung	<p>Veröffentlicht kontinuierlich, wenn der UploadJob läuft.</p> <p>Einheiten: Keine. „1“ wird veröffentlicht, solange sie UploadJob sich in diesem Status befindet.</p>
	FatalError	<p>Wird veröffentlicht, wenn die UploadJob fatalen Fehler auftreten.</p> <p>Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff; margin-top: 10px;"> <p> Note Weitere Informationen finden Sie in den Protokollen.</p> </div>
	Completed	<p>Veröffentlicht, wenn der abgeschlossen UploadJob ist.</p> <p>Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.</p>
Stream-Name	PercentageSpaceUsed	<p>Dies ist der Prozentsatz, der vom gesamten Speicherplatz, der in Amazon Kinesis Video Streams Edge Agent-Konfigurationen für Aufnahmemedien zugewiesen wurde, genutzt wird. Weitere Informationen finden Sie unter the section called “LocalSizeConfig”.</p> <p>Einheiten: Prozentsatz (Skala 0—1).</p>

Dimensionen	Status	Beschreibung
Thing-Name	Am Leben	<p>Veröffentlicht jede Minute vom Amazon Kinesis Video Streams Edge Agent aus, unabhängig davon, welche Konfigurationen darauf ausgeführt werden.</p> <p>Dies kann verwendet werden, um zu ermitteln, ob der Amazon Kinesis Video Streams Edge Agent aktiv ist und bereit ist, Konfigurationen zu akzeptieren.</p> <p>Einheiten: Keine. „1“ wird jede Minute veröffentlicht.</p>
	RecordJobs.HealthyJobCount	<p>Gesamtzahl der laufenden und geplanten Aufzeichnungsaufträge auf dem Amazon Kinesis Video Streams Edge Agent.</p> <p>Einheiten: Anzahl.</p>
	UploadJobs.HealthyJobCount	<p>Gesamtzahl der laufenden und geplanten Upload-Jobs auf dem Amazon Kinesis Video Streams Edge Agent.</p> <p>Einheiten: Anzahl.</p>
	RecordJobs.UnhealthyJobCount	<p>Gesamtzahl der aktuell fehlerhaften Aufzeichnungsaufträge.</p> <p>Einheiten: Anzahl.</p>
	UploadJobs.UnhealthyJobCount	<p>Gesamtzahl der aktuell fehlerhaften Upload-Jobs.</p> <p>Einheiten: Anzahl.</p>
	RecordJobs.RunningJobCount	<p>Gesamtzahl der aktiv ausgeführten Rekordjobs.</p> <p>Einheiten: Anzahl.</p>
	UploadJobs.RunningJobCount	<p>Gesamtzahl der aktiv ausgeführten Upload-Jobs.</p> <p>Einheiten: Anzahl.</p>

Dimensionen	Status	Beschreibung
	RecordJobsEdgeConfigCount	Gesamtzahl der Datensatzkonfigurationen, die auf dem Amazon Kinesis Video Streams Edge Agent in Bearbeitung sind. Einheiten: Anzahl.
	UploadJobsEdgeConfigCount	Gesamtzahl der Upload-Konfigurationen, die auf dem Amazon Kinesis Video Streams Edge Agent in Bearbeitung sind. Einheiten: Anzahl.

CloudWatch Leitfaden für Kennzahlen für Amazon Kinesis Video Streams Edge Agent

CloudWatch Metriken können nützlich sein, um Antworten auf die folgenden Fragen zu finden:

Themen

- [Hat der Amazon Kinesis Video Streams Edge Agent genügend Speicherplatz für Aufzeichnungen?](#)
- [Ist der Amazon Kinesis Video Streams Edge Agent am Leben?](#)
- [Gibt es ungesunde Jobs?](#)
- [Müssen irgendwelche Jobs von außen eingreifen?](#)

Hat der Amazon Kinesis Video Streams Edge Agent genügend Speicherplatz für Aufzeichnungen?

Relevante Metriken: PercentageSpaceUsed

Aktion: Keine Aktion erforderlich.

Ist der Amazon Kinesis Video Streams Edge Agent am Leben?

Relevante Metriken: Alive

Maßnahme: Wenn Sie diese Metrik zu irgendeinem Zeitpunkt nicht mehr erhalten, bedeutet dies, dass der Amazon Kinesis Video Streams Edge Agent auf eine oder mehrere der folgenden Ursachen gestoßen ist:

- Ein Problem mit der Anwendungslaufzeit: Speicher- oder andere Ressourcenbeschränkungen, Fehler usw.
- Das AWS IoT Gerät, auf dem der Agent ausgeführt wird, wenn es heruntergefahren, abgestürzt oder beendet wurde
- Das AWS IoT Gerät hat keine Netzwerkverbindung


Gibt es ungesunde Jobs?

Relevante Metriken:

- `RecordJobs.UnhealthyJobCount`
- `UploadJobs.UnhealthyJobCount`

Aktion: Untersuchen Sie die Protokolle und suchen Sie nach der `FatalError` Metrik.

- Wenn die `FatalError` Metrik vorhanden ist, ist ein schwerwiegender Fehler aufgetreten und Sie müssen den Job manuell neu starten. Überprüfen Sie die Protokolle und beheben Sie das `ProblemStartEdgeConfigurationUpdate`, bevor Sie den Job manuell neu starten.
- Wenn die `FatalError` Metrik nicht vorhanden ist, ist ein vorübergehender (nicht schwerwiegender) Fehler aufgetreten, und der Amazon Kinesis Video Streams Edge Agent versucht den Job erneut.

 Note

Verwenden Sie, damit der Agent erneut versucht, einen Auftrag mit schwerwiegendem Fehler auszuführen. [the section called “StartEdgeConfigurationUpdate”](#)

Müssen irgendwelche Jobs von außen eingreifen?

Relevante Metriken:

- `PercentageSpaceUsed`— Wenn dieser Wert einen bestimmten Wert überschreitet, wird der Aufnahmevorgang angehalten und erst wieder aufgenommen, wenn Speicherplatz verfügbar ist (wenn das Medium nicht mehr aufbewahrt wird). Sie können eine aktualisierte Konfiguration mit einem höheren Wert senden, um den Job `MaxLocalMediaSizeInMB` sofort zu aktualisieren.
- `RecordJob.FatalError/UploadJob.FatalError`— Untersuchen Sie die Protokolle des Agenten und senden Sie die Konfiguration erneut, damit der Job wieder aufgenommen werden kann.

Aktion: Führen Sie einen API-Aufruf mit der Konfiguration durch, um Jobs neu zu starten, bei denen dieses Problem auftritt.

Protokollieren Amazon Kinesis Video Streams Streams-API-Aufrufen mit AWS CloudTrail

Amazon Kinesis Video Streams arbeitet mit einem Service AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS-Service in Amazon Kinesis Video Streams bereitstellt. CloudTrail erfasst alle API-Aufrufe für Amazon Kinesis Video Streams als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der Amazon Kinesis Video Streams Streams-Konsole und Code-Aufrufe an die Amazon Kinesis Video Streams Streams-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übertragung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Kinesis Video Streams. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon Kinesis Video Streams gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen darüber CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Amazon Kinesis Video Streams und CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn unterstützte Ereignisaktivitäten in Amazon Kinesis Video Streams auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen im CloudTrail Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Amazon Kinesis Video Streams, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere konfigurieren, AWS-Services um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Amazon Kinesis Video Streams unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.

- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

Weitere Informationen finden Sie unter dem [CloudTrailUserIdentity-Element](#).

Beispiel: Einträge in der Amazon Kinesis Video Streams Streams-Protokolldatei

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anforderungsparameter. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die [CreateStream](#)Aktion demonstriert.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2018-05-25T00:16:31Z",
      "eventSource": "kinesisvideo.amazonaws.com",
      "eventName": "CreateStream",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
      "requestParameters": {
        "streamName": "VideoStream",

```

```

        "dataRetentionInHours": 2,
        "mediaType": "mediaType",
        "kmsKeyId": "arn:aws:kms::us-east-1:123456789012:alias",
"deviceName": "my-device"
    },
    "responseElements": {
"streamARN":arn:aws:kinesisvideo:us-east-1:123456789012:stream/VideoStream/12345"
        },
        "requestID": "db6c59f8-c757-11e3-bc3b-57923b443c1c",
        "eventID": "b7acfc0-6ca9-4ee1-a3d7-c4e8d420d99b"
    },
    {
        "eventVersion": "1.05",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
        "eventTime": "2018-05-25:17:06Z",
        "eventSource": "kinesisvideo.amazonaws.com",
        "eventName": "DeleteStream",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
        "requestParameters": {
            "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/
VideoStream/12345",
            "currentVersion": "keqrjeqkj9"
        },
        "responseElements": null,
        "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
        "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
    },
    {
        "eventVersion": "1.05",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",

```

```

        "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:02Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "DescribeStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
        "streamName": "VideoStream"
    },
    "responseElements": null,
    "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:03Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "GetDataEndpoint",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
        "streamName": "VideoStream",
        "apiName": "LIST_FRAGMENTS"
    },
    "responseElements": null,
    "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",

```

```
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-05-25T00:16:56Z",
  "eventSource": "kinesisvideo.amazonaws.com",
  "eventName": "ListStreams",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
  "requestParameters": {
    "maxResults": 100,
    "streamNameCondition": {"comparisonValue":"MyVideoStream"
comparisonOperator":"BEGINS_WITH"}}
  },
  "responseElements": null,
  "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
  "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
}
]
```

Service Quotas für Kinesis Video Streams

Kinesis Video Streams verfügt über die folgenden Service Quotas:

Important

Die folgenden Servicekontingente sind entweder Soft [s], die durch Senden eines Support-Tickets aktualisiert werden können, oder Hard [h], die nicht erhöht werden können. In den folgenden Tabellen sehen Sie [s] und [h] neben dem individuellen Servicekontingent.


API-Servicekontingente der Steuerebene

Im folgenden Abschnitt werden Service Quotas für APIs auf Steuerebene beschrieben. TPS steht für Transaktionen pro Sekunde.

Wenn ein Anforderungslimit auf Konto- oder Ressourcenebene erreicht ist, `ClientLimitExceededException` wird eine ausgelöst.

API-Servicekontingente der Steuerebene

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream-Ebene	Relevante Ausnahmen und Hinweise
the section called "CreateStream"	50 TPS [s]	10 000 Streams pro Konto [s] in den Regionen USA Ost (Nord-Virginia) und USA West (Oregon). 5 000 Streams pro Konto [s] in allen		Diese API kann über Geräte, CLIs, SDK-gesteuerten Zugriff und die Konsole aufgerufen werden. Es ist nur ein API-Aufruf erfolgreich, wenn der Stream noch nicht vorhanden ist.

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
		<p>anderen unterstützten Regionen.</p> <div data-bbox="591 478 792 1852" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>Dieses Limit kann auf bis zu 100 000 (oder mehr) Streams pro Konto [s] erhöht werden. Melden Sie sich bei der AWS Management Console unter an <u>https://console.a</u></p> </div>		

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
		<p>ws.amazon.com/ und reichen Sie einen Fall zur Erhöhung des Service-Limits ein, damit Kinesis Video Streams eine Erhöhung dieses Limits beantragen kann.</p>		
<p>the section called “DeleteEdgeConfiguration”</p>	<p>10 TPS [h]</p>	<p>N/A</p>	<p>1 TPS [h]</p>	

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
the section called “DeleteStream”	50 TPS [h]	N/A	5 TPS [h]	
the section called “DescribeEdgeConfiguration”	50 TPS [h]	N/A	5 TPS [h]	
the section called “DescribeImageGenerationConfiguration”	50 TPS [h]	N/A	5 TPS [h]	
the section called “DescribeMappedResourceConfiguration”	50 TPS [h]	N/A	5 TPS [h]	
the section called “DescribeNotificationConfiguration”	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
the section called "Describe Stream"	300 TPS [h]	N/A	5 TPS [h]	
the section called "GetDataEndpoint"	300 TPS [h]	N/A	5 TPS [h]	Wird alle 45 Minuten aufgerufen, um das Streaming-Token für die meisten PutMedia/GetMedia-Anwendungsfälle zu aktualisieren. Das Caching von Datenendpunkten ist sicher, wenn die Anwendung sie bei einem Fehler erneut lädt.
the section called "ListEdgeAgentConfigurations"	50 TPS [h]	N/A	N/A	
the section called "ListStreams"	50 TPS [h]	N/A		
the section called "ListTagsForStream"	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
the section called “StartEdgeConfigurationUpdate”	10 TPS [h]	N/A	1 TPS [h]	
the section called “TagStream”	50 TPS [h]	N/A	5 TPS [h]	
the section called “UntagStream”	50 TPS [h]	N/A	5 TPS [h]	
the section called “UpdateDataRetention”	50 TPS [h]	N/A	5 TPS [h]	
the section called “UpdateImageGenerationConfiguration”	50 TPS [h]	N/A	5 TPS [h]	
the section called “UpdateNotificationConfiguration”	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anforderung	Kontolimit: Streams	Grenzwerte auf Stream- Ebene	Relevante Ausnahmen und Hinweise
the section called "UpdateStream"	50 TPS [h]	N/A	5 TPS [h]	

API-Servicekontingente für Medien und archivierte Medien

Im folgenden Abschnitt werden Service Quotas für Medien und archivierte Medien-APIs beschrieben.

Wenn ein Anforderungslimit auf Konto- oder Ressourcenebene erreicht ist, `ClientLimitExceededException` wird ausgelöst.

Wenn ein Grenzwert auf Verbindungsebene erreicht wird, wird die Ausnahme `ConnectionLimitExceededException` ausgelöst.


Die folgenden Fehler und Bestätigungen werden ausgelöst, wenn ein Grenzwert auf Fragmentebene erreicht wird:


- Eine `MIN_FRAGMENT_DURATION_REACHED`-Bestätigung wird zurückgegeben, wenn ein Fragment unterhalb der Mindestdauer liegt.
- Eine `MAX_FRAGMENT_DURATION_REACHED`-Bestätigung wird zurückgegeben, wenn ein Fragment oberhalb der Höchstdauer liegt.
- Eine `MAX_FRAGMENT_SIZE`-Bestätigung wird zurückgegeben, wenn ein Fragment die maximale Datengröße überschreitet.
- Eine `FragmentLimitExceeded`-Ausnahme wird ausgelöst, wenn ein Fragmentgrenzwert in einer `GetMediaForFragmentList`-Operation erreicht wird.

API-Servicekontingente auf Datenebene

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
the section called "PutMedia"	5 TPS [h]	1 [s]	12,5 MB/Sekunde oder 100 Mbit/s [s] pro Stream	<ul style="list-style-type: none"> • Minimale Fragmentdauer: 1 Sekunde [h] • Maximale Fragmentdauer: 20 Sekunden [h] • Maximale Fragmentgröße: 50 MB [h] • Maximale Anzahl von Tracks: 3 [s] • Maximale Anzahl gesendeter Fragmente pro Sekunde: 5 [h] 	Eine typische PutMedia-Anforderung enthält die Daten für mehrere Sekunden, was zu einem niedrigeren TPS pro Stream führt. Wenn es mehrere gleichzeitige Verbindungen gibt, die die Kontingente überschreiten, wird die letzte Verbindung akzeptiert.

API	Grenzwert e auf Stream-Eb- ene	Grenzwert e auf Verbindun- gsebene	Bandbreit- enbegrenz- ung	Grenzwert e auf Fragmente- bene	Relevante Ausnahmen und Hinweise
					<ul style="list-style-type: none"> • Maximales Fragmentanzahl limit: 10 Tags [h]
the section called "GetClip"	N/A	N/A	100 MB Größenbeschränkung [h]	Maximale Anzahl der Fragmente : 200 [h]	
the section called "GetDASHStreamingSessionURL"	25 TPS [h]	N/A	–	N/A	
the section called "GetHLSStreamingSessionURL"	25 TPS [h]	N/A	–	–	

API	Grenzwert e auf Stream-Eb- ene	Grenzwert e auf Verbindun- gsebene	Bandbreit- enbegrenz- ung	Grenzwert e auf Fragmente- bene	Relevante Ausnahmen und Hinweise
the section called “GetImages”	–	N/A	100 MB [h]	N/A	Die maximale Anzahl von Bildern pro Anforderung beträgt 100 [h]. <div data-bbox="1156 577 1466 1008"> <p> Note Der Mindestwert für <code>SamplingInterval</code> ist 200 Millisekunden (ms), was 5 Bildern pro Sekunde entspricht.</p> </div>

API	Grenzwert e auf Stream-Ebene	Grenzwert e auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwert e auf Fragmentebene	Relevante Ausnahmen und Hinweise
the section called "GetMedia"	5 TPS [h]	3 [s]	25 MB/s oder 200 Mbit/s [s]	Maximale Anzahl gesendeter Fragmente pro Sekunde: 6 [h]	<p>Ein eindeutig verbrauchender Client sollte nicht mehr als zwei oder drei TPS benötigen, da die Anwendung nach dem Herstellen der Verbindung kontinuierlich lesen sollte.</p> <p>Wenn ein typisches Fragment ungefähr 5 MB groß ist, bedeutet dieses Limit ~75 MBps pro Kinesis-Videostream. Ein solcher Stream hat eine ausgehende Bitrate, die etwa doppelt so hoch ist wie die maximale eingehende Bitrate des Streams.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>GetMedia wird nicht für die HLS/DASH-Wiedergabe verwendet.</p> </div>

API	Grenzwert e auf Stream-Ebene	Grenzwert e auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwert e auf Fragmentebene	Relevante Ausnahmen und Hinweise
the section called “GetMediaForFragmentList”	N/A	5 [s]	25 MB/s oder 200 Mbit/s [s]	Maximale Anzahl der Fragmente: 1000 [h]	Fünf fragmentasierte verbundene Anwendungen können gleichzeitig aufrufen <code>GetMediaForFragmentList</code> . Weitere Verbindungen werden abgelehnt.

API-Servicekontingente für Videowiedergabeprotokolle

API	Grenzwerte auf Sitzungsebene	Grenzwerte auf Fragmentebene
GetDASHManifestPlaylist	5 TPS [h]	Maximale Anzahl von Fragmenten pro Wiedergabepalette: 5000 [h]
GetHLSMasterPlaylist	5 TPS [h]	N/A
GetHLSMediaPlaylist	5 TPS [h]	Maximale Anzahl von Fragmenten pro Wiedergabepalette: 5000 [h]
GetMP4InitFragment	5 TPS [h]	N/A
GetMP4MediaFragment	20 TPS [h]	N/A
GetTSFragment	20 TPS [h]	N/A

Kontingente für Fragment-Metadaten und Fragment-Medien

[APIs von Kinesis Video Streams für den Zugriff auf archivierte Medien](#) werden auf der Grundlage der Anzahl der angeforderten Fragmente und nicht der Anzahl der API-Aufrufe gedrosselt. APIs werden sowohl durch die Anzahl der Fragmentmetadaten als auch durch die Anzahl der angeforderten Fragmentmedien begrenzt. Die Fragmentmetadaten und Fragmentmedienkontingente werden pro Stream angewendet. Mit anderen Worten, Anforderungen für Fragmentmetadaten oder Medien in einem Stream gelten nicht für die Kontingente eines anderen Streams. Innerhalb eines bestimmten Streams wird jedes Kontingent jedoch für mehrere APIs freigegeben. Dies bedeutet, dass Anforderungen für Fragmente über verschiedene APIs hinweg für einen bestimmten Stream aus demselben Kontingent verbrauchen. Wenn entweder das Fragment-Metadaten- oder Fragment-Medienkontingent für einen Stream überschritten wird, gibt die API eine `ClientLimitExceededException` zurück. Die folgenden Tabellen zeigen, wie die APIs von jedem der beiden Kontingenttypen verbrauchen. Nehmen Sie für die zweite Spalte in diesen Tabellen an, dass, wenn ein Stream ein Kontingent von N hat, die APIs N Punkte haben, die von diesem Kontingenttyp für diesen Stream verwendet werden sollen. Die `GetClip` API wird in beiden Tabellen angezeigt.

Nutzung des Fragment-Metadatenkontingents

API	Anzahl der pro Anforderung verbrauchten Kontingentpunkte	Gemeinsam genutztes Kontingent (N)
the section called “ListFragments”	Wert des <code>MaxResults</code> Parameters	10 000 Kontingentpunkte pro Sekunde, pro Stream [h]
the section called “GetClip”	Anzahl der Fragmente im resultierenden Fragment	
GetHLSMediaPlaylist	Wert des <code>MaxMediaPlaylistFragmentResults</code> Parameters	
GetDASHManifest	Wert des <code>MaxManifestFragmentResults</code> Parameters	

API	Anzahl der pro Anforderung verbrauchten Kontingentpunkte	Gemeinsam genutztes Kontingent (N)
the section called "GetImages"	Wert von 400 + maximale Anzahl der angeforderten Bilder	

Nutzung von Fragment-Medienkontingenten

API	Anzahl der pro Anforderung verbrauchten Kontingentpunkte	Gemeinsam genutztes Kontingent (N)
the section called "GetMediaForFragmentList"	Anzahl der Fragmente im Parameter Fragmente	500 Kontingentpunkte pro Sekunde, pro Stream [h]
the section called "GetClip"	Anzahl der Fragmente im resultierenden Fragment	
GetMP4MediaFragment	1	
GetTSFragment	1	
the section called "GetImages"	Maximale Anzahl angeforderter Bilder	

Bei einem Kontingent von 500 Fragmentmedien pro Sekunde werden beispielsweise die folgenden Aufrufmuster für einen bestimmten Stream unterstützt:

- 5 Anforderungen pro Sekunde an GetClip mit 100 Fragmenten in jedem -CLI.
- 100 Anfragen pro Sekunde an GetClip mit 5 Fragmenten in jedem -CLI.
- 2 Anforderungen pro Sekunde an GetClip mit 100 Fragmenten in jedem clip und 3 Anforderungen pro Sekunde an GetMediaForFragmentList in jedem clip.
- 400 Anforderungen pro Sekunde an GetMP4MediaFragment und 100 Anforderungen pro Sekunde an GetTSFragment.

Diese Kontingente haben eine wichtige Auswirkung auf die Anzahl der HLS- und MPEG-DASH-Sitzungen, die pro Stream unterstützt werden können. Die Anzahl der HLS- und DASH-Sitzungen, die von Media Playern zu einem bestimmten Zeitpunkt verwendet werden können, ist unbegrenzt. Daher ist es wichtig, dass die Wiedergabeanwendung nicht zu viele Sitzungen gleichzeitig verwendet. In den folgenden beiden Beispielen wird beschrieben, wie Sie die Anzahl der gleichzeitigen Wiedergabesitzungen ermitteln, die unterstützt werden können:

Beispiel 1: Live-Streaming

In einem Live-Streaming-Szenario mit HLS mit Fragmenten mit einer Dauer von 1 Sekunde, einer Audio- und Videospur und `MaxMediaPlaylistFragmentResults` einem Wert von fünf führt ein Media Player in der Regel zwei Aufrufe an `GetHLSMediaPlaylist` pro Sekunde aus. Ein Aufruf bezieht sich auf die neuesten Videometadaten und ein weiterer auf die entsprechenden Audiometadaten. Die beiden Aufrufe verbrauchen jeweils fünf Fragmentmetadaten-Kontingentpunkte. Es führt außerdem zwei Aufrufe an `GetMP4MediaFragment` pro Sekunde durch: einen Aufruf für das neueste Video und einen weiteren für das entsprechende Audio. Jeder Aufruf verbraucht ein einzelnes Fragment-Medientoken, sodass insgesamt zwei Token verbraucht werden.

In diesem Szenario können bis zu 250 gleichzeitige Wiedergabesitzungen unterstützt werden. Bei 250 Sitzungen verbraucht dieses Szenario 2 500 Fragmentmetadaten-Kontingentpunkte pro Sekunde (deutlich unter dem 10 000-Kontingent) und 500 Fragment-Medienkontingentpunkte pro Sekunde.

Beispiel 2: On-Demand-Wiedergabe

In einem On-Demand-Wiedergabeszenario eines früheren Ereignisses mit MPEG-DASH, einer Audio- und Videospur und `MaxManifestFragmentResults` einer Einstellung auf 1 000, ruft ein Media Player in der Regel `GetDASHManifest` einmal zu Beginn der Sitzung auf (wobei 1 000 Fragmentmetadatenkontingentpunkte verbraucht werden), und ruft `GetMP4MediaFragment` mit einer Geschwindigkeit von bis zu 5-mal pro Sekunde auf (wobei 5 Fragmente Medienkontingentpunkte verbraucht werden), bis alle Fragmente geladen werden. In diesem Szenario können bis zu 10 neue Sitzungen pro Sekunde gestartet werden (rechte Seite mit dem Kontingent von 10 000 Fragmentmetadaten pro Sekunde), und bis zu 100 Sitzungen können aktiv Fragmentmedien mit einer Rate von 5 pro Sekunde laden (rechte Seite mit dem Kontingent von 500 Fragmentmedien pro Sekunde).

Sie können `ArchivedFragmentsConsumed.Metadata` und `verwendenArchivedFragmentsConsumed.Media`, um den Verbrauch von Fragment-Metadaten bzw. Fragment-Medienkontingentpunkten zu überwachen. Weitere Informationen zur Überwachung finden Sie unter [Überwachen](#).

Fragment-Metadatenkontingente

Die folgenden Servicekontingente gelten für das Hinzufügen von Fragmentmetadaten zu Fragmenten in einem Kinesis-Videostream:

- Sie können einem Fragment bis zu 10 Metadatenelemente voranstellen.
- Der Metadaten-Name eines Fragments kann bis zu 128 Bytes lang sein.
- Der Metadaten-Wert eines Fragments kann bis zu 256 Bytes lang sein.
- Ein Fragment-Metadatenname darf nicht mit der Zeichenfolge „“ beginnenAWS. Wenn ein entsprechendes Metadatenelement hinzugefügt wird, gibt die Methode `putFragmentMetadata` im PIC einen `STATUS_INVALID_METADATA_NAME`-Fehler (Fehlercode `0x52000077`) zurück. Ihre Anwendung kann dann entweder den Fehler ignorieren (der PIC fügt das Metadatenelement nicht hinzu) oder auf den Fehler reagieren.

Stream-Tags

Diese Metadaten-Schlüssel-Wert-Paare gelten für die gesamte Kinesis-Video-Streams-Ressource, nicht für einzelne Fragmente, die in einem Kinesis-Video-Stream enthalten sind.

Jeder Kinesis-Videostream unterstützt bis zu 50 Tags.

Einschränkungen [the section called “TagStream”](#) für die Schlüssel und Werte von Stream-Tags finden Sie unter .

Problembhebung bei Kinesis Video Streams

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme mit Amazon Kinesis Video Streams zu beheben.

Themen

- [Fehlerbehebung bei allgemeinen Problemen](#)
- [Behebung von API-Problemen](#)
- [Behebung von HLS-Problemen](#)
- [Behebung von Java-Problemen](#)
- [Behebung von Problemen mit der Producer Library](#)
- [Behebung von Problemen mit der Stream Parser-Bibliothek](#)

Fehlerbehebung bei allgemeinen Problemen

In diesem Abschnitt werden allgemeine Probleme beschrieben, die bei der Arbeit mit Kinesis Video Streams auftreten können.

Problembereiche

- [Latenz zu hoch](#)

Latenz zu hoch

Die Latenz kann durch die Dauer der Fragmente verursacht werden, die an den Kinesis Video Streams Streams-Dienst gesendet werden. Eine Möglichkeit, die Latenzzeit zwischen Producer und Service zu reduzieren, besteht darin, die Medien-Pipeline so zu konfigurieren, dass kürzere Fragmentlaufzeiten entstehen.

Um die Anzahl der in jedem Fragment gesendeten Frames zu reduzieren, reduzieren Sie den folgenden Wert in `inkinesis_video_gstreamer_sample_app.cpp`:

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512, NULL);
```


Note

Die Latenzen sind im Mozilla Firefox-Browser aufgrund der internen Implementierung des Video-Renderings höher.

Behebung von API-Problemen

In diesem Abschnitt werden API-Probleme beschrieben, die bei der Arbeit mit Kinesis Video Streams auftreten können.

Problembereiche

- [Fehler: „Unbekannte Optionen“](#)
- [Fehler: „Unable to determine service/operation name to be authorized“ \(Zu autorisierender Service-/Operationsname kann nicht ermittelt werden\)](#)
- [Fehler: „Failed to put a frame in the stream“ \(Es konnte kein Frame in den Stream eingefügt werden\)](#)
- [Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“](#)
- [Fehler: "STATUS_STORE_OUT_OF_MEMORY"](#)

Fehler: „Unbekannte Optionen“

`GetMedia` und `GetMediaForFragmentList` können mit dem folgenden Fehler fehlschlagen:

```
Unknown options: <filename>.mkv
```

Dieser Fehler tritt auf, wenn Sie den AWS CLI mit einem output Typ von `json` konfiguriert haben. Konfigurieren Sie den AWS CLI mit dem Standardausgabety `(none)` neu. Informationen zur Konfiguration von finden [Sie AWS CLI unter configure](#) in der AWS CLI Befehlsreferenz.

Fehler: „Unable to determine service/operation name to be authorized“ (Zu autorisierender Service-/Operationsname kann nicht ermittelt werden)

`GetMedia` kann mit dem folgenden Fehler fehlschlagen:

```
Unable to determine service/operation name to be authorized
```

Dieser Fehler kann auftreten, wenn der Endpunkt nicht korrekt angegeben ist. Wenn Sie den Endpunkt erhalten, stellen Sie sicher, dass Sie den folgenden Parameter in den `GetDataEndpoint` Aufruf aufnehmen, je nachdem, welche API aufgerufen werden soll:

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

Fehler: „Failed to put a frame in the stream“ (Es konnte kein Frame in den Stream eingefügt werden)

`PutMedia` kann mit dem folgenden Fehler fehlschlagen:

```
Failed to put a frame in the stream
```

Dieser Fehler kann auftreten, wenn für den Dienst keine Konnektivität oder Berechtigungen verfügbar sind. Führen Sie Folgendes in der AWS CLI aus und stellen Sie sicher, dass die Stream-Informationen abgerufen werden können:

```
aws kinesishvideo describe-stream --stream-name StreamName --endpoint https://ServiceEndpoint.kinesishvideo.region.amazonaws.com
```

Schlägt der Aufruf fehl, finden Sie [AWS CLI weitere Informationen unter Problembehandlung](#).

Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“

`PutMedia` kann mit dem folgenden Fehler fehlschlagen:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was received
```

Dieser Fehler kann auftreten, wenn `PushbackInputStream` nicht korrekt implementiert ist. Stellen Sie sicher, dass die `unread()` Methoden korrekt implementiert sind.

Fehler: "STATUS_STORE_OUT_OF_MEMORY"

PutMedia kann mit dem folgenden Fehler fehlschlagen:

```
The content store is out of memory.
```

Dieser Fehler tritt auf, wenn der Inhaltsspeicher nicht ausreichend groß dimensioniert wurde. Um den Inhaltsspeicher zu vergrößern, erhöhen Sie den Wert von `StorageInfo.storageSize`. Weitere Informationen finden Sie unter [StorageInfo](#).

Behebung von HLS-Problemen

Falls Ihr Videostream nicht korrekt wiedergegeben wird, finden Sie weitere Informationen unter [the section called "Behebung von HLS-Problemen"](#).

Behebung von Java-Problemen

In diesem Abschnitt wird beschrieben, wie Sie häufig auftretende Java-Probleme beheben können, die bei der Arbeit mit Kinesis Video Streams auftreten.

Problembereiche

- [Aktivieren von Java-Protokollen](#)

Aktivieren von Java-Protokollen

Um Probleme mit Java-Beispielen und -Bibliotheken zu beheben, ist es hilfreich, die Debug-Logs zu aktivieren und zu untersuchen. Gehen Sie zum Aktivieren der Debug-Protokolle folgendermaßen vor:

1. Fügen Sie `log4j` in die Datei `pom.xml` im Knoten `dependencies` ein:

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. Erstellen Sie im Verzeichnis `target/classes` eine Datei namens `log4j.properties` mit folgendem Inhalt:

```
# Root logger option
log4j.rootLogger=DEBUG, stdout

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:
%L - %m%n

log4j.logger.org.apache.http.wire=DEBUG
```

Die Debug-Protokolle werden anschließend zur IDE-Konsole ausgegeben.

Behebung von Problemen mit der Producer Library

In diesem Abschnitt werden Probleme beschrieben, die bei der Verwendung von [Producer-Bibliotheken](#) auftreten können.

Problembereiche

- [Das Producer SDK kann nicht kompiliert werden](#)
- [Der Videostream wird nicht in der Konsole angezeigt](#)
- [Fehler: „Security token included in the request is invalid“ \(Das Sicherheits-Token der Anfrage ist ungültig\) beim Streamen von Daten mit der GStreamer-Demo-Anwendung](#)
- [Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“](#)
- [GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt“](#)
- [Fehler: „Failed to allocate heap“ \(Heap konnte nicht zugeordnet werden\) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi](#)
- [Fehler: „Illegal Instruction“ \(Ungültige Anweisung\) beim Ausführen der GStreamer-Demo auf Raspberry Pi](#)
- [Die Kamera kann nicht auf Raspberry Pi geladen werden.](#)
- [Kamera kann auf macOS High Sierra nicht gefunden werden.](#)
- [Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden](#)
- [Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung](#)

- [Zeitstempel-/Zeitraum-Assertion während der Laufzeit auf Raspberry Pi](#)
- [Assertion auf `gst_value_set_fraction_range_full` auf Raspberry Pi](#)
- [STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA \(0x3200000d\)-Fehler unter Android](#)
- [Fehler: Die maximale Fragmentdauer wurde erreicht](#)
- ["Ungültiger Objektname abgelaufen", Fehler bei der Verwendung der IoT-Autorisierung](#)

Das Producer SDK kann nicht kompiliert werden

Überprüfen Sie, ob die erforderlichen Bibliotheken in Ihrem Pfad vorhanden sind. Zur Bestätigung führen Sie den folgenden Befehl aus:

```
$ env | grep LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/local/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/
kinesis-video-native-build/downloads/local/lib
```

Der Videostream wird nicht in der Konsole angezeigt

Damit der Videostream in der Konsole angezeigt werden kann, muss er mit H.264 im AvCC-Format codiert worden sein. Prüfen Sie Folgendes, wenn der Stream nicht angezeigt wird:

- Für [NAL Adaptation Flags](#) ist `NAL_ADAPTATION_ANNEXB_NALS` | `NAL_ADAPTATION_ANNEXB_CPD_NALS` eingestellt, wenn der Originalstream im Format Annex-B vorliegt. Dies ist der Standardwert im `StreamDefinition`-Konstruktor.
- Sie stellen die Code Private Data ordnungsgemäß bereit. Für H.264 sind dies der Sequenzparametersatz (SPS) und der Bildparametersatz (PPS). Abhängig von der Medienquelle können diese Daten separat aus der Medienquelle abgerufen oder im Frame codiert werden.

Viele einfache Streams liegen im folgenden Format vor (Ab ist der Annex-B-Startcode – 001 oder 0001):

```
Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/
B-frame) Ab(P/B-frame)
```

Der CPD (Codec Private Data) kann, sofern H.264 als SPS und PPS im Stream enthalten ist, an das AvCC-Format angepasst werden. Sofern die Medienpipeline die CPD nicht separat angibt, kann die Anwendung die CPD aus dem Frame extrahieren, indem sie nach dem ersten Idr-Frame

sucht (der die SPS und PPS enthalten sollte), die beiden NALUs (die dann Ab (Sps) Ab (Pps) sein werden) extrahiert und im CPD in `insertStreamDefinition`.

Fehler: „Security token included in the request is invalid“ (Das Sicherheits-Token der Anfrage ist ungültig) beim Streamen von Daten mit der GStreamer-Demo-Anwendung

Wenn dieser Fehler auftritt, liegt ein Problem mit Ihren Anmeldeinformationen vor. Überprüfen Sie Folgendes:

- Wenn Sie temporäre Anmeldeinformationen verwenden, müssen Sie den Sitzungs-Token angeben.
- Überprüfen Sie, ob Ihre temporären Anmeldeinformationen abgelaufen sind.
- Vergewissern Sie sich, dass Sie über die richtigen Rechte verfügen.
- Vergewissern Sie sich unter MacOS, dass Sie keine Anmeldeinformationen im Schlüsselbund zwischengespeichert haben.

Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“

Wenn dieser Fehler auftritt, sind die Zeitstempel im Quell-Stream nicht richtig gesetzt. Gehen Sie wie folgt vor:

- Verwenden Sie die neueste SDK-Version, in der es möglicherweise schon ein Update gibt, das Ihr Problem behebt.
- Stellen Sie für den hochwertigen Stream eine höhere Bitrate ein und korrigieren Sie jeglichen Jitter im Quellstream, sofern die Kamera dies unterstützt.

GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt“

Das Streamen kann unter OS X mit der folgenden Meldung abgebrochen werden:

```
Debugging information: gstbasesrc.c(2939): void gst_base_src_loop(GstPad *) (): /
GstPipeline:test-pipeline/GstAutoVideoSrc:source/GstAVFVideoSrc:source-actual-src-
avfvide:
streaming stopped, reason not-negotiated (-4)
```

Eine mögliche Problemumgebung besteht darin, die Parameter für die Bildrate aus dem `gst_caps_new_simple` Aufruf zu entfernen in: `kinesis_video_gstreamer_sample_app.cpp`

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

Fehler: „Failed to allocate heap“ (Heap konnte nicht zugeordnet werden) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi

Die GStreamer-Beispielanwendung versucht, 512 MB RAM zuzuweisen, die möglicherweise nicht auf Ihrem System verfügbar sind. Sie können die Größe des zugewiesenen Speicherplatzes verringern, indem Sie den folgenden Wert in `KinesisVideoProducer.cpp` verkleinern:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Fehler: „Illegal Instruction“ (Ungültige Anweisung) beim Ausführen der GStreamer-Demo auf Raspberry Pi

Wenn beim Ausführen der GStreamer-Demo der folgende Fehler auftritt, überprüfen Sie, ob Sie die Anwendung für die richtige Version Ihres Geräts kompiliert haben. (Stellen Sie beispielsweise sicher, dass Sie nicht für Raspberry Pi 3 kompilieren, wenn Sie auf Raspberry Pi 2 laufen.)

```
INFO - Initializing curl.
Illegal instruction
```

Die Kamera kann nicht auf Raspberry Pi geladen werden.

Um zu überprüfen, ob die Kamera geladen wurde, führen Sie die folgenden Schritte aus:

```
$ ls /dev/video*
```

Ist die Suche ergebnislos, führen Sie die folgenden Schritte aus:

```
$ vcgencmd get_camera
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
supported=1 detected=1
```

Wenn der Treiber die Kamera nicht erkennt, gehen Sie wie folgt vor:

1. Überprüfen Sie die technische Einrichtung der Kamera und prüfen Sie, ob sie richtig angeschlossen ist.
2. Führen Sie die folgenden Schritte aus, um die Firmware zu aktualisieren:

```
$ sudo rpi-update
```

3. Starten Sie das Gerät neu.
4. Führen Sie die folgenden Schritte aus, um den Treiber zu laden:

```
$ sudo modprobe bcm2835-v4l2
```

5. Überprüfen Sie, ob die Kamera erkannt wurde:

```
$ ls /dev/video*
```

Kamera kann auf macOS High Sierra nicht gefunden werden.

Unter macOS High Sierra kann die Demo-Anwendung die Kamera nicht finden, wenn mehr als eine Kamera verfügbar ist.

Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden

Um diesen Fehler zu beheben, aktualisieren Sie Ihre Installation von Xcode auf die neueste Version.

Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung

Um Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung zu beheben, kopieren Sie [diese Zertifikatdatei](#) zu `/etc/ssl/cert.pem`.

Zeitstempel-/Zeitraum-Assertion während der Laufzeit auf Raspberry Pi

Wenn zur Laufzeit eine Zeitstempel-/Zeitraum-Assertion auftritt, aktualisieren Sie die Firmware und starten Sie das Gerät neu:

```
$ sudo rpi-update
$ sudo reboot
```

Assertion auf `gst_value_set_fraction_range_full` auf Raspberry Pi

Die folgende Assertion erscheint, wenn der `uv4l`-Service ausgeführt wird:

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
denominator_end) < 0' failed
```

Wenn dies der Fall ist, stoppen Sie den `uv4l`-Service und starten Sie die Anwendung neu.

STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d)-Fehler unter Android

Der folgende Fehler tritt auf, wenn [NAL Adaptation Flags](#) für den Medien-Stream fehlerhaft ist:

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Wenn dieser Fehler auftritt, geben Sie das richtige `.withNalAdaptationFlags`-Flag für Ihr Medium (Beispiel: `NAL_ADAPTATION_ANNEXB_CPD_NALS`) an. Geben Sie dieses Flag in der folgenden Zeile des [Android Producer-Bibliothek](#) an:

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/.java#L169> `AmazonKinesisVideoDemoAppStreamConfigurationFragment`

Fehler: Die maximale Fragmentdauer wurde erreicht

Dieser Fehler tritt auf, wenn ein Medienfragment in einem Stream die maximale Fragmentdauer überschreitet. Informationen zur maximalen Fragmentdauer finden Sie im [the section called “API-Servicekontingente für Medien und archivierte Medien”](#) Abschnitt.

Versuchen Sie, das Problem wie folgt zu beheben:

- Wenn Sie eine Webcam/USB-Kamera verwenden, führen Sie einen der folgenden Schritte aus:
 - Wenn Sie die Keyframe-basierte Fragmentierung verwenden, stellen Sie den Encoder so ein, dass er Keyframes innerhalb von 10 Sekunden bereitstellt.
 - Wenn Sie keine Keyframe-basierte Fragmentierung verwenden, legen Sie bei der Definition des Stream-Ins die maximale Fragmentdauer auf einen Wert fest [Schritt 2: Schreiben und untersuchen Sie den Code](#), der weniger als 10 Sekunden beträgt.
 - Wenn Sie Software-Encoder (wie x264) in der GStreamer-Pipeline verwenden, können Sie das key-int-max Attribut innerhalb von 10 Sekunden auf einen Wert setzen. Stellen Sie es beispielsweise key-int-max auf 60 und die FPS auf 30 ein, um Keyframes alle 2 Sekunden zu aktivieren.
- Wenn Sie eine RPI-Kamera verwenden, legen Sie das Keyframe-Intervall-Attribut auf weniger als 10 Sekunden fest.
- Wenn Sie eine IP-Kamera (RTSP) verwenden, setzen Sie die GOP-Größe auf 60.

"Ungültiger Objektname abgelaufen", Fehler bei der Verwendung der IoT-Autorisierung

Um diesen Fehler (HTTP Error 403: Response: {"message":"Invalid thing name passed"}) zu vermeiden, wenn Sie IoT-Anmeldeinformationen für die Autorisierung verwenden, stellen Sie sicher, dass der Wert von stream-name (ein erforderlicher Parameter des kvssink Elements) mit dem Wert von identisch ist `istiot-thingname`. Weitere Informationen finden Sie unter [Referenz zu GStreamer-Elementparametern](#).

Behebung von Problemen mit der Stream Parser-Bibliothek

In diesem Abschnitt werden Probleme beschrieben, die bei der Verwendung von ["Stream Parser"-Bibliothek](#) auftreten können.

Problembereiche

- [Es kann nicht auf einen einzelnen Frame aus dem Stream zugegriffen werden](#)
- [Fragmentdekodierungsfehler](#)

Es kann nicht auf einen einzelnen Frame aus dem Stream zugegriffen werden

Um auf einen einzelnen Frame aus einer Streaming-Quelle in Ihrer Consumer-Anwendung zuzugreifen, stellen Sie sicher, dass Ihr Stream die richtigen privaten Codec-Daten enthält. Informationen über das Format der Daten in einem Stream finden Sie unter [Datenmodell](#).

[In der folgenden Testdatei auf der GitHub Website erfahren Sie, wie Sie private Codec-Daten für den Zugriff auf einen Frame verwenden können: `.java KinesisVideoRendererExampleTest`](#)

Fragmentdekodierungsfehler

Wenn Ihre Fragmente nicht korrekt in einem H.264-Format und auf einem vom Browser unterstützten Level kodiert sind, wird möglicherweise die folgende Fehlermeldung angezeigt, wenn Sie Ihren Stream in der Konsole wiedergeben:

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264 content
```

Wenn dies passiert, überprüfen Sie Folgendes:

- Die Auflösung der Frames stimmt mit der in den Codec Private Data angegebenen Auflösung überein.
- Das H.264-Profil und das Level der kodierten Frames stimmt mit dem Profil und dem Level überein, die in den Codec Private Data angegeben sind.
- Der Browser unterstützt die Profil/Level-Kombination. Die meisten aktuellen Browser unterstützen alle Profil- und Level-Kombinationen.
- Die Zeitstempel sind präzise und in der richtigen Reihenfolge, und es werden keine doppelten Zeitstempel erstellt.
- Ihre Anwendung kodiert die Frame-Daten mit dem H.264-Format.

Dokumentverlauf für Amazon Kinesis Video Streams

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von Amazon Kinesis Video Streams beschrieben.

- Aktuelle API-Version: 2017-11-29
- Letzte Aktualisierung der Dokumentation: 27. Juni 2023

Änderung	Beschreibung	Datum
Edge-to-Cloud-Verbindung von Amazon Kinesis Video Streams Edge-Agent	Neue Feature-Version. Weitere Informationen finden Sie unter Edge-Agent .	27. Juni 2023
Erste Schritte: Daten an einen Kinesis-Videostream senden	Grundlegendes Tutorial zum Senden von Mediendaten von einer Kamera an einen Kinesis-Videostream. Weitere Informationen finden Sie unter Daten an einen Amazon Kinesis Kinesis-Videostream senden .	21. Januar 2019
Bibliotheksvorlage zur Integration mit SageMaker	Beispielanwendung für Kinesis Video Streams, die verwendet wird, SageMaker um zu erkennen, wann bestimmte Objekte in einem Amazon Kinesis-Videostream erscheinen. Weitere Informationen finden Sie unter SageMaker .	19. November 2018
Streaming-Metadaten	Sie können das Producer SDK verwenden, um Metadaten in einen Kinesis-Videostream	28. September 2018

Änderung	Beschreibung	Datum
	m einzubetten. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .	
C++ Producer SDK-Protokollierung	Sie können die Protokollierung für C++ Producer SDK-Anwendungen konfigurieren. Weitere Informationen finden Sie unter Verwenden der Protokollierung mit dem C++ Producer SDK .	18. Juli 2018
HLS-Video-Streaming	Sie können jetzt einen Kinesis-Videostream mithilfe von HTTP Live Streaming ansehen. Weitere Informationen finden Sie unter Wiedergabe von Kinesis Video Streams .	13. Juli 2018
Streaming aus einer RTSP-Quelle	Beispielanwendung für Kinesis Video Streams, die in einem Docker-Container ausgeführt wird und Video von einer RTSP-Quelle streamt. Weitere Informationen finden Sie unter RTSP und Docker .	20. Juni 2018
C++ Producer SDK GStreamer-Plugin	Zeigt, wie das C++ Producer-Bibliothek erstellt werden muss, um es als GStreamer-Ziel zu verwenden. Weitere Informationen finden Sie unter GStreamer .	15. Juni 2018

Änderung	Beschreibung	Datum
Producer SDK-Callbacks-Referenzdokumentation	Referenzdokumentation für die vom Produzentenbibliotheken von Kinesis Video Streams verwendeten Callbacks. Weitere Informationen finden Sie unter SDK-Rückrufe von Produzenten .	12. Juni 2018
Systemanforderungen	Dokumentation für Arbeitsspeicher- und Speicheranforderungen für Producer-Geräte und SDK. Weitere Informationen finden Sie unter Kinesis Video Streams Systemanforderungen .	30. Mai 2018
CloudTrail-Support	Dokumentation CloudTrail zur Überwachung der API-Nutzung. Weitere Informationen finden Sie unter Protokollieren Amazon Kinesis Video Streams Streams-API-Aufrufen mit AWS CloudTrail .	24. Mai 2018
Producer SDK-Strukturen – Referenzdokumentation	Referenzdokumentation für die vom Produzentenbibliotheken von Kinesis Video Streams verwendeten Strukturen. Weitere Informationen erhalten Sie unter SDK-Strukturen für Hersteller und Kinesis-VideoStream-Strukturen .	7. Mai 2018

Änderung	Beschreibung	Datum
Renderer-Beispiel-Dokumentation	Dokumentation für die Renderer-Beispielanwendung, die zeigt, wie Frames aus einem Kinesis-Video-Stream dekodiert und angezeigt werden. Weitere Informationen finden Sie unter Beispiel: Analysieren und Rendern von Kinesis-Video-Streams-Fragmenten .	15. März 2018
Producer-SDK-Limits-Referenzdokumentation	Informationen zu Einschränkungen hinsichtlich Operationen in der C++ Producer-Bibliothek . Weitere Informationen finden Sie unter Beschränkungen des Producer-SDK .	13. März 2018
Überwachung	Informationen zur Überwachung der Metriken und API-Aufrufe von Kinesis Video Streams mithilfe von Amazon CloudWatch und AWS CloudTrail. Weitere Informationen finden Sie unter Überwachung von Amazon Kinesis Video Streams .	5. Februar 2018
Network Abstraction Layer (NAL) Adaptation Flag-Referenz	Informationen zum Festlegen von NAL-Anpassungsflags beim Konsumieren von Video-Streaming. Weitere Informationen finden Sie unter NAL Adaptation Flags .	15. Januar 2018

Änderung	Beschreibung	Datum
Android-Unterstützung für Video-Streaming	Kinesis Video Streams unterstützt jetzt das Streamen von Videos von Android-Geräten. Weitere Informationen finden Sie unter Android Producer-Bibliothek .	12. Januar 2018
Kinesis-Video-Beispiel-Dokumentation	Dokumentation für die Kinesis Video-Beispielanwendung, die zeigt, wie die Parser-Bibliothek für Kinesis-Videostreams in einer Anwendung verwendet wird. Weitere Informationen finden Sie unter KinesisVideoExample .	9. Januar 2018
Kinesis Video Streams-Dokumentation veröffentlicht	Dies ist die erste veröffentlichte Version des Amazon Kinesis Video Streams-Entwicklerhandbuchs.	29. November 2017

API-Referenz

Die Abschnitte unter diesem Knoten enthalten die API-Referenzdokumentation. Verwenden Sie das Inhaltsverzeichnis im linken Bereich, um zu den verschiedenen API-Referenzabschnitten zu gelangen.

Aktionen

Die folgenden Aktionen werden von Amazon Kinesis Video Streams unterstützt:

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)

- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)
- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Media unterstützt:

- [GetMedia](#)
- [PutMedia](#)

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Archived Media unterstützt:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

Die folgenden Aktionen werden von Amazon Kinesis Video Signaling Channels unterstützt:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

Die folgenden Aktionen werden von Amazon Kinesis Video WebRTC Storage unterstützt:

- [JoinStorageSession](#)

Amazon Kinesis Video Streams

Die folgenden Aktionen werden von Amazon Kinesis Video Streams unterstützt:

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)
- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)

- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)

CreateSignalingChannel

Bedienung: Amazon Kinesis Video Streams

Erzeugt einen Signalkanal.

CreateSignalingChannel ist ein asynchroner Vorgang.

Anforderungssyntax

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelName": "string",
  "ChannelType": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelName

Ein Name für den Signalisierungskanal, den Sie erstellen. Dieser Wert muss für jedes AWS-Konto und jede AWS-Region eindeutig sein.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Ja

ChannelType

Ein Typ des Signalisierungskanals, den Sie erstellen. Derzeit ist SINGLE_MASTER der einzige unterstützte Kanaltyp.

Typ: Zeichenfolge

Zulässige Werte: SINGLE_MASTER | FULL_MESH

Required: No

SingleMasterConfiguration

Eine Struktur, die die Konfiguration für den SINGLE_MASTER Kanaltyp enthält.

Typ: [SingleMasterConfiguration](#) Objekt

Required: No

Tags

Eine Reihe von Tags (Schlüssel-Wert-Paare), die Sie diesem Kanal zuordnen möchten.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 50 Elemente.

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelARN": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

ChannelARN

Der Amazon-Ressourcenname (ARN) des erstellten Kanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

AccountChannelLimitExceededException

Sie haben in dieser Region die maximale Anzahl an aktiven Signalkanälen dafür AWS-Konto erreicht.

HTTP Status Code: 400

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

TagsPerResourceExceededLimitException

Sie haben das Limit an Tags überschritten, die Sie der Ressource zuordnen können. Ein Kinesis-Videostream kann bis zu 50 Tags unterstützen.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

CreateStream

Bedienung: Amazon Kinesis Video Streams

Erzeugt einen neuen Kinesis-Videostream.

Wenn Sie einen neuen Stream erstellen, weist Kinesis Video Streams ihm eine Versionsnummer zu. Wenn Sie die Metadaten des Streams ändern, aktualisiert Kinesis Video Streams die Version.

CreateStream ist ein asynchroner Vorgang.

Weitere Information zur Funktionsweise des Services finden Sie unter [Funktionsweise](#).

Sie müssen über Berechtigungen für die `KinesisVideo:CreateStream`-Aktion verfügen.

Anforderungssyntax

```
POST /createStream HTTP/1.1
Content-type: application/json

{
  "DataRetentionInHours": number,
  "DeviceName": "string",
  "KmsKeyId": "string",
  "MediaType": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[DataRetentionInHours](#)

Die Anzahl der Stunden, für die Sie die Daten im Stream behalten möchten. Kinesis Video Streams speichert die Daten in einem Datenspeicher, der dem Datenstrom zugeordnet ist.

Der Standardwert ist 0. Er gibt an, dass der Stream keine Daten beibehält.

Wenn der `DataRetentionInHours` Wert 0 ist, können Verbraucher immer noch die Fragmente verwenden, die im Service-Host-Puffer verbleiben. Dieser Puffer hat ein Aufbewahrungszeitlimit von 5 Minuten und ein Aufbewahrungsspeicherlimit von 200 MB. Fragmente werden aus dem Puffer entfernt, wenn einer der Grenzwerte erreicht ist.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Required: No

DeviceName

Der Name des Geräts, das in den Stream schreibt.

Note

In der aktuellen Implementierung verwendet Kinesis Video Streams diesen Namen nicht.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

KmsKeyId

Die ID des Schlüssels AWS Key Management Service (AWS KMS), den Kinesis Video Streams zum Verschlüsseln von Stream-Daten verwenden soll.

Wenn keine Schlüssel-ID angegeben ist, wird standardmäßig der von Kinesis Video verwaltete Schlüssel (`AWS/kinesisvideo`) verwendet.

Weitere Informationen finden Sie unter. [DescribeKey](#)

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 2048 Zeichen.

Pattern: . +

Required: No

MediaType

Der Medientyp des Streams. Nutzer des Streams können diese Informationen bei der Verarbeitung des Streams verwenden. Weitere Informationen zu Medientypen finden Sie unter [Medientypen](#). Wenn Sie sich für die Angabe von `contentType` entscheiden, finden Sie Richtlinien unter [Benennungsvoraussetzungen](#).

Zu den gültigen Werten gehören beispielsweise „video/h264“ und „video/h264, audio/aac“.

Dieser Parameter ist optional; der Standardwert ist (oder leer in JSON). `null`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Required: No

StreamName

Ein Name für den Stream, den Sie erstellen.

Der Streamname ist eine Kennung für den Stream und muss für jedes Konto und jede Region eindeutig sein.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Ja

Tags

Eine Liste von Tags, die dem angegebenen Stream zugeordnet werden sollen. Jedes Tag ist ein Schlüssel-Wert-Paar (der Wert ist optional).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 50 Elementen.

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: $^([\backslash p\{L}\backslash p\{Z}\backslash p\{N}_ \. : / = + \ - @]^*)\$$

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertemuster: $[\backslash p\{L}\backslash p\{Z}\backslash p\{N}_ \. : / = + \ - @]^*$

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "StreamARN": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[StreamARN](#)

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: $arn:[a-z\d-]+ : kinesisvideo:[a-z0-9-]+ : [0-9]+ : [a-z]+ / [a-zA-Z0-9_ \. -]+ / [0-9]+$

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccountStreamLimitExceededException

Die Anzahl der für das Konto erstellten Streams ist zu hoch.

HTTP Status Code: 400

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

DeviceStreamLimitExceededException

Nicht implementiert.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidDeviceException

Nicht implementiert.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API, um den Status der Ressource zu bestimmen.

HTTP Status Code: 400

TagsPerResourceExceededLimitException

Sie haben das Limit an Tags überschritten, die Sie der Ressource zuordnen können. Ein Kinesis-Videostream kann bis zu 50 Tags unterstützen.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DeleteEdgeConfiguration

Bedienung: Amazon Kinesis Video Streams

Eine asynchrone API, die die bestehende Edge-Konfiguration eines Streams sowie die entsprechenden Medien aus dem Edge-Agent löscht.

Wenn Sie diese API aufrufen, wird der Synchronisierungsstatus auf gesetzt. DELETING Ein Löschvorgang wird gestartet, bei dem aktive Edge-Jobs gestoppt und alle Medien vom Edge-Gerät gelöscht werden. Die Dauer des Löschvorgangs hängt von der Gesamtmenge der gespeicherten Medien ab. Schlägt der Löschvorgang fehl, ändert sich der Synchronisierungsstatus aufDELETE_FAILED. Sie müssen den Löschvorgang erneut versuchen.

Wenn der Löschvorgang erfolgreich abgeschlossen wurde, kann nicht mehr auf die Edge-Konfiguration zugegriffen werden.

Note

Diese API ist in der Region AWS Afrika (Kapstadt), af-south-1, nicht verfügbar.

Anforderungssyntax

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream. Geben Sie entweder den oder den anStreamName. StreamARN

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, aus dem die Edge-Konfiguration gelöscht werden soll. Geben Sie entweder den StreamName oder den anStreamARN.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

StreamEdgeConfigurationNotFoundException

Die Ausnahme wurde gerendert, wenn Amazon Kinesis Video Stream die von Ihnen angegebene Edge-Konfiguration eines Streams nicht finden kann.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DeleteSignalingChannel

Bedienung: Amazon Kinesis Video Streams

Löscht einen angegebenen Signalkanal. `DeleteSignalingChannel` ist eine asynchrone Operation. Wenn Sie die aktuelle Version des Kanals nicht angeben, wird die neueste Version gelöscht.

Anforderungssyntax

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Signalkanals, den Sie löschen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

CurrentVersion

Die aktuelle Version des Signalkanals, den Sie löschen möchten. Sie können die aktuelle Version abrufen, indem Sie die `ListSignalingChannels` API-Operationen `DescribeSignalingChannel` oder aufrufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene

Eingang StreamARN oder Eingang nicht im Status Aktiv ChannelARN befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

`ResourceNotFoundException`

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

`VersionMismatchException`

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die [DescribeStream](#) API, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DeleteStream

Bedienung: Amazon Kinesis Video Streams

Löscht einen Kinesis-Videostream und die darin enthaltenen Daten.

Diese Methode markiert den Stream zum Löschen und macht den Zugriff auf die Daten im Stream sofort unmöglich.

Um sicherzustellen, dass Sie über die neueste Version des Streams verfügen, bevor Sie ihn löschen, können Sie die Stream-Version angeben. Kinesis Video Streams weist jedem Stream eine Version zu. Wenn Sie einen Stream aktualisieren, weist Kinesis Video Streams eine neue Versionsnummer zu. Verwenden Sie die API, um die neueste Stream-Version zu erhalten. `DescribeStream`

Diese Operation setzt eine Berechtigung für die `KinesisVideo:DeleteStream`-Aktion voraus.

Anforderungssyntax

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "StreamARN": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

CurrentVersion

Optional: Die Version des Streams, die Sie löschen möchten.

Geben Sie die Version als Sicherheitsmaßnahme an, um sicherzustellen, dass Sie den richtigen Stream löschen. Verwenden Sie die `DescribeStream` API, um die Stream-Version abzurufen.

Wenn nicht angegeben, `CreationTime` wird nur überprüft, bevor der Stream gelöscht wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: [a-zA-Z0-9]+

Required: No

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, den Sie löschen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

VersionMismatchException

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die [DescribeStream](#) API, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)

- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeEdgeConfiguration

Bedienung: Amazon Kinesis Video Streams

Beschreibt die Edge-Konfiguration eines Streams, die mithilfe der `StartEdgeConfigurationUpdate` API festgelegt wurde, und den aktuellen Status der Recorder- und Uploader-Jobs des Edge-Agenten. Verwenden Sie diese API, um den Status der Konfiguration abzurufen und festzustellen, ob die Konfiguration mit dem Edge-Agent synchronisiert ist. Verwenden Sie diese API, um den Zustand des Edge-Agenten zu bewerten.

Note

Diese API ist in der Region AWS Afrika (Kapstadt), `af-south-1`, nicht verfügbar.

Anforderungssyntax

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream. Geben Sie entweder den oder den `anStreamName`. `StreamARN`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, dessen Edge-Konfiguration Sie aktualisieren möchten. Geben Sie entweder den StreamName oder den anStreamARN.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "EdgeAgentStatus": {
    "LastRecorderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "RecorderStatus": "string"
    },
    "LastUploaderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "UploaderStatus": "string"
    }
  },
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,

```

```
    "LocalSizeConfig": {
      "MaxLocalMediaSizeInMB": number,
      "StrategyOnFullSize": "string"
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      },
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    }
  },
  "FailedStatusDetails": "string",
  "LastUpdatedTime": number,
  "StreamARN": "string",
  "StreamName": "string",
  "SyncStatus": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

CreationTime

Der Zeitstempel, zu dem die Edge-Konfiguration eines Streams zum ersten Mal erstellt wurde.

Typ: Zeitstempel

EdgeAgentStatus

Ein Objekt, das die neuesten Statusdetails für die Recorder- und Uploader-Jobs eines Edge-Agenten enthält. Verwenden Sie diese Informationen, um den aktuellen Zustand eines Edge-Agents zu ermitteln.

Typ: [EdgeAgentStatus](#) Objekt

EdgeConfig

Eine Beschreibung der Edge-Konfiguration des Streams, die für die Synchronisierung mit der Edge Agent IoT Greengrass-Komponente verwendet wird. Die Edge Agent-Komponente wird auf einem IoT Hub-Gerät ausgeführt, das bei Ihnen vor Ort eingerichtet ist.

Typ: [EdgeConfig](#) Objekt

FailedStatusDetails

Eine Beschreibung des generierten Fehlerstatus.

Typ: Zeichenfolge

LastUpdatedTime

Der Zeitstempel, zu dem die Edge-Konfiguration eines Streams zuletzt aktualisiert wurde.

Typ: Zeitstempel

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

Der Name des Streams, von dem aus die Edge-Konfiguration aktualisiert wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

SyncStatus

Der neueste Status des Edge-Konfigurationsupdates.

Typ: Zeichenfolge

Zulässige Werte: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

StreamEdgeConfigurationNotFoundException

Die Ausnahme wurde gerendert, wenn Amazon Kinesis Video Stream die von Ihnen angegebene Edge-Konfiguration eines Streams nicht finden kann.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeImageGenerationConfiguration

Bedienung: Amazon Kinesis Video Streams

Ruft das ImageGenerationConfiguration für einen bestimmten Kinesis-Videostream ab.

Anforderungssyntax

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, aus dem die Konfiguration zur Bilderzeugung abgerufen werden soll. Sie müssen entweder den StreamName oder den StreamARN angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, aus dem die Konfiguration zur Image-Generierung abgerufen werden soll. Sie müssen entweder den StreamName oder den angebenStreamARN.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[ImageGenerationConfiguration](#)

Die Struktur, die die Informationen enthält, die für die Übertragung von Kinesis-Videostream (KVS) -Bildern erforderlich sind. Wenn diese Struktur Null ist, wird die Konfiguration aus dem Stream gelöscht.

Typ: [ImageGenerationConfiguration](#) Objekt

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)

- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeMappedResourceConfiguration

Bedienung: Amazon Kinesis Video Streams

Gibt die aktuellsten Informationen über den Stream zurück. Das `streamName` oder `streamARN` sollte in der Eingabe angegeben werden.

Anforderungssyntax

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

MaxResults

Die maximale Anzahl von Ergebnissen, die in der Antwort zurückgegeben werden sollen.

Typ: Ganzzahl

Gültiger Bereich: Fester Wert von 1.

Required: No

NextToken

Das Token, das Sie in Ihrer nächsten Anfrage angeben müssen, um weitere Ergebnisse zu erhalten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Name des -Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "MappedResourceConfigurationList": [
    {
      "ARN": "string",
      "Type": "string"
    }
  ],
  "NextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[MappedResourceConfigurationList](#)

Eine Struktur, die die Eigenschaften der Medienspeicherkonfiguration kapselt oder enthält.

Typ: Array von [MappedResourceConfigurationListItem](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 1 Element.

[NextToken](#)

Das Token, das in der NextToken Anforderung verwendet wurde, um die nächsten Ergebnisse abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeMediaStorageConfiguration

Bedienung: Amazon Kinesis Video Streams

Gibt die aktuellsten Informationen über den Kanal zurück. Geben Sie das ChannelName oder ChannelARN in der Eingabe an.

Anforderungssyntax

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Kanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

Der Name des Kanal.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

MediaStorageConfiguration

Eine Struktur, die die Konfigurationseigenschaften des Medienspeichers kapselt oder enthält.

Typ: MediaStorageConfiguration Objekt

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#)

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeNotificationConfiguration

Bedienung: Amazon Kinesis Video Streams

Ruft das `NotificationConfiguration` für einen bestimmten Kinesis-Videostream ab.

Anforderungssyntax

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, von dem Sie die Benachrichtigungskonfiguration abrufen möchten. Sie müssen entweder den `StreamName` oder den `StreamArn` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, aus dem die Benachrichtigungskonfiguration abgerufen werden soll. Sie müssen entweder den `StreamName` oder den `StreamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

NotificationConfiguration

Die Struktur, die die für Benachrichtigungen erforderlichen Informationen enthält. Wenn die Struktur Null ist, wird die Konfiguration aus dem Stream gelöscht.

Typ: NotificationConfiguration Objekt

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter Häufige Fehler.

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeSignalingChannel

Bedienung: Amazon Kinesis Video Streams

Gibt die aktuellsten Informationen über den Signalkanal zurück. Sie müssen entweder den Namen oder den Amazon-Ressourcennamen (ARN) des Kanals angeben, den Sie beschreiben möchten.

Anforderungssyntax

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der ARN des Signalkanals, den Sie beschreiben möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

Der Name des Signalkanals, den Sie beschreiben möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelInfo": {
    "ChannelARN": "string",
    "ChannelName": "string",
    "ChannelStatus": "string",
    "ChannelType": "string",
    "CreationTime": number,
    "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
    },
    "Version": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

ChannelInfo

Eine Struktur, die die Metadaten und Eigenschaften des angegebenen Signalkanals kapselt.

Typ: ChannelInfo Objekt

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#)

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

DescribeStream

Bedienung: Amazon Kinesis Video Streams

Gibt die aktuellsten Informationen über den angegebenen Stream zurück. Sie müssen entweder den StreamName oder den angebenStreamARN.

Anforderungssyntax

```
POST /describeStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

Name des -Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "StreamInfo": {
    "CreationTime": number,
    "DataRetentionInHours": number,
    "DeviceName": "string",
    "KmsKeyId": "string",
    "MediaType": "string",
    "Status": "string",
    "StreamARN": "string",
    "StreamName": "string",
    "Version": "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

StreamInfo

Ein Objekt, das den Stream beschreibt.

Typ: [StreamInfo](#) Objekt

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetDataEndpoint

Bedienung: Amazon Kinesis Video Streams

Ruft einen Endpunkt für einen angegebenen Stream zum Lesen oder Schreiben ab. Verwenden Sie diesen Endpunkt in Ihrer Anwendung, um aus dem angegebenen Stream zu lesen (mithilfe der `GetMediaForFragmentList` Operationen `GetMedia` oder) oder in ihn zu schreiben (mithilfe der `PutMedia` Operation).

Note

An den zurückgegebenen Endpunkt ist der API-Name nicht angehängt. Der Client muss den API-Namen zum zurückgegebenen Endpunkt hinzufügen.

Geben Sie in der Anfrage den Stream entweder mit `StreamName` oder `StreamARN`.

Anforderungssyntax

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
  "APIName": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

APIName

Der Name der API-Aktion, für die ein Endpunkt abgerufen werden soll.

Typ: Zeichenfolge

Zulässige Werte: PUT_MEDIA | GET_MEDIA | LIST_FRAGMENTS |
GET_MEDIA_FOR_FRAGMENT_LIST | GET_HLS_STREAMING_SESSION_URL |
GET_DASH_STREAMING_SESSION_URL | GET_CLIP | GET_IMAGES

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, für den Sie den Endpunkt abrufen möchten. Sie müssen entweder diesen Parameter oder a StreamName in der Anfrage angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, für den Sie den Endpunkt abrufen möchten. Sie müssen entweder diesen Parameter oder a StreamARN in der Anfrage angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "DataEndpoint": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

DataEndpoint

Der Endpunktwert. Um Daten aus dem Stream zu lesen oder Daten in den Stream zu schreiben, geben Sie diesen Endpunkt in Ihrer Anwendung an.

Typ: Zeichenfolge

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetSignalingChannelEndpoint

Bedienung: Amazon Kinesis Video Streams

Stellt einen Endpunkt für den angegebenen Signalkanal zum Senden und Empfangen von Nachrichten bereit. Diese API verwendet den `SingleMasterChannelEndpointConfiguration` Eingabeparameter, der aus den `Role` Eigenschaften `Protocols` und besteht.

`Protocols` wird verwendet, um den Kommunikationsmechanismus zu bestimmen. Wenn Sie beispielsweise WSS als Protokoll angeben, erzeugt diese API einen sicheren WebSocket-Endpunkt. Wenn Sie HTTPS als Protokoll angeben, generiert diese API einen HTTPS-Endpunkt.

`Role` bestimmt die Messaging-Berechtigungen. Eine MASTER Rolle führt dazu, dass diese API einen Endpunkt generiert, über den ein Client mit allen Zuschauern auf dem Kanal kommunizieren kann. Eine VIEWER Rolle führt dazu, dass diese API einen Endpunkt generiert, über den ein Client nur mit einem kommunizieren kann MASTER.

Anforderungssyntax

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "SingleMasterChannelEndpointConfiguration": {
    "Protocols": [ "string" ],
    "Role": "string"
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Signalisierungskanals, für den Sie einen Endpunkt abrufen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

[SingleMasterChannelEndpointConfiguration](#)

Eine Struktur, die die Endpunktkonfiguration für den SINGLE_MASTER Kanaltyp enthält.

Typ: [SingleMasterChannelEndpointConfiguration](#) Objekt

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ResourceEndpointList": [
    {
      "Protocol": "string",
      "ResourceEndpoint": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[ResourceEndpointList](#)

Eine Liste von Endpunkten für den angegebenen Signalkanal.

Typ: Array von [ResourceEndpointListItem](#)-Objekten

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv` `ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API, um den Status der Ressource zu bestimmen.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ListEdgeAgentConfigurations

Bedienung: Amazon Kinesis Video Streams

Gibt ein Array von Edge-Konfigurationen zurück, die dem angegebenen Edge-Agenten zugeordnet sind.

In der Anfrage müssen Sie den Edge-Agent angeben `HubDeviceArn`.

Note

Diese API ist in der Region AWS Afrika (Kapstadt), `af-south-1`, nicht verfügbar.

Anforderungssyntax

```
POST /listEdgeAgentConfigurations HTTP/1.1
Content-type: application/json

{
  "HubDeviceArn": "string",
  "MaxResults": number,
  "NextToken": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

HubDeviceArn

Das „Internet der Dinge (IoT) -Ding“ Arn of the Edge Agent.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Erforderlich: Ja

MaxResults

Die maximale Anzahl von Edge-Konfigurationen, die in der Antwort zurückgegeben werden sollen. Der Standardwert ist 5.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert 10.

Required: No

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis einer `ListEdgeAgentConfigurations` Operation gekürzt wird, gibt der Aufruf `NextToken` in der Antwort den zurück. Um einen weiteren Stapel von Edge-Konfigurationen zu erhalten, geben Sie dieses Token in Ihrer nächsten Anfrage an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "EdgeConfigs": [
    {
      "CreationTime": number,
      "EdgeConfig": {
        "DeletionConfig": {
          "DeleteAfterUpload": boolean,
          "EdgeRetentionInHours": number,
          "LocalSizeConfig": {
```

```

        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
    }
},
"HubDeviceArn": "string",
"RecorderConfig": {
    "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
    },
    "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
    }
},
"UploaderConfig": {
    "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
    }
},
"FailedStatusDetails": "string",
"LastUpdatedTime": number,
"StreamARN": "string",
"StreamName": "string",
"SyncStatus": "string"
}
],
"NextToken": "string"
}

```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

EdgeConfigs

Eine Beschreibung der Edge-Konfiguration eines einzelnen Streams.

Typ: Array von [ListEdgeAgentConfigurationsEdgeConfig](#)-Objekten

NextToken

Wenn die Antwort gekürzt wird, gibt der Aufruf dieses Element mit einem bestimmten Token zurück. Verwenden Sie dieses Token in Ihrer nächsten Anfrage, um den nächsten Stapel von Edge-Konfigurationen abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ListSignalingChannels

Bedienung: Amazon Kinesis Video Streams

Gibt ein Array von `ChannelInfo` Objekten zurück. Jedes Objekt beschreibt einen Signalkanal. Um nur die Kanäle abzurufen, die eine bestimmte Bedingung erfüllen, können Sie a `channelNameCondition` angeben.

Anforderungssyntax

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
  "ChannelNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelNameCondition

Optional: Gibt nur die Kanäle zurück, die eine bestimmte Bedingung erfüllen.

Typ: ChannelNameCondition Objekt

Required: No

MaxResults

Die maximale Anzahl von Kanälen, die in der Antwort zurückgegeben werden sollen. Die Standardeinstellung ist 500.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 10000.

Required: No

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis einer `ListSignalingChannels` Operation gekürzt wird, gibt der Aufruf das `NextToken` in der Antwort zurück. Um einen weiteren Stapel von Kanälen zu erhalten, geben Sie dieses Token in Ihrer nächsten Anfrage an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelInfoList": [
    {
      "ChannelARN": "string",
      "ChannelName": "string",
      "ChannelStatus": "string",
      "ChannelType": "string",
      "CreationTime": number,
      "SingleMasterConfiguration": {
        "MessageTtlSeconds": number
      },
      "Version": "string"
    }
  ],
  "NextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

ChannelInfoList

Ein Array von ChannelInfo-Objekten.

Typ: Array von [ChannelInfo](#)-Objekten

NextToken

Wenn die Antwort gekürzt wird, gibt der Aufruf dieses Element mit einem Token zurück. Verwenden Sie dieses Token in Ihrer nächsten Anfrage, um den nächsten Stapel von Streams abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: [a-zA-Z0-9+/=]*

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ListStreams

Bedienung: Amazon Kinesis Video Streams

Gibt ein Array von `StreamInfo` Objekten zurück. Jedes Objekt beschreibt einen Stream. Um nur Streams abzurufen, die eine bestimmte Bedingung erfüllen, können Sie `streamNameCondition` angeben.

Anforderungssyntax

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string",
  "StreamNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

MaxResults

Die maximale Anzahl von Streams, die in der Antwort zurückgegeben werden sollen. Die Standardeinstellung ist 10.000.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 10000.

Required: No

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis einer `ListStreams` Operation gekürzt wird, gibt der Aufruf das `NextToken` in der Antwort zurück. Um einen weiteren Stapel von Streams zu erhalten, geben Sie dieses Token in Ihrer nächsten Anfrage an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamNameCondition

Optional: Gibt nur Streams zurück, die eine bestimmte Bedingung erfüllen. Derzeit können Sie nur das Präfix eines Streamnamens als Bedingung angeben.

Typ: [StreamNameCondition](#) Objekt

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "StreamInfoList": [
    {
      "CreationTime": number,
      "DataRetentionInHours": number,
      "DeviceName": "string",
      "KmsKeyId": "string",
      "MediaType": "string",
      "Status": "string",
      "StreamARN": "string",
      "StreamName": "string",
      "Version": "string"
    }
  ]
}
```

```
]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

NextToken

Wenn die Antwort gekürzt ist, gibt der Aufruf dieses Element mit einem Token zurück. Verwenden Sie dieses Token in Ihrer nächsten Anfrage, um den nächsten Stapel von Streams abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: [a-zA-Z0-9+/=]*

StreamInfoList

Ein Array von `StreamInfo`-Objekten.

Typ: Array von [StreamInfo](#)-Objekten

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ListTagsForResource

Bedienung: Amazon Kinesis Video Streams

Gibt eine Liste von Tags zurück, die dem angegebenen Signalkanal zugeordnet sind.

Anforderungssyntax

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
  "NextToken": "string",
  "ResourceARN": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis eines `ListTagsForResource` Aufrufs gekürzt wird, enthält die Antwort ein Token, das Sie in der nächsten Anforderung verwenden können, um den nächsten Stapel von Tags abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

ResourceARN

Der Amazon-Ressourcenname (ARN) des Signalisierungskanals, für den Sie Tags auflisten möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis eines `ListTagsForResource` Aufrufs gekürzt wird, enthält die Antwort ein Token, das Sie in der nächsten Anfrage verwenden können, um den nächsten Satz von Tags abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Tags

Eine Zuordnung von Tag-Schlüsseln und -Werten, die dem angegebenen Signalkanal zugeordnet sind.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 50 Elementen.

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: $^([\backslash p\{L\}\backslash p\{Z\}\backslash p\{N\}_\cdot : / = + \backslash - @]^*)\$$

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertemuster: $[\backslash p\{L\}\backslash p\{Z\}\backslash p\{N\}_\cdot : / = + \backslash - @]^*$

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

ListTagsForStream

Bedienung: Amazon Kinesis Video Streams

Gibt eine Liste von Tags zurück, die dem angegebenen Stream zugeordnet sind.

In der Anfrage müssen Sie entweder den `StreamName` oder den `StreamARN` angeben.

Anforderungssyntax

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis eines `ListTagsForStream` Aufrufs gekürzt wird, enthält die Antwort ein Token, das Sie in der nächsten Anforderung verwenden können, um den nächsten Stapel von Tags abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, für den Sie Tags auflisten möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, für den Sie Tags auflisten möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis eines `ListTags` Aufrufs gekürzt wird, enthält die Antwort ein Token, das Sie in der nächsten Anforderung verwenden können, um den nächsten Satz von Tags abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge von 512.

Pattern: `[a-zA-Z0-9+/=]*`

Tags

Eine Zuordnung von Tag-Schlüsseln und -Werten, die dem angegebenen Stream zugeordnet sind.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 50 Elementen.

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *) $`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertemuster: `[\p{L}\p{Z}\p{N}_ . : / = + \ - @] *`

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidResourceFormatException

Das Format von `StreamARN` ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

StartEdgeConfigurationUpdate

Bedienung: Amazon Kinesis Video Streams

Eine asynchrone API, die die bestehende Edge-Konfiguration eines Streams aktualisiert. Der Kinesis Video Stream synchronisiert die Edge-Konfiguration des Streams mit der Edge Agent IoT Greengrass-Komponente, die auf einem bei Ihnen vor Ort eingerichteten IoT Hub-Gerät ausgeführt wird. Die Dauer der Synchronisierung kann variieren und hängt von der Konnektivität des Hub-Geräts ab. Das SyncStatus wird aktualisiert, sobald die Edge-Konfiguration bestätigt und mit dem Edge-Agenten synchronisiert wird.

Wenn diese API zum ersten Mal aufgerufen wird, wird eine neue Edge-Konfiguration für den Stream erstellt, und der Synchronisierungsstatus wird auf gesetzt. SYNCING Sie müssen warten, bis der Synchronisierungsstatus einen Terminalstatus wie: oder erreicht hat IN_SYNC SYNC_FAILED, bevor Sie diese API erneut verwenden können. Wenn Sie diese API während des Synchronisierungsvorgangs aufrufen, ResourceInUseException wird ausgelöst. Die Verbindung zwischen der Edge-Konfiguration des Streams und dem Edge-Agenten wird für 15 Minuten erneut versucht. Nach 15 Minuten wechselt der Status in den SYNC_FAILED Status.

Um eine Edge-Konfiguration von einem Gerät auf ein anderes zu verschieben, verwenden Sie, [DeleteEdgeConfiguration](#) um die aktuelle Edge-Konfiguration zu löschen. Sie können dann StartEdgeConfigurationUpdate mit einem aktualisierten Hub-Geräte-ARN aufrufen.

Note

Diese API ist in der Region AWS Afrika (Kapstadt), af-south-1, nicht verfügbar.

Anforderungssyntax

```
POST /startEdgeConfigurationUpdate HTTP/1.1
Content-type: application/json
```

```
{
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    }
  }
}
```

```

    }
  },
  "HubDeviceArn": "string",
  "RecorderConfig": {
    "MediaSourceConfig": {
      "MediaUriSecretArn": "string",
      "MediaUriType": "string"
    },
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  }
},
"StreamARN": "string",
"StreamName": "string"
}

```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[EdgeConfig](#)

Die Edge-Konfigurationsdetails, die zum Aufrufen des Aktualisierungsvorgangs erforderlich sind.

Typ: [EdgeConfig](#) Objekt

Erforderlich: Ja

[StreamARN](#)

Der Amazon-Ressourcenname (ARN) des -Stream. Geben Sie entweder den `StreamName` oder den `anStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, dessen Edge-Konfiguration Sie aktualisieren möchten. Geben Sie entweder den StreamName oder den anStreamARN.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      }
    }
  }
}
```

```
    },
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  }
},
"FailedStatusDetails": "string",
"LastUpdatedTime": number,
"StreamARN": "string",
"StreamName": "string",
"SyncStatus": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

CreationTime

Der Zeitstempel, zu dem die Edge-Konfiguration eines Streams zum ersten Mal erstellt wurde.

Typ: Zeitstempel

EdgeConfig

Eine Beschreibung der Edge-Konfiguration des Streams, die für die Synchronisierung mit der Edge Agent IoT Greengrass-Komponente verwendet wird. Die Edge Agent-Komponente wird auf einem IoT Hub-Gerät ausgeführt, das bei Ihnen vor Ort eingerichtet ist.

Typ: [EdgeConfig](#) Objekt

FailedStatusDetails

Eine Beschreibung des generierten Fehlerstatus.

Typ: Zeichenfolge

LastUpdatedTime

Der Zeitstempel, zu dem die Edge-Konfiguration eines Streams zuletzt aktualisiert wurde.

Typ: Zeitstempel

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

Der Name des Streams, von dem aus die Edge-Konfiguration aktualisiert wurde.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

SyncStatus

Der aktuelle Synchronisierungsstatus der Edge-Konfiguration des Streams. Wenn Sie diese API aufrufen, wird der Synchronisierungsstatus auf den SYNCING Status gesetzt. Verwenden Sie die `DescribeEdgeConfiguration` API, um den neuesten Status der Edge-Konfiguration abzurufen.

Typ: Zeichenfolge

Zulässige Werte: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NoDataRetentionException

Die Stream-Datenspeicherung in Stunden ist gleich Null.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

TagResource

Bedienung: Amazon Kinesis Video Streams

Fügt einem Signalkanal ein oder mehrere Tags hinzu. Ein Tag ist ein Schlüssel-Wert-Paar (der Wert ist optional), das Sie definieren und Ressourcen zuweisen AWS können. Wenn Sie ein Tag angeben, das bereits vorhanden ist, wird der Tag-Wert durch den Wert ersetzt, den Sie in der Anfrage angeben. Weitere Informationen finden Sie unter [Verwenden von Cost Allocation Tags](#) im AWS Billing and Cost Managementand Cost Management-Benutzerhandbuch.

Anforderungssyntax

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ResourceARN

Der Amazon-Ressourcenname (ARN) des Signalisierungskanals, zu dem Sie Tags hinzufügen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

Tags

Eine Liste von Tags, die dem angegebenen Signalkanal zugeordnet werden sollen. Jeder Tag ist ein Schlüssel/Wert-Paar.

Typ: Array von [Tag](#)-Objekten

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 50 Elemente.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

TagsPerResourceExceededLimitException

Sie haben das Limit an Tags überschritten, die Sie der Ressource zuordnen können. Ein Kinesis-Videostream kann bis zu 50 Tags unterstützen.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

TagStream

Bedienung: Amazon Kinesis Video Streams

Fügt einem Stream ein oder mehrere Tags hinzu. Ein Tag ist ein Schlüssel-Wert-Paar (der Wert ist optional), das Sie definieren und Ressourcen zuweisen AWS können. Wenn Sie ein Tag angeben, das bereits vorhanden ist, wird der Tag-Wert durch den Wert ersetzt, den Sie in der Anfrage angeben. Weitere Informationen finden Sie unter [Verwenden von Cost Allocation Tags](#) im AWS Billing and Cost Management and Cost Management-Benutzerhandbuch.

Sie müssen entweder den `StreamName` oder den `StreamARN` angeben.

Diese Operation setzt eine Berechtigung für die `KinesisVideo:TagStream`-Aktion voraus.

Ein Kinesis-Videostream kann bis zu 50 Tags unterstützen.

Anforderungssyntax

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[StreamARN](#)

Der Amazon-Ressourcenname (ARN) der Ressource, zu der Sie das Tag oder die Tags hinzufügen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, zu dem Sie das Tag oder die Tags hinzufügen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Tags

Eine Liste von Tags, die dem angegebenen Stream zugeordnet werden sollen. Jedes Tag ist ein Schlüssel-Wert-Paar (der Wert ist optional).

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 50 Elementen.

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *) $`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertemuster: `[\p{L}\p{Z}\p{N}_ . : / = + \ - @] *`

Erforderlich: Ja

Antwortsyntax

HTTP/1.1 200

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidResourceFormatException

Das Format von `StreamARN` ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

TagsPerResourceExceededLimitException

Sie haben das Limit an Tags überschritten, die Sie der Ressource zuordnen können. Ein Kinesis-Videostream kann bis zu 50 Tags unterstützen.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UntagResource

Bedienung: Amazon Kinesis Video Streams

Entfernt ein oder mehrere Tags aus einem Signalkanal. Geben Sie in der Anfrage nur einen oder mehrere Tag-Schlüssel an, geben Sie nicht den Wert an. Wenn Sie einen Tag-Schlüssel angeben, der nicht existiert, wird er ignoriert.

Anforderungssyntax

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "TagKeyList": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ResourceARN

Der Amazon-Ressourcenname (ARN) des Signalkanals, aus dem Sie Tags entfernen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

TagKeyList

Eine Liste der Schlüssel der Tags, die Sie entfernen möchten.

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *)$`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UntagStream

Bedienung: Amazon Kinesis Video Streams

Entfernt ein oder mehrere Tags aus einem Stream. Geben Sie in der Anfrage nur einen oder mehrere Tag-Schlüssel an, geben Sie nicht den Wert an. Wenn Sie einen Tag-Schlüssel angeben, der nicht existiert, wird er ignoriert.

In der Anfrage müssen Sie das `StreamName` oder `angebenStreamARN`.

Anforderungssyntax

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "TagKeyList": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, aus dem Sie Tags entfernen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, aus dem Sie Tags entfernen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

TagKeyList

Eine Liste der Schlüssel der Tags, die Sie entfernen möchten.

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *) $`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidResourceFormatException

Das Format von StreamARN ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateDataRetention

Bedienung: Amazon Kinesis Video Streams

Erhöht oder verringert die Datenaufbewahrungsdauer des Streams um den von Ihnen angegebenen Wert. Um anzugeben, ob Sie die Datenaufbewahrungsdauer verlängern oder verkürzen möchten, geben Sie den `Operation` Parameter im Anfragetext an. In der Anfrage müssen Sie entweder den `StreamName` oder den `StreamARN` angeben.

Diese Operation setzt eine Berechtigung für die `KinesisVideo:UpdateDataRetention`-Aktion voraus.

Eine Änderung der Datenaufbewahrungsdauer wirkt sich wie folgt auf die Daten im Stream aus:

- Wenn die Datenaufbewahrungsfrist verlängert wird, werden bestehende Daten für die neue Aufbewahrungsfrist aufbewahrt. Wenn beispielsweise die Datenaufbewahrungsdauer von einer Stunde auf sieben Stunden verlängert wird, werden alle vorhandenen Daten sieben Stunden lang aufbewahrt.
- Wenn die Datenaufbewahrungsdauer verkürzt wird, werden die vorhandenen Daten für die neue Aufbewahrungsfrist aufbewahrt. Wenn beispielsweise die Datenaufbewahrungsdauer von sieben Stunden auf eine Stunde verkürzt wird, werden alle vorhandenen Daten für eine Stunde aufbewahrt, und alle Daten, die älter als eine Stunde sind, werden sofort gelöscht.

Anforderungssyntax

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DataRetentionChangeInHours": number,
  "Operation": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

CurrentVersion

Die Version des Streams, dessen Aufbewahrungszeitraum Sie ändern möchten. Um die Version abzurufen, rufen Sie entweder die `DescribeStream` oder die `ListStreams` API auf.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

Erforderlich: Ja

DataRetentionChangeInHours

Die Anzahl der Stunden, um die die aktuelle Aufbewahrung angepasst werden soll. Der von Ihnen angegebene Wert wird zum aktuellen Wert addiert oder von diesem subtrahiert, abhängig von `operation`.

Der Mindestwert für die Datenspeicherung ist 0 und der Höchstwert ist 87600 (zehn Jahre).

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1.

Erforderlich: Ja

Operation

Gibt an, ob Sie die Aufbewahrungsdauer verlängern oder verkürzen möchten.

Typ: Zeichenfolge

Zulässige Werte: `INCREASE_DATA_RETENTION` | `DECREASE_DATA_RETENTION`

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, dessen Aufbewahrungszeitraum Sie ändern möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, dessen Aufbewahrungszeitraum Sie ändern möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

VersionMismatchException

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die [DescribeStream](#) API, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)

- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateImageGenerationConfiguration

Bedienung: Amazon Kinesis Video Streams

Aktualisiert die ImageProcessingConfiguration Felder StreamInfo und.

Anforderungssyntax

```
POST /updateImageGenerationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ImageGenerationConfiguration

Die Struktur, die die Informationen enthält, die für die Lieferung von KVS-Images erforderlich sind. Wenn die Struktur Null ist, wird die Konfiguration aus dem Stream gelöscht.

Typ: [ImageGenerationConfiguration](#) Objekt

Required: No

[StreamARN](#)

Der Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, von dem aus Sie die Konfiguration der Image-Generierung aktualisieren möchten. Sie müssen entweder den `StreamName` oder den `StreamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

[StreamName](#)

Der Name des Streams, aus dem die Konfiguration der Image-Generierung aktualisiert werden soll. Sie müssen entweder den `StreamName` oder den `StreamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NoDataRetentionException

Die Stream-Datenspeicherung in Stunden ist gleich Null.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API, um den Status der Ressource zu bestimmen.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateMediaStorageConfiguration

Bedienung: Amazon Kinesis Video Streams

Ordnet einem Stream a `SignalingChannel` zu, um die Medien zu speichern. Es gibt zwei Signalmodi, die Sie angeben können:

- Wenn `StorageStatus` aktiviert, werden die Daten im `StreamARN` bereitgestellten Ordner gespeichert. Damit WebRTC Ingestion funktioniert, muss für den Stream die Datenspeicherung aktiviert sein.
- Wenn deaktiviert `StorageStatus` ist, werden keine Daten gespeichert und der `StreamARN` Parameter wird nicht benötigt.

Important

Wenn diese Option aktiviert `StorageStatus` ist, treten keine direkten Verbindungen peer-to-peer (Master-Viewer) mehr auf. Peers stellen eine direkte Verbindung zur Speichersitzung her. Sie müssen die `JoinStorageSession` API aufrufen, um das Senden eines SDP-Angebots auszulösen und eine Verbindung zwischen einem Peer und der Speichersitzung herzustellen.

Anforderungssyntax

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Kanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

MediaStorageConfiguration

Eine Struktur, die die Konfigurationseigenschaften des Medienspeichers kapselt oder enthält.

Typ: [MediaStorageConfiguration](#) Objekt

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#)

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NoDataRetentionException

Die Stream-Datenspeicherung in Stunden ist gleich Null.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateNotificationConfiguration

Bedienung: Amazon Kinesis Video Streams

Aktualisiert die Benachrichtigungsinformationen für einen Stream.

Anforderungssyntax

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[NotificationConfiguration](#)

Die Struktur, die die für Benachrichtigungen erforderlichen Informationen enthält. Wenn die Struktur Null ist, wird die Konfiguration aus dem Stream gelöscht.

Typ: [NotificationConfiguration](#) Objekt

Required: No

[StreamARN](#)

Der Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, von dem aus Sie die Benachrichtigungskonfiguration aktualisieren möchten. Sie müssen entweder den `StreamName` oder den `StreamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, von dem aus die Benachrichtigungskonfiguration aktualisiert werden soll. Sie müssen entweder den `StreamName` oder den `angebenStreamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NoDataRetentionException

Die Stream-Datenspeicherung in Stunden ist gleich Null.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status Aktiv `ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API, um den Status der Ressource zu bestimmen.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für V3 JavaScript](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateSignalingChannel

Bedienung: Amazon Kinesis Video Streams

Aktualisiert den vorhandenen Signalkanal. Dies ist ein asynchroner Vorgang, dessen Abschluss einige Zeit in Anspruch nimmt.

Wenn der `MessageTtlSeconds` Wert aktualisiert (entweder erhöht oder reduziert) wird, gilt er nur für neue Nachrichten, die über diesen Kanal gesendet wurden, nachdem er aktualisiert wurde. Bestehende Nachrichten sind weiterhin gemäß dem vorherigen `MessageTtlSeconds` Wert abgelaufen.

Anforderungssyntax

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Signalkanals, den Sie aktualisieren möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

CurrentVersion

Die aktuelle Version des Signalkanals, den Sie aktualisieren möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: [a-zA-Z0-9]+

Erforderlich: Ja

SingleMasterConfiguration

Die Struktur, die die Konfiguration für den SINGLE_MASTER Typ des Signalkanals enthält, den Sie aktualisieren möchten.

Typ: SingleMasterConfiguration Objekt

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status `Aktiv ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

VersionMismatchException

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die [DescribeStream](#) API, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)

- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

UpdateStream

Bedienung: Amazon Kinesis Video Streams

Aktualisiert Stream-Metadaten wie den Gerätenamen und den Medientyp.

Sie müssen den Stream-Namen oder den Amazon-Ressourcennamen (ARN) des Streams angeben.

Um sicherzustellen, dass Sie über die neueste Version des Streams verfügen, bevor Sie ihn aktualisieren, können Sie die Stream-Version angeben. Kinesis Video Streams weist jedem Stream eine Version zu. Wenn Sie einen Stream aktualisieren, weist Kinesis Video Streams eine neue Versionsnummer zu. Verwenden Sie die API, um die neueste Stream-Version zu erhalten.

DescribeStream

UpdateStream ist ein asynchroner Vorgang, dessen Abschluss einige Zeit in Anspruch nimmt.

Anforderungssyntax

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DeviceName": "string",
  "MediaType": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

CurrentVersion

Die Version des Streams, dessen Metadaten Sie aktualisieren möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: [a-zA-Z0-9]+

Erforderlich: Ja

DeviceName

Der Name des Geräts, das in den Stream schreibt.

Note

In der aktuellen Implementierung verwendet Kinesis Video Streams diesen Namen nicht.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

MediaType

Der Medientyp des Streams. Wird verwendet `MediaType`, um den Inhaltstyp anzugeben, den der Stream für die Nutzer des Streams enthält. Weitere Informationen zu Medientypen finden Sie unter [Medientypen](#). Wenn Sie das `MediaType` angeben möchten, finden Sie weitere Informationen unter [Benennungsvoraussetzungen](#).

Um Video auf der Konsole abzuspielen, müssen Sie den richtigen Videotyp angeben. Wenn es sich bei dem Video im Stream beispielsweise um H.264 handelt, geben Sie `video/h264` als `anMediaType`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: [\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*

Required: No

StreamARN

Der ARN des Streams, dessen Metadaten Sie aktualisieren möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

Der Name des Streams, dessen Metadaten Sie aktualisieren möchten.

Der Streamname ist eine Kennung für den Stream und muss für jedes Konto und jede Region eindeutig sein.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceInUseException

Wenn der Eingang `StreamARN` oder `ChannelARN` Eingang bereits einer anderen Kinesis Video Stream-Ressource zugeordnet `CLOUD_STORAGE_MODE` ist oder wenn sich der angegebene Eingang `StreamARN` oder Eingang nicht im Status Aktiv `ChannelARN` befindet, versuchen Sie es mit einer der folgenden Methoden:

1. Die `DescribeMediaStorageConfiguration` API, um zu ermitteln, welchem Stream der angegebene Kanal zugeordnet ist.
2. Die `DescribeMappedResourceConfiguration` API zur Bestimmung des Kanals, dem der angegebene Stream zugeordnet ist.
3. Die `DescribeStream` oder `DescribeSignalingChannel` API zur Bestimmung des Status der Ressource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams kann den von Ihnen angegebenen Stream nicht finden.

HTTP Status Code: 404

VersionMismatchException

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die [DescribeStream](#) API, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Medien für Amazon Kinesis Video Streams

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Media unterstützt:

- [GetMedia](#)
- [PutMedia](#)

GetMedia

Service: Amazon Kinesis Video Streams Media

Verwenden Sie diese API, um Medieninhalte aus einem Kinesis-Videostream abzurufen. In der Anfrage identifizieren Sie den Streamnamen oder den Stream-Amazon-Ressourcennamen (ARN) und den Startblock. Kinesis Video Streams gibt dann einen Stream von Chunks zurück, der nach Fragmentnummer sortiert ist.

Note

Sie müssen zuerst die `GetDataEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Senden Sie dann die `GetMedia` Anfragen mit dem Parameter [--endpoint-url an diesen Endpunkt](#).

Wenn Sie Mediendaten (Fragmente) in einen Stream einfügen, speichert Kinesis Video Streams jedes eingehende Fragment und die zugehörigen Metadaten in einem sogenannten „Chunk“. Weitere Informationen finden Sie unter [PutMedia](#). Die `GetMedia` API gibt einen Stream dieser Chunks zurück, der mit dem Chunk beginnt, den Sie in der Anfrage angeben.

Bei der Verwendung der API gelten die folgenden Beschränkungen: `GetMedia`

- Ein Client kann `GetMedia` bis zu fünf Mal pro Sekunde pro Stream aufrufen.
- Kinesis Video Streams sendet während einer Sitzung Mediendaten mit einer Geschwindigkeit von bis zu 25 Megabyte pro Sekunde (oder 200 Megabit pro Sekunde). `GetMedia`

Note

Der Statuscode der `GetMedia` HTTP-Antwort wird sofort zurückgegeben, aber das Lesen der Nutzdaten der HTTP-Antwort wird nach 3 Sekunden beendet, wenn keine aufgenommenen Fragmente für die Wiedergabe verfügbar sind.

Note

Wenn nach dem Aufrufen einer Kinesis Video Streams Streams-Medien-API ein Fehler ausgelöst wird, enthält dieser zusätzlich zum HTTP-Statuscode und dem Antworttext die folgenden Informationen:

- `x-amz-ErrorTypeHTTP-Header` — enthält zusätzlich zu dem, was der HTTP-Statuscode bietet, einen spezifischeren Fehlertyp.
- `x-amz-RequestIdHTTP-Header` — Wenn Sie ein Problem melden möchten AWS, kann das Support-Team das Problem anhand der Anforderungs-ID besser diagnostizieren.

Sowohl der HTTP-Statuscode als auch der `ErrorType` Header können verwendet werden, um programmatische Entscheidungen darüber zu treffen, ob und unter welchen Bedingungen Fehler wiederholt werden können. Außerdem können Informationen darüber bereitgestellt werden, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erneut erfolgreich zu versuchen.

[Weitere Informationen finden Sie im Abschnitt Fehler am Ende dieses Themas sowie unter Häufige Fehler.](#)

Anforderungssyntax

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
  "StartSelector": {
    "AfterFragmentNumber": "string",
    "ContinuationToken": "string",
    "StartSelectorType": "string",
    "StartTimestamp": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

StartSelector

Identifiziert den Startblock, der aus dem angegebenen Stream abgerufen werden soll.

Typ: [StartSelector](#) Objekt

Erforderlich: Ja

StreamARN

Der ARN des Streams, von dem Sie den Medieninhalt abrufen möchten. Wenn Sie den nicht angeben `streamARN`, müssen Sie den `streamName` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Kinesis-Videostreams, von dem Sie den Medieninhalt abrufen möchten. Wenn Sie den nicht angeben `streamName`, müssen Sie den `streamARN` angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-Type: ContentType
```

```
Payload
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die Antwort gibt die folgenden HTTP-Header zurück.

ContentType

Der Inhaltstyp des angeforderten Mediums.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[a-zA-Z0-9_\.\-]+$`

Die Antwort gibt folgendes als HTTP-Hauptteil zurück.

Payload

Die Payload, die Kinesis Video Streams zurückgibt, ist eine Sequenz von Chunks aus dem angegebenen Stream. Weitere Informationen zu den Chunks finden Sie unter [PutMedia](#). Die Chunks, die Kinesis Video Streams im `GetMedia` Aufruf zurückgibt, enthalten auch die folgenden zusätzlichen Matroska-Tags (MKV):

- `AWS_KINESISVIDEO_CONTINUATION_TOKEN` (UTF-8-Zeichenfolge) — Falls Ihr `GetMedia` Aufruf beendet wird, können Sie dieses Fortsetzungstoken in Ihrer nächsten Anfrage verwenden, um den nächsten Abschnitt abzurufen, in dem die letzte Anfrage beendet wurde.
- `AWS_KINESISVIDEO_MILLIS_BEHIND_NOW` (UTF-8-Zeichenfolge) — Client-Anwendungen können diesen Tag-Wert verwenden, um zu ermitteln, wie weit der in der Antwort zurückgegebene Chunk vom letzten Chunk im Stream zurückliegt.
- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` — Im Chunk zurückgegebene Fragmentnummer.
- `AWS_KINESISVIDEO_SERVER_TIMESTAMP` — Serverzeitstempel des Fragments.
- `AWS_KINESISVIDEO_PRODUCER_TIMESTAMP` — Generator-Zeitstempel des Fragments.

Die folgenden Tags sind vorhanden, wenn ein Fehler auftritt:

- `AWS_KINESISVIDEO_ERROR_CODE` — Zeichenkettenbeschreibung eines Fehlers, der zum Abbruch geführt hat. `GetMedia`
- `AWS_KINESISVIDEO_ERROR_ID`: Ganzzahlcode des Fehlers.

Die Fehlercodes lauten wie folgt:

- 3002 — Fehler beim Schreiben in den Stream
- 4000 — Das angeforderte Fragment wurde nicht gefunden

- 4500 — Zugriff für den KMS-Schlüssel des Streams verweigert
- 4501 — Der KMS-Schlüssel des Streams ist deaktiviert
- 4502 — Validierungsfehler beim KMS-Schlüssel des Streams
- 4503 — Der im Stream angegebene KMS-Schlüssel ist nicht verfügbar
- 4504 — Ungültige Verwendung des im Stream angegebenen KMS-Schlüssels
- 4505 — Ungültiger Status des im Stream angegebenen KMS-Schlüssels
- 4506 — Der im Stream angegebene KMS-Schlüssel konnte nicht gefunden werden
- 5000 — Interner Fehler

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

ConnectionLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der zulässigen Client-Verbindungen überschritten haben.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidEndpointException

Der Aufrufer hat einen falschen Endpunkt verwendet, um Daten in einen Stream zu schreiben. Bei Empfang einer solchen Ausnahme muss der Benutzer `GetDataEndpoint` mit `APIName` set to `PUT_MEDIA` aufrufen und den Endpunkt aus der Antwort verwenden, um den nächsten `PutMedia` Aufruf aufzurufen.

HTTP Status Code: 400

NotAuthorizedException

Der Aufrufer ist nicht autorisiert, eine Operation mit dem angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

Statuscode: 404, Der Stream mit dem angegebenen Namen existiert nicht.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

PutMedia

Service: Amazon Kinesis Video Streams Media

Verwenden Sie diese API, um Mediendaten an einen Kinesis-Videostream zu senden.

Note

Sie müssen zuerst die `GetDataEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Senden Sie dann die `PutMedia` Anfragen mit dem Parameter [--endpoint-url an diesen Endpunkt](#).

In der Anfrage verwenden Sie die HTTP-Header, um Parameterinformationen bereitzustellen, z. B. Streamname, Zeitstempel und ob der Zeitstempelwert absolut oder relativ zu dem Zeitpunkt ist, zu dem der Producer mit der Aufnahme begonnen hat. Sie verwenden den Anfragetext, um die Mediendaten zu senden. Kinesis Video Streams unterstützt nur das Matroska-Containerformat (MKV) für das Senden von Mediendaten über diese API.

Sie haben die folgenden Optionen für das Senden von Daten mit dieser API:

- **Mediendaten in Echtzeit senden:** Eine Sicherheitskamera kann beispielsweise Bilder in Echtzeit senden, während sie sie generiert. Dieser Ansatz minimiert die Latenz zwischen der Videoaufnahme und den über das Kabel gesendeten Daten. Dies wird als kontinuierlicher Produzent bezeichnet. In diesem Fall kann eine Verbraucheranwendung den Stream in Echtzeit oder bei Bedarf lesen.
- **Mediendaten offline (stapelweise) senden:** Eine Körperkamera kann beispielsweise stundenlang Videos aufnehmen und auf dem Gerät speichern. Später, wenn Sie die Kamera an den Docking-Anschluss anschließen, kann die Kamera eine `PutMedia` Sitzung starten, um Daten an einen Kinesis-Videostream zu senden. In diesem Szenario ist Latenz kein Problem.

Beachten Sie bei der Verwendung dieser API die folgenden Überlegungen:

- Sie müssen entweder `streamName` oder `streamARN` angeben, aber nicht beides.
- Um die Medien auf der Konsole oder über HLS abspielen zu können, sollte Track 1 jedes Fragments H.264-codiertes Video enthalten, die `CodeCid` in den Fragment-Metadaten sollte „V_MPEG/ISO/AVC“ lauten und die Fragment-Metadaten sollten private H.264-Codec-Daten im AVCC-Format enthalten. Optional sollte Track 2 jedes Fragments AAC-codiertes Audio enthalten,

die CodeCid in den Fragment-Metadaten sollte „A_AAC“ lauten und die Fragment-Metadaten sollten private AAC-Codec-Daten enthalten.

- Möglicherweise finden Sie es einfacher, eine einzelne PutMedia Sitzung mit langer Laufzeit zu verwenden und eine große Anzahl von Mediendatenfragmenten in der Payload zu senden. Für jedes empfangene Fragment sendet Kinesis Video Streams eine oder mehrere Bestätigungen. Mögliche Netzwerkprobleme können dazu führen, dass Sie nicht alle diese Bestätigungen erhalten, sobald sie generiert werden.
- Sie können mehrere aufeinanderfolgende PutMedia Sitzungen mit jeweils weniger Fragmenten wählen, um sicherzustellen, dass Sie alle Bestätigungen vom Dienst in Echtzeit erhalten.

Note

Wenn Sie in mehreren gleichzeitigen PutMedia Sitzungen Daten an denselben Stream senden, werden die Medienfragmente im Stream verschachtelt. Sie sollten sicherstellen, dass dies in Ihrem Anwendungsszenario in Ordnung ist.

Bei der Verwendung der PutMedia API gelten die folgenden Beschränkungen:

- Ein Client kann PutMedia bis zu fünf Mal pro Sekunde pro Stream aufrufen.
- Ein Client kann bis zu fünf Fragmente pro Sekunde pro Stream senden.
- Kinesis Video Streams liest Mediendaten mit einer Geschwindigkeit von bis zu 12,5 MB/Sekunde oder 100 Mbit/s während einer Sitzung. PutMedia

Beachten Sie die folgenden Einschränkungen. In diesen Fällen sendet Kinesis Video Streams die Fehlerbestätigung in der Antwort.

- Fragmente mit Zeitcodes, die den maximal zulässigen Grenzwert überschreiten, und die mehr als 50 MB an Daten enthalten, sind nicht zulässig.
- Fragmente, die mehr als drei Spuren enthalten, sind nicht zulässig. Jeder Frame in jedem Fragment muss dieselbe Spurnummer haben wie eine der im Fragment-Header definierten Spuren. Darüber hinaus muss jedes Fragment mindestens einen Frame für jede im Fragment-Header definierte Spur enthalten.
- Jedes Fragment muss mindestens einen Frame für jede in den Fragment-Metadaten definierte Spur enthalten.

- Der früheste Frame-Zeitstempel in einem Fragment muss nach dem letzten Frame-Zeitstempel im vorherigen Fragment liegen.
- Ein MKV-Stream, der mehr als ein MKV-Segment enthält oder unzulässige MKV-Elemente (`wietrack*`) enthält, führt ebenfalls zur Fehlerbestätigung.

Kinesis Video Streams speichert jedes eingehende Fragment und die zugehörigen Metadaten in einem sogenannten „Chunk“. Die Fragment-Metadaten umfassen Folgendes:

- Die MKV-Header, die zu Beginn der Anfrage bereitgestellt wurden `PutMedia`
- Die folgenden Kinesis Video Streams-spezifischen Metadaten für das Fragment:
 - `server_timestamp`- Zeitstempel, zu dem Kinesis Video Streams mit dem Empfang des Fragments begonnen hat.
 - `producer_timestamp`- Zeitstempel, wann der Produzent mit der Aufnahme des Fragments begonnen hat. Kinesis Video Streams verwendet drei in der Anfrage empfangene Informationen, um diesen Wert zu berechnen.
 - Der Timecode-Wert des Fragments, der zusammen mit dem Fragment im Hauptteil der Anfrage empfangen wurde.
 - Zwei Anforderungsheader: `producerStartTimeStamp` (als der Produzent mit der Aufnahme begonnen hat) und `fragmentTimeCodeType` (ob der Fragment-Timecode in der Payload absolut oder relativ ist).

Kinesis Video Streams berechnet dann den `producer_timestamp` für das Fragment wie folgt:

Wenn es relativ `fragmentTimeCodeType` ist, dann

$$\text{producer_timestamp} = \text{producerStartTimeStamp} + \text{Fragment-Timecode}$$

Wenn `fragmentTimeCodeType` es absolut ist, dann

$$\text{producer_timestamp} = \text{Fragment-Timecode (in Millisekunden umgewandelt)}$$

- Von Kinesis Video Streams zugewiesene eindeutige Fragmentnummer.

Note

Wenn Sie die `GetMedia` Anfrage stellen, gibt Kinesis Video Streams einen Stream dieser Chunks zurück. Der Client kann die Metadaten nach Bedarf verarbeiten.

Note

Dieser Vorgang ist nur für das AWS SDK for Java verfügbar. Er wird in AWS SDKs für andere Sprachen nicht unterstützt.

Note

Kinesis Video Streams analysiert und validiert die privaten Codec-Daten während der Aufnahme und Archivierung über die API nicht. PutMedia KVS extrahiert und validiert die erforderlichen Informationen aus den privaten Codec-Daten für das Verpacken von MPEG-TS- und MP4-Fragmenten, wenn der Stream über die HLS-APIs abgerufen wird.

Note

Wenn nach dem Aufrufen einer Kinesis Video Streams Streams-Medien-API ein Fehler ausgelöst wird, enthält dieser zusätzlich zum HTTP-Statuscode und dem Antworttext die folgenden Informationen:

- `x-amz-ErrorType`HTTP-Header — enthält zusätzlich zu dem, was der HTTP-Statuscode bietet, einen spezifischeren Fehlertyp.
- `x-amz-RequestId`HTTP-Header — Wenn Sie ein Problem melden möchten AWS, kann das Support-Team das Problem anhand der Anforderungs-ID besser diagnostizieren.

Sowohl der HTTP-Statuscode als auch der `ErrorType` Header können verwendet werden, um programmatische Entscheidungen darüber zu treffen, ob und unter welchen Bedingungen Fehler wiederholt werden können. Außerdem können Informationen darüber bereitgestellt werden, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erneut erfolgreich zu versuchen.

Weitere Informationen finden Sie im Abschnitt Fehler am Ende dieses Themas sowie unter [Häufige Fehler](#).

Anforderungssyntax

```
POST /putMedia HTTP/1.1
x-amzn-stream-name: StreamName
x-amzn-stream-arn: StreamARN
x-amzn-fragment-timecode-type: FragmentTimecodeType
x-amzn-producer-start-timestamp: ProducerStartTimestamp
```

Payload

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

[FragmentTimecodeType](#)

Sie übergeben diesen Wert als `x-amzn-fragment-timecode-type` HTTP-Header.

Gibt an, ob die Timecodes in den Fragmenten (Payload, HTTP-Anforderungstext) absolut oder relativ zu sind. `producerStartTimestamp` Kinesis Video Streams verwendet diese Informationen, um die `producer_timestamp` für das in der Anfrage empfangene Fragment zu berechnen, wie in der API-Übersicht beschrieben.

Zulässige Werte: ABSOLUTE | RELATIVE

Erforderlich: Ja

[ProducerStartTimestamp](#)

Sie übergeben diesen Wert als `x-amzn-producer-start-timestamp` HTTP-Header.

Dies ist der Producer-Zeitstempel, zu dem der Producer mit der Aufnahme der Medien begonnen hat (nicht der Zeitstempel der spezifischen Fragmente in der Anfrage).

[StreamARN](#)

Sie übergeben diesen Wert als `x-amzn-stream-arn` HTTP-Header.

Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, in den Sie den Medieninhalt schreiben möchten. Wenn Sie den nicht angeben `streamARN`, müssen Sie den `streamName` angeben.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

Sie übergeben diesen Wert als `x-amzn-stream-name` HTTP-Header.

Name des Kinesis-Videostreams, in den Sie den Medieninhalt schreiben möchten. Wenn Sie den nicht angeben `streamName`, müssen Sie den `streamARN` angeben.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Anforderungstext

Die Anfrage akzeptiert die folgenden Binärdaten.

Payload

Der Medieninhalt, der in den Kinesis-Videostream geschrieben werden soll. In der aktuellen Implementierung unterstützt Kinesis Video Streams nur das Matroska (MKV) -Containerformat mit einem einzigen MKV-Segment. Ein Segment kann einen oder mehrere Cluster enthalten.

Note

Jeder MKV-Cluster ist einem Kinesis-Videostream-Fragment zugeordnet. Die von Ihnen gewählte Clusterdauer wird zur Fragmentdauer.

Antwortsyntax

```
HTTP/1.1 200
```

Payload

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.


Die Antwort gibt folgendes als HTTP-Hauptteil zurück.

Payload

Nachdem Kinesis Video Streams erfolgreich eine PutMedia Anfrage empfangen hat, validiert der Dienst die Anforderungsheader. Der Dienst beginnt dann mit dem Lesen der Payload und sendet zunächst eine HTTP 200-Antwort.

Der Dienst gibt dann einen Stream zurück, der eine Reihe von JSON-Objekten (AcknowledgementObjekten) enthält, die durch Zeilenumbrüche getrennt sind. Die Bestätigungen werden auf derselben Verbindung empfangen, über die die Mediendaten gesendet werden. Für eine Anfrage kann es viele Bestätigungen geben. PutMedia Jedes Acknowledgement besteht aus den folgenden Schlüssel-Wert-Paaren:

- **AckEventType**- Ereignistyp, für den die Bestätigung steht.
 - **Pufferung**: Kinesis Video Streams hat begonnen, das Fragment zu empfangen. Kinesis Video Streams sendet die erste Buffering-Bestätigung, wenn das erste Byte von Fragmentdaten empfangen wird.
 - **Empfangen**: Kinesis Video Streams hat das gesamte Fragment empfangen. Wenn Sie den Stream nicht so konfiguriert haben, dass die Daten dauerhaft gespeichert werden, kann der Producer die Pufferung des Fragments beenden, sobald er diese Bestätigung erhält.
 - **Persistent**: Kinesis Video Streams hat das Fragment beibehalten (z. B. in Amazon S3). Sie erhalten diese Bestätigung, wenn Sie den Stream so konfiguriert haben, dass die Daten dauerhaft gespeichert werden. Nachdem Sie diese Bestätigung erhalten haben, kann der Producer die Pufferung des Fragments beenden.
 - **Fehler**: Bei Kinesis Video Streams ist bei der Verarbeitung des Fragments ein Fehler aufgetreten. Sie können den Fehlercode überprüfen und die nächste Vorgehensweise festlegen.
 - **Inaktiv**: Die PutMedia Sitzung ist im Gange. Kinesis Video Streams empfängt derzeit jedoch keine Daten. Kinesis Video Streams sendet diese Bestätigung in regelmäßigen Abständen für bis zu 30 Sekunden nach den letzten empfangenen Daten. Wenn innerhalb der 30 Sekunden keine Daten empfangen werden, schließt Kinesis Video Streams die Anfrage.

 Note


Anhand dieser Bestätigung kann ein Produzent feststellen, ob die PutMedia Verbindung aktiv ist, auch wenn keine Daten gesendet werden.

- `FragmentTimecode`— Fragment-Timecode, für den die Bestätigung gesendet wird.

Das Element kann fehlen, wenn es im Leerlauf ist **AckEventType**.

- `FragmentNumber`- Von Kinesis Video Streams generierte Fragmentnummer, für die die Bestätigung gesendet wird.
- `ErrorId` und `ErrorCode` — Falls `jaError`, enthält dieses Feld den `AckEventType` entsprechenden Fehlercode. Im Folgenden finden Sie eine Liste der Fehler-IDs und der entsprechenden Fehlercodes und Fehlermeldungen:
 - 4000 - `STREAM_READ_ERROR` - Fehler beim Lesen des Datenstroms.
 - 4001 - `MAX_FRAGMENT_SIZE_REACHED` — Die Fragmentgröße liegt über dem zulässigen Höchstwert von 50 MB.
 - 4002 - `MAX_FRAGMENT_DURATION_REACHED` — Die Fragmentdauer liegt über dem maximal zulässigen Grenzwert.
 - 4003 - `MAX_CONNECTION_DURATION_REACHED` — Die Verbindungsdauer ist größer als der maximal zulässige Schwellenwert.
 - 4004 - `FRAGMENT_TIMECODE_LESSER_THAN_PREVIOUS` — Der Fragment-Timecode ist kleiner als der Timecode des vorherigen Timecodes (innerhalb eines Anrufs können Sie Fragmente nicht in der falschen Reihenfolge senden). PutMedia
 - 4005 - `MORE_THAN_ALLOWED_TRACKS_FOUND` - In MKV wurde mehr als ein Titel gefunden. (veraltet)
 - 4006 - `INVALID_MKV_DATA` - Der Eingabestream konnte nicht als gültiges MKV-Format analysiert werden.
 - 4007 - `INVALID_PRODUCER_TIMESTAMP` — Ungültiger Producer-Zeitstempel.
 - 4008 - `STREAM_NOT_ACTIVE` - Stream existiert nicht mehr (gelöscht).
 - 4009 - `FRAGMENT_METADATA_LIMIT_REACHED` — Das Limit für Fragment-Metadaten wurde erreicht. [Weitere Informationen finden Sie im Abschnitt Grenzwerte im Entwicklerhandbuch.](#)
 - 4010 - `TRACK_NUMBER_MISMATCH` - Die Titelnummer in einem MKV-Frame stimmte nicht mit den Titeln im MKV-Header überein.

- 4011 - FRAMES_MISSING_FOR_TRACK - Das Fragment enthielt keine Frames für mindestens einen der Tracks im MKV-Header.
- 4012 - INVALID_FRAGMENT_METADATA - Der Name der Fragment-Metadaten darf nicht mit der Zeichenfolge beginnen. AWS_
- 4500 - KMS_KEY_ACCESS_DENIED — Der Zugriff auf den angegebenen KMS-Schlüssel des Streams wurde verweigert.
- 4501 - KMS_KEY_DISABLED — Der für den Stream angegebene KMS-Schlüssel ist deaktiviert.
- 4502 - KMS_KEY_VALIDATION_ERROR — Der angegebene KMS-Schlüssel des Streams konnte nicht überprüft werden.
- 4503 - KMS_KEY_UNAVAILABLE — Der angegebene KMS-Schlüssel für den Stream ist nicht verfügbar.
- 4504 - KMS_KEY_INVALID_USAGE — Ungültige Verwendung des angegebenen KMS-Schlüssels für den Stream.
- 4505 - KMS_KEY_INVALID_STATE — Der angegebene KMS-Schlüssel des Streams befindet sich in einem ungültigen Zustand.
- 4506 - KMS_KEY_NOT_FOUND — Der angegebene KMS-Schlüssel des Streams wurde nicht gefunden.
- 5000 - INTERNAL_ERROR - Interner Dienstfehler.
- 5001 - ARCHIVAL_ERROR - Kinesis Video Streams konnte keine Fragmente im Datenspeicher speichern.

 Note

Der Producer sollte beim Senden der Nutzdaten für eine lang andauernde PutMedia Anfrage die Antwort zur Bestätigung lesen. Ein Producer kann aufgrund der Pufferung auf einem zwischengeschalteten Proxyserver mehrere Bestätigungen gleichzeitig erhalten. Ein Produzent, der zeitnahe Bestätigungen erhalten möchte, kann in jeder Anfrage weniger Fragmente senden. PutMedia

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

ConnectionLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der zulässigen Client-Verbindungen überschritten haben.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidEndpointException

Der Aufrufer hat einen falschen Endpunkt verwendet, um Daten in einen Stream zu schreiben. Bei Empfang einer solchen Ausnahme muss der Benutzer `GetDataEndpoint` mit `APIName set to` aufrufen `PUT_MEDIA` und den Endpunkt aus der Antwort verwenden, um den nächsten `PutMedia` Aufruf aufzurufen.

HTTP Status Code: 400

NotAuthorizedException

Der Aufrufer ist nicht autorisiert, eine Operation mit dem angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

Statuscode: 404, Der Stream mit dem angegebenen Namen existiert nicht.

HTTP Status Code: 404

Beispiele

Format der Bestätigung

Das Format der Bestätigung lautet wie folgt:

```
{
  Acknowledgement : {
    "EventType": enum
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
    "ErrorId" : String
  }
}
```

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK for Ruby V3](#)

Amazon Kinesis Video Streams

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Archived Media unterstützt:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

GetClip

Service: Amazon Kinesis Video Streams Archived Media

Lädt eine MP4-Datei (clip) mit den archivierten On-Demand-Medien aus dem angegebenen Videostream über den angegebenen Zeitraum herunter.

Sowohl die - StreamName als auch die StreamARN-Parameter sind optional, aber Sie müssen entweder den - StreamName oder den -StreamARN angeben, wenn Sie diesen API-Vorgang aufrufen.

Note

Sie müssen zuerst die GetDataEndpoint-API aufrufen, um einen Endpunkt abzurufen. Senden Sie dann die GetClip Anfragen an diesen Endpunkt mit dem [Parameter --endpoint-url](#) .

Ein Amazon Kinesis Video Stream hat die folgenden Anforderungen für die Bereitstellung von Daten über MP4:

- [Anforderungen an die Videowiedergabespur](#) .
- Der Datenerhaltzeitraum muss größer als 0 sein.
- Die Videospur jedes Fragments muss private Codec-Daten in Advanced Video Coding (AVC) für H.264 und HEVC für H.265 enthalten. Weitere Informationen finden Sie unter [MPEG-4-Spezifikation ISO/IEC 14496-15](#). Informationen zur Adaptierung von Stream-Daten für ein bestimmtes Format finden Sie unter [NAL Adaptation Flags](#).
- Die Audiospur (falls vorhanden) jedes Fragments muss private Codec-Daten im AAC-Format ([AAC-Spezifikation ISO/IEC 13818-7](#)) oder dem [MS Wave-Format](#) enthalten.

Sie können die Menge der ausgehenden Daten überwachen, indem Sie die GetClip.OutgoingBytes Amazon- CloudWatch Metrik überwachen. Informationen zur Verwendung von CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter [Überwachung von Kinesis Video Streams](#). Preisinformationen finden Sie unter [Amazon Kinesis Video Streams – Preise](#) und [AWS Preise](#). Für ausgehende AWS Daten fallen Gebühren an.

Anforderungssyntax

```
POST /getClip HTTP/1.1
```

```
Content-type: application/json

{
  "ClipFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ClipFragmentSelector

Der Zeitraum des angeforderten Textes und die Quelle der Zeitstempel.

Typ: [ClipFragmentSelector](#) Objekt

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, für den der Medienclip abgerufen werden soll.

Sie müssen entweder den StreamName oder den StreamARN angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, für den der Medienclip abgerufen werden soll.

Sie müssen entweder den StreamName oder den StreamARN angeben.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die Antwort gibt die folgenden HTTP-Header zurück.

ContentType

Der Inhaltstyp der Medien im angeforderten -CLI.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[a-zA-Z0-9_\.\-]+$`

Die Antwort gibt folgendes als HTTP-Hauptteil zurück.

Payload

Herkömmliche MP4-Datei, die den Medienclip aus dem angegebenen Videostream enthält. Die Ausgabe enthält die ersten 100 MB oder die ersten 200 Fragmente des angegebenen Startzeitstempels. Weitere Informationen finden Sie unter [Limits für Kinesis Video Streams](#).

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anforderung gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Weitere Informationen zu Limits finden Sie unter [Limits für Kinesis Video Streams](#).

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Einschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einer der Spuren des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

InvalidMediaFrameException

Ein oder mehrere Frames im angeforderten -CLI konnten nicht basierend auf dem angegebenen Codec analysiert werden.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einer der Spuren des Videostreams wurden keine privaten Codec-Daten gefunden.

HTTP Status Code: 400

NoDataRetentionException

GetImages wurde für einen Stream angefordert, der keine Daten speichert (d. h. einen DataRetentionInHours von 0 hat).

HTTP Status Code: 400

NotAuthorizedException

Statuscode: 403, Der Aufrufer ist nicht autorisiert, eine Operation für den angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages löst diesen Fehler aus, wenn Kinesis Video Streams den angegebenen Stream nicht finden kann.

GetHLSStreamingSessionURL und GetDASHStreamingSessionURL geben diesen Fehler aus, wenn eine Sitzung mit einem PlaybackMode von ON_DEMAND oder für einen Stream ohne Fragmente innerhalb des angeforderten Zeitraums angefordert LIVE_REPLAY wird oder wenn eine Sitzung mit einem PlaybackMode von für einen Stream ohne Fragmente innerhalb der letzten 30 Sekunden angefordert LIVE wird.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

Der Typ der Medien (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte nicht anhand der Codec-IDs der Spuren im ersten Fragment für eine Wiedergabesitzung bestimmt werden. Die Codec-ID für Spur 1 sollte sein V_MPEG/ISO/AVC und optional sollte die Codec-ID für Spur 2 sein A_AAC.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)

- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetDASHStreamingSessionURL

Service: Amazon Kinesis Video Streams Archived Media

Ruft eine URL für MPEG Dynamic Adaptive Streaming über HTTP (DASH) für den Stream ab. Anschließend können Sie die URL in einem Media Player öffnen, um den Stream-Inhalt anzuzeigen.

Sowohl die `StreamName` `StreamARN` Parameter als auch sind optional, aber Sie müssen entweder die `StreamName` oder die angeben, `StreamARN` wenn Sie diesen API-Vorgang aufrufen.

Ein Amazon Kinesis Video Stream hat die folgenden Anforderungen für die Bereitstellung von Daten über MPEG-DASH:

- [Anforderungen an die Videowiedergabespur](#) .
- Der Datenerhaltzeitraum muss größer als 0 sein.
- Die Videospur jedes Fragments muss private Codec-Daten in Advanced Video Coding (AVC) für H.264 und HEVC für H.265 enthalten. Weitere Informationen finden Sie unter [MPEG-4-Spezifikation ISO/IEC 14496-15](#). Informationen zur Adaptierung von Stream-Daten für ein bestimmtes Format finden Sie unter [NAL Adaptation Flags](#).
- Die Audiospur (falls vorhanden) jedes Fragments muss private Codec-Daten im AAC-Format ([AAC-Spezifikation ISO/IEC 13818-7](#)) oder dem [MS Wave-Format](#) enthalten.

Das folgende Verfahren zeigt, wie MPEG-DASH mit Kinesis Video Streams verwendet wird:

1. Rufen Sie die `GetDataEndpoint` -API auf, um einen Endpunkt abzurufen. Senden Sie dann die `GetDASHStreamingSessionURL` Anfragen an diesen Endpunkt mit dem [Parameter `--endpoint-url`](#) .
2. Rufen Sie die MPEG-DASH-URL mit `abGetDASHStreamingSessionURL`. Kinesis Video Streams erstellt eine MPEG-DASH-Streaming-Sitzung, die für den Zugriff auf Inhalte in einem Stream mithilfe des MPEG-DASH-Protokolls verwendet wird. `GetDASHStreamingSessionURL` gibt eine authentifizierte URL (die ein verschlüsseltes Sitzungstoken enthält) für das MPEG-DASH-Manifest der Sitzung zurück (die Stammressource, die für das Streaming mit MPEG-DASH erforderlich ist).

Note

Teilen oder speichern Sie dieses Token nicht, wenn eine nicht autorisierte Entität darauf zugreifen kann. Das Token bietet Zugriff auf den Inhalt des Streams. Schützen Sie das

Token mit den gleichen Maßnahmen, die Sie mit Ihren - AWS Anmeldeinformationen verwenden.

Die Medien, die über das Manifest zur Verfügung gestellt werden, bestehen nur aus dem angeforderten Stream, dem Zeitbereich und dem Format. Es werden keine anderen Mediendaten (z. B. Frames außerhalb des angeforderten Fensters oder alternative Bitraten) zur Verfügung gestellt.

3. Geben Sie die URL (mit dem verschlüsselten Sitzungstoken) für das MPEG-DASH-Manifest an einen Media Player an, der das MPEG-DASH-Protokoll unterstützt. Kinesis Video Streams stellt das Initialisierungsfragment und die Medienfragmente über die Manifest-URL zur Verfügung. Das Initialisierungsfragment enthält die privaten Codec-Daten für den Stream sowie andere Daten, die zum Einrichten des Video- oder Audiodecoders und Renderers benötigt werden. Die Medienfragmente enthalten codierte Videoframes oder codierte Audiobeispiele.
4. Der Media Player empfängt die authentifizierte URL und fordert normal Stream-Metadaten und Mediendaten an. Wenn der Media Player Daten anfordert, ruft er die folgenden Aktionen auf:
 - **GetDASHManifest** : Ruft ein MPEG DASH-Manifest ab, das die Metadaten für die Medien enthält, die Sie wiedergeben möchten.
 - **GetMP4InitFragment**: Ruft das MP4-Initialisierungsfragment ab. Der Media Player lädt normalerweise das Initialisierungsfragment, bevor er Medienfragmente lädt. Dieses Fragment enthält die MP4-Atome „fyp“ und „moov“ sowie die untergeordneten Atome, die zur Initialisierung des Media Player-Decoders erforderlich sind.

Das Initialisierungsfragment entspricht keinem Fragment in einem Kinesis-Videostream. Es enthält nur die privaten Codec-Daten für den Stream und den jeweiligen Track, die der Media Player zum Dekodieren der Medienframes benötigt.


- **GetMP4MediaFragment**: Ruft MP4-Medienfragmente ab. Diese Fragmente enthalten die MP4-Atome „moof“ und „mdat“ und ihre untergeordneten Atome, die die Medienframes des codierten Fragments und ihre Zeitstempel enthalten.

Note

Nachdem das erste Medienfragment in einer Streaming-Sitzung verfügbar gemacht wurde, führen alle Fragmente, die nicht dieselben privaten Codec-Daten enthalten, dazu, dass ein Fehler zurückgegeben wird, wenn diese verschiedenen Medienfragmente geladen werden. Daher sollten sich die privaten Codec-Daten nicht zwischen

Fragmenten in einer Sitzung ändern. Dies bedeutet auch, dass die Sitzung fehlschlägt, wenn sich die Fragmente in einem Stream von nur Video zu Audio und Video ändern.


Daten, die mit dieser Aktion abgerufen wurden, sind abrechenbar. Weitere Informationen finden Sie unter [Preise](#).

 Note

Einschränkungen für MPEG-DASH-Sitzungen finden Sie unter [Limits für Kinesis Video Streams](#).

Sie können die Datenmenge überwachen, die der Media Player verbraucht, indem Sie die `GetMP4MediaFragment.OutgoingBytes` Amazon-CloudWatch Metrik überwachen. Informationen zur Verwendung von CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter [Überwachung von Kinesis Video Streams](#). Preisinformationen finden Sie unter [Amazon Kinesis Video Streams – Preise](#) und [AWS Preise](#). Es fallen Gebühren sowohl für HLS-Sitzungen als auch für ausgehende AWS Daten an.

Weitere Informationen zu HLS finden Sie unter [HTTP-Live-Streaming](#) auf der [Apple-Developer-Website](#).

 Important

Wenn nach dem Aufruf einer Kinesis Video Streams-API für archivierte Medien zusätzlich zum HTTP-Statuscode und dem Antworttext ein Fehler ausgelöst wird, enthält er die folgenden Informationen:

- `x-amz-ErrorType` HTTP-Header – enthält zusätzlich zu dem, was der HTTP-Statuscode bietet, einen spezifischeren Fehlertyp.
- `x-amz-RequestId` HTTP-Header – wenn Sie ein Problem an AWS das Support-Team melden möchten, kann das Problem besser diagnostiziert werden, wenn die Anforderungs-ID angegeben ist.

Sowohl der HTTP-Statuscode als auch der `-ErrorType` Header können verwendet werden, um programmgesteuerte Entscheidungen darüber zu treffen, ob Fehler wiederholbar sind und

unter welchen Bedingungen, und Informationen darüber bereitzustellen, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erfolgreich erneut zu versuchen.

Weitere Informationen finden Sie im Abschnitt Fehler unten in diesem Thema sowie unter [Häufige Fehler](#).

Anforderungssyntax

```
POST /getDASHStreamingSessionURL HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "DASHFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimestamp": number,
      "StartTimestamp": number
    }
  },
  "DisplayFragmentNumber": "string",
  "DisplayFragmentTimestamp": "string",
  "Expires": number,
  "MaxManifestFragmentResults": number,
  "PlaybackMode": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

[DASHFragmentSelector](#)

Der Zeitraum des angeforderten Fragments und die Quelle der Zeitstempel.

Dieser Parameter ist erforderlich, wenn `ON_DEMAND` oder `PlaybackMode` ist `LIVE_REPLAY`. Dieser Parameter ist optional, wenn `PlaybackMode` ist `LIVE`. Wenn `PlaybackMode` ist `LIVE`, `FragmentSelectorType` kann festgelegt werden, aber der `TimestampRange` sollte nicht festgelegt werden. Wenn `ON_DEMAND` oder `PlaybackMode` ist `LIVE_REPLAY`, `TimestampRange` müssen `FragmentSelectorType` sowohl als auch festgelegt werden.

Typ: [DASHFragmentSelector](#) Objekt

Erforderlich: Nein

[DisplayFragmentNumber](#)

Fragmente werden in der Manifestdatei anhand ihrer Sequenznummer in der Sitzung identifiziert. Wenn auf `DisplayFragmentNumber` ist `ALWAYS`, wird jedem S-Element in der Manifestdatei die Kinesis-Video-Streams-Fragmentnummer mit dem Attributnamen „kvs:fn“ hinzugefügt. Diese Fragmentnummern können für die Protokollierung oder für die Verwendung mit anderen APIs (z. B. `GetMedia` und `GetMediaForFragmentList`) verwendet werden. Ein benutzerdefinierter MPEG-DASH-Medienplayer ist erforderlich, um dieses benutzerdefinierte Attribut zu nutzen.

Der Standardwert ist `NEVER`.

Typ: Zeichenfolge

Zulässige Werte: `ALWAYS` | `NEVER`

Erforderlich: Nein

[DisplayFragmentTimestamp](#)

Gemäß der MPEG-DASH-Spezifikation kann die Uhrzeit der Hintergrunduhr von Fragmenten in der Manifestdatei mithilfe von Attributen im Manifest selbst abgeleitet werden. In der Regel behandeln MPEG-DASH-kompatible Media Player jedoch Lücken in der Medienzeitleiste nicht richtig. Kinesis Video Streams passt die Medienzeitleiste in der Manifestdatei an, um die Wiedergabe von Medien mit Unterbrechungen zu ermöglichen. Daher kann die aus der Manifestdatei abgeleitete Uhrzeit ungenau sein. Wenn auf `DisplayFragmentTimestamp` ist `ALWAYS`, wird jedem S-Element in der Manifestdatei der genaue Fragment-Zeitstempel mit dem Attributnamen „kvs:ts“ hinzugefügt. Ein benutzerdefinierter MPEG-DASH-Medienplayer ist erforderlich, um dieses benutzerdefinierte Attribut zu nutzen.

Der Standardwert ist `NEVER`. Wenn [DASHFragmentSelector](#) ist `SERVER_TIMESTAMP`, sind die Zeitstempel die Server-Startzeitstempel. Wenn gleich [DASHFragmentSelector](#) ist `PRODUCER_TIMESTAMP`, sind die Zeitstempel die Startzeitstempel des Produzenten.

Typ: Zeichenfolge

Zulässige Werte: ALWAYS | NEVER

Erforderlich: Nein

Expires

Die Zeit in Sekunden, bis die angeforderte Sitzung abläuft. Dieser Wert kann zwischen 300 (5 Minuten) und 43200 (12 Stunden) liegen.

Wenn eine Sitzung abläuft, `GetMP4MediaFragment` können für diese Sitzung keine neuen Aufrufe an `GetDashManifest` `GetMP4InitFragment`, oder getätigt werden.

Der Standardwert ist 300 (5 Minuten).

Typ: Ganzzahl

Gültiger Bereich: Mindestwert von 300. Maximalwert 43200.

Erforderlich: Nein

MaxManifestFragmentResults

Die maximale Anzahl von Fragmenten, die im MPEG-DASH-Manifest zurückgegeben werden.

Wenn der `PlaybackMode` ist `LIVE`, werden die neuesten Fragmente bis zu diesem Wert zurückgegeben. Wenn der `PlaybackMode` ist `ON_DEMAND`, werden die ältesten Fragmente bis zu dieser maximalen Anzahl zurückgegeben.

Wenn in einem Live-MPEG-DASH-Manifest eine höhere Anzahl von Fragmenten verfügbar ist, puffern Videoplayer häufig Inhalte, bevor sie mit der Wiedergabe beginnen. Eine Erhöhung der Puffergröße erhöht die Wiedergabelatenz, verringert jedoch die Wahrscheinlichkeit, dass während der Wiedergabe eine erneute Pufferung stattfindet. Wir empfehlen, dass ein Live-MPEG-DASH-Manifest mindestens 3 Fragmente und maximal 10 Fragmente enthält.

Der Standardwert ist 5 Fragmente, wenn `LIVE` oder `PlaybackMode` ist `LIVE_REPLAY`, und 1 000, wenn `PlaybackMode` ist `ON_DEMAND`.

Der Maximalwert von 1 000 Fragmenten entspricht mehr als 16 Minuten Video in Streams mit Fragmenten von 1 Sekunde und mehr als 2 1/2 Stunden Video in Streams mit Fragmenten von 10 Sekunden.

Type: Long

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 5000.

Erforderlich: Nein

PlaybackMode

Gibt an, ob Live-, Live-Wiedergabe- oder archivierte On-Demand-Daten abgerufen werden sollen.

Zu den drei Sitzungstypen gehören die folgenden Funktionen:

- **LIVE** : Für Sitzungen dieses Typs wird das MPEG-DASH-Manifest kontinuierlich mit den neuesten Fragmenten aktualisiert, sobald sie verfügbar sind. Wir empfehlen, dass der Media Player ein neues Manifest in einem Intervall von einer Sekunde abrufen. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel eine „Live“-Benachrichtigung an, ohne Scrubber-Steuererelemente für die Auswahl der Position im Wiedergabefenster, die angezeigt werden soll.

Note

Im -LIVEModus sind die neuesten verfügbaren Fragmente in einem MPEG-DASH-Manifest enthalten, auch wenn eine Lücke zwischen Fragmenten besteht (d. h. wenn ein Fragment fehlt). Eine solche Lücke kann dazu führen, dass ein Media Player angehalten wird oder zu einem Sprung der Wiedergabe führt. In diesem Modus werden Fragmente nicht zum MPEG-DASH-Manifest hinzugefügt, wenn sie älter als das neueste Fragment in der Wiedergabeliste sind. Wenn das fehlende Fragment verfügbar wird, nachdem ein nachfolgendes Fragment zum Manifest hinzugefügt wurde, wird das ältere Fragment nicht hinzugefügt und die Lücke wird nicht gefüllt.

- **LIVE_REPLAY** : Bei Sitzungen dieses Typs wird das MPEG-DASH-Manifest ähnlich aktualisiert wie im -LIVEModus, mit der Ausnahme, dass es mit dem Einschließen von Fragmenten aus einer bestimmten Startzeit beginnt. Anstatt beim Aufnehmen Fragmente hinzuzufügen, werden Fragmente hinzugefügt, wenn die Dauer des nächsten Fragments abgelaufen ist. Wenn die Fragmente in der Sitzung beispielsweise zwei Sekunden lang sind, wird dem Manifest alle zwei Sekunden ein neues Fragment hinzugefügt. Dieser Modus ist nützlich, um die Wiedergabe von zu starten, wenn ein Ereignis erkannt wird, und Live-Streaming-Medien fortzusetzen, die zum Zeitpunkt der Sitzungserstellung noch nicht aufgenommen wurden. Dieser Modus ist auch nützlich, um zuvor archivierte Medien zu streamen, ohne durch das Limit von 1 000 Fragmenten im ON_DEMAND Modus eingeschränkt zu werden.

- **ON_DEMAND** : Für Sitzungen dieses Typs enthält das MPEG-DASH-Manifest alle Fragmente für die Sitzung bis zu der in angegebenen `ZahlMaxManifestFragmentResults`. Das Manifest darf für jede Sitzung nur einmal abgerufen werden. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel ein Scrubber-Steuerelement zur Auswahl der Position im Wiedergabefenster an, die angezeigt werden soll.

In allen Wiedergabemodi, wenn `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und wenn es mehrere Fragmente mit demselben Startzeitstempel gibt, ist das Fragment mit der größeren Fragmentnummer (d. h. dem neueren Fragment) im MPEG-DASH-Manifest enthalten. Die anderen Fragmente sind nicht enthalten. Fragmente mit unterschiedlichen Zeitstempeln, aber überlappenden Dauern sind weiterhin im MPEG-DASH-Manifest enthalten. Dies kann zu unerwartetem Verhalten im Media Player führen.

Der Standardwert ist `LIVE`.

Typ: Zeichenfolge

Zulässige Werte: `LIVE` | `LIVE_REPLAY` | `ON_DEMAND`

Erforderlich: Nein

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, für den die MPEG-DASH-Manifest-URL abgerufen werden soll.

Sie müssen entweder die `StreamName` oder die `angebenStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, für den die MPEG-DASH-Manifest-URL abgerufen werden soll.

Sie müssen entweder die `StreamName` oder die `angebenStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "DASHStreamingSessionURL": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

DASHStreamingSessionURL

Die URL (mit dem Sitzungstoken), die ein Media Player zum Abrufen des MPEG-DASH-Manifests verwenden kann.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anforderung gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Weitere Informationen zu Limits finden Sie unter [Limits für Kinesis Video Streams](#).

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Einschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einer der Spuren des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einer der Spuren des Videostreams wurden keine privaten Codec-Daten gefunden.

HTTP Status Code: 400

NoDataRetentionException

GetImages wurde für einen Stream angefordert, der keine Daten speichert (d. h. einen `DataRetentionInHours` von 0 hat).

HTTP Status Code: 400

NotAuthorizedException

Statuscode: 403, Der Aufrufer ist nicht autorisiert, eine Operation für den angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages gibt diesen Fehler aus, wenn Kinesis Video Streams den angegebenen Stream nicht finden kann.

GetHLSStreamingSessionURL und GetDASHStreamingSessionURL geben diesen Fehler aus, wenn eine Sitzung mit einem `PlaybackMode` von `ON_DEMAND` oder für einen Stream ohne Fragmente innerhalb des angeforderten Zeitraums angefordert `LIVE_REPLAY` wird oder wenn eine Sitzung mit einem `PlaybackMode` von `LIVE` für einen Stream ohne Fragmente innerhalb der letzten 30 Sekunden angefordert `LIVE` wird.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

Der Typ der Medien (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte nicht anhand der Codec-IDs der Spuren im ersten Fragment für eine Wiedergabesitzung bestimmt werden. Die Codec-ID für Spur 1 sollte sein V_MPEG/ISO/AVC und optional sollte die Codec-ID für Spur 2 seinA_AAC.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetHLSStreamingSessionURL

Service: Amazon Kinesis Video Streams Archived Media

Ruft eine HTTP Live Streaming (HLS)-URL für den Stream ab. Anschließend können Sie die URL in einem Browser oder Media Player öffnen, um den Stream-Inhalt anzuzeigen.

Sowohl die `StreamName` `StreamARN` Parameter als auch sind optional, aber Sie müssen entweder die `StreamName` oder die angeben, `StreamARN` wenn Sie diesen API-Vorgang aufrufen.


Ein Amazon Kinesis Video Stream hat die folgenden Anforderungen für die Bereitstellung von Daten über HLS:

- [Anforderungen an die Videowiedergabespur](#) .
- Der Datenerhaltzeitraum muss größer als 0 sein.
- Die Videospur jedes Fragments muss private Codec-Daten im Advanced Video Coding (AVC) für das H.264-Format oder HEVC für das H.265-Format ([MPEG-4-Spezifikation ISO/IEC 14496-15](#)) enthalten. Informationen zur Adaptierung von Stream-Daten für ein bestimmtes Format finden Sie unter [NAL Adaptation Flags](#).
- Die Audiospur (falls vorhanden) jedes Fragments muss private Codec-Daten im AAC-Format enthalten ([AAC-Spezifikation ISO/IEC 13818-7](#)).

HLS-Sitzungen von Kinesis Video Streams enthalten Fragmente in der fragmentierten MPEG-4-Form (auch als fMP4 oder CMAF bezeichnet) oder der MPEG-2-Form (auch als TS-Blöcke bezeichnet, die die HLS-Spezifikation ebenfalls unterstützt). Weitere Informationen zu HLS-Fragmenttypen finden Sie in der [HLS-Spezifikation](#) .

Das folgende Verfahren zeigt, wie Sie HLS mit Kinesis Video Streams verwenden:

1. Rufen Sie die `GetDataEndpoint` -API auf, um einen Endpunkt abzurufen. Senden Sie dann die `GetHLSStreamingSessionURL` Anfragen an diesen Endpunkt mit dem [Parameter `--endpoint-url`](#) .
2. Rufen Sie die HLS-URL mit `abGetHLSStreamingSessionURL`. Kinesis Video Streams erstellt eine HLS-Streaming-Sitzung, die für den Zugriff auf Inhalte in einem Stream mithilfe des HLS-Protokolls verwendet wird. `GetHLSStreamingSessionURL` gibt eine authentifizierte URL (die ein verschlüsseltes Sitzungstoken enthält) für die HLS-Master-Wiedergabeliste der Sitzung (die Stammressource, die für das Streaming mit HLS erforderlich ist) zurück.

 Note

Teilen oder speichern Sie dieses Token nicht, wenn eine nicht autorisierte Entität darauf zugreifen könnte. Das Token bietet Zugriff auf den Inhalt des Streams. Schützen Sie das Token mit den gleichen Maßnahmen, die Sie mit Ihren AWS -Anmeldeinformationen verwenden würden.

Die Medien, die über die Wiedergabeliste zur Verfügung gestellt werden, bestehen nur aus dem angeforderten Stream, dem Zeitbereich und dem Format. Es werden keine anderen Mediendaten (z. B. Frames außerhalb des angeforderten Fensters oder alternative Bitraten) zur Verfügung gestellt.

3. Geben Sie die URL (die das verschlüsselte Sitzungstoken enthält) für die HLS-Master-Wiedergabeliste an einen Media Player an, der das HLS-Protokoll unterstützt. Kinesis Video Streams stellt die HLS-Medienwiedergabeliste, das Initialisierungsfragment und die Medienfragmente über die URL der Hauptwiedergabeliste zur Verfügung. Das Initialisierungsfragment enthält die privaten Codec-Daten für den Stream sowie andere Daten, die zum Einrichten des Video- oder Audiodecoders und Renderers benötigt werden. Die Medienfragmente enthalten H.264-kodierte Videoframes oder AAC-kodierte Audiobeispiele.
4. Der Media Player empfängt die authentifizierte URL und fordert normal Stream-Metadaten und Mediendaten an. Wenn der Media Player Daten anfordert, ruft er die folgenden Aktionen auf:
 - `GetHLSMasterPlaylist` : Ruft eine HLS-Master-Wiedergabeliste ab, die eine URL für die `GetHLSMediaPlaylist` Aktion für jeden Track und zusätzliche Metadaten für den Media Player enthält, einschließlich der geschätzten Bitrate und Auflösung.
 - `GetHLSMediaPlaylist` : Ruft eine HLS-Medienwiedergabeliste ab, die eine URL für den Zugriff auf das MP4-Initialisierungsfragment mit der `GetMP4InitFragment` Aktion und URLs für den Zugriff auf die MP4-Medienfragmente mit den `GetMP4MediaFragment` Aktionen enthält. Die HLS-Medienwiedergabeliste enthält auch Metadaten über den Stream, den der LIVE Player zum Abspielen benötigt, z. B. ob oder `PlaybackMode` ist `ON_DEMAND`. Die HLS-Medienwiedergabeliste ist in der Regel statisch für Sitzungen mit einem `PlaybackType` von `ON_DEMAND`. Die HLS-Medienwiedergabeliste wird kontinuierlich mit neuen Fragmenten für Sitzungen mit einem `PlaybackType` von `aktualisiertLIVE`. Es gibt eine eigene HLS-Medienwiedergabeliste für die Videospur und die Audiospur (falls zutreffend), die MP4-Medien-URLs für die spezifische Spur enthält.

- **GetMP4InitFragment:** Ruft das MP4-Initialisierungsfragment ab. Der Media Player lädt normalerweise das Initialisierungsfragment, bevor er Medienfragmente lädt. Dieses Fragment enthält die MP4-Atome „ftyp“ und „moov“ sowie die untergeordneten Atome, die zur Initialisierung des Media Player-Decoders erforderlich sind.

Das Initialisierungsfragment entspricht keinem Fragment in einem Kinesis-Videostream. Es enthält nur die privaten Codec-Daten für den Stream und den jeweiligen Track, die der Media Player zum Dekodieren der Medienframes benötigt.


- **GetMP4MediaFragment:** Ruft MP4-Medienfragmente ab. Diese Fragmente enthalten die MP4-Atome „moof“ und „mdat“ und ihre untergeordneten Atome, die die Medienframes des codierten Fragments und ihre Zeitstempel enthalten.

Note

Für die HLS-Streaming-Sitzung werden Änderungen an privaten Daten (CPD) im Rahmen von In-Track-Codec unterstützt. Nachdem das erste Medienfragment in einer Streaming-Sitzung verfügbar gemacht wurde, können Fragmente CPD-Änderungen für jeden Track enthalten. Daher können die Fragmente in einer Sitzung eine andere Auflösung, Bitrate oder andere Informationen im CPD haben, ohne die Wiedergabe zu unterbrechen. Jede Änderung am Track-Nummer- oder Track-Codec-Format kann jedoch einen Fehler zurückgeben, wenn diese verschiedenen Medienfragmente geladen werden. Beispielsweise schlägt das Streaming fehl, wenn sich die Fragmente im Stream von nur Video zu Audio und Video ändern oder wenn eine AAC-Audiospur zu einer ALAW-Audiospur geändert wird. Für jede Streaming-Sitzung sind nur 500 CPD-Änderungen zulässig.

Daten, die mit dieser Aktion abgerufen wurden, sind abrechenbar. Weitere Informationen finden Sie unter [-Preise](#).

- **GetTSFragment :** Ruft MPEG-TS-Fragmente ab, die sowohl Initialisierungs- als auch Mediendaten für alle Spuren im Stream enthalten.

 Note


Wenn der `ContainerFormat` ist `MPEG_TS`, wird diese API anstelle von `GetMP4InitFragment` und verwendet, `GetMP4MediaFragment` um Stream-Medien abzurufen.

Daten, die mit dieser Aktion abgerufen wurden, sind abrechenbar. Weitere Informationen finden Sie unter [Preise für Kinesis Video Streams](#).

Eine Streaming-Sitzungs-URL darf nicht zwischen Playern geteilt werden. Der Service drosselt eine Sitzung möglicherweise, wenn mehrere Media Player sie gemeinsam nutzen. Verbindungslimits finden Sie unter [Kinesis-Video-Streams-Limits](#).

Sie können die Datenmenge überwachen, die der Media Player verbraucht, indem Sie die `GetMP4MediaFragment.OutgoingBytes` Amazon- CloudWatch Metrik überwachen. Informationen zur Verwendung von CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter [Überwachung von Kinesis Video Streams](#). Preisinformationen finden Sie unter [Amazon Kinesis Video Streams – Preise](#) und [AWS – Preise](#). Es fallen Gebühren sowohl für HLS-Sitzungen als auch für ausgehende AWS Daten an.

Weitere Informationen zu HLS finden Sie unter [HTTP-Live-Streaming](#) auf der [Apple-Developer-Website](#).

 Important

Wenn nach dem Aufruf einer Kinesis Video Streams-API für archivierte Medien zusätzlich zum HTTP-Statuscode und dem Antworttext ein Fehler ausgelöst wird, enthält er die folgenden Informationen:

- `x-amz-ErrorType` HTTP-Header – enthält zusätzlich zu dem, was der HTTP-Statuscode bietet, einen spezifischeren Fehlertyp.
- `x-amz-RequestId` HTTP-Header – Wenn Sie ein Problem an melden möchten AWS, kann das Support-Team das Problem besser diagnostizieren, wenn es die Anforderungs-ID erhält.

Sowohl der HTTP-Statuscode als auch der - ErrorType Header können verwendet werden, um programmgesteuerte Entscheidungen darüber zu treffen, ob Fehler wiederholbar sind und unter welchen Bedingungen, und Informationen darüber bereitzustellen, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erfolgreich erneut zu versuchen.

Weitere Informationen finden Sie im Abschnitt Fehler unten in diesem Thema sowie unter [Häufige Fehler](#).

Anforderungssyntax

```
POST /getHLSStreamingSessionURL HTTP/1.1
Content-type: application/json

{
  "ContainerFormat": "string",
  "DiscontinuityMode": "string",
  "DisplayFragmentTimestamp": "string",
  "Expires": number,
  "HLSFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimestamp": number,
      "StartTimestamp": number
    }
  },
  "MaxMediaPlaylistFragmentResults": number,
  "PlaybackMode": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ContainerFormat

Gibt an, welches Format zum Verpacken der Medien verwendet werden soll. Durch die Angabe des `FRAGMENTED_MP4` Containerformats werden die Medien in MP4-Fragmente (fMP4 oder CMAF) verpackt. Dies ist die empfohlene Paketierung, da nur ein minimaler Paketierungsaufwand entsteht. Die andere Containerformatoption ist `MPEG_TS`. HLS hat MPEG-TS-Blöcke seit ihrer Veröffentlichung unterstützt und ist manchmal die einzige unterstützte Paketierung auf älteren HLS-Playern. MPEG TS hat in der Regel einen Verpackungsaufwand von 5–25 Prozent. Das bedeutet, dass MPEG TS in der Regel 5–25 Prozent mehr Bandbreite und Kosten erfordert als fMP4.

Der Standardwert ist `FRAGMENTED_MP4`.

Typ: Zeichenfolge

Zulässige Werte: `FRAGMENTED_MP4` | `MPEG_TS`

Erforderlich: Nein

DiscontinuityMode

Gibt an, wann Flags, die Unterbrechungen zwischen Fragmenten markieren, zu den Medienwiedergabelisten hinzugefügt werden.

Media Player erstellen in der Regel eine Zeitleiste mit abzuspielenden Medieninhalten, basierend auf den Zeitstempeln jedes Fragments. Das bedeutet, dass die Media [HLSFragmentSelector](#)-Player-Zeitleiste bei Überschneidungen oder Lücken zwischen Fragmenten (wie bei üblich `SERVER_TIMESTAMP`) auch an einigen Stellen kleine Lücken zwischen Fragmenten aufweist und Frames an anderen Stellen überschreibt. Lücken in der Timeline des Media Players können dazu führen, dass die Wiedergabe ins Stocken gerät, und Überschneidungen können dazu führen, dass die Wiedergabe jitterig ist. Wenn es Unterbrechungs-Flags zwischen Fragmenten gibt, wird erwartet, dass der Media Player die Zeitleiste zurücksetzt, was dazu führt, dass das nächste Fragment unmittelbar nach dem vorherigen Fragment abgespielt wird.

Die folgenden Modi werden unterstützt:

- **ALWAYS:** Zwischen jedem Fragment in der HLS-Medienwiedergabeliste wird eine Unterbrechungsmarkierung platziert. Es wird empfohlen, den Wert zu verwenden `ALWAYS`, wenn die Fragment-Zeitstempel nicht korrekt sind.
- **NEVER:** Es werden keine Abbruchmarkierungen platziert. Es wird empfohlen, einen Wert von zu verwenden, `NEVER` um sicherzustellen, dass die Timeline des Media Players den Zeitstempeln des Produzenten am genauesten entspricht.

- **ON_DISCONTINUITY**: Eine Unterbrechungsmarkierung wird zwischen Fragmenten platziert, die eine Lücke oder Überlappung von mehr als 50 Millisekunden aufweisen. Für die meisten Wiedergabeszenarien wird empfohlen, den Wert zu verwenden, **ON_DISCONTINUITY** damit die Media-Player-Zeitleiste nur zurückgesetzt wird, wenn ein erhebliches Problem mit der Medienzeitleiste auftritt (z. B. ein fehlendes Fragment).

Der Standardwert ist `ON_DISCONTINUITY` wenn auf festgelegt [HLSFragmentSelector](#) ist `SERVER_TIMESTAMP`, und `NEVER` wenn es auf festgelegt ist `PRODUCER_TIMESTAMP`.

Typ: Zeichenfolge

Zulässige Werte: `ALWAYS` | `NEVER` | `ON_DISCONTINUITY`

Erforderlich: Nein

[DisplayFragmentTimestamp](#)

Gibt an, wann die Startzeitstempel des Fragments in die HLS-Medienwiedergabeliste aufgenommen werden sollen. In der Regel melden Media Player die Playhead-Position als eine Zeit relativ zum Anfang des ersten Fragments in der Wiedergabesitzung. Wenn die Startzeitstempel jedoch in der HLS-Medienwiedergabeliste enthalten sind, melden einige Media Player den aktuellen Playhead basierend auf den Fragmentzeitstempeln möglicherweise als absolute Zeit. Dies kann nützlich sein, um ein Wiedergabeerlebnis zu erstellen, das den Zuschauern die Uhrzeit der Medien anzeigt.

Der Standardwert ist `NEVER`. Wenn [HLSFragmentSelector](#) ist `SERVER_TIMESTAMP`, sind die Zeitstempel die Server-Startzeitstempel. Wenn gleich [HLSFragmentSelector](#) ist `PRODUCER_TIMESTAMP`, sind die Zeitstempel die Startzeitstempel des Produzenten.

Typ: Zeichenfolge

Zulässige Werte: `ALWAYS` | `NEVER`

Erforderlich: Nein

[Expires](#)

Die Zeit in Sekunden, bis die angeforderte Sitzung abläuft. Dieser Wert kann zwischen 300 (5 Minuten) und 43200 (12 Stunden) liegen.

Wenn eine Sitzung abläuft, `GetTSFragment` können für diese Sitzung keine neuen Aufrufe an `GetHLSMasterPlaylist`, `GetMP4InitFragment`, `GetHLSMediaPlaylist` oder `GetMP4MediaFragment`, oder getätigt werden.

Der Standardwert ist 300 (5 Minuten).

Typ: Ganzzahl

Gültiger Bereich: Mindestwert von 300. Maximalwert 43200.

Erforderlich: Nein

[HLSFragmentSelector](#)

Der Zeitraum des angeforderten Fragments und die Quelle der Zeitstempel.

Dieser Parameter ist erforderlich, wenn `ON_DEMAND` oder `PlaybackMode` ist `LIVE_REPLAY`. Dieser Parameter ist optional, wenn `PlaybackMode` ist `LIVE`. Wenn `PlaybackMode` ist `LIVE`, `FragmentSelectorType` kann der festgelegt werden, aber der `TimestampRange` sollte nicht festgelegt werden. Wenn `ON_DEMAND` oder `PlaybackMode` ist `LIVE_REPLAY`, `TimestampRange` müssen `FragmentSelectorType` sowohl als auch festgelegt werden.

Typ: [HLSFragmentSelector](#) Objekt

Erforderlich: Nein

[MaxMediaPlaylistFragmentResults](#)

Die maximale Anzahl von Fragmenten, die in den HLS-Medienwiedergabelisten zurückgegeben werden.

Wenn der `PlaybackMode` ist `LIVE`, werden die neuesten Fragmente bis zu diesem Wert zurückgegeben. Wenn der `PlaybackMode` ist `ON_DEMAND`, werden die ältesten Fragmente bis zu dieser maximalen Anzahl zurückgegeben.

Wenn in einer Live-HLS-Medienwiedergabeliste eine höhere Anzahl von Fragmenten verfügbar ist, puffern Videoplayer häufig Inhalte, bevor sie mit der Wiedergabe beginnen. Eine Erhöhung der Puffergröße erhöht die Wiedergabelatenz, verringert jedoch die Wahrscheinlichkeit, dass während der Wiedergabe eine erneute Pufferung stattfindet. Wir empfehlen, dass eine Live-HLS-Medienwiedergabeliste mindestens 3 Fragmente und maximal 10 Fragmente hat.

Der Standardwert ist 5 Fragmente, wenn `LIVE` oder `PlaybackMode` ist `LIVE_REPLAY`, und 1 000, wenn `PlaybackMode` ist `ON_DEMAND`.

Der Maximalwert von 5 000 Fragmenten entspricht mehr als 80 Minuten Video in Streams mit Fragmenten von 1 Sekunde und mehr als 13 Stunden Video in Streams mit Fragmenten von 10 Sekunden.

Type: Long

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 5000.

Erforderlich: Nein

PlaybackMode

Gibt an, ob Live-, Live-Wiedergabe- oder archivierte On-Demand-Daten abgerufen werden sollen.

Zu den drei Sitzungstypen gehören die folgenden Funktionen:

- **LIVE** : Für Sitzungen dieses Typs wird die HLS-Medienwiedergabeliste kontinuierlich mit den neuesten Fragmenten aktualisiert, sobald sie verfügbar sind. Wir empfehlen, dass der Media Player in einem Intervall von einer Sekunde eine neue Wiedergabeliste abrufen. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel eine „Live“-Benachrichtigung an, ohne Scrubber-Steuer-element für die Auswahl der Position im Wiedergabefenster, die angezeigt werden soll.

Note

Im -LIVEModus sind die neuesten verfügbaren Fragmente in einer HLS-Medienwiedergabeliste enthalten, auch wenn eine Lücke zwischen Fragmenten besteht (d. h. wenn ein Fragment fehlt). Eine solche Lücke kann dazu führen, dass ein Media Player angehalten wird oder zu einem Sprung der Wiedergabe führt. In diesem Modus werden Fragmente nicht zur HLS-Medienwiedergabeliste hinzugefügt, wenn sie älter als das neueste Fragment in der Wiedergabeliste sind. Wenn das fehlende Fragment verfügbar ist, nachdem ein nachfolgendes Fragment zur Wiedergabeliste hinzugefügt wurde, wird das ältere Fragment nicht hinzugefügt und die Lücke wird nicht gefüllt.

- **LIVE_REPLAY** : Bei Sitzungen dieses Typs wird die HLS-Medienwiedergabeliste ähnlich aktualisiert wie die Aktualisierung für den -LIVEModus, mit der Ausnahme, dass sie mit dem Einschließen von Fragmenten aus einer bestimmten Startzeit beginnt. Anstatt bei der Aufnahme Fragmente hinzuzufügen, werden Fragmente hinzugefügt, wenn die Dauer des nächsten Fragments abgelaufen ist. Wenn die Fragmente in der Sitzung beispielsweise zwei Sekunden lang sind, wird der Medienwiedergabeliste alle zwei Sekunden ein neues Fragment hinzugefügt. Dieser Modus ist nützlich, um die Wiedergabe von zu starten, wenn ein Ereignis erkannt wird, und Live-Streaming-Medien fortzusetzen, die zum Zeitpunkt der Sitzungserstellung noch nicht aufgenommen wurden. Dieser Modus ist auch nützlich, um zuvor archivierte Medien zu

streamen, ohne durch das Limit von 1 000 Fragmenten im ON_DEMAND Modus eingeschränkt zu werden.

- **ON_DEMAND** : Für Sitzungen dieses Typs enthält die HLS-Medienwiedergabeliste alle Fragmente für die Sitzung bis zu der in angegebenen `ZahlMaxMediaPlaylistFragmentResults`. Die Wiedergabeliste darf für jede Sitzung nur einmal abgerufen werden. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel ein Scrubber-Steuerelement zur Auswahl der Position im Wiedergabefenster an, die angezeigt werden soll.

Wenn in allen Wiedergabemodi `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und wenn es mehrere Fragmente mit demselben Startzeitstempel gibt, wird das Fragment mit der größten Fragmentnummer (d. h. dem neuesten Fragment) in die HLS-Medienwiedergabeliste aufgenommen. Die anderen Fragmente sind nicht enthalten. Fragmente mit unterschiedlichen Zeitstempeln, aber überlappenden Dauern sind weiterhin in der HLS-Medienwiedergabeliste enthalten. Dies kann zu unerwartetem Verhalten im Media Player führen.

Der Standardwert ist `LIVE`.

Typ: Zeichenfolge

Zulässige Werte: `LIVE` | `LIVE_REPLAY` | `ON_DEMAND`

Erforderlich: Nein

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, für den die URL der HLS-Master-Wiedergabeliste abgerufen werden soll.

Sie müssen entweder die `StreamName` oder die `angebenStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, für den die URL der HLS-Master-Wiedergabeliste abgerufen werden soll.

Sie müssen entweder die `StreamName` oder die `angebenStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "HLSStreamingSessionURL": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[HLSStreamingSessionURL](#)

Die URL (mit dem Sitzungstoken), die ein Media Player zum Abrufen der HLS-Master-Wiedergabeliste verwenden kann.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anforderung gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Weitere Informationen zu Limits finden Sie unter [Limits für Kinesis Video Streams](#).

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Einschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einer der Spuren des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einer der Spuren des Videostreams wurden keine privaten Codec-Daten gefunden.

HTTP Status Code: 400

NoDataRetentionException

GetImages wurde für einen Stream angefordert, der keine Daten speichert (d. h. einen `DataRetentionInHours` von 0 hat).

HTTP Status Code: 400

NotAuthorizedException

Statuscode: 403, Der Aufrufer ist nicht autorisiert, eine Operation für den angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages löst diesen Fehler aus, wenn Kinesis Video Streams den angegebenen Stream nicht finden kann.

GetHLSStreamingSessionURL und GetDASHStreamingSessionURL geben diesen Fehler aus, wenn eine Sitzung mit einem PlaybackMode von ON_DEMAND oder für einen Stream ohne Fragmente innerhalb des angeforderten Zeitraums angefordert LIVE_REPLAY wird oder wenn eine Sitzung mit einem PlaybackMode von für einen Stream ohne Fragmente innerhalb der letzten 30 Sekunden angefordert LIVE wird.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

Der Typ der Medien (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte nicht anhand der Codec-IDs der Spuren im ersten Fragment für eine Wiedergabesitzung bestimmt werden. Die Codec-ID für Spur 1 sollte sein V_MPEG/ISO/AVC und optional sollte die Codec-ID für Spur 2 sein A_AAC.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetImages

Service: Amazon Kinesis Video Streams Archived Media

Ruft eine Liste von Bildern ab, die jedem Zeitstempel für einen bestimmten Zeitraum, ein Sampling-Intervall und eine Bildformatkonfiguration entsprechen.

Note

Sie müssen zuerst die GetDataEndpoint-API aufrufen, um einen Endpunkt abzurufen. Senden Sie dann die GetImages Anfragen an diesen Endpunkt mit dem [Parameter --endpoint-url](#).

[Anforderungen an die Videowiedergabespur](#)

Anforderungssyntax

```
POST /getImages HTTP/1.1
Content-type: application/json

{
  "EndTimeStamp": number,
  "Format": "string",
  "FormatConfig": {
    "string" : "string"
  },
  "HeightPixels": number,
  "ImageSelectorType": "string",
  "MaxResults": number,
  "NextToken": "string",
  "SamplingInterval": number,
  "StartTimeStamp": number,
  "StreamARN": "string",
  "StreamName": "string",
  "WidthPixels": number
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

EndTimeStamp

Der Endzeitstempel für den Bereich der zu generierenden Bilder. Wenn der Zeitraum zwischen `StartTimeStamp` und mehr als 300 Sekunden über `EndTimeStamp` liegt `StartTimeStamp`, erhalten Sie eine `IllegalArgumentException`.

Typ: Zeitstempel

Erforderlich: Ja

Format

Das Format, das zum Codieren des Bildes verwendet wird.

Typ: Zeichenfolge

Zulässige Werte: JPEG | PNG

Erforderlich: Ja

FormatConfig

Die Liste einer Schlüssel-Wert-Paarstruktur, die zusätzliche Parameter enthält, die bei der Generierung des Images angewendet werden können. Der `FormatConfig` Schlüssel ist der `JPEGQuality`, der den JPEG-Qualitätsschlüssel angibt, der zum Generieren des Bilds verwendet werden soll. Der `FormatConfig` Wert akzeptiert Ints von 1 bis 100. Wenn der Wert 1 ist, wird das Bild mit weniger Qualität und der besten Komprimierung generiert. Wenn der Wert 100 ist, wird das Image mit der besten Qualität und weniger Komprimierung generiert. Wenn kein Wert angegeben wird, wird der Standardwert des `JPEGQuality` Schlüssels auf 80 gesetzt.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 1 Element.

Gültige Schlüssel: `JPEGQuality`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertemuster: `^[a-zA-Z_0-9]+`

Erforderlich: Nein

HeightPixels

Die Höhe des Ausgabebilds, das in Verbindung mit dem `WidthPixels` Parameter verwendet wird. Wenn sowohl `HeightPixels` - als auch `WidthPixels` Parameter angegeben werden, wird das Bild gedehnt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `HeightPixels` Parameter angegeben wird, wird sein ursprüngliches Seitenverhältnis verwendet, um das `WidthPixels` Verhältnis zu berechnen. Wenn keiner der Parameter angegeben wird, wird die Originalbildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 2160.

Erforderlich: Nein

ImageSelectorType

Der Ursprung der Server- oder Producer-Zeitstempel, die zum Generieren der Images verwendet werden sollen.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Erforderlich: Ja

MaxResults

Die maximale Anzahl von Images, die von der API zurückgegeben werden sollen.

Note

Das Standardlimit beträgt 25 Images pro API-Antwort. Die Angabe eines `MaxResults` größer als dieser Wert führt zu einer Seitengröße von 25. Alle zusätzlichen Ergebnisse werden paginiert.

Type: Long

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

NextToken

Ein Token, das angibt, wo mit der Paginierung des nächsten Satzes von Images begonnen werden soll. Dies ist die `GetImages:NextToken` aus einer zuvor gekürzten Antwort.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 4096 Zeichen.

Pattern: `[a-zA-Z0-9+/\]+={0,2}`

Erforderlich: Nein

SamplingInterval

Das Zeitintervall in Millisekunden (ms), in dem die Images aus dem Stream generiert werden müssen. Der Mindestwert, der bereitgestellt werden kann, beträgt 200 ms (5 Bilder pro Sekunde). Wenn der Zeitstempelbereich kleiner als das Sampling-Intervall ist, `startTimestamp` wird das Bild aus der zurückgegeben, falls verfügbar.

Typ: Ganzzahl

Erforderlich: Ja

StartTimestamp

Der Ausgangspunkt, von dem die Bilder generiert werden sollen. Dies `StartTimestamp` muss innerhalb eines inklusiven Bereichs von Zeitstempeln liegen, damit ein Bild zurückgegeben werden kann.

Typ: Zeitstempel

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, aus dem die Images abgerufen werden sollen. Sie müssen entweder die `StreamName` oder die `angebenStreamARN`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Images abgerufen werden sollen. Sie müssen entweder die StreamName oder die angebenStreamARN.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

WidthPixels

Die Breite des Ausgabebilds, das in Verbindung mit dem HeightPixels Parameter verwendet wird. Wenn sowohl - als auch -WidthPixelsHeightPixelsParameter angegeben werden, wird das Bild so gedehnt, dass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der -WidthPixelsParameter oder nur der bereitgestellt HeightPixels wird, ValidationException wird ein ausgelöst. Wenn keiner der Parameter angegeben wird, wird die Originalbildgröße aus dem Stream zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 3840.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Images": [
    {
      "Error": "string",
```

```
    "ImageContent": "string",  
    "TimeStamp": number  
  }  
],  
"NextToken": "string"  
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

Images

Die Liste der aus dem Videostream generierten Bilder. Wenn für den angegebenen Zeitstempel keine Medien verfügbar sind, wird der NO_MEDIA Fehler in der Ausgabe aufgeführt. Wenn während der Generierung des Images ein Fehler auftritt, MEDIA_ERROR wird der in der Ausgabe als Ursache für das fehlende Image aufgeführt.

Typ: Array von [Image](#)-Objekten

NextToken

Das verschlüsselte Token, das in der Anforderung verwendet wurde, um mehr Bilder abzurufen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 4096 Zeichen.

Pattern: [a-zA-Z0-9+/\]+= {0, 2}

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anforderung gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Weitere Informationen zu Limits finden Sie unter [Limits für Kinesis Video Streams](#).

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Einschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

NotAuthorizedException

Statuscode: 403, Der Aufrufer ist nicht autorisiert, eine Operation für den angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages löst diesen Fehler aus, wenn Kinesis Video Streams den angegebenen Stream nicht finden kann.

GetHLSStreamingSessionURL und GetDASHStreamingSessionURL geben diesen Fehler aus, wenn eine Sitzung mit einem PlaybackMode von ON_DEMAND oder für einen Stream ohne Fragmente innerhalb des angeforderten Zeitraums angefordert LIVE_REPLAY wird oder wenn eine Sitzung mit einem PlaybackMode von für einen Stream ohne Fragmente innerhalb der letzten 30 Sekunden angefordert LIVE wird.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- [AWS -Befehlszeilenschnittstelle](#)
- [AWS SDK für .NET](#)
- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)

- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

GetMediaForFragmentList

Service: Amazon Kinesis Video Streams Archived Media

Ruft Medien für eine Liste von Fragmenten (angegeben durch die Fragmentnummer) aus den archivierten Daten in einem Amazon Kinesis Kinesis-Videostream ab.

Note

Sie müssen zuerst die `GetDataEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Senden Sie dann die `GetMediaForFragmentList` Anfragen mit dem Parameter `--endpoint-url` an diesen Endpunkt.

Grenzwerte finden Sie unter Limits für [Kinesis Video Streams](#).

Important

Wenn nach dem Aufrufen einer Kinesis Video Streams Streams-API für archivierte Medien ein Fehler ausgelöst wird, enthält dieser zusätzlich zum HTTP-Statuscode und dem Antworttext die folgenden Informationen:

- `x-amz-ErrorType` HTTP-Header — enthält zusätzlich zu den Angaben des HTTP-Statuscodes einen spezifischeren Fehlertyp.
- `x-amz-RequestId` HTTP-Header — Wenn Sie ein Problem melden möchten AWS, kann das Support-Team das Problem anhand der Anforderungs-ID besser diagnostizieren.

Sowohl der HTTP-Statuscode als auch der `ErrorType` Header können verwendet werden, um programmatische Entscheidungen darüber zu treffen, ob und unter welchen Bedingungen Fehler wiederholt werden können. Außerdem können Informationen darüber bereitgestellt werden, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erneut erfolgreich zu versuchen.

[Weitere Informationen finden Sie im Abschnitt Fehler am Ende dieses Themas sowie unter Häufige Fehler.](#)

Anforderungssyntax

```
POST /getMediaForFragmentList HTTP/1.1
```

```
Content-type: application/json

{
  "Fragments": [ "string" ],
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

Fragments

Eine Liste der Fragmente, für die Medien abgerufen werden sollen. Sie rufen diese Werte mit [abListFragments](#).

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 1000 Elemente.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[0-9]+$`

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, aus dem Fragmentmedien abgerufen werden sollen. Geben Sie entweder diesen Parameter oder den `StreamName` Parameter an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem Fragmentmedien abgerufen werden sollen. Geben Sie entweder diesen Parameter oder den `StreamARN` Parameter an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die Antwort gibt die folgenden HTTP-Header zurück.

ContentType

Der Inhaltstyp der angeforderten Medien.

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[a-zA-Z0-9_\.\-]+$`

Die Antwort gibt folgendes als HTTP-Hauptteil zurück.

Payload

Die Payload, die Kinesis Video Streams zurückgibt, ist eine Sequenz von Chunks aus dem angegebenen Stream. Informationen zu den Chunks finden Sie unter [PutMedia](#). Die Chunks, die

Kinesis Video Streams im `GetMediaForFragmentList` Aufruf zurückgibt, enthalten auch die folgenden zusätzlichen Matroska-Tags (MKV):

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` — Im Chunk zurückgegebene Fragmentnummer.
- `AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP` — Serverseitiger Zeitstempel des Fragments.
- `AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP` — Produzentenseitiger Zeitstempel des Fragments.

Die folgenden Tags werden hinzugefügt, wenn eine Ausnahme auftritt:

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` — Die Nummer des Fragments, das die Ausnahme ausgelöst hat.
- `AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE` — Der Integer-Code des Fehlers.
- `AWS_KINESISVIDEO_EXCEPTION_MESSAGE` — Eine Textbeschreibung der Ausnahme.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

`ClientLimitExceededException`

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Limits finden Sie unter [Kinesis Video Streams Streams-Limits](#).

HTTP Status Code: 400

`InvalidArgumentException`

Ein bestimmter Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

`NotAuthorizedException`

Statuscode: 403, Der Aufrufer ist nicht berechtigt, einen Vorgang mit dem angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages löst diesen Fehler aus, wenn Kinesis Video Streams den von Ihnen angegebenen Stream nicht finden kann.

GetHLSStreamingSessionURL und GetDASHStreamingSessionURL gibt diesen Fehler aus, wenn eine Sitzung mit einem PlaybackMode of ON_DEMAND oder für einen Stream angefordert LIVE_REPLAY wird, der innerhalb des angeforderten Zeitraums keine Fragmente enthält, oder wenn eine Sitzung mit einem PlaybackMode of für einen Stream angefordert LIVE wird, der innerhalb der letzten 30 Sekunden keine Fragmente enthält.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

ListFragments

Service: Amazon Kinesis Video Streams Archived Media

Gibt eine Liste von [Fragment](#) Objekten aus dem angegebenen Stream- und Zeitstempelbereich innerhalb der archivierten Daten zurück.

Das Auflisten von Fragmenten ist letztendlich konsistent. Das bedeutet, dass selbst wenn der Hersteller eine Bestätigung erhält, dass ein Fragment persistent ist, das Ergebnis einer Anfrage an möglicherweise nicht sofort zurückgegeben wird. `ListFragments` Die Ergebnisse sind jedoch in der Regel in weniger als einer Sekunde verfügbar.

Note

Sie müssen zuerst die `GetDataEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Senden Sie dann die `ListFragments` Anfragen mit dem Parameter [--endpoint-url an diesen Endpunkt](#).

Important

Wenn nach dem Aufrufen einer Kinesis Video Streams Streams-API für archivierte Medien ein Fehler ausgelöst wird, enthält dieser zusätzlich zum HTTP-Statuscode und dem Antworttext die folgenden Informationen:

- `x-amz-ErrorType`HTTP-Header — enthält zusätzlich zu dem, was der HTTP-Statuscode bietet, einen spezifischeren Fehlertyp.
- `x-amz-RequestId`HTTP-Header — Wenn Sie ein Problem melden möchtenAWS, kann das Support-Team das Problem anhand der Anforderungs-ID besser diagnostizieren.

Sowohl der HTTP-Statuscode als auch der `ErrorType` Header können verwendet werden, um programmatische Entscheidungen darüber zu treffen, ob und unter welchen Bedingungen Fehler wiederholt werden können. Außerdem können Informationen darüber bereitgestellt werden, welche Maßnahmen der Client-Programmierer möglicherweise ergreifen muss, um es erneut erfolgreich zu versuchen.

[Weitere Informationen finden Sie im Abschnitt Fehler am Ende dieses Themas sowie unter Häufige Fehler.](#)

Anforderungssyntax

```
POST /listFragments HTTP/1.1
Content-type: application/json

{
  "FragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

FragmentSelector

Beschreibt den Zeitstempelbereich und den Ursprung des Zeitstempels für den Bereich der zurückzugebenden Fragmente.

Note

Dies ist nur erforderlich, wenn der NextToken nicht in der API übergeben wird.

Typ: FragmentSelector Objekt

Erforderlich: Nein

MaxResults

Die Gesamtzahl der zurückzugebenden Fragmente. Wenn die Gesamtzahl der verfügbaren Fragmente den in angegebenen Wert übersteigt `max-results`, `NextToken` wird in der Ausgabe ein [ListFragments](#): bereitgestellt, mit dem Sie die Paginierung fortsetzen können.

Der Standardwert lautet 100.

Type: Long

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 1 000.

Erforderlich: Nein

NextToken

Ein Token für den Beginn der Seitennummerierung. Dies ist das [ListFragments](#): `NextToken` aus einer zuvor gekürzten Antwort.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 4096 Zeichen.

Pattern: `[a-zA-Z0-9+/]+={0,2}`

Erforderlich: Nein

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, aus dem eine Fragmentliste abgerufen werden soll. Geben Sie entweder diesen Parameter oder den `StreamName` Parameter an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem eine Fragmentliste abgerufen werden soll. Geben Sie entweder diesen Parameter oder den `StreamARN` Parameter an.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Fragments": [
    {
      "FragmentLengthInMilliseconds": number,
      "FragmentNumber": "string",
      "FragmentSizeInBytes": number,
      "ProducerTimestamp": number,
      "ServerTimestamp": number
    }
  ],
  "NextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[Fragments](#)

Eine Liste der archivierten [Fragment](#) Objekte aus dem Stream, die die Auswahlkriterien erfüllen. Die Ergebnisse sind in keiner bestimmten Reihenfolge, auch nicht seitenübergreifend.

Wenn der Stream keine Fragmente enthält, die die Auswahlkriterien erfüllen, wird eine leere Liste zurückgegeben.

Typ: Array von [Fragment](#)-Objekten

NextToken

Wenn die zurückgegebene Liste gekürzt ist, gibt der Vorgang dieses Token zurück, mit dem die nächste Ergebnisseite abgerufen werden kann. Dieser Wert ist angegeben null, wenn keine weiteren Ergebnisse zurückgegeben werden können.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 4096 Zeichen.

Pattern: `[a-zA-Z0-9+/\]+= {0, 2}`

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Limits finden Sie unter [Kinesis Video Streams Streams-Limits](#).

HTTP Status Code: 400

InvalidArgumentException

Ein angegebener Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

NotAuthorizedException

Statuscode: 403, Der Aufrufer ist nicht berechtigt, einen Vorgang mit dem angegebenen Stream auszuführen, oder das Token ist abgelaufen.

HTTP-Statuscode: 401

ResourceNotFoundException

GetImages löst diesen Fehler aus, wenn Kinesis Video Streams den von Ihnen angegebenen Stream nicht finden kann.

`GetHLSStreamingSessionURL` und `GetDASHStreamingSessionURL` gibt diesen Fehler aus, wenn eine Sitzung mit einem `PlaybackMode` of `ON_DEMAND` oder für einen Stream angefordert `LIVE_REPLAY` wird, der innerhalb des angeforderten Zeitraums keine Fragmente enthält, oder wenn eine Sitzung mit einem `PlaybackMode` of für einen Stream angefordert `LIVE` wird, der innerhalb der letzten 30 Sekunden keine Fragmente enthält.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS-SDK für PHP V3](#)
- [AWS-SDK für Python](#)
- [AWS-SDK für Ruby V3](#)

Amazon Kinesis Video Video Streams

Die folgenden Aktionen werden von Amazon Kinesis Video Signaling Channels unterstützt:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

GetIceServerConfig

Bedienung: Amazon Kinesis Video Signaling Channels

Hinweis: Bevor Sie diese API verwenden können, müssen Sie die `GetSignalingChannelEndpoint` API aufrufen, um den HTTPS-Endpunkt anzufordern. Anschließend geben Sie den Endpunkt und die Region in Ihrer `GetIceServerConfig` API-Anfrage an.

Ruft die ICE-Serverkonfigurationsinformationen (Interactive Connectivity Establishment) ab, einschließlich URIs, Benutzername und Kennwort, die zur Konfiguration der WebRTC-Verbindung verwendet werden können. Die ICE-Komponente verwendet diese Konfigurationsinformationen, um die WebRTC-Verbindung einzurichten, einschließlich der Authentifizierung beim Relay-Server Traversal Using Relays around NAT (TURN).

TURN ist ein Protokoll, das zur Verbesserung der Konnektivität von Anwendungen verwendet wird. peer-to-peer Durch die Bereitstellung eines cloudbasierten Relay-Dienstes stellt TURN sicher, dass eine Verbindung auch dann hergestellt werden kann, wenn ein oder mehrere Peers keine direkte peer-to-peer Verbindung herstellen können. Weitere Informationen finden Sie unter [Eine REST-API für den Zugriff auf TURN-Dienste](#).

Sie können diese API aufrufen, um einen Fallback-Mechanismus einzurichten, falls einer der Peers keine direkte peer-to-peer Verbindung über einen Signalkanal herstellen kann. Sie müssen den Amazon-Ressourcennamen (ARN) Ihres Signalisierungskanals angeben, um diese API aufzurufen.

Anforderungssyntax

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ClientId": "string",
  "Service": "string",
  "Username": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der ARN des Signalkanals, der für die peer-to-peer Verbindung zwischen konfigurierten Peers verwendet werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

ClientId

Eindeutiger Bezeichner für den Betrachter. Muss innerhalb des Signalkanals eindeutig sein.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Service

Gibt den gewünschten Dienst an. Derzeit TURN ist dies der einzig gültige Wert.

Typ: Zeichenfolge

Zulässige Werte: TURN

Required: No

Username

Eine optionale Benutzer-ID, die den Anmeldeinformationen zugeordnet werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Required: No

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "IceServerList": [
    {
      "Password": "string",
      "Ttl": number,
      "Uris": [ "string" ],
      "Username": "string"
    }
  ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

[IceServerList](#)

Die Liste der ICE-Serverinformationsobjekte.

Typ: Array von [IceServer](#)-Objekten

Fehler

Hinweise zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Ihre Anfrage wurde gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

InvalidClientException

Der angegebene Client ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Die angegebene Ressource wurde nicht gefunden.

HTTP Status Code: 404

SessionExpiredException

Wenn die Clientsitzung abgelaufen ist. Sobald der Client verbunden ist, ist die Sitzung 45 Minuten lang gültig. Der Client sollte sich erneut mit dem Kanal verbinden, um weiterhin Nachrichten senden/empfangen zu können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS JavaScript SDK für V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

SendAlexaOfferToMaster

Bedienung: Amazon Kinesis Video Signaling Channels

Note

Bevor Sie diese API verwenden, müssen Sie die `GetSignalingChannelEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Anschließend geben Sie den Endpunkt und die Region in Ihrer `SendAlexaOfferToMaster` API-Anfrage an.

Diese API ermöglicht es Ihnen, WebRTC-fähige Geräte mit Alexa-Anzeigegeräten zu verbinden. Wenn sie aufgerufen wird, sendet sie das SDP-Angebot (Alexa Session Description Protocol) an den Master-Peer. Das Angebot wird zugestellt, sobald der Master mit dem angegebenen Signalkanal verbunden ist. Diese API gibt die SDP-Antwort vom verbundenen Master zurück. Wenn der Master nicht mit dem Signalkanal verbunden ist, werden Anfragen zur erneuten Zustellung gestellt, bis die Nachricht abläuft.

Anforderungssyntax

```
POST /v1/send-alex-a-offer-to-master HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "MessagePayload": "string",
  "SenderClientId": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

ChannelARN

Der Amazon-Ressourcenname (ARN) des Signalkanals, über den Alexa und der Master-Peer kommunizieren.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

MessagePayload

Das Base64-kodierte SDP bietet Inhalte.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 10000.

Pattern: `[a-zA-Z0-9+/=]+`

Erforderlich: Ja

SenderClientId

Die eindeutige Kennung für den Absender-Client.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Answer": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

Answer

Der Base64-kodierte SDP-Antwortinhalt.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Die maximale Länge beträgt 10000.

Fehler

Informationen zu den Fehlern, die allen Aktionen gemeinsam sind, finden Sie unter [Häufige Fehler](#).

ClientLimitExceededException

Ihre Anfrage wurde gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

NotAuthorizedException

Der Anrufer ist nicht autorisiert, diesen Vorgang auszuführen.

HTTP-Statuscode: 401

ResourceNotFoundException

Die angegebene Ressource wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-Befehlszeilenschnittstelle](#)
- [AWS-SDK für .NET](#)
- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWSSDK für JavaScript V3](#)
- [AWS SDK für PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK für Ruby V3](#)

Amazon Kinesis Video WebRTC Storage

Folgende Aktionen werden von Amazon Kinesis Video WebRTC Video WebRTC Storage unterstützt:

- [JoinStorageSession](#)

JoinStorageSession

Service: Amazon Kinesis Video WebRTC Storage

Note

Bevor Sie diese API verwenden, müssen Sie die `GetSignalingChannelEndpoint` API aufrufen, um den WebRTC-Endpunkt anzufordern. Anschließend geben Sie den Endpunkt und die Region in Ihrer `JoinStorageSession` API-Anfrage an.

Nehmen Sie an der laufenden WebRTC-Sitzung mit Einweg-Video und/oder Mehrwege-Audio als Videoproduktionsgerät für einen Eingangskanal teil. Wenn für den Kanal keine Sitzung vorhanden ist, muss eine neue Streaming-Sitzung erstellt und der Amazon-Ressourcenname (ARN) des Signalkanals angegeben werden.

Derzeit kann ein Videoproduktionsgerät für diesen `SINGLE_MASTER` Typ sowohl Audio- als auch Videomedien in einen Stream aufnehmen. Nur Videoproduktionsgeräte können an der Sitzung teilnehmen und Medien aufnehmen.

Important

Für die WebRTC-Aufnahme sind derzeit sowohl Audio- als auch Videospuren erforderlich.

Sobald ein Master-Teilnehmer eine Verbindung über WebRTC ausgehandelt hat, wird die aufgenommene Mediensitzung im Kinesis-Videostream gespeichert. Mehrere Zuschauer können dann Medien in Echtzeit über unsere Playback-APIs wiedergeben.

Kunden können auch bestehende Kinesis Video Streams Streams-Funktionen wie HLS DASH OD-Wiedergabe, Bilderzeugung und mehr mit aufgenommenen WebRTC-Medien verwenden.

Note

Die Bereitstellung von S3-Bildern wird derzeit nicht unterstützt.

Note

Gehen Sie davon aus, dass einer Sitzung für den Kanal nur ein Client für das Videoproduktionsgerät zugeordnet werden kann. Wenn mehr als ein Client der Sitzung eines bestimmten Kanals als Videoproduktionsgerät beitrifft, hat die neueste Client-Anfrage Vorrang.

Zusätzliche Informationen

- Idempotent — Diese API ist nicht idempotent.
- Wiederholungsverhalten — Dies wird als neuer API-Aufruf gezählt.
- Gleichzeitige Aufrufe — Gleichzeitige Aufrufe sind zulässig. Pro Aufruf wird einmal ein Angebot gesendet.

Anforderungssyntax

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json

{
  "channelArn": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

channelArn

Der Amazon-Ressourcenname (ARN) des Signalisierungskanals.

Typ: Zeichenfolge

Pattern: `^arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+$`

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter [Häufige Fehler](#).

AccessDeniedException

Sie verfügen nicht über die erforderlichen Berechtigungen, um diesen Vorgang auszuführen.

HTTP Status Code: 403

ClientLimitExceededException

Kinesis Video Streams hat die Anfrage gedrosselt, weil Sie das Limit der erlaubten Client-Aufrufe überschritten haben. Versuchen Sie später, den Anruf zu tätigen.

HTTP Status Code: 400

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

ResourceNotFoundException

Die angegebene Ressource wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- [AWS -Befehlszeilenschnittstelle](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK für JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK für Python](#)
- [AWS SDK for Ruby V3](#)

Datentypen

Die folgenden Datentypen werden von Amazon Kinesis Video Streams unterstützt:

- [ChannelInfo](#)
- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)
- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)

- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)
- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

Die folgenden Datentypen werden von Amazon Kinesis Video Streams Media unterstützt:

- [StartSelector](#)

Die folgenden Datentypen werden von Amazon Kinesis Video Streams Archived Media unterstützt:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Die folgenden Datentypen werden von Amazon Kinesis Video Signaling Channels unterstützt:

- [IceServer](#)

Die folgenden Datentypen werden von Amazon Kinesis Video WebRTC Storage unterstützt:

Amazon Kinesis Video Streams

Die folgenden Datentypen werden von Amazon Kinesis Video Streams unterstützt:

- [ChannelInfo](#)
- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)
- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)
- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)
- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

ChannelInfo

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Metadaten und Eigenschaften eines Signalkanals kapselt.

Inhalt

ChannelARN

Der Amazon-Ressourcenname (ARN) des Signalisierungskanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

ChannelName

Der Name des Signalisierungskanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

ChannelStatus

Aktueller Status des Signalisierungskanals.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | UPDATING | DELETING`

Required: No

ChannelType

Der Typ des Signalisierungskanals.

Typ: Zeichenfolge

Zulässige Werte: SINGLE_MASTER | FULL_MESH

Required: No

CreationTime

Die Uhrzeit, zu der die Signalisierungskanals erstellt wurde.

Typ: Zeitstempel

Required: No

SingleMasterConfiguration

Eine Struktur, die die Konfiguration für den SINGLE_MASTER Kanaltyp enthält.

Typ: [SingleMasterConfiguration](#) Objekt

Required: No

Version

Die aktuelle Version des Signalisierungskanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: [a-zA-Z0-9]+

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ChannelNameCondition

Bedienung: Amazon Kinesis Video Streams

Ein optionaler Eingabeparameter für die `ListSignalingChannels` API. Wenn dieser Parameter beim Aufrufen angegeben wird `ListSignalingChannels`, gibt die API nur die Kanäle zurück, die eine in `ChannelNameCondition` angegebene Bedingung erfüllen.

Inhalt

ComparisonOperator

Ein Vergleichsoperator. Derzeit können Sie nur den `BEGINS_WITH` Operator angeben, der Signalkanäle findet, deren Namen mit einem bestimmten Präfix beginnen.

Typ: Zeichenfolge

Zulässige Werte: `BEGINS_WITH`

Required: No

ComparisonValue

Ein Wert zum Vergleich.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DeletionConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, die erforderlich sind, um die Verbindung des Streams aus dem Edge-Agenten zu löschen.

Inhalt

DeleteAfterUpload

Der `boolean` Wert, der angibt, ob Sie das Medium zum Löschen markieren möchten oder nicht, sobald es in die Kinesis Video Stream Cloud hochgeladen wurde. Die Mediendateien können gelöscht werden, wenn einer der Löschkonfigurationswerte auf gesetzt ist `true`, z. B. wenn das Limit für die `EdgeRetentionInHours`, oder die `MaxLocalMediaSizeInMB`, erreicht wurde.

Da der Standardwert auf gesetzt ist `true`, konfigurieren Sie den Uploader-Zeitplan so, dass die Mediendateien nicht gelöscht werden, bevor sie zum ersten Mal in die AWS Cloud hochgeladen werden.

Typ: Boolesch

Required: No

EdgeRetentionInHours

Die Anzahl der Stunden, für die Sie die Daten im Stream auf dem Edge-Agenten aufbewahren möchten. Der Standardwert der Aufbewahrungszeit beträgt 720 Stunden, was 30 Tagen entspricht.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von Maximaler Wert von 60.

Required: No

LocalSizeConfig

Der Wert der lokalen Größe, der zum Löschen der Edge-Konfiguration erforderlich ist.

Typ: [LocalSizeConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

EdgeAgentStatus

Bedienung: Amazon Kinesis Video Streams

Ein Objekt, das die neuesten Statusdetails für die Recorder- und Uploader-Jobs eines Edge-Agenten enthält. Verwenden Sie diese Informationen, um den aktuellen Zustand eines Edge-Agenten zu ermitteln.

Inhalt

LastRecorderStatus

Der neueste Status des Edge-Aufzeichnungsauftrags eines Streams.

Typ: [LastRecorderStatus](#) Objekt

Required: No

LastUploaderStatus

Der neueste Status des Edge-to-Cloud-Uploader-Jobs eines Streams.

Typ: [LastUploaderStatus](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

EdgeConfig

Bedienung: Amazon Kinesis Video Streams

Eine Beschreibung der Edge-Konfiguration des Streams, die für die Synchronisierung mit der Edge Agent IoT Greengrass-Komponente verwendet wird. Die Edge Agent-Komponente wird auf einem IoT-Hub-Gerät ausgeführt, das bei Ihnen vor Ort eingerichtet ist.

Inhalt

HubDeviceArn

Das „Internet der Dinge (IoT)“ ist der Kern des Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Erforderlich: Ja

RecorderConfig

Die Rekorderkonfiguration besteht aus den lokalen `MediaSourceConfig` Details, die als Anmeldeinformationen für den Zugriff auf die lokalen Mediendateien verwendet werden, die auf der Kamera gestreamt werden.

Typ: [RecorderConfig](#) Objekt

Erforderlich: Ja

DeletionConfig

Die Löschkonfiguration besteht aus den Angaben zur Aufbewahrungszeit (`EdgeRetentionInHours`) und der lokalen Größenkonfiguration (`LocalSizeConfig`), die für den Löschvorgang verwendet wurden.

Typ: [DeletionConfig](#) Objekt

Required: No

UploaderConfig

Die Uploader-Konfiguration enthält die `ScheduleExpression` Details, die verwendet werden, um Uploadjobs für die aufgezeichneten Mediendateien vom Edge-Agenten auf einen Kinesis-Videostream zu planen.

Typ: [UploaderConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ImageGenerationConfiguration

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die für die Lieferung der KVS-Images erforderlichen Informationen enthält. Bei Null wird die Konfiguration aus dem Stream gelöscht.

Inhalt

DestinationConfig

Die Struktur, die die Informationen enthält, die für die Lieferung von Bildern an einen Kunden erforderlich sind.

Typ: [ImageGenerationDestinationConfig](#) Objekt

Erforderlich: Ja

Format

Das akzeptierte Bildformat.

Typ: Zeichenfolge

Zulässige Werte: JPEG | PNG

Erforderlich: Ja

ImageSelectorType

Der Ursprung der Server- oder Producer-Zeitstempel, die zum Generieren der Bilder verwendet werden sollen.

Typ: Zeichenfolge

Zulässige Werte: SERVER_TIMESTAMP | PRODUCER_TIMESTAMP

Erforderlich: Ja

SamplingInterval

Das Zeitintervall in Millisekunden (ms), in dem die Bilder aus dem Stream generiert werden müssen. Der Mindestwert, der angegeben werden kann, ist 200 ms. Wenn der Zeitstempelbereich kleiner als das Abtastintervall ist, wird das Bild aus dem `StartTimeStamp` zurückgegeben, falls verfügbar.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert von 3000. Höchstwert von 20000.

Erforderlich: Ja

Status

Gibt an, ob der `ContinuousImageGenerationConfigurationsAPI` ist aktiviert oder deaktiviert.

Typ: Zeichenfolge

Zulässige Werte: ENABLED | DISABLED

Erforderlich: Ja

FormatConfig

Die Liste einer Schlüssel-Wert-Paarstruktur, die zusätzliche Parameter enthält, die bei der Generierung des Bildes angewendet werden können. Der `FormatConfigSchlüssel` ist der `JPEGQuality`, der den JPEG-Qualitätsschlüssel angibt, der zum Generieren des Bildes verwendet werden soll. Der `FormatConfigvalue` akzeptiert Ganzzahlen von 1 bis 100. Wenn der Wert 1 ist, wird das Bild mit geringerer Qualität und der besten Komprimierung generiert. Wenn der Wert 100 ist, wird das Bild mit der besten Qualität und geringerer Komprimierung generiert. Wenn kein Wert angegeben wird, ist der Standardwert des `JPEGQuality` Der Schlüssel wird auf 80 gesetzt.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Karteneinträge: Maximale Anzahl von 1 Element.

Gültige Schlüssel: `JPEGQuality`

Einschränkungen der Wertelänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Wertmuster: `^[a-zA-Z_0-9]+`

Required: No

HeightPixels

Die Höhe des Ausgabebildes, das in Verbindung mit dem verwendet wird `WidthPixelsParameter`. Wenn beide `HeightPixels` und `WidthPixelsParameter`

werden bereitgestellt, das Bild wird gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `HeightPixels` Parameter bereitgestellt, sein ursprüngliches Seitenverhältnis wird zur Berechnung des `WidthPixels` Verhältnis. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert von 2160.

Required: No

WidthPixels

Die Breite des Ausgabebildes, das in Verbindung mit dem verwendet wird `HeightPixels` Parameter. Wenn beide `WidthPixels` und `HeightPixels` Parameter werden bereitgestellt, das Bild wird gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `WidthPixels` Parameter bereitgestellt, sein ursprüngliches Seitenverhältnis wird zur Berechnung des `HeightPixels` Verhältnis. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert von 3840.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ImageGenerationDestinationConfig

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die Informationen enthält, die für die Lieferung von Bildern an einen Kunden erforderlich sind.

Inhalt

DestinationRegion

Die AWS Region des S3-Buckets, in die Bilder geliefert werden. Dies `DestinationRegion` muss mit der Region übereinstimmen, in der sich der Stream befindet.

Typ: Zeichenfolge

Längenbeschränkungen: Länge beträgt 999 Zeichen, Länge beträgt 9

Längenbeschränkungen: Länge beträgt 14 Zeichen, Länge beträgt 14 Zeichen, Länge von 14

Pattern: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Erforderlich: Ja

Uri

Der URI der Uniform Resource Identifier (URI), der Identifier (URI) des ers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 255 Zeichen.

Pattern: `^[a-zA-Z_0-9]+:(//)?([/+])/?([^*] *)$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

LastRecorderStatus

Bedienung: Amazon Kinesis Video Streams

Der neueste Status des Edge-Aufzeichnungsauftrags eines Streams.

Inhalt

JobStatusDetails

Eine Beschreibung des letzten Status eines Aufzeichnungsauftrags.

Typ: Zeichenfolge

Required: No

LastCollectedTime

Der Zeitstempel, zu dem der Aufzeichnungsauftrag zuletzt ausgeführt und die Medien auf der lokalen Festplatte gespeichert wurden.

Typ: Zeitstempel

Required: No

LastUpdatedTime

Der Zeitstempel, zu dem der Rekorderstatus zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

RecorderStatus

Der Status des letzten Aufzeichnungsauftrags.

Typ: Zeichenfolge

Zulässige Werte: SUCCESS | USER_ERROR | SYSTEM_ERROR

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

LastUploaderStatus

Bedienung: Amazon Kinesis Video Streams

Der neueste Status des Edge-to-Cloud-Uploader-Jobs eines Streams.

Inhalt

JobStatusDetails

Eine Beschreibung des letzten Status eines Uploader-Jobs.

Typ: Zeichenfolge

Required: No

LastCollectedTime

Der Zeitstempel, an dem der Uploader-Job zuletzt ausgeführt und die Medien in der Cloud gesammelt wurden.

Typ: Zeitstempel

Required: No

LastUpdatedTime

Der Zeitstempel, zu dem der Uploader-Status zuletzt aktualisiert wurde.

Typ: Zeitstempel

Required: No

UploaderStatus

Der Status des letzten Uploader-Jobs.

Typ: Zeichenfolge

Zulässige Werte: SUCCESS | USER_ERROR | SYSTEM_ERROR

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ListEdgeAgentConfigurationsEdgeConfig

Bedienung: Amazon Kinesis Video Streams

Eine Beschreibung der Edge-Konfiguration eines einzelnen Streams.

Inhalt

CreationTime

Der Zeitstempel, als der Stream die Edge-Konfiguration zum ersten Mal erstellt hat.

Typ: Zeitstempel

Required: No

EdgeConfig

Eine Beschreibung der Edge-Konfiguration des Streams, die für die Synchronisierung mit der Edge Agent IoT Greengrass-Komponente verwendet wird. Die Edge Agent-Komponente wird auf einem IoT-Hub-Gerät ausgeführt, das bei Ihnen vor Ort eingerichtet ist.

Typ: [EdgeConfig](#) Objekt

Required: No

FailedStatusDetails

Eine Beschreibung des generierten Fehlerstatus.

Typ: Zeichenfolge

Required: No

LastUpdatedTime

Der Zeitstempel, als der Stream die Edge-Konfiguration zuletzt aktualisiert hat.

Typ: Zeitstempel

Required: No

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Name des -Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

SyncStatus

Der aktuelle Synchronisierungsstatus der Edge-Konfiguration des Streams.

Typ: Zeichenfolge

Zulässige Werte: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

LocalSizeConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, die die maximale Größe der Medien (`MaxLocalMediaSizeInMB`) beinhalten, die Sie für einen Stream auf dem Edge-Agenten speichern möchten, sowie die Strategie, die verwendet werden sollte (`StrategyOnFullSize`), wenn die maximale Größe eines Streams erreicht wurde.

Inhalt

MaxLocalMediaSizeInMB

Die maximale Gesamtgröße der Medien, die Sie für einen Stream auf dem Edge Agent speichern möchten.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 64. Maximaler Wert von 2000.

Required: No

StrategyOnFullSize

Die Strategie, die ausgeführt werden soll, wenn das `MaxLocalMediaSizeInMB` Limit eines Streams erreicht ist.

Typ: Zeichenfolge

Zulässige Werte: `DELETE_OLDEST_MEDIA` | `DENY_NEW_MEDIA`

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MappedResourceConfigurationListItem

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Konfigurationseigenschaften des Medienspeichers kapselt oder enthält.

Inhalt

ARN

Der Amazon-Ressourcenname (ARN) der Kinesis Data Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

Type

Der Typ der zugehörigen Ressource für den Kinesis-Videostream.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MediaSourceConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, die aus den Anmeldeinformationen bestehen, die für den Zugriff auf die Mediendateien erforderlich sind (`MediaUriSecretArn` und `MediaUriType`), die auf die Kamera gestreamt werden.

Inhalt

MediaUriSecretArn

Der AWS Secrets Manager Manager-ARN für den Benutzernamen und das Passwort der Kamera oder einen lokalen Speicherort für Mediendateien.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 20. Maximale Länge beträgt 2048 Zeichen.

Pattern: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

Erforderlich: Ja

MediaUriType

Der URI (Uniform Resource Identifier). Der `FILE_URI` Wert kann verwendet werden, um lokale Mediendateien zu streamen.

Note

Die Vorschau unterstützt nur das URI-Format der `RTSP_URI` Medienquelle.

Typ: Zeichenfolge

Zulässige Werte: `RTSP_URI` | `FILE_URI`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

MediaStorageConfiguration

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Eigenschaften der Medienspeicherkonfiguration kapselt oder enthält.

- Wenn diese Option aktiviert `StorageStatus` ist, werden die Daten in der `StreamARN` bereitgestellten Datei gespeichert. Damit WebRTC Ingestion funktioniert, muss für den Stream die Datenspeicherung aktiviert sein.
- Wenn deaktiviert `StorageStatus` ist, werden keine Daten gespeichert und der `StreamARN` Parameter wird nicht benötigt.

Inhalt

Status

Der Status der Medienspeicherkonfiguration.

Typ: Zeichenfolge

Zulässige Werte: ENABLED | DISABLED

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des -Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

NotificationConfiguration

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die Benachrichtigungsinformationen für die Lieferung der KVS-Images enthält. Wenn dieser Parameter Null ist, wird die Konfiguration aus dem Stream gelöscht.

Inhalt

DestinationConfig

Die Zielinformationen, die für die Übermittlung einer Benachrichtigung an einen Kunden erforderlich sind.

Typ: [NotificationDestinationConfig](#) Objekt

Erforderlich: Ja

Status

Gibt an, ob eine Benachrichtigungskonfiguration aktiviert oder deaktiviert ist.

Typ: Zeichenfolge

Zulässige Werte: ENABLED | DISABLED

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

NotificationDestinationConfig

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die Informationen enthält, die für die Übermittlung einer Benachrichtigung an einen Kunden erforderlich sind.

Inhalt

Uri

Der URI (Uniform Resource Identifier (URI) des Bestellers.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 255 Zeichen.

Pattern: `^[a-zA-Z_0-9]+:(//)?([/])/?([^*] *)$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

RecorderConfig

Bedienung: Amazon Kinesis Video Streams

Die Recorder-Konfiguration besteht aus den `localMediaSourceConfig` Details, die als Anmeldeinformationen für den Zugriff auf die lokalen Mediendateien verwendet werden, die auf der Kamera gestreamt werden.

Inhalt

MediaSourceConfig

Die Konfigurationsdetails, die aus den Anmeldeinformationen bestehen, die für den Zugriff auf die an die Kamera gestreamten Mediendateien erforderlich sind (`MediaUriSecretArn` und `MediaUriType`).

Typ: [MediaSourceConfig](#) Objekt

Erforderlich: Ja

ScheduleConfig

Die Konfiguration, die aus dem `ScheduleExpression` und den `DurationInMinutes` Details besteht, die den Zeitplan für die Aufnahme von einer Kamera oder einer lokalen Mediendatei auf den Edge Agent festlegen. Wenn das `ScheduleExpression` Attribut nicht angegeben wird, wird der Edge-Agent immer in den Aufnahmemodus versetzt.

Typ: [ScheduleConfig](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ResourceEndpointListItem

Bedienung: Amazon Kinesis Video Streams

Ein Objekt, das den Endpunkt der von der `GetSignalingChannelEndpoint` API zurückgegebenen Signalkanals beschreibt.

Der Medienserver-Endpoint entspricht dem WEBRTC Protokoll.

Inhalt

Protocol

Das Protokoll des von der `GetSignalingChannelEndpoint` API zurückgegebenen Signalkanals.

Typ: Zeichenfolge

Zulässige Werte: WSS | HTTPS | WEBRTC

Required: No

ResourceEndpoint

Der Endpunkt des von der `GetSignalingChannelEndpoint` API zurückgegebenen Signalkanals.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ScheduleConfig

Bedienung: Amazon Kinesis Video Streams

Mit dieser API können Sie die Dauer angeben, die die Kamera oder die lokale Mediendatei auf dem Edge Agent aufzeichnen soll. Das `ScheduleConfig` besteht aus den Attributen `ScheduleExpression` und den `DurationInMinutes` Attributen.

Wenn der nicht in der angegebenen `ScheduleConfig` ist `RecorderConfig`, wird der Edge-Agent immer in den Aufnahmemodus versetzt.

Wenn der nicht in der angegebenen `ScheduleConfig` ist `UploaderConfig`, lädt der Edge-Agent in regelmäßigen Abständen (alle 1 Stunde) hoch.

Inhalt

DurationInSeconds

Die Gesamtdauer für die Aufzeichnung der Medien. Wenn das `ScheduleExpression` Attribut angegeben wird, sollte das `DurationInSeconds` Attribut auch angegeben werden.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 60. Wert von 60.

Erforderlich: Ja

ScheduleExpression

Der Quartz-Cron-Ausdruck, der sich um die Planung von Aufzeichnungsaufträgen von der Kamera oder einer lokalen Mediendatei auf den Edge Agent kümmert. Wenn der für nicht vorgesehen `ScheduleExpression` ist `RecorderConfig`, wird der Edge-Agent immer in den Aufnahmemodus versetzt.

Weitere Informationen zu Quartz finden Sie [auf der Seite Cron Trigger Tutorial](#), um die gültigen Ausdrücke und ihre Verwendung zu verstehen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 100 Zeichen.

Pattern: `[^\n]{1,100}`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

SingleMasterChannelEndpointConfiguration

Bedienung: Amazon Kinesis Video Streams

Ein Objekt, das die Endpunktkonfiguration für den SINGLE_MASTER Kanaltyp enthält.

Inhalt

Protocols

Diese Eigenschaft wird verwendet, um die Art der Kommunikation über diesen SINGLE_MASTER Signalkanal zu bestimmen. Wenn angegeben, WSS gibt diese API einen WebSocket-Endpunkt zurück. Wenn angegeben, HTTPS gibt diese API einen HTTPS Endpunkt zurück.

Typ: Zeichenfolge-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 5 Elemente.

Zulässige Werte: WSS | HTTPS | WEBRTC

Required: No

Role

Diese Eigenschaft wird verwendet, um die Nachrichtenberechtigungen in diesem SINGLE_MASTER Signalkanal zu bestimmen. Wenn angegeben, MASTER gibt diese API einen Endpunkt zurück, den ein Kunde verwenden kann, um Angebote von allen Zuschauern auf diesem Signalkanal zu empfangen und Antworten an diese zu senden. Wenn angegeben, VIEWER gibt diese API einen Endpunkt zurück, den ein Client nur verwenden kann, um Angebote an einen anderen MASTER Client auf diesem Signalkanal zu senden.

Typ: Zeichenfolge

Zulässige Werte: MASTER | VIEWER

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)

- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

SingleMasterConfiguration

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Konfiguration des SINGLE_MASTER Kanalfensters enthält.

Inhalt

MessageTtlSeconds

Der Zeitraum, in dem ein Signalisierungskanal nicht zugestellte Nachrichten aufbewahrt, bevor sie verworfen werden.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 5. Maximaler Wert.

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

StreamInfo

Bedienung: Amazon Kinesis Video Streams

Ein Kinesis-Videodatenstrom.

Inhalt

CreationTime

Ein Zeitstempel, der angibt, wann der Stream erstellt wurde.

Typ: Zeitstempel

Required: No

DataRetentionInHours

Wie lange der Stream Daten speichert, in Stunden.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Required: No

DeviceName

Der Name des Geräts, das dem Stream zugeordnet ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Nein

KmsKeyId

Die ID des AWS Key Management Service (AWS KMS)-Schlüssels, die Kinesis Video Streams zum Verschlüsseln von Daten im Stream verwendet.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 2048 Zeichen.

Pattern: . +

Erforderlich: Nein

MediaType

Der MediaType des Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w\-\.\.]+/[\w\-\.\.]+(, [\w\-\.\.]+/[\w\-\.\.]+)*`

Erforderlich: Nein

Status

Der Status des Datenstroms.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | UPDATING | DELETING`

Required: No

StreamARN

Der Amazon-Ressourcenname (ARN) des Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 1024 Zeichen.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Nein

StreamName

Name des Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Nein

Version

Die Version des Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 64 Zeichen.

Pattern: [a-zA-Z0-9]+

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

StreamNameCondition

Bedienung: Amazon Kinesis Video Streams

Gibt die Bedingung an, die Streams erfüllen müssen, damit sie zurückgegeben werden, wenn Sie Streams auflisten (siehe `ListStreams` API). Eine Bedingung hat eine Vergleichsoperation und einen Wert. Derzeit können Sie nur den `BEGINS_WITH` Operator angeben, der Streams findet, deren Namen mit einem bestimmten Präfix beginnen.

Inhalt

ComparisonOperator

Ein Vergleichsoperator. Derzeit können Sie nur den `BEGINS_WITH` Operator angeben, der Streams findet, deren Namen mit einem bestimmten Präfix beginnen.

Typ: Zeichenfolge

Zulässige Werte: `BEGINS_WITH`

Required: No

ComparisonValue

Ein Wert zum Vergleich.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)

- [AWS SDK für Ruby V3](#)

Tag

Bedienung: Amazon Kinesis Video Streams

Ein Schlüssel/Wert-Paar, das dem angegebenen Signalisierungskanal zugeordnet ist.

Inhalt

Key

Der Schlüssel des Tags, das dem angegebenen Signalisierungskanal zugeordnet ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *)$`

Erforderlich: Ja

Value

Der Wert des Tags, das dem angegebenen Signalisierungskanal zugeordnet ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 0. Maximale Länge beträgt 256 Zeichen.

Pattern: `[\p{L}\p{Z}\p{N}_ . : / = + \ - @] *`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

UploaderConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfiguration, die aus dem `ScheduleExpression` und den `DurationInMinutes` Details besteht, die den Zeitplan für die Aufnahme von einer Kamera oder einer lokalen Mediendatei auf den Edge Agent festlegen. Wenn der nicht in der angegebenen `ScheduleConfig` ist `UploaderConfig`, lädt der Edge-Agent in regelmäßigen Abständen (alle 1 Stunde) hoch.

Inhalt

ScheduleConfig

Die Konfiguration, die aus dem `ScheduleExpression` und den `DurationInMinutes` Details besteht, die den Zeitplan für die Aufnahme von einer Kamera oder einer lokalen Mediendatei auf den Edge Agent festlegen. Wenn dies nicht vorgesehen `ScheduleConfig` ist `UploaderConfig`, lädt der Edge-Agent in regelmäßigen Abständen (alle 1 Stunde) hoch.

Typ: [ScheduleConfig](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Medien Amazon Kinesis Video Streams

Die folgenden Datentypen werden von Amazon Kinesis Video Streams Media unterstützt:

- [StartSelector](#)

StartSelector

Bedienung: Amazon Kinesis Video Streams Media

Identifiziert den Abschnitt im Kinesis-Videostream, in dem die `GetMedia` API mit der Rückgabe von Mediendaten beginnen soll. Sie haben die folgenden Optionen für Ihre Start-Chunk:

- Wählen Sie den neuesten (oder ältesten) Chunk aus.
- Identifizieren Sie einen bestimmten Teil. Sie können einen bestimmten Chunk identifizieren, indem Sie entweder eine Fragmentnummer oder einen Zeitstempel (Server oder Produzent) angeben.
- Die Metadaten jedes Chunks enthalten ein Continuation Token als Matroska (MKV) -Tag (`AWK_KINESISVIDEO_CONTINUATION_TOKEN`). Wenn Ihre vorherige `GetMedia` Anfrage beendet wurde, können Sie diesen Tag-Wert in Ihrer nächsten `GetMedia` Anfrage verwenden. Die API beginnt dann mit der Rückgabe von Chunks und beginnt dort, wo die letzte API endete.

Inhalt

StartSelectorType

Identifiziert das Fragment im Kinesis-Videostream, von dem Sie mit dem Abrufen der Daten beginnen möchten.

- `JETZT` — Beginne mit dem neuesten Teil des Streams.
- `FRÜHESTENS` — Beginne mit dem frühesten verfügbaren Block im Stream.
- `FRAGMENT_NUMBER` — Beginne mit dem Chunk nach einem bestimmten Fragment. Sie müssen außerdem den Parameter `AfterFragmentNumber` angeben.
- `PRODUCER_TIMESTAMP` oder `SERVER_TIMESTAMP` — Beginnt mit dem Chunk, der ein Fragment mit dem angegebenen Producer- oder Server-Zeitstempel enthält. Sie geben den Zeitstempel an, indem Sie hinzufügen `StartTimeStamp`.
- `CONTINUATION_TOKEN` — Lesen Sie mit dem angegebenen Fortführungstoken.

Note

Wenn Sie `NOW`, `ERIEEST` oder `CONTINUATION_TOKEN` als das `startSelectorType` wählen, geben Sie keine zusätzlichen Informationen in der `startSelector` an.

Typ: Zeichenfolge

Zulässige Werte: FRAGMENT_NUMBER | SERVER_TIMESTAMP | PRODUCER_TIMESTAMP | NOW | EARLIEST | CONTINUATION_TOKEN

Erforderlich: Ja

AfterFragmentNumber

Gibt die Fragmentnummer an, von der aus die GetMedia API mit der Rückgabe der Fragmente beginnen soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[0-9]+$`

Erforderlich: Nein

ContinuationToken

Fortführungstoken, das Kinesis Video Streams in der vorherigen GetMedia Antwort zurückgegeben hat. Die GetMedia API beginnt dann mit dem Chunk, der durch das Continuation Token identifiziert wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[a-zA-Z0-9_\.\-]+$`

Erforderlich: Nein

StartTimeStamp

Ein Zeitstempelwert. Dieser Wert ist erforderlich, wenn Sie PRODUCER_TIMESTAMP oder SERVER_TIMESTAMP als den wählen. startSelectorType Die GetMedia API beginnt dann mit dem Chunk, der das Fragment enthält, das den angegebenen Zeitstempel hat.

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Amazon Kinesis Video Streams Streams-Archivierte Medien

Die folgenden Datentypen werden von Amazon Kinesis Video Streams Archived Media unterstützt:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

ClipFragmentSelector

Bedienung: Amazon Kinesis Video Streams Archived Media

Beschreibt den Zeitstempelbereich und den Zeitstempelursprung einer Reihe von Fragmenten.

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das bedeutet, dass, wenn Produzenten einen Strom von Fragmenten mit Produzenten-Zeitstempeln produzieren, die ungefähr der wahren Uhrzeit entsprechen, der Clip alle Fragmente innerhalb des angeforderten Zeitstempelbereichs enthält. Wenn einige Fragmente innerhalb derselben Zeitspanne und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

Inhalt

FragmentSelectorType

Die Herkunft der zu verwendenden Zeitstempel (Server oder Producer).

Typ: Zeichenfolge

Zulässige Werte: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

Erforderlich: Ja

TimestampRange

Der Bereich der zurückzugebenden Zeitstempel.

Typ: [ClipTimestampRange](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

ClipTimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Inhalt

EndTimeStamp

Das Ende des Zeitstempelbereichs für das angeforderte Medium.

Dieser Wert muss innerhalb von 24 Stunden nach dem angegebenen `StartTimeStamp` Wert liegen und er muss nach dem angegebenen `StartTimeStamp` Wert liegen.

Wenn `FragmentSelectorType` für die Anfrage gilt `SERVER_TIMESTAMP`, muss dieser Wert in der Vergangenheit liegen.

Der angegebene Bereich versteht sich inklusive der angegebenen Werte. Der `EndTimeStamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimeStamp` Wert beginnen und über diesen hinausgehen, sind in der Sitzung enthalten.

Typ: Zeitstempel

Erforderlich: Ja

StartTimeStamp

Der Startzeitstempel im Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Nur Fragmente, die genau am oder danach beginnen, `StartTimeStamp` sind in der Sitzung enthalten. Fragmente, die davor beginnen `StartTimeStamp` und darüber hinaus andauern, sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` `jaSERVER_TIMESTAMP`, `StartTimeStamp` muss der später als der Stream Head sein.

Typ: Zeitstempel

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DASHFragmentSelector

Bedienung: Amazon Kinesis Video Streams Archived Media

Enthält den Bereich der Zeitstempel für die angeforderten Medien und die Quelle der Zeitstempel.

Inhalt

FragmentSelectorType

Die Quelle der Zeitstempel für die angeforderten Medien.

Wenn `FragmentSelectorType` ist auf gesetzt `PRODUCER_TIMESTAMP` und `getDashStreamingSession URL: PlaybackMode` ist `ON_DEMAND` oder `LIVE_REPLAY`, wird das erste Fragment, das mit einem Producer-Zeitstempel innerhalb des angegebenen `FragmentSelector` Zeitstempels aufgenommen wurde, in die `TimestampRange` Medien-Playlist aufgenommen. Zusätzlich sind die Fragmente mit Producer-Zeitstempeln innerhalb des `TimestampRange` unmittelbar auf das erste Fragment folgenden Ingested (bis zum `StreamingSessionMaxManifestFragmentResultsGetDash-URL-Wert`) enthalten.

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das bedeutet, dass, wenn Produzenten einen Strom von Fragmenten mit Herstellerzeitstempeln produzieren, die ungefähr der wahren Uhrzeit entsprechen, das MPEG-DASH-Manifest alle Fragmente innerhalb des angeforderten Zeitstempelbereichs enthält. Wenn einige Fragmente innerhalb derselben Zeitspanne und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

When `FragmentSelectorType` ist auf gesetzt `PRODUCER_TIMESTAMP` und `getDashStreamingSession URL: PlaybackMode` is `LIVE`, werden die Timestamps des Produzenten in den MP4-Fragmenten und zur Deduplizierung verwendet. Die zuletzt aufgenommenen Fragmente, die auf Serverzeitstempeln basieren, sind jedoch im MPEG-DASH-Manifest enthalten. Das bedeutet, dass selbst wenn Fragmente, die in der Vergangenheit aufgenommen wurden, einen Producer-Zeitstempel mit aktuellen Werten haben, sie nicht in der HLS-Medien-Playlist enthalten sind.

Der Standardwert ist `SERVER_TIMESTAMP`.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Required: No

TimestampRange

Der Beginn und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Dieser Wert sollte nicht vorhanden sein, wenn PlaybackType jaLIVE.

Typ: [DASHTimestampRange](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

DASHTimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Beginn und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Wenn ja, sollte dieser Wert nicht `PLAYBACK_TYPE_LIVE` sein.

Die Werte in `DASHTimestampRange` sind inklusiv. Fragmente, die genau an oder nach der Startzeit beginnen, sind in der Sitzung enthalten. Fragmente, die vor der Startzeit beginnen und über diese hinaus andauern, sind nicht in der Sitzung enthalten.

Inhalt

EndTimeStamp

Das Ende des Zeitstempelbereichs für das angeforderte Medium. Dieser Wert muss innerhalb von 24 Stunden nach dem angegebenen `StartTimeStamp` Wert liegen und er muss nach dem angegebenen `StartTimeStamp` Wert liegen.

Wenn `FragmentSelectorType` für die Anfrage gilt `SERVER_TIMESTAMP`, muss dieser Wert in der Vergangenheit liegen.

Der `EndTimeStamp` Wert ist für den `ON_DEMAND` Modus erforderlich, für den `LIVE_REPLAY` Modus jedoch optional. Wenn der nicht für den `LIVE_REPLAY` Modus gesetzt `EndTimeStamp` ist, enthält die Sitzung weiterhin neu aufgenommene Fragmente, bis die Sitzung abläuft.

Note

Der angegebene Bereich versteht sich inklusive der angegebenen Werte. Der `EndTimeStamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimeStamp` Wert beginnen und über diesen hinausgehen, sind in der Sitzung enthalten.

Typ: Zeitstempel

Required: No

StartTimeStamp

Der Beginn des Zeitstempelbereichs für das angeforderte Medium.

Wenn der `DASHTimestampRange` Wert angegeben ist, ist der `StartTimeStamp` Wert erforderlich.

Nur Fragmente, die genau am oder danach beginnen, `StartTimeStamp` sind in der Sitzung enthalten. Fragmente, die davor beginnen `StartTimeStamp` und darüber hinaus andauern, sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` `jaSERVER_TIMESTAMP`, `StartTimeStamp` muss der später als der Stream Head sein.

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Fragment

Service: Amazon Kinesis Video Streams Archived Media

Stellt ein Videosegment oder andere zeitlich begrenzte Daten dar.

Inhalt

FragmentLengthInMilliseconds

Die Wiedergabedauer oder ein anderer Zeitwert, der dem Fragment zugeordnet ist.

Type: Long

Erforderlich: Nein

FragmentNumber

Die eindeutige Kennung des Fragments. Dieser Wert nimmt je nach Reihenfolge der Aufnahme monoton zu.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[0-9]+$`

Erforderlich: Nein

FragmentSizeInBytes

Die Gesamtgröße des Fragments, einschließlich Informationen über das Fragment und die enthaltenen Mediendaten.

Type: Long

Erforderlich: Nein

ProducerTimestamp

Der Zeitstempel des Produzenten, der dem Fragment entspricht, in Millisekunden.

Typ: Zeitstempel

Erforderlich: Nein

ServerTimestamp

Der Zeitstempel des AWS Servers, der dem Fragment entspricht, in Millisekunden.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS-SDK für Ruby V3](#)

FragmentSelector

Bedienung: Amazon Kinesis Video Streams Archived Media

Beschreibt den Zeitstempelbereich und den Zeitstempelursprung einer Reihe von Fragmenten.

Es werden nur Fragmente mit einem Startzeitstempel größer als oder gleich der angegebenen Startzeit und kleiner oder genauso groß wie die letzte Zeit sein. Zum Beispiel, wenn ein Stream Fragmente mit den folgenden Startzeitstempeln enthält:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

Ein Fragmentauswahlbereich mit einer Startzeit von 00:00:01 und einer Endzeit von 00:00:04 würde die Fragmente mit den Startzeiten 00:00:02 und 00:00:04 zurückgeben.

Inhalt

FragmentSelectorType

Die Herkunft der zu verwendenden Zeitstempel (Server oder Producer).

Typ: Zeichenfolge

Zulässige Werte: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

Erforderlich: Ja

TimestampRange

Der Bereich der zurückzugebenden Zeitstempel.

Typ: [TimestampRange](#) Objekt

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HLSFragmentSelector

Bedienung: Amazon Kinesis Video Streams Archived Media

Enthält den Bereich der Zeitstempel für die angeforderten Medien und die Quelle der Zeitstempel.

Inhalt

FragmentSelectorType

Die Quelle der Zeitstempel für die angeforderten Medien.

When `FragmentSelectorType` ist auf gesetzt `PRODUCER_TIMESTAMP` und [GetHLS StreamingSession URL: PlaybackMode](#) ist `ON_DEMAND` oder `LIVE_REPLAY`, wird das erste Fragment, das mit einem Producer-Zeitstempel innerhalb des angegebenen Zeitstempels aufgenommen wurde, in die [FragmentSelectorTimestampRangeMedien-Playlist](#) aufgenommen. Zusätzlich sind die Fragmente mit Producer-Zeitstempeln innerhalb des `TimestampRange` unmittelbar auf das erste Fragment folgenden [Ingested \(bis zur StreamingSession GetHLS-URL: MaxMediaPlaylistFragmentResults](#) value) enthalten.

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das bedeutet, dass, wenn Produzenten einen Strom von Fragmenten mit Herstellerzeitstempeln produzieren, die ungefähr der wahren Uhrzeit entsprechen, die HLS-Medien-Playlists alle Fragmente innerhalb des angeforderten Zeitstempelbereichs enthalten. Wenn einige Fragmente innerhalb derselben Zeitspanne und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

When `FragmentSelectorType` ist auf gesetzt `PRODUCER_TIMESTAMP` und [GetHLS StreamingSession URL: PlaybackMode](#) is `LIVE`, werden die Timestamps des Produzenten in den MP4-Fragmenten und zur Deduplizierung verwendet. Die zuletzt aufgenommenen Fragmente, die auf Serverzeitstempeln basieren, sind jedoch in der HLS-Medien-Playlist enthalten. Das bedeutet, dass selbst wenn Fragmente, die in der Vergangenheit aufgenommen wurden, einen Producer-Zeitstempel mit aktuellen Werten haben, sie nicht in der HLS-Medien-Playlist enthalten sind.

Der Standardwert ist `SERVER_TIMESTAMP`.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Required: No

TimestampRange

Der Beginn und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Dieser Wert sollte nicht vorhanden sein, wenn PlaybackType jaLIVE.

Typ: [HLSTimestampRange](#) Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

HLSTimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Beginn und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Wenn ja, sollte dieser Wert nicht `PLAYBACK_TYPE_LIVE` sein.

Inhalt

EndTimeStamp

Das Ende des Zeitstempelbereichs für das angeforderte Medium. Dieser Wert muss innerhalb von 24 Stunden nach dem angegebenen `StartTimeStamp` Wert liegen und er muss nach dem angegebenen `StartTimeStamp` Wert liegen.

Wenn `FragmentSelectorType` für die Anfrage gilt `SERVER_TIMESTAMP`, muss dieser Wert in der Vergangenheit liegen.

Der `EndTimeStamp` Wert ist für den `ON_DEMAND` Modus erforderlich, für den `LIVE_REPLAY` Modus jedoch optional. Wenn der nicht für den `LIVE_REPLAY` Modus gesetzt `EndTimeStamp` ist, enthält die Sitzung weiterhin neu aufgenommene Fragmente, bis die Sitzung abläuft.

Note

Der angegebene Bereich versteht sich inklusive der angegebenen Werte. Der `EndTimeStamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimeStamp` Wert beginnen und über diesen hinausgehen, sind in der Sitzung enthalten.

Typ: Zeitstempel

Required: No

StartTimeStamp

Der Beginn des Zeitstempelbereichs für das angeforderte Medium.

Wenn der `HLSTimestampRange` Wert angegeben ist, ist der `StartTimeStamp` Wert erforderlich.

Nur Fragmente, die genau am oder danach beginnen, `StartTimeStamp` sind in der Sitzung enthalten. Fragmente, die davor beginnen `StartTimeStamp` und darüber

hinaus andauern, sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` `jaSERVER_TIMESTAMP`, `StartTimeStamp` muss der später als der Stream Head sein.

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Image

Bedienung: Amazon Kinesis Video Streams Archived Media

Eine Struktur, die die `TimestampError`, und enthält `ImageContent`.

Inhalt

Error

Die Fehlermeldung, die angezeigt wird, wenn das Bild für den angegebenen Zeitstempel aufgrund eines Fehlers, der nicht ausprobiert werden kann, nicht extrahiert wurde. Ein Fehler wird zurückgegeben, wenn:

- Für das angegebene Objekt gibt es kein `MediumTimestamp`.
- Das Medium für die angegebene Zeit erlaubt es nicht, ein Bild zu extrahieren. In diesem Fall handelt es sich bei dem Medium nur um Audio, oder es wurde das falsche Medium aufgenommen.

Typ: Zeichenfolge

Zulässige Werte: `NO_MEDIA` | `MEDIA_ERROR`

Required: No

ImageContent

Ein Attribut des Image Objekts, das Base64-kodiert ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 1 000 Zeichen.

Required: No

TimeStamp

Ein Attribut des Image Objekts, das verwendet wird, um ein Bild aus dem Videostream zu extrahieren. Dieses Feld wird verwendet, um Lücken in Bildern zu verwalten oder das Paginierungsfenster besser zu verstehen.

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK für Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

TimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Inhalt

EndTimeStamp

Der Endzeitstempel im Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Typ: Zeitstempel

Erforderlich: Ja

StartTimeStamp

Der Startzeitstempel im Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Typ: Zeitstempel

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Amazon Kinesis Video Video Kinesis Video Streams

Die folgenden Datentypen werden von Amazon Kinesis Video Kinesis Video Kinesis Video Streams unterstützt:

- [IceServer](#)

IceServer

Bedienung: Amazon Kinesis Video Signaling Channels

Eine Struktur für die Verbindungsdaten des ICE-Servers.

Inhalt

Password

Ein Passwort für die Anmeldung am ICE-Server.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Nein

Ttl

Die Anzahl der Sekunden, in denen: Die Anzahl der Sekunden, in denen:

Typ: Ganzzahl

Gültiger Bereich: Maximaler Wert von 86400.

Required: No

Uris

Eine Reihe von URIs in der in der [ID angegebenen Form. petithuguenin-behave-turn-uris](#)spezifikation. Diese URIs stellen die verschiedenen Adressen und/oder Protokolle bereit, mit denen der TURN-Server erreicht werden kann.

Typ: Zeichenfolge-Array

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Required: No

Username

Ein Benutzername für die Anmeldung am ICE-Server.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Pattern: [a-zA-Z0-9_.-]+

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- [AWS-SDK für C++](#)
- [AWS-SDK for Go](#)
- [AWS-SDK für Java V2](#)
- [AWS SDK für Ruby V3](#)

Amazon Kinesis Video WebRTC Speicher

Die folgenden Datentypen werden von Amazon Kinesis Video WebRTC unterstützt:

Häufige Fehler

In diesem Abschnitt sind Fehler aufgeführt, die häufig bei den API-Aktionen aller AWS-Services auftreten. Informationen zu Fehlern, die spezifisch für eine API-Aktion für diesen Service sind, finden Sie unter dem Thema für diese API-Aktion.

AccessDeniedException

Sie haben keinen ausreichenden Zugriff zum Durchführen dieser Aktion.

HTTP Status Code: 400

IncompleteSignature

Die Anforderungssignatur entspricht nicht den AWS-Standards.

HTTP Status Code: 400

InternalFailure

Die Anforderungsverarbeitung ist fehlgeschlagen, da ein unbekannter Fehler, eine Ausnahme oder ein Fehler aufgetreten ist.

HTTP Status Code: 500

InvalidAction

Die angeforderte Aktion oder Operation ist ungültig. Überprüfen Sie, ob die Aktion ordnungsgemäß eingegeben wurde.

HTTP Status Code: 400

InvalidClientTokenId

Das angegebene X.509-Zertifikat oder die AWS-Zugriffsschlüssel-ID ist nicht in unseren Datensätzen vorhanden.

HTTP Status Code: 403

NotAuthorized

Sie haben keine Berechtigung zum Ausführen dieser Aktion.

HTTP Status Code: 400

OptInRequired

Die AWS-Zugriffsschlüssel-ID benötigt ein Abonnement für den Service.

HTTP Status Code: 403

RequestExpired

Die Anforderung hat den Service mehr als 15 Minuten nach dem Datumstempel oder mehr als 15 Minuten nach dem Ablaufdatum der Anforderung erreicht (z. B. für vorseignierte URLs) oder der Datumstempel auf der Anforderung liegt mehr als 15 Minuten in der Zukunft.

HTTP Status Code: 400

ServiceUnavailable

Die Anforderung ist aufgrund eines temporären Fehlers des Servers fehlgeschlagen.

HTTP Status Code: 503

ThrottlingException

Die Anforderung wurde aufgrund der Drosselung von Anforderungen abgelehnt.

HTTP Status Code: 400

ValidationError

Die Eingabe erfüllt nicht die von einem AWS-Service definierten Einschränkungen.

HTTP Status Code: 400

Geläufige Parameter

Die folgende Liste enthält die Parameter, die alle Aktionen zum Signieren von Signature-Version-4-Anforderungen mit einer Abfragezeichenfolge verwenden. Alle aktionsspezifischen Parameter werden im Thema für diese Aktion aufgelistet. Weitere Informationen zu Signature Version 4 finden Sie unter [Signieren von AWS API-Anfragen](#) im IAM-Benutzerhandbuch.

Action

Die auszuführende Aktion.

Typ: Zeichenfolge

Erforderlich: Ja

Version

Die API-Version, für die die Anforderung geschrieben wurde, ausgedrückt im Format JJJJ-MM-TT.

Typ: Zeichenfolge

Erforderlich: Ja

X-Amz-Algorithm

Der Hashalgorithmus, den Sie zum Erstellen der Anforderungssignatur verwendet haben.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Zulässige Werte: AWS4-HMAC-SHA256

Required: Conditional

X-Amz-Credential

Der Wert des Anmeldeinformationsumfangs. Dabei handelt es sich um eine Zeichenfolge, die Ihren Zugriffsschlüssel, das Datum, die gewünschte Region und eine Zeichenfolge zur Beendigung („aws4_request“) beinhaltet. Der Wert wird im folgenden Format ausgedrückt: Zugriffsschlüssel/JJJJMMTT/Region/Service/aws4_request.

Weitere Informationen finden Sie unter [Erstellen einer signierten AWS API-Anfrage](#) im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Date

Das Datum, das zum Erstellen der Signatur verwendet wird. Das Format muss das ISO 8601-Basisformat (JJJJMMTT'T'SSMSS'Z') sein. Die folgende Datumszeit ist beispielsweise ein gültiger X-Amz-Date-Wert: 20120325T120000Z.

Bedingung: X-Amz-Date ist bei allen Anforderungen optional. Damit kann das Datum überschrieben werden, das zum Signieren von Anforderungen verwendet wird. Wenn der Date-Header im ISO 8601-Basisformat angegeben ist, ist X-Amz-Date nicht erforderlich. Wenn X-Amz-Date verwendet wird, überschreibt es immer den Wert des Date-Headers. Weitere Informationen finden Sie unter [Elemente einer AWS API-Anforderungssignatur](#) im IAM-Benutzerhandbuch.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Security-Token

Das temporäre Sicherheitstoken, das durch einen Anruf von AWS Security Token Service (AWS STS) abgerufen wurde. Eine Liste der Services, die temporäre Sicherheitsanmeldeinformationen von unterstützen AWS STS [AWS-Services, finden Sie unter, die mit IAM arbeiten](#) im IAM-Benutzerhandbuch.

Bedingung: Wenn Sie temporäre Sicherheitsanmeldeinformationen von nutzenAWS STS, müssen Sie das Sicherheitstoken einschließen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Signature

Gibt die hex-codierte Signatur an, die aus der zu signierenden Zeichenfolge und dem abgeleiteten Signaturschlüssel berechnet wurde.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-SignedHeaders

Gibt alle HTTP-Header an, die als Teil der kanonischen Anforderung enthalten waren. Weitere Informationen zur Angabe signierter Header finden Sie unter [Erstellen einer signiertenAWS API-Anfrage](#) im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

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.