aws

Entwicklerhandbuch

Amazon Kinesis Video Streams

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

Amazon Kinesis Video Streams: Entwicklerhandbuch

Copyright © 2025 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 Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Kinesis Video Streams? ... 1
Verfügbarkeit in Regionen ... 2
Funktionsweise ... 3
API- und Produzentenbibliotheken ... 4
Datenmodell ... 8
Systemanforderungen ... 17
Anforderungen an die Kamera ... 18
Getestete Betriebssysteme ... 19
SDK-Speicheranforderungen ... 19
Kontingente ... 19
API-Dienstkontingente auf Kontrollebene ... 19
API-Dienstkontingente für Medien und archivierte Medien ... 23
Fragment-Metadaten und Kontingente für fragmentierte Medien ... 29
Kontingente für Streaming-Metadaten ... 32
SDK-Kontingente für Produzenten ... 32
Einrichten eines -Kontos ... 36
Melden Sie sich an für ein AWS-Konto ... 37
Erstellen eines Benutzers mit Administratorzugriff ... 37
Erstellen Sie einen Schlüssel AWS-Konto ... 39
Vorteile ... 39
Erste Schritte ... 41
Erstellen Sie einen Amazon Kinesis Kinesis-Videostream ... 41
Erstellen Sie mit der Konsole einen Videostream ... 42
Erstellen Sie einen Videostream mit dem AWS CLI ... 42
Daten an einen Amazon Kinesis Kinesis-Videostream senden ... 42
Erstellen Sie das SDK und die Beispiele ... 43
Führen Sie die Beispiele aus, um Medien auf Kinesis Video Streams hochzuladen ... 47
Überprüfen Sie die Bestätigungsobjekte ... 49
Mediendaten konsumieren ... 49
Medien in der Konsole anzeigen ... 49
Konsumieren Sie Mediendaten mit HLS ... 49
Zu Kinesis Video Streams hochladen ... 50
Produzent/Kunde von Kinesis Video Streams ... 50
Produzentenbibliothek für Kinesis Video Streams ... 51

Verstehen Sie, was Producer-Bibliotheken sind ... 52
Java ... 52
Verfahren: Verwenden Sie das Java Producer SDK ... 53
Voraussetzungen ... 53
Laden Sie den Code herunter und konfigurieren Sie ihn ... 54
Schreiben und untersuchen Sie den Code ... 55
Bereinigen von -Ressourcen ... 57
Führen Sie den Code aus und überprüfen Sie ihn ... 57
Android ... 58
Verfahren: Verwenden Sie das Android Producer SDK ... 59
Voraussetzungen ... 59
Laden Sie den Code herunter und konfigurieren Sie ihn ... 63
Untersuchen Sie den Code ... 64
Führen Sie den Code aus und überprüfen Sie ihn ... 66
C++ ... 68
Objektmodell ... 68
Fügen Sie Medien in den Stream ein ... 68
Callback-Schnittstellen ... 69
Verfahren: Verwenden Sie das C++-Producer-SDK ... 69
Voraussetzungen ... 69
Laden Sie den Code herunter und konfigurieren Sie ihn ... 71
Schreiben und untersuchen Sie den Code ... 76
Führen Sie den Code aus und überprüfen Sie ihn ... 83
Verwenden Sie das SDK als GStreamer Plugin ... 84
Verwenden Sie das C++-Producer-SDK als GStreamer Plugin in einem Docker-Container ... 84
Verwenden Sie die Protokollierung ... 84
C-Produzent ... 85
Objektmodell ... 85
Fügt Medien in den Stream ein ... 86
Verfahren: Verwenden Sie das C-Producer-SDK ... 87
Voraussetzungen ... 87
Laden Sie den Code herunter ... 88
Schreiben und untersuchen Sie den Code ... 89
Führen Sie den Code aus und überprüfen Sie ihn ... 91
C++ auf Raspberry Pi ... 93
Voraussetzungen ... 94

Erstellen eines Benutzers 94
Treten Sie Ihrem Netzwerk bei 96
Verbindung per Fernzugriff herstellen 97
Konfiguriere die Kamera 97
Installieren Sie die Softwarevoraussetzungen 100
Laden Sie das C++-Producer-SDK herunter und erstellen Sie es 101
Streamen Sie einen Livestream 104
Medien abspielen 113
Fehlerbehebung 116
Fehlercodereferenz 121
Von PutFrame Callbacks — Platform Independent Code (PIC) zurückgegebene Fehler und Statuscodes 121
Von PutFrame Callbacks zurückgegebene Fehler und Statuscodes — C-Producer-Bibliothek 176
NAL-Anpassungsflags 183
Produzentenstrukturen 185
DeviceInfo/DefaultDeviceInfoProvider 185
StorageInfo 185
Stream-Strukturen 187
StreamDefinition/StreamInfo 187
ClientMetrics 207
StreamMetrics 208
Rückrufe 210
ClientCallbackProvider 211
StreamCallbackProvider 211
ClientCallbacks 212
Callback-Implementierungen, um das Streaming erneut zu versuchen 217
Streaming-Metadaten verwenden 218
Hinzufügen von Metadaten zu einem Kinesis-Videostream 219
Videowiedergabe 222
Anforderungen für die Wiedergabe 223
GetClip 223
Holen Sie sich die SessionURL DASHStreaming 224
Holen Sie sich die SessionURL HLSStreaming 225
GetImages 227
Wiedergabe mit HLS 227

Verwenden Sie die AWS CLI , um die URL einer HLS-Streaming-Sitzung abzurufen 228
Beispiel: Verwenden Sie HLS in HTML und JavaScript 232
Fehlerbehebung bei HLS-Problemen 236
Wiedergabe mit MPEG-DASH 239
Beispiel: Verwendung von MPEG-DASH in HTML und JavaScript 239
Richten Sie Benachrichtigungen ein 244
Benachrichtigungskonfigurationen verwalten 244
UpdateNotificationConfiguration 244
DescribeNotificationConfiguration 245
Über MKV-Tags des Herstellers 245
Syntax für MKV-Tags von Produzenten 245
Beschränkungen für MKV-Tags 246
Amazon SNS SNS-Nachrichten 246
Nutzlast für Amazon SNS-Themen 246
Ihre Amazon SNS SNS-Nachrichten anzeigen 247
Extrahieren Sie Bilder aus Videostreams 249
Automatisierte Bildgenerierung (Amazon S3 S3-Lieferung) 249
UpdateImageGenerationConfiguration 250
DescribeImageGenerationConfiguration 252
MKV-Tags für Hersteller 245
Fügen Sie Metadaten-Tags im Producer-SDK hinzu mit PutEventMetaData 254
Einschränkungen 254
S3-Objekt-Metadaten 254
Amazon S3 S3-Objektpfad (Bild) 255
Amazon S3 S3-URI-Empfehlungen zum Schutz vor Drosselung 255
Greifen Sie auf Videoanalysen zu 257
Metadaten konsumieren 257
Streamen Sie mithilfe der Parser-Bibliothek 258
Voraussetzungen 259
Laden Sie den Code herunter 259
Untersuchen Sie den Code 260
Führen Sie den Code aus 267
Überwachen 267
Überwachen Sie Metriken mit CloudWatch 268
Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch 288
API-Aufrufe protokollieren mit CloudTrail 294

Beschränkungen für Streaming-Metadaten ... 299
Planen Sie die Videoaufnahme und Speicherung ... 300
API-Operationen für Amazon Kinesis Video Streams Edge Agent ... 301
Überwachen des Amazon Kinesis Video Streams Edge Agent ... 301
Im AWS IoT Greengrass Nicht-Modus bereitstellen ... 301
Abhängigkeiten installieren ... 302
Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs ... 304
Erstellen Sie eine IAM-Berechtigungsrichtlinie ... 306
Erstellen einer IAM-Rolle ... 308
Erstellen Sie den AWS IoT Rollenalias ... 310
Erstellen Sie die AWS IoT Richtlinie ... 310
Erstelle ein AWS IoT Ding und hol dir AWS IoT Core Anmeldeinformationen ... 312
Erstellen Sie den Edge-Agenten ... 315
Installieren Sie den CloudWatch Agenten ... 326
Führen Sie den Edge-Agent als nativen Prozess aus ... 329
Bereitstellen auf AWS IoT Greengrass ... 332
Erstellen Sie eine Ubuntu-Instanz ... 332
Richten Sie das AWS IoT Greengrass Kerngerät ein ... 334
Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs ... 335
Fügen Sie der TES-Rolle Berechtigungen hinzu ... 337
Installieren Sie die Secret Manager-Komponente ... 340
Stellen Sie den Edge-Agenten auf dem Gerät bereit ... 343
Installieren Sie die AWS IoT Greengrass Log Manager-Komponente ... 353
Häufig gestellte Fragen ... 357
Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent? ... 357
Unterstützt der Amazon Kinesis Video Streams Edge Agent H.265-Medien? ... 357
Funktioniert der Amazon Kinesis Video Streams Edge Agent in AL2? ... 357
Wie kann ich mehrere Streams innerhalb des Objekts oder AWS IoT Geräts ausführen? ... 358
Wie kann ich eine bearbeiten, `StartEdgeConfigurationUpdate` nachdem sie gesendet wurde? ... 358
Haben Sie Beispiele für Gemeinsamkeiten`ScheduleConfigs`? ... 358
Gibt es ein maximales Stream-Limit? ... 359
Wie starte ich einen Job neu, bei dem ein Fehler aufgetreten ist? ... 359
Wie überwache ich den Zustand meines Amazon Kinesis Video Streams Edge Agent? ... 360
Video über eine VPC streamen ... 361
Zusätzliche Informationen ... 361

VPC-Endpunktverfahren ... 362
Beispiele ... 364
Beispiele: Daten an Kinesis Video Streams senden ... 364
Beispiele: Daten aus Kinesis Video Streams abrufen ... 364
Beispiele: Wiedergeben von Videodaten ... 364
Voraussetzungen ... 364
GStreamer Plug-in — kvssink ... 365
Laden Sie das GStreamer Element herunter, erstellen Sie es und konfigurieren Sie es ... 366
Führen Sie das Element aus GStreamer ... 367
Startbefehle ... 367
Führen Sie das GStreamer Element in einem Docker-Container aus ... 370
Parameter-Referenz ... 372
PutMedia API ... 388
Laden Sie den Code herunter und konfigurieren Sie ihn ... 389
Schreiben und untersuchen Sie den Code ... 390
Führen Sie den Code aus und überprüfen Sie ihn ... 392
RTSP und Docker ... 393
Video-Tutorials ... 394
Voraussetzungen ... 394
Erstellen Sie das Docker-Image ... 394
Führen Sie die RTSP-Beispielanwendung aus ... 395
Renderer ... 396
Voraussetzungen ... 397
Renderer-Beispiel ausführen ... 397
So funktioniert's ... 398
API-Referenz ... 400
Aktionen ... 400
Amazon Kinesis Video Streams ... 402
Amazon Kinesis Video Streams Streams-Medien ... 530
Amazon Kinesis Video Streams — Archivierte Medien ... 548
Amazon Kinesis Kinesis-Videosignalkanäle ... 600
Amazon Kinesis Video WebRTC-Speicher ... 609
Datentypen ... 619
Amazon Kinesis Video Streams ... 620
Amazon Kinesis Video Streams Streams-Medien ... 663
Amazon Kinesis Video Streams — Archivierte Medien ... 666

Amazon Kinesis Kinesis-Videosignalkanäle ... 684
Amazon Kinesis Video WebRTC-Speicher ... 686
Häufige Fehler ... 686
Geläufige Parameter ... 688
Sicherheit ... 691
Datenschutz ... 692
Was ist serverseitige Verschlüsselung für Kinesis Video Streams? ... 692
Überlegungen zu Kosten, Regionen und Leistung ... 692
Wie fange ich mit der serverseitigen Verschlüsselung an? ... 693
Einen vom Kunden verwalteten Schlüssel erstellen und verwenden ... 694
Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels ... 695
Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe von IAM ... 697
Richtliniensyntax ... 697
Aktionen für Kinesis Video Streams ... 698
Amazon-Ressourcennamen (ARNs) für Kinesis Video Streams ... 699
Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren ... 699
Beispielrichtlinien ... 703
Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe AWS IoT ... 705
AWS IoT ThingName als Streamname ... 706
AWS IoT CertificateId als Streamname ... 713
Verwenden Sie AWS IoT Anmeldeinformationen, um zu einem fest codierten Stream-Namen zu streamen ... 714
Compliance-Validierung ... 716
Ausfallsicherheit ... 717
Sicherheit der Infrastruktur ... 717
Bewährte Methoden für die Sicherheit ... 718
Implementieren des Zugriffs mit geringsten Berechtigungen ... 718
Verwenden von IAM-Rollen ... 718
Wird CloudTrail zur Überwachung von API-Aufrufen verwendet ... 719
Fehlerbehebung ... 720
Allgemeine Probleme ... 720
Latenz zu hoch ... 720
Probleme mit der API ... 721
Fehler: „Unbekannte Optionen“ ... 721
Fehler: „Unable to determine service/operation name to be authorized“ (Zu autorisierender Service-/Operationsname kann nicht ermittelt werden) ... 721

Fehler: „Failed to put a frame in the stream“ (Es konnte kein Frame in den Stream eingefügt werden) ... 722
Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“ ... 722
Fehler: "STATUS_STORE_OUT_OF_MEMORY" ... 723
Fehler: „Die Anmeldeinformationen sollten auf eine gültige Region beschränkt sein.“ ... 723
HLS-Probleme ... 723
Probleme mit Java ... 723
Aktivieren von Java-Protokollen ... 724
Probleme mit der Producer-Bibliothek ... 724
Das Producer-SDK kann nicht kompiliert werden ... 725
Der Videostream wird nicht in der Konsole angezeigt ... 725
Fehler: „Das in der Anfrage enthaltene Sicherheitstoken ist ungültig“ beim Streamen von Daten mit der Demo-Anwendung GStreamer ... 726
Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“ ... 726
GStreamer Die Anwendung wird unter OS X mit der Meldung „Streaming gestoppt, Grund nicht ausgehandelt“ beendet ... 727
Fehler: „Heap konnte nicht zugewiesen werden“ beim Erstellen des Kinesis Video Client in der GStreamer Demo auf dem Raspberry Pi ... 727
Fehler: „Ungültiger Befehl“ beim Ausführen der GStreamer Demo auf dem Raspberry Pi ... 728
Die Kamera kann nicht auf Raspberry Pi geladen werden. ... 728
Kamera kann auf macOS High Sierra nicht gefunden werden. ... 729
Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden ... 729
Curl-Fehler beim Ausführen der Demo-Anwendung GStreamer ... 729
Zeitstempel-/Zeitraum-Assertion während der Laufzeit auf Raspberry Pi ... 729
Assertion auf gst_value_set_fraction_range_full auf Raspberry Pi ... 729
STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d)-Fehler unter Android ... 729
Fehler: Die maximale Fragmentdauer wurde erreicht ... 730
"Ungültiger Objektname abgelaufen", Fehler bei der Verwendung der IoT-Autorisierung ... 731
Probleme mit der Stream-Parser-Bibliothek ... 731
Es kann nicht auf einen einzelnen Frame aus dem Stream zugegriffen werden ... 731
Fragmentdekodierungsfehler ... 731
Probleme mit dem Netzwerk ... 732
Dokumentverlauf ... 733
... dccxxxviii

Was ist Amazon 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 Consoleüberwachen 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 Plug-in — kvssink 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 indexiert 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 EC2 Amazon-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.

Verfügbarkeit in Regionen

Amazon Kinesis Video Streams ist in den folgenden Regionen verfügbar:

Name der Region	AWS Regionalcode
USA Ost (Ohio)	us-east-2
USA Ost (Nord-Virginia)	us-east-1
USA West (Oregon)	us-west-2
AWS GovCloud (US-Ost)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1
Afrika (Kapstadt)	af-south-1
Asien-Pazifik (Hongkong)	ap-east-1
Asien-Pazifik (Mumbai)	ap-south-1
Asien-Pazifik (Seoul)	ap-northeast-2
Asien-Pazifik (Singapur)	ap-southeast-1
Asien-Pazifik (Sydney)	ap-southeast-2
Asien-Pazifik (Tokio)	ap-northeast-1
Kanada (Zentral)	ca-central-1
China (Beijing)	cn-north-1
Europa (Frankfurt)	eu-central-1
Europa (Irland)	eu-west-1
Europa (London)	eu-west-2
Europa (Paris)	eu-west-3

Name der Region	AWS Regionalcode
Südamerika (São Paulo)	sa-east-1

Kinesis Video Streams: So funktioniert's

Themen

- Unterstützung für Kinesis Video Streams API und Producer-Bibliotheken
- 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 bietet 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 überträgt. Ein Produzent kann ein beliebiges Videogerät sein, z. B. eine Sicherheitskamera, eine am Körper getragene Kamera, eine Smartphone-Kamera oder eine Armaturenbrettkamera. 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 Streams-Produzentenbibliotheken — Eine Reihe von Software und Bibliotheken, die Sie auf Ihren Geräten installieren und konfigurieren können. Sie können diese Bibliotheken verwenden, um Videos auf verschiedene Arten sicher zu verbinden und sie zuverlässig zu streamen, z. B. in Echtzeit, nachdem sie einige Sekunden lang zwischengespeichert wurden, oder als after-the-fact Medien-Uploads.
- Kinesis-Videostream — Eine Ressource, die Sie verwenden können, um Live-Videodaten zu transportieren, optional zu speichern und die Daten sowohl in Echtzeit als auch auf Batch- oder

Einmalbasis für den Konsum verfügbar zu machen. In einer typischen Konfiguration veröffentlicht in einem Kinesis-Videostream nur ein Produzent Daten.

Der Stream kann Audio-, Video- und ähnliche zeitcodierte Datenstreams transportieren, z. B. Tiefenmessungsfeeds, RADAR-Feeds und mehr. Sie erstellen einen Kinesis-Videostream mithilfe von AWS Management Console oder programmgesteuert mithilfe von. 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 niedriger Latenz erforderlich ist. Sie können diese Verbraucheranwendungen für die Ausführung auf EC2 Amazon-Instances erstellen.
  - Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an— Ermöglicht Kinesis Video Streams Streams-Anwendungen das zuverlässige Abrufen von Medien aus dem Kinesis-Videostream mit geringer Latenz. Zudem werden die Frame-Grenzen in den Medien analysiert, damit Anwendungen sich auf die Verarbeitung und Analyse der Frames konzentrieren können.

Unterstützung für Kinesis Video Streams API und Producer-Bibliotheken

Mit Kinesis Video Streams APIs können Sie Streams erstellen und verwalten sowie Mediendaten in und aus einem Stream lesen oder schreiben. 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 Producer-Bibliotheken, die Sie in Ihrem Anwendungscode verwenden können, um Daten aus Ihren Medienquellen zu extrahieren und in Ihren Kinesis-Videostream hochzuladen.

Themen

- Kinesis Video Streams Videostreams-API
- Muster für die Erkennung von Endpunkten
- Producer-Bibliotheken

Kinesis Video Streams Videostreams-API

Kinesis Video Streams ermöglicht APIs die Erstellung und Verwaltung von Kinesis Video Streams. Es ermöglicht auch APIs das Lesen und Schreiben von Mediendaten in einen Stream wie folgt:

- 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 eintreffen, weist Kinesis Video Streams in aufsteigender Reihenfolge eine eindeutige Fragmentnummer zu. Es speichert auch produzenten- und serverseitige Zeitstempel für jedes Fragment als Kinesis Video Streams-spezifische Metadaten.
- Verbraucher APIs — Verbraucher können Folgendes 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 wurden, `GetMedia` wechselt es zum Lesen von Fragmenten aus einem speicherinternen 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. Nutzer, die offline sind, können sich dafür entscheiden, bestimmte Medienfragmente oder Videobereiche explizit abzurufen, indem sie und kombinieren. `ListFragments` `GetMediaFromFragmentList` APIs `ListFragments`sund `GetMediaFromFragmentList` ermöglichen es einer Anwendung, Videosegmente für einen bestimmten Zeitraum oder Fragmentbereich zu identifizieren und diese Fragmente dann entweder sequentiell oder parallel zur Verarbeitung abzurufen. 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 Fragmentnummer, produzenten- und serverseitige Zeitstempel usw.

  2. Fragment-Metadatenliste verwenden und Fragmente in beliebiger Reihenfolge abrufen. Um beispielsweise alle Fragmente für den Tag zu verarbeiten, könnte der Verbraucher die Liste in Unterlisten aufteilen und Mitarbeiter (z. B. mehrere EC2 Amazon-Instances) die Fragmente parallel mithilfe von abrufen und parallel verarbeiten lassen. `GetMediaFromFragmentList`

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 produzentenseitige Zeitstempel

Wenn ein Verbraucher Medienmetadaten anfordert, gibt Kinesis Video Streams einen Stream von Chunks 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.

Muster für die Erkennung von Endpunkten

REST der Steuerungsebene APIs

Verwenden Sie die Kinesis Video Streams Streams-Dienstendpunkte APIs, um auf den REST der Kinesis Video Streams Streams-Kontrollebene zuzugreifen.

REST der Datenebene APIs

Kinesis Video Streams basiert auf einer Mobilfunkarchitektur, um eine bessere Skalierung und Isolierung des Datenverkehrs zu gewährleisten. Da jeder Stream einer bestimmten Zelle in einer Region zugeordnet ist, muss Ihre Anwendung die richtigen zellenspezifischen Endpunkte verwenden, denen Ihr Stream zugeordnet wurde. Wenn Sie auf die Datenebene REST zugreifen APIs, müssen Sie die richtigen Endpunkte selbst verwalten und zuordnen. Dieser Prozess, das Endpoint Discovery Pattern, wird im Folgenden beschrieben:

1. Das Endpoint Discovery Pattern beginnt mit dem Aufruf einer der `GetEndpoints` Aktionen. Diese Aktionen gehören zur Kontrollebene.
    1. Wenn Sie die Endpunkte für die the section called “Amazon Kinesis Video Streams — Archivierte Medien” Dienste the section called “Amazon Kinesis Video Streams Streams-Medien” oder abrufen, verwenden Sie. the section called “GetDataEndpoint”
    2. Wenn Sie die Endpunkte fürthe section called “Amazon Kinesis Kinesis-Videosignalkanäle”, oder Kinesis Video Signaling abrufenthe section called “Amazon Kinesis Video WebRTC-Speicher”, verwenden Sie. the section called “GetSignalingChannelEndpoint”
2. Zwischenspeichern Sie den Endpunkt und verwenden Sie ihn erneut.
3. Wenn der zwischengespeicherte Endpunkt nicht mehr funktioniert, rufen Sie erneut auf, um den Endpunkt `GetEndpoints` zu aktualisieren.

Producer-Bibliotheken

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 Zu Kinesis Video Streams hochladen.

Kinesis Video Streams Streams-Datenmodell

Das Zu Kinesis Video Streams hochladen und die the section called “Streamen Sie mithilfe der 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 Zu Kinesis Video Streams hochladen 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++ 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/kinesisvideo`	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 `video/h264` 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	Gibt die von Frame-Tim estamps verwendete Skala an. Der Standardwert beträgt 1 Millisekunde. Wenn Sie `0` festlegen, wird ebenfalls der Standardwert von 1 Millisekunde festgelegt. Dieser Wert kann zwischen 100 Nanosekunden und einer Sekunde liegen. Weitere Informationen finden Sie TimecodeScalein der Matroska-Dokumentation.	

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

Element	Beschreibung	Typische Werte
fragment_acks	Fall`true`, werden Bestätigungen (ACKs) gesendet, wenn Kinesis Video Streams die Daten empfängt. ACKs Sie können mit den Rückrufen oder empfangen werden. `KinesisVideoStreamFragmentAck` `KinesisVideoStreamParseFragmentAck`	true
restart_on_error	Gibt an, ob der Stream die Übertragung nach einem Fehler fortsetzen soll.	true
nal_adaptation_flags	Gibt an, ob der Inhalt NAL (Network Abstraction Layer)-Adaptions- oder Codec-interne Daten enthält. Gültige Flags sind `NAL_ADAPTATION_ANNEXB_NALS` und `NAL_ADAPTATION_ANNEXB_CPD_NALS` .	NAL_ADAPTATION_ANNEXB_NALS
frame_rate	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 `0` wird der Standardwert `24` zugewiesen.	24

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 CodecID 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 `MkvStatics.cpp` enthält das `gMkvTrackVideoBits` Array in die Pixelbreite und -höhe für den Frame.	
codecPrivateDataGröße	Die Größe der Daten im Parameter `codecPrivateData`	
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-Bibliothek 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 angezeigt zu werden AWS Management Console, müssen die Daten im H.264-Codec codiert sein. Kinesis Video Streams kann jedoch zeitserialisierte Datenströme in jedem Format empfangen.

Systemanforderungen für Amazon Kinesis Video Streams

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 umfasst 512 MB. 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 Zu Kinesis Video Streams hochladen besteht ein Mindestspeicherbedarf von 170 MB und eine empfohlener Speicherbedarf von 512 MB.

Kontingente für den Amazon Kinesis Video Streams Streams-Service

Kinesis Video Streams hat die folgenden Servicequotas:

> **Important**
>
> Die folgenden Servicekontingente sind entweder weich [s], die durch Einreichung eines Supporttickets aktualisiert werden können, oder hart [h], die nicht erhöht werden können. In den folgenden Tabellen werden neben den einzelnen Servicekontingenten [s] und [h] angezeigt.

API-Dienstkontingente auf Kontrollebene

Im folgenden Abschnitt werden Dienstkontingente für die Kontrollebene beschrieben APIs. TPS steht für Transaktionen pro Sekunde.

Wenn ein Anforderungslimit auf Konto- oder Ressourcenebene erreicht wird, wird a ausgelöst. `ClientLimitExceededException`

API	Kontolimit: Anfrage	Kontolimit: Streams	Grenzwerte auf Stream-Ebene	Relevante Ausnahmen und Hinweise
CreateStream	50 TPS [s]	10.000 Streams pro Konto (en) in allen unterstützten Regionen. **Note** Dieses Limit kann auf bis zu 100 000 (oder mehr) Streams pro Konto [s] erhöht werden. Melden Sie sich AWS Management Console		Geräte CLIs, SDK-gesteuerter Zugriff und die Konsole können alle diese API aufrufen. Es ist nur ein API-Aufruf erfolgreich, wenn der Stream noch nicht vorhanden ist.

API	Kontolimit: Anfrage	Kontolimit: Streams	Grenzwerte auf Stream-Ebene	Relevante Ausnahmen und Hinweise
		unter an https://console.aws.amazo.com/ und fordern Sie eine Erhöhung dieses Limits an.		
DeleteEdgeConfiguration	10 TPS [h]	N/A	1 TPS [h]	
DeleteStream	50 TPS [h]	N/A	5 TPS [h]	
DescribeEdgeConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeImageGenerationConfiguration	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anfrage	Kontolimit: Streams	Grenzwerte auf Stream-Ebene	Relevante Ausnahmen und Hinweise
DescribeMappedResourceConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeNotificationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeStream	300 TPS [h]	N/A	5 TPS [h]	
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.
ListEdgeAgentConfigurations	50 TPS [h]	N/A	N/A	
ListStreams	50 TPS [h]	N/A		
ListTagsForStream	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anfrage	Kontolimit: Streams	Grenzwerte auf Stream-Ebene	Relevante Ausnahmen und Hinweise
StartEdgeConfigurationUpdate	10 TPS [h]	N/A	1 TPS [h]	
TagStream	50 TPS [h]	N/A	5 TPS [h]	
UntagStream	50 TPS [h]	N/A	5 TPS [h]	
UpdateDataRetention	50 TPS [h]	N/A	5 TPS [h]	
UpdateImageGenerationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
UpdateNotificationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
UpdateStream	50 TPS [h]	N/A	5 TPS [h]	

API-Dienstkontingente für Medien und archivierte Medien

Im folgenden Abschnitt werden Dienstkontingente für Medien und archivierte Medien beschrieben. APIs

Wenn ein Anforderungslimit auf Konto- oder Ressourcenebene erreicht wird, wird a ausgelöst. `ClientLimitExceededException`

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-Dienstkontingente auf Datenebene

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
PutMedia	5 TPS [h]	1 [h]	12,5 MB/Sekunde oder 100 Mbit/s [s] pro Stream	• Minimale Fragmentdauer: 1 Sekunde [h] • Maximale Fragmentdauer: 20 Sekunden [h] • Maximale Fragmentgröße: 50 MB [h]	PutMediaAnfragen sind Streaming-Verbindungen mit langer Laufzeit. Sie müssen nicht für jedes Datenelement eine neue Verbindung öffnen, da Sie mehrere Fragmente in einer einzigen dauerhaften Verbindung senden können. Wenn Sie versuchen, mehrere Verbindungen gleichzeitig `PutMedia` herzustellen, drosselt Kinesis Video Streams die letzten Verbindungen mit einer Fehlermeldung.

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
				• Maximale Anzahl von Tracks: 3 [s] • Maximale Anzahl gesendeter Fragmente pro Sekunde: 5 [h] • Maximales Limit für Fragment-Metadaten: 10 Tags [h]	`ConnectionLimitExceededException`
GetClip	N/A	N/A	100 MB Größenbeschränkung [h]	Maximale Anzahl der Fragmente: 200 [h]	
Holen DASHStreaming Sie sich die SessionURL	25 TPS [h]	N/A	–	N/A	

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
Holen Sie sich die SessionURL HLSStreaming	25 TPS [h]	N/A	–	–	
GetImages	–	N/A	100 MB [h]	N/A	Die maximale Anzahl von Bildern pro Anfrage beträgt 100 [h]. **Note** Der Mindestwert für `SamplingInterval` ist 200 Millisekunden (ms), was 5 Bildern pro Sekunde entspricht.

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
GetMedia	5 TPS [h]	3 [h]	25 MB/s oder 200 Mbit/s [s]	Pro Sekunde werden maximal 5 Fragmente gesendet [h]	GetMediaAnfragen sind Streaming-Verbindungen mit langer Laufzeit. Sie müssen nicht für jedes Datenelement eine neue Verbindung öffnen, da Sie mehrere Fragmente in einer einzigen dauerhaften Verbindung senden können. Wenn Sie versuchen, mehr als drei `GetMedia` Verbindungen gleichzeitig herzustellen, drosselt Kinesis Video Streams die letzten Verbindungen mit einer Fehlermeldung. `ConnectionLimitExceededException` Wenn ein typisches Fragment ungefähr 5 MB groß ist, bedeutet diese Grenze ~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.

API	Grenzwerte auf Stream-Ebene	Grenzwerte auf Verbindungsebene	Bandbreitenbegrenzung	Grenzwerte auf Fragmentebene	Relevante Ausnahmen und Hinweise
					Note `GetMedia`wird nicht für HLS/DASH die Wiedergabe verwendet.
GetMediaForFragmentList	N/A	5 [s]	25 MB/s oder 200 Mbit/s [s]	Maximale Anzahl der Fragmente: 1000 [h]	Fünf fragmentbasierte, verbrauchende Anwendungen können gleichzeitig aufgerufen werden. `GetMediaForFragmentList` Weitere Verbindungen werden abgelehnt.

API-Dienstkontingente für das Videowiedergabeprotokoll

API	Grenzwerte auf Sitzungsebene	Grenzwerte auf Fragmentebene
DASHManifestPlaylist abrufen	5 TPS [h]	Maximale Anzahl von Fragmenten pro Playlist: 5.000 [h]
HLSMasterPlaylist abrufen	5 TPS [h]	N/A
HLSMediaPlaylist abrufen	5 TPS [h]	Maximale Anzahl von Fragmenten pro Playlist: 5.000 [h]

API	Grenzwerte auf Sitzungse bene	Grenzwerte auf Fragmente bene
Holen MP4 InitFragment	5 TPS [h]	N/A
Holen MP4 MediaFragment	20 TPS [h]	N/A
Holen TSFragment	20 TPS [h]	N/A

Fragment-Metadaten und Kontingente für fragmentierte Medien

Kinesis Video Streams APIs für den Zugriff auf archivierte Medien werden anhand der Anzahl der angeforderten Fragmente und nicht anhand der Anzahl der API-Aufrufe gedrosselt. APIs sind sowohl durch die Anzahl der Fragment-Metadaten als auch durch die Anzahl der angeforderten Fragmentmedien ratenbegrenzt. Die Kontingente für Fragmentmetadaten und Fragmentmedien werden pro Stream angewendet. Mit anderen Worten, Anfragen nach Fragmentmetadaten oder Medien in einem Stream gelten nicht für die Kontingente eines anderen Streams. Innerhalb eines bestimmten Streams wird jedes Kontingent jedoch von mehreren geteilt APIs. Das bedeutet, dass für einen bestimmten Stream Anfragen nach Fragmenten aus verschiedenen Quellen dasselbe Kontingent APIs verbrauchen. Wenn entweder das Fragmentmetadaten- oder das Fragmentmedienkontingent für einen Stream überschritten wird, gibt die API a zurück`ClientLimitExceededException`. Die folgenden Tabellen zeigen, wie die beiden Kontingenttypen APIs verbrauchen. Gehen Sie für die zweite Spalte in diesen Tabellen davon aus, dass ein Stream ein Kontingent von N hat, das bedeutet, dass er N Punkte von diesem Quotentyp für diesen Stream verbrauchen APIs muss. Die `GetClip` API erscheint in beiden Tabellen.

Kontingentverbrauch von Fragment-Metadaten

API	Anzahl der pro Anfrage verbrauchten Kontingen tpunkte	Geteiltes Kontingent (N)
`ListFragments`	Wert des `MaxResults` Parameters	10.000 Quotenpunkte pro Sekunde, pro Stream [h]
`GetClip`	Anzahl der Fragmente im resultierenden Clip	

API	Anzahl der pro Anfrage verbrauchten Kontingentpunkte	Geteiltes Kontingent (N)
`GetHLSMediaPlaylist`	Wert des `MaxMediaPlaylistFragmentResults` Parameters	
`GetDASHManifest`	Wert des `MaxManifestFragmentResults` Parameters	
`GetImages`	Wert von 400 + maximale Anzahl angeforderter Bilder	

Verbrauch fragmentarischer Medienkontingente

API	Anzahl der pro Anfrage verbrauchten Kontingentpunkte	Geteiltes Kontingent (N)
`GetMediaForFragmentList`	Anzahl der Fragmente im Fragments-Parameter	500 Quotenpunkte pro Sekunde, pro Stream [h]
`GetClip`	Anzahl der Fragmente im resultierenden Clip	
`GetMP4MediaFragment`	1	
`GetTSFragment`	1	
`GetImages`	Maximale Anzahl angeforderter Bilder	

Bei einem Kontingent von 500 Fragmentmedien pro Sekunde werden beispielsweise die folgenden Aufrufmuster für einen bestimmten Stream unterstützt:

- 5 Anfragen pro Sekunde bis `GetClip` zu 100 Fragmente in jedem Clip.
- 100 Anfragen pro Sekunde bis `GetClip` mit 5 Fragmenten in jedem Clip.
- 2 Anfragen pro Sekunde `GetClip` mit 100 Fragmenten in jedem Clip und 3 Anfragen pro Sekunde `GetMediaForFragmentList` in jedem Clip.
- 400 Anfragen pro Sekunde an `GetMP4MediaFragment` und 100 Anfragen pro Sekunde an`GetTSFragment`.

Diese Kontingente haben wichtige Auswirkungen auf die Anzahl der HLS- und MPEG-DASH-Sitzungen, die pro Stream unterstützt werden können. Die Anzahl der HLS- und DASH-Sitzungen, die von Mediaplayern zu einem bestimmten Zeitpunkt verwendet werden können, ist unbegrenzt. Daher ist es wichtig, dass die Wiedergabeanwendung nicht zulässt, dass zu viele Sitzungen gleichzeitig verwendet werden. 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, Fragmenten mit einer Dauer von 1 Sekunde, einer Audio- und Videospur und einer `MaxMediaPlaylistFragmentResults` Einstellung von fünf führt ein Media Player in der Regel zwei Aufrufe `GetHLSMediaPlaylist` pro Sekunde durch. Ein Anruf bezieht sich auf die neuesten Videometadaten und ein anderer auf die entsprechenden Audio-Metadaten. Die beiden Aufrufe verbrauchen jeweils fünf Quotenpunkte für Fragmentmetadaten. Außerdem werden zwei Aufrufe `GetMP4MediaFragment` pro Sekunde getätigt: ein Anruf für das neueste Video und ein weiterer für das entsprechende Audio. Jeder Anruf 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 Kontingent von 10.000) und 500 Fragmentmedienkontingentpunkte pro Sekunde.

Beispiel 2: Wiedergabe auf Abruf

In einem On-Demand-Wiedergabe-Szenario eines vergangenen Ereignisses mit MPEG-DASH, einer Audio- und Videospur, die auf 1.000 `MaxManifestFragmentResults` eingestellt ist, ruft ein Media Player normalerweise `GetDASHManifest` einmal zu Beginn der Sitzung auf (verbraucht 1.000 Fragmentmetadaten-Quota-Punkte) und ruft mit einer `GetMP4MediaFragment` Geschwindigkeit von bis zu 5 Mal pro Sekunde auf (verbraucht 5 Fragment-Medienkontingentpunkte), bis alle Fragmente geladen sind. In diesem Szenario können bis zu 10 neue Sitzungen pro Sekunde gestartet werden (genau bei der Quote von 10.000 Fragmentmetadaten pro Sekunde), und bis zu 100 Sitzungen

können Fragmentmedien mit einer Geschwindigkeit von 5 pro Sekunde aktiv laden (genau bei der Quote von 500 Fragmentmedien pro Sekunde).

Sie können `ArchivedFragmentsConsumed.Metadata` und verwenden`ArchivedFragmentsConsumed.Media`, um den Verbrauch von Fragment-Metadaten bzw. Quotenpunkten für Fragmentmedien zu überwachen. Hinweise zur Überwachung finden Sie unterthe section called “Überwachen”.

Kontingente für Streaming-Metadaten

Die folgenden Dienstkontingente gelten für das Hinzufügen von Streaming-Metadaten zu 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.
- Der Name eines Fragment-Metadaten darf nicht mit der Zeichenfolge "`AWS`" beginnen. 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.

SDK-Kontingente für Produzenten

Die folgende Tabelle enthält die aktuellen Kontingente für Werte im SDK. Weitere Informationen finden Sie unter Zu Kinesis Video Streams hochladen.

> **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-

Wert	Limit	Hinweise
		Objekt erstellen kann. Dies ist ein weiches Limits (Sie können eine Erhöhung beantragen). Es garantier t, 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-Na mens)	256 Zeichen	
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 Stammverz eichnispfades)	4,096 Zeichen	
Max auth info length (Max. Länge der Authentifizierungs informationen)	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	

Wert	Limit	Hinweise
Max tag value length (Max. Länge des Tag-Wertes)	1,024 Zeichen	
Min security token period (Min. Zeitraum des Sicherheits-Tokens)	30 Sekunden	
Security token grace period (Übergangsfrist des Sicherheits-Tokens)	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.
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 Kontingente für den Amazon Kinesis Video Streams Streams-Service.
Max fragment duration (Max. Fragmentdauer)	20 Sekunden	Weitere Informationen finden Sie unter Kontingente für den Amazon Kinesis Video Streams Streams-Service.

Wert	Limit	Hinweise
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-Funktion gesendeten Empfangsbestätigung.
Max content type string length (Max. Zeichenfolgenlänge des Inhaltstyps)	128 Zeichen	
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.

Wert	Limit	Hinweise
Max timecode scale value length (Max. Länge des Timecode-Skalierungswertes)	1 Sekunde	Der maximale Timecode-Skalierungswert zur Repräsentierung 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	
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	

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 Benutzers mit Administratorzugriff
- 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/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscodes auf der Telefontastatur.

   Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontoserstellt. 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. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie 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 Consoleals 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 Benutzers mit Administratorzugriff

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 Benutzerzugriff.

   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.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.

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

Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

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.

Vorteile

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 AWS Marketplace verarbeiten und analysieren. Sie können Kinesis Video Streams verwenden `Get` APIs , 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 ().AWS 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 -Preisrechner.

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

- Erstellen Sie einen Amazon Kinesis Kinesis-Videostream
- Daten an einen Amazon Kinesis Kinesis-Videostream senden
- Mediendaten konsumieren

Erstellen Sie einen Amazon Kinesis Kinesis-Videostream

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 unterhttps://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 Verwenden Sie die C++-Producer-Bibliothek als Beispiel: GStreamer SDK-Plug-In für Kinesis Video Streams Producer — kvssink Plug-in verwendet.

Um Medien von einer Vielzahl von Geräten auf verschiedenen Betriebssystemen zu senden, verwendet dieses Tutorial die Kinesis Video Streams C++ Producer-Bibliothek und ein Open-Source-Medienframework GStreamer, 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
- Überprüfen Sie die Bestätigungsobjekte

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
- Wird CMakezum Generieren von Makefiles verwendet
- 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, offen 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 aus.

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/awslabs/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. Wird verwendet 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

11. (Optional) Sie können dem Startskript Ihrer Shell die Einstellung der Umgebungsvariablen GST_PLUGIN_PATH hinzufügen. Dadurch wird sichergestellt, dass GST_PLUGIN_PATH während einer neuen Terminalsitzung richtig eingestellt ist. In lautet das AWS Cloud9 Startskript der Shell:. `~/.bashrc`

    Führen Sie den folgenden Befehl aus, um den Befehl an das Ende des Startskripts der Shell anzuhängen.

    ```
    echo "export GST_PLUGIN_PATH=~/environment/amazon-kinesis-video-streams-
    producer-sdk-cpp/build" >> ~/.bashrc
    ```

    Geben Sie Folgendes ein, um das Startskript der Shell auszuführen:

    ```
    source ~/.bashrc
    ```

    Vergewissern Sie sich, dass GST_PLUGIN_PATH gesetzt ist.

    ```
    echo $GST_PLUGIN_PATH
    ```

    Wenn Sie die Ausgabe korrekt eingestellt haben, wird die folgende Ausgabe angezeigt. Wenn die Ausgabe leer ist, ist die Umgebungsvariable nicht richtig gesetzt.

    ```
    /home/ubuntu/environment/amazon-kinesis-video-streams-producer-sdk-cpp/build
    ```

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_ACCESS_KEY=YourSecretKey
export AWS_DEFAULT_REGION=YourAWSRegion
```

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
```

Videospezifikationen:

- Auflösung: 1280 x 720 Pixel
- Bildrate - 30 Bilder pro Sekunde
- Dauer - 14,0 Sekunden
- Videokodierung - H.264, in Track 1
- Keyframes — Alle 3 Sekunden, was zu einer Fragmentdauer (auch bekannt als Größe einer Gruppe von Bildern (GoP)) von 3 Sekunden führt, wobei das letzte Fragment 2 Sekunden lang ist.

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 unterthe section called “Erstellen Sie einen Amazon Kinesis Kinesis-Videostream”.

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

Sample video from GStreamer

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

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

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

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

Diese GStreamer Pipeline generiert einen Live-Testvideostream mit einem Standardtestmuster, das mit 10 Bildern pro Sekunde und einer Auflösung von 640 x 480 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. `kvssink`nimmt 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 beenden, 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: GStreamer SDK-Plug-In für Kinesis Video Streams Producer — kvssink

Überprüfen Sie die Bestätigungsobjekte

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":"123456789012345678
```

Wenn die Bestätigung „Ja“ `EventType` 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 Videowiedergabe.

Zu Kinesis Video Streams hochladen

Die Amazon Kinesis Video Streams Streams-Producer-Bibliotheken sind eine Reihe von Bibliotheken im Kinesis Video Streams Streams-Producer-SDK. Der Client verwendet die Bibliotheken und das SDK, um die geräteinterne Anwendung für die sichere Verbindung zu Kinesis Video Streams zu erstellen und Mediendaten zu streamen, die in der Konsole oder in Client-Anwendungen in Echtzeit angezeigt werden können.

Mediendaten können auf folgende Arten gestreamt werden:

- In Echtzeit
- Nachdem ich es für ein paar Sekunden gepuffert habe
- Nach den Medien-Uploads

Nachdem Sie einen Kinesis Video Streams 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 die folgenden Komponenten:

- Produzent/Kunde von Kinesis Video Streams
- Produzentenbibliothek für Kinesis Video Streams

Produzent/Kunde von Kinesis Video Streams

Der Kinesis Video Streams Producer-Client umfasst eine einzige `KinesisVideoClient` Klasse. Diese Klasse verwaltet Medienquellen, empfängt Daten von den Quellen und verwaltet den Stream-Lebenszyklus, wenn 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 Hardware 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 Infrarot RADARs - oder Tiefenkameras

Kinesis Video Streams bietet keine integrierten Implementierungen für medienproduzierende Geräte wie Kameras. Um Daten von diesen Geräten zu extrahieren, müssen Sie den entsprechenden Code selbst implementieren, d. h. eine benutzerdefinierte Medienquelle erstellen. Anschließend können Sie Ihre benutzerdefinierten Medienquellen explizit bei registrieren`KinesisVideoClient`, wodurch die Daten in Kinesis Video Streams hochgeladen werden.

Der Kinesis Video Streams Producer Client ist für Java- und Android-Anwendungen verfügbar. Weitere Informationen erhalten Sie unter Verwenden Sie die Java-Producer-Bibliothek und Verwenden Sie die Android Producer-Bibliothek.

Produzentenbibliothek für Kinesis Video Streams

Die Kinesis Video Streams Producer-Bibliothek 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 Geräteressourcen.

Die Kinesis Video Streams Producer-Bibliothek implementiert die Zustandsmaschine 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 können sich aus den folgenden Gründen dafür entscheiden, die Kinesis Video Streams Streams-Producer-Bibliothek 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 Streams-Produzentenbibliothek für Android-, C-, C++- und Java-Anwendungen verfügbar. Weitere Informationen zu den unterstützten Sprachen finden Sie in den folgenden verwandten Themen.

Verstehen Sie, was Producer-Bibliotheken sind

Verwenden Sie die Java-Producer-Bibliothek

Verwenden Sie die Android Producer-Bibliothek

Verwenden Sie die C++-Producer-Bibliothek

Verwenden Sie die C-Producer-Bibliothek

Verwenden Sie das C++-Producer-SDK auf dem Raspberry Pi

Verwenden Sie die Java-Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte Java-Producer-Bibliothek verwenden, um Anwendungscode mit minimaler Konfiguration zu schreiben, um Mediendaten von einem Gerät an einen Kinesis-Videostream zu senden.

Führen Sie die folgenden Schritte aus, um Ihren Code in Kinesis Video Streams zu integrieren, sodass 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.

Verfahren: Verwenden Sie das Java Producer SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Java Producer-Client in Ihrer Java-Anwendung verwenden, um Daten an Ihren Kinesis-Videostream 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:

- Laden Sie den Code herunter und konfigurieren Sie ihn
- Schreiben und untersuchen Sie den Code
- Führen Sie den Code aus und überprüfen Sie ihn

Voraussetzungen

Bevor Sie das Java Producer SDK einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer Profildatei mit den AWS Anmeldeinformationen einrichten. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein. Weitere Informationen finden Sie unter AWS Anmeldeinformationen und Region für die Entwicklung einrichten in der AWS SDK für 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).

- Sie `NativeLibraryPath` müssen Ihre `KinesisVideoProducerJNI` Datei enthalten, die unter https://github.com/awslabs/amazon-kinesis-video-streams- verfügbar istproducer-sdk-cpp. Die Namenserweiterung 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 `src/main/resources/lib` unter https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-java .git verfügbar. Für andere Umgebungen kompilieren Sie die Datei C++.

Laden Sie den Code der Java-Producer-Bibliothek herunter und konfigurieren Sie ihn

In diesem Abschnitt der Java-Producer-Bibliotheksprozedur laden Sie den Java-Beispielcode herunter, importieren das Projekt in Ihre Java-IDE und konfigurieren die Bibliotheksspeicherorte.

Voraussetzungen und weitere Informationen 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/awslabs/amazon-kinesis-video-streams-producer-sdk-java
   ```

2. Öffnen Sie die Java Integrated Development Environment (IDE), die Sie verwenden (z. B. Eclipse oder JetBrains IntelliJ IDEA), und importieren Sie das Apache Maven-Projekt, das Sie heruntergeladen haben:

   - 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 Anmeldeinformationen. AWS 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 für Java-Referenz.

Schreiben und untersuchen Sie den Code

In diesem Abschnitt der Java-Producer-Bibliotheksprozedur 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 das `MediaSource` und es beginnt, Daten an den Client zu senden.

Die folgenden Abschnitte stellen Details bereit.

Erstellen Sie eine Instanz 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 = ABCDEFGHIJKLMOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

Erstellen Sie eine Instanz 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 Kinesis Video Streams Streams-Java-Bibliothek stellt beispielsweise die `ImageFileMediaSource` Implementierung der `MediaSource` Schnittstelle bereit. Diese Klasse liest nur Daten aus einer Reihe von Mediendateien und nicht aus einem Kinesis-Videostream. Sie können sie jedoch zum Testen des Codes verwenden.

```
final MediaSource bytesMediaSource = createImageFileMediaSource();
```

Registrieren Sie das 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 Sie die Medienquelle

Starten Sie die Medienquelle, damit sie mit der Generierung von Daten beginnen und sie an den Client senden kann.

```
bytesMediaSource.start();
```

Bereinigen von -Ressourcen

Um Speicherlecks zu vermeiden, gehen Sie wie folgt vor, um die Registrierung einer Medienquelle auf dem Client aufzuheben und den Client freizugeben.

```
try {
    kinesisVideoClient.unregisterMediaSource(mediaSource);
    kinesisVideoClient.free();
} catch (final KinesisVideoException e) {
    throw new RuntimeException(e);
}
```

Wenn Sie dem Cache Elemente hinzugefügt haben `CachedInfoMultiAuthServiceCallbacks`, zum Beispiel:

```
serviceCallbacks.addStreamInfoToCache(streamName, streamInfo);
serviceCallbacks.addStreamingEndpointToCache(streamName, dataEndpoint);
```

Leeren Sie den Cache, wenn Sie fertig sind:

```
serviceCallbacks.removeStreamFromCache(streamName);
```

Führen Sie den Code aus und überprüfen Sie ihn

Gehen Sie wie folgt vor, um den Java-Test-Harness für die Java-Producer-Bibliothek auszuführen.

1. Wählen Sie DemoAppMain.
2. Wählen Sie „Ausführen“, „AusführenDemoAppMain“.

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 Kinesis Video Streams Streams-Konsole an AWS Management Console und öffnen Sie sie.

   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 Sie die Android Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte Android-Producer-Bibliothek verwenden, um Anwendungscode mit minimaler Konfiguration zu schreiben, um 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, sodass 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.

Verfahren: Verwenden Sie das Android Producer SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Android Producer-Client in Ihrer Android-Anwendung verwenden, um Daten an Ihren Kinesis-Videostream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- the section called “Voraussetzungen”
- the section called “Laden Sie den Code herunter und konfigurieren Sie ihn”
- the section called “Untersuchen Sie den Code”
- the section called “Führen Sie den Code aus und überprüfen Sie ihn”

Voraussetzungen

Wir empfehlen Android Studio für das Überprüfen, Bearbeiten und Ausführen des Anwendungscodes. Wir empfehlen die Verwendung der neuesten stabilen Version.

Im Beispielcode geben Sie Amazon Cognito Cognito-Anmeldeinformationen an.

Gehen Sie wie folgt vor, um einen Amazon Cognito Cognito-Benutzerpool und einen Identitätspool einzurichten.

- Richten Sie einen Benutzerpool ein
- Richten Sie einen Identitätspool ein

Richten Sie einen Benutzerpool ein

Einrichten eines Benutzerpools

1. Melden Sie sich bei der Amazon Cognito Cognito-Konsole an und überprüfen Sie, ob die Region korrekt ist.
2. Wählen Sie in der linken Navigation Benutzerpools aus.
3. Wählen Sie im Abschnitt Benutzerpools die Option Benutzerpool erstellen aus.
4. Füllen Sie die folgenden Abschnitte aus:

    a. Schritt 1: Anmeldeerlebnis konfigurieren — Wählen Sie im Abschnitt Anmeldeoptionen für den Cognito-Benutzerpool die entsprechenden Optionen aus.

Klicken Sie auf Weiter.

b. Schritt 2: Sicherheitsanforderungen konfigurieren — Wählen Sie die entsprechenden Optionen aus.

   Klicken Sie auf Weiter.

c. Schritt 3: Anmeldevorgang konfigurieren — Wählen Sie die entsprechenden Optionen aus.

   Klicken Sie auf Weiter.

d. Schritt 4: Nachrichtenzustellung konfigurieren — Wählen Sie die entsprechenden Optionen aus.

   Wählen Sie im Auswahlfeld für die IAM-Rolle eine vorhandene Rolle aus oder erstellen Sie eine neue Rolle.

   Klicken Sie auf Weiter.

e. Schritt 5: Integrieren Sie Ihre App — Wählen Sie die entsprechenden Optionen aus.

   Wählen Sie im Feld Erster App-Client die Option Vertraulicher Kunde 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.

5. Wählen Sie auf der Seite Benutzerpools den Pool aus, den Sie gerade erstellt haben.

   Kopieren Sie die Benutzerpool-ID und notieren Sie sich diese für später. In der `awsconfiguration.json` Datei ist das`CognitoUserPool.Default.PoolId`.

6. Wählen Sie die Registerkarte App-Integration und gehen Sie zum Ende der Seite.

7. Wählen Sie im Abschnitt App-Client-Liste den Namen des App-Clients aus, den Sie gerade erstellt haben.

   Kopieren Sie die Client-ID und notieren Sie sich dies für später. In der `awsconfiguration.json` Datei ist das`CognitoUserPool.Default.AppClientId`.

8. Zeigen Sie dem Client das Geheimnis und notieren Sie sich das für später. In der `awsconfiguration.json` Datei ist das`CognitoUserPool.Default.AppClientSecret`.

Richten Sie einen Identitätspool ein

Einrichten eines Identitäten-Pools

1. Melden Sie sich bei der Amazon Cognito Cognito-Konsole an und überprüfen Sie, ob die Region korrekt ist.
2. Wählen Sie in der linken Navigation Identity Pools aus.
3. Wählen Sie Identitätspool erstellen.
4. Konfigurieren Sie den Identitätspool.

   a. Schritt 1: Identity Pool Trust konfigurieren — Füllen Sie die folgenden Abschnitte aus:

      - Benutzerzugriff — Wählen Sie Authentifizierter Zugriff
      - Authentifizierte Identitätsquellen — Wählen Sie den Amazon Cognito Cognito-Benutzerpool

      Klicken Sie auf Weiter.

   b. Schritt 2: Berechtigungen konfigurieren — Füllen Sie im Abschnitt Authentifizierte Rolle die folgenden Felder aus:

      - IAM-Rolle — Wählen Sie Neue IAM-Rolle erstellen
      - IAM-Rollenname — Geben Sie einen Namen ein und notieren Sie sich diesen für einen späteren Schritt.

      Klicken Sie auf Weiter.

   c. Schritt 3: Identitätsanbieter Connect — Füllen Sie im Abschnitt Benutzerpooldetails 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: Eigenschaften konfigurieren — Geben Sie einen Namen in das Feld Identitätspoolname ein.

      Klicken Sie auf Weiter.

e. Schritt 5: Überprüfen und erstellen — Überprüfen Sie Ihre Auswahl in den einzelnen Abschnitten und wählen Sie dann Identitätspool 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 das`CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Aktualisieren Sie die Berechtigungen für die IAM-Rolle.

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/

   b. Wählen Sie in der linken Navigation die Option Rollen aus.

   c. Suchen Sie die Rolle, die Sie oben erstellt haben, und wählen Sie sie aus.

   > **Note**
   >
   > Verwenden Sie bei Bedarf die Suchleiste.

   d. Wählen Sie die beigefügte Berechtigungsrichtlinie aus.

      Wählen Sie Bearbeiten aus.

   e. Wählen Sie die Registerkarte JSON aus und ersetzen Sie die Richtlinie durch Folgendes:

      JSON

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

```
}
```

Klicken Sie auf Weiter.

f. Aktivieren Sie das Kästchen neben Diese neue Version als Standard festlegen, falls es noch nicht ausgewählt ist.

Wählen Sie Änderungen speichern aus.

Laden Sie den Code der Android-Producer-Bibliothek herunter und konfigurieren Sie ihn

In diesem Abschnitt des Verfahrens zur Android-Producer-Bibliothek laden Sie den Android-Beispielcode herunter und öffnen das Projekt in Android Studio.

Voraussetzungen und weitere Informationen zu diesem Beispiel finden Sie unter Verwenden der Android-Producer-Bibliothek.

1. Erstellen Sie ein Verzeichnis und klonen Sie es dann AWS Mobile SDK for Android aus dem GitHub Repository.

   ```
   git clone https://github.com/awslabs/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 im `CredentialsProvider` Knoten die Identitätspool-ID aus der Prozedur So richten Sie einen Identitätspool im Abschnitt Voraussetzungen ein und geben Sie Ihre AWS-Region (z. B.**`us-west-2`**) ein.

   Geben Sie im `CognitoUserPool` Knoten den geheimen App-Client, die App-Client-ID und die Pool-ID aus der Prozedur So richten Sie einen Benutzerpool im Abschnitt Voraussetzungen ein und geben Sie Ihre AWS-Region (z. B.**`us-west-2`**) an.

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": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmno",
      "AppClientId": "0123456789abcdefghijklmnop",
      "PoolId": "us-west-2_qRsTuVwXy",
      "Region": "us-west-2"
    }
  }
}
```

7. Aktualisieren Sie das `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` mit Ihrer Region (im folgenden Beispiel ist es auf US_WEST_2 gesetzt):

```
public class KinesisVideoDemoApp extends Application {
    public static final String TAG = KinesisVideoDemoApp.class.getSimpleName();
    public static Regions KINESIS_VIDEO_REGION = Regions.US_WEST_2;
```

Informationen zu AWS-Region Konstanten finden Sie unter Regionen.

Untersuchen Sie den Code

In diesem Abschnitt der Android-Producer-Bibliotheksprozedur 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 das`MediaSource`, und es beginnt, Daten an den Client zu senden.

Die folgenden Abschnitte stellen Details bereit.

Erstellen Sie eine Instanz 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 Instanz von`AWSCredentialsProvider`, die Ihre Amazon Cognito Cognito-Anmeldeinformationen aus der `awsconfiguration.json` Datei liest, die Sie im vorherigen Abschnitt geändert haben.

Erstellen Sie eine Instanz 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().isEndcoderHardwareAccelerated())
                .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 Sie die 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();
```

Führen Sie den Code aus und überprüfen Sie ihn

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 Plussymbol (+), Android-App. Geben Sie im Feld Name **`AmazonKinesisVideoDemoApp`** ein. Wählen Sie im Pulldown-Menü „Modul“. AmazonKinesisVideoDemoApp 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 AWSKinesisVideoDemoAppAnwendung 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 Ihrer Benutzerpool-Seite in der Amazon Cognito 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. Wählen Sie Ok aus.
9. Behalten Sie auf der nächsten Seite die Standardwerte bei und wählen Sie Stream.
10. Melden Sie sich bei der Kinesis Video Streams Streams-Konsole in der Region USA West (Oregon) an AWS Management Console und öffnen Sie sie.

    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 namens demo-stream.

Verwenden Sie die C++-Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte C++-Producer-Bibliothek 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 Verfügung, um das Senden von Daten an einen Kinesis-Videostream zu verwalten:

- 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 Medieninhaltstyp.

Fügen Sie Medien in den Stream ein

Sie können die in 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 wurden, wenn sie einen Stream erstellt, eine Stream-Beschreibung abruft und einen Stream löscht.
- Schnittstelle für Ereignisse im Client-Bereitschaftsstatus oder bei niedrigem Speicherbedarf (**`ClientCallbackProvider`**): Die Bibliothek ruft Ereignisse auf dieser Schnittstelle auf, wenn der Client bereit ist oder wenn er feststellt, dass ihm möglicherweise nicht mehr genügend Speicherplatz oder Arbeitsspeicher zur Verfügung steht.
- Callback-Schnittstelle für Stream-Ereignisse (**`StreamCallbackProvider`**): Die Bibliothek ruft Ereignisse auf dieser Schnittstelle auf, wenn Stream-Ereignisse auftreten, z. B. wenn der Stream in den Bereitschaftszustand wechselt, Frames gelöscht wurden oder Streamfehler auftreten.

Kinesis Video Streams bietet Standardimplementierungen für diese Schnittstellen. Sie können auch Ihre eigene benutzerdefinierte Implementierung bereitstellen, z. B. wenn Sie eine benutzerdefinierte Netzwerklogik benötigen oder wenn Sie der Benutzeroberfläche einen niedrigen Speicherzustand anzeigen möchten.

Weitere Informationen zu Callbacks in den Producer-Bibliotheken finden Sie unter. SDK-Rückrufe von Produzenten

Verfahren: Verwenden Sie das C++-Producer-SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Streams-Client und Medienquellen in einer C++-Anwendung verwenden, um Daten an Ihren Kinesis-Videostream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

Voraussetzungen

Bevor Sie das C++-Producer-SDK einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Anmeldeinformationen: Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer Profildatei mit den AWS Anmeldeinformationen einrichten. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein.

  Weitere Informationen finden Sie unter AWS Anmeldeinformationen und Region für die Entwicklung einrichten.

- Integration des Zertifikatsspeichers: Die Kinesis Video Streams Streams-Producer-Bibliothek muss eine Vertrauensstellung mit 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
  - 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:
  - Git: `sudo apt install git`
  - CMake: `sudo apt install cmake`
  - G ++: `sudo apt install g++`
  - pkg-konfiguration: `sudo apt install pkg-config`
  - OpenJDK: `sudo apt install openjdk-8-jdk`

    > **Note**
    >
    > Dies ist nur erforderlich, wenn Sie Java Native Interface (JNI) erstellen.

  - Legen Sie die `JAVA_HOME`-Umgebungsvariable fest: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Laden Sie den Code der C++-Producer-Bibliothek herunter und konfigurieren Sie ihn

Informationen zum Herunterladen und Konfigurieren der C++-Producer-Bibliothek finden Sie unter Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin und JNI.

Voraussetzungen und weitere Informationen zu diesem Beispiel finden Sie unter. the section called “C++”

CMake Argumente

Im Folgenden finden Sie eine Referenztabelle für die SDK-spezifischen CMake Argumente von C++ Producer. Sie können die CMake Standardoptionen auch an CMake übergeben.

> **Important**
>
> Diese sind alle optional.

Flaggen zum Ein- oder Ausschließen bestimmter Funktionen

CMake Argument	Typ	Standard	Erklärung
`BUILD_DEPENDENCIES`	Boolesch	ON	Erstellen Sie Abhängigkeiten aus der Quelle. Verwenden Sie andernfalls die Abhängigkeiten, die bereits auf dem System installiert sind. Wenn eine der erforderlichen Abhängigkeiten nicht gefunden werden konnte, wird ein Fehler zurückgegeben.

CMake Argument	Typ	Standard	Erklärung
`BUILD_GSTREAMER_PLUGIN`	Boolesch	OFF	Baut das `kvssink` GStreamer Plugin auf.
`BUILD_JNI`	Boolesch	OFF	Erstellt das Java Native Interface (JNI), um diesen Code aus einer Java-Laufzeitumgebung aufrufen zu können.
`ALIGNED_MEMORY_MODEL`	Boolesch	OFF	Wenn die Speicherzuweisungen an 8-Byte-Grenzen ausgerichtet werden sollen. Einige Architekturen erlauben keinen unausgerichteten Speicherzugriff.
`CONSTRAINED_DEVICE`	Boolesch	OFF	Nur Nicht-Windows. Wenn ON, wird die Thread-Stack-Größe auf `0.5 MiB` gesetzt. Wird für Alpine Linux-Builds benötigt. Andernfalls wird die Standardeinstellung des Betriebssystems verwendet.

CMake Argument	Typ	Standard	Erklärung
BUILD_STATIC	Boolesch	OFF	Erstellen Sie Bibliotheken und ausführbare Dateien als gemeinsam genutzte (OFF) oder statische Dateien (ON).
ADD_MUCLIBC	Boolesch	OFF	Verlinken Sie auf uClibc statt auf die Standard-C-Bibliothek, bei der es sich um eine kleinere C-Standardbibliothek handelt, die für eingebettete Systeme konzipiert ist.
OPEN_SRC_INSTALL_PREFIX	String	.. /quelloffen/lokal	Speicherort für die Installation der Open-Source-Abhängigkeiten, falls sie aus dem Quellcode erstellt werden.

Flaggen für die Cross-Compilierung

> **Important**
>
> Legen Sie diese fest, wenn sich die CPU-Architekturen Ihres Ziel- und Hostcomputers unterscheiden.

CMake Argument	Typ	Standard	Erklärung
`BUILD_LOG 4CPLUS_HOST`	String	""	Erstellen Sie die `log4cplus` Abhängigkeit für die angegebene CPU-Architektur. Wenn nicht gesetzt, `log4cplus` wird die CPU-Architektur des Host-Computers automatisch erkannt und verwendet.
`BUILD_OPE NSSL_PLATFORM`	String	""	Erstellen Sie die `OpenSSL` Abhängigk eit für die angegeben e CPU-Architektur. Wenn nicht gesetzt, `OpenSSL` wird die CPU-Architektur des Host-Computers automatisch erkannt und verwendet.

Flaggen im Zusammenhang mit Tests

CMake Argument	Typ	Standard	Erklärung
`BUILD_TEST`	Boolesch	OFF	Erstellen Sie die Einheiten- und Integrationstests. Um alle Tests auszuführ en, führen Sie die Ausführung `./tst/ producerTest`

CMake Argument	Typ	Standard	Erklärung
			vom Build-Verzeichnis aus. AWS Für die Durchführung der Tests sind Anmeldeinformationen erforderlich.
CODE_COVERAGE	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. Aktiviert die Erfassung der Codeabdeckung mit gcov und die Generierung von Berichten.
COMPILER_WARNINGS	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. Aktiviert alle Compiler-Warnungen.
ADDRESS_SANITIZER	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. AddressSanitizerBaue mit.
MEMORY_SANITIZER	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. MemorySanitizerBaue mit.
THREAD_SANITIZER	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. ThreadSanitizerBaue mit.

CMake Argument	Typ	Standard	Erklärung
`UNDEFINED_BEHAVIOR_SANITIZER`	Boolesch	OFF	Nur für GNU/Clang-Compiler verfügbar. UndefinedBehaviorSanitizerBaue mit.

Um diese CMake Argumente zu verwenden, übergeben Sie sie nach dem `cmake ..` Befehl als durch Leerzeichen getrennte Liste von `-D`*`key`*`=`*`value`* Paaren. Zum Beispiel:

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -
DALIGNED_MEMORY_MODEL=ON
```

CMake sucht nach der Compiler-Toolchain, indem er der Variablen folgt. `$PATH` Stellen Sie vor der Ausführung die `CXX` Umgebungsvariablen `CC` und so ein CMake, dass explizit festgelegt wird, welche Toolchain für Cross-Compilierung verwendet werden soll.

Schreiben und untersuchen Sie den Code

In diesem Abschnitt von untersuchen Sie den Code im C++-Testcode (`tst/ProducerTestFixture.h`und anderen Dateien). the section called “C++” 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 noch kein Stream mit demselben Namen vorhanden ist.
- Rufen Sie `putFrame` im `KinesisVideoStream` für jeden Daten-Frame auf, sobald er verfügbar wird, um den Frame an den Stream zu senden.

In den folgenden Abschnitten finden Sie weitere Informationen zu diesem Codierungsmuster.

Erstellen Sie eine 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 ist 3 (Streams) * 3 (Minuten) * 60 (Sekunden in einer Minute) * 10000 (KB) * 1,2 (Defragmentierungszuschuss) = 194,4 MB ~ 200 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 ermöglicht.
- Die AWS-Region („us-west-2"). Der Service-Endpunkt wird anhand der Region bestimmt.

Erstellen Sie eine 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 Konsumentenanwendungen 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 the section called “Datenschutz”.
- 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.

Fügen Sie dem Kinesis-Videostream eine Audiospur hinzu

Sie können einer Videotrack-Stream-Definition Details zur Audiospur hinzufügen, indem Sie die `addTrack` folgende Methode verwenden`StreamDefinition`:

```
stream_definition->addTrack(DEFAULT_AUDIO_TRACKID, DEFAULT_AUDIO_TRACK_NAME,
 DEFAULT_AUDIO_CODEC_ID, MKV_TRACK_INFO_TYPE_AUDIO);
```

Die `addTrack` Methode erfordert die folgenden Parameter:

- Track-ID (wie eine für Audio). Dies sollte ein eindeutiger Wert ungleich Null sein.
- Benutzerdefinierter Titelname (z. B. „Audio“ für den Audiotrack).
- Codec-ID für diesen Titel (z. B. für den Audiotrack „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 Objekt erstellt haben, während Sie die Startmethode in aufrufen. KinesisVideoStream KinesisVideoStream

Fügen Sie einen Frame in den Kinesis-Videostream ein

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 vorangegangene 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.
- Zeitstempel der Präsentation.
- Dauer des Frames (in 100-ns-Einheiten).
- Größe des Frames in Byte.
- Framedaten.

Weitere Informationen zum Format des Frames erhalten Sie unter the section called “Datenmodell”.

Fügen Sie einen KinesisVideoFrame in einen bestimmten Track ein von KinesisVideoStream

Sie können die `PutFrameHelper` Klasse verwenden, um Frame-Daten in eine bestimmte Spur einzufügen. Rufen Sie zunächst den `getFrameDataBuffer` auf, um einen Zeiger auf einen der vorab zugewiesenen Puffer zum Ausfüllen der `KinesisVideoFrame` Daten zu erhalten. Dann können Sie den aufrufen, `putFrameMultiTrack` um den `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 `putFrameMultiTrack` Methode verwendet einen Warteschlangenmechanismus, um sicherzustellen, dass die MKV-Fragmente monoton ansteigende 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 Audio-Frames in der Warteschlange.
- Maximale Anzahl von Videoframes in der Warteschlange.
- Größe, die einem einzelnen Audioframe zugewiesen werden soll.
- Größe, die einem einzelnen Videoframe zugewiesen werden soll.

Zugriff auf Metriken und Metrikprotokollierung

Das C++ Producer SDK umfasst Funktionen für Metriken und die Protokollierung von Metriken.

Sie können die `getKinesisVideoStreamMetrics` API-Operationen `getKinesisVideoMetrics` und 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 (der Speicher, 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 Inhaltsspeicher.
- 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 Frame-Daten an Kinesis Video Streams gesendet werden).
- overallViewDuration: Der Unterschied in 100 ns-Einheiten zwischen dem Kopf der Inhaltsansicht (wenn Frames codiert sind) und dem Ende (wenn Frames aus dem Speicher gelöscht werden, entweder weil der gesamte zugewiesene Speicherplatz für die Inhaltsansicht überschritten wurde oder weil eine `PersistedAck` Nachricht von Kinesis Video Streams empfangen wird und Frames, von denen bekannt ist, dass sie persistent sind, gelöscht 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 zuletzt gemessene Rate des Streams in Bildern pro Sekunde.
- currentTransferRate: Die zuletzt 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);
```

Führen Sie den Code aus und überprüfen Sie ihn

Informationen zum Ausführen und Überprüfen des Codes für finden Sie in den folgenden betriebssystemspezifischen Anweisungen: the section called “C++”

- 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.

Verwenden Sie das C++ Producer SDK als GStreamer Plugin

GStreamerist ein beliebtes Medien-Framework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medien-Pipelines durch die Kombination modularer Plugins zu erstellen. Das Kinesis Video GStreamer Streams-Plugin optimiert die Integration Ihrer bestehenden GStreamer Medienpipeline mit Kinesis Video Streams.

Informationen zur Verwendung des C++-Producer-SDK als GStreamer Plugin finden Sie unter. Beispiel: GStreamer SDK-Plug-In für Kinesis Video Streams Producer — kvssink

Verwenden Sie das C++-Producer-SDK als GStreamer Plugin in einem Docker-Container

GStreamerist ein beliebtes Medien-Framework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medien-Pipelines durch die Kombination modularer Plugins zu erstellen. Das Kinesis Video GStreamer Streams-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.

Informationen zur Verwendung des C++-Producer-SDK als GStreamer Plugin in einem Docker-Container finden Sie unter. Führen Sie das GStreamer Element in einem Docker-Container aus

Verwenden Sie die Protokollierung mit dem C++-Producer-SDK

Sie konfigurieren die Protokollierung für C++-Producer-SDK-Anwendungen in der `kvs_log_configuration` Datei im `kinesis-video-native-build` Ordner.

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 Sie die C-Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte C-Producer-Bibliothek verwenden, um Anwendungscode zu schreiben, um Mediendaten von einem Gerät an einen Kinesis-Videostream zu senden.

Objektmodell

Die Kinesis Video Streams C Producer-Bibliothek basiert auf einer gemeinsamen Komponente namens Platform Independent Codebase (PIC), die GitHub unter https://github.com/awslabs/amazon-kinesis-video-streams -pic/ verfügbar ist. Das PIC enthält plattformunabhängige Geschäftslogik für die grundlegenden Komponenten. Die Kinesis Video Streams C-Producer-Bibliothek umschließt PIC mit einer zusätzlichen API-Ebene, die szenario- und plattformspezifische Rückrufe und Ereignisse ermöglicht. Die Kinesis Video Streams C Producer-Bibliothek besteht aus den folgenden Komponenten, die auf PIC aufbauen:

- Anbieter von Geräteinformationen — Macht die `DeviceInfo` Struktur verfügbar, die direkt an die PIC-API übergeben werden kann. Sie können eine Reihe von Anbietern konfigurieren, einschließlich eines für Anwendungsszenarien optimierten Anbieters, der den Inhaltsspeicher auf der Grundlage der Anzahl und der Typen von Streams, die Ihre Anwendung verarbeitet, und der Menge der erforderlichen Pufferung, die auf der Grundlage der Menge des verfügbaren RAM konfiguriert ist, optimieren kann.

- Stream-Informationsanbieter — Macht die `StreamInfo` Struktur verfügbar, die direkt an die PIC-API übergeben werden kann. Es gibt eine Reihe von Anbietern, die für die Anwendungstypen und die gängigen Arten von Streaming-Szenarien spezifisch sind. Dazu gehören Anbieter wie Video, Audio sowie Audio- und Video-Multitrack. Für jedes dieser Szenarien gibt es Standardwerte, 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 (AWS Credentials-API) und Callbacks, bei denen das Streaming bei Fehlern wiederholt werden soll. Die Callback Provider-API benötigt für die Konfiguration eine Reihe von Argumenten, wie z. B. die und die Autorisierungsinformationen. AWS-Region Dies erfolgt mithilfe von IoT-Zertifikaten oder mithilfe von AWS AccessKeyId, SecretKey, oder 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. Außerdem optimiert es die Paketierung der Frame-Metadaten in der PIC-Framestruktur, 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 verwaltet Rückrufe, um über Kinesis Video Streams Streams-Ereignisse zu berichten.
- KinesisVideoStream— Stellt Informationen zu den Parametern des Videostreams wie Name, Datenaufbewahrungsdauer und Medieninhaltstyp dar.

Fügt Medien in den Stream ein

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.

Verfahren: Verwenden Sie das C-Producer-SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video Streams Streams-Client und Medienquellen in einer C-Anwendung verwenden, um H.264-kodierte Videoframes an Ihren Kinesis-Videostream zu senden.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

- Laden Sie den C-Producer-Bibliothekscode herunter
- Schreiben und untersuchen Sie den Code
- Führen Sie den Code aus und überprüfen Sie ihn

Voraussetzungen

Bevor Sie das C Producer SDK einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Anmeldeinformationen — Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer Profildatei für AWS Anmeldeinformationen eingerichtet haben. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein.

  Weitere Informationen finden Sie unter AWS Anmeldeinformationen und Region für die Entwicklung einrichten.
- Integration des Zertifikatsspeichers — Die Kinesis Video Streams Streams-Producer-Bibliothek muss eine Vertrauensstellung mit 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
  - 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:
  - Git: `sudo apt install git`
  - CMake: `sudo apt install cmake`
  - G ++: `sudo apt install g++`
  - pkg-konfiguration: `sudo apt install pkg-config`
  - OpenJDK: `sudo apt install openjdk-8-jdk`
  - Legen Sie die `JAVA_HOME`-Umgebungsvariable fest: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Laden Sie den C-Producer-Bibliothekscode herunter

In diesem Abschnitt laden Sie die Low-Level-Bibliotheken herunter. Die Voraussetzungen und anderen Details für dieses Beispiel finden Sie unter the section called “C++”.

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 und eine Build-Umgebung installieren. CMake
   > 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).

Schreiben und untersuchen Sie den Code

In diesem Abschnitt untersuchen Sie den Code der Beispielanwendung `KvsVideoOnlyStreamingSample.c` im `samples` Ordner des Repositorys https://github.com/awslabs/amazon-kinesis-video-streams-producer-c. GitHub Sie haben diesen Code im vorherigen Schritt heruntergeladen. Dieses Beispiel zeigt, wie Sie die C-Producer-Bibliothek verwenden, um H.264-kodierte Videoframes 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 von privaten Codec-Daten und Versetzen des Streams in den Status READY.
- Hauptschleife:
  - Abrufen des Frame von der Medien-Pipeline mit dem Zeitstempel und Flags.
  - Senden des Frames an den. KinesisVideoStream
- Abbau:
  - Stoppen (Synchronisieren) KinesisVideoStream, Freigeben KinesisVideoStream, KinesisVideoClient Freigeben.

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 Anmeldeinformationen zu erstellen. AWS

```
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 Ihren Gerätespeicher 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));
```

Führen Sie den Code aus und überprüfen Sie ihn

Gehen Sie wie folgt vorthe section called “C++”, um den Code für auszuführen und zu überprüfen:

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`- ermöglicht die Berichterstattung über die Netzabdeckung.
- `-DCOMPILER_WARNINGS`- aktiviert alle Compiler-Warnungen.
- `-DADDRESS_SANITIZER`- baue mit AddressSanitizer.
- `-DMEMORY_SANITIZER`- baue mit MemorySanitizer.
- `-DTHREAD_SANITIZER`- baue mit ThreadSanitizer.
- `-DUNDEFINED_BEHAVIOR_SANITIZER`- baue mit UndefinedBehaviorSanitizer.
- `-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 Befehl 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-kodierte Videoframes innerhalb des Ordners `samples/h264SampleFrames` an Kinesis Video Streams. Mit dem folgenden Befehl werden die Videoframes zehn Sekunden lang in einer Schleife an Kinesis Video Streams gesendet:

   ```
   ./kinesis_video_cproducer_video_only_sample YourStreamName 10
   ```

   Wenn Sie H.264-kodierte 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 Sie das C++-Producer-SDK auf dem 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 Schritte beinhalten auch, wie Sie die Installation mithilfe der GStreamer Demo-Anwendung überprüfen können.

Themen

- Voraussetzungen
- Erstellen Sie einen IAM-Benutzer mit der Berechtigung, in Kinesis Video Streams zu schreiben
- Verbinden Sie Ihren Raspberry Pi mit Ihrem Wi-Fi-Netzwerk
- Stellen Sie eine Remoteverbindung zu Ihrem Raspberry Pi her
- Konfigurieren Sie die Raspberry Pi-Kamera
- Installieren Sie die Softwarevoraussetzungen
- Laden Sie das Kinesis Video Streams C++ Producer SDK herunter und erstellen Sie es
- Streamen Sie Video in Ihren Kinesis-Videostream
- Spielen Sie Medien aus Ihrem Kinesis-Videostream ab
- Behebung von Build-Problemen im C++-Producer-SDK für Raspberry Pi

Voraussetzungen

Bevor Sie das C++-Producer-SDK auf Ihrem Raspberry Pi einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Ein Raspberry Pi-Gerät mit der folgenden Konfiguration:
  - Board-Version: 3 Modell B oder höher.
  - Ein angeschlossenes Kameramodul oder eine angeschlossene USB-Kamera (Webcam).
  - 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.
- Und AWS-Konto mit einem Kinesis-Videostream. Weitere Informationen finden Sie unter Erste Schritte mit Kinesis-Videostreams.

Erstellen Sie einen IAM-Benutzer mit der Berechtigung, in Kinesis Video Streams zu schreiben

Falls Sie dies noch nicht getan haben, richten Sie einen AWS Identity and Access Management (IAM-) Benutzer ein, der berechtigt ist, in einen Kinesis-Videostream zu schreiben.

Diese Verfahren sollen Ihnen helfen, schnell mit der Verwendung eines AWS Zugriffsschlüsselpaars zu beginnen. Geräte können X.509-Zertifikate verwenden, um eine Verbindung herzustellen AWS IoT. Weitere Informationen the section called “Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe AWS IoT” 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 -producer festlegen die Option Bestehende Richtlinien direkt anhängen 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 gibt Ihrem Benutzer die Erlaubnis, Daten zu erstellen und in Kinesis-Videostreams zu schreiben.

    JSON

```
{
  "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 Sie Ihren Raspberry Pi mit Ihrem Wi-Fi-Netzwerk

Fahren Sie bei Anschluss eines Monitors und einer Tastatur mit Konfigurieren Sie die Raspberry Pi-Kamera.

Diese Anweisungen sollen Ihnen helfen, Ihren Raspberry Pi einzurichten, wenn er im Headless-Modus betrieben wird, d. h. ohne angeschlossene Tastatur, Monitor oder Netzwerkkabel. Folgen Sie den nachstehenden Anweisungen, um Ihren Raspberry Pi so zu konfigurieren, dass er automatisch versucht, eine Verbindung zum angegebenen Netzwerk herzustellen, sodass Ihr Host-Computer eine SSH-Verbindung herstellen kann.

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.

Stellen Sie eine Remoteverbindung zu Ihrem Raspberry Pi her

Fahren Sie bei Anschluss eines Monitors und einer Tastatur mit Konfigurieren Sie die Raspberry Pi-Kamera.

Diese Anweisungen sollen Ihnen helfen, Ihren Raspberry Pi einzurichten, wenn er im Headless-Modus betrieben wird, d. h. ohne angeschlossene Tastatur, Monitor oder Netzwerkkabel. Folgen Sie den nachstehenden Anweisungen, um Ihren Raspberry Pi in Ihrem Netzwerk zu finden und von Ihrem Host-Computer aus eine SSH-Verbindung herzustellen.

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 Sie die Raspberry Pi-Kamera

Gehen Sie wie folgt vor, um das Raspberry Pi-Kameramodul so zu konfigurieren, dass es Video vom Gerät an einen Kinesis-Videostream sendet.

> **Note**
>
> Wenn Sie eine USB-Webcam verwenden, fahren Sie mit fortthe section called “Installieren Sie die Softwarevoraussetzungen”.

Camera module 1

Folgen Sie diesen Anweisungen, um die Moduldatei zu aktualisieren, die Kameraschnittstelle zu aktivieren und die Funktionalität Ihrer Kamera zu überprüfen. Durch das Aktualisieren der Moduldatei wird dem Raspberry Pi mitgeteilt, welche Kernelmodule beim Booten geladen werden sollen. Der Kameratreiber wird standardmäßig nicht geladen, um Systemressourcen auf Raspberry Pi-Geräten zu schonen, die keine Kamera verwenden.

1. Öffnen Sie einen Editor, um Änderungen an der Moduldatei vorzunehmen. Öffnen Sie das Terminal und bearbeiten Sie die Datei mit dem `nano` Editor mit dem folgenden Befehl:

   ```
   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 schließen Sie den Editor. Um mit dem `nano` Editor zu speichern und zu beenden, verwenden Sie 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 „Schnittstellenoptionen“, „Legacy Camera“. In älteren Versionen des Raspbian-Betriebssystems befindet sich diese Menüoption möglicherweise unter Schnittstellenoptionen, 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 und zeigt `test.jpg` Informationsmeldungen an.

Camera module 2 or 3

Wenn Sie ein Kameramodul 2 verwenden, verwenden Sie entweder `bcm2835-v4l2` (Legacy) oder `libcamera` (Modern). Der `libcamera` Stack wird jedoch für eine bessere Unterstützung und bessere Funktionen empfohlen. Gehen Sie wie folgt vor, um zu bestätigen, dass er up-to-date auf Ihrem System installiert `libcamera` ist.

1. libcamera sollte auf Ihrem Raspberry Pi vorinstalliert sein. Suchen Sie nach Updates und aktualisieren Sie auf die neueste Version, um Bugfixes und Sicherheitsupdates zu erhalten. Öffnen Sie das Terminal und geben Sie die folgenden Befehle ein:

```
sudo apt-get update
sudo apt-get upgrade
```

2. Starten Sie Ihr System neu, damit die Updates wirksam werden.

```
sudo reboot
```

3. Testen Sie Ihre Kamera. Diese Anwendung startet einen Kamera-Vorschau-Stream und zeigt ihn auf dem Bildschirm an.

```
libcamera-hello
```

Wenn Sie Probleme mit Ihrem Kameramodul haben, finden Sie Informationen zur Fehlerbehebung in der Raspberry Pi-Dokumentation.

Installieren Sie die Softwarevoraussetzungen

Für das C++ Producer SDK müssen Sie die folgenden Softwarevoraussetzungen auf dem Raspberry Pi installieren.

1. Aktualisieren Sie die Paketliste und installieren Sie die Bibliotheken, die für die Erstellung des SDK benötigt werden. Öffnen Sie das Terminal und geben Sie die folgenden Befehle ein:

```
sudo apt-get update
sudo apt-get 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. Wenn Sie den `libcamera` Stack verwenden, installieren Sie auch das `libcamerasrc` GStreamer Plugin. Dieses GStreamer Plugin ist standardmäßig nicht installiert.

```
sudo apt-get install gstreamer1.0-libcamera
```

3. 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
```

Laden Sie das Kinesis Video Streams C++ Producer SDK herunter und erstellen Sie es

Gehen Sie wie folgt vor, um das Kinesis Video Streams C++ Producer SDK herunterzuladen und zu erstellen. Vergewissern Sie sich, dass Sie die erforderlichen Softwarekomponenten installiert haben. Die entsprechenden Schritte finden Sie unterthe section called “Installieren Sie die Softwarevoraussetzungen”.

1. Navigieren Sie zum Download-Verzeichnis. Öffnen Sie ein Terminal und wechseln Sie zu Ihrem bevorzugten Download-Verzeichnis.

   Zum Beispiel:

   ```
   cd ~/Downloads
   ```

2. Klonen Sie das SDK-Repository. Verwenden Sie den `git clone` Befehl, um das SDK aus dem GitHub Repository herunterzuladen. Type:

   ```
   git clone https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp.git --single-branch -b master kvs-producer-sdk-cpp
   ```

   Mit diesem Befehl wird nur ein einziger Zweig (der `master` Zweig) geklont, wodurch Größe und Dauer des Downloads reduziert werden. Außerdem platziert er den heruntergeladenen Inhalt in einem Ordner namens `kvs-producer-sdk-cpp` innerhalb des aktuellen Verzeichnisses.

3. Überprüfen Sie den Download. Sobald der Klonvorgang abgeschlossen ist, listen Sie den Inhalt des `kvs-producer-sdk-cpp` Ordners auf, um zu überprüfen, ob das SDK heruntergeladen wurde.

   ```
   ls kvs-producer-sdk-cpp
   ```

4. Bereiten Sie ein Build-Verzeichnis vor. Type:

   ```
   mkdir -p kvs-producer-sdk-cpp/build
   cd kvs-producer-sdk-cpp/build
   ```

5. Konfigurieren Sie den Build. Führen Sie den folgenden `cmake` Befehl aus, um die Build-Umgebung mit bestimmten Optionen zu konfigurieren:

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -
DALIGNED_MEMORY_MODEL=ON
```

CMakeverwendet die folgenden Optionen, um die entsprechenden zu generierenMakefiles:

- Verwendung des Projektordners (`..`) als Quellverzeichnis.
- Das aktuelle Verzeichnis (`.`) (`build/`) wird für die Build-Ausgabe verwendet.
- `-DBUILD_GSTREAMER_PLUGIN=ON`ermöglicht das Erstellen des GStreamer Plugins kvssink.
- `-DBUILD_DEPENDENCIES=OFF`deaktiviert das Erstellen externer Abhängigkeiten aus dem Quellcode. Das Projekt sucht und verwendet die externen Abhängigkeiten, die in einem vorherigen Schritt installiert wurden.
- `-DALIGNED_MEMORY_MODEL=ON`deaktiviert das unausgerichtete Speichermodell. Ein unausgerichteter Speicherzugriff wird von bestimmten Raspberry Pi-Geräten nicht unterstützt.

> **Note**
>
> Eine vollständige Liste der CMake Argumente finden Sie unterthe section called “Laden Sie den Code herunter und konfigurieren Sie ihn”.

6. Erstellen Sie das Projekt. Verwenden Sie nach der Konfiguration des Builds den `make` Befehl zum Kompilieren mit dem Befehl `Makefile` generated by`cmake`.

```
make -j$(nproc)
```

Das `-j` Argument to `make` ermöglicht es, mehrere Kompilierungsjobs parallel auszuführen. Um die Build-Zeiten zu reduzieren, verwenden Sie den `nproc` Befehl, um die Anzahl der CPU-Kerne auf Ihrem Raspberry Pi dynamisch zu berechnen.

7. Bestätigen Sie, dass dies vorhanden `libgstkvssink.so` ist.

   Listet die Dateien im aktuellen Verzeichnis auf.

   Aufforderung:

```
ls
```

Antwort:

```
CMakeCache.txt       dependency                          kvs_gstreamer_sample
CMakeFiles           kvs_gstreamer_audio_video_sample    kvssink_gstreamer_sample
Makefile             kvs_gstreamer_file_uploader_sample  libKinesisVideoProducer.so
cmake_install.cmake  kvs_gstreamer_multistream_sample    libgstkvssink.so
```

8. Bestätigen Sie, dass das geladen werden GStreamer kann`kvssink`.

   Setzen Sie die `GST_PLUGIN_PATH` Umgebungsvariable auf das Verzeichnis, das enthält`libgstkvssink.so`.

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

   Habe GStreamer geladen`kvssink`:

   ```
   gst-inspect-1.0 kvssink
   ```

   Sie sollten einige Unterlagen über sehen`kvssink`. Verwenden Sie die Pfeiltasten, um zu navigieren, und drücken Sieq, um den Vorgang zu beenden.

9. (Optional) Aktualisieren Sie das Startskript Ihrer Shell um die Einstellung der `GST_PLUGIN_PATH` Umgebungsvariablen. Dadurch `GST_PLUGIN_PATH` wird sichergestellt, dass sie während einer neuen Terminalsitzung richtig eingestellt ist. Auf Raspberry Pi-Geräten lautet das Startskript der Shell`~/.bashrc`.

   Führen Sie den folgenden Befehl aus, um den Befehl an das Ende des Startskripts der Shell anzuhängen.

   ```
   echo "export GST_PLUGIN_PATH=~/Downloads/kvs-producer-sdk-cpp/build" >> ~/.bashrc
   ```

   Geben Sie Folgendes ein, um das Startskript der Shell auszuführen, oder schließen Sie die aktuelle Shell und öffnen Sie eine neue.

   ```
   source ~/.bashrc
   ```

   Bestätigen Sie, dass das gesetzt `GST_PLUGIN_PATH` ist, und Sie können es laden`kvssink`.

```
echo $GST_PLUGIN_PATH
```

```
gst-inspect-1.0 kvssink
```

Streamen Sie Video in Ihren Kinesis-Videostream

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 Sie einen IAM-Benutzer mit der Berechtigung, in Kinesis Video Streams zu schreiben erstellt wurden
- GStreamer ist in der Lage, das `kvssink` Plugin zu finden. Weitere Informationen finden Sie unter the section called “Laden Sie das C++-Producer-SDK herunter und erstellen Sie es”.

1. Stellen Sie die Anmeldeinformationen und die Region ein.

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

   Weitere Authentifizierungsmethoden finden Sie unterthe section called “Geben Sie Anmeldeinformationen ein für `kvssink`”.

   > **Note**
   >
   > Das C++-Producer-SDK verwendet standardmäßig die Region USA West (Oregon`us-west-2`) (). Um die Standardeinstellung zu verwenden, AWS-Region erstellen Sie Ihren Kinesis-Videostream in der Region USA West (Oregon).
   > Um eine andere Region für Ihren Kinesis-Videostream zu verwenden, legen Sie die folgende Umgebungsvariable auf Ihre Region fest (z. B.*us-east-1*):
   >
   > ```
   > export AWS_DEFAULT_REGION=us-east-1
   > ```

2. Wählen Sie je nach Eingabemedium eine der folgenden Optionen aus:

Sample GStreamer video

Diese GStreamer Pipeline generiert einen Live-Testvideostream mit einem Standardtestmuster, das mit 10 Bildern pro Sekunde und einer Auflösung von 640 x 480 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 Keyframes werden höchstens alle 10 Frames generiert, was zu einer Fragmentdauer (auch bekannt als Größe einer Bildergruppe (GoP)) von 1 Sekunde. kvssink nimmt 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 beenden, wählen Sie das Terminalfenster aus und drücken Sie STRG+C.

Die GStreamer Beispielvideopipeline sieht wie folgt aus:

USB web cam

Führen Sie den folgenden Befehl aus, damit Ihre USB-Kamera GStreamer automatisch erkannt wird:

```
gst-launch-1.0 autovideosrc \
  ! videoconvert \
  ! video/x-raw,format=I420,width=640,height=480 \
  ! x264enc bframes=0 key-int-max=45 tune=zerolatency byte-stream=true speed-
preset=ultrafast \
  ! h264parse \
  ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
  ! kvssink stream-name="YourStreamname"
```

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

Anstatt die GStreamer automatische Erkennung zu verwenden, können Sie es `v4l2src` mit einer bestimmten Gerätekennung verwenden. Führen Sie den folgenden Befehl aus:

```
gst-device-monitor-1.0
```

In der Ausgabe sehen Sie einige Geräte und den Beginn einer GStreamer Pipeline zur Verwendung des Geräts:

```
Device found:

    name  : H264 USB Camera: USB Camera
    class : Video/Source
    caps  : video/x-h264, stream-format=(string)byte-stream,
 alignment=(string)au, width=(int)1920, height=(int)1080, pixel-aspect-
ratio=(fraction)1/1, colorimetry=(string){ 2:4:7:1 }, framerate=(fraction)
{ 30/1, 25/1, 15/1 };
            ...
    properties:
        device.path = /dev/video4
        udev-probed = false
        device.api = v4l2
        v4l2.device.driver = uvcvideo
        v4l2.device.card = "H264\ USB\ Camera:\ USB\ Camera"
        v4l2.device.bus_info = usb-3f980000.usb-1.3
        v4l2.device.version = 265767 (0x00040e27)
        v4l2.device.capabilities = 2216689665 (0x84200001)
        v4l2.device.device_caps = 69206017 (0x04200001)
    gst-launch-1.0 v4l2src device=/dev/video4 ! ...
```

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

Raspberry Pi camera module 1

Wenn Sie das Pi-Kameramodul 1 oder das Pi-Kameramodul 2 mit verwenden`bcm2835-v4l2`, verwenden Sie Folgendes:

```
gst-launch-1.0 v4l2src device=/dev/video0 \
```

```
  ! videoconvert \
  ! video/x-raw,format=I420,width=640,height=480 \
  ! x264enc bframes=0 key-int-max=45 bitrate=500 tune=zerolatency \
  ! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
  ! kvssink stream-name="YourStreamname"
```

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

Raspberry Pi camera module 2 or 3

Wenn Sie den modernen `libcamera` Stack verwenden, verwenden Sie die folgende GStreamer Pipeline:

```
gst-launch-1.0 libcamerasrc \
  ! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \
  ! videoconvert \
  ! x264enc speed-preset=ultrafast tune=zerolatency byte-stream=true key-int-
max=75 \
  ! video/x-h264,level='(string)4' \
  ! h264parse \
  ! video/x-h264,stream-
format=avc,alignment=au,width=640,height=480,framerate=30/1 \
  ! kvssink stream-name="YourStreamname"
```

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

Verwenden Sie Hardware

Einige Raspberry Pi-Modelle sind mit hardwarebeschleunigten H.264-Encodern ausgestattet. Sie können sie anstelle von `x264enc`, einem Software-Encoder, verwenden.

1. Stellen Sie sicher, dass die GStreamer Plugins installiert sind:

```
sudo apt-get install gstreamer1.0-tools gstreamer1.0-plugins-bad
```

2. Type:

```
gst-inspect-1.0 | grep h264
```

Stellen Sie fest, ob die folgenden Elemente verfügbar sind:

- omxh264enc
- v4l2h264enc

Wenn sie verfügbar sind, können Sie sie verwenden. Hier sind einige Pipeline-Beispiele, die diese Elemente verwenden:

`omxh264enc`:

```
gst-launch-1.0 v4l2src device=/dev/video0 \
  ! videoconvert \
  ! video/x-raw,format=I420,width=640,height=480 \
  ! omxh264enc control-rate=2 target-bitrate=512000 periodicity-idr=45 inline-
header=FALSE \
  ! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
  ! kvssink stream-name="raspberry"
```

`v4l2h264enc`und**`v4l2convert`**:

```
gst-launch-1.0 libcamerasrc \
  ! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \
  ! v4l2convert \
  ! v4l2h264enc extra-controls="controls,repeat_sequence_header=1" \
  ! video/x-h264,level='(string)4' \
  ! h264parse \
  ! video/x-h264,stream-format=avc,alignment=au,width=640,height=480,framerate=30/1
 \
  ! kvssink stream-name="test-stream"
```

Probleme mit der Laufzeit

Im Folgenden finden Sie einige häufig auftretende Laufzeitprobleme und deren Behebung.

Kein solches Element „xxxxxxxxx“

Wenn du eine Fehlermeldung wie die folgende erhältst, bedeutet das, dass dir ein GStreamer Plugin fehlt:

```
WARNING: erroneous pipeline: no element "videoconvert"
```

Auflösung

Ermitteln Sie anhand des fehlenden Elements die entsprechende Aktion:

- `kvssink`: Siehe the section called “Laden Sie das C++-Producer-SDK herunter und erstellen Sie es”.
- `libcamerasrc`: Informationen the section called “Fehler „Die Aktivierung des Pufferpools ist fehlgeschlagen“” zur Installation des `libcamerasrc` GStreamer Elements finden Sie unter.
- `omxh264encoderv4l2h264enc`:

  Folgen Siethe section called “Installieren Sie die Softwarevoraussetzungen”, um alle GStreamer Bibliotheken zu installieren. Wenn Sie sie alle installiert haben und diese Elemente nicht angezeigt werden, bedeutet dies, dass Ihr Raspberry Pi nicht über die erforderliche Hardware verfügt. Verwenden Sie `x264enc` stattdessen den Software-Encoder.
- Andere: Folgen Sie diesen Anweisungenthe section called “Installieren Sie die Softwarevoraussetzungen”, um alle GStreamer Bibliotheken zu installieren. Verschiedene GStreamer Elemente befinden sich in den verschiedenen GStreamer Plugin-Gruppen (gut, schlecht, hässlich). Stellen Sie also sicher, dass Sie sie alle installieren.

Fehler „Die Aktivierung des Pufferpools ist fehlgeschlagen“

Wenn Sie einen Fehler wie den folgenden erhalten, bedeutet dies, dass die verwendete Pipeline verwendet wird`v4l2src`, aber sie sollte `libcamerasrc` stattdessen verwenden.

```
ERROR bufferpool gstbufferpool.c:572:gst_buffer_pool_set_active:source:pool0:src start
 failed
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Failed to allocate
 required memory.
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Buffer pool
 activation failed
WARN basesrc gstbasesrc.c:3352:gst_base_src_prepare_allocation: Subclass failed to
 decide allocation
Error received from element source: Failed to allocate required memory.
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: Internal data stream error.
Debugging information: ../sys/v4l2/gstv4l2src.c(976): gst_v4l2src_decide_allocation
 (): /GstPipeline:live-kinesis-pipeline/GstV4l2Src:source:
```

```
Buffer pool activation failed
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: streaming stopped, reason not-
negotiated (-4)
```

Wenn Sie beispielsweise die folgende Pipeline verwenden, bei der das Kameramodul 2 nicht `libcamerasrc` installiert GStreamer ist, kann dieser Fehler auftreten, wenn Sie versuchen, automatisch zu erkennen, welche Elemente verwendet werden sollen.

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

Auflösung

Stellen Sie sicher, dass das installiert `libcamerasrc` ist, und verwenden Sie es nicht als `v4l2src` Quellelement. Geben Sie Folgendes ein, um das `libcamerasrc` GStreamer Element zu installieren:

```
sudo apt-get update
sudo apt-get install gstreamer1.0-libcamera
```

Wenn Sie `libcamerasrc` das `autovideosrc` Element nach der Installation verwenden, GStreamer sollten Sie automatisch zur Verwendung der richtigen Quelle wechseln, `libcamerasrc` anstatt von`v4l2src`.

Busfehler

Wenn Sie kurz nach dem Start einen Busfehler erhalten `kvssink` (normalerweise etwa zu dem Zeitpunkt, zu dem der HTTP-Aufruf `PutMedia` abgeschlossen ist), bedeutet dies, dass Ihr Raspberry Pi keinen unausgerichteten Speicherzugriff unterstützt. Die Protokolle werden wie folgt aussehen:

```
INFO Camera camera.cpp:1197 configuring streams: (0) 640x480-YUV420
INFO RPI pisp.cpp:1450 Sensor: /base/axi/pcie@120000/rp1/i2c@88000/imx708@1a - Selected
 sensor format: 1536x864-SBGGR10_1X10 - Selected CFE format: 1536x864-PC1B
[INFO ] kinesisVideoStreamFormatChanged(): Stream format changed.
[DEBUG] setRequestHeader(): Appending header to request: user-agent -> AWS-SDK-KVS-CPP-
CLIENT/3.4.2/1.5.3 GCC/12.2.0 Linux/6.6.51+rpt-rpi-v8 aarch64 CPPSDK
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-stream-name -> demo-
stream
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-producer-start-
timestamp -> 1732012345.678
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-
acknowledgment-required -> 1
```

```
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-timecode-type
 -> ABSOLUTE
[DEBUG] setRequestHeader(): Appending header to request: transfer-encoding -> chunked
[DEBUG] setRequestHeader(): Appending header to request: connection -> keep-alive
[INFO ] putStreamResultEvent(): Put stream result event. New upload handle 0
[WARN ] notifyDataAvailable(): [demo-stream] Failed to un-pause curl with error: 43.
 Curl object 0xe2f6f418
Bus error
```

Kinesis Video Streams PIC verwendet unausgerichteten Speicherzugriff, um die Speichernutzung zu optimieren, was nicht von allen Geräten unterstützt wird.

Auflösung

Um das SDK im Modus „Aligned Memory Access“ zu verwenden, müssen Sie das `ALIGNED_MEMORY_MODEL` CMake Flag `ON` beim Kompilieren explizit auf setzen`kvssink`, da es standardmäßig auf eingestellt ist. `OFF` Eine detailliertere the section called “Laden Sie das C++-Producer-SDK herunter und erstellen Sie es” Anleitung finden Sie unter.

Der Zeitstempel friert ein und die Pipeline gerät ins Stocken

Bei `x264enc` der Verwendung in einer GStreamer Pipeline können Situationen auftreten, in denen sich die Timeline der Pipeline erheblich verlangsamt oder innerhalb weniger Sekunden vollständig zum Stillstand kommt.

Dies liegt daran, dass die `x264enc` Standardeinstellungen zu einer hohen Kodierungslatenz führen können, die die Kapazität des Standardeingabepuffers übersteigt. Infolgedessen füllt sich der Eingabepuffer, wodurch Upstream-Elemente blockiert werden und die Pipeline zum Stillstand kommt.

Weitere Informationen finden Sie in der GStreamer -Dokumentation.

Auflösung

Konfigurieren Sie `x264enc` mit der `zerolatency` Tuning-Option. Dadurch wird die Latenz bei der Kodierung erheblich reduziert, da für Echtzeitszenarien optimiert wird und sichergestellt wird, dass Frames schneller verarbeitet und ausgegeben werden.

Beispielkonfiguration:

```
... ! x264enc tune=zerolatency byte-stream=true speed-preset=ultrafast bframes=0 key-
int-max=60 ! ...
```

> **Note**
>
> Diese Lösung verhindert zwar effektiv ein Blockieren der Pipeline, kann sich jedoch auf die Effizienz und Qualität der Kodierung auswirken. Für Szenarien, die sowohl eine geringe Latenz als auch eine hohe Qualität erfordern, sollten Sie alternative Ansätze in Betracht ziehen, z. B. Hardwareoptimierungen oder die Suche nach einer Webcam, die H.264 direkt ausgibt, und diesen Kodierungsschritt überspringen.
> Weitere Informationen finden Sie unter the section called “Verwenden Sie Hardware”.

Es können nicht mehrere Pipelines gleichzeitig von demselben **`v4l2`** Gerät aus ausgeführt werden

Auf Geräte wie `/dev/video0` kann jeweils nur von einem Prozess zugegriffen werden. Wenn mehrere Prozesse gleichzeitig versuchen, darauf zuzugreifen, wartet der zweite, bis der erste abgeschlossen ist.

Auflösung

Erstellen Sie ein Loopback-Gerät, sodass mehrere Prozesse die Loopback-Schnittstelle gleichzeitig verwenden können. Weitere Informationen finden Sie unter Stack Exchange.

Interner Datenstream-Fehler

Wenn Sie eine GStreamer Pipeline erstellen, verbinden Sie Elemente, indem Sie das Quell-Pad eines Elements mit dem Sink-Pad eines anderen Elements verknüpfen. Dieser Verbindungsprozess ermöglicht den Datenfluss vom Quellelement zum Senkenelement und bildet so eine Datenpipeline.

Die Fehlermeldung „Pad Link failed“ im Protokoll weist darauf hin, dass beim Versuch, eine Verbindung (Link) zwischen den Pads zweier Elemente in Ihrer Pipeline herzustellen, ein Problem GStreamer aufgetreten ist.

```
Pad link failed
Error received from element udpsrc0: Internal data stream error.
```

Auflösung

Stellen Sie fest, welche Elemente nicht miteinander verknüpft werden können. Um den Umfang der Pipeline einzugrenzen, entfernen Sie Elemente aus der Pipeline. Ersetzen Sie das Element ganz rechts durch Elemente `fakesink` und entfernen Sie sie nacheinander.

Möglicherweise müssen Sie die Capsfilter-Elemente anpassen und and/or ändern, welche Elemente Ihre Pipeline verwendet.

Häufig wird nach einem `framerate` oder gefragt`resolution`, das die Kamera nicht unterstützt. Verwenden Sie `gst-device-monitor-1.0` im Terminal, um die unterstützten `frameratesresolutions`, und zu erhalten`formats`. Sie können das GStreamer Videoscale-Element verwenden, um die Videoauflösung anzupassen, und mit Videorate können Sie die Videobildrate anpassen.

Geben `gst-inspect-1.0 element-name` Sie in das Terminal ein, um die unterstützten Formate für ein einzelnes GStreamer Element zu überprüfen.

Spielen Sie Medien aus Ihrem Kinesis-Videostream ab

Öffnen Sie die Kinesis Video Streams Streams-Konsole und wählen Sie den Stream-Namen für den Stream aus, den Sie erstellt haben.

Der vom Raspberry Pi gesendete Videostream erscheint in der Konsole.

> **Note**
>
> Es kann einige Sekunden dauern, bis das Video in der Konsole angezeigt wird.

Sobald der Stream abgespielt wird, können Sie mit den folgenden Funktionen in der Konsole experimentieren:

- Führen Sie mit den Navigationssteuerelementen im Bereich Video preview (Video-Vorschau) einen Rück- oder Vorlauf des Streams durch.
- Überprüfen Sie im Abschnitt Stream-Informationen den Codec, die Auflösung und die Bitrate des Streams. Die Werte für Auflösung und Bitrate sind auf dem Raspberry Pi bewusst niedrig eingestellt, um die Bandbreitennutzung für dieses Tutorial zu minimieren.

  Um die CloudWatch Amazon-Metriken anzuzeigen, die für Ihren Stream erstellt werden, wählen Sie Stream-Metriken anzeigen in 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 Serverseitige Verschlüsselung, dass Ihre Daten im Ruhezustand mit einem Schlüssel verschlüsselt werden, der von AWS Key Management Service (AWS KMS) verwaltet wird.

Probleme bei der Wiedergabe

Im Folgenden finden Sie einige häufig auftretende Wiedergabeprobleme und deren Behebung.

Keine Medien, aber es gibt PERSISTENTE Acks in den Protokollen

Wenn in den Protokollen PERSISTED Acks angezeigt werden, hat Kinesis Video Streams die Medien, die von hochgeladen wurden, erfolgreich aufgenommen und gespeichert. `kvssink` Von Kinesis Video Streams erhaltene Acks sehen so aus. Sehen Sie sich in der JSON-Datei den Wert für den `"EventType"` Schlüssel an.

```
{"EventType":"RECEIVED","FragmentTimecode":252200,"FragmentNumber":"123456789012345678901234567
{"EventType":"BUFFERING","FragmentTimecode":252467,"FragmentNumber":"123456789012345678901234567
{"EventType":"RECEIVED","FragmentTimecode":252467,"FragmentNumber":"123456789012345678901234567
{"EventType":"BUFFERING","FragmentTimecode":253000,"FragmentNumber":"123456789012345678901234567
{"EventType":"PERSISTED","FragmentTimecode":252200,"FragmentNumber":"123456789012345678901234567
{"EventType":"PERSISTED","FragmentTimecode":252467,"FragmentNumber":"123456789012345678901234567
```

Auflösung

Warten Sie in der Kinesis Video Streams Streams-Konsole ein oder zwei Minuten und verwenden Sie dann den Doppelpfeil nach rechts. Wenn kein Medium angezeigt wird, überprüfen Sie, ob Ihr Stream in die richtige Region gesendet wird, und überprüfen Sie die Schreibweise des Streamnamens. Du findest diese Informationen in den Protokollen.

the section called “Geben Sie eine Region an `kvssink`”Weitere Informationen darüber, wie kvssink bestimmt, welche Region verwendet werden soll, finden Sie unter.

Das Laden des Mediums dauert sehr lange AWS Management Console

> **Important**
>
> Das Wiedergabeerlebnis auf der Konsole unterscheidet sich vom Wiedergabeerlebnis mit HLS und DASH. Verwenden Sie auch die von Media Player gehostete Beispielwebseite, GitHub um die Wiedergabe zu testen. Den Quellcode für die Webseite finden Sie hier.

Das Laden von Medien in der Konsole kann aufgrund einer schlechten Netzwerkbandbreite oder eines eingeschränkten Geräts langsam sein. Dies kann aber auch mit der Videokodierung und Fragmentierung zusammenhängen.

Grundlagen der Videokodierung:

- H.264- und H.265-Encoder verwenden Key-Frames (I-Frames) und Predicted-Frames (P-Frames) für eine effiziente Komprimierung.
- Key-Frames enthalten vollständige Bilddaten, während P-Frames nur Änderungen gegenüber früheren Frames enthalten.
- Das „Key-Frame-Intervall“ bestimmt, wie oft Keyframes im Videostream vorkommen.

Fragmentierung beim Streaming:

- In Kinesis Video Streams beginnen neue Fragmente mit jedem Keyframe. Weitere Informationen finden Sie unter the section called “Datenmodell”.
- Die Fragmentlänge (in Sekunden) kann wie folgt geschätzt werden: Key-Frame-Intervall — Bildrate

  Beispiel:

  Für einen Stream mit einem Keyframe-Intervall von 30 Bildern pro Sekunde und einer Bildrate von 15 Bildern pro Sekunde:

  Fragmentlänge = 30 ÷15 = 2 Sekunden

Aufgrund größerer Keyframe-Intervalle erhöhen längere Fragmente die Latenz bei Streaming-Medien.

Auflösung

Um die Ladezeiten zu verbessern, sollten Sie erwägen, das Keyframe-Intervall zu reduzieren. Dadurch entstehen kürzere Fragmente, was die Latenz verringert, aber auch die Größe der Videodatei erhöht.

Für das `x264enc` GStreamer Element können Sie das Keyframe-Intervall explizit über die `key-int-max` folgende Eigenschaft festlegen:

```
x264enc bframes=0 key-int-max=60
```

Beachten Sie bei der Überprüfung der Protokollausgabe, wie oft der hochladende Client Daten ACKs von Kinesis Video Streams empfängt. Je mehr Keyframes generiert werden, desto mehr ACKs werden zurückgegeben.

Das Medium ist verzerrt oder weist Artefakte auf

Um dieses Problem zu beheben, stellen Sie sicher, dass alle Kabel fest angeschlossen sind. Überprüfen Sie die Ausgabe von `libcamera-hello` (oder `raspistill` für ältere Pi-Kameras) für Kameramodule.

Ersetzen Sie es in Ihrer GStreamer Pipeline `kvssink` durch `autovideosink` oder `matroskamux` und`filesink`. Zum Beispiel:

```
... x264enc tune=zerolatency speed-preset=ultrafast bframes=0 key-int-max=60 byte-
stream=true ! h264parse ! matroskamux ! filesink location=output.mkv
```

Prüfen Sie anhand der Ausgabedatei `filesink` oder des Media Players, der sich bei der Verwendung `autovideosink` öffnet, ob die Artefakte ebenfalls vorhanden sind.

Überprüfen Sie auch die Ausgabe der folgenden Pipeline:

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

Durch das Hinzufügen von Elementen zu Ihrer Pipeline, z. B. Dewarp, können Sie die Ausgaben der Fischaugenkamera korrigieren.

Prüfen Sie die unterstützten Ausgabecodecs für Ihre Kamera und passen Sie die Elemente nach Bedarf an.

Wenn Ihre USB-Kamera beispielsweise nur die JPEG-Ausgabe unterstützt, müssen Sie die `jpegdec` Elemente `jpegparse` und verwenden, um das Medium zu transformieren, bevor Sie es mit H.264 kodieren. `x264enc` Suchen Sie in den GStreamer Foren nach Unterstützung für andere Benutzer mit ähnlichen Pipelines und/oder Webcam-Setups.

Behebung von Build-Problemen im C++-Producer-SDK für Raspberry Pi

Wenn Sie auf ein Build-Problem stoßen und verschiedene CMake Argumente ausprobieren möchten, stellen Sie sicher, dass Sie einen sauberen Build durchführen. Löschen Sie die `build` Ordner `open-sourcedependency`, und, bevor Sie es erneut versuchen.

Probleme mit OpenSSL erstellen

Wenn Sie eine Ausgabe ähnlich der folgenden erhalten, deutet dies darauf hin, dass OpenSSL Ihre Systemarchitektur falsch erkannt hat.

```
crypto/md5/md5-aarch64.S: Assembler messages:
crypto/md5/md5-aarch64.S:3: Error: unrecognized symbol type ""
crypto/md5/md5-aarch64.S:6: Error: bad instruction `stp x19,x20,[sp,#-80]!'
crypto/md5/md5-aarch64.S:7: Error: bad instruction `stp x21,x22,[sp,#16]'
crypto/md5/md5-aarch64.S:8: Error: bad instruction `stp x23,x24,[sp,#32]'
crypto/md5/md5-aarch64.S:9: Error: bad instruction `stp x25,x26,[sp,#48]'
```

In diesem Beispiel wird versucht, eine 64-Bit-Version (`linux-aarch64`) zu erstellen, obwohl dieser Raspberry Pi tatsächlich eine 32-Bit-Version ist. Einige Raspberry Pi-Geräte haben einen 64-Bit-Kernel, aber einen 32-Bit-Benutzerbereich.

Ermitteln Sie, für welche Architektur OpenSSL zu bauen versucht. Sie finden die Protokollzeile während des `configure` Schritts für OpenSSL:

```
[ 33%] Performing update step for 'project_libopenssl'
-- Already at requested tag: OpenSSL_1_1_1t
[ 44%] No patch step for 'project_libopenssl'
[ 55%] Performing configure step for 'project_libopenssl'
Operating system: x86_64-whatever-linux2
Configuring OpenSSL version 1.1.1t (0x1010114fL) for linux-x86_64
Using os-specific seed configuration
Creating configdata.pm
Creating Makefile
```

Überprüfen Sie die Architektur Ihres Systems:

- Überprüfen Sie die Kernel-Bitanzahl: run `uname -m`
- Überprüfen Sie den Benutzerbereich bit-ness: run `getconf LONG_BIT`

Sie können Ihre CPU-Informationen auch mit den Befehlen `cat /proc/cpuinfo` oder `lscpu` überprüfen.

Auflösung

Um dieses Problem zu beheben, fügen Sie beim Erstellen das folgende CMake Argument hinzu, um sicherzustellen, dass OpenSSL korrekt für die 32-Bit-ARM-Architektur erstellt wird:

```
-DBUILD_OPENSSL_PLATFORM=linux-armv4
```

Beheben Sie Probleme **kvssink** beim Laden in GStreamer

Bestätigen GST_PLUGIN_PATH

Stellen Sie sicher, dass die GST_PLUGIN_PATH Umgebungsvariable in Ihrer aktuellen Shell-Sitzung auf das Verzeichnis verweist, das enthältkvssink. Umgebungsvariablen sind sitzungsspezifisch, sodass Sie sie für jede neue Sitzung festlegen müssen. Informationen dazu, wie Sie diese Änderung dauerhaft vornehmen können, finden Sie unter „Aktualisieren Sie das Startskript Ihrer Shell, sodass es die Einstellung der Umgebungsvariablen GST_PLUGIN_PATH einschließt“.

Fehler: Die gemeinsam genutzte Objektdatei kann nicht geöffnet werden: Keine solche Datei oder kein solches Verzeichnis

Wenn der Fehler auftrittCannot open shared object file: No such file or directory, führen Sie den folgenden Befehl aus:

```
gst-inspect-1.0 /path/to/libgstkvssink.so
```

Wenn Sie die folgende Ausgabe erhalten, bedeutet dies, dass der dynamische Linker die erforderlichen Bibliotheken für kvssink nicht finden kann. Dies ist in der Regel auf folgende Ursachen zurückzuführen:

- Umzug kvssink an einen anderen Ort als den, an dem es gebaut wurde.
- Cross-Compilierung für die falsche CPU-Architektur.
- Eine erforderliche Abhängigkeit fehlt.

Ausgabe:

```
WARNING: erroneous pipeline: no element "kvssink"
error while loading shared libraries: libcproducer.so: cannot open shared object file:
 No such file or directory
```

Auflösung

Fügen Sie für verschobene Bibliotheken das Verzeichnis hinzu, das die fehlenden Bibliotheken enthältLD_LIBRARY_PATH.

Im Stammverzeichnis des ursprünglichen Repositorys können Sie mit dem `find` Hilfsprogramm nach der fehlenden Bibliothek suchen. Geben Sie im Terminal Folgendes ein:

```
find . -name "*libcproducer*"
```

Ausgabe:

```
./build/dependency/libkvscproducer/kvscproducer-src/libcproducer.so
```

Das Dateipfadtrennzeichen auf Linux-Geräten ist`:`. Mit dem folgenden Befehl wird ein neuer Ordnerpfad an die bestehende `LD_LIBRARY_PATH` Umgebungsvariable angehängt, wobei alle vorherigen Werte beibehalten werden.

Geben Sie in Ihrem Terminal Folgendes ein:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/build/dependency/libkvscproducer/
kvscproducer-src
```

> **Important**
>
> Umgebungsvariablen sind sitzungsspezifisch. Um Änderungen sitzungsübergreifend beizubehalten, ändern Sie das Startskript Ihrer Shell.

Möglicherweise müssen Sie das auch zu Ihrem `open-source/local/lib` `$LD_LIBRARY_PATH` hinzufügen.

Fehler:. /path/to/libcproducer.so.1: Ungültiger ELF-Header

Wenn Sie diesen Fehler beim Laden gemeinsam genutzter Bibliotheken erhalten, kann dies an defekten symbolischen Links () `symlinks` liegen. Symlinks können kaputt gehen, wenn das Betriebssystem des Host-Computers nicht mit dem des Zielcomputers übereinstimmt. Zum Beispiel Cross-Compiling auf einem MacBook für einen Raspberry Pi.

Eine weitere mögliche Ursache ist, dass die erstellten Binärdateien für die falsche Architektur bestimmt waren. Zum Beispiel, wenn die Binärdateien für x86 erstellt wurden (Raspberry Pi verwendet ARM CPUs).

Navigieren Sie zum Speicherort der Bibliothek, der im Fehler angegeben wurde, und geben Sie: ein`ls -la`, um die Bibliothek `symlinks` zu überprüfen.

Antwort:

```
drwxr-xr-x  16 me  staff      512 Sep 10 17:16 .
drwxr-xr-x   7 me  staff      224 Jan  6 23:46 ..
drwxr-xr-x   4 me  staff      128 Sep 10 17:16 engines-1.1
-rwxr-xr-x   1 me  staff  2294496 Sep 10 17:16 libcrypto.1.1.so
-rw-r--r--   1 me  staff  4002848 Sep 10 17:16 libcrypto.a
lrwxr-xr-x   1 me  staff       19 Sep 10 17:16 libcrypto.so -> libcrypto.1.1.so
-rwxr-xr-x   1 me  staff   631176 Sep 10 17:12 liblog4cplus-2.0.3.so
lrwxr-xr-x   1 me  staff       24 Sep 10 17:12 liblog4cplus.so -> liblog4cplus-2.0.3.so
-rwxr-xr-x   1 me  staff     1012 Sep 10 17:12 liblog4cplus.a
-rwxr-xr-x   1 me  staff   694328 Sep 10 17:12 liblog4cplusU-2.0.3.so
lrwxr-xr-x   1 me  staff       25 Sep 10 17:12 liblog4cplusU.dylib ->
 liblog4cplusU-2.0.3.so
-rwxr-xr-x   1 me  staff     1017 Sep 10 17:12 liblog4cplusU.a
-rwxr-xr-x   1 me  staff   536416 Sep 10 17:16 libssl.1.1.so
-rw-r--r--   1 me  staff   795184 Sep 10 17:16 libssl.a
lrwxr-xr-x   1 me  staff       16 Sep 10 17:16 libssl.so -> libssl.1.1.so
drwxr-xr-x   6 me  staff      192 Sep 10 17:16 pkgconfig
```

In der obigen Beispielausgabe `symlinks` sind sie nicht defekt. Bei Broken zeigen `symlinks` keine Pfeile auf ihre Ziele.

Auflösung

Es gibt zwei Optionen, um die Symlinks zu korrigieren:

- Empfehlung: Erstellen Sie das `symlink` mit dem `ln` Befehl neu. Type:

  ```
  ln -s /path/to/actual/library /path/to/symlink
  ```

- Kopieren Sie die eigentliche Bibliotheksdatei und benennen Sie sie entsprechend um. `symlink`

  > **Note**
  >
  > Diese Option führt zu einer erhöhten Speichernutzung.

Es hat sich bewährt, auf demselben Betriebssystem mit Tools wie Docker zu kompilieren, um Probleme bei der Cross-Compilierung zu vermeiden.

Fehlende Abhängigkeiten:

Wenn der fehlende Bibliotheksname mit beginnt`libkvs`, finden Sie im Abschnitt „verschobene Bibliotheken“ weiter oben Informationen zur Installation der Kinesis Video Streams Streams-Bibliotheken vom Host-Gerät auf das Zielgerät.

Andernfalls gehen Sie wie folgt vor, the section called “Installieren Sie die Softwarevoraussetzungen” um sicherzustellen, dass alle erforderlichen Open-Source-Softwarekomponenten auf dem Zielgerät installiert sind.

Fehlercodereferenz

Dieser Abschnitt enthält Fehler- und Statuscodeinformationen für das Zu Kinesis Video Streams hochladen.

Weitere Informationen zu Lösungen für gängige Probleme finden Sie unter Fehlerbehebung.

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 Bibliothek zurückgegebene Fehler- und Statuscodes MKVGen
- 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` wie unter SDK-Kontingente für Produzenten 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`
`0x52000003`	`STATUS_INVALID_DEVICE_NAME_LENGTH`	Ungültige Länge des Gerätenamens.	Beziehen Sie sich auf die maximale Länge von Gerätenamen in Zeichen, die in angegeben istSDK-Kontingente für Produzenten.
`0x52000004`	`STATUS_INVALID_DEVICE_INFO_VERSION`	Ungültige Version der `DeviceInfo` - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
`0x52000005`	`STATUS_MAX_TAG_COUNT`	Die maximale Tag-Anzahl wurde erreicht.	Beziehen Sie sich auf die aktuelle maximale Anzahl von Tags, die in angegeben istSDK-Kontingente für Produzenten.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
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 istSDK-Kontingente für Produzenten.
0x5200000a	STATUS_INVALID_STORAGE_SIZE	Es wurde eine ungültige Speichergröße angegeben.	Die Speichergröße in Byte muss innerhalb der unter SDK-Kontingente für Produzenten angegebenen Limits liegen.
0x5200000b	STATUS_INVALID_ROOT_DIRECTORY_LENGTH	Ungültige Länge der Stammverzeichnis-Zeichenfolge.	Beziehen Sie sich auf die maximale Pfadlänge des Stammverzeichnisses, die in angegeben istSDK-Kontingente für Produzenten.
0x5200000c	STATUS_INVALID_SPILL_RATIO	Ungültiges Überlaufverhältnis.	Geben Sie die Überlaufquote als Prozentsatz zwischen 0 und 100 an.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Ungültige Version der StorageInfo -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
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 GetStreamingEndpoint -API-Aufruf fehlschlägt und die Client-Anwendung ihn ignoriert und weiterhin Frames in den Stream leitet.
0x5200000f	STATUS_SERVICE_CALL_CALLBACKS_MISSING	In der Callbacks -Struktur fehlen Funktions-Einstiegspunkte 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_DESCRIBE_STREAM_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 zu löschenden Stream 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_ARG_ERROR	Für den Dienstaufruf 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. Es kann in verschiedenen Kontexten unterschiedliche Bedeutungen haben, aber die wahrscheinliche Ursache ist die Verwendung von, APIs bevor der Stream erstellt wird. Die Verwendung des SDK bestätigt, dass der Stream zuerst erstellt wurde.
0x52000018	STATUS_INVALID_AUTH_LEN	Ungültige Länge der Authentifizierungsinformationen.	Die aktuellen Werte werden unter SDK-Kontingente für Produzenten 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` -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 SDK-Kontingente für Produzenten 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_STORE_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 Speichergröße 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_CALL_UNKNOWN_ERROR	Von dem Netzwerk-Stack wurde ein unbekannter oder generischer Fehler zurückgegeben.	Weitere detaillierte Informationen finden Sie in den Protokollen.
0x52000032	STATUS_SERVICE_CALL_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_CALL_CLIENT_LIMIT_ERROR	Client-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000034	STATUS_SERVICE_CALL_DEVICE_LIMIT_ERROR	Geräte-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000035	STATUS_SERVICE_CALL_STREAM_LIMIT_ERROR	Stream-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000036	STATUS_SERVICE_CALL_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_CALL_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 SDK-Kontingente für Produzenten angegeben werden.
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	Ungültige Länge des Tag-Wertes.	Nehmen Sie auf die Limits des Tag-Wertes Bezug, die unter SDK-Kontingente für Produzenten 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	Ungültige benutzerdefinierte Daten, die PIC aufrufen APIs.	In einem PIC-Aufruf wurden ungültige benutzerdefinierte Daten angegeben APIs. 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 SDK-Kontingente für Produzenten 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 oder SecretAccessKey) 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_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 übergeordnete Ebene SDKs (z. B. C++ oder Java) implementiert 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 SDK-Kontingente für Produzenten.
0x5200004a	STATUS_END_OF_STREAM	Ende des Streams-(EOS)-Indikator.	Gibt im API-Aufruf `GetStreamData` 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_IN VALID_RET ENTION_PERIOD	Ungültiger Aufbewahrungszeitraum.	In der StreamInfo - Struktur wird ein ungültiger Aufbewahrungszeitraum angegeben. Weitere Informationen zum gültigen Wertbereich für den Aufbewahrungszeitraum finden Sie unter SDK-Kontingente für Produzenten.
0x5200004d	STATUS_IN VALID_ACK _KEY_START	Ungültig FragmentAck .	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_IN VALID_ACK _DUPLICAT E_KEY_NAME	Ungültig FragmentAck .	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_IN VALID_ACK _INVALID_ VALUE_START	Ungültig FragmentAck .	Fehler beim Analysier en der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüssel wert-Startindikators. Die Codefragment-ACK-Z eichenfolge ist möglicher weise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000050	STATUS_IN VALID_ACK _INVALID_ VALUE_END	Ungültig FragmentAck .	Fehler beim Analysier en der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüssel wert-Endindikators. Die Codefragment-ACK-Z eichenfolge ist möglicher weise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000051	STATUS_IN VALID_PAR SED_ACK_TYPE	Ungültig FragmentAck .	Fehler beim Analysier en der Codefragment-ACK-Zeichenfolge, da ein ungültiges ACK-Typ angegeben wurde.
0x52000052	STATUS_ST REAM_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 `StreamMetrics` -Struktur.	Geben Sie die korrekte aktuelle Version der `StreamMetrics` -Struktur an.
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Ungültige Version der `ClientMetrics` -Struktur.	Geben Sie die korrekte aktuelle Version der `ClientMetrics` -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 `FragmentAck` -Struktur wird ein ungültiger ACK-Typ angegeben.	Die in der `FragmentAck` -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_FREED_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 Zuweisungsgröße 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_ERR_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_ACK_ERR_FRAGMENT_SIZE_REACHED	Die maximale Fragmentgröße wurde erreicht.	Die maximale Fragmentgröße in Byte ist unter SDK-Kontingente für Produzenten 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 SDK-Kontingente für Produzenten 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. SDK-Kontingente für Produzenten 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_ACK_ERR_FRAGMENT_TIMECODE_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_ACK_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_ACK_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_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	Ungültiger Producer-Zeitstempel.	Der Dienst gibt diesen ACK-Fehler zurück, wenn die Uhr des Producers wesentlich vorgeht. Auf höherer Ebene SDKs (z. B. Java oder C++) wird eine Version der Systemuhr verwendet, um den aktuellen Zeitrückruf von PIC zu erfüllen. 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_ACK_ERR_STREAM_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_ACK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS Fehler „Zugriff verweigert“.	Wird zurückgegeben, wenn das Konto nicht auf den angegebenen Schlüssel zugreifen kann.
0x52000066	STATUS_ACK_ERR_KMS_KEY_DISABLED	AWS KMS Der Schlüssel ist deaktiviert.	Der angegebene Schlüssel wurde deaktiviert.
0x52000067	STATUS_ACK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS Fehler bei der Schlüsselüberprüfung.	Allgemeiner Validierungsfehler. Weitere Informationen finden Sie in der AWS Key Management Service - API-Referenz.
0x52000068	STATUS_ACK_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_ACK_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_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS ungültiger 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_ACK_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 unbekannten 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 SDK-Kontingente für Produzenten.
0x52000074	STATUS_MAX_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 Metadatenelementen. 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_PUT_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_MAX_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_ERR_TRACK_NUMBER_MISMATCH	Die Track-Nummer des ACK-Fehlers ist falsch zugeordnet.	
0x5200007d	STATUS_ACK_ERR_FRAMES_MISSING_FOR_TRACK	Frames fehlen für einen Track.	
0x5200007e	STATUS_ACK_ERR_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_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUTMEDIA_LAST_PERSIST_ACK_NOT_RECEIVED		
0x52000089	STATUS_NON_ALIGNED_HEAP_WITH_IN_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_IN VALID_IMA GE_METADA TA_KEY_LENGTH		
0x5200008f	STATUS_IN VALID_IMA GE_METADA TA_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 Fehler- und Statusinformationscodes sind vielen gemeinsam APIs.

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
0x00000001	0x1	STATUS_NULL_ARG	Für ein obligator isches Argument

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
			wurde NULL übergeben.
0x00000002	0x2	STATUS_INVALID_ARG	Für ein Argument wurde ein ungültiger Wert angegeben.
0x00000003	0x3	STATUS_INVALID_ARG_LEN	Es wurde eine ungültige Länge für ein Argument angegeben.
0x00000004	0x4	STATUS_NOT_ENOUGH_MEMORY	Es konnte nicht genug Speicher zugewiesen werden.
0x00000005	0x5	STATUS_BUFFER_TOO_SMALL	Die angegebene Puffergröße ist zu klein.
0x00000006	0x6	STATUS_UNEXPECTED_EOF	Es wurde ein unerwartetes Dateiende erreicht.
0x00000007	0x7	STATUS_FORMAT_ERROR	Es ist ein ungültiges Format aufgetreten.
0x00000008	0 x 8	STATUS_INVALID_HANDLE_ERROR	Ungültiger Handle-Wert.
0x00000009	0x9	STATUS_OPEN_FILE_FAILED	Fehler beim Öffnen einer Datei.
0x0000000a	0xa	STATUS_READ_FILE_FAILED	Fehler beim Lesen aus einer Datei.

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
0x0000000b	0xb	`STATUS_WRITE_TO_FILE_FAILED`	Fehler beim Schreiben in eine Datei.
0x0000000c	0xc	`STATUS_INTERNAL_ERROR`	Es ist ein ungewöhnlicher interner Fehler aufgetreten, der möglicherweise auf einen SDK- oder Service-API-Fehler hinweist.
0x0000000d	0xD	`STATUS_INVALID_OPERATION`	Eine Operation ist ungültig oder nicht zulässig.
0x0000000e	0xe	`STATUS_NOT_IMPLEMENTED`	Die Funktion ist nicht implementiert.
0x0000000f	0xf	`STATUS_OPERATION_TIMED_OUT`	Bei der Operation ist eine Zeitüberschreitung aufgetreten.
0x00000010	0x10	`STATUS_NOT_FOUND`	Eine erforderliche Ressource wurde nicht gefunden.
0x00000011	0 x 11	`STATUS_CREATE_THREAD_FAILED`	Fehler beim Erstellen eines Threads.

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
0x00000012	0 x 12	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	0 x 13	STATUS_THREAD_INVALID_ARG	Es wurden ungültige Thread-Attribute angegeben, oder ein anderer Thread wartet bereits darauf, diesem Thread beizutreten.
0x00000014	0 x 14	STATUS_THREAD_PERMISSIONS	Keine Berechtigung zum Festlegen der in Thread-Attributen angegebenen Scheduling-Richtlinie und der in Thread-Attributen angegebenen Parameter
0x00000015	0 x 15	STATUS_THREAD_DEADLOCKED	Ein Deadlock wurde erkannt oder der beitretende Thread spezifiziert den aufrufenden Thread.

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
0x00000016	0 x 16	`STATUS_THREAD_DOES_NOT_EXIST`	Der Thread mit der angegebenen Thread-ID konnte nicht gefunden werden.
0x00000017	0 x 17	`STATUS_JOIN_THREAD_FAILED`	Beim Thread-Join-Vorgang wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x00000018	0 x 18	`STATUS_WAIT_FAILED`	Die maximale Wartezeit für die Bedingungsvariable wurde überschritten.
0x00000019	0 x 19	`STATUS_CANCEL_THREAD_FAILED`	Beim Vorgang zum Abbrechen des Threads wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x0000001a	0 x 1 a	`STATUS_THREAD_IS_NOT_JOINABLE`	Der Thread-Join-Vorgang wird für einen Thread angefordert, der nicht verknüpft werden kann.

Code	Code ohne führende Nullen	Fehlermeldung	Beschreibung
0x0000001b	0 x 1 b	`STATUS_DETACH_THREAD_FAILED`	Beim Vorgang zum Trennen des Threads wurde ein unbekannter oder generischer Fehler zurückgegeben.
0x0000001c	0x1 c	`STATUS_THREAD_ATTR_INIT_FAILED`	Das Thread-Attribute-Objekt konnte nicht initialisiert werden.
0x0000001d	0x1d	`STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED`	Die Stackgröße für das Thread-Attribute-Objekt konnte nicht festgelegt werden.
0x0000001e	0x1e	`STATUS_MEMORY_NOT_FREED`	Wird nur in den Tests verwendet. Zeigt an, dass nicht der gesamte angeforderte Speicher freigegeben wurde.
0x0000001f	0x1f	`STATUS_INVALID_THREAD_PARAMS_VERSION`	Ungültige "ThreadParams" Strukturversion. Geben Sie die korrekte aktuelle Version der Struktur an.

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
0x10000001	STATUS_HEAP_FLAGS_ERROR	Es wurde eine ungültige Kombination von Flags angegeben.
0x10000002	STATUS_HEAP_NOT_INITIALIZED	Eine Operation wurde vor der Heap-Initialisierung versucht.
0x10000003	STATUS_HEAP_CORRUPTED	Der Heap wurde beschädigt oder das Sicherheits-Frequenzband (im Debug-Modus) wurde überschrieben. Ein Pufferüberlauf im Client-Code könnte eine Heap-Beschädigung verursachen.
0x10000004	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.
0x10000005	STATUS_HEAP_VRAM_LIB_REOPEN	Fehler beim Öffnen der VRAM-Bibliothek.
0x10000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	Fehler beim Laden des INIT-Funktionsexports.
0x10000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	Fehler beim Laden des ALLOC-Funktionsexports.
0x10000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	Fehler beim Laden des FREE-Funktionsexports.

Code	Fehlermeldung	Beschreibung
0x10000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	Fehler beim Laden des LOCK-Funktionsexports.
0x1000000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	Fehler beim Laden des UNLOCK-Funktionsexports.
0x1000000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	Fehler beim Laden des UNINIT-Funktionsexports.
0x1000000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	Fehler beim Laden des GETMAX-Funktionsexports.
0x1000000d	STATUS_HEAP_DIRECT_MEM_INIT	Fehler bei der Initialisierung des Haupt-Heap-Pools im Hybrid-Heap.
0x1000000e	STATUS_HEAP_VRAM_INIT_FAILED	Die dynamische VRAM-Initialisierung ist fehlgeschlagen.
0x1000000f	STATUS_HEAP_LIBRARY_FREE_FAILED	Fehler beim Aufheben der Zuweisung und Freisetzen der VRAM-Bibliothek.
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	Die VRAM-Zuweisung ist fehlgeschlagen.
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	Die VRAM-Freisetzung ist fehlgeschlagen.
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	Die VRAM-Zuordnung ist fehlgeschlagen.
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	Das Aufheben der VRAM-Zuordnung ist fehlgeschlagen.
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	Das Aufheben der VRAM-Initialisierung ist fehlgeschlagen.

Code	Fehlermeldung	Beschreibung
0x10000015	`STATUS_INVALID_ALLOCATION_SIZE`	
0x10000016	`STATUS_HEAP_REALLOC_ERROR`	
0x10000017	`STATUS_HEAP_FILE_HEAP_FILE_CORRUPT`	

Von der Bibliothek zurückgegebene Fehler- und Statuscodes MKVGen

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 Sie sicher, dass Dauer, Größe und Frame-Daten gültig sind und innerhalb der unter angegebenen Grenzen liegenSDK-Kontingente für Produzenten.
0x32000002	`STATUS_MKV_INVALID_FRAME_TIMESTAMP`	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

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
		oder ein Stabilitätsproblem des Encoders. Informationen zur Problembehebung finden Sie unter Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“
0x32000003	`STATUS_MKV_INVALID_CLUSTER_DURATION`	Es wurde eine ungültige Fragment-Dauer angegeben wurde. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.
0x32000004	`STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH`	Ungültige Zeichenfolgenlänge des Inhaltstyps. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.
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 SDK-Kontingente für Produzenten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000007	`STATUS_MKV_INVALID_TRACK_NAME_LENGTH`	Ungültige Zeichenfolgenlänge des Tracknamens. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.
0x32000008	`STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH`	Ungültige Länge privater Codec-Daten. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.
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 SDK-Kontingente für Produzenten.
0x3200000b	`STATUS_MKV_MAX_FRAME_TIMECODE`	Der Frame-Timecode ist größer als das maximale Limit. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
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_FRAME_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 Emulationsverhütungssequenz 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_FRAME_DATA (0x3200000d)-Fehler unter Android.
0x3200000e	`STATUS_MKV_INVALID_AVCC_NALU_IN_FRAME_DATA`	Ungültiges AVCC NALU Packaging, 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`	Sowohl die Anpassung von AVCC als auch von Anhang B wurden spezifiziert. NALUs 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`	HEVC fehlt NALUs im Sequenzparametersatz (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 des privaten Betawerts für den AnnexBb 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	Private Codec-Daten fehlen in Anhang B. NALUs
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Ungültiger privater Codec in Anhang 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_UID	Ungültige Track-UID.
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_FALIED
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 SDK-Kontingente für Produzenten.
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	Es wurde eine ungültige Dauer der Inhaltsansicht angegeben. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.

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_VIEW_INVALID_TIMESTAMP`	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:. `` `DTS(n) >= DTS(n-1) + 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 abzuhelfen, geben Sie in der `StreamInfo.StreamCaps` -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

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 `StreamCallbacks` -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Ungültige Version der `AuthCallbacks` -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	Ungültige Version der `ApiCallbacks` -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Ungültige Version der `AwsCredentials` -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_DESCRIBE_STREAM_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_LEN_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_LEN_EXCEEDED	Die maximale Länge der Namensendung des Agenten wurde erreicht.	
0x15000014	STATUS_MAX_CUSTOM_USER_AGENT_LEN_EXCEEDED	Die maximale Länge des Kundenbenutzeragenten wurde erreicht.	
0x15000015	STATUS_INVALID_USER_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_FREE_FUNC		

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_MAX_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_LWS_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CA_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_BEING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RESET_FAILED		

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x16000001	STATUS_CURL_PERFORM_FAILED	CURL hat einen Fehlercode zurückgegeben.	Weitere Informationen finden Sie in den Protokollen. Ein häufiger CURL-Fehler ist „Hostname konnte nicht aufgelöst werden. „, überprüfen Sie die Internetverbindung des Geräts. Ein weiterer häufiger Fehler ist ein 403-Fehlercode. Dies weist darauf hin, dass IoT-Zertifikate nicht korrekt erstellt oder angegeben wurden. Überprüfen Sie, ob die Dateipfade zu den IoT-Zertifikaten und Berechtigungen korrekt eingestellt sind. Weitere Informationen finden Sie unter the section called “Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe AWS IoT”.
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH	Beim Abrufen von IoT-Anmeldeinformationen wurde eine Antwort mit der Länge 0 erhalten.	Überprüfen Sie das AWS Health-Dashboard und versuchen Sie es später erneut.

Code	Fehlermeldung	Beschreibung	Empfohlene Aktion
0x16000003	STATUS_IOT_NULL_AWS_CREDS	Das vom Endpunkt für IoT-Anmeldeinformationen zurückgegebene JSON enthielt das Anmeldeinformationsobjekt nicht.	Weitere Informationen finden Sie im JSON-Element „Nachricht“.
0x16000004	STATUS_IOT_INVALID_URI_LEN	Die URL, die an die Funktion zum Abrufen von IoT-Anmeldeinformationen übergeben wurde, hat keine Länge zwischen 1 und 10.000.	Überprüfen Sie die URL, die an diese Funktion übergeben wurde.
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT	Das Element „Ablauf“ in der JSON-Datei für das Abrufen von IoT-Anmeldeinformationen hat nicht das Format:YYYY-MM-DDTHH:mm:SSZ .	Überprüfen Sie das AWS Health-Dashboard und versuchen Sie es später erneut.

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-Einheiten) durch zwei Byte mit Nullen, gefolgt von einem oder drei Byte mit Nullen, gefolgt von der Zahl 1 (so genannter Startcode, z. B. 00000001).
- Das AVCC-Format umschließt ebenfalls NALUs, aber jeder NALU ist ein Wert vorangestellt, der die Größe der NALU angibt (normalerweise vier Byte).

Viele Encoder produzieren das Annex-B-Bitstream-Format. Einige Bitstream-Prozessoren auf höherer Ebene (z. B. eine Playback-Engine oder der Media Source Extensions (MSE) -Player im) verwenden das AWS Management Console AVCC-Format für ihre Frames.

Der Codec Private Data (CPD), also SPS/PPS (Sequence Parameter Set/Picture Parameter Set) für den H.264-Codec, kann auch im Annex-B- oder AVCC-Format vorliegen. Bei der CPD unterscheiden sich die Formate jedoch von den zuvor beschriebenen.

Die Flags weisen das SDK an, das für Frame-Daten und CPD wie folgt NALUs an AVCC oder Annex-B anzupassen:

Flag	Anpassung
`NAL_ADAPTATION_FLAG_NONE`	Keine Anpassung.
`NAL_ADAPTATION_ANNEXB_NALS`	Passen Sie Anhang B an AVCC NALUs an. NALUs
`NAL_ADAPTATION_AVCC_NALS`	Passen Sie AVCC an Anhang B NALUs an. NALUs
`NAL_ADAPTATION_ANNEXB_CPD_NALS`	Passen Sie Anhang B NALUs für den Codec private Daten an das AVCC-Format an. NALUs
`NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS`	Passen Sie Anhang B NALUs für den Codec an und stellen Sie private Daten in das AVCC-Format ein. NALUs

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 DefaultDeviceInfoProviderObjekte DeviceInfound 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.

```
Size = NumberOfStreams * AverageFrameSize * FramesPerSecond * BufferDurationInSeconds *
 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.

```
Size = (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) *
 1.2 = 65059200 = ~ 66MB.
```

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 bei 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. Dies kann auch passieren, wenn die Anwendungsbestätigungen (ACKs) nicht verfügbar sind oder die ACKs persistenten Bestätigungen verzögert werden. In diesem Fall werden die Puffer bis zur Kapazität der Pufferdauer aufgefüllt, 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
stream_name	`string`	Ein optionaler Streamname.	Wenn kein Name angegeben wird, wird

Feld	Datentyp	Beschreibung	Standardwert
		Weitere Informationen zur Länge des Streamnamens finden Sie unter SDK-Kontingente für Produzenten. Jeder Stream sollte einen eindeutigen Namen haben.	ein zufälliger Name generiert.
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 die vorhandenen Tags angehängt.	Keine Tags

Feld	Datentyp	Beschreibung	Standardwert
kms_key_id	`string`	Die AWS KMS Schlüssel-ID, die für die Verschlüsselung des Streams verwendet werden soll. Weitere Informationen finden Sie unter Datenschutz in Kinesis Video Streams.	Der KMS-Standardschlüssel (`aws/kinesisvideo`).
streaming_type	`STREAMING_TYPE` - Aufzählung	Der einzige unterstützte Wert ist `STREAMING_TYPE_REALTIME` .	
content_type	`string`	Inhaltsformat des Streams. Die Kinesis Video Streams Streams-Konsole kann Inhalte in diesem `video/h264` Format wiedergeben.	`video/h264`

Feld	Datentyp	Beschreibung	Standardwert
max_latency	`duration< uint64_t, milli>`	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 **`0`** wird festgelegt, dass der Callback für Stream-Latenzdruck aufgerufen wird.	`milliseconds::zero()`

Feld	Datentyp	Beschreibung	Standardwert
fragment_duration	`duration< uint64_t>`	Die gewünschte Fragmentdauer in Sekunden. Dieser Wert wird in Kombination mit dem `key_frame _fragmentation` - Wert verwendet . Wenn dieser Wert lautet`false`, 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 `key_frame _fragmentation = false` kommt es nach Ablauf dieser Dauer zur Fragmentierung eines Keyframes mit dem Ergebnis zweisekün diger Fragmente.	2

Feld	Datentyp	Beschreibung	Standardwert
timecode_scale	`duration< uint64_t, milli>`	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 ~= 32 Sekunden lang ist. Dieser Wert liegt über der maximalen Fragmentdauer von 10 Sekunden, die	1

Feld	Datentyp	Beschreibung	Standardwert
		in Kontingente für den Amazon Kinesis Video Streams Streams-Service angegeben ist.	
key_frame_fragmentation	`bool`	Gibt an, ob Fragmente für einen Keyframe erstellt werden. Wenn der Wert `true` ist, produziert das SDK jedes Mal einen Start des Fragments, wenn ein Keyframe auftritt. Wenn`false`, wartet Kinesis Video Streams mindestens darauf `fragment_duration` und erzeugt ein neues Fragment auf dem darauffolgenden Schlüsselbild.	`true`

Feld	Datentyp	Beschreibung	Standardwert
frame_timecodes	`bool`	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`false`, wird sichergestellt, dass die Frames bei der Eingabe in Kinesis Video Streams mit einem Zeitstempel versehen werden.	`true`

Feld	Datentyp	Beschreibung	Standardwert
absolute_fragment_times	`bool`	Kinesis Video Streams verwendet MKV als zugrunde liegenden Verpackungsmechanismus. In der MKV-Spezifikation müssen Frame-Timecodes relativ zum Anfang des Clusters (Fragments) sein. Die Cluster-Timecodes können jedoch entweder absolut oder relativ zum Startzeitpunkt des Streams sein. Wenn die Zeitstempel relativ sind, verwendet der Aufruf von `PutMedia` 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.	`true`

Feld	Datentyp	Beschreibung	Standardwert
fragment_acks	`bool`	Ob Fragmente auf Anwendungsebene ACKs (Bestätigungen) empfangen werden sollen.	`true`, was bedeutet, dass das SDK das empfängt ACKs und entsprechend handelt.
restart_on_error	`bool`	Wird bei bestimmten Fehlern neu gestartet ?	`true` 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 leistungsschwachen/kleinen Geräten möglicher weise auf `false` eingestellt werden, um CPU-Takte zu sparen. Andernfalls empfehlen wir, diesen Wert nicht `false` zu verwenden.	`true`

Feld	Datentyp	Beschreibung	Standardwert
nal_adaptation_flags	`uint32_t`	Gibt die Anpassungs-Flags für die Netzwerk-Abstraktionsschichteinheit (Network Abstraction Layer Unit, NALU) an. Wenn der Bitstream H.264-codiert ist, kann er anschließend als Rohdatei oder als Paket verarbeitet werden. NALUs 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, weil 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 Konsolenwiedergabe verwendet die Browser-MSE (Media Source	Standardmäßig wird das Format Annex-B für das Format AVCC sowohl für die Framedaten als auch für die CPD (Codec Private Data) adaptiert.

Feld	Datentyp	Beschreibung	Standardwert
		Extensions), um den Stream im Format AVCC zu decodiere n und wiederzug eben. Für H.264 (und für M-JPEG und H.265), stellt das SDK Adaptionsfunktionen bereit. Viele einfache Streams liegen in folgendem Format vor. In diesem Beispiel ist Ab der Annex-B-Startcode (001 oder 0001). `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)` 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	

Feld	Datentyp	Beschreibung	Standardwert
		an das AVCC-Form at angepasst werden. Sofern die Medien-Pipeline die CPD nicht separat ausgibt, kann die Anwendung die CPD aus dem Frame extrahieren. Dazu sucht er nach dem ersten IDR-Frame (der SPS und PPS enthalten sollte), extrahiert die beiden NALUs (die sind`Ab(Sps)Ab(Pps)`) und fügt ihn in die CPD ein. `StreamDef inition` Weitere Informationen finden Sie unter the section called “NAL-Anpassungsflags”.	
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`buffer_duration` , 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 `max_latency` 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 rückwärts bewegt 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 Fragmentstart 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 Verbindun g 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 `StreamDefinition.codecPrivateData` 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. `KinesisVideoStream.start(cpd)`	`null`
codecPrivateDataGröße	`uint32_t`	Die CPD-Puffergröße	0

ClientMetrics

Das ClientMetricsObjekt wird durch Aufrufen `getKinesisVideoMetrics` gefüllt.

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 Speichergröße 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 angerechn

Feld	Datentyp	Beschreibung
		et. Dieser Speicher wird mit dem Makro `MEMALLOC` zugeordnet, das überschri eben 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 `STREAM_ME TRICS_CURRENT_VERS ION` .
currentViewDuration	`UINT64`	Die Dauer der akkumulie rten 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 `max_latency` angegeben

Feld	Datentyp	Beschreibung
		e Dauer wird`StreamDef inition` , wird der Stream-Latenz-Callback aufgerufe n, sofern er angegeben ist. Die Dauer wird in Einheiten von 100 ns – die Standardz eiteinheit für den PIC-Layer – angegeben.
overallViewDuration	`UINT64`	Die Gesamtanzeigedauer. Wenn der Stream ohne ACKs oder Persistenz konfiguri ert ist, nimmt dieser Wert zu, wenn die Frames in den Kinesis-Videostream eingefügt werden, und entspricht dem `buffer_duration` in der. `StreamDefinition` Wenn ACKs diese Option aktiviert ist und das persisten te 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 Standardz eiteinheit für den PIC-Layer – angegeben.
currentViewSize	`UINT64`	Die Größe des aktuellen Puffers in Byte.

Feld	Datentyp	Beschreibung
overallViewSize	`UINT64`	Die Gesamtanzeigegröße in Byte.
currentFrameRate	`UINT64`	Die beobachtete Bildrate für den aktuellen Stream.
currentTransferRate	`UINT64`	Die beobachtete Übertragungsrate 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.
- StreamCallbackProviderund 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 the section called “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
`CreateDeviceFunc`	Derzeit nicht im Backend implementiert. Dieser Aufruf schlägt fehl, wenn er mit Java oder C++ ausgeführt wird. Andere Clients führen eine	Backend API

Funktion	Beschreibung	Typ
	plattformspezifische Initialisierung durch.	
`CreateStreamFunc`	Wird beim Erstellen des Streams aufgerufen	Backend API
`DescribeStreamFunc`	Wird aufgerufen, wenn `DescribeStream` aufgerufen wird	Backend API
`GetStreamingEndpointFunc`	Wird aufgerufen, wenn `GetStreamingEndpoint` aufgerufen wird	Backend API
`GetStreamingTokenFunc`	Wird aufgerufen, wenn `GetStreamingToken` aufgerufen wird	Backend API
`PutStreamFunc`	Wird aufgerufen, wenn `PutStream` aufgerufen wird	Backend API
`TagResourceFunc`	Wird aufgerufen, wenn `TagResource` aufgerufen wird	Backend API
`CreateMutexFunc`	Erstellt einen Synchronisierungs-Mutex.	Synchronisierung
`FreeMutexFunc`	Gibt den Mutex frei.	Synchronisierung
`LockMutexFunc`	Sperrt den Synchronisierungs-Mutex.	Synchronisierung
`TryLockMutexFunc`	Versucht den Mutex zu sperren. Derzeit nicht implementiert.	Synchronisierung

Funktion	Beschreibung	Typ
`UnlockMutexFunc`	Hebt die Sperrung des Mutex auf.	Synchronisierung
`ClientReadyFunc`	Wird aufgerufen, wenn der Client bereit ist.	Benachrichtigung
`DroppedFrameReportFunc`	Meldet, wenn ein Frame verworfen wird.	Benachrichtigung
`DroppedFragmentReportFunc`	Meldet, wenn ein Fragment verworfen wird. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Benachrichtigung
`FragmentAckReceivedFunc`	Wird aufgerufen, wenn eine Fragment-ACK (Puffern, Empfang, persistent und Fehler) empfangen wird.	Benachrichtigung
`StorageOverflowPressureFunc`	Wird aufgerufen, wenn die Speichernutzung den Wert `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD` , also 5 Prozent der Gesamtspeichergröße, unterschreitet.	Benachrichtigung
`StreamClosedFunc`	Wird aufgerufen, wenn die letzten Bits der verbleibenden Frames gestreamt werden.	Benachrichtigung

Funktion	Beschreibung	Typ
`StreamConnectionStaleFunc`	Wird aufgerufen, wenn der Stream veraltet ist. In diesem Fall sendet der Produzent Daten an den Service, empfängt jedoch keine Bestätigungen.	Benachrichtigung
`StreamDataAvailableFunc`	Wird aufgerufen, wenn Stream-Daten verfügbar sind.	Benachrichtigung
`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 `max_latency` 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

Funktion	Beschreibung	Typ
`GetDeviceCertificateFunc`	Gibt das Gerätezertifikat zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration
`GetDeviceFingerprintFunc`	Gibt die Geräteidentifikation zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration
`GetRandomNumberFunc`	Gibt eine zufällige Zahl zwischen 0 und `RAND_MAX` zurück.	Plattformintegration
`GetSecurityTokenFunc`	Gibt das Sicherheitstoken zurück, das an die Funktionen übergeben wurde, die mit der Backend-API kommunizieren. Die Implementierung kann die serialisierten Werte für `AccessKeyId` , `SecretKeyId` und das Sitzungstoken angeben.	Plattformintegration
`LogPrintFunc`	Protokolliert eine Textzeile mit dem Tag und der Protokoll ebene. Weitere Informationen finden Sie unter `PlatformUtils.h` .	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 bei https://github.com/awslabs/amazon-kinesis-video-streamsproducer-sdk-cpp-/ blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine .c nach Bedarf verwenden. Beachten Sie, dass es keine Möglichkeit gibt, Frames, die aufgrund eines Netzwerkausfalls 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 (gepuffertes 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/awslabs/amazon-kinesis-video-streams-producer- c/blob/master/src/source/ConnectionStaleStateMachine .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.

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 reagierenSehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an.

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.

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 mithilfe von hinzugefügten Metadatenelemente Zu Kinesis Video Streams hochladen unterscheiden sich von dem Tagging auf Stream-Ebene, das mitthe section called “TagStream”, the section called “UntagStream” und APIs implementiert wurde. the section called “ListTagsForStream”

API für Streaming-Metadaten

Sie können die folgenden Operationen im Producer-SDK verwenden, um Streaming-Metadaten zu implementieren.

Themen

- PIC
- C++-Produzenten-SDK
- SDK für Java-Produzenten
- Persistente und nicht persistente Metadaten

PIC

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

C++-Produzenten-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);
```

SDK für Java-Produzenten

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.

Bei persistenten Metadaten sammelt das Producer-SDK die Metadatenelemente in der Metadatenwarteschlange auf die gleiche Weise 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).

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 unterStreamen Sie mithilfe der 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 einem der vier unten APIs aufgeführten Formate nicht unterstützt wird, verwenden Sie `GetMedia`oder `GetMediaForFragmentList`, da für sie keine Einschränkungen hinsichtlich des Tracktyps gelten.

Themen

- GetClip Anforderungen
- Rufen Sie die DASHStreaming SessionURL-Anforderungen ab
- Rufen Sie die HLSStreaming SessionURL-Anforderungen ab
- 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

Beschreibung von Track 1	Codec-ID von Track 1	Beschreibung von Track 2	Codec-ID für Track 2
H.265-Video	V_ MPEGH/ISO/HEVC	AAC-Audio	A_AAC

Important

Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten codecspezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. CPD-Änderungen zwischen den Zielfragmenten des resultierenden Clips werden nicht unterstützt. Die CPD muss auf den abgefragten Medien konsistent bleiben, andernfalls wird ein Fehler zurückgegeben.

Important

Änderungen nachverfolgen wird nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Es wird ein Fehler zurückgegeben, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

Rufen Sie die DASHStreaming SessionURL-Anforderungen ab

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

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	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

Important

Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten codecspezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. CPD-Änderungen werden während einer Streaming-Sitzung nicht unterstützt. Die CPD muss in den abgefragten Medien konsistent bleiben.

Important

Änderungen nachverfolgen wird nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Das Streaming schlägt fehl, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

Rufen Sie die HLSStreaming SessionURL-Anforderungen ab

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

Note

Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten codecspezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. Sowohl für TS- als auch für CPD-Änderungen werden während einer MP4 Streaming-Sitzung unterstützt. Daher können die Fragmente in einer Sitzung unterschiedliche Informationen in der CPD enthalten, ohne dass die Wiedergabe unterbrochen wird. Für jede Streaming-Sitzung sind nur 500 CPD-Änderungen zulässig.

> **Important**
>
> Änderungen nachverfolgen werden nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Das Streaming schlägt fehl, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

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 branchenübliches 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 Get SessionUrl HLSStreaming. 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
- Fehlerbehebung bei HLS-Problemen

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

Das folgende Verfahren zeigt, wie Sie die verwenden, AWS CLI um 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 kinesisvideo get-data-endpoint \
  --api-name GET_HLS_STREAMING_SESSION_URL \
  --stream-name YourStreamName
```

   Sie erhalten eine Antwort, die wie die folgende 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 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
   ```

   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 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 durch die HLS-Fragmentauswahl angegebenen Medien. 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 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 die folgende 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 Diese statische Webseite vereinfacht das Testen und Experimentieren mit HLS- und MPEG-DASH-Ausgaben aus Amazon Kinesis Kinesis-Videostreams. Die Beispielseite bietet Eingabefelder für die folgenden Parameter:

- AWS-Region: Die Region, in der sich Ihr Kinesis-Videostream befindet
- Stream-Name: Der Name Ihres Kinesis Video Streams
- Wiedergabemodus: Der HLS-Wiedergabemodus (LIVE, LIVE_REPLAY oder ON_DEMAND)
- Fragmentauswahltyp: Die Methode, die zur Auswahl von Fragmenten verwendet wird (SERVER_TIMESTAMP oder PRODUCER_TIMESTAMP)
- Fragmentnummer: Die Startfragmentnummer (falls zutreffend)
- Containerformat: Das Format des Mediencontainers (FRAGMENTED_ MP4 oder MPEG_TS)

Die Anwendung ruft diese Werte aus den Eingabefeldern auf der HTML-Seite ab und verwendet sie, um die Anforderung für eine HLS-Streaming-Sitzung zu erstellen, die auf der Seite angezeigt wird.

Erläuterung der Themen mit 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)
    .promise();
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:

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

Fehlerbehebung bei 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-Stream-Sitzung angeben. Probieren Sie verschiedene Kombinationen von Werten für die folgenden Parameter aus:
  - `ContainerFormat`
  - `PlaybackMode`
  - `FragmentSelectorType`
  - `DiscontinuityMode`
  - `MaxMediaPlaylistFragmentResults`

  Einige Mediaplayer (wie HTML5 auch mobile Player) unterstützen HLS normalerweise nur mit dem MP4 F-Containerformat. Andere Mediaplayer (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 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 Get SessionUrl. DASHStreaming Diese Aktion gibt eine URL (mit einem Sitzungs-Token) 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 “Holen Sie sich die SessionURL DASHStreaming”
- 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 Streams-Client für die MPEG-DASH-Wiedergabe einrichten
- Rufen Sie den Endpunkt für archivierte Inhalte von Kinesis Video Streams für die MPEG-DASH-Wiedergabe ab

- Rufen Sie die URL der MPEG-DASH-Streaming-Sitzung ab
- Zeigen Sie das Streaming-Video mit MPEG-DASH-Wiedergabe an
- Abgeschlossenes Beispiel

Den Kinesis Video Streams 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 Endpunkt für archivierte Inhalte von Kinesis Video Streams 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);
```

Rufen Sie die URL der MPEG-DASH-Streaming-Sitzung ab

Wenn Sie den Endpunkt für archivierte Inhalte haben, rufen Sie die Get DASHStreaming SessionUrl 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);
```

Zeigen Sie das Streaming-Video mit MPEG-DASH-Wiedergabe an

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');
}
```

Abgeschlossenes Beispiel

Sie können den fertigen Beispielcode unter herunterladen oder ansehen GitHub.

Benachrichtigungen in Kinesis Video Streams einrichten

Wenn ein Medienfragment zur Nutzung verfügbar ist, benachrichtigt Kinesis Video Streams die Kunden mithilfe von Amazon Simple Notification Service (Amazon SNS) -Benachrichtigungen.

> **Note**
>
> Amazon Kinesis Video Streams verwendet Amazon SNS Standard Topics für die Kommunikation. FIFO-Themen werden derzeit nicht unterstützt.

In den folgenden Themen wird erklärt, wie Sie mit Benachrichtigungen beginnen können.

Themen

- Benachrichtigungskonfigurationen verwalten
- Über MKV-Tags des Herstellers
- Amazon SNS SNS-Nachrichten

Benachrichtigungskonfigurationen verwalten

Verwenden Sie `UpdateNotificationConfiguration` und, um Benachrichtigungskonfigurationen zu verwalten`DescribeNotificationConfiguration`. Weitere Informationen hierzu finden Sie unten.

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 DescribeNotificationConfigurationim Amazon Kinesis Video Streams Developer Guide.

Über MKV-Tags des Herstellers

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 MKV-Tags von Produzenten

```
|+ 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

Amazon SNS SNS-Nachrichten

Dieses Thema enthält weitere Informationen zu Amazon SNS SNS-Nachrichten und Themen-Payloads.

Themen

- Nutzlast für Amazon SNS-Themen
- Ihre Amazon SNS SNS-Nachrichten anzeigen

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. Fragen Sie nach Nachrichten ab. Dieser Befehl sollte alle von der `PutMedia` Sitzung generierten Benachrichtigungen anzeigen. Informationen zu Umfragen finden Sie unter Amazon SQS Short and Long Polling.

Extrahieren Sie Bilder aus Videostreams

Sie können Amazon Kinesis Video Streams APIs verwenden, SDKs um Ihnen beim Extrahieren von Bildern aus Ihren Videostreams zu helfen. 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 automatisierte Bildextraktion aus Metadaten-Tags in einem aufgenommenen Video.

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

- On-Demand-Bildgenerierung (GetImages) — Mit dieser API können Kunden ein einzelnes Bild oder mehrere Bilder aus Videos extrahieren, die in Kinesis Video Streams gespeichert sind.
- the section called “Automatisierte Bildgenerierung (Amazon S3 S3-Lieferung)”- Konfigurieren Sie Kinesis Video Streams so, dass Bilder anhand von Tags in einem hochgeladenen Video automatisch in Echtzeit aus Videodaten extrahiert und an den vom Kunden angegebenen S3-Bucket gesendet werden.

Automatisierte Bildgenerierung (Amazon S3 S3-Lieferung)

Derzeit betreiben und verwalten Kunden ihre eigene Pipeline zur Bildtranskodierung, um Bilder für verschiedene Zwecke wie Scrubbing, Bildvorschau, Ausführung von ML-Modellen auf Bildern und mehr zu erstellen. Kinesis Video Streams bietet die Möglichkeit, die Bilder zu transkodieren und bereitzustellen. Kinesis Video Streams extrahiert basierend auf einem Tag automatisch Bilder aus Videodaten in Echtzeit und liefert die Bilder an einen vom Kunden angegebenen S3-Bucket.

Themen

- UpdateImageGenerationConfiguration
- DescribeImageGenerationConfiguration
- MKV-Tags für Hersteller
- Fügen Sie Metadaten-Tags im Producer-SDK hinzu mit PutEventMetaData
- Einschränkungen
- S3-Objekt-Metadaten
- Amazon S3 S3-Objektpfad (Bild)

- Amazon S3 S3-URI-Empfehlungen zum Schutz vor Drosselung

UpdateImageGenerationConfiguration

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

1. Erstellen Sie einen S3-Bucket für die Imagegenerierung auf der Grundlage der im SDK hinzugefügten Tags mithilfe der neuen API. Notieren Sie sich den S3-URI, der im nächsten Schritt erforderlich ist, wenn Sie die Konfigurationen der Image-Generierung für die Streams aktualisieren.
2. Erstellen Sie eine JSON-Datei namens 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 den UpdateImageGenerationConfigurationAPI-Vorgang aufrufen AWS CLI , um den zuvor erstellten Amazon S3 S3-ARN hinzuzufügen und den Status in zu ENABLED ändern.

```
aws kinesisvideo 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 für die Imagegenerierung dauert es mindestens 1 Minute, bis der Workflow zur Image-Generierung gestartet wird. Warten Sie `PutMedia` nach dem Aktualisierungsaufruf mindestens 1 Minute, bevor Sie den Vorgang aufrufen.

DescribeImageGenerationConfiguration

Um Konfigurationen zur Image-Generierung anzuzeigen, die bereits für einen Stream festgelegt wurden, können Kunden wie folgt eine `DescribeImageGenerationConfiguration` Anfrage stellen.

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'
    }
  }
```

Weitere Informationen zu dieser `DescribeImageGenerationConfiguration` Funktion finden Sie DescribeImageGenerationConfiguration im Amazon Kinesis Video Streams Developer Guide.

MKV-Tags für Hersteller

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 in diesem Code. Beim Aufrufen dieser API fügt das SDK zusammen mit den Fragmentdaten einen Satz vordefinierter MKV-Tags hinzu. Kinesis Video Streams erkennt diese speziellen MKV-Tags und leitet den Bildgenerierungs-Workflow auf der Grundlage der Bildverarbeitungskonfiguration dieses Streams ein.

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

Syntax für MKV-Tags von Produzenten

```
|+ 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
```

Fügen Sie Metadaten-Tags im Producer-SDK hinzu mit PutEventMetaData

Die `PutEventMetaData` Funktion hängt eine MKV-Datei an, die einem Ereignis zugeordnet ist. `PutEventMetaData`benötigt zwei Parameter. Der erste Parameter ist ein Ereignis, dessen Wert aus der `STREAM_EVENT_TYPE` Enumeration stammt. Der zweite Parameter, `pStreamEventMetadata`, ist optional und kann verwendet werden, um zusätzliche Metadaten als Schlüssel-Wert-Paare einzubeziehen. Es gibt eine Obergrenze 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	Einstellbar
Länge des Bildpräfixes	256	Nein
Optionale Länge des Metadatenschlüssels	128	Nein
Optionale Länge des Metadatenwerts	256	Nein
Maximale Anzahl optionaler Metadaten	10	Ja

S3-Objekt-Metadaten

Standardmäßig legt Kinesis Video Streams die Fragmentnummer, den Produzenten und den Serverzeitstempel des als Amazon S3 S3-Objektmetadaten generierten Bildes fest. Wenn zusätzliche

Fragmentdaten in den MKV-Tags angegeben sind, werden diese Tags auch zu den Amazon S3 S3-Objektmetadaten hinzugefügt. Das folgende Beispiel zeigt die korrekte Syntax für die Amazon S3 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,
}
```

Amazon S3 S3-Objektpfad (Bild)

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

Format:

ImagePrefix_AccountID_StreamName_ImageTimecode_RandomID.file-extension

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 S3-URI-Empfehlungen zum Schutz vor Drosselung

Wenn Sie Tausende von Bildern auf Amazon S3 schreiben, besteht die Gefahr einer Drosselung. Weitere Informationen finden Sie unter Grenzwerte für S3-Präfix-Put-Anfragen.

Ein Amazon S3 S3-Präfix beginnt mit einem PUT-Limit von 3.500 PUT-Anfragen pro Sekunde und wird im Laufe der Zeit für eindeutige Präfixe schrittweise erhöht. Vermeiden Sie die Verwendung von Datum und Uhrzeit als Amazon S3 S3-Präfixe. Zeitkodierte Daten wirken sich jeweils auf ein Präfix aus und ändern sich zudem regelmäßig, wodurch frühere Präfix-Skalierungen ungültig werden. Um eine schnellere, konsistente Amazon S3-Skalierung zu ermöglichen, empfehlen wir, der Amazon S3-Ziel-URI ein zufälliges Präfix wie einen Hex-Code oder eine UUID hinzuzufügen. Hex-Code-Präfixe teilen Ihre Anfragen beispielsweise nach dem Zufallsprinzip auf 16 verschiedene Präfixe auf (ein Präfix für jedes eindeutige Hex-Zeichen), was 56.000 PUT-Anfragen pro Sekunde ermöglicht, nachdem Amazon S3 automatisch skaliert hat.

Greifen Sie auf Videoanalysen zu

Dieser Abschnitt enthält Informationen darüber, wie Sie mithilfe der Parser-Bibliothek und Amazon CloudWatch auf Videoanalysen zugreifen können.

Themen

- Verwenden Sie in einen Kinesis-Videostream eingebettete Metadaten
- Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an
- Überwachung von Amazon Kinesis Video Streams
- Beschränkungen für Streaming-Metadaten

Verwenden Sie in einen Kinesis-Videostream eingebettete Metadaten

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 Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an 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.

Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an

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 die Kinesis-Videostream-Parser-Bibliothek verwendet wird.

Die Bibliothek umfasst auch Tests, die die Verwendung der Tools verdeutlichen.

Voraussetzungen

Sie benötigen Folgendes, um die Kinesis-Videostream-Parser-Bibliothek untersuchen und verwenden zu können:

- Ein Amazon Web Services (AWS) -Konto. Falls Sie noch kein Konto haben AWS-Konto, finden Sie weitere Informationen unterthe section called “Melden Sie sich an für ein AWS-Konto”.
- Eine integrierte Java-Entwicklungsumgebung (IDE) wie Eclipse Java Neon oder JetBrains IntelliJ Idea.
- Java 11, wie Amazon Corretto 11.

Laden Sie den Code herunter

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 the section called “Streamen Sie mithilfe der 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 Apache Maven-Projekt, das Sie heruntergeladen haben:

   - 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.

Untersuchen Sie den Code

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 `Streaming MkvReader`-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 wie private Codec-Daten, Pixelbreite oder Pixelhöhe enthalten.

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
 (MkvTypeInfos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
 {
                    MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
                    Frame frame =
 ((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
                    MkvTrackMetadata trackMetadata =
 fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
                    assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
                }
                if
 (MkvTypeInfos.SEGMENT.equals(mkvElement.get().getElementMetaData().getTypeInfo())) {
                    if (mkvElement.get() instanceof MkvEndMasterElement) {
                        if (segmentCount < continuationTokens.size()) {
                            Optional<String> continuationToken =
 fragmentVisitor.getContinuationToken();
                            Assert.assertTrue(continuationToken.isPresent());
                            Assert.assertEquals(continuationTokens.get(segmentCount),
 continuationToken.get());
                        }
                        segmentCount++;
                    }
                }
            }

        }
```

Das vorhergehende Beispiel weist folgendes Codierungsmuster auf:

- Ein `FragmentMetadataVisitor`-Objekt wird zum Analysieren der Daten und ein StreamingMkvReader-Objekt zum Bereitstellen der Daten erstellt.

- Bei jedem `MkvElement` im Stream wird geprüft, ob dessen Metadaten den Typ `SIMPLEBLOCK` haben.
- Ist dies der Fall, wird das `MkvDataElement` aus dem `MkvElement` abgerufen.
- Das `Frame`-Objekt (Mediendaten) wird aus `MkvDataElement` abgerufen.
- Das `MkvTrackMetadata`-Element des `Frame`-Objekts wird aus dem `FragmentMetadataVisitor`-Objekt abgerufen.
- Die folgenden Daten werden aus den `Frame`- und `MkvTrackMetadata`-Objekten abgerufen und geprüft:
  - Track-Nummer
  - Framehöhe in Pixeln
  - Framebreite in Pixeln
  - 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, kleiner 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
 (MkvTypeInfos.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 die Kinesis-Videostream-Parser-Bibliothek verwendet wird.

Diese Klasse führt die folgenden Operationen aus:

- Erzeugt einen Kinesis-Videostream. Wenn ein Stream mit dem angegebenen Namen bereits vorhanden ist, wird der Stream gelöscht und neu erstellt.
- Aufrufe PutMediazum Streamen von Videofragmenten in den Kinesis-Videostream.
- Aufrufe GetMediazum Streamen von Videofragmenten aus dem Kinesis-Videostream.
- 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"));
```

Rufen Sie an PutMedia

Das folgende Codebeispiel (aus der `PutMediaWorker.java` Datei) ruft den Stream PutMediaauf:

```
 putMedia.putMedia(new PutMediaRequest().withStreamName(streamName)
.withFragmentTimecodeType(FragmentTimecodeType.RELATIVE)
.withProducerStartTimestamp(new Date())
.withPayload(inputStream), new PutMediaAckResponseHandler() {
...
});
```

Rufen Sie GetMedia

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) ruft den Stream GetMediaauf:

```
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 mit GetMedia StreamingMkvReader

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) erstellt eine StreamingMkvReader und verwendet sie, um das Ergebnis der GetMediaOperation 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.

Fragmente abrufen 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.
- A`CompositeMkvElementVisitor`, das das FragmentMetadataVisitorthe section called “OutputSegmentMerger”, und zu einem einzigen Element zusammensetzt, `LogVisitor` den 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 den übergeben`GetMediaWorker`, der wiederum an den übergeben wird`ExecutorService`, der 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);
```

Führen Sie den Code aus

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`

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 Sie Amazon Kinesis Video Streams Streams-Metriken mit CloudWatch
- Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch
- API-Aufrufe von Amazon Kinesis Video Streams protokollieren mit AWS CloudTrail

Überwachen Sie 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
`ArchivedFragmentsConsumed.Media`	Die Anzahl der Quotenpunkte für Fragmentmedien, die APIs von allen genutzt wurden. Eine Erläuterung des Konzepts der Quotenpunkte finden Sie unterthe section called “Fragment-Metadaten und Kontingente für fragmentierte Medien”. Einheiten: Anzahl
`ArchivedFragmentsConsumed.Metadata`	Die Anzahl der Fragment-Metadatenquotenpunkte, die von allen genutzt wurden APIs. Eine Erläuterung des Konzepts der Quotenpunkte finden Sie unterthe section called “Fragment-Metadaten und Kontingente für fragmentierte Medien”. Einheiten: Anzahl
`PutMedia.Requests`	Die Anzahl der `PutMedia` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl
`PutMedia.IncomingBytes`	Die Anzahl der Bytes, die als Teil von `PutMedia` für den Stream empfangen wurden. Einheiten: Byte
`PutMedia.IncomingFragments`	Die Anzahl der vollständigen Fragmente, die als Teil von `PutMedia` für den Stream empfangen wurden.

Metrik	Beschreibung
	Einheiten: Anzahl
`PutMedia.IncomingFrames`	Die Anzahl der vollständigen Frames, die als Teil von `PutMedia` für den Stream empfangen wurden. Einheiten: Anzahl
`PutMedia.ActiveConnections`	Die Gesamtzahl der Verbindungen zum Service-H ost. Einheiten: Anzahl
`PutMedia.ConnectionErrors`	Die Fehler beim `PutMedia` Verbindungsaufbau für den Stream. Einheiten: Anzahl
`PutMedia.FragmentIngestionL atency`	Der Zeitunterschied zwischen dem Zeitpunkt, an dem das erste und das letzte Byte eines Fragments von Amazon Kinesis Video Streams empfangen werden. Einheiten: Millisekunden
`PutMedia.FragmentPersistLat ency`	Die Zeit, die vergeht, bis die vollständigen Fragmentd aten empfangen und archiviert wurden. Einheiten: Anzahl
`PutMedia.Latency`	Der Zeitunterschied zwischen der Anfrage und der HTTP-Antwort InletService beim Verbindungsaufbau. Einheiten: Anzahl
`PutMedia.BufferingAckLatency`	Der Zeitunterschied zwischen dem Empfang des ersten Byte eines neuen Fragments durch Amazon Kinesis Video Streams und dem Senden des Buffering ACK für das Fragment. Einheiten: Millisekunden

Metrik	Beschreibung
`PutMedia.ReceivedAckLatency`	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. Einheiten: Millisekunden
`PutMedia.PersistedAckLatency`	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. Einheiten: Millisekunden
`PutMedia.ErrorAckCount`	Die Anzahl der Fehler, die während des Streams ACKs `PutMedia` gesendet wurden. Einheiten: Anzahl
`PutMedia.Success`	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. Einheiten: Anzahl
`GetMedia.Requests`	Die Anzahl der `GetMedia` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl
`GetMedia.OutgoingBytes`	Die Gesamtzahl der Byte, die vom Dienst als Teil der `GetMedia` API für einen bestimmten Stream gesendet wurden. Einheiten: Byte

Metrik	Beschreibung
`GetMedia.OutgoingFragments`	Die Anzahl der Fragmente, die während der Bearbeitung `GetMedia` für den Stream gesendet wurden. Einheiten: Anzahl
`GetMedia.OutgoingFrames`	Die Anzahl der Frames, die während des `GetMedia` angegebenen Streams gesendet wurden. Einheiten: Anzahl
`GetMedia.MillisBehindNow`	Der Zeitunterschied zwischen dem aktuellen Serverzeitstempel und dem Serverzeitstempel des zuletzt gesendeten Fragments. Einheiten: Millisekunden
`GetMedia.ConnectionErrors`	Die Anzahl der Verbindungen, die nicht erfolgreich aufgebaut wurden. Einheiten: Anzahl

Metrik	Beschreibung
`GetMedia.Success`	1 für jedes erfolgreich gesendete Fragment; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter Protokollierung der Zusammenfassung von AWS Anfragen IDs und Antworten. Einheiten: Anzahl
`GetMediaForFragmentList.OutgoingBytes`	Die Gesamtzahl der Byte, die vom Dienst als Teil der `GetMediaForFragmentList` API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
`GetMediaForFragmentList.OutgoingFragments`	Die Gesamtzahl der Fragmente, die vom Dienst als Teil der `GetMediaForFragmentList` API für einen bestimmten Stream gesendet wurden. Einheiten: Anzahl
`GetMediaForFragmentList.OutgoingFrames`	Die Gesamtzahl der Frames, die vom Service als Teil der `GetMediaForFragmentList` API für einen bestimmten Stream gesendet wurden. Einheiten: Anzahl

Metrik	Beschreibung
`GetMediaForFragmentList.Requests`	Die Anzahl der `GetMediaForFragmentList` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl
`GetMediaForFragmentList.Success`	1 für jedes erfolgreich gesendete Fragment; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`ListFragments.Latency`	Die Latenz der `ListFragments` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`ListFragments.Requests`	Die Anzahl der `ListFragments` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`ListFragments.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetHLSStreamingSessionURL.Latency`	Die Latenz der `GetHLSStreamingSessionURL` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetHLSStreamingSessionURL.Requests`	Die Anzahl der `GetHLSStreamingSessionURL` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetHLSStreamingSessionURL.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetHLSMasterPlaylist.Latency`	Die Latenz der `GetHLSMasterPlaylist` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetHLSMasterPlaylist.Requests`	Die Anzahl der `GetHLSMasterPlaylist` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetHLSMasterPlaylist.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetHLSMediaPlaylist.Latency`	Die Latenz der `GetHLSMediaPlaylist` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetHLSMediaPlaylist.Requests`	Die Anzahl der `GetHLSMediaPlaylist` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetHLSMediaPlaylist.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetMP4InitFragment.Latency`	Die Latenz der `GetMP4InitFragment` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetMP4InitFragment.Requests`	Die Anzahl der `GetMP4InitFragment` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetMP4InitFragment.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetMP4MediaFragment.Latency`	Die Latenz der `GetMP4MediaFragment` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetMP4MediaFragment.Requests`	Die Anzahl der `GetMP4MediaFragment` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetMP4MediaFragment.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetMP4MediaFragment.OutgoingBytes`	Die Gesamtzahl der Byte, die vom Dienst als Teil der `GetMP4MediaFragment` API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
`GetTSFragment.Latency`	Die Latenz der `GetTSFragment` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetTSFragment.Requests`	Die Anzahl der `GetTSFragment` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetTSFragment.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetTSFragment.OutgoingBytes`	Die Gesamtzahl der Byte, die vom Dienst als Teil der `GetTSFragment` API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
`GetDASHStreamingSessionURL.Latency`	Die Latenz der `GetDASHStreamingSessionURL` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetDASHStreamingSessionURL.Requests`	Die Anzahl der `GetDASHStreamingSessionURL` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetDASHStreamingSessionURL.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetDASHManifest.Latency`	Die Latenz der `GetDASHManifest` API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
`GetDASHManifest.Requests`	Die Anzahl der `GetDASHManifest` API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Beschreibung
`GetDASHManifest.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetClip.Latency`	Die Latenz der GetClip API-Aufrufe für den angegebenen Videostreamnamen. Einheiten: Millisekunden
`GetClip.Requests`	Die Anzahl der GetClip API-Anfragen für einen bestimmten Videostream. Einheiten: Anzahl

Metrik	Beschreibung
`GetClip.Success`	1 für jede erfolgreiche Anfrage; 0 für jeden Fehler. Der durchschnittliche Wert zeigt die Erfolgsrate an. **Note** Ausfälle umfassen 400- (Benutzer) und 500-Fehler (System). Weitere Informationen zum Aktivieren einer Zusammenfassung von Anfragen und Antworten, einschließlich Anfragen, finden Sie unter AWS IDs Protokollierung der Zusammenfassung von Anfragen und Antworten. Einheiten: Anzahl
`GetClip.OutgoingBytes`	Die Gesamtzahl der Byte, die vom Dienst als Teil der GetClip API für einen bestimmten Videostream gesendet wurden. Einheiten: Byte

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 gibt`PutMedia.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 kommt`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 der Wert von `PutMedia.FragmentPersistLatency` oder zunimmt`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 wieder Verbindungen von Verbrauchern herstellen, 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 gibt`GetMedia.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 kommt`GetMediaForFragmentList.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.

Überwachen Sie den 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	Veröffentlicht, wenn ein `RecordJob` fataler Fehler auftritt. Einheiten: Keine. „1" wird einmal veröffentlicht, wenn dieses Ereignis eintritt. **Note** Weitere Informationen finden Sie in den Protokollen.
	Completed	Veröffentlicht, wenn a abgeschlossen `RecordJob` ist. Einheiten: Keine. „1" wird einmal veröffentlicht, wenn dieses Ereignis eintritt.

Dimensionen	Status	Beschreibung
Name des Streams, `UploadJob`	In Ausführung	Veröffentlicht kontinuierlich, wenn der `UploadJob` läuft. Einheiten: Keine. „1" wird veröffentlicht, solange sie `UploadJob` sich in diesem Status befindet.
	FatalError	Wird veröffentlicht, wenn die `UploadJob` fatalen Fehler auftreten. Einheiten: Keine. „1" wird einmal veröffentlicht, wenn dieses Ereignis eintritt. **Note** Weitere Informationen finden Sie in den Protokollen.
	Completed	Veröffentlicht, wenn der abgeschlossen `UploadJob` ist. Einheiten: Keine. „1" wird einmal veröffentlicht, wenn dieses Ereignis eintritt.
Stream-Name	Percentag eSpaceUsed	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”. Einheiten: Prozentsatz (Skala 0—1).

Dimensionen	Status	Beschreibung
Thing-Name	Am Leben	Veröffentlicht jede Minute vom Amazon Kinesis Video Streams Edge Agent aus, unabhängig davon, welche Konfigurationen darauf ausgeführt werden. Dies kann verwendet werden, um zu ermitteln, ob der Amazon Kinesis Video Streams Edge Agent aktiv ist und bereit ist, Konfigurationen zu akzeptieren. Einheiten: Keine. „1" wird jede Minute veröffentlicht.
	RecordJobs.HealthyJobCount	Gesamtzahl der laufenden und geplanten Aufzeichnungsaufträge auf dem Amazon Kinesis Video Streams Edge Agent. Einheiten: Anzahl.
	UploadJobs.HealthyJobCount	Gesamtzahl der laufenden und geplanten Upload-Jobs auf dem Amazon Kinesis Video Streams Edge Agent. Einheiten: Anzahl.
	RecordJobs.UnhealthyJobCount	Gesamtzahl der aktuell fehlerhaften Aufzeichnungsaufträge. Einheiten: Anzahl.
	UploadJobs.UnhealthyJobCount	Gesamtzahl der aktuell fehlerhaften Upload-Jobs. Einheiten: Anzahl.
	RecordJobs.RunningJobCount	Gesamtzahl der aktiv ausgeführten Rekordjobs. Einheiten: Anzahl.
	UploadJobs.RunningJobCount	Gesamtzahl der aktiv ausgeführten Upload-Jobs. Einheiten: Anzahl.

Dimensionen	Status	Beschreibung
	RecordJobs.EdgeConfigCount	Gesamtzahl der Datensatzkonfigurationen, die auf dem Amazon Kinesis Video Streams Edge Agent in Bearbeitung sind. Einheiten: Anzahl.
	UploadJobs.EdgeConfigCount	Gesamtzahl der Upload-Konfigurationen, die auf dem Amazon Kinesis Video Streams Edge Agent in Bearbeitung sind. Einheiten: Anzahl.

CloudWatch Leitfaden für Metriken für Amazon Kinesis Video Streams Edge Agent

CloudWatch Metriken können nützlich sein, um Antworten auf die folgenden Fragen zu finden:

Themen

- Verfügt der Amazon Kinesis Video Streams Edge Agent über ausreichend Speicherplatz für Aufzeichnungen?
- Ist der Amazon Kinesis Video Streams Edge Agent am Leben?
- Gibt es ungesunde Jobs?
- Erfordern irgendwelche Jobs externe Eingriffe?

Verfügt der Amazon Kinesis Video Streams Edge Agent über ausreichend Speicherplatz für Aufzeichnungen?

Relevante Kennzahlen: `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. Prüfen Sie die Protokolle und beheben Sie das Problem`StartEdgeConfigurationUpdate`, 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”

Erfordern irgendwelche Jobs externe Eingriffe?

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.

API-Aufrufe von Amazon Kinesis Video Streams protokollieren mit AWS CloudTrail

Amazon Kinesis Video Streams ist ein 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 Bereitstellung 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 CloudTrail userIdentity-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 ist eine einzelne Anforderung aus einer beliebigen Quelle und enthält Informationen zur angeforderten Aktion, zu Datum und Uhrzeit der Aktion, zu den Anforderungsparametern usw. CloudTrail -Protokolldateien stellen kein geordnetes Stack-Trace der öffentlichen API-Aufrufe dar. Daher werden sie nicht in einer bestimmten Reihenfolge angezeigt.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die CreateStreamAktion 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": "b7acfcd0-6ca9-4ee1-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"
        }
    ]
}
```

Beschränkungen für Streaming-Metadaten

Weitere Informationen zu den Beschränkungen, die the section called “Kontingente für Streaming-Metadaten” für das Hinzufügen von Streaming-Metadaten zu einem Kinesis-Videostream gelten, finden Sie unter.

Planen Sie die Videoaufnahme und Speicherung mit dem Amazon Kinesis Video Streams Edge Agent

Amazon Kinesis Video Streams bietet eine effiziente und kostengünstige Möglichkeit, eine Verbindung zu IP-Kameras vor Ort beim Kunden herzustellen. Mit dem Amazon Kinesis Video Streams Edge Agent können Sie Videos von den Kameras lokal aufnehmen 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 kurze Formular aus, um auf den Amazon Kinesis Video Streams Edge Agent zuzugreifen.

Sie können den Amazon Kinesis Video Streams Edge Agent herunterladen und ihn auf Ihren lokalen Edge-Computing-Geräten bereitstellen. Sie können sie auch problemlos in Docker-Containern bereitstellen, die auf EC2 Amazon-Instances ausgeführt werden. Nach der Bereitstellung können Sie die Amazon Kinesis Video Streams Streams-API verwenden, um die Konfigurationen für Videoaufzeichnung und Cloud-Upload 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 Amazon Kinesis Video Streams Edge Agent als AWS IoT Greengrass Komponente auf jedem AWS IoT Greengrass zertifizierten Gerät installieren. Weitere Informationen AWS IoT Greengrass dazu finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch.
- An AWS Snowball Edge: Sie können den Amazon Kinesis Video Streams Edge Agent auf Snowball Edge-Geräten ausführen. Weitere Informationen finden Sie im AWS Snowball Edge Edge-Entwicklerhandbuch.
- Bei einer nativen AWS IoT Bereitstellung: Sie können den Amazon Kinesis Video Streams Edge Agent nativ auf jeder Recheninstanz installieren. Das Edge-SDK verwendet AWS IoT Corefür die Verwaltung von Edge über den. the section called “Amazon Kinesis Video Streams”

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

Themen

- API-Operationen für Amazon Kinesis Video Streams Edge Agent
- Überwachen des Amazon Kinesis Video Streams Edge Agent
- Im AWS IoT Greengrass Nicht-Modus bereitstellen
- Stellen Sie den Amazon Kinesis Video Streams Edge Agent bereit für AWS IoT Greengrass
- Häufig gestellte Fragen zu Amazon Kinesis Video Streams Edge Agent

API-Operationen für Amazon Kinesis Video Streams Edge Agent

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

- the section called “StartEdgeConfigurationUpdate”
- the section called “DescribeEdgeConfiguration”
- the section called “DeleteEdgeConfiguration”
- the section called “ListEdgeAgentConfigurations”

Überwachen des Amazon Kinesis Video Streams Edge Agent

Informationen zur Überwachung Ihres Amazon Kinesis Video Streams Edge-Agenten finden Sie unterthe section called “Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch”.

Im AWS IoT Greengrass Nicht-Modus bereitstellen

Dieser Abschnitt enthält eine umfassende Anleitung zur Verwendung von Amazon Kinesis Video Streams außerhalb der AWS IoT Greengrass Umgebung. Unabhängig davon, ob Sie mit Edge-Geräten oder anderen Plattformen arbeiten, helfen Ihnen diese Informationen dabei, Kinesis Video Streams effektiv einzurichten und zu nutzen.

Sie finden detaillierte Informationen zu:

- Einrichtung der -Entwicklungsumgebung
- Einen Kinesis-Videostream erstellen
- Herunterladen und Kompilieren des Kinesis Video Streams Producer SDK
- Schreiben und Prüfen einer Beispielanwendung
- Die Beispielanwendung ausführen

Fahren Sie mit den folgenden Schritten fort, um den Amazon Kinesis Video Streams Edge Agent mit AWS IoT MQTT als eigenständige Bereitstellung auszuführen.

Themen

- Abhängigkeiten installieren
- Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs
- Erstellen Sie eine IAM-Berechtigungsrichtlinie
- Erstellen einer IAM-Rolle
- Erstellen Sie den Rollenalias AWS IoT
- Erstellen Sie die AWS IoT Richtlinie
- Erstelle ein AWS IoT Ding und erhalte AWS IoT Core Zugangsdaten
- Erstellen Sie den Amazon Kinesis Video Streams Edge Agent
- Installieren Sie den CloudWatch Agenten auf dem Gerät
- Führen Sie den Amazon Kinesis Video Streams Edge Agent als nativen Prozess aus

Abhängigkeiten installieren

Bevor Sie das Amazon Kinesis Video Streams Streams-Producer-SDK verwenden können, müssen Sie Ihre Entwicklungsumgebung mit den erforderlichen Abhängigkeiten einrichten. Diese Seite führt Sie durch den Prozess der Installation der erforderlichen Softwarekomponenten und Bibliotheken auf Ihrem System.

> **Note**
>
> Eine Liste der unterstützten Betriebssysteme finden Sie unterthe section called “Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent?”.

Installieren Sie Abhängigkeiten auf dem Gerät

1. Um den Amazon Kinesis Video Streams Edge Agent auszuführen, installieren Sie die folgenden geeigneten 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` von der Quelle aus.

   ```
   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
   ```

   `apache-maven-3.9.2`Von der Quelle aus installieren.

```
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 Sie auf einem Bildschirm darüber informiert werden, dass einige Dienste neu gestartet werden müssen, drücken Sie die Eingabetaste, um OK auszuwählen.

Weitere Informationen finden Sie im Amazon Corretto 11-Benutzerhandbuch.

2. Installieren Sie das. AWS Command Line Interface Informationen zur Installation oder Aktualisierung der neuesten Version der AWS CLI Verfahren finden Sie im AWS Command Line Interface Benutzerhandbuch.

Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs

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

Amazon Kinesis Video Streams erstellen

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

Öffnen Sie in der AWS Management Console die Amazon Kinesis Video Streams Streams-Konsole. Wählen Sie in der linken Navigationsleiste Videostreams aus.

Weitere Informationen finden Sie unter the section called “Erstellen Sie einen Amazon Kinesis Kinesis-Videostream”.

Erstelle Geheimnisse in AWS Secrets Manager

Öffnen Sie in der AWS Management Console die AWS Secrets Manager Konsole. Wählen Sie in der linken Navigationsleiste Secrets aus.

Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.

1. Wählen Sie Store a new secret (Ein neues Secret speichern).
   a. Schritt 1: Wählen Sie den geheimen Typ
      - 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 es falsch eingeben, funktioniert die Anwendung nicht.

        Wert:*`Your MediaURI`*.

        Example

        Beispiel:`rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/YourCameraMediaURI`.
   b. Schritt 2: Secret konfigurieren. Geben Sie diesem Geheimnis einen Namen. Nennen Sie es, wie Sie wollen.
   c. Schritt 3: Rotation konfigurieren — optional. Wählen Sie Weiter aus.
   d. Schritt 4: Überprüfen. Wählen Sie Store (Speichern) aus.
2. Wenn Ihr Geheimnis nicht sofort angezeigt wird, klicken Sie auf die Schaltfläche „Aktualisieren“.

   Wählen Sie den Namen Ihres Geheimnisses. Notieren Sie sich den geheimen ARN.
3. Wiederholen Sie diesen Vorgang für jedes MediaURI, von dem Sie streamen möchten.

> **Note**
>
> Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der EC2 Amazon-Instance aus darauf zugreifen oder wenn Sie unverwaltet laufen, während Sie mit dem VPN verbunden sind.

> **Important**
>
> Die RTSP-URL Ihrer Kamera sollte Videos im H.264-Format streamen. Die Fragmentdauer darf den unter genannten Grenzwert nicht überschreiten. the section called “SDK-Kontingente für Produzenten”
> Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.

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

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

Erstellen Sie eine IAM-Berechtigungsrichtlinie

Gehen Sie wie folgt vor, um eine IAM-Richtlinie zu erstellen. Diese Berechtigungsrichtlinie ermöglicht die selektive Zugriffskontrolle (eine Teilmenge der unterstützten Operationen) für eine AWS Ressource. In diesem Fall handelt es sich bei den AWS Ressourcen um die Videostreams, zu denen der Amazon Kinesis Video Streams Edge Agent streamen soll. Zu den Ressourcen gehören auch die AWS Secrets Manager Geheimnisse, die der Amazon Kinesis Video Streams Edge Agent abrufen kann. Weitere Informationen finden Sie unter IAM-Richtlinien.

Erstellen Sie eine Richtlinie mithilfe des JSON-Richtlinien-Editors

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 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:

JSON

```
{
    "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 das ARNs für die Videostreams und `arn:aws:secretsmanager:*:*:secret:*` ersetzen Sie es durch das, ARNs das die MediaURI-Secrets enthält, die Sie in erstellt haben. the section called “Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs” Verwenden Sie die ARNs für die Geheimnisse, 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 Policy-Restrukturierung im IAM-Benutzerhandbuch.

7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen und optional eine 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.

Erstellen einer IAM-Rolle

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

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

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 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:

   JSON

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

4. Wählen Sie das Kästchen neben der IAM-Richtlinie aus, die Sie in the section called “Erstellen Sie eine IAM-Berechtigungsrichtlinie” erstellt haben.
5. Wählen Sie Weiter aus.
6. Geben Sie einen Rollennamen oder ein Rollennamensuffix ein, damit Sie den Zweck dieser Rolle leichter erkennen können.

   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 anhängen.

   Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Tagging IAM-Ressourcen im IAM-Benutzerhandbuch.
9. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Erstellen Sie den Rollenalias AWS IoT

Gehen Sie wie folgt vor, um einen AWS IoT Rollenalias für die IAM-Rolle zu erstellen, die Sie in erstellt haben. the section called “Erstellen einer IAM-Rolle” Ein Rollenalias ist ein alternatives Datenmodell, das auf die IAM-Rolle verweist. Eine Anfrage eines Anbieters für AWS IoT Anmeldeinformationen muss einen Rollenalias enthalten, der angibt, welche IAM-Rolle übernommen werden soll, um temporäre Anmeldeinformationen von AWS Security Token Service ()AWS STS zu erhalten. Weitere Informationen finden Sie unter So verwenden Sie ein Zertifikat, um ein Sicherheitstoken zu erhalten.

Erstellen Sie den AWS IoT Rollenalias

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. Stellen Sie sicher, dass die entsprechende Region ausgewählt ist.
3. Wählen Sie in der linken Navigationsleiste Sicherheit und dann Rollenaliase 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, in der Sie sie erstellt haben. the section called “Erstellen einer IAM-Rolle”
7. Wählen Sie Erstellen aus. Auf der nächsten Seite sehen Sie einen Hinweis, dass Ihr Rollenalias erfolgreich erstellt wurde.
8. Suchen Sie nach dem neu erstellten Rollenalias und wählen Sie ihn aus. Notieren Sie sich den Rollenalias ARN. Sie benötigen dies für die AWS IoT Richtlinie im nächsten Schritt.

Erstellen Sie die AWS IoT Richtlinie

Gehen Sie wie folgt vor, um eine AWS IoT Richtlinie zu erstellen, die an das Gerätezertifikat angehängt wird. Dadurch erhalten Sie Berechtigungen für AWS IoT Funktionen und können mithilfe des Zertifikats den Rollenalias übernehmen.

Mithilfe von 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 ausführen können:

- Connect zum AWS IoT Core Message Broker her
- Senden und Empfangen von MQTT-Nachrichten
- Den Geräteschatten einer Sache abrufen oder aktualisieren

Weitere Informationen finden Sie unter AWS IoT Core Richtlinien.

Verwenden Sie den AWS IoT Richtlinien-Editor, um eine AWS IoT Richtlinie zu erstellen

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 im linken Navigationsbereich 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 Tagging your AWS IoT resources im AWS IoT Core Developer Guide.
6. Wählen Sie den Tab JSON.
7. Fügen Sie das folgende JSON-Richtliniendokument ein:

   JSON

```
{
    "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**
>
> `your-role-alias-arn`Ersetzen Sie es durch den ARN des Rollenalias, in dem Sie ihn erstellt habenthe section called “Erstellen Sie den AWS IoT Rollenalias”.

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

Erstelle ein AWS IoT Ding und erhalte AWS IoT Core Zugangsdaten

Zu diesem Zeitpunkt haben Sie Folgendes erstellt:

- Eine IAM-Berechtigungsrichtlinie. Siehe the section called “Erstellen Sie eine IAM-Berechtigungsrichtlinie”.
- Eine IAM-Rolle mit angehängter Berechtigungsrichtlinie. Siehe the section called “Erstellen einer IAM-Rolle”.
- Ein AWS IoT Rollenalias für die IAM-Rolle. Siehe the section called “Erstellen Sie den AWS IoT Rollenalias”.
- Eine AWS IoT Richtlinie, die derzeit keiner AWS Ressource zugeordnet ist. Siehe the section called “Erstellen Sie die AWS IoT Richtlinie”.

Um eine AWS IoT Sache zu erstellen und zu registrieren und AWS IoT Core Zugangsdaten zu erhalten

1. Registrieren Sie das Gerät als ein AWS IoT Ding 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 Navigationsleiste Alle Geräte und dann Dinge aus.

   d. Wähle „Dinge erstellen“.

   e. Wähle „Einzelnes Ding erstellen“ und dann „Weiter“.

      1. Schritt 1. Geben Sie die Eigenschaften der Dinge an

         Geben Sie einen Namen für Ihr Ding ein und wählen Sie dann Weiter.

      2. Schritt 2. Gerätezertifikat konfigurieren

         Wählen Sie Neues Zertifikat automatisch generieren (empfohlen) und dann Weiter aus.

      3. Schritt 3. Ordnen Sie dem Zertifikat Richtlinien zu

         Suchen Sie nach der Berechtigungsrichtlinie, die Sie in erstellt habenthe section called “Erstellen Sie die AWS IoT Richtlinie”.

         Aktivieren Sie das Kontrollkästchen neben Ihrer Richtlinie und wählen Sie Ding erstellen aus.

   f. Laden Sie in dem daraufhin angezeigten Fenster die folgenden Dateien herunter:

      - Gerätezertifikat. Dies ist das X.509-Zertifikat.
      - Öffentliche Schlüsseldatei
      - 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 Ding erfolgreich erstellt wurde.

h. Übertrage die oben heruntergeladenen Dateien auf dein AWS IoT Ding, falls sie nicht schon da sind.

2. Besorgen Sie sich den Endpunkt des Credential Providers für Ihr AWS Konto.

   AWS CLI

   Führen Sie den folgenden Befehl aus:

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

   AWS Management Console

   Führen Sie in AWS CloudShellden folgenden Befehl aus:

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

   Notieren Sie sich diese Informationen für einen späteren Schritt.

3. Besorgen Sie sich den Gerätedatenendpunkt für Ihr AWS Konto.

   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 Core Konsole unter https://console.aws.amazon.com/iot/.
   2. Wählen Sie in der linken Navigationsleiste Einstellungen aus.
   3. Suchen Sie den Endpunkt Gerätedaten.

   Notieren Sie sich diese Informationen für einen späteren Schritt.

4. (Optional) Vergewissern Sie sich, dass 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 So verwenden Sie ein Zertifikat, um ein Sicherheitstoken zu erhalten.

Erstellen Sie den Amazon Kinesis Video Streams Edge Agent

Erstellen Sie den Amazon Kinesis Video Streams Edge Agent

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, suchen Sie in Ihrer E-Mail nach dem Download-Link. Wenn Sie das Formular nicht ausgefüllt haben, füllen Sie es hier aus.
2. Überprüfen Sie die Prüfsumme.
3. Extrahieren Sie die Binärdateien und die JAR-Datei auf Ihrem Gerät.

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

   Nach dem Extrahieren 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 sein, dass er die neueste Binär-Versionsnummer widerspiegelt. Bei einer Version 1.0.0 wird der Ordnername beispielsweise auf 1.0.0 festgelegt.

4. Erstellen Sie das Abhängigkeiten-Jar.

> **Note**
>
> Das im Lieferumfang von enthaltene JAR `kvs-edge-agent.tar.gz` enthält keine Abhängigkeiten. Gehen Sie wie folgt vor, um diese Bibliotheken zu erstellen.

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

   Typ `mvn clean package`.

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

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

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

6. Legen Sie Umgebungsvariablen mit den Werten aus den vorherigen Schritten fest. Die folgende Tabelle enthält Beschreibungen der Variablen.

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_REGION`	Ja	Die Region, die verwendet wird. Beispiel: us-west-2
`AWS_IOT_CA_CERT`	Ja	Dateipfad zum CA-Zertifikat, das verwendet wird, um über TLS eine Vertrauensstellung mit dem Back-End-Dienst herzustellen. Beispiel: `/file/path/to/AmazonRootCA1.pem`
`AWS_IOT_CORE_CERT`	Ja	Dateipfad zum X.509-Zertifikat. Beispiel: `/file/path/to/certificateID-certificate .pem.crt`
`AWS_IOT_CORE_CREDENTIAL_ENDPOINT`	Ja	Der Endpoint Provider-Endpunkt für AWS IoT Core Anmeldeinformationen für Ihr AWS Konto. Beispiel: `credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com`

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_IOT_CORE_DATA_ATS_ENDPOINT`	Ja	Der Endpunkt der AWS IoT Core Datenebene für Ihr AWS Konto. Beispiel: *`data-account-specific-prefix`* `.iot.`*`aws-region`* `.amazonaws.com`
`AWS_IOT_CORE_PRIVATE_KEY`	Ja	Dateipfad zu dem privaten Schlüssel, der im öffentlichen/privaten key pair verwendet wird. Weitere Informationen finden Sie unter Schlüsselverwaltung in. AWS IoT Beispiel: */`file/path/to/certificateID-private`* `.pem.key`
`AWS_IOT_CORE_ROLE_ALIAS`	Ja	Der Name des Rollenalias, der auf die AWS IAM-Rolle verweist, mit der eine Verbindung hergestellt werden soll AWS IoT Core. Beispiel: `kvs-edge-role-alias`

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_IOT_CORE_THING_NAME`	Ja	Der Name des Objekts AWS IoT , auf dem die Anwendung ausgeführt wird. Beispiel: `my-edge-device-thing`
`GST_PLUGIN_PATH`	Ja	Dateipfad, der auf den Ordner verweist, der die `gstkvssink` und die `IngestorPipelineJNI` plattformabhängigen Bibliotheken enthält. Lassen Sie uns diese GStreamer Plugins laden. Weitere Informationen finden Sie unter the section called “Laden Sie das GStreamer Element herunter, erstellen Sie es und konfigurieren Sie es”. Beispiel: `/`*`download-location`*` /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*`EdgeAgentVersion`*` /`

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`LD_LIBRARY_PATH`	Ja	Dateipfad, der auf das Verzeichnis verweist, das die `cproducer` und die `KinesisVideoProducer` plattformabhängigen Bibliotheken enthält. Beispiel: */download-location* `/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/` *EdgeAgent Version* `/lib/`
`AWS_KVS_EDGE_CLOUDWATCH_ENABLED`	Nein	Legt fest, ob der Amazon Kinesis Video Streams Edge Agent Statistiken zur Auftragsintegrität auf Amazon CloudWatch veröffentlicht. Zulässige Werte:`TRUE`/`FALSE`(Groß- und Kleinschreibung wird nicht beachtet). Der Standardwert ist, `FALSE` falls nicht angegeben. Beispiel: FALSE

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_KVS_EDGE_LOG_LEVEL`	Nein	Die Ebene der Protokollierung der Amazon Kinesis Video Streams Edge Agent-Ausgaben. Zulässige Werte: • OFF • ALL • TÖDLICH • ERROR • WARN • INFO, Standard, falls nicht angegeben • DEBUG • TRACE Beispiel: `INFO`
`AWS_KVS_EDGE_LOG_MAX_FILE_SIZE`	Nein	Sobald die Protokolldatei diese Größe erreicht hat, erfolgt ein Rollover. • Min.: 0 • Maximal: 10000 • Standard: 20, falls nicht angegeben • Einheiten: Megabyte (MB) Beispiel: 5

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY`	Nein	Der Dateipfad, der auf das Verzeichnis verweist, in dem die Amazon Kinesis Video Streams Edge Agent-Protokolle ausgegeben werden. Der Standardwert ist, `./log` falls nicht angegeben. Beispiel: */file/path/*
`AWS_KVS_EDGE_LOG_ROLLOVER_COUNT`	Nein	Die Anzahl der Rollover-Logs, die vor dem Löschen beibehalten werden sollen. • Min.: 1 • Maximal: 100 • Standard: 10, falls nicht angegeben Beispiel: 20
`AWS_KVS_EDGE_RECORDING_DIRECTORY`	Nein	Dateipfad, der auf das Verzeichnis verweist, in das aufgezeichnete Medien geschrieben werden. Standardmäßig wird das aktuelle Verzeichnis verwendet, falls nicht angegeben. Beispiel: */file/path/*

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`GST_DEBUG`	Nein	Gibt die Ebene der GStreamer auszugebenden Protokolle an. Weitere Informationen finden Sie in der GStreamer -Dokumentation. Beispiel: 0
`GST_DEBUG_FILE`	Nein	Gibt die Ausgabedatei der GStreamer Debug-Logs an. Wenn diese Option nicht gesetzt ist, werden Debug-Logs mit dem Standardfehler ausgegeben. Weitere Informationen finden Sie in der GStreamer -Dokumentation. Beispiel: `/tmp/gstreamer-logging .log`

7. Leeren 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
`java.library.path`	Nein	Dateipfad, der auf den Ordner verweist, der die `gstkvssink` und die

Java-Eigenschaftsname	Erforderlich	Beschreibung
		`IngestorPipelineJNI` abhängigen Bibliotheken 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 `-D`*`java-property-name`*`=`*`value`* sie dem Java-Befehl hinzu, mit dem das JAR ausgeführt wurde.

Zum Beispiel:

```
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 aus wie/*download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*.

9. Senden Sie Konfigurationen mit dem 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 Uhr bis 16:59:59 Uhr aufzeichnet (je nach Systemzeit auf Ihrem Gerät). AWS IoT Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 Uhr 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 das AWS CLI Feld ein:

```
aws kinesisvideo 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.

Installieren Sie den CloudWatch Agenten auf dem Gerät

> **Note**
>
> Beachten Sie die CloudWatchKontingente.

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 hochgeladen werden. CloudWatch Dieser Schritt ist optional.

Anweisungen zur Installation des CloudWatch Agenten auf Ihrem Gerät finden Sie im CloudWatch Amazon-Benutzerhandbuch.

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

> **Important**
>
> `file_path`Bei den folgenden Konfigurationen wird davon ausgegangen, dass der Standardspeicherort für die Protokollausgabe verwendet wird.

Der verwendete Dateipfad setzt voraus, dass Sie den Amazon Kinesis Video Streams Edge Agent vom folgenden Standort aus ausführen:*download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*version*.

- Um den CloudWatch Agenten so zu konfigurieren, dass er Protokolle hochlädt und RAM- und CPU-Metriken auf das Gerät zurücksendet, 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/java_kvs.log",
            "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
            "log_stream_name": "YourIotThingName-java_kvs.log"
          },
          {
            "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"
          }
        ]
      }
    }
  }
}
```

- Verwenden Sie die folgende Konfiguration, um nur die Protokolle hochzuladen und nicht die RAM- und CPU-Daten des Geräts zu erfassen:

```
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/java_kvs.log",
            "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
            "log_stream_name": "YourIotThingName-java_kvs.log"
          },
```

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

Führen Sie den Amazon Kinesis Video Streams Edge Agent als nativen Prozess aus

Richten Sie den Amazon Kinesis Video Streams Edge Agent als Systemd-Service ein. Dieser Schritt ist optional.

`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 Amazon Kinesis Video Streams Edge Agent als nativen Prozess aus

1. Erstellen Sie eine neue Datei in `/etc/systemd/system` und geben Sie ihr *`aws.kinesisvideo.edge-runtime-agent`*`.service` einen Namen.

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` Dienstkonfigurationsdatei akzeptierten Parametern finden Sie in der Dokumentation.

> **Note**
>
> Fügen Sie die erforderlichen Umgebungsvariablen am `...` Speicherort hinzu, wie unter beschriebenthe section called “Erstellen Sie den Edge-Agenten”.

2. Laden Sie die Servicedateien neu, um den neuen Dienst einzubeziehen.

   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-Dienstes, um sicherzustellen, dass 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. Untersuchen Sie die Protokolle auf Fehler.

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

6. Geben Sie `systemctl --help` ein, um die vollständige Liste der Optionen anzuzeigen, mit denen der Prozess verwaltet werden soll`systemctl`.

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

   - Um neu zu starten, geben `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service` Sie ein.
   - Geben Sie ein, um den Vorgang zu beenden`sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
   - Um bei jedem Neustart des Geräts automatisch zu starten, geben Sie ein`sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Stellen Sie den Amazon Kinesis Video Streams Edge Agent bereit für AWS IoT Greengrass

Dieser Abschnitt enthält eine umfassende Anleitung zur Verwendung von Amazon Kinesis Video Streams mit AWS IoT Greengrass. Durch die Kombination dieser Dienste können Sie Videos effizient von Edge-Geräten in die Cloud streamen und so eine Vielzahl von Anwendungen in den Bereichen IoT, Überwachung und mehr ermöglichen.

Sie finden detaillierte Informationen zu:

- Einrichtung der -Entwicklungsumgebung
- Einen Kinesis-Videostream erstellen
- Eine Lambda-Funktion erstellen und verpacken
- Konfiguration des Kinesis Video Streams Streams-Core-Geräts
- Bereitstellung auf dem Kerngerät
- Dein Stream wird verifiziert

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

Themen

- Erstellen Sie eine EC2 Ubuntu-Amazon-Instanz
- Richten Sie das AWS IoT Greengrass V2 Kerngerät auf dem Gerät ein
- Erstellen Sie die Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera RTSP URLs
- Fügen Sie der Rolle Token Exchange Service (TES) Berechtigungen hinzu
- Installieren Sie die AWS IoT Greengrass Secret Manager-Komponente auf dem Gerät
- Stellen Sie die Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent-Komponente auf dem Gerät bereit
- Installieren Sie die AWS IoT Greengrass Log Manager-Komponente auf dem Gerät

Erstellen Sie eine EC2 Ubuntu-Amazon-Instanz

Gehen Sie wie folgt vor, um eine EC2 Ubuntu-Amazon-Instance zu erstellen.

Erstellen Sie eine EC2 Ubuntu-Amazon-Instanz

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-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 Instanz ein.
   - Anwendungs- und Betriebssystem-Images (Amazon Machine Image) — Wählen Sie Ubuntu aus.
   - Instanztyp — Wählen Sie t2.large aus.
   - Schlüsselpaar-Login — Erstellen Sie Ihr eigenes key pair.
   - Netzwerkeinstellungen — Behalten Sie die Standardeinstellung bei.
   - Speicher konfigurieren — Erhöhen Sie das Volumen auf 256 GiB.
   - Erweiterte Einstellungen — Behalten Sie die Standardeinstellung bei.
3. Starten Sie die Instance und greifen Sie per SSH auf sie zu.

   Gehen Sie wie folgt vor:

   1. Wählen Sie im linken Navigationsbereich Instances und dann die Instance-ID aus.
   2. Wählen Sie oben rechts Connect 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 ausführen, erhalten Sie die Meldung „Die Authentizität des Hosts (...) kann nicht festgestellt werden“. Geben Sie ja ein.
4. Installieren Sie Systembibliotheken, um den Amazon Kinesis Video Streams Edge Agent 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 Sie auf einem Bildschirm darüber informiert werden, dass einige Dienste neu gestartet werden müssen, drücken Sie die Eingabetaste, um OK auszuwählen.

Weitere Informationen finden Sie im Amazon Corretto 11-Benutzerhandbuch.

Richten Sie das AWS IoT Greengrass V2 Kerngerät auf dem Gerät ein

Gehen Sie wie folgt vor, um die AWS IoT Greengrass Nucleus-Kernsoftware auf der EC2 Amazon-Instance zu installieren.

Richten Sie das AWS IoT Greengrass Core-Gerät ein

1. Melden Sie sich bei AWS Management Console, an 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 Kerngerät einrichten aus.
4. Führen Sie die Schritte auf dem Bildschirm aus.

   - Schritt 1: Registrieren Sie ein Greengrass Core-Gerät. Geben Sie einen Namen für das Gerät ein.
   - Schritt 2: Zu einer Dinggruppe hinzufügen, um eine kontinuierliche Bereitstellung anzuwenden. Wählen Sie Keine Gruppe aus.
   - Schritt 3: Installieren Sie die Greengrass Core-Software. Wählen Sie Linux aus.
     - Schritt 3.1: Installieren Sie Java auf dem Gerät

       Java ist als Teil von installiertthe section called “Erstellen Sie eine Ubuntu-Instanz”. Kehren Sie zu diesem Schritt zurück, wenn Sie Java noch nicht installiert haben.

- Schritt 3.2: Kopieren Sie die AWS Anmeldeinformationen auf das Gerät

  Öffnen Sie die `bash/zsh` Option und fügen Sie die Exportbefehle in die EC2 Amazon-Instanz ein.

- Schritt 3.3: Führen Sie das Installationsprogramm aus

  1. Kopieren Sie die Befehle Installer herunterladen und Installer ausführen in der EC2 Ubuntu-Amazon-Instance und führen Sie sie aus.

     > **Note**
     >
     > Der Befehl Run the installer wird automatisch auf der Grundlage des Namens aktualisiert, den Sie in einem vorherigen Schritt ausgewählt haben.

  2. Notieren Sie sich die Rolle des Token Exchange Service (TES), die erstellt wurde. Sie benötigen sie später.

     > **Note**
     >
     > Standardmäßig heißt die erstellte Rolle GreenGrassV2 TokenExchangeRole.

Erstellen Sie die Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera RTSP URLs

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

Amazon Kinesis Video Streams erstellen

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

Öffnen Sie in der AWS Management Console die Amazon Kinesis Video Streams Streams-Konsole. Wählen Sie in der linken Navigationsleiste Videostreams aus.

Weitere Informationen finden Sie unter the section called “Erstellen Sie einen Amazon Kinesis Kinesis-Videostream”.

Erstelle Geheimnisse in AWS Secrets Manager

Öffnen Sie in der AWS Management Console die AWS Secrets Manager Konsole. Wählen Sie in der linken Navigationsleiste Secrets aus.

Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.

1. Wählen Sie Store a new secret (Ein neues Secret speichern).
   a. Schritt 1: Wählen Sie den geheimen Typ
      - 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 es falsch eingeben, funktioniert die Anwendung nicht.

        Wert:*`Your MediaURI`*.

        Example

        Beispiel:`rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/YourCameraMediaURI`.
   b. Schritt 2: Secret konfigurieren. Geben Sie diesem Geheimnis einen Namen. Nennen Sie es, wie Sie wollen.
   c. Schritt 3: Rotation konfigurieren — optional. Wählen Sie Weiter aus.
   d. Schritt 4: Überprüfen. Wählen Sie Store (Speichern) aus.
2. Wenn Ihr Geheimnis nicht sofort angezeigt wird, klicken Sie auf die Schaltfläche „Aktualisieren“.

   Wählen Sie den Namen Ihres Geheimnisses. Notieren Sie sich den geheimen ARN.
3. Wiederholen Sie diesen Vorgang für jedes MediaURI, von dem Sie streamen möchten.

> **Note**
>
> Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der EC2 Amazon-Instance aus darauf zugreifen oder wenn Sie unverwaltet laufen, während Sie mit dem VPN verbunden sind.
>
> > **Important**
> >
> > Die RTSP-URL Ihrer Kamera sollte Videos im H.264-Format streamen. Die Fragmentdauer darf den unter genannten Grenzwert nicht überschreiten. the section called “SDK-Kontingente für Produzenten”
> > Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.
>
> Führen Sie `gst-discoverer-1.0 Your RtspUrl` das Programm aus, um sicherzustellen, dass Ihre Kamera von Ihrem Gerät aus erreichbar ist.

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

Fügen Sie der Rolle Token Exchange Service (TES) Berechtigungen hinzu

Erteilen Sie dem Gerät, das die Rechte zum Einsehen der Geheimnisse annimmt, die Rolle des Token Exchange Service (TES). Dies ist erforderlich, damit die AWS Secrets Manager AWS IoT Greengrass Komponente ordnungsgemäß funktioniert.

Fügen Sie der TES-Rolle Berechtigungen hinzu

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 im linken Navigationsbereich Rollen aus 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 anhängen 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 das ARNs für die Streams, die Sie in einem vorherigen Schritt erstellt haben.

   JSON

```
{
    "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. Geben Sie Ihrer Richtlinie einen Namen 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 angehängt haben.

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

   Aktivieren Sie das Kontrollkästchen und wählen Sie Richtlinien anhängen aus.

   Auf dem nächsten Bildschirm sehen Sie den Hinweis, dass die Richtlinie erfolgreich an die Rolle angehängt wurde.

10. Erstellen Sie eine weitere Richtlinie und fügen Sie sie hinzu, diesmal für Ihre Geheimnisse.

    Ersetzen Sie die Richtlinie durch Folgendes:

    > **Note**
    >
    > `arn:aws:secretsmanager:*:*:secret:*`Ersetzen Sie es durch das ARNs, das die MediaURI-Geheimnisse enthält, die Sie in erstellt haben. the section called “Erstellen Sie Ressourcen für Ihre IP-Kamera RTSP URLs”

    JSON

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

11. Erstellen Sie eine weitere Richtlinie und fügen Sie sie hinzu, diesmal für Amazon CloudWatch Metriken. Ersetzen Sie die Richtlinie durch Folgendes:

    JSON

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

Installieren Sie die AWS IoT Greengrass Secret Manager-Komponente auf dem Gerät

Für den Amazon Kinesis Video Streams Edge Agent muss zuerst die AWS IoT Greengrass Secret Manager-Komponente auf dem Gerät installiert werden.

Installieren Sie die 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 Navigationsleiste Greengrass-Geräte, Deployments aus.

   Wählen Sie die Bereitstellung mit demselben Ziel wie die Bereitstellung, in der wir sie erstellt haben. the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”
3. Wählen Sie in der Drop-down-Liste „Aktionen“ in der oberen rechten Ecke die Option Überarbeiten aus.

Wählen Sie in dem daraufhin angezeigten Pop-up die Option Bereitstellung überarbeiten aus.

4. Füllen Sie die folgenden Abschnitte aus:

   - Schritt 1: Ziel angeben. Wählen Sie Weiter aus.
   - Schritt 2: Komponenten auswählen.
     - Stellen Sie sicher, dass die AWS.GreenGrass.cli-Komponente ausgewählt ist. Deinstallieren Sie diese Komponente nicht.
     - Schalten Sie die Option Nur ausgewählte Komponenten anzeigen ein und suchen Sie nach aws.greengrass. SecretManager.
     - Markieren Sie das Kästchen neben aws.greengrass. SecretManagerund wählen Sie dann Weiter.
   - Schritt 3: Komponenten konfigurieren. Konfigurieren Sie die AWS IoT Greengrass Secret Manager-Komponente so, dass sie die Secrets aus der AWS IoT Greengrass Umgebung herunterlädt.

     Wählen Sie aws.greengrass aus. SecretManagerKomponente und wählen Sie dann Komponente konfigurieren.

     Aktualisieren Sie auf dem daraufhin angezeigten Bildschirm das AWS Secrets Manager ARNs Feld Konfiguration zum Zusammenführen.

     > **Note**
     >
     > `arn:aws:secretsmanager:*:*:secret:*`Ersetzen Sie es durch das ARNs der Geheimnisse, die Sie in erstellt habenthe section called “Erstellen Sie 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 zur Secret Manager-Konfiguration im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

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

- Schritt 4: Konfigurieren Sie erweiterte Einstellungen. Klicken Sie auf Weiter.
- Schritt 5: Überprüfen. Wählen Sie Bereitstellen aus.

5. Vergewissern Sie sich, dass die AWS Secrets Manager Komponente und die Berechtigungen korrekt installiert wurden.

   Geben Sie auf der EC2 Ubuntu-Amazon-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. Untersuchen Sie die AWS IoT Greengrass Kernprotokolle.

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

   Überprüfen Sie, ob Fehler bei der Bereitstellung vorliegen.

   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 Hauptdienst neu zu starten.

   Wenn der Bereitstellungsfehler auf fehlende Berechtigungen zurückzuführen ist, überprüfen Sie den the section called “Fügen Sie der TES-Rolle Berechtigungen hinzu” Abschnitt, um sicherzustellen, dass die TES-Rolle über die richtigen Berechtigungen verfügt. Wiederholen Sie dann diesen Abschnitt.

Aktualisieren Sie die Secrets in der AWS IoT Greengrass Secret Manager-Komponente

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

Um die Geheimnisse der AWS IoT Greengrass Secret Manager-Komponente zu aktualisieren, folgen Sie den vorherigen Schritten 1—6 mit der folgenden Änderung.

Schritt 3: Komponenten konfigurieren. Konfigurieren Sie die AWS IoT Greengrass Secret Manager-Komponente so, dass sie die Secrets aus der AWS IoT Greengrass Umgebung herunterlädt.

Wählen Sie aws.greengrass aus. SecretManagerKomponente und wählen Sie dann Komponente konfigurieren.

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

Weitere Informationen finden Sie unter Updates zurücksetzen.

Stellen Sie die Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent-Komponente auf dem Gerät bereit

Gehen Sie wie folgt vor, um die Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent-Komponente auf dem Gerät bereitzustellen:

Stellen Sie die Komponente bereit

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

   Wenn Sie das Interessenformular für Amazon Kinesis Video Streams Edge Agent ausgefüllt haben, suchen Sie in Ihrer E-Mail nach dem Download-Link. Wenn Sie das Formular nicht ausgefüllt haben, füllen Sie es hier aus.
2. Überprüfen Sie die Prüfsumme.
3. Extrahieren Sie die Binärdateien und die JAR-Datei auf Ihrem Gerät.

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

Nach dem Extrahieren 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 sein, dass er die neueste Binär-Versionsnummer widerspiegelt. Bei einer Version 1.0.0 wird der Ordnername beispielsweise auf 1.0.0 festgelegt.

4. Erstellen Sie das Abhängigkeiten-Jar.

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

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

Typ `mvn clean package`.

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

5. Platzieren Sie die Datei 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 Greengrass Modus:

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_REGION`	Ja	Die Region, die verwendet wird. Beispiel: us-west-2 AWS IoT Greengrass Die Kernsoftware legt diesen Wert automatisch für Sie fest. Weitere Informationen finden Sie im Referenzthema „Umgebungsvariable für Komponenten“ im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`GST_PLUGIN_PATH`	Ja	Dateipfad, der auf den Ordner verweist, der die `gstkvssink` und die `IngestorPipelineJNI` plattformabhängigen Bibliotheken enthält. Dadurch können diese Plugins GStreamer geladen werden. Weitere Informationen finden Sie unter the section called “Laden Sie das GStreamer Element herunter, erstellen Sie es und konfigurieren Sie es”. Beispiel: `/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /`

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`LD_LIBRARY_PATH`	Ja	Dateipfad, der auf das Verzeichnis verweist, das die `cproducer` und die `KinesisVideoProducer` plattformabhängigen Bibliotheken enthält. Beispiel: */download-location* /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ *EdgeAgentVersion* /lib/
`AWS_KVS_EDGE_CLOUDWATCH_ENABLED`	Nein	Legt fest, ob der Amazon Kinesis Video Streams Edge Agent Statistiken zur Auftragsintegrität auf Amazon CloudWatch veröffentlicht. Zulässige Werte:`TRUE`/`FALSE`(Groß- und Kleinschreibung wird nicht beachtet). Der Standardwert ist, `FALSE` falls nicht angegeben. Beispiel: FALSE

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_KVS_EDGE_LOG_LEVEL`	Nein	Die Ebene der Protokollierung der Amazon Kinesis Video Streams Edge Agent-Ausgaben. Zulässige Werte: • OFF • ALL • TÖDLICH • ERROR • WARN • INFO, Standard, falls nicht angegeben • DEBUG • TRACE Beispiel: `INFO`
`AWS_KVS_EDGE_LOG_MAX_FILE_SIZE`	Nein	Sobald die Protokolldatei diese Größe erreicht hat, erfolgt ein Rollover. • Min.: 1 • Maximal: 100 • Standard: 20, falls nicht angegeben • Einheiten: Megabyte (MB) Beispiel: 5

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY`	Nein	Der Dateipfad, der auf das Verzeichnis verweist, in dem die Amazon Kinesis Video Streams Edge Agent-Protokolle ausgegeben werden. Der Standardwert ist, `./log` falls nicht angegeben. Beispiel: */file/path/*
`AWS_KVS_EDGE_LOG_ROLLOVER_COUNT`	Nein	Die Anzahl der Rollover-Logs, die vor dem Löschen beibehalten werden sollen. • Min.: 1 • Maximal: 100 • Standard: 10, falls nicht angegeben Beispiel: 20
`AWS_KVS_EDGE_RECORDING_DIRECTORY`	Nein	Dateipfad, der auf das Verzeichnis verweist, in das aufgezeichnete Medien geschrieben werden. Standardmäßig wird das aktuelle Verzeichnis verwendet, falls nicht angegeben. Beispiel: */file/path/*

Name der Umgebungsvariablen	Erforderlich	Beschreibung
`GREENGRASS_ROOT_DIRECTORY`	Nein	Der Dateipfad zum AWS IoT Greengrass Stammverzeichnis. Dies ist standardmäßig, `/greengrass/v2/` falls nicht angegeben. Beispiel: */file/path/*
`GST_DEBUG`	Nein	Gibt die Ebene der GStreamer auszugebenden Protokolle an. Weitere Informationen finden Sie in der GStreamer -Dokumentation. Beispiel: 0
`GST_DEBUG_FILE`	Nein	Gibt die Ausgabedatei der GStreamer Debug-Logs an. Wenn diese Option nicht gesetzt ist, werden Debug-Logs mit dem Standardfehler ausgegeben. Weitere Informationen finden Sie in der GStreamer -Dokumentation. Beispiel: */tmp/gstreamer-logging* `.log`

Öffnen `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` und ändern Sie das Run-Skript, um eine der obigen Umgebungsvariablen hinzuzufügen.

> **Important**
>
> Stellen Sie sicher, dass das geänderte Ausführungsskript keine Tabulatorzeichen enthält. Die AWS IoT Greengrass Kernsoftware wird das Rezept nicht lesen können.

7. Stellen Sie die Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent-Komponente 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 des AWS IoT Greengrass Version 2 Entwicklerhandbuchs:

   - AWS IoT Greengrass CLI-Befehle
   - Stellen Sie AWS IoT Greengrass Komponenten auf Geräten bereit

8. Senden Sie Konfigurationen mit dem 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 Uhr bis 16:59:59 Uhr aufzeichnet (je nach Systemzeit auf Ihrem Gerät). AWS IoT Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 Uhr 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 das Feld ein AWS 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.

Installieren Sie die AWS IoT Greengrass Log Manager-Komponente auf dem Gerät

> **Note**
>
> Beachten Sie die CloudWatchKontingente.

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

Installieren Sie die 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 an AWS Management Console 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, die in erstellt wurdethe section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”. Verwenden Sie bei Bedarf die Suchleiste.

   d. Wählen Sie die `GreengrassV2TokenExchangeRoleAccess` Richtlinie aus.

   e. Wählen Sie die Registerkarte JSON aus und vergewissern Sie sich, dass die Richtlinie wie folgt aussieht:

      JSON

```
{
    "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 existiert oder einige erforderliche Berechtigungen fehlen, erstellen Sie eine neue IAM-Richtlinie mit diesen Berechtigungen und fügen Sie sie der TES-Rolle hinzu, die in the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein” erstellt wurde.

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 Navigationsleiste Greengrass-Geräte, Deployments aus.

   Wählen Sie die Bereitstellung mit demselben Ziel wie die Installation, in der Sie sie erstellt haben. the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”
4. Wählen Sie in der oberen rechten Ecke Aktionen und anschließend Überarbeiten aus.

   Wählen Sie in dem daraufhin angezeigten Pop-up die Option Bereitstellung überarbeiten aus.
5. Füllen Sie die folgenden Abschnitte aus:

   a. Schritt 1: Ziel angeben. Wählen Sie Weiter aus.

   b. Schritt 2: Komponenten auswählen.

      i. Stellen Sie sicher, dass die Komponente aws.greengrass.cli und aws.greengrass vorhanden sind. SecretManagerKomponenten sind immer noch ausgewählt.

      > **Important**
      >
      > Deinstallieren Sie diese Komponenten nicht.

      ii. Schalten Sie die Option Nur ausgewählte Komponenten anzeigen ein und suchen Sie nach aws.greengrass. LogManager.

      iii. Wählen Sie das Kästchen neben aws.greengrass aus. LogManagerund wählen Sie dann Weiter.

c. Schritt 3: Komponenten konfigurieren. Konfigurieren Sie die AWS IoT Greengrass Log-Manager-Komponente so, dass sie die vom Amazon Kinesis Video Streams Edge Agent generierten Protokolle hochlädt.

Wählen Sie aws.greengrass aus. LogManagerKomponente und wählen Sie dann Komponente konfigurieren.

Fügen Sie auf dem daraufhin angezeigten Bildschirm die folgende Log Manager-Konfiguration in das Feld Zusammenzufügende Konfiguration 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**
>
> `logFileDirectoryPath`Bei der vorherigen Konfiguration wird davon ausgegangen, dass der Standardspeicherort für die Protokollausgabe verwendet wird.

> **Note**
>
> Weitere Informationen zu den einzelnen Parametern für die Log Manager-Konfiguration finden Sie im Abschnitt Log Manager im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

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

d. Schritt 4: Konfigurieren Sie erweiterte 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 entsprechenden Berechtigungen korrekt installiert wurden.

7. Geben Sie auf der EC2 Ubuntu-Amazon-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. Untersuchen Sie die AWS IoT Greengrass Kernprotokolle.

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

   Überprüfen Sie, ob Fehler bei der Bereitstellung vorliegen.

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

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

Wenn der Bereitstellungsfehler auf fehlende Berechtigungen zurückzuführen ist, überprüfen the section called “Fügen Sie der TES-Rolle Berechtigungen hinzu” Sie, ob die TES-Rolle über die erforderlichen Berechtigungen verfügt. Wiederholen Sie dann diesen Abschnitt.

Häufig gestellte Fragen zu Amazon Kinesis Video Streams Edge Agent

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

- 2.x
  - AMD64
- 18.x
  - ARM

AL2

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

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

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

Funktioniert der Amazon Kinesis Video Streams Edge Agent in AL2?

Ja.

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

Senden Sie ein weiteres the section called “StartEdgeConfigurationUpdate” an denselben`HubDeviceArn`, aber anderen Amazon Kinesis Video Streams AWS Secrets Manager ARNs Streams/.

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

Senden Sie eine Aktualisierung the section called “StartEdgeConfigurationUpdate” desselben `HubDeviceArn` mit demselben Amazon Kinesis Video Streams Streams-ARN. Wenn die Anwendung die Nachricht von Amazon Kinesis Video Streams empfängt, überschreibt sie die vorherige Konfiguration für diesen Stream. Die Änderungen werden dann stattfinden.

Haben Sie Beispiele für Gemeinsamkeiten**`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)`	
Jeden Tag von 9:00:00 Uhr bis 16:59:59 Uhr	`0 0 9-16 * * ? *`	3599
An Wochentagen von 9:00:00 Uhr bis 16:59:59 Uhr	`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

Beschreibung	ScheduleExpression	DurationInSeconds
9:00:00 Uhr — 16:59:59 Uhr am Wochenende	`0 0 9-16 ? * SAT,SUN *`	3599
An Wochentagen von 22:00:00 Uhr bis 23:59:59 Uhr	`0 0 22,23 ? * MON-FRI *`	3599
9:00:00 Uhr — 10:00:00 Uhr jeden Tag	`0 0 9 * * ? *`	3600
Jeden Tag von 16:00:00 Uhr bis 17:59:59 Uhr	`0 0 16-17 * * ? *`	3599

Weitere Beispiele finden Sie in der Quartz-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 von the section called “StartEdgeConfigurationUpdate” wird die Anzahl der Streams des Geräts nicht erhöht.

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

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

Informationen dazu, welche Jobs manuell neu gestartet werden müssen, finden Sie in the section called “Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch” der FatalErrorMetrik unter.

Senden Sie den erneutthe section called “StartEdgeConfigurationUpdate”, um den Job 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 “Überwachen Sie den 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
- Streamen ARNs
- 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 PrivateLink Hintergrund
- 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 die gesamte Bandbreite 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 PrivateLink Dokumentation.

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":"*"
      }
   ]
}
```

Beispiele für Amazon Kinesis Video Streams

Die folgenden Codebeispiele zeigen, wie Sie mit der Kinesis Video Streams Streams-API arbeiten:

Beispiele: Daten an Kinesis Video Streams senden

- Beispiel: GStreamer SDK-Plug-In für Kinesis Video Streams Producer — kvssink: Zeigt, wie das Kinesis Video Streams Streams-Producer-SDK für die Verwendung als GStreamer Ziel erstellt wird.
- Führen Sie das GStreamer Element in einem Docker-Container aus: Zeigt, wie Sie ein vorgefertigtes Docker-Image verwenden, um RTSP-Video (Real-Time Streaming Protocol) 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 API PutMedia: Zeigt, wie Sie mithilfe der Verwenden Sie die Java-Producer-Bibliothek API Daten an Kinesis Video Streams senden können, die bereits in einem Containerformat (MKV) vorliegen. PutMedia

Beispiele: Daten aus Kinesis Video Streams abrufen

- KinesisVideoExample: Zeigt, wie Videofragmente mithilfe der Kinesis Video Streams Parser Library analysiert und protokolliert werden.
- Beispiel: Analysieren und Rendern von Kinesis Video Streams Streams-Fragmenten: Zeigt, wie Kinesis-Videostream-Fragmente mit JCodecund analysiert und gerendert werden. JFrame

Beispiele: Wiedergeben von Videodaten

- Beispiel: Verwenden Sie HLS in HTML und JavaScript: Zeigt, wie eine HLS-Streaming-Sitzung für einen Kinesis-Videostream abgerufen und auf einer Webseite wiedergegeben wird.

Voraussetzungen

- Im Beispielcode geben Sie Anmeldeinformationen an, indem Sie ein Profil angeben, das Sie in Ihrer Profildatei mit den AWS Anmeldeinformationen festgelegt haben, oder

indem Sie Anmeldeinformationen in den Java-Systemeigenschaften Ihrer integrierten Entwicklungsumgebung (IDE) angeben. Falls Sie dies noch nicht getan haben, richten Sie zunächst Ihre Anmeldeinformationen ein. Weitere Informationen finden Sie unter AWS Zugangsdaten und Region für die Entwicklung einrichten.

- Wir empfehlen, eine der folgenden Java-IDEs zum Bearbeiten und Ausführen des Codes zu verwenden:
  - Eclipse Java Neon
  - JetBrains IntelliJ IDEA

Beispiel: GStreamer SDK-Plug-In für Kinesis Video Streams Producer — kvssink

In diesem Thema wird beschrieben, wie Sie das Amazon Kinesis Video Streams Streams-Producer-SDK zur Verwendung als GStreamer Plugin erstellen.

Themen

- Laden Sie das GStreamer Element herunter, erstellen Sie es und konfigurieren Sie es
- Führen Sie das Element aus GStreamer
- Beispiele für Startbefehle GStreamer
- Führen Sie das GStreamer Element in einem Docker-Container aus
- GStreamer Referenz zu den Elementparametern

GStreamerist ein beliebtes Medien-Framework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medien-Pipelines durch die Kombination modularer Plug-ins zu erstellen. Das Kinesis Video GStreamer Streams-Plugin optimiert die Integration Ihrer bestehenden GStreamer Medienpipeline mit Kinesis Video Streams. Nach der Integration GStreamer können Sie Videos von einer Webcam oder einer RTSP-Kamera (Real Time Streaming Protocol) an Kinesis Video Streams streamen, um sie in Echtzeit oder später wiederzugeben, zu speichern und weiter zu analysieren.

Das GStreamer Plugin verwaltet automatisch die Übertragung Ihres Videostreams zu Kinesis Video Streams, indem es die vom Kinesis Video Streams Streams-Producer-SDK bereitgestellten Funktionen in einem GStreamer Senkenelement kapselt,. `kvssink` Das GStreamer Framework bietet eine verwaltete Standardumgebung für die Erstellung des Medienflusses von einem Gerät

wie einer Kamera oder einer anderen Videoquelle zur weiteren Verarbeitung, zum Rendern oder Speichern.

Die GStreamer Pipeline besteht in der Regel aus der Verbindung zwischen einer Quelle (Videokamera) und dem Senkenelement (entweder ein Player zum Rendern des Videos oder ein 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 aufbauen, die Video von einer Videoquelle streamen kann, z. B. einer Webkamera oder einem RTSP-Stream, der normalerweise über Zwischencodierungsstufen (mit H.264-Kodierung) mit Kinesis Video Streams verbunden ist. Wenn Ihr Videostream als Kinesis-Videostream verfügbar ist, können Sie ihn the section called “Streamen Sie mithilfe der Parser-Bibliothek” für die weitere Verarbeitung, Wiedergabe, Speicherung oder Analyse Ihres Videostreams verwenden.

Laden Sie das GStreamer Element herunter, erstellen Sie es und konfigurieren Sie es

Das GStreamer Plugin-Beispiel ist im C++-Producer-SDK von Kinesis Video Streams enthalten. Weitere Informationen zu den Voraussetzungen und Downloadoptionen für das SDK finden Sie unter Laden Sie den Code der C++-Producer-Bibliothek herunter und konfigurieren Sie ihn.

Sie können die GStreamer Producer-SDK-Senke 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 befinden`GST_PLUGIN_PATH`. Führen Sie den folgenden Befehl aus:

```
export GST_PLUGIN_PATH=`pwd`/build
```

> **Note**
>
> Unter macOS können Sie nur Videos von einer Netzwerk-Kamera streamen, wenn sie GStreamer in einem Docker-Container ausgeführt werden. Video-Streaming von einer USB-Kamera auf macOS in einem Docker-Container wird nicht unterstützt.

Führen Sie das Element aus GStreamer

Verwenden Sie GStreamer den `gst-launch-1.0` Befehl, um das Producer-SDK-Element von Kinesis Video Streams als Senke zu verwenden. Verwenden Sie Upstream-Elemente, die für das zu verwendende 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.

Neben der Bereitstellung von Anmeldeinformationen und der Angabe einer Region hat das `kvssink` Element den folgenden erforderlichen Parameter:

- `stream-name`— Der Name des Ziels Kinesis Video Streams.

Weitere Informationen zu optionalen `kvssink`-Parametern finden Sie unter GStreamer Referenz zu den Elementparametern.

Aktuelle Informationen zu GStreamer Plug-ins und Parametern finden Sie unter GStreamer Plugins. Sie können auch den Namen eines Elements oder Plugins `gst-inspect-1.0` gefolgt vom Namen eines GStreamer Elements oder Plugins verwenden, um dessen Informationen auszudrucken und zu überprüfen, ob es auf Ihrem Gerät verfügbar ist:

```
gst-inspect-1.0 kvssink
```

Wenn das Erstellen `kvssink` fehlgeschlagen ist oder GST_PLUGIN_PATH nicht richtig gesetzt ist, sieht Ihre Ausgabe ungefähr so aus:

```
No such element or plugin 'kvssink'
```

Beispiele für Startbefehle GStreamer

Die folgenden Beispiele zeigen, wie Sie das `kvssink` GStreamer Plugin verwenden, um Videos von verschiedenen Gerätetypen zu streamen.

Beispiel 1: Streamen Sie Video von einer RTSP-Kamera auf Ubuntu

Der folgende Befehl erstellt unter Ubuntu eine GStreamer Pipeline, die mithilfe des GStreamer rtspsrc-Plugins von einer Netzwerk-RTSP-Kamera streamt:

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
 rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Beispiel 2: Video von einer USB-Kamera auf Ubuntu kodieren und streamen

Der folgende Befehl erstellt eine GStreamer Pipeline auf Ubuntu, die den Stream von einer USB-Kamera im H.264-Format codiert und an Kinesis Video Streams streamt. In diesem Beispiel wird das v4l2src-Plugin verwendet. GStreamer

```
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 Sie vorcodiertes Video von einer USB-Kamera auf Ubuntu

Mit dem folgenden Befehl wird auf Ubuntu eine GStreamer Pipeline erstellt, die Videos, die die Kamera bereits im H.264-Format codiert hat, an Kinesis Video Streams streamt. In diesem Beispiel wird das v4l2src-Plugin verwendet. GStreamer

```
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 Netzwerk-Kamera auf macOS streamen

Mit dem folgenden Befehl wird auf macOS eine GStreamer Pipeline erstellt, die Video von einer Netzwerkkamera zu Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer rtspsrc-Plugin verwendet.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
 rtph264depay ! h264parse ! 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 Sie Video von einer Netzwerk-Kamera unter Windows

Mit dem folgenden Befehl wird unter Windows eine GStreamer Pipeline erstellt, die Video von einer Netzwerk-Kamera zu Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer rtspsrc-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: Streamen Sie Video von einer Kamera auf dem Raspberry Pi

Der folgende Befehl erstellt eine GStreamer Pipeline auf dem Raspberry Pi, die Video an Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer v4l2src-Plugin 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: Streamen Sie sowohl Audio als auch 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 8: Streamen Sie sowohl Audio als auch Video von Gerätequellen 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 9: Laden Sie eine MKV-Datei hoch, 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. Sie benötigen eine MKV-Testdatei mit h.264- und AAC-codierten Medien.

Führen Sie das GStreamer Element in einem Docker-Container aus

Docker ist eine Plattform für die Entwicklung, Bereitstellung und das Ausführen von Anwendungen mithilfe von Containern. Durch die Verwendung von Docker zur Erstellung der GStreamer Pipeline wird die Betriebsumgebung für Kinesis Video Streams standardisiert, was die Erstellung und Verwendung der Anwendung optimiert.

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 einem der unten angegebenen `docker pull` Befehle aus der Amazon Elastic Container Registry herunterladen.

Gehen Sie wie GStreamer folgt vor, um das Producer-SDK-Element von Kinesis Video Streams als Senke in einem Docker-Container auszuführen:

Themen

- Authentifizieren Sie Ihren Docker-Client
- Herunterladen des Docker-Images für Ubuntu, macOS, Windows oder Raspberry Pi
- Führen Sie das Docker-Image aus

Authentifizieren Sie Ihren Docker-Client

Authentifizieren Sie Ihren Docker-Client bei der Amazon-ECR-Registrierung, aus der Sie Ihr Image abrufen möchten. Sie müssen Authentifizierungstoken für jede verwendete Registrierung abrufen. Tokens sind 12 Stunden gültig. Weitere Informationen finden Sie unter Registry-Authentifizierung im Benutzerhandbuch zu Amazon-Elastic-Container-Registry.

Example : Authentifizieren mit Amazon ECR

Um sich bei Amazon ECR zu authentifizieren, kopieren Sie den folgenden Befehl und fügen Sie ihn wie gezeigt ein.

```
sudo 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

```
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
```

Führen Sie das Docker-Image aus

Führen Sie abhängig vom verwendeten Betriebssystem einen der folgenden Befehle aus, um das Docker-Image auszuführen:

Führen Sie das Docker-Image auf Ubuntu aus

```
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
```

Führen Sie das Docker-Image auf macOS aus

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/
kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Führen Sie das Docker-Image unter Windows aus

```
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
```

Führen Sie das Docker-Image auf Raspberry Pi aus

```
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 für die 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, um `kvssink` mithilfe der `gst-launch-1.0` eine Pipeline auszuführen, die für Ihr Gerät und Ihre Videoquelle geeignet ist. Beispiele für Pipelines finden Sie unterBeispiele für Startbefehle GStreamer .

GStreamer Referenz zu den Elementparametern

Um Video an das C++-SDK von Amazon Kinesis Video Streams Producer zu senden, geben Sie es `kvssink` als Senke oder Endziel der Pipeline an. Diese Referenz enthält Informationen zu den

erforderlichen und optionalen `kvssink`-Parametern. Weitere Informationen finden Sie unter the section called “GStreamer Plug-in — kvssink”.

Topics

- the section called “Geben Sie Anmeldeinformationen ein für `kvssink`”
- the section called “Geben Sie eine Region an `kvssink`”
- the section called “`kvssink`optionale Parameter”

Geben Sie Anmeldeinformationen ein für **`kvssink`**

Damit das `kvssink` GStreamer Element Anfragen stellen kann AWS, müssen Sie AWS Anmeldeinformationen angeben, die es beim Aufrufen des Amazon Kinesis Video Streams Streams-Service verwenden soll. Die Anbieterkette für Anmeldeinformationen sucht in der folgenden Reihenfolge nach Anmeldeinformationen:

1. AWS IoT Referenzen

Informationen zum Einrichten von AWS IoT Anmeldeinformationen finden Sie unterthe section called “Steuern des Zugriffs auf Kinesis Video Streams Streams-Ressourcen mithilfe AWS IoT”.

Der `iot-credentials` Parameterwert muss mit einer kommagetrennten Liste der folgenden *`key value`* =-Paare beginnen `iot-certificate,` und darauf folgen.

Schlüssel	Erforderlich	Beschreibung
`ca-path`	Ja	Dateipfad zum CA-Zertifikat, das verwendet wird, um über TLS eine Vertrauensstellung mit dem Back-End-Dienst herzustellen. Example Beispiel: */file/path/to/certificate.pem*
`cert-path`	Ja	Dateipfad zum X.509-Zertifikat.

Schlüssel	Erforderlich	Beschreibung
		Example Beispiel: `/file/path/to/certificateID -certificate.pem.crt`
`endpoint`	Ja	Der Endpoint Provider-Endpunkt für AWS IoT Core Anmeldeinformationen für Ihr AWS Konto. Weitere Informationen finden Sie im AWS IoT Entwicklerhandbuch. Example Beispiel: `credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com`
`key-path`	Ja	Dateipfad zu dem privaten Schlüssel, der im public/private key pair verwendet wird. Example Beispiel: `/file/path/to/certificateID -private.pem.key`

Schlüssel	Erforderlich	Beschreibung
`role-aliases`	Ja	Der Name des Rollenalias, der auf die AWS IAM-Rolle verweist, mit der eine Verbindung hergestellt werden soll AWS IoT Core. Example Beispiel: *KvsCameraIoTRoleAlias*
`iot-thing-name`	Nein	Das `iot-thing-name` ist optional. Wenn nicht angegeben, `iot-thing-name` wird der `stream-name` Parameterwert verwendet. Example Beispiel: *kvs_example_camera*

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 können, 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 Wert für das Sitzungstoken an, wenn Sie temporäre Sicherheitsanmeldeinformationen direkt aus AWS STS Vorgängen 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 für future Sitzungen persistent zu machen, legen Sie sie im Startskript Ihrer Shell fest.

3. **`access-key`**, **`secret-key`** Parameter

Um Anmeldeinformationen direkt als `kvssink` Parameter anzugeben, legen Sie die folgenden Parameter fest:

`kvssink`Name des Parameters	Erforderlich	Beschreibung
`access-key`	Ja	Der AWS Zugriffsschlüssel, der für den Zugriff auf Amazon Kinesis Video Streams verwendet wird.

kvssinkName des Parameters	Erforderlich	Beschreibung
`secret-key`	Ja	Der AWS geheime Schlüssel, der dem Zugriffsschlüssel zugeordnet ist.
`session-token`	Nein	Gibt den erforderlichen Wert für das Sitzungstoken an, wenn Sie temporäre Sicherheitsanmeldeinformationen direkt aus AWS STS Vorgängen verwenden.

Example

Verwenden statischer Anmeldeinformationen:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE"
```

Example

Temporäre Anmeldeinformationen verwenden:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE" session-token="STEXAMPLE"
```

4. Anmeldeinformationsdatei

> **Important**
>
> Wenn Sie eine der vorherigen Methoden ausgewählt haben, können Sie den `credential-path` `kvssink` Parameter nicht verwenden.

kvssinkName des Parameters	Erforderlich	Beschreibung
`credential-path`	Ja	Pfad zur Textdatei, die Anmeldeinformationen in einem bestimmten Format enthält.

Die Textdatei muss Anmeldeinformationen in einem der folgenden Formate enthalten:

- ANMELDEINFORMATIONEN *`YourAccessKey YourSecretKey`*
- REFERENZEN *`YourAccessKey Expiration 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 sie in zu verwenden`kvssink`, geben Sie Folgendes ein:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 credential-path="/home/ubuntu/credentials.txt"
```

> **Note**
>
> Die Ablaufzeit sollte in future mindestens 5 + 30 + 3 = 38 Sekunden betragen. Die Kulanzzeit ist definiert als die `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` Variable in `IotCredentialProvider.h`. Wenn die Anmeldeinformationen beim Start zu kurz vor dem Ablauf liegen`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 Anmeldeinformationsdatei vor Ablauf der Ablaufzeit abzüglich der Kulanzzeit von einer externen Quelle aktualisiert werden.

Geben Sie eine Region an **`kvssink`**

Die Reihenfolge der Suche nach Regionen ist wie folgt:

1. `AWS_DEFAULT_REGION`Die Umgebungsvariable wird zuerst überprüft. Wenn sie gesetzt ist, wird diese Region zur Konfiguration des Clients verwendet.
2. `aws-region`Der Parameter wird als Nächstes überprüft. Wenn er gesetzt ist, wird diese Region zur Konfiguration des Clients verwendet.
3. Wenn keine der vorherigen Methoden verwendet wurde, ist die `kvssink` Standardeinstellung. `us-west-2`

`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
`stream-name`	Der Name des Amazon Kinesis Kinesis-Ziel-Video streams. **Important** Wenn kein Streamnam e angegeben ist, wird der Standard-		

Parameter	Beschreibung	Einheit/Typ	Standard
	Streamname verwendet: „DEFAULT_STREAM“. Falls ein Stream mit diesem Standardnamen noch nicht existiert, wird er erstellt.		
`absolute-fragment-times`	Gibt an, ob absolute Fragmentzeiten verwendet werden.	Boolesch	true

Parameter	Beschreibung	Einheit/Typ	Standard
`access-key`	Der AWS Zugriffss chlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird. Sie müssen entweder AWS Anmeldein formationen festgeleg t haben oder diesen Parameter angeben. Geben Sie Folgendes ein, um diese Informationen bereitzustellen: `export AWS_ACCES S_KEY_ID=`		
`avg-bandwidth-bps`	Die erwartete durchschnittliche Bandbreite für den Stream.	Bits pro Sekunde	4194304

Parameter	Beschreibung	Einheit/Typ	Standard
`aws-region`	Die AWS-Region zu verwendenden. **Note** Sie können der Region auch die `AWS_DEFAULT_REGION` Umgebungsvariable zur Verfügung stellen. Die Umgebungsvariablen haben Vorrang, wenn sowohl die Umgebungsvariable als auch die kvssink-Parameter gesetzt sind. **Important** Die Region wird standardmäßig verwendet,	String	`"us-west-2"`

Parameter	Beschreibung	Einheit/Typ	Standard
	`us-west-2` sofern nicht anders angegeben.		
`buffer-duration`	Laufzeit des Streampuffers	Sekunden	120
`codec-id`	Die Codec-ID des Streams	String	`"V_MPEG4/ISO/AVC"`
`connection-staleness`	Die Zeit nach der der Stream Staleness Callback aufgerufen wird.	Sekunden	60
`content-type`	Der Inhaltstyp des Streams	String	`"video/h264"`
`fragment-acks`	Ob Fragment verwendet werden soll. ACKs	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-Call back:	Boolesch	true

Parameter	Beschreibung	Einheit/Typ	Standard
`key-frame-fragmentation`	Gibt an, ob Fragmente für einen Keyframe erstellt werden.	Boolesch	true
`log-config`	Der Protokollkonfigurationspfad	String	`"../kvs_log_configuration"`
`max-latency`	Die maximale Latenz für den Stream	Sekunden	60
`recalculate-metrics`	Gibt an, ob die Metriken neu berechnet werden.	Boolesch	true
`replay-duration`	Die Laufzeit, um die der aktuelle Reader bei Auftreten eines Fehlers "zurückgespult" werden muss, um die Wiedergabe zu wiederholen	Sekunden	40
`restart-on-error`	Gibt an, ob beim Auftreten von Fehlern ein Neustart ausgeführt wird.	Boolesch	true
`retention-period`	Die Zeitdauer, für die der Stream aufbewahrt wird	Stunden	2

Parameter	Beschreibung	Einheit/Typ	Standard
`rotation-period`	Der Schlüssel rotationszeitraum. Weitere Informati onen finden Sie unter Rotierende AWS KMS Schlüssel.	Sekunden	3600
`secret-key`	Der AWS geheime Schlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird. Sie müssen entweder AWS Anmeldein formationen festgeleg t haben oder diesen Parameter angeben. `export AWS_SECRE T_ACCESS_KEY=`		
`session-token`	Gibt den erforderl ichen Wert für das Sitzungstoken an, wenn Sie temporäre Sicherheitsanmelde informationen direkt aus AWS STS Vorgä ngen verwenden.		

Parameter	Beschreibung	Einheit/Typ	Standard
`storage-size`	Die Gerätespeichergröße in Mebibyte (MiB). Weitere Informationen zum Konfigurieren von Speichergeräten finden Sie unter StorageInfo.	Mebibyte (MiB)	128
`streaming-type`	Der Streaming-Typ. Gültige Werte sind: • 0: Echtzeit • 1: nahezu in Echtzeit (derzeit nicht unterstützt) • 2: offline	Enum `GstKvsSinkStreamingType`	0: Echtzeit
`timecode-scale`	Die MKV-Timecode-Skala	Millisekunden	1
`track-name`	Der Name des MKV-Tracks.	String	`"kinesis_video"`

Parameter	Beschreibung	Einheit/Typ	Standard
`iot-certificate`	AWS IoT Anmeldeinformationen, die im `kvssink` Element verwendet werden sollen. `iot-certificate` akzeptiert die folgenden Schlüssel und Werte: **Note** Das `iot-thing-name` ist optional. Wenn nicht angegeben, `iot-thing-name` wird der `stream-name` Parameterwert verwendet. • `endpoint=iotcredentialsproviderendpoint` • `cert-path=/localdirectorypat`	String	Keine

Parameter	Beschreibung	Einheit/Typ	Standard
	h /to/certi ficate • key-path=/ localdir ectorypath / to/private/ key • ca-path=/ localdir ectorypath/ to/ca-cert • role-alia ses =role-alia ses • iot-thing- name=YourIotTh ingName		

Beispiel: Senden von Daten an Kinesis Video Streams mithilfe der API PutMedia

Dieses Beispiel zeigt, wie die PutMediaAPI verwendet wird. Es zeigt, wie Daten gesendet werden, die bereits in einem Containerformat (MKV) vorliegen. Wenn Ihre Daten vor dem Senden in ein Containerformat zusammengestellt werden müssen (z. B. wenn Sie Kameravideodaten in Frames zusammenfügen), finden Sie weitere Informationen unter. Zu Kinesis Video Streams hochladen

> **Note**
>
> Die `PutMedia` Operation ist nur in 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:

- Laden Sie den Code herunter und konfigurieren Sie ihn
- Schreiben und untersuchen Sie den Code
- Führen Sie den Code aus und überprüfen Sie ihn

Laden Sie den Code herunter und konfigurieren Sie ihn

Folgen Sie den Schritten, um den Java-Beispielcode herunterzuladen, das Projekt in Ihre Java-IDE zu importieren, die Bibliotheksspeicherorte zu konfigurieren und den Code so zu konfigurieren, dass er Ihre AWS Anmeldeinformationen verwendet.

1. Erstellen Sie ein Verzeichnis und klonen Sie den Beispielquellcode aus dem GitHub Repository. Das `PutMedia`-Beispiel ist Bestandteil der Java.

   ```
   git clone https://github.com/awslabs/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 anschließend „Projektstruktur“.

     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.

Schreiben und untersuchen Sie den Code

Das `PutMedia`-API-Beispiel () weist folgendes Programmiermuster auf:`PutMediaDemo`

Themen

- Erstelle den PutMediaClient
- Streamen von Medien und Anhalten des Threads

Die Codebeispiele in diesem Abschnitt stammen aus der Klasse `PutMediaDemo`.

Erstelle den PutMediaClient

Für die Erstellung des `PutMediaClient` Objekts sind die folgenden Parameter erforderlich:

- 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 Sie die 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 dieVerwenden Sie die Java-Producer-Bibliothek, die Kinesis Video Streams Streams-Konsole oder die AWS CLI verwenden.

- 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();
```

Führen Sie den Code aus und überprüfen Sie ihn

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 Streams-Konsole oder mithilfe von. 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)
   ```

5. Erstellen Sie eine classpath-Zeichenfolge mit den Abhängigkeiten zum lokalen Repository in einer Datei:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. 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)
```

7. Führen Sie die Demo von der Befehlszeile aus wie folgt aus und geben Sie Ihre AWS Anmeldeinformationen ein:

```
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
```

8. Öffnen Sie die Kinesis Video Streams 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

Das C++ enthält eine Definition für einen Docker-Container, der eine Verbindung zu einer RTSP-Netzwerkkamera (Real-Time Streaming Protocol) herstellt. Durch die Verwendung von Docker wird die Betriebsumgebung für Kinesis Video Streams standardisiert, wodurch die Erstellung und Verwendung der Anwendung optimiert wird.

Im folgenden Verfahren wird gezeigt, wie Sie die RTSP-Demoanwendung einrichten und verwenden.

Themen

- Video-Tutorials
- Voraussetzungen
- Erstellen Sie das Docker-Image
- Führen Sie die RTSP-Beispielanwendung aus

Video-Tutorials

Dieses Video zeigt, wie Sie einen Raspberry Pi einrichten, um RTSP-Feeds an die AWS Cloud und an Amazon Kinesis Video Streams zu senden. Dies ist eine end-to-end Demonstration.

Dieses Video zeigt, wie Sie Bilder aus einem Feed aufnehmen, um mithilfe von Computer Vision und Amazon Rekognition die Bilder zu verarbeiten und Benachrichtigungen zu senden.

Voraussetzungen

Um die RTSP-Beispielanwendung Kinesis Video Streams ausführen zu können, 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-Netzwerkkameraquelle: Weitere Informationen zu empfohlenen Kameras finden Sie unter Systemanforderungen.

Erstellen Sie das Docker-Image

Erstellen Sie zunächst das Docker-Image, in dem die Demo-Anwendung ausgeführt werden soll.

1. Klonen Sie das Amazon Kinesis Video Streams Streams-Demo-Repository.

   ```
   git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
   ```

2. Wechseln Sie in das Verzeichnis, das die Dockerfile enthält. In diesem Fall ist es das Verzeichnis docker-rtsp.

   ```
   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 kennzeichnet es als rtspdockertest.

   ```
   docker build -t rtspdockertest .
   ```

4. Führen Sie den Befehl aus `docker images` und suchen Sie nach der Bild-ID, die mit gekennzeichnet ist. `rtspdockertest`

   In der Beispielausgabe unten `IMAGE ID` ist der beispielsweise`54f0d65f69b2`.

   ```
   REPOSITORY        TAG       IMAGE ID        CREATED           PLATFORM       SIZE
          BLOB SIZE
   rtspdockertest    latest    54f0d65f69b2    10 minutes ago    linux/arm64    653.1
    MiB    292.4 MiB
   ```

   Sie werden dies in einem späteren Schritt benötigen.

Führen Sie die RTSP-Beispielanwendung aus

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

Führen Sie die RTSP-Beispielanwendung aus

1. Starten Sie den Amazon Kinesis Video Streams Docker-Container 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 Kinesis-Videostreams und die URL der RTSP-Netzwerkkamera ein.

   > **Important**
   >
   > Wenn Sie temporäre Anmeldeinformationen verwenden, müssen Sie auch Ihre angeben. `AWS_SESSION_TOKEN` Sehen Sie sich das zweite Beispiel unten an.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Temporäre Anmeldeinformationen:

```
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
```

3. Melden Sie sich bei der Kinesis Video Streams Streams-Konsole an AWS Management Console und öffnen Sie sie.

   Sehen Sie sich den Stream an.

4. Um den Docker-Container zu verlassen, schließen Sie das Terminalfenster oder geben Sie ein`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=Your AWS Region; ./
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

Beispiel: Analysieren und Rendern von Kinesis Video Streams Streams-Fragmenten

Die Streamen Sie mithilfe der Parser-Bibliothek enthält eine Demo-Anwendung mit dem Namen`KinesisVideoRendererExample`, die das Parsen und Rendern von Amazon Kinesis Kinesis-Videostream-Fragmenten demonstriert. Das Beispiel dient JCodeczur Dekodierung der H.264-codierten Frames, die mit der Anwendung aufgenommen wurden. Beispiel: GStreamer SDK-

Plug-In für Kinesis Video Streams Producer — kvssink Nachdem der Frame mit dekodiert wurde JCodec, wird das sichtbare Bild mit gerendert. JFrame

Dieses Beispiel veranschaulicht, wie Sie folgende Aufgaben ausführen:

- Rufen Sie mithilfe der `GetMedia` API Frames aus einem Kinesis-Videostream ab und rendern Sie den Stream zur Anzeige.
- Zeigen Sie den Videoinhalt von Streams in einer benutzerdefinierten Anwendung an, anstatt die Kinesis Video Streams 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, die vor der Anzeige nicht dekodiert werden müssen.

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 Idea.

Renderer-Beispiel ausführen

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. Vergewissern Sie sich, dass die Option Ausgabeverzeichnis bei Neuerstellung löschen ausgewählt ist, und wählen Sie dann Build, Build Project 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
- Analysieren von MKV-Fragmenten in Frames
- Den Frame dekodieren und anzeigen

Senden von MKV-Daten

Im Beispiel werden MKV-Beispieldaten aus der `rendering_example_video.mkv` Datei gesendet, `PutMedia` um Videodaten an einen Stream mit dem Namen zu senden. 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 Rufen Sie an PutMedia in der Dokumentation zu Streamen Sie mithilfe der Parser-Bibliothek.

Analysieren von MKV-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 Rufen Sie GetMedia in der Dokumentation zu Streamen Sie mithilfe der Parser-Bibliothek.

Den Frame dekodieren und anzeigen

Das Beispiel dekodiert dann den Frame und zeigt ihn an mit. JFrame

Das folgende Code-Beispiel stammt aus der Klasse `KinesisVideoFrameViewer`, die `JFrame` erweitert:

```
 public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

Das Bild wird als Instanz von java.awt.image angezeigt. BufferedImage. Beispiele, die die Arbeit mit `BufferedImage` veranschaulichen, finden Sie unter Lesen/Laden eines Image.

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
- JoinStorageSessionAsViewer

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. Es muss für jedes AWS-Konto Band einzigartig sein AWS-Region.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. 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`

Erforderlich: Nein

SingleMasterConfiguration

Eine Struktur, die die Konfiguration für den `SINGLE_MASTER` Kanaltyp enthält. Die Standardkonfiguration für die Gültigkeitsdauer der Kanalnachricht beträgt 60 Sekunden (1 Minute).

Typ: SingleMasterConfiguration Objekt

Erforderlich: Nein

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.

Erforderlich: Nein

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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for 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. Das Minimum ist 1 Stunde.

Wenn der `DataRetentionInHours` Wert 0 ist, können Verbraucher immer noch die Fragmente verwenden, die im Service-Host-Puffer verbleiben. Dieser Puffer hat eine Aufbewahrungszeit von 5 Minuten und eine Speicherbegrenzung von 200 MB. Fragmente werden aus dem Puffer entfernt, wenn einer der Grenzwerte erreicht ist.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 2048 Zeichen.

Pattern: `.+`

Erforderlich: Nein

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 entscheiden`MediaType`, finden Sie Richtlinien unter Benennungsvoraussetzungen.

Zu den gültigen Werten gehören beispielsweise "video/h264" and "video/h264,audio/aac“.

Dieser Parameter ist optional; der Standardwert ist `null` (oder leer in JSON).

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Erforderlich: Nein

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 beträgt 1 Zeichen. 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: `^([\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: Nein

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

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, 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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 auf`DELETE_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 an`StreamName`. `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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Edge-Konfiguration gelöscht werden soll. Geben Sie entweder den `StreamName` oder den an`StreamARN`.

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
```

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for 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 beträgt 1 Zeichen. Maximale Länge beträgt 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

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

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-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 DescribeStreamAPI, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for 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 der überprüft, bevor der Stream gelöscht wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

Erforderlich: Nein

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

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

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, 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

VersionMismatchException

Die von Ihnen angegebene Stream-Version ist nicht die neueste Version. Verwenden Sie die DescribeStreamAPI, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 an`StreamName`. `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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, dessen Edge-Konfiguration Sie aktualisieren möchten. Geben Sie entweder den `StreamName` oder den an`StreamARN`.

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

{
   "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 beträgt 1 Zeichen. 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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Konfiguration zur Image-Generierung abgerufen werden soll. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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

{
   "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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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.

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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]+`

Erforderlich: Nein

ChannelName

Der Name des Kanal.

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

{
   "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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Benachrichtigungskonfiguration abgerufen werden soll. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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

{
   "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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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]+`

Erforderlich: Nein

ChannelName

Der Name des Signalkanals, den Sie beschreiben möchten.

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

{
   "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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeStream

Bedienung: Amazon Kinesis Video Streams

Gibt die aktuellsten Informationen über den angegebenen Stream zurück. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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]+`

Erforderlich: Nein

StreamName

Name des -Streams.

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

{
   "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

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

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 an`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]+`

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

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

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

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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. Wenn Sie `WEBRTC` als Protokoll angeben, der Signalkanal jedoch nicht für die Aufnahme konfiguriert ist, erhalten Sie den Fehler. `InvalidArgumentException`

`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

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-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

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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.

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

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

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 angeben`ChannelNameCondition`.

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

Erforderlich: Nein

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.

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

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-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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 a angeben`StreamNameCondition`.

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. Der Standardwert ist 10 000.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximaler Wert von 10000.

Erforderlich: Nein

NextToken

Wenn Sie diesen Parameter angeben und das Ergebnis einer `ListStreams` Operation gekürzt wird, gibt der Aufruf `NextToken` in der Antwort den 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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

StreamInfoList

Ein Array von `StreamInfo`-Objekten.

Typ: Array von StreamInfo-Objekten

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

InvalidArgumentException

Der Wert für diesen Eingabeparameter ist ungültig.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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 beträgt 1024 Zeichen.

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: `^([\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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 angeben`StreamARN`.

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 beträgt 1024 Zeichen.

Pattern: `[a-zA-Z0-9+/=]*`

Erforderlich: Nein

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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, für den Sie Tags auflisten möchten.

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

{
   "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 beträgt 1024 Zeichen.

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

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

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 IoT Hub-Gerät ausgeführt wird und bei Ihnen vor Ort eingerichtet 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_SYNCSYNC_FAILED`, bevor Sie diese API erneut verwenden können. Wenn Sie diese API während des Synchronisierungsvorgangs aufrufen, `ResourceInUseException` wird a 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 an`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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, dessen Edge-Konfiguration Sie aktualisieren möchten. Geben Sie entweder den `StreamName` oder den an`StreamARN`.

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

{
   "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 beträgt 1 Zeichen. 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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 spezifizieren, das bereits existiert, wird der Tag-Wert durch den Wert ersetzt, den Sie in der Anfrage angeben. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Tags im AWS Fakturierung und Kostenmanagement and 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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 spezifizieren, das bereits existiert, wird der Tag-Wert durch den Wert ersetzt, den Sie in der Anfrage angeben. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Tags im AWS Fakturierung und Kostenmanagement and Cost Management-Benutzerhandbuch.

Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, zu dem Sie das Tag oder die Tags hinzufügen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

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

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

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 Signalisierungskanals, 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: Zeichenfolgen-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. 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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 angeben`StreamARN`.

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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem Sie Tags entfernen möchten.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

TagKeyList

Eine Liste der Schlüssel der Tags, die Sie entfernen möchten.

Typ: Zeichenfolgen-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. 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

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

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 angeben`StreamARN`.

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 beträgt 1 Zeichen. Maximale Länge beträgt 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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, dessen Aufbewahrungszeitraum Sie ändern möchten.

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
```

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.

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 DescribeStreamAPI, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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

Erforderlich: Nein

StreamARN

Der Amazon-Ressourcenname (ARN) des Kinesis-Videostreams, von dem aus Sie die Konfiguration der Imagegenerierung 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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Konfiguration der Image-Generierung aktualisiert werden soll. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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
```

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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

Erforderlich: Nein

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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, von dem aus die Benachrichtigungskonfiguration aktualisiert werden soll. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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
```

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-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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 beträgt 1 Zeichen. Maximale Länge beträgt 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. Dieser Parameter und die Kanalnachrichten time-to-live sind für Kanäle mit dem `SINGLE_MASTER` Kanaltyp erforderlich.

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

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-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 DescribeStreamAPI, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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 beträgt 1 Zeichen. Maximale Länge beträgt 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 beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

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 an`MediaType`.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Erforderlich: Nein

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]+`

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

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

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

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 DescribeStreamAPI, um die neueste Version zu erhalten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Streams-Medien

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Media unterstützt:

- GetMedia
- PutMedia

GetMedia

Bedienung: 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, sortiert nach Fragmentnummer.

> **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, beginnend mit dem Chunk, den Sie in der Anfrage angeben.

Die `GetMedia` API ist so konzipiert, dass sie als Streaming-API über eine lang andauernde Verbindung funktioniert. Sie ist nicht für die herkömmliche Verwendung vorgesehen, RESTful bei der für jedes Fragment eine neue HTTP-Verbindung hergestellt und geschlossen wird. Wenn Sie die `GetMedia` API aufrufen, überträgt Kinesis Video Streams Fragmente kontinuierlich über eine persistente Verbindung mit langer Laufzeit unter Verwendung der HTTP-Chunked-Transfercodierung.

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**
>
> Wird `GetMedia` als Streaming-Verbindung mit langer Laufzeit verwendet, um mehrere Fragmente in einer einzigen persistenten Verbindung abzurufen. Verwenden Sie die `GetMedia` API nicht RESTful auf herkömmliche Weise, bei der Sie für jedes Fragment eine neue HTTP-Verbindung einrichten und schließen. Wenn Sie versuchen, mehr als drei

`GetMedia` Verbindungen gleichzeitig herzustellen, drosselt Kinesis Video Streams die letzten Verbindungen mit einem Fehler. `ConnectionLimitExceededException`

> **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-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 /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 angeben`streamName`.

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 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 des angeforderten Mediums.

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

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` Anruf 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 - Die im Chunk zurückgegebene Fragmentnummer.
- AWS_KINESISVIDEO_SERVER_TIMESTAMP - Serverzeitstempel des Fragments.
- AWS_KINESISVIDEO_PRODUCER_TIMESTAMP - Zeitstempel des Produzenten des Fragments.

Die folgenden Tags sind vorhanden, falls ein Fehler auftritt:

- AWS_KINESISVIDEO_ERROR_CODE - Zeichenkettenbeschreibung eines Fehlers, der GetMedia zum Abbruch geführt hat.
- 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 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 berechtigt, 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

PutMedia

Bedienung: 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-kodiertes 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.

- Die `PutMedia` API ist so konzipiert, dass sie als Streaming-API über eine lang andauernde Verbindung funktioniert. Sie ist nicht für die herkömmliche Verwendung vorgesehen, RESTful bei der für jedes Fragment eine neue HTTP-Verbindung hergestellt und geschlossen wird. Verwenden Sie bei Verwendung der `PutMedia` API die HTTP-Chunked-Transfercodierung, um Fragmente kontinuierlich über eine persistente Verbindung zu senden.
- Für jedes in einer `PutMedia` Sitzung empfangene Fragment sendet Kinesis Video Streams eine oder mehrere Bestätigungen. Mögliche Überlegungen zum clientseitigen Netzwerk können dazu führen, dass Sie nicht alle diese Bestätigungen erhalten, sobald sie generiert werden.

> **Note**
>
> Wird `PutMedia` als Streaming-Verbindung mit langer Laufzeit verwendet, um mehrere Fragmente in einer einzigen persistenten Verbindung zu senden. Wenn Sie versuchen, mehrere Verbindungen gleichzeitig `PutMedia` herzustellen, drosselt Kinesis Video Streams die letzten Verbindungen mit einem Fehler. `ConnectionLimitExceededException`

Bei der Verwendung der API gelten die folgenden Beschränkungen: `PutMedia`

- 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 (wie`track*`) 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

    `producer_timestamp`= `producerStartTimeStamp` + Fragment-Timecode

    Wenn `fragmentTimeCodeType` es absolut ist, dann

    `producer_timestamp`= Fragment-Timecode (in Millisekunden umgewandelt)
  - Eindeutige Fragmentnummer, die von Kinesis Video Streams zugewiesen wurde.

> **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. In AWS SDKs anderen Sprachen wird er 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 MPEG-TS und Fragment-Paketierung, wenn der Stream über die HLS abgerufen wird. MP4 APIs

> **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 (`Acknowledgement`Objekten) 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 ja`Error`, 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 einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams — Archivierte Medien

Die folgenden Aktionen werden von Amazon Kinesis Video Streams Archived Media unterstützt:

- GetClip
- GetDASHStreamingSessionURL
- GetHLSStreamingSessionURL
- GetImages
- GetMediaForFragmentList
- ListFragments

GetClip

Bedienung: 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.

StreamName Sowohl der Parameter als auch der 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 zu erhalten. Senden Sie dann die `GetClip` Anfragen mit dem Parameter --endpoint-url an diesen Endpunkt.

Für einen Amazon Kinesis Kinesis-Videostream gelten die folgenden Anforderungen für die Bereitstellung von Daten über MP4:

- Anforderungen an den Titel für die Videowiedergabe.
- 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` CloudWatch Amazon-Metrik überwachen. Informationen CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter Kinesis Video Streams überwachen. Preisinformationen finden Sie unter Amazon Kinesis Video Streams — Preise und AWS Preise. Für ausgehende AWS Daten fallen Gebühren an.

> **Important**
>
> Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten codecspezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. CPD-Änderungen zwischen den Zielfragmenten des resultierenden Clips werden nicht unterstützt. Die CPD muss auf den abgefragten Medien konsistent bleiben, andernfalls wird ein Fehler zurückgegeben.

> **Important**
>
> Änderungen nachverfolgen wird nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Es wird ein Fehler zurückgegeben, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

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 Zeitbereich des angeforderten Clips 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 Clip.

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 Kinesis Video Streams Streams-Kontingente.

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 Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einem der Tracks des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

InvalidMediaFrameException

Ein oder mehrere Frames im angeforderten Clip konnten nicht auf der Grundlage des angegebenen Codecs analysiert werden.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einem 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 Wert `DataRetentionInHours` von 0 hat).

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

UnsupportedStreamMediaTypeException

Der Medientyp (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte anhand IDs des Codec der Titel im ersten Fragment für eine Wiedergabesitzung nicht bestimmt werden. Die Codec-ID für Track 1 sollte `V_MPEG/ISO/AVC` und optional die Codec-ID für Track 2 sein. `A_AAC`

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetDASHStreamingSessionURL

Bedienung: Amazon Kinesis Video Streams Archived Media

Ruft eine MPEG-URL für dynamisches adaptives Streaming über HTTP (DASH) für den Stream ab. Sie können die URL dann in einem Media Player öffnen, um den Inhalt des Streams anzusehen.

`StreamName`Sowohl der als auch der `StreamARN` Parameter sind optional, aber Sie müssen entweder den `StreamName` oder den angeben, `StreamARN` wenn Sie diesen API-Vorgang aufrufen.

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

- Anforderungen an den Titel für die Videowiedergabe.
- 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 zu erhalten. Senden Sie dann die `GetDASHStreamingSessionURL` Anfragen mit dem Parameter --endpoint-url an diesen Endpunkt.
2. Rufen Sie die MPEG-DASH-URL ab mit. `GetDASHStreamingSessionURL` 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 Root-Ressource, die für das Streaming mit MPEG-DASH benötigt wird).

> **Note**
>
> Teilen oder speichern Sie dieses Token nicht dort, wo eine nicht autorisierte Entität darauf zugreifen kann. Das Token ermöglicht den Zugriff auf den Inhalt des Streams. Schützen

Sie das Token mit denselben Maßnahmen, die Sie für Ihre AWS Anmeldeinformationen verwenden.

Die Medien, die über das Manifest verfügbar gemacht werden, bestehen nur aus dem angeforderten Stream, Zeitbereich und Format. Es werden keine anderen Mediendaten (wie 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 weiter, 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 und andere Daten, die zur Einrichtung des Video- oder Audiodecoders und Renderers benötigt werden. Die Medienfragmente enthalten kodierte Videoframes oder kodierte Audiobeispiele.
4. Der Media Player empfängt die authentifizierte URL und fordert Stream-Metadaten und Mediendaten normal an. Wenn der Media Player Daten anfordert, ruft er die folgenden Aktionen auf:
   - AbrufenDASHManifest: Ruft ein MPEG DASH-Manifest ab, das die Metadaten für die Medien enthält, die Sie wiedergeben möchten.
   - Abrufen MP4InitFragment: Ruft das MP4 Initialisierungsfragment ab. Der Media Player lädt normalerweise das Initialisierungsfragment, bevor er Medienfragmente lädt. Dieses Fragment enthält die Atome `fytp` "" und "`moov`" sowie die untergeordneten MP4 Atome, die zur Initialisierung des Media Player-Decoders benötigt werden.

     Das Initialisierungsfragment entspricht keinem Fragment in einem Kinesis-Videostream. Es enthält nur die privaten Codec-Daten für den Stream und den jeweiligen Titel, die der Media Player benötigt, um die Medienframes zu dekodieren.
   - Abrufen MP4MediaFragment: Ruft MP4 Medienfragmente ab. Diese Fragmente enthalten die MP4 Atome `moof` "" und "`mdat`" und ihre untergeordneten Atome, die die Medienframes und ihre Zeitstempel des codierten Fragments enthalten.

     Important

     Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten Codec-spezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. CPD-Änderungen

werden während einer Streaming-Sitzung nicht unterstützt. Die CPD muss in den abgefragten Medien konsistent bleiben.

> **Important**
>
> Änderungen nachverfolgen wird nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Das Streaming schlägt fehl, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

Mit dieser Aktion abgerufene Daten sind abrechnungsfähig. Einzelheiten finden Sie unter Preise.

> **Note**
>
> Einschränkungen, die für MPEG-DASH-Sitzungen gelten, finden Sie unter Kinesis Video Streams Streams-Kontingente.

Sie können die Datenmenge, die der Media Player verbraucht, überwachen, indem Sie die `GetMP4MediaFragment.OutgoingBytes` CloudWatch Amazon-Metrik überwachen. Informationen CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter Kinesis Video Streams überwachen. Preisinformationen finden Sie unter Amazon Kinesis Video Streams — Preise und AWS Preise. Sowohl für HLS-Sitzungen als auch für ausgehende AWS Daten fallen Gebühren an.

Weitere Informationen zu HLS finden Sie unter HTTP Live Streaming auf der Apple Developer-Website.

> **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 AWS dem Support-Team ein Problem melden möchten, können Sie 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, sowie Informationen darüber bereitzustellen, 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 /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 Zeitbereich des angeforderten Fragments und die Quelle der Zeitstempel.

Dieser Parameter ist erforderlich, wenn er `ON_DEMAND` oder `PlaybackMode` `LIVE_REPLAY` ist. Dieser Parameter ist optional, falls PlaybackMode ja `LIVE`. Wenn `PlaybackMode` ja`LIVE`, `FragmentSelectorType` kann er gesetzt werden, `TimestampRange` sollte aber nicht gesetzt werden. Wenn `PlaybackMode` ist `ON_DEMAND` oder`LIVE_REPLAY`, `TimestampRange` müssen beide `FragmentSelectorType` gesetzt werden.

Typ: DASHFragmentSelector Objekt

Erforderlich: Nein

DisplayFragmentNumber

Fragmente werden in der Manifestdatei anhand ihrer Sequenznummer in der Sitzung identifiziert. Wenn auf gesetzt DisplayFragmentNumber ist`ALWAYS`, wird die Kinesis Video Streams Streams-Fragmentnummer zu jedem S-Element in der Manifestdatei mit dem Attributnamen „kvs:fn“ hinzugefügt. Diese Fragmentnummern können für die Protokollierung oder für die Verwendung mit anderen Fragmenten APIs (z. B. `GetMedia` und) verwendet werden. `GetMediaForFragmentList` Um dieses benutzerdefinierte Attribut nutzen zu können, ist ein benutzerdefinierter MPEG-DASH Media Player erforderlich.

Der Standardwert ist `NEVER`.

Typ: Zeichenfolge

Zulässige Werte: `ALWAYS | NEVER`

Erforderlich: Nein

DisplayFragmentTimestamp

Gemäß der MPEG-DASH-Spezifikation kann die Uhrzeit der Wanduhr von Fragmenten in der Manifestdatei anhand von Attributen im Manifest selbst abgeleitet werden. In der Regel können MPEG-DASH-kompatible Mediaplayer Lücken in der Medien-Timeline jedoch nicht richtig behandeln. Kinesis Video Streams passt die Medien-Timeline in der Manifestdatei an, um die Wiedergabe von Medien mit Unterbrechungen zu ermöglichen. Daher kann die

aus der Manifestdatei abgeleitete Uhrzeit der Wanduhr ungenau sein. Wenn auf gesetzt DisplayFragmentTimestamp ist`ALWAYS`, wird jedem S-Element in der Manifestdatei mit dem Attributnamen „kvs:ts“ der genaue Fragment-Zeitstempel hinzugefügt. Um dieses benutzerdefinierte Attribut nutzen zu können, ist ein benutzerdefinierter MPEG-DASH Media Player erforderlich.

Der Standardwert ist `NEVER`. Wenn dies der DASHFragmentSelector Fall ist`SERVER_TIMESTAMP`, sind die Zeitstempel die Startzeitstempel des Servers. In ähnlicher Weise sind die Zeitstempel die `PRODUCER_TIMESTAMP` Startzeitstempel des Produzenten, wenn dies der Fall DASHFragmentSelector ist.

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 keine neuen Aufrufe für diese Sitzung getätigt werden. `GetDashManifest` `GetMP4InitFragment`

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 ja `PlaybackModeLIVE`, werden die neuesten Fragmente bis zu diesem Wert zurückgegeben. Wenn der `PlaybackMode` Wert ist`ON_DEMAND`, werden die ältesten Fragmente zurückgegeben, bis zu dieser Höchstzahl.

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 Wiedergabe-Latenz, verringert jedoch die Wahrscheinlichkeit, dass während der Wiedergabe eine Zwischenpufferung auftritt. Wir empfehlen, dass ein Live-MPEG-DASH-Manifest mindestens 3 Fragmente und maximal 10 Fragmente enthält.

Die Standardeinstellung ist 5 Fragmente, wenn `PlaybackMode` es `LIVE` oder ist`LIVE_REPLAY`, und 1.000, wenn es ist. `PlaybackMode` `ON_DEMAND`

Der Höchstwert von 1.000 Fragmenten entspricht mehr als 16 Minuten Video bei Streams mit 1-Sekunden-Fragmenten und mehr als 2 1/2 Stunden Video bei Streams mit 10-Sekunden-Fragmenten.

Type: Long

Gültiger Bereich: Mindestwert 1. Der Höchstwert ist 5000.

Erforderlich: Nein

PlaybackMode

Ob Live-, Live-Wiedergabe- oder archivierte On-Demand-Daten abgerufen werden sollen.

Zu den Funktionen der drei Sitzungstypen gehören:

- **`LIVE`**: Für Sitzungen dieses Typs wird das MPEG-DASH-Manifest kontinuierlich mit den neuesten Fragmenten aktualisiert, sobald diese verfügbar sind. Wir empfehlen, dass der Media Player alle eine Sekunde ein neues Manifest abruft. Wenn diese Art von Sitzung in einem Media Player wiedergegeben wird, zeigt die Benutzeroberfläche in der Regel eine Live-Benachrichtigung an, ohne dass die gewünschte Position im Wiedergabefenster ausgewählt werden kann.

  > **Note**
  >
  > Im `LIVE` Modus sind die neuesten verfügbaren Fragmente in einem MPEG-DASH-Manifest enthalten, auch wenn zwischen den Fragmenten eine Lücke besteht (d. h. wenn ein Fragment fehlt). Eine solche Lücke kann dazu führen, dass ein Media Player stoppt oder die Wiedergabe unterbrochen wird. In diesem Modus werden Fragmente nicht zum MPEG-DASH-Manifest hinzugefügt, wenn sie älter als das neueste Fragment in der Playlist sind. Wenn das fehlende Fragment verfügbar wird, nachdem ein weiteres Fragment zum Manifest hinzugefügt wurde, wird das ältere Fragment nicht hinzugefügt und die Lücke wird nicht gefüllt.

- **`LIVE_REPLAY`**: Bei Sessions dieses Typs wird das MPEG-DASH-Manifest ähnlich aktualisiert, wie es für den `LIVE` Modus aktualisiert wird, außer dass es zunächst Fragmente aus einer bestimmten Startzeit einbezieht. Anstatt Fragmente hinzuzufügen, wenn sie aufgenommen werden, werden Fragmente hinzugefügt, wenn die Dauer des nächsten Fragments verstrichen 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 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. Dieser Modus ist auch nützlich, um zuvor archivierte Medien zu streamen, ohne durch die Obergrenze von 1.000 Fragmenten in diesem `ON_DEMAND` Modus eingeschränkt zu sein.
- **`ON_DEMAND`**: Für Sitzungen dieses Typs enthält das MPEG-DASH-Manifest alle Fragmente für die Sitzung, bis zu der Anzahl, die in angegeben ist. `MaxManifestFragmentResults` Das Manifest muss für jede Sitzung nur einmal abgerufen werden. Wenn diese Art von Sitzung in einem Media Player wiedergegeben wird, zeigt die Benutzeroberfläche normalerweise ein Steuerelement an, mit dem Sie die anzuzeigende Position im Wiedergabefenster auswählen können.

Wenn dies der Fall `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und mehrere Fragmente mit demselben Startzeitstempel vorhanden sind, ist in allen Wiedergabemodi das Fragment mit der größeren Fragmentnummer (d. h. das neuere Fragment) im MPEG-DASH-Manifest enthalten. Die anderen Fragmente sind nicht enthalten. Fragmente mit unterschiedlichen Zeitstempeln, aber überlappender Dauer 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 den oder den `StreamName` angeben. `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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, für den die MPEG-DASH-Manifest-URL abgerufen werden soll.

Sie müssen entweder den oder den `StreamName` angeben. `StreamARN`

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 (die das Sitzungstoken enthält), die ein Media Player verwenden kann, um das MPEG-DASH-Manifest abzurufen.

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 Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einem der Tracks des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einem 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 Wert `DataRetentionInHours` von 0 hat).

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

UnsupportedStreamMediaTypeException

Der Medientyp (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte anhand IDs des Codec der Titel im ersten Fragment für eine Wiedergabesitzung nicht bestimmt werden. Die Codec-ID für Track 1 sollte `V_MPEG/ISO/AVC` und optional die Codec-ID für Track 2 sein. `A_AAC`

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetHLSStreamingSessionURL

Bedienung: Amazon Kinesis Video Streams Archived Media

Ruft eine HTTP Live Streaming (HLS) -URL für den Stream ab. Sie können die URL dann in einem Browser oder Media Player öffnen, um den Stream-Inhalt anzusehen.

`StreamName`Sowohl der als auch der `StreamARN` Parameter sind optional, aber Sie müssen entweder den `StreamName` oder den angeben, `StreamARN` wenn Sie diesen API-Vorgang aufrufen.

Für einen Amazon Kinesis Kinesis-Videostream gelten die folgenden Anforderungen für die Bereitstellung von Daten über HLS:

- Anforderungen an den Titel für die Videowiedergabe.
- 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 (AAC-Spezifikation ISO/IEC 13818-7) enthalten.

Kinesis Video Streams HLS-Sitzungen enthalten Fragmente in der fragmentierten MPEG-4-Form (auch als f MP4 oder CMAF bezeichnet) oder der MPEG-2-Form (auch TS-Chunks genannt, die von der HLS-Spezifikation ebenfalls unterstützt werden). Weitere Informationen zu HLS-Fragmenttypen finden Sie in der HLS-Spezifikation.

Das folgende Verfahren zeigt, wie HLS mit Kinesis Video Streams verwendet wird:

1. Rufen Sie die `GetDataEndpoint` API auf, um einen Endpunkt zu erhalten. Senden Sie dann die `GetHLSStreamingSessionURL` Anfragen mit dem Parameter --endpoint-url an diesen Endpunkt.
2. Rufen Sie die HLS-URL ab mit. `GetHLSStreamingSessionURL` 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-Playlist der Sitzung zurück (die Root-Ressource, die für das Streaming mit HLS benötigt wird).

> **Note**
>
> Teilen oder speichern Sie dieses Token nicht, wenn 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.

Die Medien, die über die Playlist verfügbar gemacht werden, bestehen nur aus dem angeforderten Stream, Zeitraum und Format. Es werden keine anderen Mediendaten (wie Frames außerhalb des angeforderten Fensters oder alternative Bitraten) zur Verfügung gestellt.

3. Geben Sie die URL (mit dem verschlüsselten Sitzungstoken) für die HLS-Master-Playlist an einen Media Player weiter, der das HLS-Protokoll unterstützt. Kinesis Video Streams stellt die HLS-Medien-Playlist, das Initialisierungsfragment und die Medienfragmente über die URL der Master-Playlist zur Verfügung. Das Initialisierungsfragment enthält die privaten Codec-Daten für den Stream und andere Daten, die zur Einrichtung 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 Stream-Metadaten und Mediendaten normal an. Wenn der Media Player Daten anfordert, ruft er die folgenden Aktionen auf:
   - HLSMasterPlaylist abrufen: Ruft eine HLS-Master-Playlist ab, die eine URL für die `GetHLSMediaPlaylist` Aktion für jeden Titel und zusätzliche Metadaten für den Media Player, einschließlich der geschätzten Bitrate und Auflösung, enthält.
   - HLSMediaPlaylist abrufen: Ruft eine HLS-Medien-Playlist ab, die eine URL für den Zugriff auf das MP4 Initialisierungsfragment mit der `GetMP4InitFragment` Aktion und URLs für den Zugriff auf die Medienfragmente mit den MP4 Aktionen enthält. `GetMP4MediaFragment` Die HLS-Medien-Playlist enthält auch Metadaten über den Stream, den der Player zum Abspielen benötigt, z. B. ob der Stream ist oder. `PlaybackMode` `LIVE` `ON_DEMAND` Die HLS-Medien-Playlist ist in der Regel statisch für Sessions mit einem `PlaybackType` von. `ON_DEMAND` Die HLS-Medien-Playlist wird kontinuierlich mit neuen Fragmenten für Sitzungen mit einem `PlaybackType` von aktualisiert. `LIVE` Es gibt eine eigene HLS-Medien-Playlist für den Videotrack und den Audiotrack (falls zutreffend), die MP4 Medien URLs für den jeweiligen Titel enthält.

- Abrufen MP4InitFragment: Ruft das MP4 Initialisierungsfragment ab. Der Media Player lädt normalerweise das Initialisierungsfragment, bevor er Medienfragmente lädt. Dieses Fragment enthält die Atome `fytp` "" und "`moov`" sowie die untergeordneten MP4 Atome, die zur Initialisierung des Media Player-Decoders benötigt werden.

  Das Initialisierungsfragment entspricht keinem Fragment in einem Kinesis-Videostream. Es enthält nur die privaten Codec-Daten für den Stream und den jeweiligen Titel, die der Media Player benötigt, um die Medienframes zu dekodieren.

- Abrufen MP4MediaFragment: Ruft MP4 Medienfragmente ab. Diese Fragmente enthalten die MP4 Atome `moof` "" und "`mdat`" und ihre untergeordneten Atome, die die Medienframes und ihre Zeitstempel des codierten Fragments enthalten.

  > **Note**
  >
  > Die in jedem Fragment enthaltenen privaten Codec Data (CPD) enthalten Codec-spezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Dekodierung des Fragments erforderlich sind. Sowohl für TS- als auch für CPD-Änderungen werden während einer MP4 Streaming-Sitzung unterstützt. Daher können die Fragmente in einer Sitzung unterschiedliche Informationen in der CPD enthalten, ohne dass die Wiedergabe unterbrochen wird. Für jede Streaming-Sitzung sind nur 500 CPD-Änderungen zulässig.

  > **Important**
  >
  > Änderungen nachverfolgen werden nicht unterstützt. Die Titel müssen in den abgefragten Medien einheitlich sein. Das Streaming schlägt fehl, wenn die Fragmente im Stream nicht mehr nur Video enthalten, sondern sowohl Audio als auch Video enthalten, oder wenn eine AAC-Audiospur in eine A-Law-Audiospur umgewandelt wird.

  Mit dieser Aktion abgerufene Daten sind abrechnungsfähig. Weitere Informationen finden Sie unter -Preise.

- AbrufenTSFragment: Ruft MPEG TS-Fragmente ab, die sowohl Initialisierungs- als auch Mediendaten für alle Titel im Stream enthalten.

> **Note**
>
> Falls ja`MPEG_TS`, `ContainerFormat` wird diese API anstelle von `GetMP4InitFragment` und zum Abrufen von `GetMP4MediaFragment` Stream-Medien verwendet.

Mit dieser Aktion abgerufene Daten sind fakturierbar. Weitere Informationen finden Sie unter Preise für Kinesis Video Streams.

Die URL einer Streaming-Sitzung darf nicht von Spielern gemeinsam genutzt werden. Der Dienst drosselt möglicherweise eine Sitzung, wenn sie von mehreren Mediaplayern gemeinsam genutzt wird. Verbindungslimits finden Sie unter Kinesis Video Streams Streams-Kontingente.

Sie können die Datenmenge, die der Media Player verbraucht, überwachen, indem Sie die `GetMP4MediaFragment.OutgoingBytes` CloudWatch Amazon-Metrik überwachen. Informationen CloudWatch zur Überwachung von Kinesis Video Streams finden Sie unter Kinesis Video Streams überwachen. Preisinformationen finden Sie unter Amazon Kinesis Video Streams — Preise und AWS Preise. Sowohl für HLS-Sitzungen als auch für ausgehende AWS Daten fallen Gebühren an.

Die Beispiele für die Videowiedergabe finden Sie in der Dokumentation: Verwenden Sie die AWS CLI , um die URL einer HLS-Streaming-Sitzung abzurufen undBeispiel: Verwenden Sie HLS in HTML und JavaScript.

Weitere Informationen zu HLS finden Sie unter HTTP Live Streaming auf der Apple Developer-Website.

> **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 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 /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 für die Verpackung der Medien verwendet werden soll. Durch Angabe des `FRAGMENTED_MP4` Containerformats werden die Medien in MP4 Fragmente (f MP4 oder CMAF) verpackt. Dies ist die empfohlene Verpackung, da nur ein minimaler Verpackungsaufwand anfällt. Die andere Option für das Containerformat ist`MPEG_TS`. HLS unterstützt seit seiner Veröffentlichung MPEG TS-Chunks und ist manchmal das einzige unterstützte Paket auf älteren HLS-Playern. MPEG TS hat in der Regel einen Paketierungsaufwand von 5 bis 25 Prozent. Das bedeutet, dass MPEG TS in der Regel 5 bis 25 Prozent mehr Bandbreite und Kosten benötigt als f. MP4

Der Standardwert ist `FRAGMENTED_MP4`.

Typ: Zeichenfolge

Zulässige Werte: `FRAGMENTED_MP4 | MPEG_TS`

Erforderlich: Nein

DiscontinuityMode

Gibt an, wann Markierungen, die Unterbrechungen zwischen Fragmenten kennzeichnen, zu den Medien-Playlisten hinzugefügt werden.

Media-Player erstellen in der Regel eine Zeitleiste mit den abzuspielenden Medieninhalten, die auf den Zeitstempeln der einzelnen Fragmente basiert. Das heißt, wenn es Überschneidungen oder Lücken zwischen Fragmenten gibt (was bei Einstellung üblich HLSFragmentSelector ist`SERVER_TIMESTAMP`), weist die Timeline des Media Players an einigen Stellen auch kleine Lücken zwischen den Fragmenten auf und überschreibt an anderen Stellen Frames. Lücken in der Timeline des Media Players können dazu führen, dass die Wiedergabe ins Stocken gerät, und Überlappungen können dazu führen, dass die Wiedergabe ruckelig wird. Wenn zwischen Fragmenten Unterbrechungen auftreten, wird erwartet, dass der Media Player die Timeline zurücksetzt, sodass das nächste Fragment unmittelbar nach dem vorherigen Fragment abgespielt wird.

Die folgenden Modi werden unterstützt:

- `ALWAYS`: Zwischen jedem Fragment in der HLS-Medien-Playlist wird eine Diskontinuitätsmarkierung platziert. Es wird empfohlen, den Wert von zu verwenden, `ALWAYS` wenn die Zeitstempel der Fragmente nicht korrekt sind.

- NEVER: Es sind nirgends Diskontinuitätsmarkierungen platziert. Es wird empfohlen, den Wert von zu verwenden, NEVER um sicherzustellen, dass die Timeline des Media Players möglichst genau den Zeitstempeln des Produzenten entspricht.
- ON_DISCONTINUITY: Zwischen Fragmenten, die eine Lücke oder Überlappung von mehr als 50 Millisekunden aufweisen, wird ein Diskontinuitätsmarker platziert. Für die meisten Wiedergabeszenarien wird empfohlen, den Wert von zu verwenden, ON_DISCONTINUITY sodass die Media Player-Zeitleiste nur zurückgesetzt wird, wenn ein erhebliches Problem mit der Medien-Timeline vorliegt (z. B. ein fehlendes Fragment).

Die Standardeinstellung istSERVER_TIMESTAMP, ALWAYS wann auf eingestellt HLSFragmentSelector ist und NEVER wann es auf eingestellt istPRODUCER_TIMESTAMP.

Typ: Zeichenfolge

Zulässige Werte: ALWAYS | NEVER | ON_DISCONTINUITY

Erforderlich: Nein

DisplayFragmentTimestamp

Gibt an, wann die Startzeitstempel des Fragments in die HLS-Medien-Playlist aufgenommen werden sollen. In der Regel geben Media Player die Position des Abspielkopfs als Zeit relativ zum Beginn des ersten Fragments in der Wiedergabesitzung an. Wenn die Startzeitstempel jedoch in der HLS-Medien-Playlist enthalten sind, geben einige Media Player den aktuellen Abspielkopf möglicherweise als absolute Zeit an, die auf den Zeitstempeln des Fragments basiert. Dies kann nützlich sein, um ein Wiedergabeerlebnis zu schaffen, bei dem den Zuschauern die Uhrzeit des Mediums angezeigt wird, zu der die Wandzeit angezeigt wird.

Der Standardwert ist NEVER. Wenn dies der HLSFragmentSelector Fall istSERVER_TIMESTAMP, sind die Zeitstempel die Startzeitstempel des Servers. In ähnlicher Weise sind die Zeitstempel die PRODUCER_TIMESTAMP Startzeitstempel des Produzenten, wenn dies der Fall HLSFragmentSelector ist.

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 `GetHLSMasterPlaylist` Sitzung keine neuen Aufrufe von `GetHLSMediaPlaylist` `GetMP4InitFragmentGetMP4MediaFragment`,,, oder getätigt werden.

Der Standardwert ist 300 (5 Minuten).

Typ: Ganzzahl

Gültiger Bereich: Mindestwert von 300. Maximalwert 43200.

Erforderlich: Nein

HLSFragmentSelector

Der Zeitbereich des angeforderten Fragments und die Quelle der Zeitstempel.

Dieser Parameter ist erforderlich, wenn er `ON_DEMAND` oder `PlaybackMode` `LIVE_REPLAY` ist. Dieser Parameter ist optional, falls PlaybackMode ja `LIVE`. Wenn `PlaybackMode` ja`LIVE`, `FragmentSelectorType` kann er gesetzt werden, `TimestampRange` sollte aber nicht gesetzt werden. Wenn `PlaybackMode` ist `ON_DEMAND` oder`LIVE_REPLAY`, `TimestampRange` müssen beide `FragmentSelectorType` gesetzt werden.

Typ: HLSFragmentSelector Objekt

Erforderlich: Nein

MaxMediaPlaylistFragmentResults

Die maximale Anzahl von Fragmenten, die in den HLS-Medien-Playlisten zurückgegeben werden.

Wenn ja `PlaybackModeLIVE`, werden die neuesten Fragmente bis zu diesem Wert zurückgegeben. Wenn der `PlaybackMode` Wert ist`ON_DEMAND`, werden die ältesten Fragmente zurückgegeben, bis zu dieser Höchstzahl.

Wenn in einer Live-HLS-Medien-Playlist eine höhere Anzahl von Fragmenten verfügbar ist, puffern Videoplayer häufig Inhalte, bevor sie mit der Wiedergabe beginnen. Eine Erhöhung der Puffergröße erhöht die Wiedergabe-Latenz, verringert jedoch die Wahrscheinlichkeit, dass

während der Wiedergabe eine Zwischenpufferung auftritt. Wir empfehlen, dass eine Live-HLS-Medien-Playlist mindestens 3 Fragmente und maximal 10 Fragmente enthält.

Die Standardeinstellung ist 5 Fragmente, falls `PlaybackMode` es `LIVE` oder ist`LIVE_REPLAY`, und 1.000, wenn ja`PlaybackMode`. `ON_DEMAND`

Der Höchstwert von 5.000 Fragmenten entspricht mehr als 80 Minuten Video bei Streams mit 1-Sekunden-Fragmenten und mehr als 13 Stunden Video bei Streams mit 10-Sekunden-Fragmenten.

Type: Long

Gültiger Bereich: Mindestwert 1. Der Höchstwert ist 5000.

Erforderlich: Nein

PlaybackMode

Ob Live-, Live-Wiedergabe- oder archivierte On-Demand-Daten abgerufen werden sollen.

Zu den Funktionen der drei Sitzungstypen gehören:

- **`LIVE`**: Bei Sitzungen dieses Typs wird die HLS-Medien-Playlist ständig mit den neuesten Fragmenten aktualisiert, sobald sie verfügbar sind. Wir empfehlen, dass der Media Player alle eine Sekunde lang eine neue Playlist abruft. Wenn diese Art von Sitzung in einem Media Player abgespielt wird, zeigt die Benutzeroberfläche in der Regel eine Live-Benachrichtigung an, ohne dass die gewünschte Position im Wiedergabefenster ausgewählt werden kann.

  > **Note**
  >
  > Im `LIVE` Modus werden die neuesten verfügbaren Fragmente in eine HLS-Medien-Playlist aufgenommen, auch wenn zwischen den Fragmenten eine Lücke besteht (d. h. wenn ein Fragment fehlt). Eine solche Lücke kann dazu führen, dass ein Media Player angehalten wird oder die Wiedergabe unterbrochen wird. In diesem Modus werden Fragmente nicht zur HLS-Medien-Playlist hinzugefügt, wenn sie älter als das neueste Fragment in der Playlist sind. Wenn das fehlende Fragment verfügbar wird, nachdem ein weiteres Fragment zur Playlist hinzugefügt wurde, wird das ältere Fragment nicht hinzugefügt und die Lücke wird nicht gefüllt.

- **`LIVE_REPLAY`**: Bei Sessions dieses Typs wird die HLS-Medien-Playlist ähnlich wie bei der Aktualisierung für den `LIVE` Modus aktualisiert, außer dass sie damit beginnt, Fragmente

ab einer bestimmten Startzeit einzubeziehen. Fragmente werden nicht bei der Aufnahme hinzugefügt, sondern Fragmente werden hinzugefügt, wenn die Dauer des nächsten Fragments verstrichen ist. Wenn die Fragmente in der Sitzung beispielsweise zwei Sekunden lang sind, wird der Medien-Playlist alle zwei Sekunden ein neues Fragment hinzugefügt. 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. Dieser Modus ist auch nützlich, um zuvor archivierte Medien zu streamen, ohne durch die Obergrenze von 1.000 Fragmenten in diesem `ON_DEMAND` Modus eingeschränkt zu sein.

- **`ON_DEMAND`**: Bei Sitzungen dieses Typs enthält die HLS-Medien-Playlist alle Fragmente für die Sitzung, bis zu der Anzahl, die in `MaxMediaPlaylistFragmentResults` angegeben ist. Die Playlist 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 Steuerelement an, mit dem Sie die anzuzeigende Position im Wiedergabefenster auswählen können.

Wenn dies der Fall `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und mehrere Fragmente mit demselben Startzeitstempel vorhanden sind, ist in allen Wiedergabemodi das Fragment mit der größten Fragmentnummer (d. h. das neueste Fragment) in der HLS-Medien-Playlist enthalten. Die anderen Fragmente sind nicht enthalten. Fragmente mit unterschiedlichen Zeitstempeln, aber überlappender Dauer sind weiterhin in der HLS-Medien-Playlist 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-Playlist 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 die URL der HLS-Master-Playlist 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: 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 (die das Sitzungstoken enthält), mit der ein Media Player die HLS-Master-Playlist abrufen 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 Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

InvalidCodecPrivateDataException

Die privaten Codec-Daten in mindestens einem der Tracks des Videostreams sind für diesen Vorgang nicht gültig.

HTTP Status Code: 400

MissingCodecPrivateDataException

In mindestens einem 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 Wert `DataRetentionInHours` von 0 hat).

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

UnsupportedStreamMediaTypeException

Der Medientyp (z. B. h.264- oder h.265-Video oder AAC- oder G.711-Audio) konnte anhand IDs des Codec der Titel im ersten Fragment für eine Wiedergabesitzung nicht bestimmt werden. Die Codec-ID für Track 1 sollte `V_MPEG/ISO/AVC` und optional die Codec-ID für Track 2 sein. `A_AAC`

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetImages

Bedienung: Amazon Kinesis Video Streams Archived Media

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.

GetImages ruft außerdem eine Liste von Bildern ab, die jedem Zeitstempel für einen bestimmten Zeitraum, ein bestimmtes Abtastintervall und eine bestimmte Bildformatkonfiguration entsprechen.

> **Note**
>
> Sie müssen zuerst die `GetDataEndpoint` API aufrufen, um einen Endpunkt zu erhalten. Senden Sie dann die `GetImages` Anfragen mit dem Parameter --endpoint-url an diesen Endpunkt.

Anforderungen an den Titel für die Videowiedergabe.

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. Liegt der Zeitbereich zwischen `StartTimestamp` und `EndTimestamp` mehr als 300 Sekunden darüber`StartTimestamp`, erhalten Sie eine`IllegalArgumentException`.

Typ: Zeitstempel

Erforderlich: Ja

Format

Das Format, das zum Codieren des Bilds verwendet wird.

Typ: Zeichenfolge

Zulässige Werte: `JPEG | PNG`

Erforderlich: Ja

FormatConfig

Die Liste einer Struktur mit Schlüssel-Wert-Paaren, die zusätzliche Parameter enthält, die bei der Generierung des Bildes angewendet werden können. Der `FormatConfig` Schlüssel ist der`JPEGQuality`, der den JPEG-Qualitätsschlüssel angibt, der zur Generierung des Bildes verwendet werden soll. Der `FormatConfig` Wert 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 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`

Längenbeschränkungen für Werte: 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 `HeightPixels` sowohl als auch `WidthPixels` Parameter angegeben sind, wird das Bild gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `HeightPixels` Parameter angegeben wird, wird das ursprüngliche Seitenverhältnis zur Berechnung des `WidthPixels` Verhältnisses verwendet. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Maximalwert von 2160.

Erforderlich: Nein

ImageSelectorType

Der Ursprung der Server- oder Producer-Zeitstempel, die zur Generierung der Bilder verwendet werden sollen.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Erforderlich: Ja

MaxResults

Die maximale Anzahl von Bildern, die von der API zurückgegeben werden sollen.

> **Note**
>
> Das Standardlimit liegt bei 25 Bildern pro API-Antwort. Wenn Sie einen `MaxResults` Wert angeben, der über diesem Wert liegt, ergibt sich eine Seitengröße von 25. Alle weiteren Ergebnisse werden paginiert.

Type: Long

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

Erforderlich: Nein

NextToken

Ein Token, das angibt, wo mit der Paginierung der nächsten Gruppe von Bildern begonnen werden soll. Dies ist das Ergebnis `GetImages:NextToken` 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 Bilder aus dem Stream generiert werden müssen. Der Mindestwert, der bereitgestellt werden kann, ist 200 ms (5 Bilder pro Sekunde). Wenn der Zeitstempelbereich unter dem Abtastintervall liegt, `startTimestamp` wird das Bild von zurückgegeben, sofern verfügbar.

Typ: Ganzzahl

Erforderlich: Nein

StartTimestamp

Der Startpunkt, von dem aus die Bilder generiert werden sollen. Damit ein Bild zurückgegeben werden kann, `StartTimestamp` muss dieser Wert innerhalb eines gültigen Zeitstempelbereichs liegen.

Typ: Zeitstempel

Erforderlich: Ja

StreamARN

Der Amazon-Ressourcenname (ARN) des Streams, aus dem die Bilder abgerufen werden sollen. Sie müssen entweder den `StreamName` oder den angeben`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]+`

Erforderlich: Nein

StreamName

Der Name des Streams, aus dem die Bilder abgerufen werden sollen. Sie müssen entweder den `StreamName` oder den angeben`StreamARN`.

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 `WidthPixels` sowohl als auch `HeightPixels` Parameter angegeben sind, wird das Bild gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `WidthPixels` Parameter angegeben wird oder wenn nur der angegeben `HeightPixels` wird, `ValidationException` wird a ausgelöst. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße aus dem Stream zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Der Maximalwert ist 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 Bildes ein Fehler auftritt, `MEDIA_ERROR` wird dieser in der Ausgabe als Ursache für das fehlende Bild aufgeführt.

Typ: Array von Image-Objekten

NextToken

Das verschlüsselte Token, das in der Anfrage zum Abrufen weiterer Bilder verwendet wurde.

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 Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

HTTP Status Code: 400

InvalidArgumentException

Ein bestimmter Parameter überschreitet seine Beschränkungen, wird nicht unterstützt oder kann nicht verwendet werden.

HTTP Status Code: 400

NoDataRetentionException

`GetImages`wurde für einen Stream angefordert, der keine Daten speichert (d. h. den Wert `DataRetentionInHours` 0 hat).

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetMediaForFragmentList

Bedienung: 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 Kinesis Video Streams Streams-Kontingente.

> **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: Zeichenfolgen-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 1000 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. 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 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 angeforderten Medien.

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

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 - Die Fragmentnummer wurde im Chunk zurückgegeben.
- AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP - Serverseitiger Zeitstempel des Fragments.
- AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP - Herstellerseitiger Zeitstempel des Fragments.

Die folgenden Tags werden hinzugefügt, falls 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 Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListFragments

Bedienung: 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 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 /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 beträgt 1 Zeichen. 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 beträgt 1 Zeichen. 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 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 Anfrage gedrosselt, weil Sie ein Limit überschritten haben. Versuchen Sie später, den Anruf zu tätigen. Informationen zu Grenzwerten finden Sie unter Kinesis Video Streams Streams-Kontingente.

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Amazon Kinesis Kinesis-Videosignalkanäle

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 URIs, einschließlich 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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Service

Gibt den gewünschten Dienst an. Derzeit `TURN` ist dies der einzig gültige Wert.

Typ: Zeichenfolge

Zulässige Werte: `TURN`

Erforderlich: Nein

Username

Eine optionale Benutzer-ID, die den Anmeldeinformationen zugeordnet werden soll.

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

{
   "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

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for 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-alexa-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 beträgt 1 Zeichen. 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 beträgt 1 Zeichen. 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 beträgt 1 Zeichen. Die maximale Länge beträgt 10000.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Amazon Kinesis Video WebRTC-Speicher

Die folgenden Aktionen werden von Amazon Kinesis Video WebRTC Storage unterstützt:

- JoinStorageSession
- JoinStorageSessionAsViewer

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 Einwegvideo und and/or Mehrwege-Audio als Videoproduktionsgerät für einen Eingangskanal teil. Wenn es keine bestehende Sitzung für den Kanal gibt, erstellen Sie eine neue Streaming-Sitzung und geben Sie den Amazon-Ressourcennamen (ARN) des Signalisierungskanals an.

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. Aktuelle Anforderungen:
>
> - Videospur: H.264
> - Audiospur: Opus

Das resultierende aufgenommene Video im Kinesis-Videostream hat die folgenden Parameter: H.264-Video und AAC-Audio.

Sobald ein Master-Teilnehmer eine Verbindung über WebRTC ausgehandelt hat, wird die aufgenommene Mediensitzung im Kinesis-Videostream gespeichert. Über unser Playback können dann mehrere Zuschauer Medien in Echtzeit wiedergeben. APIs

Sie können auch vorhandene Kinesis Video Streams Streams-Funktionen wie `HLS` `DASH` OD-Wiedergabe, Bilderzeugung über GetImagesund mehr mit aufgenommenen WebRTC-Medien verwenden.

> **Note**
>
> Die Bereitstellung von S3-Bildern und Benachrichtigungen werden derzeit nicht unterstützt.

> **Note**
>
> Gehen Sie davon aus, dass einer Sitzung für den Kanal nur ein Client für ein Videoproduktionsgerät zugeordnet werden kann. Wenn mehr als ein Client der Sitzung eines bestimmten Kanals als Videoproduktionsgerät beitritt, 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 Anruf 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.

> **Important**
>
> Beachten Sie die Großschreibung dieses Eingabeparameters.

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

JoinStorageSessionAsViewer

Service: Amazon Kinesis Video WebRTC Storage

> **Note**
>
> WebRTC-Ingest mit Multi-Viewer-Unterstützung (Preview) wird in der Vorschauversion gemäß den AWS Servicebedingungen bereitgestellt und kann sich ändern. Es ist derzeit nur in us-east-1 (IAD) verfügbar.
> Um an der Vorschau teilzunehmen, senden Sie uns eine E-Mail an -preview@amazon.com. kvs-webrtc-multi-view

> **Note**
>
> Bevor Sie diese API verwenden können, müssen Sie die `GetSignalingChannelEndpoint` API aufrufen, um den WEBRTC-Endpunkt anzufordern. Anschließend geben Sie den Endpunkt und die Region in Ihrer `JoinStorageSessionAsViewer` API-Anfrage an.

`JoinStorageSessionAsViewer`ermöglicht es einem Zuschauer, an einer laufenden, in der Cloud aufgezeichneten WebRTC-Streaming-Sitzung teilzunehmen. Diese API initiiert eine WebRTC-Verbindung, indem sie ein SDP-Angebot und ICE-Kandidaten zwischen dem Zuschauer und dem Aufzeichnungsagenten sendet, sodass der Zuschauer Echtzeitvideos vom Master über den Aufzeichnungsagenten empfangen und an der bidirektionalen Audiokommunikation über den Aufzeichnungsagenten teilnehmen kann. Sobald die Verbindung hergestellt ist, wird das Audio des Zuschauers, sofern bereitgestellt, an alle anderen verbundenen Peers, einschließlich des Master-Teilnehmers, weitergeleitet und in den WebRTC-Stream integriert, der im Kinesis Video-Stream gespeichert wird.

> **Important**
>
> Zuschauer können derzeit keine Videotracks senden. Zuschauer können eine optionale Audiospur oder gar keine Titel senden.
> Aktuelle Anforderungen für das Senden von Zuschauern und Teilnehmern:
>
> - Videotrack: Nicht unterstützt
> - Audiospur (optional): Opus

Wenn der Master-Teilnehmer derzeit mit dem Videoproduktionsgerät verbunden ist, hat das resultierende aufgenommene Video im Kinesis-Videostream die folgenden Parameter: H.264-Video und AAC-Audio.

> **Note**
>
> Die Viewer-Teilnehmer stellen direkt eine Verbindung zur Speichersitzung her und nicht direkt mit dem Master-Teilnehmer. Die Speichersitzung kümmert sich um das Mischen, Duplizieren und Weiterleiten der Medien an das entsprechende Ziel.

> **Note**
>
> Wenn der Master-Teilnehmer nicht anwesend ist, können sich die Zuschauer nicht gegenseitig hören.

Sobald der Zuschauer-Teilnehmer eine Verbindung über WebRTC ausgehandelt hat, wird die aufgenommene Audiositzung im Kinesis-Videostream gespeichert, solange der Master-Teilnehmer auch mit der Speichersitzung verbunden ist.

Sie können auch vorhandene Kinesis Video Streams Streams-Funktionen wie HLS DASH OD-Wiedergabe, Bilderzeugung über GetImagesund mehr mit aufgenommenen WebRTC-Medien verwenden.

> **Note**
>
> Die Bereitstellung von S3-Bildern und Benachrichtigungen werden derzeit nicht unterstützt.

> **Note**
>
> Gehen Sie davon aus, dass einer Sitzung für den Kanal nur ein Client für ein Videoproduktionsgerät zugeordnet werden kann. Wenn mehr als ein Client der Sitzung eines bestimmten Kanals als Videoproduktionsgerät beitritt, hat die neueste Client-Anfrage Vorrang.

Beschränkungen

Die aktuellen Grenzwerte gelten:

- Maximale Anzahl von Zuschauern: 3
- Maximale Zeit, für die Zuschauerteilnehmer mit der Speichersitzung verbunden bleiben, ohne dass ein Master-Teilnehmer anwesend ist: 3 Minuten

> **Important**
>
> Wenn ein Zuschauer die Verbindung zur Speichersitzung trennt (die Peer-Verbindung schließt), bleibt sein Kontingent (Zuschauerlimit) 1 Minute lang verbraucht. Während dieses Zeitraums von einer Minute kann der Zuschauer diese API mit derselben Client-ID aufrufen, um wieder an der Sitzung teilzunehmen, ohne ein zusätzliches Zuschauerkontingent zu verbrauchen. Nach 1 Minute wird das Zuschauerkontingent freigegeben und steht anderen Zuschauern zur Teilnahme zur Verfügung.

Zusätzliche Informationen

- Idempotent — Diese API ist nicht idempotent.
- Wiederholungsverhalten — Dies wird als neuer API-Aufruf gezählt.
- Gleichzeitige Aufrufe — Gleichzeitige Aufrufe sind zulässig. Pro Anruf wird einmal ein Angebot gesendet.

Anforderungssyntax

```
POST /joinStorageSessionAsViewer HTTP/1.1
Content-type: application/json

{
   "channelArn": "string",
   "clientId": "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.

> **Important**
>
> Beachten Sie die Großschreibung dieses Eingabeparameters.

Typ: Zeichenfolge

Pattern: `arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erforderlich: Ja

clientId

Die eindeutige Kennung für den Absender-Client.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin

- 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 Signalkanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

ChannelStatus

Aktueller Status des Signalkanals.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | UPDATING | DELETING`

Erforderlich: Nein

ChannelType

Der Typ des Signalkanals.

Typ: Zeichenfolge

Zulässige Werte: `SINGLE_MASTER | FULL_MESH`

Erforderlich: Nein

CreationTime

Der Zeitpunkt, zu dem der Signalkanal erstellt wurde.

Typ: Zeitstempel

Erforderlich: Nein

SingleMasterConfiguration

Eine Struktur, die die Konfiguration für den `SINGLE_MASTER` Kanaltyp enthält.

Typ: SingleMasterConfiguration Objekt

Erforderlich: Nein

Version

Die aktuelle Version des Signalkanals.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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 unter angegebene Bedingung erfüllen. `ChannelNameCondition`

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`

Erforderlich: Nein

ComparisonValue

Ein zu vergleichender Wert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeletionConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, die erforderlich sind, um die Verbindung des Streams vom Edge-Agent zu löschen.

Inhalt

DeleteAfterUpload

Der `boolean` Wert, der verwendet wird, um anzugeben, ob Sie die Medien zum Löschen markieren möchten, nachdem sie in die Kinesis Video Stream-Cloud hochgeladen wurden. Die Mediendateien können gelöscht werden, wenn einer der Konfigurationswerte für das Löschen auf festgelegt ist`true`, z. B. wenn das Limit für`EdgeRetentionInHours`, oder`MaxLocalMediaSizeInMB`, erreicht wurde.

Da der Standardwert auf eingestellt ist`true`, konfigurieren Sie den Uploader-Zeitplan so, dass die Mediendateien nicht gelöscht werden, bevor sie zum ersten Mal in die AWS Cloud hochgeladen wurden.

Typ: Boolesch

Erforderlich: Nein

EdgeRetentionInHours

Die Anzahl der Stunden, für die Sie die Daten im Stream auf dem Edge Agent behalten möchten. Der Standardwert für die Aufbewahrungszeit beträgt 720 Stunden, was 30 Tagen entspricht.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Der Maximalwert ist 720.

Erforderlich: Nein

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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-Agents zu ermitteln.

Inhalt

LastRecorderStatus

Der aktuelle Status des Edge-Aufzeichnungsauftrags eines Streams.

Typ: LastRecorderStatus Objekt

Erforderlich: Nein

LastUploaderStatus

Der aktuelle 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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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)“ -Arn 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 zur Konfiguration der lokalen Größe (`LocalSizeConfig`), die für den Löschvorgang verwendet werden.

Typ: DeletionConfig Objekt

Erforderlich: Nein

UploaderConfig

Die Uploader-Konfiguration enthält die `ScheduleExpression` Details, die verwendet werden, um Upload-Jobs für die aufgezeichneten Mediendateien vom Edge Agent auf einen Kinesis Video Stream zu planen.

Typ: UploaderConfig Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImageGenerationConfiguration

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die Informationen enthält, die für die Lieferung von KVS-Images erforderlich sind. 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 bereitgestellt werden kann, ist 200 ms. Wenn der Zeitstempelbereich unter dem Abtastintervall liegt, `StartTimestamp` wird das Bild von zurückgegeben, sofern verfügbar.

Typ: Ganzzahl

Erforderlich: Ja

Status

Gibt an, ob die `ContinuousImageGenerationConfigurations` API aktiviert oder deaktiviert ist.

Typ: Zeichenfolge

Zulässige Werte: `ENABLED | DISABLED`

Erforderlich: Ja

FormatConfig

Die Liste einer Struktur mit Schlüssel-Wert-Paaren, 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 zur Generierung des Bildes verwendet werden soll. Der `FormatConfig` Wert 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 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`

Längenbeschränkungen für Werte: 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 `HeightPixels` sowohl als auch `WidthPixels` Parameter angegeben sind, wird das Bild gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `HeightPixels` Parameter angegeben wird, wird das ursprüngliche Seitenverhältnis zur

Berechnung des `WidthPixels` Verhältnisses verwendet. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Der Maximalwert ist 2160.

Erforderlich: Nein

WidthPixels

Die Breite des Ausgabebilds, das in Verbindung mit dem `HeightPixels` Parameter verwendet wird. Wenn `WidthPixels` sowohl als auch `HeightPixels` Parameter angegeben sind, wird das Bild gestreckt, sodass es dem angegebenen Seitenverhältnis entspricht. Wenn nur der `WidthPixels` Parameter angegeben wird, wird das ursprüngliche Seitenverhältnis zur Berechnung des `HeightPixels` Verhältnisses verwendet. Wenn keiner der Parameter angegeben wird, wird die ursprüngliche Bildgröße zurückgegeben.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 1. Der Maximalwert ist 3840.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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: Mindestlänge von 9. Maximale Länge von 14.

Pattern: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Erforderlich: Ja

Uri

Der Uniform Resource Identifier (URI), der angibt, wohin die Bilder geliefert werden.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LastRecorderStatus

Bedienung: Amazon Kinesis Video Streams

Der aktuelle Status des Edge-Aufzeichnungsjobs eines Streams.

Inhalt

JobStatusDetails

Eine Beschreibung des letzten Status eines Rekorder-Jobs.

Typ: Zeichenfolge

Erforderlich: Nein

LastCollectedTime

Der Zeitstempel, zu dem der Rekorderjob zuletzt ausgeführt und die Medien auf der lokalen Festplatte gespeichert wurden.

Typ: Zeitstempel

Erforderlich: Nein

LastUpdatedTime

Der Zeitstempel, zu dem der Rekorderstatus zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

RecorderStatus

Der Status des letzten Rekorder-Jobs.

Typ: Zeichenfolge

Zulässige Werte: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LastUploaderStatus

Bedienung: Amazon Kinesis Video Streams

Der aktuelle Status des Edge-to-Cloud-Uploader-Jobs eines Streams.

Inhalt

JobStatusDetails

Eine Beschreibung des letzten Status eines Uploader-Jobs.

Typ: Zeichenfolge

Erforderlich: Nein

LastCollectedTime

Der Zeitstempel, zu dem der Uploader-Job zuletzt ausgeführt und die Medien in der Cloud gesammelt wurden.

Typ: Zeitstempel

Erforderlich: Nein

LastUpdatedTime

Der Zeitstempel, zu dem der Uploader-Status zuletzt aktualisiert wurde.

Typ: Zeitstempel

Erforderlich: Nein

UploaderStatus

Der Status des letzten Uploader-Jobs.

Typ: Zeichenfolge

Zulässige Werte: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListEdgeAgentConfigurationsEdgeConfig

Bedienung: Amazon Kinesis Video Streams

Eine Beschreibung der Edge-Konfiguration eines einzelnen Streams.

Inhalt

CreationTime

Der Zeitstempel, zu dem der Stream die Edge-Konfiguration zum ersten Mal erstellt hat.

Typ: Zeitstempel

Erforderlich: Nein

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

Erforderlich: Nein

FailedStatusDetails

Eine Beschreibung des generierten Fehlerstatus.

Typ: Zeichenfolge

Erforderlich: Nein

LastUpdatedTime

Der Zeitstempel, zu dem der Stream die Edge-Konfiguration zuletzt aktualisiert hat.

Typ: Zeitstempel

Erforderlich: Nein

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 beträgt 1 Zeichen. 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`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocalSizeConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, einschließlich der maximalen Größe der Medien (`MaxLocalMediaSizeInMB`), die Sie für einen Stream auf dem Edge-Agent 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 von 64. Höchstwert von 2000000.

Erforderlich: Nein

StrategyOnFullSize

Die Strategie, die ausgeführt werden soll, wenn das `MaxLocalMediaSizeInMB` Limit eines Streams erreicht wird.

Typ: Zeichenfolge

Zulässige Werte: `DELETE_OLDEST_MEDIA | DENY_NEW_MEDIA`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MappedResourceConfigurationListItem

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Eigenschaften der Medienspeicherkonfiguration kapselt oder enthält.

Inhalt

ARN

Der Amazon-Ressourcenname (ARN) der Kinesis Video Stream-Ressource, die dem Stream zugeordnet ist.

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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaSourceConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfigurationsdetails, die aus den Anmeldeinformationen bestehen, die (`MediaUriSecretArn`und`MediaUriType`) für den Zugriff auf die Mediendateien erforderlich sind, die zur Kamera gestreamt werden.

Inhalt

MediaUriSecretArn

Der AWS Secrets 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 Typ Uniform Resource Identifier (URI). 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationConfiguration

Bedienung: Amazon Kinesis Video Streams

Verwenden Sie diese API, um Amazon Simple Notification Service (Amazon SNS) - Benachrichtigungen zu konfigurieren, wenn Fragmente in einem Stream verfügbar werden. Wenn dieser Parameter Null ist, wird die Konfiguration aus dem Stream gelöscht.

Weitere Informationen finden Sie unter Benachrichtigungen in Kinesis Video Streams.

Inhalt

DestinationConfig

Die Zielinformationen, die für die Zustellung einer Benachrichtigung an einen Kunden erforderlich sind.

Typ: NotificationDestinationConfig Objekt

Erforderlich: Ja

Status

Zeigt 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationDestinationConfig

Bedienung: Amazon Kinesis Video Streams

Die Struktur, die die Informationen enthält, die für die Zustellung einer Benachrichtigung an einen Kunden erforderlich sind.

Inhalt

Uri

Der Uniform Resource Identifier (URI), der angibt, wohin die Bilder geliefert werden.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RecorderConfig

Bedienung: Amazon Kinesis Video Streams

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.

Inhalt

MediaSourceConfig

Die Konfigurationsdetails, die aus den Anmeldeinformationen bestehen, die (`MediaUriSecretArn`und`MediaUriType`) für den Zugriff auf die zur Kamera gestreamten Mediendateien erforderlich sind.

Typ: MediaSourceConfig Objekt

Erforderlich: Ja

ScheduleConfig

Die Konfiguration, die aus den `ScheduleExpression` und den `DurationInMinutes` Details besteht, die den Zeitplan für die Aufzeichnung von einer Kamera oder einer lokalen Mediendatei auf dem Edge Agent angeben. 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourceEndpointListItem

Bedienung: Amazon Kinesis Video Streams

Ein Objekt, das den Endpunkt des von der `GetSignalingChannelEndpoint` API zurückgegebenen Signalkanals beschreibt.

Der Endpunkt des Medienservers entspricht dem `WEBRTC` Protokoll.

Inhalt

Protocol

Das Protokoll des von der `GetSignalingChannelEndpoint` API zurückgegebenen Signalkanals.

Typ: Zeichenfolge

Zulässige Werte: `WSS | HTTPS | WEBRTC`

Erforderlich: Nein

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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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 `ScheduleExpression` und den `DurationInMinutes` Attributen.

Wenn das nicht in der angegeben `ScheduleConfig` ist`RecorderConfig`, wird der Edge-Agent immer in den Aufnahmemodus versetzt.

Wenn das nicht in der angegeben `ScheduleConfig` ist`UploaderConfig`, lädt der Edge-Agent in regelmäßigen Abständen (alle 1 Stunde) Daten hoch.

Inhalt

DurationInSeconds

Die Gesamtdauer der Medienaufnahme. Wenn das `ScheduleExpression` Attribut angegeben ist, sollte das `DurationInSeconds` Attribut auch angegeben werden.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 60. Maximalwert von 3600.

Erforderlich: Ja

ScheduleExpression

Der Quartz-Cron-Ausdruck, der sich um die Planung von Aufträgen für die Aufzeichnung von der Kamera oder einer lokalen Mediendatei auf dem Edge Agent kümmert. Wenn der nicht für bereitgestellt `ScheduleExpression` wird`RecorderConfig`, wird der Edge-Agent immer in den Aufnahmemodus versetzt.

Weitere Informationen zu Quartz finden Sie auf der Cron-Trigger-Tutorialseite, um die gültigen Ausdrücke und ihre Verwendung zu verstehen.

Typ: Zeichenfolge

Längenbeschränkungen: Mindestlänge von 11. Maximale Länge beträgt 100 Zeichen.

Pattern: `[^\n]{11,100}`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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: Zeichenfolgen-Array

Array-Mitglieder: Die Mindestanzahl beträgt 1 Element. Die maximale Anzahl beträgt 5 Elemente.

Zulässige Werte: `WSS | HTTPS | WEBRTC`

Erforderlich: Nein

Role

Diese Eigenschaft wird verwendet, um Messaging-Berechtigungen in diesem `SINGLE_MASTER` Signalkanal zu ermitteln. Wenn angegeben, `MASTER` gibt diese API einen Endpunkt zurück, über den ein Client Angebote von beliebigen Zuschauern auf diesem Signalkanal empfangen und Antworten an diese senden kann. 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`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

SingleMasterConfiguration

Bedienung: Amazon Kinesis Video Streams

Eine Struktur, die die Konfiguration für den `SINGLE_MASTER` Kanaltyp enthält.

Inhalt

MessageTtlSeconds

Der Zeitraum (in Sekunden), in dem ein Signalkanal nicht zugestellte Nachrichten aufbewahrt, bevor sie verworfen werden. Wird verwendet, um diesen Wert UpdateSignalingChannel zu aktualisieren.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert von 5. Maximalwert von 120.

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamInfo

Bedienung: Amazon Kinesis Video Streams

Ein Objekt, das einen Kinesis-Videostream beschreibt.

Inhalt

CreationTime

Ein Zeitstempel, der angibt, wann der Stream erstellt wurde.

Typ: Zeitstempel

Erforderlich: Nein

DataRetentionInHours

Wie lange der Stream Daten speichert, in Stunden.

Typ: Ganzzahl

Gültiger Bereich: Mindestwert 0.

Erforderlich: Nein

DeviceName

Der Name des Geräts, das dem Stream zugeordnet ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

KmsKeyId

Die ID des Schlüssels AWS Key Management Service (AWS KMS), den Kinesis Video Streams verwendet, um Daten im Stream zu verschlüsseln.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 2048 Zeichen.

Pattern: `.+`

Erforderlich: Nein

MediaType

Der `MediaType` des Stream.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Erforderlich: Nein

Status

Der Status des Streams.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | UPDATING | DELETING`

Erforderlich: Nein

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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Version

Die Version des Streams.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 64 Zeichen.

Pattern: `[a-zA-Z0-9]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamNameCondition

Bedienung: Amazon Kinesis Video Streams

Gibt die Bedingung an, die Streams erfüllen müssen, um zurückgegeben zu 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`

Erforderlich: Nein

ComparisonValue

Ein zu vergleichender Wert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

Bedienung: Amazon Kinesis Video Streams

Ein Schlüssel- und Wertepaar, das dem angegebenen Signalkanal zugeordnet ist.

Inhalt

Key

Der Schlüssel des Tags, das dem angegebenen Signalkanal zugeordnet ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Erforderlich: Ja

Value

Der Wert des Tags, das dem angegebenen Signalkanal 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UploaderConfig

Bedienung: Amazon Kinesis Video Streams

Die Konfiguration, die aus dem `ScheduleExpression` und den `DurationInMinutes` Details besteht, die den Zeitplan für die Aufzeichnung von einer Kamera oder einer lokalen Mediendatei auf dem Edge Agent spezifizieren. Wenn das nicht in der angegeben `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 Aufzeichnung von einer Kamera oder einer lokalen Mediendatei auf dem Edge Agent spezifizieren. Wenn das hier nicht angegeben `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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Streams-Medien

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, an dem die `GetMedia` API beginnen soll, Mediendaten zurückzugeben. Sie haben die folgenden Optionen, um den Startblock zu identifizieren:

- Wählen Sie den neuesten (oder ältesten) Chunk.
- Identifizieren Sie einen bestimmten Block. Sie können einen bestimmten Block identifizieren, indem Sie entweder eine Fragmentnummer oder einen Zeitstempel (Server oder Hersteller) angeben.
- Die Metadaten jedes Chunks enthalten ein Fortsetzungstoken als Matroska-Tag (MKV) (). `AWS_KINESISVIDEO_CONTINUATION_TOKEN` Wenn Ihre vorherige `GetMedia` Anfrage beendet wurde, können Sie diesen Tag-Wert in Ihrer nächsten Anfrage verwenden. `GetMedia` Die API beginnt dann, Chunks zurückzugeben, und zwar dort, wo die letzte API geendet hat.

Inhalt

StartSelectorType

Identifiziert das Fragment im Kinesis-Videostream, aus dem Sie die Daten abrufen möchten.

- JETZT — Beginnen Sie mit dem neuesten Abschnitt im Stream.
- FRÜHESTER — Beginne mit dem frühesten verfügbaren Teil im Stream.
- FRAGMENT_NUMBER — Beginnt 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 Produzenten- oder Serverzeitstempel enthält. Sie geben `StartTimestamp` den Zeitstempel an, indem Sie hinzufügen.
- CONTINUATION_TOKEN — Lesen mit dem angegebenen Fortsetzungstoken.

> **Note**
>
> Wenn Sie NOW, EARLY oder CONTINUATION_TOKEN als Wert wählen`startSelectorType`, geben Sie in der keine zusätzlichen Informationen an. `startSelector`

Typ: Zeichenfolge

Zulässige Werte: `FRAGMENT_NUMBER | SERVER_TIMESTAMP | PRODUCER_TIMESTAMP | NOW | EARLIEST | CONTINUATION_TOKEN`

Erforderlich: Ja

AfterFragmentNumber

Gibt die Fragmentnummer an, ab der die `GetMedia` API die Fragmente zurückgeben soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^[0-9]+$`

Erforderlich: Nein

ContinuationToken

Fortsetzungstoken, das Kinesis Video Streams in der vorherigen `GetMedia` Antwort zurückgegeben hat. Die `GetMedia` API beginnt dann mit dem Chunk, der durch das Fortsetzungstoken identifiziert wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. 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. `startSelectorType` Die `GetMedia` API beginnt dann mit dem Chunk, der das Fragment mit dem angegebenen Zeitstempel enthält.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Video 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 Ursprung des Zeitstempels einer Reihe von Fragmenten.

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das heißt, wenn Produzenten einen Stream von Fragmenten mit Produzenten-Zeitstempeln erzeugen, die ungefähr der tatsächlichen Uhrzeit entsprechen, enthält der Clip alle Fragmente innerhalb des angeforderten Zeitstempelbereichs. Wenn einige Fragmente innerhalb desselben Zeitbereichs und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

Inhalt

FragmentSelectorType

Der Ursprung 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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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 Wert liegen`StartTimestamp`, und er muss nach dem angegebenen `StartTimestamp` Wert liegen. Wenn `FragmentSelectorType` für die Anfrage ein Wert `SERVER_TIMESTAMP` angegeben ist, muss dieser Wert in der Vergangenheit liegen.

Dieser Wert ist inklusiv. Der `EndTimestamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimestamp` Wert beginnen und nach diesem Wert andauern, werden in die Sitzung aufgenommen.

Typ: Zeitstempel

Erforderlich: Ja

StartTimestamp

Der Startzeitstempel im Bereich der Zeitstempel, für den Fragmente zurückgegeben werden sollen.

Nur Fragmente, die genau um oder danach beginnen, `StartTimestamp` werden in die Sitzung aufgenommen. Fragmente, die davor beginnen `StartTimestamp` und danach andauern, sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` ja`SERVER_TIMESTAMP`, `StartTimestamp` muss der später als der Streamkopf sein.

Typ: Zeitstempel

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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 auf gesetzt `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und Get DASHStreaming SessionUrl: `ON_DEMAND` oder PlaybackMode ist, wird das erste Fragment`LIVE_REPLAY`, das mit einem Producer-Zeitstempel innerhalb des angegebenen FragmentSelector: aufgenommen wurde, in die TimestampRange Medien-Playlist aufgenommen. Darüber hinaus sind die Fragmente mit Producer-Zeitstempeln innerhalb des `TimestampRange` aufgenommenen Fragments unmittelbar nach dem ersten Fragment enthalten (bis zum Wert „Get DASHStreaming sessionUrl:“). MaxManifestFragmentResults

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das bedeutet, dass das MPEG-DASH-Manifest alle Fragmente innerhalb des angeforderten Zeitstempelbereichs enthält, wenn Produzenten einen Stream von Fragmenten mit Produzenten-Zeitstempeln erzeugen, die ungefähr der tatsächlichen Uhrzeit entsprechen. Wenn einige Fragmente innerhalb desselben Zeitraums und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

Wenn auf gesetzt `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und Get DASHStreaming SessionUrl: PlaybackMode ist`LIVE`, werden die Producer-Zeitstempel in den Fragmenten und für die MP4 Deduplizierung verwendet. Die zuletzt aufgenommenen Fragmente, die auf Serverzeitstempeln basieren, sind jedoch im MPEG-DASH-Manifest enthalten. Das bedeutet, dass Fragmente, die in der Vergangenheit aufgenommen wurden, zwar Producer-Zeitstempel mit aktuellen Werten haben, sie aber nicht in der HLS-Medien-Playlist enthalten sind.

Der Standardwert ist `SERVER_TIMESTAMP`.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Erforderlich: Nein

TimestampRange

Der Anfang und das Ende des Zeitstempelbereichs für die angeforderten Medien.

Dieser Wert sollte nicht vorhanden sein, wenn er vorhanden `PlaybackType` ist`LIVE`.

Typ: DASHTimestampRange Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DASHTimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Anfang und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Dieser Wert sollte nicht vorhanden sein, wenn er vorhanden `PlaybackType` ist`LIVE`.

Die Werte in `DASHimestampRange` sind inklusiv. Fragmente, die genau an oder nach der Startzeit beginnen, sind in der Sitzung enthalten. Fragmente, die vor der Startzeit beginnen und nach der Startzeit 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 Wert liegen`StartTimestamp`, und er muss nach dem angegebenen `StartTimestamp` Wert liegen.

Wenn `FragmentSelectorType` für die Anfrage ein Wert `SERVER_TIMESTAMP` angegeben ist, 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 festgelegt `EndTimestamp` ist, enthält die Sitzung weiterhin neu aufgenommene Fragmente, bis die Sitzung abläuft.

> **Note**
>
> Dieser Wert ist inklusive. Der `EndTimestamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimestamp` Wert beginnen und nach diesem Wert andauern, werden in die Sitzung aufgenommen.

Typ: Zeitstempel

Erforderlich: Nein

StartTimestamp

Der Beginn des Zeitstempelbereichs für das angeforderte Medium.

Wenn der `DASHTimestampRange` Wert angegeben ist, ist der `StartTimestamp` Wert erforderlich.

Nur Fragmente, die genau bei oder danach beginnen, `StartTimestamp` werden in die Sitzung aufgenommen. Fragmente, die davor beginnen `StartTimestamp` und danach andauern, sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` ja`SERVER_TIMESTAMP`, `StartTimestamp` muss der später als der Stream Head liegen.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Fragment

Bedienung: 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 beträgt 1 Zeichen. 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 einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FragmentSelector

Bedienung: Amazon Kinesis Video Streams Archived Media

Beschreibt den Zeitstempelbereich und den Ursprung des Zeitstempels einer Reihe von Fragmenten.

Es werden nur Fragmente zurückgegeben, deren Startzeitstempel größer oder gleich der angegebenen Startzeit und kleiner oder gleich der Endzeit ist. 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

Der Ursprung 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 einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for 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.

Wenn auf gesetzt `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und Get HLSStreaming SessionUrl: `ON_DEMAND` oder PlaybackMode ist, wird das erste Fragment`LIVE_REPLAY`, das mit einem Producer-Zeitstempel innerhalb des angegebenen FragmentSelector: aufgenommen wurde, in die TimestampRange Medien-Playlist aufgenommen. Darüber hinaus sind die Fragmente mit Producer-Zeitstempeln innerhalb des `TimestampRange` aufgenommenen Fragments unmittelbar nach dem ersten Fragment enthalten (bis zum Wert „Get HLSStreaming sessionUrl:“). MaxMediaPlaylistFragmentResults

Fragmente mit doppelten Producer-Zeitstempeln werden dedupliziert. Das heißt, wenn Produzenten einen Stream von Fragmenten mit Produzenten-Zeitstempeln erzeugen, die ungefähr der tatsächlichen Uhrzeit entsprechen, enthalten die HLS-Medien-Playlisten alle Fragmente innerhalb des angeforderten Zeitstempelbereichs. Wenn einige Fragmente innerhalb desselben Zeitraums und zu sehr unterschiedlichen Zeitpunkten aufgenommen werden, wird nur die älteste aufgenommene Fragmentsammlung zurückgegeben.

Wenn auf gesetzt `FragmentSelectorType` ist `PRODUCER_TIMESTAMP` und Get HLSStreaming SessionUrl: PlaybackMode ist`LIVE`, werden die Producer-Zeitstempel in den Fragmenten und für die MP4 Deduplizierung verwendet. Die zuletzt aufgenommenen Fragmente, die auf Serverzeitstempeln basieren, sind jedoch in der HLS-Medien-Playlist enthalten. Das bedeutet, dass Fragmente, die in der Vergangenheit aufgenommen wurden, zwar mit Producer-Zeitstempeln und aktuellen Werten versehen sind, aber nicht in der HLS-Medien-Playlist enthalten sind.

Der Standardwert ist `SERVER_TIMESTAMP`.

Typ: Zeichenfolge

Zulässige Werte: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Erforderlich: Nein

TimestampRange

Der Anfang und das Ende des Zeitstempelbereichs für die angeforderten Medien.

Dieser Wert sollte nicht vorhanden sein, wenn er vorhanden `PlaybackType` ist`LIVE`.

Typ: HLSTimestampRange Objekt

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HLSTimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Anfang und das Ende des Zeitstempelbereichs für das angeforderte Medium.

Dieser Wert sollte nicht vorhanden sein, wenn er vorhanden `PlaybackType` ist`LIVE`.

Inhalt

EndTimestamp

Das Ende des Zeitstempelbereichs für das angeforderte Medium. Dieser Wert muss innerhalb von 24 Stunden nach dem angegebenen Wert liegen`StartTimestamp`, und er muss nach dem angegebenen `StartTimestamp` Wert liegen.

Wenn `FragmentSelectorType` für die Anfrage ein Wert `SERVER_TIMESTAMP` angegeben ist, 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 festgelegt `EndTimestamp` ist, enthält die Sitzung weiterhin neu aufgenommene Fragmente, bis die Sitzung abläuft.

> **Note**
>
> Dieser Wert ist inklusive. Der `EndTimestamp` wird mit dem (Start-) Zeitstempel des Fragments verglichen. Fragmente, die vor dem `EndTimestamp` Wert beginnen und nach diesem Wert andauern, werden in die Sitzung aufgenommen.

Typ: Zeitstempel

Erforderlich: Nein

StartTimestamp

Der Beginn des Zeitstempelbereichs für das angeforderte Medium.

Wenn der `HLSTimestampRange` Wert angegeben ist, ist der `StartTimestamp` Wert erforderlich.

Nur Fragmente, die genau bei oder danach beginnen, `StartTimestamp` werden in die Sitzung aufgenommen. Fragmente, die davor beginnen `StartTimestamp` und danach andauern,

sind nicht in der Sitzung enthalten. Wenn `FragmentSelectorType` ja`SERVER_TIMESTAMP`, `StartTimestamp` muss der später als der Streamkopf sein.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Image

Bedienung: Amazon Kinesis Video Streams Archived Media

Eine Struktur, die die `TimestampError`, und enthält`ImageContent`.

Inhalt

Error

Die Fehlermeldung, die angezeigt wurde, wenn das Bild für den angegebenen Zeitstempel aufgrund eines Fehlers, der nicht versucht werden konnte, nicht extrahiert wurde. Ein Fehler wird zurückgegeben, wenn:

- Für die angegebene Datei ist kein Medium vorhanden`Timestamp`.
- Das Medium für den angegebenen Zeitraum 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`

Erforderlich: Nein

ImageContent

Ein Attribut des `Image` Objekts, das Base64-kodiert ist.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Die maximale Länge beträgt 6291456.

Erforderlich: Nein

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 schließen oder um das Paginierungsfenster besser zu verstehen.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TimestampRange

Bedienung: Amazon Kinesis Video Streams Archived Media

Der Bereich der Zeitstempel, für die Fragmente zurückgegeben werden sollen.

Inhalt

EndTimestamp

Der letzte Zeitstempel im Bereich der Zeitstempel, für den Fragmente zurückgegeben werden sollen.

Typ: Zeitstempel

Erforderlich: Ja

StartTimestamp

Der Startzeitstempel im Bereich der Zeitstempel, für den Fragmente zurückgegeben werden sollen.

Typ: Zeitstempel

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Kinesis-Videosignalkanäle

Die folgenden Datentypen werden von Amazon Kinesis Video Signaling Channels unterstützt:

- IceServer

IceServer

Service: 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 beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Ttl

Gibt die Zeitspanne (in Sekunden) an, die der Benutzername und das Passwort gültig sind.

Typ: Ganzzahl

Gültiger Bereich: Fester Wert von 300.

Erforderlich: Nein

Uris

Ein Array von URIs, in der im I-D angegebenen Form. petithuguenin-behave-turn-urisspezifikation. Diese URIs stellen die verschiedenen Adressen und/oder Protokolle bereit, die verwendet werden können, um den TURN-Server zu erreichen.

Typ: Zeichenfolgen-Array

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Username

Ein Benutzername für die Anmeldung am ICE-Server.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Pattern: `[a-zA-Z0-9_.-]+`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie unter:

- AWS SDK for C++
- AWS SDK for Java
- AWS SDK for Ruby V3

Amazon Kinesis Video WebRTC-Speicher

Die folgenden Datentypen werden von Amazon Kinesis Video WebRTC Storage unterstützt:

Häufige Fehler

In diesem Abschnitt werden die Fehler aufgeführt, die bei den API-Aktionen aller AWS Dienste häufig 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 angegebene AWS Zugriffsschlüssel-ID ist in unseren Aufzeichnungen nicht vorhanden.

HTTP Status Code: 403

NotAuthorized

Sie haben keine Berechtigung zum Ausführen dieser Aktion.

HTTP Status Code: 400

OptInRequired

Für die AWS Zugriffsschlüssel-ID ist ein Abonnement für den Dienst erforderlich.

HTTP Status Code: 403

RequestExpired

Die Anfrage erreichte den Service mehr als 15 Minuten nach dem Datumsstempel auf der Anfrage oder mehr als 15 Minuten nach dem Ablaufdatum der Anfrage (z. B. bei vorsignierter Anfrage URLs), oder der Datumsstempel auf der Anfrage liegt mehr als 15 Minuten in der future.

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 Dienst angegebenen 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 Anfrage geschrieben wurde, ausgedrückt im Format YYYY-MM-DD.

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 im IAM-Benutzerhandbuch unter Erstellen einer signierten AWS API-Anfrage.

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'SSMMSS'Z') sein. Beispielsweise ist die folgende Datums- und Uhrzeitangabe 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-Grundformat angegeben ist, X-Amz-Date ist dies nicht erforderlich. Wenn verwendet X-Amz-Date wird, überschreibt er 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 Aufruf von AWS Security Token Service (AWS STS) abgerufen wurde. Eine Liste der Services, die temporäre Sicherheits-Anmeldeinformationen

von AWS STS unterstützen, finden Sie im IAM-Benutzerhandbuch unter AWS-Services , die mit IAM funktionieren.

Bedingung: Wenn Sie temporäre Sicherheitsanmeldedaten von verwenden AWS STS, müssen Sie das Sicherheitstoken angeben.

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 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

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 AWS IoT
- 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?
- Einen vom Kunden verwalteten Schlüssel erstellen und verwenden
- Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels

Was ist serverseitige Verschlüsselung für Kinesis Video Streams?

Die serverseitige Verschlüsselung ist eine Funktion in Kinesis Video Streams, mit der Daten automatisch mit einem von Ihnen angegebenen AWS KMS Schlüssel verschlüsselt werden, bevor sie im Ruhezustand gespeichert werden. 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 `aws/kinesisvideo` Standard-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/kinesisvideo`) verwaltet wird, ist kostenlos.
- Benutzergenerierte KMS-Schlüssel sind kostenpflichtig AWS KMS key . Weitere Informationen finden Sie unter AWS Key Management Service – Preise.

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 Von AWS verwalteter Schlüssel (von Kinesis Video Streams bereitgestellt) verwendet.

Ein vom Benutzer bereitgestellter KMS-Schlüssel muss einem Kinesis-Videostream zugewiesen werden, wenn er erstellt wird. Sie können einem Stream später mithilfe der UpdateStreamAPI 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 CreateStreamAPI die Schlüssel-ID im `KmsKeyId` Parameter an.

Einen vom Kunden verwalteten 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.

Einen vom Kunden verwalteten 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 Liste der vom Kunden verwalteten Schlüssel zurück.

Verwenden eines vom Kunden verwalteten Schlüssels

Nachdem Ihren Verbrauchern, Produzenten und Administratoren die richtigen Berechtigungen zugewiesen wurden, können Sie benutzerdefinierte KMS-Schlüssel in Ihrem eigenen AWS-Konto oder einem anderen System verwenden AWS-Konto. Alle KMS-Schlüssel in Ihrem Konto werden in der Liste der vom Kunden verwalteten Schlüssel 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/kinesisvideo`) verwenden, ist der Schlüssel erst in Ihrem Konto vorhanden, wenn der erste `PutMedia` `GetMedia` OR-Vorgang ausgeführt wird.

Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels

Bevor Sie serverseitige Verschlüsselung mit einem vom Kunden verwalteten Schlüssel verwenden können, müssen Sie KMS-Schlüsselrichtlinien so konfigurieren, dass sie die Verschlüsselung von Streams sowie die Verschlüsselung und Entschlüsselung von Stream-Datensätzen 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 einen vom Kunden verwalteten Schlüssel verwenden, stellen Sie sicher, dass Ihre Kinesis-Videostream-Produzenten und -Verbraucher (IAM-Principals) Benutzer gemäß der AWS KMS Standardschlüsselrichtlinie 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-Richtlinienerklärung gilt für die Ressourcen, die Sie mithilfe ihrer angeben. ARNs

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

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

Zum Beispiel:

```
"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 Streams in Kinesis Video Streams durchzuführen. Die folgende Übersicht beschreibt die allgemeinen Schritte, um den kontenübergreifenden Zugriff auf Videostreams zu gewähren:

1. Rufen Sie die 12-stellige Konto-ID des Kontos ab, dem Sie Berechtigungen zur Ausführung von Vorgängen mit der in Ihrem Konto erstellten Stream-Ressource gewähren möchten.

   Beispiel: In den folgenden Schritten verwenden wir 111111111111 als Konto-ID für das Konto, dem Sie die Erlaubnis erteilen möchten, und 999999999999 als ID für Ihre Kinesis Video Streams
2. Erstellen Sie eine IAM-verwaltete Richtlinie für das Konto, dem der Stream gehört (999999999999), die die Zugriffsebene ermöglicht, die Sie gewähren möchten.

   Beispielrichtlinie:

   JSON

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesisvideo:GetDataEndpoint",
                "kinesisvideo:DescribeStream",
                "kinesisvideo:PutMedia"
            ],
            "Resource": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/
custom-stream-name/1613732218179"
        }
    ]
}
```

Weitere Beispielrichtlinien für Kinesis Video Streams Streams-Ressourcen finden Sie Beispielrichtlinien im nächsten Abschnitt.

3. Erstellen Sie eine Rolle in dem Konto, dem der Stream gehört (999999999999), und geben Sie das Konto an, für das Sie Berechtigungen gewähren möchten (111111111111). Dadurch wird der Rolle eine vertrauenswürdige Entität hinzugefügt.

   Beispiel für eine vertrauenswürdige Richtlinie:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

   Hängen Sie die Richtlinie, die Sie im vorherigen Schritt erstellt haben, an diese Rolle an.

Sie haben jetzt eine Rolle im Konto 999999999999 erstellt, die über die Berechtigung für Operationen wie `DescribeStream`GetDataEndpoint, und für einen Stream-Ressourcen-ARN in `PutMedia` der verwalteten Richtlinie verfügt. Diese neue Rolle vertraut auch darauf, dass das andere Konto, 111111111111, diese Rolle übernimmt.

> **Important**
>
> Notieren Sie sich den Rollen-ARN, Sie benötigen ihn im nächsten Schritt.

4. Erstellen Sie eine verwaltete Richtlinie für das andere Konto, 111111111111, die die `AssumeRole` Aktion für die Rolle ermöglicht, die Sie im vorherigen Schritt für das Konto 999999999999 erstellt haben. Sie müssen die Rolle ARN aus dem vorherigen Schritt erwähnen.

   Beispiel für eine Richtlinie:

   JSON

   ```
   {
       "Version": "2012-10-17",
       "Statement": {
           "Effect": "Allow",
           "Action": "sts:AssumeRole",
           "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
       }
   }
   ```

5. Ordnen Sie die im vorherigen Schritt erstellte Richtlinie einer IAM-Entität zu, z. B. einer Rolle oder einem Benutzer im Konto 111111111111. Dieser Benutzer hat jetzt die Berechtigung, die Rolle im Konto 999999999999 anzunehmen. `CustomRoleName`

   Die Anmeldeinformationen dieses Benutzers rufen die AWS STS `AssumeRole` API auf, um die Sitzungsanmeldeinformationen abzurufen, die anschließend verwendet werden, um Kinesis Video Streams in dem Stream aufzurufen, der APIs im Konto 999999999999 erstellt wurde.

   ```
   aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/CustomRoleName" --
   role-session-name "kvs-cross-account-assume-role"
   {
       "Credentials": {
           "AccessKeyId": "",
   ```

```
        "SecretAccessKey": "",
        "SessionToken": "",
        "Expiration": ""
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "",
        "Arn": ""
    }
}
```

6. Legen Sie den Zugriffsschlüssel, den geheimen Schlüssel und die Sitzungsanmeldedaten auf der Grundlage der vorherigen Einstellungen in der Umgebung fest.

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=
```

7. Führen Sie Kinesis Video Streams aus APIs , um den Datenendpunkt für den Stream im Konto 999999999999 zu beschreiben und abzurufen.

```
aws kinesisvideo describe-stream --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179"
{
    "StreamInfo": {
        "StreamName": "custom-stream-name",
        "StreamARN": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-
stream-name/1613732218179",
        "KmsKeyId": "arn:aws:kms:us-west-2:999999999999:alias/aws/kinesisvideo",
        "Version": "abcd",
        "Status": "ACTIVE",
        "CreationTime": "2018-02-19T10:56:58.179000+00:00",
        "DataRetentionInHours": 24
    }
}

aws kinesisvideo get-data-endpoint --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179" --api-name "PUT_MEDIA"
{
    "DataEndpoint": "https://s-b12345.kinesisvideo.us-west-2.amazonaws.com"
}
```

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

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.

JSON

```
{
    "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.

JSON

```
{
    "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.

JSON

```
{
    "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 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 aber 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 CertificateId 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:

   JSON

```
{
    "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 Folgendes `iam-permission-document.json` verwenden:

JSON

```
{
    "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 angegeben ist. (`${credentials-iot:ThingName}`) Dieser Platzhalter nimmt den Wert des AWS IoT Thing-Attributs an, `ThingName` wenn der Anbieter für 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`
```

Die Ausgabe-JSON enthält die `accessKey`, `secretKey`, und die`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++” 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`):

  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 das an den Anbieter `thingname` für Anmeldeinformationen weitergeben. AWS IoT Sie können `getenv` es verwenden, um den Dingnamen an die Demo-Anwendung zu übergeben, ähnlich wie bei der Übergabe der anderen AWS IoT Attribute. 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`):

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"
```

Konformitätsprüfung für Amazon Kinesis Video Streams

Informationen darüber, ob AWS-Service ein in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services 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 herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen 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:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen 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— Auf diese AWS-Service Weise erhalten Sie 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. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.

- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- 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 `PutMedia`, `GetStreamingEndpoint`, 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 “API-Aufrufe protokollieren mit CloudTrail”.

Problembehebung bei Kinesis Video Streams

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme mit Amazon Kinesis Video Streams zu beheben.

Themen

- Allgemeine Probleme
- Probleme mit der API
- HLS-Probleme
- Probleme mit Java
- Probleme mit der Producer-Bibliothek
- Probleme mit der Stream-Parser-Bibliothek
- Probleme mit dem Netzwerk

Allgemeine Probleme

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`kinesis_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.

Probleme mit der API

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: „Die Anmeldeinformationen sollten auf eine gültige Region beschränkt sein.“

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 Standardausgabetyp (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 kinesisvideo describe-stream --stream-name StreamName --endpoint https://
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Schlägt der Anruf 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 Inhaltesspeicher nicht ausreichend groß dimensioniert wurde. Um den Inhaltsspeicher zu vergrößern, erhöhen Sie den Wert von `StorageInfo.storageSize`. Weitere Informationen finden Sie unter StorageInfo.

Fehler: „Die Anmeldeinformationen sollten auf eine gültige Region beschränkt sein.“

Dieser Fehler tritt auf, wenn die Signaturregion nicht mit der Endpunktregion übereinstimmt.

Wenn Sie `us-west-2` beispielsweise Ihre Signierregion angeben, aber versuchen, eine Verbindung zum Endpunkt `kinesisvideo.us-east-1.amazonaws.com` (`us-east-1`) herzustellen, wird dieser Fehler angezeigt.

In einigen Anwendungen, wie kvssink, ist die Regions-Fallback-Kette standardmäßig auf eingestellt. `us-west-2` Stellen Sie sicher, dass Sie Ihre Region entsprechend der von Ihnen verwendeten Anwendung richtig eingestellt haben.

HLS-Probleme

Falls Ihr Videostream nicht korrekt wiedergegeben wird, finden Sie weitere Informationen unterthe section called “Fehlerbehebung bei HLS-Problemen”.

Probleme mit Java

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.

Probleme mit der Producer-Bibliothek

In diesem Abschnitt werden Probleme beschrieben, die bei der Verwendung von Zu Kinesis Video Streams hochladen auftreten können.

Problembereiche

- Das Producer-SDK kann nicht kompiliert werden
- Der Videostream wird nicht in der Konsole angezeigt

- Fehler: „Das in der Anfrage enthaltene Sicherheitstoken ist ungültig“ beim Streamen von Daten mit der Demo-Anwendung GStreamer
- Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“
- GStreamer Die Anwendung wird unter OS X mit der Meldung „Streaming gestoppt, Grund nicht ausgehandelt“ beendet
- Fehler: „Heap konnte nicht zugewiesen werden“ beim Erstellen des Kinesis Video Client in der GStreamer Demo auf dem Raspberry Pi
- Fehler: „Ungültiger Befehl“ beim Ausführen der GStreamer Demo auf dem 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 Demo-Anwendung GStreamer
- 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-Anpassungsflags 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, wenn 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 extrahiert NALUs (die dann Ab (Sps) Ab (Pps) sein werden) und sie im CPD in einfügt`StreamDefinition`.

Fehler: „Das in der Anfrage enthaltene Sicherheitstoken ist ungültig“ beim Streamen von Daten mit der Demo-Anwendung GStreamer

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 qualitativ hochwertigen Stream eine höhere Bitrate ein und korrigieren Sie jeglichen Jitter im Quellstream, sofern die Kamera dies unterstützt.

GStreamer Die Anwendung wird unter OS X mit der Meldung „Streaming gestoppt, Grund nicht ausgehandelt“ beendet

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 Problemumgehung 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: „Heap konnte nicht zugewiesen werden“ beim Erstellen des Kinesis Video Client in der GStreamer Demo auf dem Raspberry Pi

Die GStreamer Beispielanwendung versucht, 512 MB RAM zuzuweisen, die auf Ihrem System möglicherweise nicht 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: „Ungültiger Befehl“ beim Ausführen der GStreamer Demo auf dem Raspberry Pi

Wenn beim Ausführen der GStreamer Demo der folgende Fehler auftritt, stellen Sie sicher, dass 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 Demo-Anwendung GStreamer

Um Curl-Fehler bei der Ausführung der GStreamer Demo-Anwendung zu beheben, kopieren Sie diese Zertifikatsdatei nach. `/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-Anpassungsflags 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 an:

https://github.com/awslabs/aws-sdk-android-samples/blob/master/AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/StreamConfigurationFragment.java#L169

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-Dienstkontingente 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 festSchreiben 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. Legen Sie beispielsweise den Wert 60 und key-int-max die Bildrate pro Sekunde auf 30 fest, 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`iot-thingname`. Weitere Informationen finden Sie unter GStreamer Referenz zu den Elementparametern.

Probleme mit der Stream-Parser-Bibliothek

In diesem Abschnitt werden Probleme beschrieben, die bei der Verwendung von Streamen Sie mithilfe der 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 unterstützt 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.

Probleme mit dem Netzwerk

Wenn beim Versuch, eine Verbindung zu Kinesis Video Streams herzustellen, Verbindungsfehler wie „Verbindungstimeout“ oder „Verbindung fehlgeschlagen“ angezeigt werden, kann dies an Einschränkungen des IP-Adressbereichs in Ihrer Netzwerkkonfiguration liegen.

Wenn in Ihrem Setup IP-Adressbereichsbeschränkungen für Kinesis Video Streams gelten, aktualisieren Sie Ihre Netzwerkkonfiguration, sodass die IP-Adressbereiche von Kinesis Video Streams zugelassen werden.

> **Important**
>
> Die IP-Bereichsliste ist keine vollständige Liste der IP-Adressen von Kinesis Video Streams. Geben Sie die angezeigten IP-Adressbereiche an und beachten Sie, dass sich die IP-Adressen im Laufe der Zeit ändern können.

Weitere Informationen finden Sie unter AWS IP-Bereiche. Folgen Sie dem Abonnementverfahren, um benachrichtigt zu werden, wenn sich IP-Bereiche ändern.

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: 6. Januar 2025

Änderung	Beschreibung	Datum
C++ auf Raspberry Pi	Aktualisierte Dokumentation für die Verwendung des C++-Producer-SDK auf Raspberry Pi.	6. Januar 2025
Verbindung zum Edge-Agenten Edge-to-Cloud von Amazon Kinesis Video Streams	Neue Funktionsversion. Weitere Informationen finden Sie unter Planen Sie die Videoaufnahme und Speicherung.	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
Streaming-Metadaten	Sie können das Producer SDK verwenden, um Metadaten in einen Kinesis-Videostrea m einzubetten. Weitere Informationen finden Sie unter Verwenden von Streaming-	28. September 2018

Änderung	Beschreibung	Datum
	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 Sie die Protokollierung mit dem C++-Producer-SDK.	18. Juli 2018
HLS-Video-Streaming	Sie können jetzt einen Kinesis-Videostream mit 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-Plugin GStreamer	Zeigt, wie das erstellt wirdC++, um es als GStreamer Ziel zu verwenden. Weitere Informationen finden Sie unter GStreamer Plug-in — kvssink.	15. Juni 2018

Änderung	Beschreibung	Datum
Producer SDK-Callbacks-Referenzdokumentation	Referenzdokumentation für die vom Zu Kinesis Video Streams hochladen verwendeten Callbacks. Weitere Informationen finden Sie unter SDK-Rückrufe von Produzenten.	12. Juni 2018
Systemanforderungen	Dokumentation für Arbeitsspeicher- und Speichernforderungen für Producer-Geräte und SDK. Weitere Informationen finden Sie unter Systemanforderungen für Amazon Kinesis Video Streams.	30. Mai 2018
CloudTrail Unterstützung	Dokumentation CloudTrail zur Überwachung der API-Nutzung. Weitere Informationen finden Sie unter API-Aufrufe von Amazon Kinesis Video Streams protokollieren mit AWS CloudTrail.	24. Mai 2018
Producer SDK-Strukturen – Referenzdokumentation	Referenzdokumentation für die vom Zu Kinesis Video Streams hochladen 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-Videostream dekodiert und angezeigt werden. Weitere Informationen finden Sie unter Beispiel: Analysieren und Rendern von Kinesis Video Streams Streams-Fragmenten.	15. März 2018
Producer-SDK-Limits-Referenzdokumentation	Informationen zu Einschränkungen hinsichtlich Operationen in der C++. Weitere Informationen finden Sie unter SDK-Kontingente für Produzenten.	13. März 2018
Überwachen	Informationen zur Überwachung von Kinesis Video Streams Streams-Metriken und API-Aufrufen 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-Anpassungsflags.	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.	12. Januar 2018
Kinesis-Video-Beispiel-Dokumentation	Dokumentation für die Kinesis Video-Beispielanwendung, in der gezeigt wird, wie die Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an 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

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.