



Entwicklerhandbuch

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: Entwicklerhandbuch

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die 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	1
Funktionsweise	3
API- und Producer-Bibliotheken	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-metadata und Kontingente für fragmentierte Medien	29
Kontingente für Streaming-Metadaten	32
SDK-Kontingente für Produzenten	32
Mehrstufiger Speicher	36
-Übersicht	37
Die wichtigsten Konzepte	37
Tiered Storage einrichten	38
Bewährte Methoden	38
Einrichten eines -Kontos	38
Melden Sie sich an für AWS-Konto	39
Erstellen Sie einen AWS-Konto Schlüssel	39
Vorteile	39
Erste Schritte	41
Erstellen Sie einen Amazon Kinesis-Videostream	41
Erstellen Sie einen Videostream mit der Konsole	42
Erstellen Sie einen Videostream mit dem AWS CLI	42
Daten an einen Amazon Kinesis-Videostream senden	42
Erstellen Sie das SDK und die Beispiele	43
Führen Sie die Samples aus, um Medien in 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
Vorgehensweise: 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
Vorgehensweise: 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	93
Erstellen eines Benutzers	94
Treten Sie Ihrem Netzwerk bei	96
Fernverbindung herstellen	96
Konfigurieren Sie die Kamera	97
Installieren Sie die Softwarevoraussetzungen	99
Laden Sie das C++ Producer SDK herunter und erstellen Sie es	100
Streamen Sie einen Livestream	103
Medien wiedergeben	115
Fehlerbehebung	119
Fehlercodereferenz	123
Von PutFrame Callbacks — Platform Independent Code (PIC) zurückgegebene Fehler und Statuscodes	124
Von PutFrame Callbacks zurückgegebene Fehler und Statuscodes — C-Producer-Bibliothek	179
NAL-Anpassungsflags	186
Produzentenstrukturen	188
DeviceInfo/DefaultDeviceInfoProvider	188
StorageInfo	188
Stream-Strukturen	190
StreamDefinition/StreamInfo	190
ClientMetrics	210
StreamMetrics	211
Rückrufe	213
ClientCallbackProvider	214
StreamCallbackProvider	214
ClientCallbacks	215
Callback-Implementierungen, um das Streaming erneut zu versuchen	220
Streaming-Metadaten verwenden	221
Hinzufügen von Metadaten zu einem Kinesis-Videostream	222
Verwenden Sie IPv6 mit Kinesis Video Streams	224
Konfigurieren Sie die AWS SDK für IPv6	224
Das Kinesis Video Streams Producer SDK für IPv6 konfigurieren	226

Konfigurieren Sie das AWS CLI für IPv6	228
Beispiele für Konfigurationen	229
Überlegungen	229
Kunden, die von der Aktualisierung auf IPv6 betroffen sind	230
Fehlerbehebung	231
Videowiedergabe	233
Anforderungen an die Wiedergabe	234
GetClip	234
GetDASHStreamingSessionURL	235
GetHLSStreamingSessionURL	236
GetImages	237
Wiedergabe mit HLS	238
Benutze das AWS CLI um eine URL für eine HLS-Streaming-Sitzung abzurufen	238
Beispiel: Verwenden Sie HLS in HTML und JavaScript	242
Behebung von HLS-Problemen	247
Wiedergabe mit MPEG-DASH	249
Beispiel: Verwendung in HTML und MPEG-DASH JavaScript	250
Richten Sie Benachrichtigungen ein	254
Benachrichtigungskonfigurationen verwalten	254
UpdateNotificationConfiguration	254
DescribeNotificationConfiguration	255
Über die MKV-Tags des Herstellers	255
Syntax für Hersteller-MKV-Tags	255
Grenzwerte für MKV-Tags	256
Amazon SNS-Nachrichten	256
Payload zum Thema Amazon SNS	256
Sehen Sie sich Ihre Amazon SNS-Nachrichten an	257
Cross-account Veröffentlichung von Amazon SNS-Benachrichtigungen	258
Identity-based Konfiguration der Richtlinie	258
Resource-based Konfiguration der Richtlinie	259
Anforderungen und Überlegungen	260
Extrahieren Sie Bilder aus Videostreams	261
Real-time Generierung von Bildern	261
Hinzufügen von Tags zur Bildgenerierung zu Fragmenten	264
Amazon S3-Objektpfad (Bild)	267
Bildmetadaten werden abgerufen	268

Amazon S3-URI-Empfehlungen zum Schutz vor Drosselung	268
Fehlerbehebung	269
Greifen Sie auf Videoanalysen zu	272
Metadaten konsumieren	272
Streamen Sie mithilfe der Parser-Bibliothek	273
Voraussetzungen	274
Laden Sie den Code herunter	274
Untersuchen Sie den Code	275
Führen Sie den Code aus	282
Überwachen	282
Überwachen Sie Metriken mit CloudWatch	283
Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch	303
API-Aufrufe protokollieren mit CloudTrail	308
Beschränkungen für Streaming-Metadaten	313
Planen Sie die Videoaufnahme und -speicherung	314
API-Operationen für Amazon Kinesis Video Streams Edge Agent	315
Überwachung des Amazon Kinesis Video Streams Edge Agent	315
Im AWS IoT Greengrass Nicht-Modus bereitstellen	315
Abhängigkeiten installieren	316
Erstellen Sie Ressourcen für Ihre IP-Kamera-RTSP-URLs	318
Erstellen Sie eine IAM-Berechtigungsrichtlinie	320
Erstellen einer IAM-Rolle	322
Erstellen Sie den AWS IoT Rollenalias	324
Erstellen Sie die AWS IoT Richtlinie	324
Erstelle AWS IoT etwas und erhalte AWS IoT Core Anmeldeinformationen	326
Erstellen Sie den Edge-Agenten	329
Installieren Sie den CloudWatch Agenten	340
Führen Sie Edge Agent als nativen Prozess aus	343
Bereitstellen auf AWS IoT Greengrass	346
Erstellen Sie eine Ubuntu-Instanz	347
Richten Sie das AWS IoT Greengrass Kerngerät ein	348
Erstellen Sie Ressourcen für Ihre IP-Kamera-RTSP-URLs	350
Fügen Sie der TES-Rolle Berechtigungen hinzu	352
Installieren Sie die Secret Manager-Komponente	355
Stellen Sie Edge Agent auf dem Gerät bereit	358
Installieren Sie die AWS IoT Greengrass Log-Manager-Komponente	368

Häufig gestellte Fragen	372
Welche Betriebssysteme unterstützt Amazon Kinesis Video Streams Edge Agent?	372
Unterstützt der Amazon Kinesis Video Streams Edge Agent Medien? H.265	372
Funktioniert der Amazon Kinesis Video Streams Edge Agent in AL2?	372
Wie kann ich mehrere Streams innerhalb von ausführen AWS IoT Ding oder Gerät?	373
Wie kann ich eine bearbeiten, StartEdgeConfigurationUpdate nachdem sie gesendet wurde?	373
Haben Sie Beispiele für Gemeinsamkeiten ScheduleConfigs?	373
Gibt es ein maximales Stream-Limit?	374
Wie starte ich einen Job neu, bei dem ein Fehler aufgetreten ist?	374
Wie überwache ich den Zustand meines Amazon Kinesis Video Streams Edge Agent?	374
Streamen Sie Videos über eine VPC	375
Zusätzliche Informationen	375
VPC-Endpunktverfahren	376
Beispiele	378
Beispiele: Senden von Daten an Kinesis Video Streams	378
Beispiele: Abrufen von Daten aus Kinesis Video Streams	378
Beispiele: Wiedergeben von Videodaten	378
Voraussetzungen	379
GStreamer-Plugin — kvssink	379
Laden Sie das GStreamer-Element herunter, erstellen Sie es und konfigurieren Sie es	380
Führen Sie das GStreamer-Element aus	381
Startbefehle	382
Führen Sie das GStreamer-Element in einem Docker-Container aus	384
Parameterreferenz	387
PutMedia API	402
Laden Sie den Code herunter und konfigurieren Sie ihn	403
Schreiben Sie den Code und untersuchen Sie ihn	404
Führen Sie den Code aus und überprüfen Sie ihn	406
RTSP und Docker	407
Video-Tutorials	408
Voraussetzungen	408
Erstellen Sie das Docker-Image	408
Führen Sie die RTSP-Beispielanwendung aus	409
Renderer	410
Voraussetzungen	411

Ausführen des Renderer-Beispiels	411
So funktioniert's	412
Sicherheit	414
Datenschutz	415
Was ist serverseitige Verschlüsselung für Kinesis Video Streams?	415
Überlegungen zu Kosten, Regionen und Leistung	415
Wie fange ich mit der serverseitigen Verschlüsselung an?	416
Erstellen und Verwenden eines vom Kunden verwalteten Schlüssels	417
Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels	418
Steuerung des Zugriffs auf Kinesis Video Streams-Ressourcen mithilfe von IAM	420
Richtliniensyntax	420
Aktionen für Kinesis Video Streams	421
Amazon-Ressourcennamen (ARNs) für Kinesis Video Streams	422
Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren	422
Beispielrichtlinien	426
Steuern Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT	428
AWS IoT ThingName als Stream-Name	429
AWS IoT CertificateId als Stream-Name	436
Verwenden Sie AWS IoT Anmeldeinformationen zum Streamen an einen fest codierten Stream-Namen	438
Compliance-Validierung	439
Ausfallsicherheit	440
Infrastruktursicherheit	440
Bewährte Methoden für die Sicherheit	440
Implementieren des Zugriffs mit geringsten Berechtigungen	441
Verwenden von IAM-Rollen	441
Wird CloudTrail zur Überwachung von API-Aufrufen verwendet	442
Fehlerbehebung	443
Allgemeine Probleme	443
Latenz zu hoch	443
API-Probleme	444
Fehler: „Unbekannte Optionen“	444
Fehler: „Der service/operation Name, der autorisiert werden soll, konnte nicht bestimmt werden“	444
Fehler: „Failed to put a frame in the stream“ (Es konnte kein Frame in den Stream eingefügt werden)	445

Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“	445
Fehler: "STATUS_STORE_OUT_OF_MEMORY"	446
Fehler: „Die Anmeldeinformationen sollten auf eine gültige Region beschränkt sein.“	446
HLS-Probleme	446
Probleme mit Java	446
Aktivieren von Java-Protokollen	447
Probleme mit der Producer-Bibliothek	447
Das Producer-SDK kann nicht kompiliert werden	448
Der Videostream wird nicht in der Konsole angezeigt	448
Fehler: „Security token included in the request is invalid“ (Das Sicherheits-Token der Anfrage ist ungültig) beim Streamen von Daten mit der GStreamer-Demo-Anwendung	449
Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“	450
GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt“	450
Fehler: „Failed to allocate heap“ (Heap konnte nicht zugeordnet werden) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi	451
Fehler: „Illegal Instruction“ (Ungültige Anweisung) beim Ausführen der GStreamer-Demo auf Raspberry Pi	451
Die Kamera kann nicht auf Raspberry Pi geladen werden.	451
Kamera kann auf macOS High Sierra nicht gefunden werden.	452
Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden	452
Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung	452
Timestamp/range Behauptung zur Laufzeit auf Raspberry Pi	452
Assertion auf gst_value_set_fraction_range_full auf Raspberry Pi	452
STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d)-Fehler unter Android	453
Fehler: Die maximale Fragmentdauer wurde erreicht	453
"Ungültiger Objektname abgelaufen", Fehler bei der Verwendung der IoT-Autorisierung	454
Probleme mit der Stream-Parser-Bibliothek	454
Es kann nicht auf einen einzelnen Frame aus dem Stream zugegriffen werden	454
Fragmentdekodierungsfehler	455
Probleme mit dem Netzwerk	455
Dokumentverlauf	457
.....	cdlxii

Was ist Amazon Kinesis Video Streams?

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

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

Mit Kinesis Video Streams können Sie riesige Mengen an Live-Videodaten aus Millionen von Quellen 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. Wenn Live-Videostreams aus diesen Quellen in einen Kinesis-Videostream übertragen werden, können Sie Anwendungen erstellen, um Bild für Bild in Echtzeit auf die Daten zuzugreifen, um eine Verarbeitung mit niedriger Latenz zu ermöglichen. Kinesis Video Streams ist quellenunabhängig. Sie können mithilfe der [GStreamer-Plugin — kvssink](#) Bibliothek Videos von der Webcam eines Computers oder mithilfe des Echtzeit-Streaming-Protokolls (RTSP) von einer Kamera in Ihrem Netzwerk 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 der Zeitstempel des Produzenten und der Aufnahme. Sie können Anwendungen erstellen, die die Videodaten in regelmäßigen Abständen stapelweise verarbeiten, oder Sie können Anwendungen erstellen, die für verschiedene Anwendungsfälle einen einmaligen Zugriff auf historische Daten benötigen.

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

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-East)	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 (Malaysia)	ap-southeast-3
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
Europa (Spanien)	eu-south-2

Name der Region	AWS Regionalcode
Naher Osten (Bahrain)	me-south-1
Südamerika (São Paulo)	sa-east-1

Kinesis Video Streams: So funktioniert es

Themen

- [Unterstützung für Kinesis Video Streams-API und Producer-Bibliotheken](#)
- [Kinesis Video Streams-Datenmodell](#)

Mit Amazon Kinesis Video Streams, einem vollständig verwalteten AWS-Service, können Sie Live-Videos von Geräten auf das streamen AWS Cloud und dauerhaft 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 einspeist. Ein Produzent kann jedes Gerät sein, das Videos generiert, 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 Producer-Bibliotheken** — Eine Reihe von Software und Bibliotheken, die Sie auf Ihren Geräten installieren und konfigurieren können. Sie können diese Bibliotheken verwenden, um Videos auf verschiedene Arten sicher zu verbinden und zuverlässig zu streamen, auch in Echtzeit, nachdem Sie sie einige Sekunden lang gepuffert haben, oder als nachträgliche Medienuploads.

- **Kinesis-Videostream** — Eine Ressource, die Sie verwenden können, um Live-Videodaten zu übertragen, optional zu speichern und die Daten sowohl in Echtzeit als auch stapelweise oder einmalig zur Nutzung zur Verfügung zu stellen. In einer typischen Konfiguration werden in einem Kinesis-Videostream nur Daten von einem Produzenten veröffentlicht.

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

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-Anwendungen bezeichnet. Sie können Anwendungen schreiben, die Daten in Kinesis Video Streams in Echtzeit verarbeiten und verarbeiten, oder nachdem die Daten gespeichert und zeitindexiert wurden, wenn eine Verarbeitung mit niedriger Latenz nicht erforderlich ist. Sie können diese Verbraucheranwendungen so erstellen, dass sie auf Amazon EC2-Instances ausgeführt werden.
- [Sehen Sie sich die Ausgabe von Kameras mithilfe der Parser-Bibliothek an](#)— Ermöglicht Kinesis Video Streams-Anwendungen, Medien zuverlässig und mit niedriger Latenz aus dem Kinesis-Videostream abzurufen. Zudem werden die Frame-Grenzen in den Medien analysiert, damit Anwendungen sich auf die Verarbeitung und Analyse der Frames konzentrieren können.

Unterstützung für Kinesis Video Streams-API und Producer-Bibliotheken

Kinesis Video Streams bietet APIs, mit denen Sie Streams erstellen und verwalten sowie Mediendaten in und aus einem Stream lesen oder schreiben können. Die Kinesis Video Streams-Konsole unterstützt neben den Verwaltungsfunktionen auch die Live- und Video-on-Demand-Wiedergabe. 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-API](#)
- [Muster zur Erkennung von Endpunkten](#)
- [Producer-Bibliotheken](#)

Kinesis Video Streams-API

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

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

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

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

Note

`GetMedia` weiß, wo sich die Fragmente befinden (archiviert im Datenspeicher oder verfügbar in Echtzeit). Wenn `GetMedia` beispielsweise feststellt, dass das Startfragment archiviert ist, wird das Zurückgeben von Fragmenten aus dem Datenspeicher gestartet. Wenn es neuere Fragmente zurückgeben muss, die noch nicht archiviert sind, wechselt `GetMedia` es zum Lesen von Fragmenten aus einem 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. Offline-Konsumenten können entscheiden, bestimmte Medienfragmente oder Bereiche von Videodaten explizit abzurufen, indem die APIs `ListFragments` und `GetMediaFromFragmentList` kombiniert werden. Mit `ListFragments` und `GetMediaFromFragmentList` kann eine Anwendung Videosegmente für einen bestimmten Zeitraum oder Fragmentbereich identifizieren und die betreffenden Fragmente dann sequenziell oder parallel zur Verarbeitung abrufen. Diese Herangehensweise eignet sich für MapReduce-Anwendungssuites, die große Datenmengen schnell parallel verarbeiten müssen.

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

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

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

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

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

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

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

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

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

Muster zur Erkennung von Endpunkten

REST-APIs der Steuerungsebene

Verwenden Sie die [Kinesis Video Streams-Dienstendpunkte](#), um auf die REST-APIs der [Kinesis Video Streams-Kontrollebene zuzugreifen](#).

REST-APIs der Datenebene

Kinesis Video Streams basiert auf einer [Mobilfunkarchitektur](#), um eine bessere Skalierung und eine bessere 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 Data Plane-REST-APIs zugreifen, müssen Sie die richtigen Endpunkte selbst verwalten und zuordnen. Dieser Prozess, das Endpoint-Discovery-Muster, wird im Folgenden beschrieben:

1. Das Endpoint-Discovery-Muster beginnt mit dem Aufruf einer der `GetEndpoints` Aktionen. Diese Aktionen gehören zur Control Plane.
 1. Wenn Sie die Endpunkte für die [Amazon Kinesis Video Streams Media API](#) oder [Amazon Kinesis Video Streams Archived Media](#) API-Services abrufen, verwenden Sie [GetDataEndpoint](#)
 2. Wenn Sie die Endpunkte für die [Amazon Kinesis Video Signaling Channels API](#), [Amazon Kinesis Video WebRTC Storage API](#) oder [Kinesis Video Signaling](#) abrufen, verwenden Sie <https://docs.aws.amazon.com/kinesisvideostreams-webrtc-dg/latest/devguide/kvswebrtc-websocket-apis.html> [GetSignalingChannelEndpoint](#)
2. Den Endpunkt zwischenspeichern und wiederverwenden.
3. Wenn der zwischengespeicherte Endpunkt nicht mehr funktioniert, rufen Sie erneut an, 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-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-Entwicklerhandbuch senden oder empfangen Daten im MKV-Format.

Der [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](#).

In den folgenden Abschnitten werden die Komponenten der von der erzeugten MKV-formatted Daten beschrieben. [C++](#)

Themen

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

Stream-Header-Elemente

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

Element	Description	Typische Werte
stream_name	Entspricht dem Namen des Kinesis-Videostreams.	my-stream
retention_period	Die Dauer, in Stunden, für die die Stream-Daten von Kinesis Video Streams gespeichert werden. Geben Sie 0 für einen Stream an, der keine Daten speichert.	24
Tags	Als Schlüssel-Wert-Paare angegebene Benutzerdaten. Diese Daten werden in der AWS-Managementkonsole 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 vorhanden, werden die Daten mit dem Kinesis-supplied Schlüssel () aws/kinesisvideo verschlüsselt.	01234567-89ab-cdef-0123-456789ab
streaming_type	Derzeit ist der einzige gültige Streaming-Typ STREAMING_TYPE_REALTIME.	STREAMING_TYPE_REALTIME

Element	Description	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 Skala an, die für Frame-Zeitstempel verwendet wird. 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 TimecodeScale in der Matroska-Dokumentation.	

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

Element	Description	Typische Werte
fragment_acks	Wenn <code>true</code> , werden Bestätigungen (ACKs) gesendet, wenn Kinesis Video Streams die Daten empfängt. Die ACKs können mit der Callback-Funktion <code>KinesisVideoStreamFragmentAck</code> oder <code>KinesisVideoStreamParseFragmentAck</code> angenommen werden.	<code>true</code>
restart_on_error	Gibt an, ob der Stream die Übertragung nach einem Fehler fortsetzen soll.	<code>true</code>
nal_adaptation_flags	Gibt an, ob der Inhalt NAL (Network Abstraction Layer)-Adaptions- oder Codec-interne Daten enthält. Gültige Flags sind <code>NAL_ADAPTATION_ANNEXB_NALS</code> und <code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code> .	<code>NAL_ADAPTATION_ANNEXB_NALS</code>
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 <code>0</code> wird der Standardwert <code>24</code> zugewiesen.	<code>24</code>

Element	Description	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. Wenn die Netzwerklatenz niedrig ist, kann dieser Wert reduziert werden. Wenn die Netzwerklatenz hoch ist, verhindert eine Erhöhung dieses Werts, dass Frames verloren gehen, bevor sie gesendet werden können, weil die Zuweisung dazu führt, dass Frames nicht in den kleineren Puffer gelangen.	

Element	Description	Typische Werte
replay_duration	Die Zeitspanne, in der der Videodatenstrom „zurückgespult“ wird, wenn die Verbindung unterbrochen wird. Dieser Wert kann Null sein, wenn der Verlust von Bildern aufgrund eines Verbindungsabbruchs kein Problem darstellt. Der Wert kann erhöht werden, wenn die nutzende 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	Description	Typische Werte
Codec PrivateData	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 die Pixelbreite und -höhe für den Frame.	
Codec PrivateDataSize	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	User-defined Segment-UUID (16 Byte).	
default_track_id	Eindeutige Zahl ungleich Null für den Track.	1

Streamen Sie Titeldaten

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

Element	Description	Typische Werte
track_name	User-defined Name des Titels. 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 https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/producer-sdk-cpp.html C++-Producer-Bibliothek MkvStatics.cpp enthält das MkvTrackVideoBitsGArray die Pixelbreite und -höhe für den Frame.	
cpd_size	Die Größe der Daten im PrivateData Codec-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 z. B. die Folgenden:
 - `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.
- **Decodierungszeitstempel:** Der Zeitstempel, zu dem dieser Frame dekodiert wurde. Wenn frühere Frames für die Decodierung von diesem Frame abhängen, könnte dieser Zeitstempel früher sein als der 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-Managementkonsole, 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), mit denen Daten codiert werden, H.264 da dadurch die Codierungslast der 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 Kodierung (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 codiert 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 APIs auf Kontrollebene beschrieben. TPS steht für Transaktionen pro Sekunde.

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

`ClientLimitExceededException`

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

API	Kontolimit: Anfrage	Kontolimit: Streams	Stream-level Limit	Relevante Ausnahmen und Hinweise
		https:// console.a ws.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]	
DescribeMappedResourceConfiguration	50 TPS [h]	N/A	5 TPS [h]	

API	Kontolimit: Anfrage	Kontolimit: Streams	Stream-level Limit	Relevante Ausnahmen und Hinweise
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]	
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]	

API	Kontolimit: Anfrage	Kontolimit: Streams	Stream-level Limit	Relevante Ausnahmen und Hinweise
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]	
UpdateStreamStorageConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeStreamStorageConfiguration	50 TPS [h]	N/A	5 TPS [h]	

API-Dienstkontingente für Medien und archivierte Medien

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

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

Wenn ein Grenzwert auf Verbindungsebene erreicht wird, wird die Ausnahme `ConnectionLimitExceededException` ausgelöst.

Die folgenden Fehler und Bestätigungen werden ausgelöst, wenn ein Grenzwert auf Fragmentebene erreicht wird:

- Eine `MIN_FRAGMENT_DURATION_REACHED`-Bestätigung wird zurückgegeben, wenn ein Fragment unterhalb der Mindestdauer liegt.
- Eine `MAX_FRAGMENT_DURATION_REACHED`-Bestätigung wird zurückgegeben, wenn ein Fragment oberhalb der Höchstdauer liegt.
- Eine `MAX_FRAGMENT_SIZE`-Bestätigung wird zurückgegeben, wenn ein Fragment die maximale Datengröße überschreitet.
- Eine `FragmentLimitExceeded`-Ausnahme wird ausgelöst, wenn ein Fragmentgrenzwert in einer `GetMediaForFragmentList`-Operation erreicht wird.

API-Dienstkontingente auf Datenebene

API	Stream-level begrenzen	Connection-level begrenzen	Bandbreitenbegrenzung	Fragment-level begrenzen	Relevante Ausnahmen und Hinweise
PutMedia	5 TPS [h]	1 [h]	12,5 MB/second oder 100 Mbit/s [s] pro Stream	• Minimale Fragmentdauer: 1 Sekunde [h] • Maximale Fragmentdauer: 20 Sekunden [h] • Maximale Fragmentgröße: 50 MB [h] • Maximale Anzahl	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, mehr als eine <code>PutMedia</code> Verbindung gleichzeitig herzustellen, drosselt Kinesis Video Streams die letzten Verbindungen mit einer Fehlermeldung.

API	Stream-level begrenzen	Connection-level begrenzen	Bandbreitenbegrenzung	Fragment-level begrenzen	Relevante Ausnahmen und Hinweise
				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]	
GetDASHStreamingSessionURL	25 TPS [h]	N/A	N/A	N/A	
GetHLSStreamingSessionURL	25 TPS [h]	N/A	N/A	N/A	

API	Stream-level begrenzen	Connection-level begrenzen	Bandbreitenbegrenzung	Fragment-level begrenzen	Relevante Ausnahmen und Hinweise
GetImages	N/A	N/A	100 MB [h]	N/A	Die maximale Anzahl von Bildern pro Anfrage beträgt 100 [h].
					 Note Der Mindestwert für <code>SamplingInterval</code> ist 200 Millisekunden (ms), was 5 Bildern pro Sekunde entspricht.

API	Stream-level begrenzen	Connection-level begrenzen	Bandbreitenbegrenzung	Fragment-level begrenzen	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. <code>ConnectionLimitExceededException</code> Wenn ein typisches Fragment etwa 5 MB groß ist, bedeutet dieser Grenzwert ~75 MB/s pro Kinesis-Videostream. Ein solcher Stream hat eine ausgehende Bitrate, die etwa doppelt so hoch ist wie die maximale eingehende Bitrate des Streams.

API	Stream-level begrenzen	Connection-level begrenzen	Bandbreitenbegrenzung	Fragment-level begrenzen	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	Session-level Limit	Fragment-level begrenzen
GetDASHManifestPlaylist	5 TPS [h]	Maximale Anzahl von Fragmenten pro Playlist: 5.000 [h]
GetHLSMasterPlaylist	5 TPS [h]	N/A
GetHLSMediaPlaylist	5 TPS [h]	Maximale Anzahl von Fragmenten pro Playlist: 5.000 [h]
GetMP4InitFragment	5 TPS [h]	N/A

API	Session-level Limit	Fragment-level begrenzen
GetMP4MediaFragment	20 TPS [h]	N/A
GetTSFragment	20 TPS [h]	N/A

Fragment-metadata und Kontingente für fragmentierte Medien

Kinesis Video Streams [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 APIs gemeinsam genutzt. Das bedeutet, dass für einen bestimmten Stream Anfragen nach Fragmenten über verschiedene APIs aus demselben Kontingent abgerufen werden. Wenn entweder das Fragmentmetadaten- oder das Fragmentmedienkontingent für einen Stream überschritten wird, gibt die API eine `ClientLimitExceededException` zurück. In den folgenden Tabellen wird dargestellt, wie die APIs die einzelnen Kontingenttypen verbrauchen. Gehen Sie für die zweite Spalte in diesen Tabellen davon aus, dass, wenn ein Stream ein Kontingent von N hat, das bedeutet, dass die APIs über N Punkte verfügen, die von diesem Quotentyp für diesen Stream verbraucht werden können. Die `GetClip` API erscheint in beiden Tabellen.

Fragment-metadata Quotenverbrauch

API	Anzahl der pro Anfrage verbrauchten Kontingentpunkte	Geteiltes Kontingent (N)
ListFragments	Wert des <code>MaxResults</code> Parameters	10.000 Quotenpunkte pro Sekunde, pro Stream [h]
GetClip	Anzahl der Fragmente im resultierenden Clip	

API	Anzahl der pro Anfrage verbrauchten Kontingen- tpunkte	Geteiltes Kontingent (N)
GetHLSMediaPlaylist	Wert des MaxMediaP laylistFragmentRes ults Parameters	
GetDASHManifest	Wert des MaxManife stFragmentResults Parameters	
GetImages	Wert von 400 + maximale Anzahl angeforderter Bilder	

Fragment-media Kontingentverbrauch

API	Anzahl der pro Anfrage verbrauchten Kontingen- tpunkte	Geteiltes Kontingent (N)
GetMediaForFragmen- tList	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 angeforde- rter 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 anGetTSFragment.

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 Aufruf 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 Aufruf für das neueste Video und ein weiterer für das entsprechende Audio. Jeder Aufruf verbraucht ein einzelnes Fragment-Medientoken, sodass insgesamt zwei Token verbraucht werden.

In diesem Szenario können bis zu 250 gleichzeitige Wiedergabesitzungen unterstützt werden. Bei 250 Sitzungen verbraucht dieses Szenario 2.500 Fragmentmetadaten-Kontingentpunkte pro Sekunde (deutlich unter dem Kontingent von 10.000) und 500 Fragmentmedienkontingentpunkte pro Sekunde.

Beispiel 2: Wiedergabe On-demand

In einem On-Demand-Wiedergabe-Szenario eines vergangenen Ereignisses mit einer Audio- und Videospur MPEG-DASH, die auf 1.000 MaxManifestFragmentResults eingestellt ist, ruft ein Media Player in der Regel zu Beginn der Sitzung GetDASHManifest einmal auf (verbraucht 1.000 Fragmentmetadaten-Kontingentpunkte) 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 `verwendenArchivedFragmentsConsumed.Media`, um den Verbrauch von Fragment-Metadaten bzw. Quotenpunkten für Fragmentmedien zu überwachen. Hinweise zur Überwachung finden Sie unter [the 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 garantiert, dass der Producer nicht versehentlich rekursiv Streams erstellt.
Max device name length (Max. Länge des Gerätenamens)	128 Zeichen	
Max tag count (Max. Tag-Anzahl)	50 pro Stream	
Max stream name length (Max. Länge des Stream-Namens)	256 Zeichen	
Min storage size (Min. Speichergröße)	10 MiB = 10 x 1024 x 1024 Byte	
Max storage size (Max. Speichergröße)	10 GiB = 10 x 1024 x 1024 x 1024 Byte	
Max root directory path length (Max. Länge des Stammverzeichnispfades)	4,096 Zeichen	
Max auth info length (Max. Länge der Authentifizierungsinformationen)	10,000 Bytes	
Max URI string length (Max. Länge der URI-Zeichenfolge)	10,000 Zeichen	
Max tag name length (Max. Länge des Tag-Namens)	128 Zeichen	

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	

Amazon Kinesis Video Streams, mehrstufiger Speicher

Mit Amazon Kinesis Video Streams Tiered Storage können Sie die Kosten für die Speicherung und Nutzung von Medien auf der Grundlage des erwarteten Nutzungsbedarfs, der Datenzugriffshäufigkeit

und der Aufbewahrungsdauer optimieren. KVS bietet zwei Speicherstufen, um Ihnen zu helfen, Kosten- und Leistungsanforderungen in Einklang zu bringen.

Themen

- [-Übersicht](#)
- [Die wichtigsten Konzepte](#)
- [Tiered Storage einrichten](#)
- [Bewährte Methoden](#)

-Übersicht

Kinesis Video Streams bietet 2 Speicherebenen:

- Kinesis Video Streams Standard (Hot) Tier — Bietet sofortigen Zugriff mit niedrigster Latenz auf Medien, auf die häufig zugegriffen wird.
- Kinesis Video Streams Warm Tier — Cost-effective für Anwendungsfälle, in denen Medien für einen längeren Zeitraum gespeichert werden und seltener abgerufen werden, bei Bedarf aber schneller Zugriff erforderlich ist.

Die wichtigsten Konzepte

Wenn Sie diese Konzepte verstehen, können Sie optimale Speicherrichtlinien konfigurieren:

- Speicherkosten — Die Warm-Stufe ist günstiger als die Hot-Stufe.
- Aufnahmekosten — Die Warm-Stufe berechnet pro 1000 Fragmente, während die Hot-Stufe die Gebühren pro 1 GB aufgenommener Medien berechnet.
- Mindestspeicherdauer — Warm Tier-Speicher wird für eine Mindestspeicherdauer von 30 Tagen in Rechnung gestellt. Wenn Medien vor Ablauf dieses Zeitraums von 30 Tagen gelöscht oder verschoben werden, fallen für Sie die normale Speichernutzungsgebühr sowie eine anteilige Gebühr für den Rest der Mindestspeicherdauer an. Für Medien, die länger als die Mindestspeicherdauer gespeichert werden, fällt keine Mindestgebühr an.
- Latenz — Leichter Anstieg (Millisekunden) bei Aufnahme und Verbrauch der Medien, wenn Medien in der Warm-Stufe gespeichert werden.

Tiered Storage einrichten

Erstellen Sie einen neuen Stream mit Warm Tier Ingestion

1. Öffnen Sie die Konsole unter. <https://console.aws.amazon.com/kinesisvideo/home>
2. Wählen Sie auf der Seite Video streams (Videostreams) die Option Create video stream (Videostream erstellen) aus.
3. Geben Sie auf der Seite Neuen Videostream erstellen *YourStreamName* den Namen des Streams ein.
4. Wählen Sie Benutzerdefinierte Konfiguration. Stellen Sie eine Datenspeicherung ein (z. B. *31 days*). Stellen Sie die Speicherstufe ein als *Warm tier*
5. Wählen Sie Create video stream (Videostream erstellen).
6. Nachdem Amazon Kinesis Video Streams den Stream erstellt hat, überprüfen Sie die Details auf der *YourStreamName* Seite.

Ihr Stream nimmt nun Medien direkt in das Warm Tier auf und bietet sofortigen Zugriff mit niedrigster Latenz.

Bewährte Methoden

- Analysieren Sie Ihre Medienzugriffsmuster, bevor Sie Speicherebenen auswählen
- Verwenden Sie Hot-Tier für Medien, auf die häufig zugegriffen wird und auf die sofort zugegriffen werden muss
- Konfigurieren Sie die Warm-Stufe für Medien, auf die gelegentlich mit akzeptabler geringer Latenz zugegriffen wird
- Berücksichtigen Sie die Gesamtbetriebskosten einschließlich Speicher- und Abrufkosten

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 AWS-Konto](#)

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

Melden Sie sich an für AWS-Konto

Um damit zu beginnen AWS, benötigen Sie eine AWS-Konto. Informationen zum Erstellen einer AWS-Konto finden Sie AWS-Konto [im AWS -Kontenverwaltung Referenzhandbuch unter](#) Erste Schritte mit einer.

Erstellen Sie einen AWS-Konto Schlüssel

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-Managementkonsole 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 die folgenden Vorteile:

- Millionen von Geräten verbinden und von dort streamen — Mit Kinesis Video Streams können Sie Video-, Audio- und andere Daten von Millionen von Geräten verbinden und streamen — von Smartphones für Privatanwender über Drohnen bis hin zu Dashcams. Sie können die Kinesis Video Streams Producer-Bibliotheken verwenden, um Ihre Geräte zu konfigurieren und zuverlässig in Echtzeit oder als nachträgliche Medien-Uploads zu streamen.
- Daten dauerhaft speichern, verschlüsseln und indexieren — 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, der auf vom Hersteller oder vom Dienst generierten Zeitstempeln basiert. Ihre Anwendungen können mithilfe des Zeitindexes 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 nutzenden Anwendungen wachsen und schrumpfen. Kinesis Video Streams übernimmt automatisch die gesamte Verwaltung und Wartung, die für die Verwaltung von Streams erforderlich ist, sodass Sie sich auf die Anwendungen und nicht auf die Infrastruktur konzentrieren können.
- Erstellen Sie Echtzeit- und Batch-Anwendungen auf Datenströmen — Sie können Kinesis Video Streams verwenden, um benutzerdefinierte Echtzeitanwendungen zu erstellen, die mit Live-Datenströmen arbeiten, und 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 oder Drittanbieter-Lösungen, mit denen Sie Ihre Streams verarbeiten und analysieren. AWS Marketplace Sie können Kinesis Video Get Streams-APIs verwenden, um mehrere Anwendungen gleichzeitig zu erstellen, die Daten in Echtzeit oder stapelweise verarbeiten.
- Daten sicherer streamen — Kinesis Video Streams verschlüsselt alle Daten, während sie den Dienst durchlaufen und wenn die Daten gespeichert werden. Kinesis Video Streams erzwingt beim Streamen von Daten von Geräten eine auf Transport Layer Security (TLS) basierende Verschlüsselung 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.
- Nutzungsbasierte Bezahlung — Weitere Informationen finden Sie unter [AWS -Preisrechner](#).

Erste Schritte mit Amazon Kinesis Video Streams

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

- Richten Sie Ihr AWS-Konto ein und erstellen Sie einen Administrator, 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, zuerst zu lesen [Kinesis Video Streams: So funktioniert es](#).

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 — Preise](#).

Topics

- [Erstellen Sie einen Amazon Kinesis-Videostream](#)
- [Senden Sie Daten an einen Amazon Kinesis-Videostream](#)
- [Mediendaten konsumieren](#)

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

Erstellen Sie einen Videostream mit der Konsole

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

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"
}
```

Senden Sie Daten an einen Amazon Kinesis-Videostream

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: Das SDK, das GStreamer-Plugin des Herstellers von Kinesis Video Streams — kvssink](#) Plug-in verwendet.

Um Medien von einer Vielzahl von Geräten mit unterschiedlichen Betriebssystemen zu senden, verwendet dieses Tutorial die Kinesis Video Streams C++ Producer Library und [GStreamer](#),

ein Open-Source-Medienframework, das den Zugriff auf Kameras und andere Medienquellen standardisiert.

Themen

- [Erstellen Sie das SDK und die Beispiele](#)
- [Führen Sie die Samples aus, um Medien in 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 stehenden Anweisungen.

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 Projektarchiv klonen
- Verwenden von CMake zum Generieren von Makefiles
- Erstellen der Binärdateien mit make

Build in AWS Cloud9

Gehen Sie zum Hochladen auf Kinesis Video Streams in AWS Cloud9 wie folgt vor. Sie müssen nichts auf Ihren Computer herunterladen.

1. In der AWS-Managementkonsole, öffnen [AWS Cloud9](#).

Wählen Sie Umgebung erstellen aus.

2. Gehen Sie auf dem Bildschirm „Umgebung erstellen“ wie folgt vor:
 - Name — Geben Sie einen Namen für Ihre neue Umgebung ein.
 - Plattform — Wählen Sie Ubuntu Server 22.04 LTS.

Sie können die anderen Felder mit den Standardauswahlen belassen.

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 zu der gesamten Software ab, die Sie mit dem Advanced Packaging Tool (APT) installieren oder aktualisieren können.

Dieser Befehl aktualisiert nicht die Software selbst, stellt jedoch sicher, dass Ihr System die neuesten verfügbaren Versionen kennt.

```
sudo apt-get update
```

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

```
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++-Produzenten-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. Verwenden Sie CMake, um Makefiles zu generieren.

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

Das Ende der erwarteten Ausgabe sieht wie folgt aus:

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

9. Verwenden Sie make, um das SDK und die Beispielanwendungen zu kompilieren sowie 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 die Einstellung der Umgebungsvariablen GST_PLUGIN_PATH zum Startskript Ihrer Shell 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
```

Bestätigen Sie, dass GST_PLUGIN_PATH gesetzt ist.

```
echo $GST_PLUGIN_PATH
```

Wenn Sie die Ausgabe richtig einstellen, 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 Samples aus, um Medien in 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
```

Technische Daten des Videos:

- 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 GOP-Größe) 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 unter [the section called “Erstellen Sie einen Amazon Kinesis-Videostream”](#).

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

Sample video from GStreamer

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

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

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

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

Diese GStreamer-Pipeline generiert einen Live-Testvideostream mit einem Standardtestmuster, das mit 10 Bildern pro Sekunde und einer Auflösung von 640x480 Pixeln ausgeführt wird. 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 GOP-Größe) von 1 Sekunde führt. `kvssink` nimmt den H.264-encoded Videostream, packt ihn in das Matroska-Containerformat (MKV) 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 aus einem RTSP-Stream von einer Kamera oder von einer USB-Kamera finden Sie unter.

[Beispiel: Das SDK, das GStreamer-Plugin des Herstellers von Kinesis Video Streams — 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 sehen. Ein Beispiel sieht wie folgt aus:

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

Wenn die Bestätigung zutrifft, bedeutet `EventType` `PERSISTED`, dass Kinesis Video Streams diesen Medienblock dauerhaft gespeichert und verschlüsselt hat, damit er abgerufen, analysiert und langfristig gespeichert werden kann.

Weitere Informationen zu Bestätigungen finden Sie unter [PutMedia](#).

Mediendaten konsumieren

Sie können Mediendaten konsumieren, indem Sie sie entweder in der Konsole anzeigen oder 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 das AWS-Managementkonsole. Wählen Sie im Kinesis Video 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 verbraucht. 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
- Nach dem Puffern für ein paar Sekunden
- 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 Infrarotkameras, Radargeräten 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 `registrierenKinesisVideoClient`, 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 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 und Mediendaten von einem Gerät an einen Kinesis-Videostream zu senden.

Gehen Sie wie folgt vor, 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 verifizieren Sie ihn](#)

Voraussetzungen

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

- Im Beispielcode geben Sie Anmeldeinformationen ein, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei einrichten. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein. Weitere Informationen finden Sie unter [AWS Anmeldeinformationen und Region für Entwicklung einrichten](#) im 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, verfügbar unter <https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp>. Die Namensweiterung für diese Datei hängt von Ihrem Betriebssystem ab:
 - `KinesisVideoProducerJNI.so` für Linux:
 - `KinesisVideoProducerJNI.dylib` für macOS
 - `KinesisVideoProducerJNI.dll` für Windows

Note

Pre-built 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 Java Producer-Bibliothekscod 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 Speicherorte der Bibliothek.

Voraussetzungen und weitere Details zu diesem Beispiel finden Sie unter [Verwenden der Java-Producer-Bibliothek](#).

1. Erstellen Sie ein Verzeichnis und klonen Sie dann den Beispielquellcode aus dem GitHub Repository.

```
git clone https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-java
```

2. Öffnen Sie die integrierte Java-Entwicklungsumgebung (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.

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

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

Registrieren 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 diese an den Client senden kann.

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

Bereinigen von Ressourcen

Um Speicherlecks zu vermeiden, gehen Sie wie folgt vor, um die Registrierung einer Medienquelle beim 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 mit dem:

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

Leere den Cache, wenn du fertig bist:

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

Führen Sie den Code aus und überprüfen Sie ihn

Gehen Sie wie folgt vor, um das [Java-Test-Harness für die Java Producer-Bibliothek](#) auszuführen.

1. Wählen Sie DemoAppMain.
2. Wählen Sie Ausführen, Ausführen 'DemoAppMain'.

3. Fügen Sie den JVM-Argumenten für die Anwendung Ihre Anmeldeinformationen hinzu:
 - Für nicht temporäre AWS Anmeldeinformationen: "-
Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey}
-Djava.library.path={NativeLibraryPath}"
 - Für temporäre AWS Anmeldeinformationen: "-
Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey}
-Daws.sessionToken={YourAwsSessionToken} -
Djava.library.path={NativeLibraryPath}"
4. Melden Sie sich bei der [Kinesis Video Streams-Konsole](#) an AWS-Managementkonsole und öffnen Sie sie.

Wählen Sie auf der Seite Manage Streams den entsprechenden Stream aus.

5. Das Beispiellvideo 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 dann die Daten 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.

Gehen Sie wie folgt vor, 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.

Vorgehensweise: 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 Anwendungscode. Wir empfehlen, die neueste stabile Version zu verwenden.

Im Beispielcode geben Sie Amazon Cognito-Anmeldeinformationen an.

Gehen Sie wie folgt vor, um einen Amazon Cognito-Benutzerpool und einen 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-Konsole an](#) und vergewissern Sie sich, dass die Region korrekt ist.
2. Wählen Sie in der linken Navigationsleiste **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 IAM-Rollenauswahlfeld 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 Client aus.

Klicken Sie auf Weiter.

- f. Schritt 6: Prüfen und erstellen — Prüfen Sie Ihre Auswahl aus den vorherigen Abschnitten und wählen Sie dann Benutzerpool erstellen aus.

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

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

- 6. Wählen Sie den Tab App-Integration und gehen Sie zum Ende der Seite.

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

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

- 8. Zeigen Sie das Client-Geheimnis an und notieren Sie es sich für später. In der `awsconfiguration.json` Datei ist `dasCognitoUserPool.Default.AppClientSecret`.

Richten Sie einen Identitätspool ein

Einrichten eines Identitäten-Pools

1. Melden Sie sich bei der [Amazon Cognito-Konsole an](#) und vergewissern Sie sich, dass die Region korrekt ist.
2. Wählen Sie in der linken Navigationsleiste **Identitätspools** aus.
3. Wählen Sie **Identitätspool erstellen**.
4. Konfigurieren Sie den Identitätspool.
 - a. Schritt 1: Vertrauensstellung im Identitätspool konfigurieren — Füllen Sie die folgenden Abschnitte aus:
 - Benutzerzugriff — Wählen Sie **Authentifizierter Zugriff**
 - Authentifizierte Identitätsquellen — Wählen Sie den **Amazon 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 verbinden — Füllen Sie im Abschnitt **Benutzerpool-Details** die folgenden Felder aus:
 - Benutzerpool-ID — Wählen Sie den Benutzerpool aus, den Sie zuvor erstellt haben.
 - App-Client-ID — Wählen Sie die App-Client-ID aus, die Sie zuvor erstellt haben.

Klicken Sie auf **Weiter**.

- d. Schritt 4: Eigenschaften konfigurieren — Geben Sie einen Namen in das **Feld für den Identitätspoolnamen** ein.

Klicken Sie auf **Weiter**.

- e. Schritt 5: Prüfen und erstellen — Prüfen 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 diese für später. In der `awsconfiguration.json` Datei ist `dasCredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Aktualisieren Sie die Berechtigungen für die IAM-Rolle.
 - a. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die IAM-Konsole unter. <https://console.aws.amazon.com/iam/>
 - b. Wählen Sie in der linken Navigationsleiste 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 angehängte 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. Wählen Sie das Kästchen neben Diese neue Version als Standard festlegen aus, falls sie nicht bereits ausgewählt ist.

Wählen Sie Änderungen speichern aus.

Laden Sie den Android Producer-Bibliothekscod herunter und konfigurieren Sie ihn

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

Voraussetzungen und weitere Details zu diesem Beispiel finden Sie unter [Verwenden der Android Producer-Bibliothek](#).

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

```
git clone https://github.com/aws-labs/aws-sdk-android-samples
```

2. Öffnen Sie [Android Studio](#).
3. Wählen Sie auf dem Öffnungsbildschirm Open an existing Android Studio project.
4. Navigieren Sie zum Verzeichnis aws-sdk-android-samples/
AmazonKinesisVideoDemoApp und wählen Sie OK.
5. Öffnen Sie die AmazonKinesisVideoDemoApp/src/main/res/raw/
awsconfiguration.json Datei.

Geben Sie im CredentialsProvider Knoten die Identitätspool-ID aus dem Abschnitt „So richten Sie einen Identitätspool ein“ im [Abschnitt „Voraussetzungen“](#) ein, und geben Sie Ihre AWS-Region (z. B. **us-west-2**) an.

Geben Sie im CognitoUserPool Knoten den geheimen App-Client-Schlüssel, die App-Client-ID und die Pool-ID aus dem [Abschnitt So richten Sie einen Benutzerpool](#) 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": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnopqrstuvwxyz",
      "AppClientId": "0123456789abcdefghijklmnopqrstuvwxyz",
      "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` festgelegt):

```
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 Regions. <https://aws-amplify.github.io/aws-sdk-android/docs/reference/com/amazonaws/regions/Regions.html>

Untersuchen Sie den Code

In diesem Abschnitt der [Android Producer-Bibliothekspozedur](#) 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 den `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-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.

Beispielsweise stellt die Kinesis Video Streams-Android-Bibliothek die [AndroidCameraMediaSource](#) Implementierung der `MediaSource` Schnittstelle bereit. Dieser Kurs liest Daten aus einer der Kameras des Speichergeräts.

Im folgenden Codebeispiel (aus der [fragment/StreamConfigurationFragment.java](#)-Datei) wird die Konfiguration für die Medien-Quelle erstellt:

```
private AndroidCameraMediaSourceConfiguration getCurrentConfiguration() {  
    return new AndroidCameraMediaSourceConfiguration(  

```

```

        AndroidCameraMediaSourceConfiguration.builder()
            .withCameraId(mCamerasDropdown.getSelectedItem().getCameraId())

        .withEncodingMimeType(mMimeTypeDropdown.getSelectedItem().getMimeType())

        .withHorizontalResolution(mResolutionDropdown.getSelectedItem().getWidth())

        .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
            .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
            .withIsEncoderHardwareAccelerated(

mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
            .withFrameRate(FRAME_RATE_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 AWSKinesisVideoDemoApp Anwendung auf dem Gerät die Option Neues Konto erstellen aus.
7. Geben Sie Werte für USERNAME, Password, Given name, Email address und Phone number ein und wählen Sie Sign up.

 Note

Für diese Werte gelten die folgenden Einschränkungen:

- Passwort: Muss eine Kombination aus Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen enthalten. Sie können diese Einschränkungen auf Ihrer Benutzerpool-Seite in der [Amazon Cognito-Konsole](#) ändern.
- E-Mail-Adresse: Muss eine gültige Adresse sein, damit Sie einen Bestätigungscode erhalten können.
- Telefonnummer: Muss das folgende Format haben: **+<Country code><Number>**, z. B. **+12065551212**.

8. Geben Sie den Code ein, den Sie per E-Mail erhalten, und wählen Sie Bestätigen. 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-Konsole](#) in der Region USA West (Oregon) an AWS-Managementkonsole und öffnen Sie die Kinesis Video Streams-Konsole.

Wählen Sie auf der Seite Manage Streams den Eintrag demo-stream.

11. Das Streaming-Video wird im eingebetteten Player wiedergegeben. Sie müssen möglicherweise kurz warten (bis zu zehn Sekunden unter normalen Bandbreiten- und Prozessorbedingungen), während die Frames angesammelt werden, bevor das Video angezeigt wird.

 Note

Wenn sich der Bildschirm des Geräts dreht (z. B. von Hochformat zu Querformat), stoppt die Anwendung das Streamen von Videos.

Das Codebeispiel erstellt einen Stream. Sobald das `MediaSource`-Objekt im Code gestartet wird, beginnt es mit dem Senden von Beispiel-Frames von der Kamera an das `KinesisVideoClient`-Objekt. Der Client sendet die Daten dann an einen Kinesis-Videostream mit dem Namen `demo-stream`.

Verwenden Sie die C++ Producer-Bibliothek

Sie können die von Amazon Kinesis Video Streams bereitgestellte C++-Produzentenbibliothek verwenden, um Anwendungscode zum Senden von Mediendaten von einem Gerät an einen Kinesis-Videostream zu schreiben.

Objektmodell

Die C++-Bibliothek bietet die folgenden Objekte zur Verwaltung des Sendens von Daten an einen Kinesis-Videostream:

- `KinesisVideoProducer`: Enthält Informationen über Ihre Medienquelle und AWS Anmeldeinformationen und verwaltet Callbacks, um über Kinesis Video Streams-Ereignisse zu berichten.
- `KinesisVideoStream`: Stellt den Kinesis-Videostream dar. Enthält Informationen zu den Parametern des Videostreams, wie Name, Aufbewahrungsdauer und Art des Medieninhalts.

Fügen Sie Medien in den Stream ein

Sie können die in der C++-Bibliothek bereitgestellten Methoden (z. B. `PutFrame`) verwenden, 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:

- Service-Callback-Schnittstelle (**CallbackProvider**): Die Bibliothek ruft Ereignisse auf, die über diese Schnittstelle abgerufen wurden, wenn sie einen Stream erstellt, eine Stream-Beschreibung abrufen und einen Stream löscht.
- Client-ready state or low storage events interface (**ClientCallbackProvider**): Die Bibliothek ruft Ereignisse auf dieser Schnittstelle auf, wenn der Client bereit ist oder wenn sie feststellt, dass ihm möglicherweise der verfügbare Speicher oder Arbeitsspeicher ausgeht.
- 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 übergeht, verworfene Frames oder Stream-Fehler 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 die Benutzeroberfläche mit einem niedrigen Speicherbedarf versehen möchten.

Weitere Informationen zu Callbacks in den Producer-Bibliotheken finden Sie unter [SDK-Rückrufe von Produzenten](#)

Vorgehensweise: Verwenden Sie das C++ Producer SDK

Dieses Verfahren zeigt, wie Sie den Kinesis Video 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 ein, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei einrichten. Falls das noch nicht geschehen ist, richten Sie zunächst das Anmeldeinformationsprofil ein.

Weitere Informationen finden Sie unter [AWS Anmeldeinformationen und Region für Entwicklung einrichten](#).

- **Integration des Zertifikatsspeichers:** Die Kinesis Video Streams Producer Library muss dem Dienst, den sie aufruft, eine Vertrauensstellung aufbauen. Dies erfolgt durch die Validierung der Zertifizierungsstellen (CAs) im öffentlichen Zertifikatsspeicher. Bei Linux-based Modellen befindet sich dieser Speicher im Verzeichnis `/etc/ssl/`.

Laden Sie das Zertifikat aus dem folgenden Speicherort in den Zertifikatsspeicher:

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installieren Sie die folgenden Build-Abhängigkeiten für macOS:
 - [Autoconf 2.69](#) (Lizenz GPLv3+/Autoconf: GNU GPL Version 3 oder höher)
 - [CMake 3.7 oder 3.8](#)
 - [Pkg-Config](#)
 - 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`
 - JDK öffnen: `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 C++ Producer-Bibliothekscode 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

Unten finden Sie eine Referenztabelle für die SDK-specific CMake-Argumente von C++ Producer. Sie können auch die [Standard-CMake-Optionen](#) 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 dem Quellcode. 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	Erstellt das kvssink GStreamer-Plugin.
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 Speicherzuweisungen an 8-Byte-Grenzen ausgerichtet werden sollen. Einige Architekturen erlauben keinen unausgerichteten Speicherzugriff.
CONSTRAINED_DEVICE	Boolesch	OFF	Non-Windows nur. Wenn ON, wird die Thread-Stack-Größe auf festgelegt 0.5 MiB. 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 (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 entwickelt wurde.
OPEN_SRC_INSTALL_PREFIX	Zeichenfolge	../öffnen- source/local	Ort, an dem die Open-Source-Abhängigkeiten installiert werden, wenn sie aus dem Quellcode erstellt werden.

Flaggen für die Kreuzkompilierung

Important

Setzen Sie diese, wenn die CPU-Architekturen Ihres Ziel- und Hostcomputers unterschiedlich sind.

CMake-Argument	Typ	Standard	Erklärung
BUILD_LOG 4CPLUS_HOST	Zeichenfolge	""	Erstellen Sie die log4cp1us Abhängigkeit für die angegebene CPU-Architektur. Wenn nicht gesetzt, log4cp1us wird die CPU-Architektur des Host-Computers automatisch erkannt und verwendet.
BUILD_OPE NSSL_PLATFORM	Zeichenfolge	""	Erstellen Sie die OpenSSL Abhängigkeit für die angegebene CPU-Architektur. Wenn nicht gesetzt, OpenSSL wird die CPU-Architektur des Host-Computers automatisch erkannt und verwendet.

Flaggen im Zusammenhang mit dem Testen

CMake-Argument	Typ	Standard	Erklärung
BUILD_TEST	Boolesch	OFF	Erstellen Sie die Einheit und die Integrationstests. Um alle Tests auszuführen, führen Sie sie ./tst/producerTest aus

CMake-Argument	Typ	Standard	Erklärung
			dem Build-Verzeichnis aus. AWS Für die Ausführung der Tests sind Anmeldeinformationen erforderlich.
CODE_COVERAGE	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Aktivieren Sie die Erfassung der Codeabdeckung mit gcov und die Berichtsgenerierung.
COMPILER_WARNINGS	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Aktiviert alle Compiler-Warnungen.
ADDRESS_SANITIZER	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Bauen Sie mit AddressSanitizer .
MEMORY_SANITIZER	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Bauen Sie mit MemorySanitizer .
THREAD_SANITIZER	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Bauen Sie mit ThreadSanitizer .

CMake-Argument	Typ	Standard	Erklärung
UNDEFINED _BEHAVIOR _SANITIZER	Boolesch	OFF	Nur für GNU/Clang Compiler verfügbar . Bauen Sie mit UndefinedBehaviorSanitizer .

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

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

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

Schreiben und untersuchen Sie den Code

In diesem Abschnitt von untersuchen Sie den Code im C++-Testsystem (`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 existiert.
- Rufen Sie `putFrame` im `KinesisVideoStream` für jeden Daten-Frame auf, sobald er verfügbar wird, um den Frame an den Stream zu senden.

Die folgenden Abschnitte enthalten weitere Informationen zu diesem Codierungsmuster.

Erstellen 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 $3 \text{ (Streams)} * 3 \text{ (Minuten)} * 60 \text{ (Sekunden in einer Minute)} * 10000 \text{ (KB)} * 1,2 \text{ (Defragmentierungszulage)} = 194,4 \text{ MB} \sim 200 \text{ MB}$.

- Ein `ClientCallbackProvider`-Objekt, das Funktionszeiger zurückgibt, die Client-spezifische Ereignisse melden.

- Ein `StreamCallbackProvider`-Objekt, das Funktionszeiger zurückgibt, die per Callback aufgerufen werden, wenn Stream-spezifische Ereignisse auftreten.
- Ein `CredentialProvider` Objekt, das den Zugriff auf Umgebungsvariablen mit AWS Anmeldeinformationen ermöglicht.
- Das 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 Konsumenten Anwendungen verwendet werden, um den richtigen Stream zu finden oder weitere Informationen über den Stream abzurufen. Die Tags können auch in der AWS-Managementkonsole 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 Videospur-Stream-Definition Audiospurdetails 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);
```

Für die `addTrack` Methode sind die folgenden Parameter erforderlich:

- Track-ID (wie eine für Audio). Dies sollte ein eindeutiger Wert ungleich Null sein.
- User-defined Titelname (z. B. „Audio“ für die Audiospur).
- Codec-ID für diesen Titel (z. B. für die Audiospur „A_AAC“).
- Spurtyp (verwenden Sie beispielsweise den Enum-Wert von `MKV_TRACK_INFO_TYPE_AUDIO` für Audio).

Wenn Sie private Codec-Daten für die Audiospur haben, können Sie sie beim Aufrufen der `addTrack`-Funktion übergeben. Sie können dem Codec auch private Daten senden, nachdem Sie das Objekt erstellt und gleichzeitig die Startmethode aufgerufen haben. `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 a KinesisVideoFrame in einen bestimmten Titel ein 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. Anschließend können Sie den aufrufen, `putFrameMultiTrack` um `KinesisVideoFrame` zusammen mit dem booleschen Wert den Typ der Frame-Daten anzugeben. 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 steigende Frame-Zeitstempel beibehalten und sich keine zwei Fragmente überschneiden. Beispielsweise sollte der MKV-Zeitstempel des ersten Frames eines Fragments immer größer sein als der MKV-Zeitstempel des letzten Frames des vorherigen Fragments.

Der `PutFrameHelper` hat die folgenden Felder:

- Maximale Anzahl von Audioframes in der Warteschlange.
- Maximale Anzahl von Videobildern in der Warteschlange.
- Größe, die einem einzelnen Audioframe zugewiesen werden soll.
- Größe, die einem einzelnen Videobild zugewiesen werden soll.

Greifen Sie auf Messwerte und Messwertprotokollierung

Das C++ Producer SDK umfasst Funktionen für Metriken und Metrikprotokollierung.

Sie können die Operationen `getKinesisVideoMetrics` und die `getKinesisVideoStreamMetrics` API 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:

- `InhaltStoreSize`: Die Gesamtgröße des Inhaltsspeichers in Byte (der Speicher, der zum Speichern von Streaming-Daten verwendet wird).
- `InhaltStoreAvailableSize`: Der verfügbare Speicher im Inhaltsspeicher in Byte.
- `InhaltStoreAllocatedSize`: Der zugewiesene Speicher im Inhaltsspeicher.
- `totalContentViewsSize`: Der gesamte Speicher, der für die Inhaltsansicht verwendet wird. Die Inhaltsansicht besteht aus einer Reihe von Informationsindizes im Inhaltsspeicher.
- `SummeFrameRate`: Die Gesamtzahl der Bilder 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:

- `aktuellViewDuration`: Die Differenz 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).
- `InsgesamtViewDuration`: Der Unterschied in 100-ns-Einheiten zwischen dem Anfang der Inhaltsansicht (wenn Frames codiert sind) und dem Ende (wenn Frames aus dem Speicher geleert werden, entweder weil der gesamte zugewiesene Speicherplatz für die Inhaltsansicht überschritten wird oder weil eine `PersistedAck` Nachricht von Kinesis Video Streams empfangen wird und Bilder, von denen bekannt ist, dass sie persistent sind, geleert werden).

- `aktuellViewSize`: Die Größe der Inhaltsansicht in Byte vom Kopf (wenn Frames codiert sind) bis zur aktuellen Position (wenn Frames an Kinesis Video Streams gesendet werden).
- `InsgesamtViewSize`: Die Gesamtgröße der Inhaltsansicht in Byte.
- `aktuellFrameRate`: Die zuletzt gemessene Rate des Streams in Bildern pro Sekunde.
- `aktuellTransferRate`: 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

Lesen Sie die folgenden OS-specific Anweisungen [the section called “C++”](#), um den Code für auszuführen und zu überprüfen:

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [Raspberry Pi OS](#)

Sie können den Traffic in Ihrem Stream überwachen, indem Sie sich die Kennzahlen ansehen, die Ihrem Stream in der CloudWatch Amazon-Konsole zugeordnet sind, z. `PutMedia.IncomingBytes` B.

Verwenden Sie das C++ Producer SDK als GStreamer-Plugin

[GStreamer](#) ist ein beliebtes Medienframework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medienpipelines zu erstellen, indem modulare Plugins kombiniert werden. Das Kinesis Video Streams GStreamer-Plugin optimiert die Integration Ihrer vorhandenen GStreamer-Medienpipeline mit Kinesis Video Streams.

Hinweise zur Verwendung des C++ Producer SDK als GStreamer-Plugin finden Sie unter. [Beispiel: Das SDK, das GStreamer-Plugin des Herstellers von Kinesis Video Streams — kvssink](#)

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

[GStreamer](#) ist ein beliebtes Medienframework, das von mehreren Kameras und Videoquellen verwendet wird, um benutzerdefinierte Medienpipelines zu erstellen, indem modulare Plugins kombiniert werden. Das Kinesis Video Streams GStreamer-Plugin optimiert die Integration Ihrer vorhandenen GStreamer-Medienpipeline mit Kinesis Video Streams.

Darüber hinaus standardisiert die Verwendung von [Docker](#) zur Erstellung der GStreamer-Pipeline die Betriebsumgebung für Kinesis Video Streams, wodurch die Erstellung und Ausführung der Anwendung optimiert wird.

Hinweise 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-Managementkonsole konfiguriert wird:

```
log4cplus.rootLogger=DEBUG, KvsConsoleAppender
```

Für eine weniger ausführliche Protokollierung können Sie als Protokollierungsebene INFO festlegen.

Um die Anwendung so zu konfigurieren, dass Protokolleinträge in eine Protokolldatei geschrieben werden, 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 unter GitHub verfügbar ist. <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/> 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-based API-Callbacks), Autorisierung (AWS Credentials-API) und Rückrufe zur Wiederholung von Streaming-Versuchen bei Fehlern. 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 Medieninhaltenstyp 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-encoded 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 AWS Anmeldeinformationsprofildatei 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-based Modellen befindet sich dieser Speicher im Verzeichnis/etc/ssl/.

Laden Sie das Zertifikat aus dem folgenden Speicherort in den Zertifikatsspeicher:

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- Installieren Sie die folgenden Build-Abhängigkeiten für macOS:
 - [Autoconf 2.69](#) (Lizenz GPLv3+/Autoconf: GNU GPL Version 3 oder höher)

- [CMake 3.7 oder 3.8](#)
- [Pkg-Config](#)
- 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/aws-labs/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`, `CMake` und eine Build-Umgebung installieren.

Weitere Informationen finden Sie unter `README.md` in <https://github.com/aws-labs/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 <https://github.com/awslabs/amazon-kinesis-video-streams-producer-c> Repositories. GitHub Sie haben diesen Code im vorherigen Schritt heruntergeladen. Dieses Beispiel zeigt, wie Sie die C-Producer-Bibliothek verwenden, um H.264-encoded 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 vor [the 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-encoded 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. Um ausführliche Protokolle zu aktivieren, definieren Sie das `HEAP_DEBUG` und, `LOG_STREAMING` C-defines indem Sie die entsprechenden Zeilen auskommentieren. `CMakeList.txt`

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 Raspberry Pi

Raspberry Pi ist ein kleiner, kostengünstiger Computer, mit dem Computergrundkenntnisse in der Programmierung beigebracht und erlernt werden können. Dieses Tutorial beschreibt, wie Sie das Amazon Kinesis Video Streams C++ Producer SDK auf einem Raspberry Pi-Gerät einrichten und verwenden können. Die Verfahrensschritte umfassen auch, wie Sie die Installation mithilfe der GStreamer-Demoanwendung überprüfen.

Themen

- [Voraussetzungen](#)
- [Erstellen Sie einen IAM-Benutzer mit der Berechtigung, in Kinesis Video Streams zu schreiben](#)
- [Verbinden Sie Ihren Raspberry Pi mit Ihrem Netzwerk Wi-Fi](#)
- [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 das 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

Wenn 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 Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT”](#) zur Konfiguration Ihres Geräts für die Verwendung der zertifikatsbasierten Authentifizierung finden Sie unter.

1. Melden Sie sich bei der an AWS-Managementkonsole 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 Set permissions for kinesis-video-raspberry-pi-producer (Berechtigungen für kinesis-video-raspberry-pi-producer festlegen) Attach existing policies directly (Bestehende Richtlinien direkt anfügen).

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 in Kinesis-Videostreams 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 Netzwerk Wi-Fi

Fahren Sie bei Anschluss eines Monitors und einer Tastatur mit [Konfigurieren Sie die Raspberry Pi-Kamera](#).

Diese Anweisungen sollen Ihnen helfen, Ihren Raspberry Pi im Headless-Modus einzurichten, d. h. ohne angeschlossene Tastatur, Monitor oder Netzkabel. 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 eine 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 psk Werte ssid und durch die Informationen für Ihr Wi-Fi 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 tritt Ihrem Wi-Fi Netzwerk bei und SSH ist 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 ausgeführt wird, d. h. ohne angeschlossene Tastatur, Monitor oder Netzkabel. 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 Wi-Fi Router Ihres Netzwerks haben, sehen Sie sich die angeschlossenen Wi-Fi 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 Wi-Fi Router Ihres Netzwerks haben, können Sie andere Software verwenden, um Geräte in Ihrem Netzwerk zu finden. [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

Folgen Sie diesen Schritten, um das [Raspberry Pi-Kameramodul so](#) zu konfigurieren, dass es Videos vom Gerät an einen Kinesis-Videostream sendet.

Note

Wenn Sie eine USB-Webcam verwenden, fahren Sie mit fort [the 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 Kamertreiber 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 verwenden Sie den folgenden Befehl, um die Datei mit dem nano Editor zu bearbeiten:

```
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. Verwenden Sie Strg+X, um den nano Editor zu speichern und zu beenden.
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-Kamera. 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 richtig 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` (veraltet) oder `libcamera` (modern). Der `libcamera` Stack wird jedoch für eine bessere Unterstützung und bessere Funktionen empfohlen. Gehen Sie wie folgt vor, um zu überprüfen, ob Ihr System auf dem neuesten Stand `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 Kameravorschau-Stream und zeigt ihn auf dem Bildschirm an.

```
libcamera-hello
```

Wenn Sie Probleme mit Ihrem Kameramodul haben, finden Sie in der [Raspberry Pi-Dokumentation Informationen](#) zur Fehlerbehebung.

Installieren Sie die Softwarevoraussetzungen

Das C++ Producer SDK erfordert, dass Sie die folgenden Softwarevoraussetzungen auf dem Raspberry Pi installieren.

1. Aktualisieren Sie die Paketliste und installieren Sie die Bibliotheken, die zum Erstellen des SDK erforderlich sind. 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. Stellen Sie sicher, dass Sie die erforderlichen Softwarekomponenten installiert haben. [the section called “Installieren Sie die Softwarevoraussetzungen”](#) Weitere Informationen zu diesen Schritten finden Sie unter.

1. Navigieren Sie zum Download-Verzeichnis. Öffnen Sie ein Terminal und wechseln Sie in Ihr bevorzugtes Download-Verzeichnis.

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 Branch (der `master` Branch) geklont, wodurch Größe und Zeit für den Download reduziert werden. Außerdem werden die heruntergeladenen Inhalte in einem Ordner mit dem Namen `kvs-producer-sdk-cpp` innerhalb des aktuellen Verzeichnisses abgelegt.

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
```

[CMake](#) verwendet die folgenden Optionen, um die entsprechenden Makefiles zu generieren:

- Verwenden Sie den Projektordner (`..`) als Quellverzeichnis.
- Verwendung des aktuellen Verzeichnisses (`.`) (`build/`) für die Build-Ausgabe.
- `-DBUILD_GSTREAMER_PLUGIN=ON` ermöglicht das Erstellen des GStreamer-Plugins `kvssink`.

- `-DBUILD_DEPENDENCIES=OFF` deaktiviert das Erstellen externer Abhängigkeiten aus der Quelle. Das Projekt findet und verwendet die externen Abhängigkeiten, die in einem vorherigen Schritt installiert wurden.
- `-DALIGNED_MEMORY_MODEL=ON` deaktiviert das Speichermodell ohne Ausrichtung. Unausgerichteter Speicherzugriff wird von bestimmten Raspberry Pi-Geräten nicht unterstützt.

Note

Eine vollständige Liste der CMake-Argumente finden Sie unter. [the 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 mithilfe des Makefile generierten `voncmake`.

```
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 `libgstkvssink.so` vorhanden ist.

Listet die Dateien im aktuellen Verzeichnis auf.

Aufforderung:

```
ls
```

Antwort:

<code>CMakeCache.txt</code>	<code>dependency</code>	<code>kvs_gstreamer_sample</code>
<code>CMakeFiles</code>	<code>kvs_gstreamer_audio_video_sample</code>	<code>kvssink_gstreamer_sample</code>
<code>Makefile</code>	<code>kvs_gstreamer_file_uploader_sample</code>	<code>libKinesisVideoProducer.so</code>
<code>cmake_install.cmake</code>	<code>kvs_gstreamer_multistream_sample</code>	<code>libgstkvssink.so</code>

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

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

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

Lassen Sie GStreamer laden: kvssink

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

Sie sollten einige Dokumentationen 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 das Setzen der GST_PLUGIN_PATH Umgebungsvariablen einzubeziehen. Dadurch GST_PLUGIN_PATH wird sichergestellt, dass es 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 ladenkvssink.

```
echo $GST_PLUGIN_PATH
```

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

Streamen Sie das 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 Plugin zu finden. kvssink Weitere Informationen finden Sie unter [the section called “Laden Sie das C++ Producer SDK herunter und erstellen Sie es”](#).

1. Legen Sie die Anmeldeinformationen und die Region fest.

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

Weitere Authentifizierungsmethoden finden Sie unter [the section called “Geben Sie Anmeldeinformationen für kvssink ein ”](#).

Note

Das C++ Producer SDK verwendet standardmäßig die Region USA West (Oregonus-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, setzen Sie die folgende Umgebungsvariable auf Ihre Region (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 640x480 Pixeln ausgeführt wird. Es wird ein Overlay hinzugefügt, das die aktuelle Systemzeit und das aktuelle Systemdatum anzeigt. Das Video wird dann in ein H.264 Format codiert und Keyframes werden höchstens alle 10 Bilder generiert, was zu einer Fragmentdauer (auch als Größe einer Gruppe von Bildern (GoP) bezeichnet) von 1 Sekunde resultiert. kvssink nimmt den H.264-encoded Videostream, verpackt ihn in das Matroska-Containerformat (MKV) 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-Pipeline für ein Beispielvideo sieht wie folgt aus:

USB web cam

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

```
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 GStreamer automatisch erkennen zu lassen, können Sie es mit einer bestimmten Geräte-ID verwenden `v4l2src`. 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 verwendenbcm2835-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.

Sample RTSP camera

In diesem Beispiel verwenden wir es, Gst-Rtsp-Server um einen Demo-RTSP-Kamera-Feed lokal zu hosten. Anschließend erstellen wir eine GStreamer-Pipeline, um diesen RTSP-Kamera-Feed in den angegebenen Kinesis-Videostream hochzuladen.

Zur Einrichtung auf Ihrem Raspberry Pi Gst-Rtsp-Server

1. Installieren Sie die notwendigen Abhängigkeitsbibliotheken, um das Gst-Rtsp-Server Projekt zu erstellen. Stellen Sie sicher, dass Sie auch die Softwarevoraussetzungen installiert haben. Geben Sie Folgendes in Ihr Terminal ein:

```
sudo apt-get update  
sudo apt-get install libgstrtspserver-1.0
```

2. Laden Sie die 1.22-Version von GStreamer auf Ihren Raspberry Pi herunter.

```
git clone https://gitlab.freedesktop.org/gstreamer/gstreamer.git --single-  
branch -b 1.22
```

3. Wechseln Sie in das Verzeichnis examples im gst-rtsp-Server.

```
cd gstreamer
cd subprojects
cd gst-rtsp-server
cd examples
```

4. Kompilieren Sie die Datei test-launch.c mit gcc in eine ausführbare Datei namens test-launch.

```
gcc -o test-launch test-launch.c `pkg-config --cflags --libs gstreamer-rtsp-server-1.0`
```

5. Führen Sie die ausführbare Datei mit den folgenden Argumenten aus. Hinweis: Es kann einige Zeit dauern, bis GStreamer zum ersten Mal geladen wird.

```
./test-launch "videotestsrc is-live=true ! video/x-raw,height=480,width=640,framerate=10/1 ! videoconvert ! x264enc tune=zerolatency bitrate=512 key-int-max=25 bframes=0 ! h264parse ! rtph264pay ! name=pay0 pt=96"
```

Die Ausgabe sollte folgendermaßen aussehen:

```
stream ready at rtsp://127.0.0.1:8554/test
```

6. Überprüfen Sie den RTSP-Videostream. Sie können jeden RTSP-Viewer verwenden. Zum Beispiel VLC Media Player. Um Ihren Live-Stream mit dem VLC Media Player anzusehen, öffnen Sie ein neues Terminal und geben Sie Folgendes ein:

```
sudo apt-get install vlc
```

um den VLC Media Player zu installieren. Geben Sie dann Folgendes ein:

```
vlc rtsp://127.0.0.1:8554/test
```

Ein VLC-Fenster sollte mit dem Live-Stream erscheinen. Wenn nicht, überprüfen Sie, ob die ausführbare Teststartdatei noch läuft, und überprüfen Sie die Ausgabe auf Fehler.

Eine andere Möglichkeit, den RTSP-Stream zu überprüfen, ist die Verwendung des Dienstprogramms gst-discoverer-1.0. Type:

```
gst-discoverer-1.0 "rtsp://127.0.0.1:8554/test"
```

Die erwartete Ausgabe sieht wie folgt aus:

```
Analyzing rtsp://127.0.0.1:8554/test
Done discovering rtsp://127.0.0.1:8554/test

Properties:
  Duration: 99:99:99.999999999
  Seekable: no
  Live: yes
  unknown #0: application/x-rtp
    video #1: H.264 (Constrained Baseline Profile)
      Stream ID:
        359314d7d4bba383223927d7e57d4244d0800e629c626be81c505055c62170e2/
  video:0:0:RTP:AVP:96
    Width: 640
    Height: 480
    Depth: 24
    Frame rate: 10/1
    Pixel aspect ratio: 1/1
    Interlaced: false
    Bitrate: 0
    Max bitrate: 0
```

Um den RTSP-Feed mit kvssink an Ihren Kinesis-Videostream zu senden

Diese GStreamer-Pipeline wird verwendet, um eine Verbindung `rtspsrc` zum RTSP-Server herzustellen, um den RTP-Videostream abzurufen. Sie übergibt die Frames an `denrtph264depay`, der die H.264-encoded Videoframes aus den RTP-Paketen extrahiert. `h264parse` gruppiert die Videobilder in das Format, das `kvssink` sie verstehen können. `kvssink` nimmt den H.264-encoded Videostream, packt ihn in das Matroska-Containerformat (MKV) und lädt ihn in Ihren Kinesis-Videostream hoch.

Führen Sie den folgenden Befehl aus:

```
gst-launch-1.0 -v rtspsrc location="rtsp://127.0.0.1:8554/test" short-
header=true \
  ! rtph264depay \
```

```
! h264parse \
! video/x-h264,format=avc,alignment=au \
! 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 verwenden, bei dem es sich um einen x264enc Software-Encoder handelt.

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"
```

v4l2h264encundv4l2convert:

```
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 „xxxxxxxx“

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 dessen, welches Element fehlt, die entsprechende Aktion:

- kvssink: Siehe [the section called “Laden Sie das C++ Producer SDK herunter und erstellen Sie es”](#).
- libcamerasrc: Siehe So [the section called “Fehler „Die Aktivierung des Pufferpools ist fehlgeschlagen”](#) installieren Sie das libcamerasrc GStreamer-Element.
- omxh264encoder: v4l2h264enc

Folgen Sie [the section called “Installieren Sie die Softwarevoraussetzungen”](#) den Anweisungen, um alle GStreamer-Bibliotheken zu installieren. Wenn Sie sie alle installiert haben und diese Elemente nicht angezeigt werden, bedeutet das, dass Ihr Raspberry Pi nicht über die entsprechende Hardware verfügt. Verwenden Sie x264enc stattdessen den Software-Encoder.

- Sonstiges: Folgen Sie [the 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 `libcamerasrc` stattdessen sollte sie verwendet werden.

```
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 ist, kann dieser Fehler auftreten, wenn GStreamer versucht, automatisch zu erkennen, welche Elemente verwendet werden sollen.

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

Auflösung

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

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

Sobald `libcamerasrc` das Element installiert ist und Sie das `autovideosrc` Element verwenden, sollte GStreamer automatisch auf die richtige Quelle umschalten. `libcamerasrc 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 das, 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 unabgestimmten Speicherzugriff, um die Speichernutzung zu optimieren. Dies wird nicht von allen Geräten unterstützt.

Auflösung

Um das SDK im angepassten Speicherzugriffsmodus zu verwenden, müssen Sie das `ALIGNED_MEMORY_MODEL` CMake-Flag `ON` beim Kompilieren explizit auf `setenvkvssink`, da es standardmäßig auf `OFF` ist. Eine detailliertere [the section called “Laden Sie das C++ Producer SDK herunter und erstellen Sie es”](#) Anleitung finden Sie unter.

Timestamp friert ein und die Pipeline kommt zum Stillstand

Bei x264enc der Verwendung in einer GStreamer-Pipeline können Situationen auftreten, in denen sich die Zeitleiste 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 Standard-Eingabepuffers übersteigt. Infolgedessen füllt sich der Eingabepuffer, was dazu führt, dass 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. Dies reduziert die Codierungslatenz erheblich, indem es für Echtzeitszenarien optimiert wird und sicherstellt, 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 das Stoppen der Pipeline, kann sich jedoch auf die Effizienz und Qualität der Codierung auswirken. Für Szenarien, die sowohl eine niedrige Latenz als auch eine hohe Qualität erfordern, sollten Sie alternative Herangehensweisen in Betracht ziehen, z. B. die Verwendung von Hardwareoptimierungen oder die Suche nach einer Webcam, die H.264 direkt ausgibt, wobei dieser Codierungsschritt übersprungen wird. 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 Prozess, bis der erste abgeschlossen ist.

Auflösung

Erstellen Sie ein Loopback-Gerät, das es mehreren Prozessen ermöglicht, die Loopback-Schnittstelle gleichzeitig zu verwenden. Weitere Informationen finden Sie unter Stack Exchange. <https://raspberrypi.stackexchange.com/questions/19630/take-picam-image-while-motion-is-running/19897#19897>

Interner Datenstromfehler

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 GStreamer beim Versuch, eine Verbindung (Link) zwischen den Pads zweier Elemente in Ihrer Pipeline herzustellen, auf ein Problem gestoßen 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 <https://gstreamer.freedesktop.org/documentation/coreelements/capsfilter.html?gi-language=c> Capsfilter-Elemente anpassen und and/or ändern, welche Elemente Ihre Pipeline verwendet.

In den meisten Fällen wird nach einem framerate oder gefragt, was resolution die Kamera nicht unterstützt. Verwenden Sie es `gst-device-monitor-1.0` im Terminal, um das unterstützte `frameratesresolutions`, und `abzurufenformats`. Sie können das [Videoscale](#) GStreamer-Element verwenden, um die Videoauflösung anzupassen, und [videorate](#), um die Videobildrate anzupassen.

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

Spiele Medien aus Ihrem Kinesis-Videostream ab

Öffnen Sie die [Kinesis Video Streams-Konsole](#) und wählen Sie den Stream-Namen für den Stream, 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.
- Überprüfe im Abschnitt Stream-Informationen den Codec, die Auflösung und die Bitrate des Streams. Die Auflösungs- und Bitratenwerte 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 unter 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 bei Server-side 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 PERSISTED Acks in den Protokollen

Wenn Sie PERSISTED Acks in den Protokollen sehen, hat Kinesis Video Streams die Medien, die von hochgeladen wurden, erfolgreich aufgenommen und gespeichert. kvssink Die von Kinesis Video Streams erhaltenen Acks sehen so aus. Sehen Sie sich im JSON den Wert für den "EventType" Schlüssel an.

[illegible]

[illegible]

Auflösung

Warten Sie ein oder zwei Minuten in der Kinesis Video Streams-Konsole und verwenden Sie dann den Doppelpfeil nach rechts. Wenn keine Medien angezeigt werden, 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 für kvssink an”Weitere Informationen darüber, wie kvssink bestimmt, welche Region verwendet werden soll, finden Sie unter.

Das Laden des Mediums in das AWS-Managementkonsole

 Important

Das Wiedergabeerlebnis auf der Konsole unterscheidet sich vom HLS- und DASH-Wiedergabeerlebnis. Verwenden Sie auch die [im Media Player](#) gehostete Beispielwebseite, GitHub um die Wiedergabe zu testen. Den Quellcode der Webseite finden Sie [hier](#).

Medien können aufgrund einer schlechten Netzwerkbandbreite oder eines eingeschränkten Geräts in der Konsole langsam geladen werden. 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. P-Frames
- Key-frames enthalten vollständige Bilddaten, enthalten aber P-frames nur Änderungen gegenüber früheren Frames.
- Das „Keyframe-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: $\text{Keyframe-Intervall} \div \text{Bildrate}$

Beispiel:

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

$\text{Fragmentlänge} = 30 \div 15 = 2 \text{ Sekunden}$

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

Auflösung

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

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

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

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

Das Medium ist verzerrt oder weist Artefakte auf

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

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

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

Prüfen Sie in der Ausgabedatei von `filesink` oder dem Media Player, der bei der Verwendung geöffnet wird `autovideosink`, ob die Artefakte ebenfalls vorhanden sind.

Überprüfen Sie auch die Ausgabe der folgenden Pipeline:

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

Wenn Sie Elemente wie [Dewarp zu Ihrer Pipeline hinzufügen](#), können Sie die Ausgaben von Fischaugen-Kameras korrigieren.

Überprüfen 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 in H.264 die Verwendung umwandeln können. `x264enc` Suchen Sie in den GStreamer-Foren nach Unterstützung für andere Benutzer mit ähnlichen and/or Pipeline-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 erhalten, die der folgenden ähnelt, 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 32-Bit 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 Log-Zeile 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
```

Verifizieren Sie die Architektur Ihres Systems:

- Überprüfen Sie die Kernel-Bitness: führen Sie aus `uname -m`
- Überprüfe die Bitanzahl des Benutzerbereichs: `run getconf LONG_BIT`

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

Auflösung

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

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

Beheben Sie Probleme beim **Laden von kvssink** 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 Folgendes enthält `kvssink`. Umgebungsvariablen sind sitzungsspezifisch, daher müssen Sie sie für jede neue Sitzung festlegen. Wie Sie diese Änderung dauerhaft vornehmen können, finden Sie unter „Aktualisieren Sie das Startskript Ihrer Shell, sodass auch die Umgebungsvariable `GST_PLUGIN_PATH` gesetzt wird“.

Fehler: Die gemeinsame Objektdatenbank kann nicht geöffnet werden: Keine solche Datei oder kein solches Verzeichnis

Wenn der Fehler auftritt `Cannot 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 geschieht in der Regel aufgrund von:

- Umzug kvssink an einen anderen Ort als den Ort, an dem es gebaut wurde.
- Cross-compiling 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 bei verschobenen Bibliotheken das Verzeichnis mit den fehlenden Bibliotheken zu `LD_LIBRARY_PATH`.

Im Stammverzeichnis des ursprünglichen Repositorys können Sie die fehlende Bibliothek mithilfe des `find` Dienstprogramms finden. Geben Sie im Terminal Folgendes ein:

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

Ausgabe:

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

Das Dateipfadtrennzeichen auf Linux-Geräten lautet: `.` Mit dem folgenden Befehl wird ein neuer Ordnerpfad an die vorhandene `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 das an defekten symbolischen Links () liegen. `symlinks` Symlinks können kaputt gehen, wenn das Betriebssystem des Host-Computers nicht mit dem des Zielcomputers übereinstimmt. Zum Beispiel Cross-Compilierung 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 gebaut wurden (Raspberry Pi verwendet ARM-CPU's).

Navigieren Sie zu dem im Fehler angegebenen Speicherort der Bibliothek und geben Sie: `ls -la` ein, 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 kaputt. Bei Broken zeigen `symlinks` keine Pfeile auf ihre Ziele.

Auflösung

Es gibt zwei Möglichkeiten, die Symlinks zu korrigieren:

- Empfohlen: 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 um, damit sie mit der `symlink` übereinstimmt.

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 Kreuzkompilierung zu vermeiden.

Fehlende Abhängigkeiten:

Wenn der Name der fehlenden Bibliothek mit `beginntlibkvs`, lesen Sie den Abschnitt „Verschiebte Bibliotheken“ weiter oben, um die Kinesis Video Streams-Bibliotheken vom Host-Gerät auf das Zielgerät zu installieren.

Gehen Sie andernfalls wie folgt vor, [the section called “Installieren Sie die Softwarevoraussetzungen”](#) um sicherzustellen, dass alle Voraussetzungen für Open-Source-Software 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 MKVGen-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Trace-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der Utils-Bibliothek zurückgegebene Fehler- und Statuscodes](#)
- [Von der View-Bibliothek zurückgegebene Fehler- und Statuscodes](#)

Von der Client-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Kinesis Video Streams Client Streams-Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Description	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.

Code	Fehlermeldung	Description	Empfohlene Aktion
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 ist SDK-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 ist SDK-Kontingente für Produzenten .
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.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	Ungültige Länge des Stream-Namens.	Beziehen Sie sich auf die maximale Länge des Stream-Namens in Zeichen, die in angegeben ist SDK-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 ist SDK-Kontingente für Produzenten .
0x5200000c	STATUS_INVALID_SPILL_RATIO	Ungültiges Überlaufverhältnis.	Geben Sie die Überlaufquote als Prozentsatz zwischen 0 und 100 an.
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Ungültige Version der StorageInfo-Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x5200000e	STATUS_INVALID_STREAM_STATE	Der Stream befindet sich in einem Zustand, in dem die aktuelle Operation nicht zulässig ist.	In den meisten Fällen tritt dieser Fehler auf, wenn das SDK nicht den Status erreicht, den es für die Ausführung des angeforderten Vorgangs benötigt. Dies tritt z. B. auf, wenn der <code>GetStream</code> <code>ingEndpoint</code> -API-Aufruf fehlschlägt und die Client-Anwendung ihn ignoriert und weiterhin Frames in den Stream leitet.
0x5200000f	STATUS_SERVICE_CALLBACKS_MISSING	In der <code>Callbacks</code> -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-Wrappers erfüllen diese Aufrufe.

Code	Fehlermeldung	Description	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	Description	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_ARGUMENT_ERROR	Für den Dienstaufwurf 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	Das 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	Das Gerät wurde nicht bereitgestellt.	Stellen Sie sicher, dass das Gerät bereitgestellt wurde.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	Der Dienst gibt zurück, dass eine generische Ressource nicht gefunden wurde.	Dieser Fehler tritt auf, wenn der Dienst die Ressource (z. B. einen Stream) nicht finden kann. Dies kann in verschiedenen Kontexten unterschiedliche Bedeutungen haben. Die wahrscheinlichste Ursache ist aber die Nutzung von APIs, bevor der Stream erstellt wurde. Durch die Verwendung des SDK wird bestätigt, dass der Stream zuerst erstellt wurde.
0x52000018	STATUS_INVALID_AUTH_LEN	Ungültige Länge der Authentifizierungsinformationen.	Die aktuellen Werte werden unter 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-API-Aufruf ist fehlgeschlagen.	Weitere Informationen zum Grund für das Fehlschlagen der Operation finden Sie in der Fehlerzeichenfolge.

Code	Fehlermeldung	Description	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	Description	Empfohlene Aktion
0x5200002e	STATUS_STREAM_OUT_OF_MEMORY	Im Inhaltsspeicher ist nicht mehr genügend Speicherplatz vorhanden.	Der Inhaltsspeicher wird von den Streams gemeinsam genutzt und sollte über genügend Kapazität zum Speichern der maximalen Dauer aller Streams + ~20 % (einschließlich Defragmentierung) verfügen. Es ist wichtig, dass der Speicher nicht überläuft. Wählen Sie Werte für die maximale Dauer pro Stream, die der kumulativen Speicherröße und den Latenztoleranzen entsprechen. Wir empfehlen, die Frames zu löschen, sobald sie aus der Inhaltsansicht herausfallen, anstatt sie einfach nur zu platzieren (Speicherdruck im Inhaltsspeicher). Das liegt daran, dass durch das Löschen der Frames die Rückrufe von Benachrichtigungen über den Stream-Druck ausgelöst werden. Die Anwendung kann dann die vorgelagerten Medien-Komponenten (wie den Encoder) so anpassen, dass die Bitrate gesenkt, Frames

Code	Fehlermeldung	Description	Empfohlene Aktion
			verworfen oder entsprechend gehandelt wird.
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	Für einen Stream sind derzeit keine Daten mehr verfügbar.	Dies ist möglicherweise ein gültiges Ergebnis, wenn die Medienpipeline die an den Dienst zu sendenden Frames langsamer produziert als der Netzwerkthread verbraucht. Higher-level Clients (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	Description	Empfohlene Aktion
0x52000033	STATUS_SERVICE_CLIENT_LIMIT_ERROR	Client-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000034	STATUS_SERVICE_DEVICE_LIMIT_ERROR	Geräte-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000035	STATUS_SERVICE_STREAM_LIMIT_ERROR	Stream-Limit.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000036	STATUS_SERVICE_RESOURCE_DELETED_ERROR	Die Ressource wurde gelöscht oder wird gerade gelöscht wird.	Wird vom Dienst zurückgegeben. Weitere Informationen finden Sie in der Kinesis Video Streams API-Referenz.
0x52000037	STATUS_SERVICE_TIMEOUT_ERROR	Zeitüberschreitung des Dienstaufrufs.	Der Aufruf einer bestimmten Dienst-API führte zu einem Timeout. Stellen Sie sicher, dass Sie über eine gültige Netzwerkverbindung verfügen. Der PIC wiederholt die Operation automatisch.

Code	Fehlermeldung	Description	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	Description	Empfohlene Aktion
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	Die TagResource -API ist fehlgeschlagen.	Der TagResource -API-Aufruf ist fehlgeschlagen. Prüfen Sie auf eine gültige Netzwerkverbindung. Weitere Informationen zum Fehler finden Sie in den Protokollen.
0x5200003e	STATUS_INVALID_CUSTOM_DATA	PIC-APIs werden mit ungültigen benutzerdefinierten Daten aufgerufen.	In einem Aufruf der PIC-APIs wurden ungültige benutzerdefinierte Daten angegeben. Dies kann nur bei Clients vorkommen, die PIC direkt nutzen.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	Ungültige CreateStreamResponse -Struktur.	Die Struktur oder die zugehörigen Felder sind ungültig (d. h. ARN ist Null oder größer als der unter SDK-Kontingente für Produzenten angegebene Wert).

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	Client-Authentifizierung fehlgeschlagen.	Der PIC konnte nach einer Reihe von Wiederholungsversuchen nicht die richtigen Authentifizierungsinformationen (AccessKeyId oderSecretAccessKey) abrufen. Überprüfen Sie die Authentifizierungsintegration. Die Beispielanwendungen verwenden Umgebungsvariablen, um Anmeldeinformationen an die C++ Producer-Bibliothek zu übergeben.
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	Das Anfordern des Sicherheits-Token ist fehlgeschlagen.	Diese Situation kann bei Clients auftreten, die PIC direkt nutzen. Der Aufruf schlägt nach einigen Wiederholungsversuchen mit diesem Fehler fehl.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	Bereitstellungsfehler.	Die Bereitstellung ist nicht implementiert.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	Der Producer Client konnte nicht erstellt werden.	Vom PIC wurde nach einigen Wiederholungsversuchen ein generischer Fehler zurückgegeben, wenn die Client-Erstellung fehlschlägt.
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	Der Producer Client konnte nicht in den Status READY (BEREIT) gebracht werden.	Wird vom PIC-Zustandsautomaten zurückgegeben, wenn der PIC nicht in den Status READY (BEREIT) gebracht werden kann. Weitere Informationen zur Ursache finden Sie in den Protokollen.
0x52000045	STATUS_TAG_RESOURCE_CLIENT_CALL_FAILED	TagResource für den Producer Client ist fehlgeschlagen.	Der API-Aufruf TagResource für den Producer Client ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Protokollen.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	Device/Producer Die Erstellung ist fehlgeschlagen.	Die SDKs auf höherer Ebene (z. B. C++ oder Java) implementieren die API zur Erstellung von Geräten oder Herstellern noch nicht. Clients, die PIC direkt nutzen, können über die Ergebnis-Benachrichtigung auf einen Fehler hinweisen.
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	Der Zeitstempel der empfangenen ACK wird nicht angezeigt .	Dieser Fehler tritt auf, wenn der Frame, der der empfangenen ACK entspricht, aus dem Inhaltsansichtsfenster fällt. Dies geschieht in der Regel bei langsamer ACK-Übermittlung. Dies kann als eine Warnung und ein Hinweis auf einen langsamen Downlink gedeutet werden.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	Ungültige Version der FragmentAck -Struktur.	Geben Sie die korrekte aktuelle Version der FragmentAck -Struktur an.

Code	Fehlermeldung	Description	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 <code>GetStreamData</code> an, dass die aktuelle Upload-Sitzung beendet ist. Dies tritt auf, wenn die Sitzung endet oder fehlerhaft ist, das Sitzungs-Token abgelaufen ist und die Sitzung rotiert wird.
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	Doppelter Stream-Name.	Mehrere Datenströme können nicht denselben Stream-Namen besitzen. Wählen Sie eine eindeutigen Namen für den Stream.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	Ungültiger Aufbewahrungszeitraum.	In der StreamInfo - Struktur wird ein ungültiger Aufbewahrungszeitraum angegeben. Weitere Informationen zum gültigen Wertebereich für den Aufbewahrungszeitraum finden Sie unter SDK-Kontingente für Produzenten .
0x5200004d	STATUS_INVALID_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_INVALID_ACK_DUPLICATE_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	Description	Empfohlene Aktion
0x5200004f	STATUS_INVALID_ACK_VALUE_START	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüsselwert-Startindikators. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000050	STATUS_INVALID_ACK_VALUE_END	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge wegen eines ungültigen Schlüsselwert-Endindikators. Die Codefragment-ACK-Zeichenfolge ist möglicherweise beschädigt. Dieser selbstkorrigierende Fehler kann als eine Warnung gedeutet werden.
0x52000051	STATUS_INVALID_ACK_TYPE	Ungültig FragmentAck .	Fehler beim Analysieren der Codefragment-ACK-Zeichenfolge, da ein ungültiges ACK-Typ angegeben wurde.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	Der Stream wurde gestoppt.	Obwohl der Stream gestoppt wurde, wird dennoch ein Frame in den Stream geleitet.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Ungültige Version der <code>StreamMetrics</code> -Struktur.	Geben Sie die korrekte aktuelle Version der <code>StreamMetrics</code> -Struktur an.
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Ungültige Version der <code>ClientMetrics</code> -Struktur.	Geben Sie die korrekte aktuelle Version der <code>ClientMetrics</code> -Struktur an.
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	Bei der Initialisierung des Producers wurde nicht der Status READY (BEREIT) erreicht.	Während der Producer Client-Initialisierung wurde nicht der Status READY (BEREIT) erreicht. Weitere Informationen finden Sie in den Protokollen.
0x52000056	STATUS_STATE_MACHINE_STATE_NOT_FOUND	Interner Fehler des Zustandsautomaten.	Kein öffentlich sichtbarer Fehler.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	In der <code>FragmentAck</code> -Struktur wird ein ungültiger ACK-Typ angegeben.	Die in der <code>FragmentAck</code> -Struktur enthaltenen ACK-Typen sollten im öffentlichen Header definiert sein.
0x52000058	STATUS_INVALID_STREAM_READY_STATE	Interner Übergangsfehler des Zustandsautomaten.	Kein öffentlich sichtbarer Fehler.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	Das Stream-Objekt wurde freigegeben, nachdem der Producer freigegeben wurde.	Es wurde versucht, ein Stream Objekt freizugeben, nachdem das Producer-Objekt freigegeben wurde. Dies kann nur bei Clients vorkommen, die PIC direkt nutzen.
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	Interner Speicherfehler.	Ein interner Fehler, der darauf hinweist, dass die tatsächliche Zuweisung sgröße aus dem Inhaltsspeicher kleiner ist als die Größe des verpackten Frames und Fragments.
0x5200005b	STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION	Interner Speicherfehler.	Die gespeicherte Größe der Zuweisung in der Inhaltsansicht ist größer als die Zuweisungsgröße im Inhaltsspeicher.
0x5200005c	STATUS_ACK_ERROR_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 im Allgemeinen auf, wenn das Backend den Stream nicht abrufen kann. Auto-restreaming kann diesen Fehler normalerweise korrigieren.

Code	Fehlermeldung	Description	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 ordnungsgemäß 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	Description	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	Description	Empfohlene Aktion
0x52000060	STATUS_ACK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC	Timecodes steigen nicht monoton an.	Das Producer-SDK erzwingt Zeitstempel, sodass Clients, die das SDK verwenden, diesen Fehler nicht erhalten sollten.
0x52000061	STATUS_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	Description	Empfohlene Aktion
0x52000062	STATUS_AC K_ERR_INV ALID_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	Description	Empfohlene Aktion
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	Ungültiger Producer-Zeitstempel.	Der Dienst gibt diesen Fehler ACK zurück, wenn die Uhr des Herstellers stark in die future abdriftet. Higher-level SDKs (z. B. Java oder C++) verwenden eine Version der Systemuhr, 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_ERROR_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	Description	Empfohlene Aktion
0x52000065	STATUS_AK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS Fehler „Zugriff verweigert“.	Wird zurückgegeben, wenn das Konto nicht auf den angegebenen Schlüssel zugreifen kann.
0x52000066	STATUS_AK_ERR_KMS_KEY_DISABLED	AWS KMS Der Schlüssel ist deaktiviert.	Der angegebene Schlüssel wurde deaktiviert.
0x52000067	STATUS_AK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS Fehler bei der Schlüsselvalidierung.	Allgemeiner Validierungsfehler. Weitere Informationen finden Sie in der AWS Key Management Service - API-Referenz .
0x52000068	STATUS_AK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key nicht verfügbar.	Der Schlüssel ist nicht verfügbar. Weitere Informationen finden Sie in der AWS Key Management Service - API-Referenz .
0x52000069	STATUS_AK_ERR_KMS_KEY_INVALID_USAGE	Ungültige Verwendung des KMS-Schlüssels.	Der AWS KMS key ist nicht für die Verwendung in diesem Kontext konfiguriert. Weitere Informationen finden Sie in der AWS Key Management Service - API-Referenz .
0x5200006a	STATUS_AK_ERR_KMS_KEY_INVALID_STATE	AWS KMS ungültiger Status.	Weitere Informationen finden Sie in der AWS Key Management Service - API-Referenz .

Code	Fehlermeldung	Description	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-Managementkonsole gelöscht.
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	Interner Fehler	Interner Fehler eines generischen Dienstes.
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR	Fragment-Archivierungsfehler.	Wird zurückgegeben, wenn der Dienst das Fragment nicht dauerhaft speichern und indizieren kann. Dies kann, wenn auch selten, aus verschiedenen Gründen auftreten. Standardmäßig wiederholt das SDK das Senden des Fragments.
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	Unbekannter Fehler.	Der Service gab einen 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	Description	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_MAXIMUM_FRAGMENT_METADATA_COUNT	Die maximale Anzahl von Metadaten elementen wurde zu einem Fragment hinzugefügt.	Ein Kinesis-Videostream kann einem Fragment bis zu 10 Metadaten elemente hinzufügen, indem entweder ein nicht persistentes Element zu einem Fragment hinzugefügt wird oder indem ein persistentes Element zur Metadaten-Warteschlange hinzugefügt wird. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	Ein Limit (maximale Metadatenanzahl, Länge des Metadatennamens oder Länge des Metadatenwerts) wurde erreicht.	Das Producer-SDK begrenzt die Anzahl und Größe von Metadaten elementen. Dieser Fehler tritt nur auf, wenn die Grenzwerte im Producer-SDK-Code geändert werden. Weitere Informationen finden Sie unter Verwenden von Streaming-Metadaten mit Kinesis Video Streams .
0x52000076	STATUS_BLOCKING_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	Description	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 fragmentierten Stream gesendet werden, der kein Schlüsselframe ist.
0x52000079	STATUS_TRACK_INFO_MISSING	Track-Informationen fehlen.	Die Titelnummer muss größer als Null sein und mit der Track-ID übereinstimmen.
0x5200007a	STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED	Maximale Track-Anzahl wurde überschritten.	Sie können maximal drei Titel pro Stream haben.
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	Der Offline-Streaming-Modus-Aufbewahrungszeitraum ist auf Null gesetzt.	Die Aufbewahrungszeit für den Offline-Streaming-Modus sollte nicht auf Null gesetzt werden.
0x5200007c	STATUS_ACK_ERROR_TRACK_NUMBER_MISMATCH	Die Track-Nummer des ACK-Fehlers ist falsch zugeordnet.	
0x5200007d	STATUS_ACK_ERROR_FRAMES_MISSING_FOR_TRACK	Frames fehlen für einen Track.	

Code	Fehlermeldung	Description	Empfohlene Aktion
0x5200007e	STATUS_ACK_ERR_MORE_THAN_ALLOWED_TRACKS_FOUND	Die maximal zulässige Anzahl von Tracks wurde überschritten.	
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		

Code	Fehlermeldung	Description	Empfohlene Aktion
0x52000085	STATUS_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUT_MEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		
0x52000089	STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS		
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		

Code	Fehlermeldung	Description	Empfohlene Aktion
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_METADATA_VALUE_LENGTH		

Von der Duration-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Duration Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

Von der gemeinsamen Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Common Bibliothek zurückgegeben werden.

Note

Diese Codes für Fehler- und Statusinformationen sind vielen APIs gemeinsam.

Code	Code ohne führende Nullen	Fehlermeldung	Description
0x00000001	0x1	STATUS_NULL_ARG	Für ein obligatorisches Argument 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.

Code	Code ohne führende Nullen	Fehlermeldung	Description
0x0000000a	0xa	STATUS_READ_FILE_FAILED	Fehler beim Lesen aus einer Datei.
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	Description
0x00000012	0 x 12	STATUS_THREAD_READ_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_READ_INVALID_ARG	Es wurden ungültige Thread-Attribute angegeben, oder ein anderer Thread wartet bereits darauf, diesem Thread beizutreten.
0x00000014	0 x 14	STATUS_THREAD_READ_PERMISSIONS	Keine Berechtigung zum Festlegen der in Thread-Attributen angegebenen Scheduling-Richtlinien und -Parameter.
0x00000015	0 x 15	STATUS_THREAD_READ_DEADLOCKED	Ein Deadlock wurde erkannt oder der beitretende Thread spezifiziert den aufrufenden Thread.

Code	Code ohne führende Nullen	Fehlermeldung	Description
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	Description
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_ATTRIBUTE_INIT_FAILED	Das Thread-Attribute-Objekt konnte nicht initialisiert werden.
0x0000001d	0x1d	STATUS_THREAD_ATTRIBUTE_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_PARAMETERS_VERSION	Ungültige "ThreadParameters" 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	Description
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	Description
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	Description
0x10000015	STATUS_INVALID_ALLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

Von der MKVGen-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der MKVGen Bibliothek zurückgegeben werden.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Ungültige Mitglieder der Frame-Datenstruktur. Stellen Sie sicher, dass Dauer, Größe und Frame-Daten gültig sind und innerhalb der unter angegebenen Grenzen liegen SDK-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 wurde ein ungültiger Annex-B Startcode gefunden. Beispielsweise wurde das Annex-B Anpassungs-Flag angegeben und der Code stößt auf eine ungültige Startsequenz mit mehr als drei Nullen. Ein gültiges Annex-B Format sollte über eine Sequenz zur Verhinderung von Emulation verfügen, um einer Sequenz von drei oder mehr Nullen im Bytestream zu entkommen. 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_AN_NEXB_AND_AVCC_SPECIFIED	Sowohl die Anpassung von AVCC als auch von NALUs wurden spezifiziert Annex-B . Machen Sie nur eine oder keine Angabe.
0x32000010	STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD	Ungültiges Annex-B CPD-Format, wenn das Annex-B Anpassungsflag angegeben ist. Stellen Sie sicher, dass die CPD ein Annex-B gültiges Format hat. Ist dies nicht der Fall, entfernen Sie das Annex-B CPD-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	Private Codec-Daten konnten nicht analysiert werden. H264/H265
0x32000013	STATUS_MKV_INVALID_H264_H265_SPS_WIDTH	Fehler beim Extrahieren der Breite privater Codec-Daten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000014	STATUS_MKV_INVALID_H264_H265_SPS_HEIGHT	Fehler beim Extrahieren der Höhe privater Codec-Daten.
0x32000015	STATUS_MKV_INVALID_H264_H265_SPS_NALU	Ungültiges H264/H265 SPS-NALU.
0x32000016	STATUS_MKV_INVALID_BIH_CPD	Ungültiges Bitmap-Info-Header-Format in den privaten Codec-Daten.
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	Ungültige Anzahl an Invalid High Efficiency Video Coding (HEVC)-Network Abstraction Layer (NALU)-Einheiten.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	Ungültiges HEVC-Format.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	Fehlende HEVC-NALUs im Sequence Parameter Set (SPS).
0x3200001a	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	Ungültige HEVC-SPS-NALU-Größe.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	Ungültige Chroma-Format-IDC.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	Ungültiges HEVC-reserviertes SPS.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	Mindestgröße 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 Annex-B NALUs.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Ungültiger privater Codec in NaLuS. Annex-B
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	Ungültige Länge des Tag-Namens. Zulässiger Wert ist größer als Null und niedriger als 128.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	Ungültige Länge des Tag-Wertes. Der gültige Wert ist größer als Null und kleiner als 256.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	Ungültige Generator-Zustands-Tags.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	Ungültiger Sampling-Häufigkeitsindex für private AAC-Codec-Daten.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	Ungültige Kanalkonfiguration für private AAC-Codec-Daten.

Code	Fehlermeldung	Beschreibung//Empfohlene Maßnahme
0x32000025	STATUS_MKV_INVALID_AAC_CPD	Ungültige private AAC-Codec-Daten.
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	Track-Informationen nicht gefunden.
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	Ungültige Segment-UUID.
0x32000028	STATUS_MKV_INVALID_TRACK_UUID	Ungültige Track-UUID.
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	
0x3200002a	STATUS_MKV_INVALID_AMS_ACM_CPD	
0x3200002b	STATUS_MKV_MISSING_SPS_FROM_H264_CPD	
0x3200002c	STATUS_MKV_MISSING_PPS_FROM_H264_CPD	
0x3200002d	STATUS_MKV_INVALID_PARENT_TYPE	

Von der Trace-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der Trace Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0x10100001	STATUS_MIN_PROFILER_BUFFER

Von der Utils-Bibliothek zurückgegebene Fehler- und Statuscodes

Die folgende Tabelle enthält Fehler- und Statusinformationen, die von Methoden in der `Utils` Bibliothek zurückgegeben werden.

Code	Fehlermeldung
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_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	Description
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	Description
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	Description
0x30000005	STATUS_CONTENT_VI W_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:. $\text{DTS}(n) \geq \text{DTS}(n-1) + \text{Duration}(n-1)$ Dieser Fehler weist oft auf einen instabilen Encoder hin. Der Encoder produziert einen Burst codierter Frames, deren Zeitstempel kleiner als die Intra-Frame-Dauer ist. Oder der Stream ist zur Verwendung von SDK-Zeitstempeln konfiguriert, und die Frames werden schneller als ihre Dauer gesendet. Um ein gewisses „Jitter“ im Encoder zu vermeiden, können Sie die Dauer des Frames in der Frame-Struktur anpassen, wenn Sie <code>putFrame()</code> oder <code>put()</code> aufrufen. <code>KinesisVideoFrame</code> Bei einigen Streams muss die Dauer für die Fehlererkennung genauer kontrolliert werden.

Code	Fehlermeldung	Description
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	Description	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 PlatformC allbacks - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Ungültige Version der ProducerC allbacks - Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Ungültige Version der StreamCallbacks -Struktur.	Geben Sie die korrekte aktuelle Version der Struktur an.

Code	Fehlermeldung	Description	Empfohlene Aktion
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.	
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	Die maximale Länge des Werts des Anforderungs-Headers wurde erreicht.	

Code	Fehlermeldung	Description	Empfohlene Aktion
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	Ungültiges JSON-Objekt für einen API-Aufruf zurückgegeben.	
0x1500000d	STATUS_CURL_INIT_FAILED	Curl-Initialisierung fehlgeschlagen.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Curl-lib-Initialisierung fehlgeschlagen.	
0x1500000f	STATUS_INVALID_DESCRIPTOR_RETURN_JSON	Ungültiges Rückgabe-JSON für DescribeStream.	
0x15000010	STATUS_HMAC_GENERATION_ERROR	Fehler bei HMAC-Generation.	
0x15000011	STATUS_IOT_FAILED	Die IoT-Autorisierung ist fehlgeschlagen.	
0x15000012	STATUS_MAXIMUM_ROLE_ALIAS_LENGTH_EXCEEDED	Die maximale Länge des Rollenalias wurde erreicht.	Geben Sie einen kürzeren Alias an.
0x15000013	STATUS_MAXIMUM_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED	Die maximale Länge der Namensendung des Agenten wurde erreicht.	

Code	Fehlermeldung	Description	Empfohlene Aktion
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_OCCURRED_IN_PAST	Der IoT-Ablaufzeitstempel liegt in der Vergangenheit.	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	Die Analyse des IoT-Ablaufs ist fehlgeschlagen.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FREE_FUNC		
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		

Code	Fehlermeldung	Description	Empfohlene Aktion
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAXIMUM_IOT_THINGING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		

Code	Fehlermeldung	Description	Empfohlene Aktion
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	Description	Empfohlene Aktion
0x16000001	STATUS_CURL_PERFORM_FAILED	CURL hat einen Code zurückgegeben, der nicht erfolgreich war.	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 Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT”.
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH	Beim Abrufen von IoT-Anmeldedaten wurde eine Antwort der Länge 0 erhalten.	Überprüfen Sie das AWS Health-Dashboard und versuchen Sie es später erneut.

Code	Fehlermeldung	Description	Empfohlene Aktion
0x16000003	STATUS_IOT_NULL_AWS_CREDS	Das vom Endpunkt mit den IoT-Anmeldeinformationen zurückgegebene JSON enthielt das Anmeldeinformationssubjekt 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 [elementare Stream](#) in einer Anwendung kann entweder im AVCC-Format oder im Annex-B oder im AVCC-Format vorliegen:

- Das Annex-BFormat begrenzt [NALUs \(Network Abstraction Layer Units\)](#) 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 Format AVCC hüllt auch NALUs ein, aber jeder NALU geht ein Wert voraus, der die Größe der NALU angibt (normalerweise vier Bytes).

Viele Encoder erzeugen das Bitstream-Format. Annex-B Einige Bitstream-Prozessoren auf höherer Ebene (z. B. eine Playback-Engine oder der [Media Source Extensions \(MSE\) -Player im AWS-Managementkonsole](#)) verwenden das AVCC-Format für ihre Frames.

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

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

Flag	Anpassung	
NAL_ADAPTATION_FLAG_NONE	Keine Anpassung.	
NAL_ADAPTATION_ANNEXB_NALS	Passen Annex-B Sie NALUs an AVCC NALUs an.	
NAL_ADAPTATION_AVCC_NALS	Passen Sie AVCC-NALUs an NALUs an. Annex-B	
NAL_ADAPTATION_ANNEXB_CPD_NALS	Passen Annex-B Sie NALUs für den Codec private Daten an NALUs im AVCC-Format an.	
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	Passen Annex-B Sie NALUs für den Codec an und rahmen Sie private Daten in NALUs im AVCC-Format ein.	

Weitere Informationen zu NALU-Typen finden Sie in Abschnitt 1.3: Einheitentypen der Netzwerkabstraktionsschicht in [RFC 3984](#).

SDK-Strukturen für Hersteller

Dieser Abschnitt enthält Informationen zu Strukturen, die Sie verwenden können, um Daten für das Kinesis Video Streams Producer-Objekt bereitzustellen.

Themen

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

Die DefaultDeviceInfoProviderObjekte DeviceInfo steuern das Verhalten des Kinesis Video Streams Producer-Objekts.

Felder für Mitglieder

- **Version** — Ein ganzzahliger Wert, mit dem sichergestellt wird, dass die richtige Version der Struktur mit der aktuellen Version der Codebasis verwendet wird. Die aktuelle Version wird mit dem Makro `DEVICE_INFO_CURRENT_VERSION` angegeben.
- **name** — Der für Menschen lesbare Name für das Gerät.
- **tagCount/tags**— Wird derzeit nicht verwendet.
- **streamCount** — Die maximale Anzahl von Streams, die das Gerät verarbeiten kann. Dadurch wird anfänglich Speicher für Zeiger auf die Streamobjekte zugewiesen, die eigentlichen Streamobjekte werden aber erst später erstellt. Standardmäßig handelt es sich um 16 Streams, dies kann jedoch in der Datei `DefaultDeviceInfoProvider.cpp` geändert werden.
- **storageInfo**: Ein Objekt, das die Hauptspeicherkonfiguration beschreibt. Weitere Informationen finden Sie unter [StorageInfo](#).

StorageInfo

Gibt die Konfiguration des Hauptspeichers für Kinesis Video Streams an.

Die Standardimplementierung basiert auf einem schnellen Heap mit geringer Fragmentierung, optimiert für das Streamen. Sie verwendet den Allocator MEMALLOC, der auf einer gegebenen Plattform überschrieben werden kann. Einige Plattformen sehen eine virtuelle Speicherzuweisung vor, ohne diese Zuweisung mit physischen Seiten zu hinterlegen. Erst bei Nutzung des Speichers werden den virtuellen Seiten physische Seiten hinterlegt. Dies führt zu wenig verfügbarem Speicher im Gesamtsystem, während der Speicher tatsächlich nicht ausgelastet wird.

Berechnen Sie die Standardspeichergröße mit folgender Formel. Der Parameter DefragmentationFactor muss auf 1,2 (20 Prozent) festgelegt werden.

$$\text{Size} = \text{NumberOfStreams} * \text{AverageFrameSize} * \text{FramesPerSecond} * \text{BufferDurationInSeconds} * \text{DefragmentationFactor}$$

Im folgenden Beispiel verfügt ein Gerät über Audio- und Videostreams. Der Audiostream besteht aus 512 Samples pro Sekunde, ein durchschnittliches Sample aus 100 Byte. Der Videostream besteht aus 25 Frames pro Sekunde, mit einem Durchschnitt von 10.000 Bytes. Jeder Stream verfügt über einen Puffer mit 3 Minuten Laufzeit.

$$\begin{aligned} \text{Size} &= (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) * \\ &1.2 = 65059200 = \sim 66\text{MB}. \end{aligned}$$

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](#). Das kann auch der Fall sein, wenn die ACKs nicht verfügbar sind oder persistente ACKs verzögert sind. In diesem Fall werden die Puffer bis zur Kapazität „Pufferdauer“ gefüllt, bevor die vorherigen Frames ausfallen.

Felder für Mitglieder

- Version — Ein ganzzahliger Wert, mit dem sichergestellt wird, dass die richtige Version der Struktur mit der aktuellen Version der Codebasis verwendet wird.

- **StorageType** — Eine `DEVICE_STORAGE_TYPE` Aufzählung, die die zugrunde liegende Unterstützung und Implementierung des Speichers angibt. Derzeit wird nur der Wert `DEVICE_STORAGE_TYPE_IN_MEM` unterstützt. `DEVICE_STORAGE_TYPE_HYBRID_FILE` wird in einer künftigen Implementierung unterstützt werden und angeben, dass der Speicher in den dateigestützten Inhaltsspeicher zurückfällt.
- **StorageSize** — Die Speichergröße in Byte, die vorab zugewiesen werden soll. Der kleinste zu reservierende Wert ist 10 MB, der größte 10 GB. (Dies wird sich in einer zukünftigen Implementierung des dateigestützten Inhaltsspeichers ändern.)
- **SpillRatio** — Ein ganzzahliger Wert, der den Prozentsatz des Speichers darstellt, der vom direkten Speichertyp (RAM) zugewiesen werden soll, im Gegensatz zum sekundären Überlaufspeicher (Dateispeicher). Derzeit nicht verwendet.
- **rootDirectory**: Der Pfad zu dem Verzeichnis, in dem sich der dateigestützte Inhaltsspeicher befindet. Derzeit nicht verwendet.

Kinesis-Videostream-Strukturen

Sie können die folgenden Strukturen verwenden, um Daten für eine Instanz eines Kinesis-Videostreams bereitzustellen.

Themen

- [StreamDefinition/StreamInfo](#)
- [ClientMetrics](#)
- [StreamMetrics](#)

StreamDefinition/StreamInfo

Das Objekt `StreamDefinition` auf C++-Ebene kapselt das Objekt `StreamInfo` im plattformunabhängigen Code und stellt im Konstruktor einige Standardwerte bereit.

Felder für Mitglieder

Feld	Datentyp	Description	Standardwert
<code>stream_name</code>	<code>string</code>	Ein optionaler Streamname.	Wenn kein Name angegeben wird, wird

Feld	Datentyp	Description	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 vorhandenen Tags angehängt.	Keine Tags

Feld	Datentyp	Description	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	Description	Standardwert
max_latency	<code>duration<uint64_t, milli></code>	Die maximale Latenz für den Stream in Millisekunden. Der Callback für Stream-Latenzdruck (sofern angegeben) wird aufgerufen, wenn die Pufferlaufzeit diesen Zeitraum übersteigt. Mit der Angabe 0 wird festgelegt, dass der Callback für Stream-Latenzdruck aufgerufen wird.	<code>milliseconds::zero()</code>

Feld	Datentyp	Description	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 lautetfalse, 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 zweisekündiger Fragmente.	2

Feld	Datentyp	Description	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	Description	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 <code>true</code> ist, produziert das SDK jedes Mal einen Start des Fragments , wenn ein Keyframe auftritt. Wenn <code>false</code> , wartet Kinesis Video Streams mindestens darauf <code>fragment_duration</code> und erzeugt ein neues Fragment auf dem darauffolgenden Schlüsselbild.	<code>true</code>

Feld	Datentyp	Description	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 angebenfalse, wird sichergestellt, dass die Frames beim Einfügen in Kinesis Video Streams mit einem Zeitstempel versehen werden.	true

Feld	Datentyp	Description	Standardwert
<code>absolute_fragment_times</code>	<code>bool</code>	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 <code>PutMedia</code> in der Service-API den optionalen Zeitstempel für den Streamanfang und passt die Cluster-Zeitstempel entsprechend an. Der Service speichert die Fragmente immer mit den absoluten Zeitstempeln.	<code>true</code>

Feld	Datentyp	Description	Standardwert
fragment_acks	bool	Ob Fragment-ACKs (Bestätigungen) auf Anwendungsebene empfangen werden sollen.	true bedeutet, dass das SDK ACKs empfängt und entsprechend verfährt.
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	Description	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. Möglicherweise müssen Sie dies auf power/footprint Geräten mit extrem niedriger Kapazität false auf einstellen, um die CPU-Zyklen zu schonen. Andernfalls empfehlen wir nicht, diesen Wert false zu verwenden.	true

Feld	Datentyp	Description	Standardwert
<code>nal_adaptation_flags</code>	<code>uint32_t</code>	Gibt die Anpassung s-Flags für die Netzwerk-Abstraktionsschichteneinheit (Network Abstraction Layer Unit, NALU) an. Wenn der Bitstream H.264 codiert ist, kann er als Rohdatens trom oder als Paket in NALUs verarbeitet werden. Diese liegen entweder im oder im AVCC-Format Annex-B vor. Die meisten elementaren Stream-Produzenten und -Konsumenten (Lese-Encoder und -Decoder) verwenden das Annex-B Format, weil es Vorteile bietet, wie z. B. die Fehlerbehebung. Higher-levelSystem e verwenden das AVCC-Format, das das Standardformat für MPEG, HLS, DASH usw. ist. Die Konsolenwiedergabe verwendet die Browser-MSE (Media Source	Standardmäßig wird das Annex-B Format sowohl für die Frame-Daten als auch für die privaten Codec-Daten an das AVCC-Format angepasst.

Feld	Datentyp	Description	Standardwert
		Extensions), um den Stream im Format AVCC zu decodieren und wiederzugeben. Für H.264 (und für M-JPEG und H.265) bietet das SDK Anpassungsmöglichkeiten. Viele einfache Streams liegen in folgendem Format vor. In diesem Beispiel Ab ist das der Annex-B Startcode (001 oder 0001). <pre>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)... Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)</pre> Im Fall von H.264 befindet sich der Codec Private Data (CPD) in den Parametern SPS (Sequence Parameter Set) und PPS (Picture Parameter Set) und	

Feld	Datentyp	Description	Standardwert
		kann an das AVCC-Format angepasst werden. Sofern die Medien-Pipeline die CPD nicht separat ausgibt, kann die Anwendung die CPD aus dem Frame extrahieren. Dazu sucht es nach dem ersten IDR-Frame (der SPS und PPS enthalten sollte), extrahiert die beiden NALUs (die es sind) <code>Ab(Sps)</code> <code>Ab(Pps)</code> und fügt sie in die CPD ein. <code>StreamDefinition</code> Weitere Informationen finden Sie unter the section called “NAL-Anpassungsflags”.	
<code>frame_rate</code>	<code>uint32_t</code>	Die erwartete Bildrate. Dieser Wert wird verwendet, um den Pufferbedarf exakter zu berechnen.	25

Feld	Datentyp	Description	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	Description	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 gelöschtbuffer_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	Description	Standardwert
		mehr auf dem lokalen Gerät gespeichert werden muss.	

Feld	Datentyp	Description	Standardwert
replay_duration	duration<uint64_t>	Die Dauer in Sekunden, bis das aktuelle Lesegerät während eines Fehlers zur Wiedergabe zurückgeschaltet wird, wenn der Neustart aktiviert ist. Der Rollback stoppt am Pufferanfang (falls das Streamen gerade erst begonnen hat oder eine persistente ACK aufgetreten ist). Das Rollback versucht, auf einem Keyframe zu landen, der einen Fragments tart bezeichnet. Wenn der Fehler, der den Neustart verursacht hat, nicht auf einen ausgefallenen Host hinweist (der Host ist noch aktiv und enthält die Frame-Daten in seinen internen Puffern), wird der Rollback beim letzten empfangenen ACK-Frame beendet. Dann wird zum nächsten Keyframe	40

Feld	Datentyp	Description	Standardwert
		vor gesprungen, da das gesamte Fragment bereits im Arbeitsspeicher des Host gespeichert ist.	
connection_staleness	duration<uint64_t>	Die Zeit in Sekunden, nach der der Stream Staleness Callback aufgerufen wird, wenn das SDK das Puffer-ACK nicht empfängt. Es zeigt an, dass die Frames vom Gerät gesendet werden, das Backend sie jedoch nicht bestätigt. Dies weist auf eine getrennte Verbindung am Zwischenpunkt oder Load Balancer hin.	30
codec_id	string	Die Codec-ID für den MKV-Track.	V_MPEG4/ISO/AVC
track_name	string	Der Name des MKV-Tracks.	kinesis_video

Feld	Datentyp	Description	Standardwert
Codec PrivateData	unsigned char*	Der CPD-Puffer (Codec Private Data). Wenn die Medienpipeline über die Informationen zu den CPD verfügt, bevor der Stream startet, kann dies in <code>StreamDefinition.codecPrivateData</code> festgelegt werden. Die Bits werden kopiert und der Puffer kann nach dem Aufruf zum Erstellen des Streams wiederverwendet oder freigegeben werden. Wenn die Daten jedoch bei der Erstellung des Streams nicht verfügbar sind, können sie in einer der Überladungen der <code>KinesisVideoStream.start(cpd)</code> Funktion festgelegt werden.	null
Codec PrivateDataSize	uint32_t	Die CPD-Puffergröße	0

ClientMetrics

Das ClientMetricsObjekt wird durch Aufrufen `getKinesisVideoMetrics` gefüllt.

Felder für Mitglieder

Feld	Datentyp	Description
Version	UINT32	Die Version der Struktur, definiert im Makro <code>CLIENT_METRICS_CURRENT_VERSION</code> .
InhaltStoreSize	UINT64	Die Inhaltsspeicher-Gesamtgröße in Byte. Dies ist der in <code>DeviceInfo.StorageInfo.storageSize</code> angegebene Wert.
Inhalt StoreAvailableSize	UINT64	Aktuell verfügbare Speichergöße in Byte.
Inhalt StoreAllocatedSize	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.
insgesamt ContentViewsSize	UINT64	Die Größe des für alle Inhaltsansichten für alle Streams zugewiesenen Speichers. Dies wird nicht auf die Speichergröße angerechnet.

Feld	Datentyp	Description
		et. Dieser Speicher wird mit dem Makro MEMALLOC zugeordnet, das überschrieben werden kann, um einen benutzerdefinierten Allocator bereitzustellen.
insgesamt FrameRate	UINT64	Die insgesamt über alle Streams beobachtete Bildrate.
insgesamt TransferRate	UINT64	Die insgesamt über alle Streams beobachtete Streamrate in Byte pro Sekunde.

StreamMetrics

Das StreamMetricsObjekt wird durch Aufrufen gefüllt `getKinesisVideoMetrics`.

Felder für Mitglieder

Feld	Datentyp	Description
Version	UINT32	Die Version der Struktur, definiert im Makro STREAM_METRICS_CURRENT_VERSION .
aktuell ViewDuration	UINT64	Die Dauer der akkumulierten Frames. Im Fall eines schnellen Netzwerks ist diese Dauer entweder Null oder die Framedauer (während der Frame übertragen wird). Wenn die Dauer länger als die in der <code>max_latency</code> angegeben

Feld	Datentyp	Description
		Die Dauer wird in <code>StreamDefinition</code> , wird der Stream-Latenz-Callback aufgerufen, sofern er angegeben ist. Die Dauer wird in Einheiten von 100 ns – die Standardzeiteinheit für den PIC-Layer – angegeben.
insgesamt ViewDuration	UINT64	Die Gesamtanzeigedauer. Wenn der Stream ohne ACKs oder Persistenz konfiguriert ist, steigt dieser Wert, wenn die Frames in den Kinesis-Videostream eingefügt werden, und entspricht dem <code>buffer_duration</code> in der <code>StreamDefinition</code> . Wenn ACKs aktiviert sind und das persistente ACK empfangen wird, wird der Puffer auf das nächste Schlüsselbild gekürzt. Das liegt daran, dass der ACK-Zeitstempel den Anfang des gesamten Fragments angibt. Die Dauer wird in Einheiten von 100 ns – die Standardzeiteinheit für den PIC-Layer – angegeben.
aktuell ViewSize	UINT64	Die Größe des aktuellen Puffers in Byte.
insgesamt ViewSize	UINT64	Die Gesamtanzeigegröße in Byte.

Feld	Datentyp	Description
aktuell FrameRate	UINT64	Die beobachtete Bildrate für den aktuellen Stream.
aktuell TransferRate	UINT64	Die beobachtete Übertragungsratesrate in Byte pro Sekunde für den aktuellen Stream.

SDK-Rückrufe von Produzenten

Die Klassen und Methoden im Amazon Kinesis Video Streams Producer SDK verwalten keine eigenen Prozesse. Sie verwenden stattdessen eingehende Funktionsaufrufe und Ereignisse, um Callbacks für die Kommunikation mit der Anwendung zu planen.

Es gibt zwei Callback-Mustern, die die Anwendung für die Interaktion mit dem SDK verwenden kann:

- **CallbackProvider**— Dieses Objekt macht jeden Callback von der PIC-Komponente (Platform-Independent Code) für die Anwendung verfügbar. Dieses Muster ermöglicht eine umfassende Funktionalität, bedeutet jedoch auch, dass die Implementierung alle öffentlichen API-Methoden und -Signaturen auf C++-Ebene verarbeiten muss.
- [StreamCallbackProvider](#) und [ClientCallbackProvider](#) — Diese Objekte machen die stream- und clientspezifischen Callbacks verfügbar, und die C++-Schicht des SDK macht die restlichen Callbacks verfügbar. Dies ist das empfohlene Callback-Muster für die Interaktion mit dem Producer SDK.

In der folgenden Abbildung ist das Objektmodell der Callback-Objekte dargestellt:

In der vorherigen Abbildung ist `DefaultCallbackProvider` von `CallbackProvider` abgeleitet (der alle Callbacks im PIC bereitstellt) und enthält `StreamCallbackProvider` und `ClientCallbackProvider`.

Dieses Thema enthält die folgenden Abschnitte:

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)

- [ClientCallbacks Struktur](#)
- [Callback-Implementierungen, um das Streaming erneut zu versuchen](#)

ClientCallbackProvider

Das `ClientCallbackProvider` -Objekt stellt Callback-Funktionen auf Client-Ebene bereit. Die Details der Funktionen sind im Abschnitt [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	Description	Typ
<code>CreateDeviceFunc</code>	Derzeit nicht im Backend implementiert. Dieser Aufruf schlägt fehl, wenn er mit Java oder C++ ausgeführt wird. Andere Clients führen eine plattformspezifische Initialisierung durch.	Backend API
<code>CreateStreamFunc</code>	Wird beim Erstellen des Streams aufgerufen	Backend API

Funktion	Description	Typ
<code>DescribeStreamFunc</code>	Wird aufgerufen, wenn <code>DescribeStream</code> aufgerufen wird	Backend API
<code>GetStreamingEndpointFunc</code>	Wird aufgerufen, wenn <code>GetStreamingEndpoint</code> aufgerufen wird	Backend API
<code>GetStreamingTokenFunc</code>	Wird aufgerufen, wenn <code>GetStreamingToken</code> aufgerufen wird	Backend API
<code>PutStreamFunc</code>	Wird aufgerufen, wenn <code>PutStream</code> aufgerufen wird	Backend API
<code>TagResourceFunc</code>	Wird aufgerufen, wenn <code>TagResource</code> aufgerufen wird	Backend API
<code>CreateMutexFunc</code>	Erstellt einen Synchronisierungs-Mutex.	Synchronisierung
<code>FreeMutexFunc</code>	Gibt den Mutex frei.	Synchronisierung
<code>LockMutexFunc</code>	Sperrt den Synchronisierungs-Mutex.	Synchronisierung
<code>TryLockMutexFunc</code>	Versucht den Mutex zu sperren. Derzeit nicht implementiert.	Synchronisierung
<code>UnlockMutexFunc</code>	Hebt die Sperrung des Mutex auf.	Synchronisierung

Funktion	Description	Typ
<code>ClientReadyFunc</code>	Wird aufgerufen, wenn der Client bereit ist.	Benachrichtigung
<code>DroppedFrameReportFunc</code>	Meldet, wenn ein Frame verworfen wird.	Benachrichtigung
<code>DroppedFragmentReportFunc</code>	Meldet, wenn ein Fragment verworfen wird. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Benachrichtigung
<code>FragmentAckReceivedFunc</code>	Wird aufgerufen, wenn eine Fragment-ACK (Puffern, Empfang, persistent und Fehler) empfangen wird.	Benachrichtigung
<code>StorageOverflowPressureFunc</code>	Wird aufgerufen, wenn die Speichernutzung den Wert <code>STORAGE_PRESSURE_NOTIFICATION_THRESHOLD</code> , also 5 Prozent der Gesamtspeichergröße, unterschreitet.	Benachrichtigung
<code>StreamClosedFunc</code>	Wird aufgerufen, wenn die letzten Bits der verbleibenden Frames gestreamt werden.	Benachrichtigung
<code>StreamConnectionStaleFunc</code>	Wird aufgerufen, wenn der Stream veraltet ist. In diesem Fall sendet der Produzent Daten an den Service, empfängt jedoch keine Bestätigungen.	Benachrichtigung

Funktion	Description	Typ
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 <code>max_latency</code> überschreitet. Weitere Informationen finden Sie unter StreamDefinition/StreamInfo .	Benachrichtigung
StreamReadyFunc	Wird aufgerufen, wenn der Stream bereit ist.	Benachrichtigung
StreamUnderflowReportFunc	Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Benachrichtigung
DeviceCertToTokenFunc	Gibt das Verbindungszertifikat als Token zurück.	Plattformintegration
GetCurrentTimeFunc	Gibt die aktuelle Zeit zurück.	Plattformintegration
GetDeviceCertificateFunc	Gibt das Gerätezertifikat zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration

Funktion	Description	Typ
<code>GetDeviceFingerprintFunc</code>	Gibt die Geräteidentifikation zurück. Diese Funktion wird derzeit nicht verwendet und ist für die future Verwendung reserviert.	Plattformintegration
<code>GetRandomNumberFunc</code>	Gibt eine zufällige Zahl zwischen 0 und <code>RAND_MAX</code> zurück.	Plattformintegration
<code>GetSecurityTokenFunc</code>	Gibt das Sicherheitstoken zurück, das an die Funktionen übergeben wurde, die mit der Backend-API kommunizieren. Die Implementierung kann die serialisierten Werte für <code>AccessKeyId</code> , <code>SecretKeyId</code> und das Sitzungstoken angeben.	Plattformintegration
<code>LogPrintFunc</code>	Protokolliert eine Textzeile mit dem Tag und der Protokoll ebene. Weitere Informationen finden Sie unter <code>PlatformUtils.h</code> .	Plattformintegration

Für die Plattformintegrationsfunktionen aus der vorherigen Tabelle ist der letzte Parameter eine `ServiceCallContext`-Struktur mit den folgenden Feldern:

- `version`: Die Version der Struktur
- `callAfter`: Eine absolute Zeit, nach der die Funktion aufgerufen wird
- `timeout`: Die Zeitüberschreitung der Operation in Einheiten von 100 Nanosekunden
- `customData`: Ein benutzerdefinierter Wert, der an den Client übergeben wird

- `pAuthInfo`: Die Anmeldeinformationen für den Aufruf. Weitere Informationen finden Sie in der folgenden (`__AuthInfo`)-Struktur.

Die Autorisierungsinformationen werden mit der Struktur `__AuthInfo` bereitgestellt. Diese kann entweder serialisierte Anmeldeinformationen oder ein anbieterspezifisches Authentifizierungstoken enthalten. Diese Struktur enthält die folgenden Felder:

- `version`: Die Version der `__AuthInfo`-Struktur
- `type`: Ein `AUTH_INFO_TYPE`-Wert, über den der Typ der Anmeldeinformationen (Zertifikat oder Sicherheitstoken) definiert wird
- `data`: Ein Byte-Array mit den Authentifizierungsinformationen
- `size`: Die Größe des `data`-Parameters
- `expiration`: Die Ablaufdauer der Anmeldeinformationen in Einheiten von 100 Nanosekunden

Callback-Implementierungen, um das Streaming erneut zu versuchen

Das Kinesis Video Producer-SDK gibt über Callback-Funktionen den Streaming-Status an. Es wird empfohlen, die folgenden Callback-Mechanismen zu implementieren, um bei vorübergehenden Netzwerkproblemen, die während des Streamings aufgetreten sind, die Wiederherstellung zu gewährleisten.

- **Stream-Latenzdruck-Callback** — Dieser Callback-Mechanismus wird ausgelöst, wenn das SDK auf eine Stream-Latenz stößt. Dies geschieht, wenn die kumulierte Puffergröße größer ist als der `MAX_LATENCY`-Wert. Wenn der Stream erstellt wurde, legt die Streaming-Anwendung `MAX_LATENCY` auf den Standardwert von 60 Sekunden fest. Die typische Implementierung für diesen Callback besteht darin, die Verbindung zurückzusetzen. Sie können die Beispielimplementierung bei <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.c> Bedarf verwenden. Beachten Sie, dass es keine Möglichkeit gibt, Frames, die aufgrund eines Netzwerkausfalls nicht zugestellt wurden, in einem Sekundärspeicher 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 bei <https://github.com/awslabs/amazon-kinesis-video-streams-producer-c/blob/master/src/source/ConnectionStaleStateMachine.c> 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 [GetMedia](#) oder [GetMediaForFragmentList](#) zur Verfügung gestellt. Es wird zusammen mit den Fragmenten für die gesamte Dauer der Aufbewahrungszeit des Streams gespeichert. Ihre nutzenden Anwendungen können anhand der Metadaten lesen, verarbeiten und darauf reagieren [Sehen 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 über das [Zu Kinesis Video Streams hochladen](#) hinzugefügten Metadatenelemente unterscheiden sich von den markierenden APIs auf Stream-Ebene, die mit [TagStream](#), [UntagStream](#) und [ListTagsForStream](#) implementiert werden.

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).

Verwenden Sie IPv6 mit Kinesis Video Streams

Sie können Kinesis Video Streams so konfigurieren, dass IPv6 sowohl für Operationen auf der Steuerungsebene als auch auf der Datenebene verwendet wird. Dadurch können Ihre Anwendungen mithilfe von IPv6-Adressen über Dual-Stack-Endpunkte mit Kinesis Video Streams Streams-Diensten kommunizieren.

Note

Für die IPv6-Unterstützung sind spezielle SDK-Versionen und Konfigurationseinstellungen erforderlich. Stellen Sie sicher, dass Ihre Kinesis Video Streams AWS SDK- und SDK-Versionen IPv6-Dual-Stack-Endpunkte unterstützen. Dual-stack Endgeräte unterstützen sowohl IPv4- als auch IPv6-Verkehr und sind für einige Dienste in einigen Regionen verfügbar.

Kinesis Video Streams unterstützt IPv6 über Dual-Stack-Endpunkte sowohl für Hersteller- als auch für Verbraucheranwendungen. Sie können Ihre Anwendungen so konfigurieren, dass sie IPv6 für API-Aufrufe auf Steuerungsebene und Streaming-Operationen auf Datenebene verwenden.

Konfigurieren Sie die AWS SDK für IPv6

Wenn Sie das AWS SDK verwenden, um Kinesis Video Streams Streams-APIs auf der Steuerungsebene in Ihrem Produktions-Setup aufzurufen, können Sie IPv6 aktivieren, indem Sie

Dual-Stack-Endpunkte konfigurieren. Das AWS SDK bietet mehrere standardisierte Methoden zur Aktivierung von Dual-Stack-Endpunkten.

Important

Wenn Dual-Stack-Endpunkte aktiviert sind, versucht das SDK, Dual-Stack-Endpunkte für Netzwerkanfragen zu verwenden. Wenn für den Dienst oder die Region kein Dual-Stack-Endpoint existiert, schlägt die Anfrage fehl.

Verwenden Sie Umgebungsvariablen

Legen Sie die folgende Umgebungsvariable fest, um IPv6-Dual-Stack-Endpunkte zu aktivieren:

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Verwenden Sie die AWS Konfigurationsdatei

Fügen Sie Ihrer AWS Konfigurationsdatei (~/.aws/config) die folgende Einstellung hinzu:

```
[default]
use_dualstack_endpoint = true
```

Verwenden Sie JVM-Systemeigenschaften (nur Java- und Kotlin-SDKs)

Legen Sie für Java- und Kotlin-Anwendungen die folgende JVM-Systemeigenschaft fest:

```
-Daws.useDualstackEndpoint=true
```

Oder programmgesteuert in Ihrem Java-Code:

```
System.setProperty("aws.useDualstackEndpoint", "true");
```

SDK-Unterstützung

Die folgenden AWS SDKs unterstützen die Dual-Stack-Endpunktkonfiguration:

SDK	Unterstützt	Methoden zur Konfiguration
AWS CLI v2	Ja	Umgebungsvariable, Konfigurationsdatei
SDK für C++	Ja	Umgebungsvariable, Konfigurationsdatei
SDK for Go V2 (1.x)	Ja	Umgebungsvariable, Konfigurationsdatei
SDK für Java 2.x	Ja	Umgebungsvariable, Konfigurationsdatei, JVM-Eigenschaft
SDK für Java 1.x	Nein	Nicht unterstützt
SDK für 3.x JavaScript	Ja	Umgebungsvariable, Konfigurationsdatei
SDK für Python (Boto3)	Ja	Umgebungsvariable, Konfigurationsdatei

Nachdem Sie Dual-Stack-Endpunkte konfiguriert haben, verwendet das AWS SDK automatisch IPv6-Endpunkte, wenn es die APIs der Kinesis Video Streams Streams-Kontrollebene aufruft.

Das Kinesis Video Streams Producer SDK für IPv6 konfigurieren

Das Kinesis Video Streams Producer SDK bietet IPv6-Konfigurationsoptionen sowohl für den Betrieb auf der Steuerungsebene als auch auf der Datenebene. Diese Einstellungen funktionieren mit der AWS SDK-Dual-Stack-Endpunktkonfiguration.

Konfigurieren Sie das C/C ++ Producer SDK

Die Standard-Endpunkte und die DNS-Auflösungskette werden in der KVS Producer-C SDK-Version 1.6.0 implementiert. Es überprüft nacheinander jeden Ort, an dem Sie die Konfiguration für diese Parameter festlegen können, und wählt dann den ersten aus, den Sie festlegen. Die vordefinierte Reihenfolge lautet wie folgt:

Weitere Informationen zu den Producer-SDKs finden Sie unter [Producer-SDK für C](#), [Producer-SDK SDK for C++](#) und [verwandten Themen zum Producer-SDK](#).

Endpunktkonfiguration

1. Der `controlPlaneUrl` Parameter für `createAbstractDefaultCallbacksProvider`.
2. CMake-Parameter für die Endpunktkonfiguration: (-
DAWS_KVS_USE_LEGACY_ENDPOINT_ONLY=TRUE,-
DAWS_KVS_USE_DUAL_STACK_ENDPOINT_ONLY=TRUE)
3. Umgebungsvariablen: () `export AWS_USE_DUALSTACK_ENDPOINT=TRUE`
 - Falls ja `AWS_USE_DUALSTACK_ENDPOINT TRUE` (ohne Berücksichtigung von Groß- und Kleinschreibung), wird der Dual-Stack-Endpoint verwendet.
4. Andernfalls wird der Legacy-Endpoint erstellt und verwendet.

Bei den Werten 2, 3 und 4 wird der Endpoint auf der Grundlage der bereitgestellten Region erstellt `createAbstractDefaultCallbacksProvider`.

DNS-Filterung

Das KVS Producer SDK legt je nach Konfiguration den entsprechenden `CURLOPT_IPRESOLVE` Parameter fest:

1. CMake-Parameter für die DNS-Auflösung: (-DAWS_KVS_IPV4_ONLY=TRUE,, -
DAWS_KVS_IPV6_ONLY=TRUE) -DAWS_KVS_IPV4_AND_IPV6_ONLY=TRUE
2. Umgebungsvariablen (`export AWS_KVS_USE_IPV4=TRUE, export
AWS_KVS_USE_IPV6=TRUE`)
3. Andernfalls findet keine Filterung statt. Sowohl IPv4- als auch IPv6-IP-Adressen können verwendet werden, sofern sie von DNS zurückgegeben werden.

Note

Wenn die DNS-Filtereinstellungen so eingestellt sind, dass sie nach IPv6-IP-Adressen filtern, die SDK-Konfiguration jedoch den Legacy-Endpoint verwendet (gibt IPv4-only Adressen zurück), schlägt die Anfrage fehl.

Das C++ Producer SDK Version 3.5.0 verwendet das Producer-C SDK 1.6.0 für die KVS-API-Aufrufe.

Konfigurieren Sie das GStreamer-Plugin

Das GStreamer-Plugin verwendet das zugrunde liegende C Producer SDK, sodass die IPv6-Konfiguration automatisch durchgeführt wird, wenn Sie das C-SDK für IPv6 wie zuvor beschrieben konfigurieren.

Eine Änderung des Codes ist nicht erforderlich. Erstellen Sie einfach das SDK mit den CMake-Parametern oder legen Sie die entsprechenden Umgebungsvariablen fest, wie im vorherigen Abschnitt beschrieben.

Auflösung der Endpunkte auf der Datenebene

Verwenden Sie für Datenebenenoperationen die `GetDataEndpoint` API, um den entsprechenden Dual-Stack-Datenebenen-Endpunkt abzurufen. Der Dienst gibt je nach Anforderungs-URL den entsprechenden Endpunkt zurück.

Beispiel:

- Wenn die `GetDataEndpoint` API-Anfrage an den Legacy-Endpunkt gestellt wird, der auf `endet.amazonaws.com`, gibt Kinesis Video Streams die Legacy-Datenebenen-Endpunkte zurück, die auf `enden.amazonaws.com`
- Wenn die `GetDataEndpoint` API-Anfrage an den Dual-Stack-Endpunkt gestellt wird, der auf `endet.api.aws`, gibt Kinesis Video Streams den Dual-Stack-Datenebenen-Endpunkt zurück, der mit `endet.api.aws`

Konfigurieren Sie das AWS CLI für IPv6

Wenn Sie den Betrieb AWS CLI für Kinesis Video Streams verwenden (in der Regel für Machbarkeitsstudien), können Sie IPv6 aktivieren, indem Sie Dual-Stack-Endpunkte konfigurieren.

Verwenden Sie eine Umgebungsvariable

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Verwenden Sie die AWS Konfigurationsdatei

Fügen Sie Ihrer AWS CLI Konfigurationsdatei (`~/.aws/config`) Folgendes hinzu:

```
[default]
```

```
use_dualstack_endpoint = true
```

Beispiele für Konfigurationen

Beispiel für ein C SDK

Um das KVS Producer-C SDK im IPV6-only Modus zu erstellen und die Konfiguration der Umgebungsvariablen zu ignorieren, verwenden Sie die folgenden Befehle, um das SDK zu erstellen:

```
cmake .. -DAWS_KVS_USE_DUAL_STACK_ENDPOINT_ONLY=TRUE -DAWS_KVS_IPV6_ONLY=TRUE  
make -j
```

Note

Wenn Sie das SDK bereits erstellt haben, müssen Sie einen sauberen Build durchführen. Löschen Sie die vorhandenen Build-, Open-Source-Ordner und Dependency-Ordner, bevor Sie die Build-Befehle ausführen.

Überlegungen

Netzwerkanforderungen

- Stellen Sie sicher, dass Ihre Netzwerkinfrastruktur IPv6-Konnektivität unterstützt
- Stellen Sie sicher, dass Ihre Sicherheitsgruppen und Netzwerk-ACLs IPv6-Verkehr zulassen
- Testen Sie die Konnektivität zu AWS IPv6-Endpunkten von Ihrer Bereitstellungsumgebung aus
- Dual-stack Endpunkte sind für einige Dienste in einigen Regionen verfügbar. Überprüfen Sie die Verfügbarkeit für Ihre Zielregionen

SDK-Kompatibilität

- Stellen Sie sicher, dass Sie eine unterstützte AWS SDK-Version verwenden
- Das AWS SDK for Java 1.x unterstützt keine Dual-Stack-Endpunktconfiguration
- Für das SDK for Go 1.x (V1) müssen Sie das Laden aus der Konfigurationsdatei aktivieren, um die Einstellungen der gemeinsam genutzten Konfigurationsdatei verwenden zu können

Testen und Validieren

Bevor Sie IPv6-enabled Kinesis Video Streams Streams-Anwendungen für die Produktion bereitstellen:

- Testen Sie die Abläufe auf der Steuerungsebene (Erstellung, Löschung, Auflistung von Streams)
- Überprüfen Sie den Betrieb der Datenebene (Aufnahme und Nutzung von Videos)
- Überprüfen Sie die Leistung und Konnektivität in Ihrer Netzwerkumgebung
- Führen Sie Canary-Tests durch, um eine konsistente IPv6-Funktionalität sicherzustellen
- Testen Sie das Failover-Verhalten, wenn Dual-Stack-Endpunkte nicht verfügbar sind

Kunden, die von der Aktualisierung auf IPv6 betroffen sind

Wenn Sie IPv6 für Kinesis Video Streams aktivieren, gibt es mehrere Bereiche, in denen Sie möglicherweise Ihre vorhandenen Konfigurationen und Richtlinien aktualisieren müssen, um die kontinuierliche Funktionalität sicherzustellen.

IAM-Richtlinien und IP-Adressfilterung

Wenn Sie die Quell-IP-Adressfilterung in Ihren IAM-Benutzerrichtlinien, Rollenrichtlinien oder ressourcenbasierten Richtlinien verwenden, müssen Sie diese Richtlinien aktualisieren, um IPv6-Adressbereiche einzubeziehen.

Important

Bestehende IAM-Richtlinien, die IPv4-CIDR-Blocks `IpAddress` oder `NotIpAddress` - Bedingungen verwenden, funktionieren nicht automatisch mit IPv6-Adressen. Sie müssen explizit IPv6-Bereiche hinzufügen, um die Zugriffskontrolle aufrechtzuerhalten.

Beispiel für ein IAM-Richtlinienupdate für IPv6:

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

```

    "Action": "kinesisvideo:*",
    "Resource": "*",
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24",
          "203.0.113.0/24",
          "2001:db8::/32"
        ]
      }
    }
  }
}

```

Protokollierung und Überwachung

IPv6-Adressen haben ein anderes Format als IPv4-Adressen, was sich auf Ihre Protokollierungs-, Überwachungs- und Analysesysteme auswirken kann.

Protokolle

Protokolle enthalten IPv6-Adressen im `sourceIPAddress` Feld, wenn Anfragen über IPv6 gestellt werden. Aktualisieren Sie Ihre Tools und Skripts zum Analysieren von Protokollen, sodass sie mit IPv6-Adressformaten umgehen können.

Beispiel für eine IPv6-Adresse in Protokollen:

```

{
  "sourceIPAddress": "2001:db8::1",
  "eventName": "CreateStream",
  "eventSource": "kinesisvideo.amazonaws.com"
}

```

Fehlerbehebung

Häufige Probleme

- Verbindungsfehler — Überprüfen Sie die IPv6-Netzwerkonnektivität und die DNS-Auflösung
- SDK-Fehler — Stellen Sie sicher, dass Sie kompatible SDK-Versionen verwenden, die Dual-Stack-Endpunkte unterstützen

- Authentifizierungsprobleme — Stellen Sie sicher, dass IAM-Richtlinien und Anmeldeinformationen mit IPv6-Endpunkten funktionieren
- Endpunkt nicht verfügbar — Wenn kein Dual-Stack-Endpunkt für den Service oder die Region existiert, schlagen Anfragen fehl

Schritte zur Überprüfung

- Überprüfen Sie, ob dies festgelegt `AWS_USE_DUALSTACK_ENDPOINT=true`
`use_dualstack_endpoint = true` ist oder sich in Ihrer Konfigurationsdatei befindet
- Stellen Sie sicher, dass die IPv6-Konfigurationsflags des Kinesis Video Streams SDK richtig gesetzt sind
- Testen Sie die Netzwerkkonnektivität zu IPv6-Endpunkten AWS
- Überprüfen Sie die Anwendungsprotokolle auf Fehlermeldungen IPv6-specific
- Vergewissern Sie sich, dass Ihre Region Dual-Stack-Endpunkte für Kinesis Video Streams unterstützt

Überprüfung der Konfiguration

Sie können Ihre Dual-Stack-Endpunktkonfiguration überprüfen, indem Sie Folgendes überprüfen:

- Umgebungsvariablen: `echo $AWS_USE_DUALSTACK_ENDPOINT`
- AWS Konfigurationsdatei: `cat ~/.aws/config | grep use_dualstack_endpoint`
- JVM-Eigenschaften (Java): Überprüfen Sie die Systemeigenschaften in Ihren Anwendungsprotokollen

Kinesis-Videostreams-Wiedergabe

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-Videostreams zu erstellen. `GetMedia` ist eine Echtzeit-API mit niedriger Latenz. Um einen Player zu erstellen, der verwendet wird `GetMedia`, musst du ihn selbst erstellen. Informationen zum Entwickeln einer Anwendung, mit der ein Kinesis-Videostream angezeigt wird `GetMedia`, finden Sie unter [Streamen Sie mithilfe der Parser-Bibliothek](#).
- **HLS** — [HTTP Live Streaming \(HLS\)](#) ist ein branchenübliches HTTP-based Medienstreaming-Kommunikationsprotokoll. Sie können HLS verwenden, um einen Kinesis-Videostream anzusehen, entweder für die Live-Wiedergabe oder zum Ansehen archivierter Videos.

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 Player eines Drittanbieters (z. B. [Google Shaka Player](#)) verwenden, um den Videostream anzuzeigen, indem Sie die URL der HLS-Streamingsitzung entweder programmgesteuert [Video.js](#) oder manuell angeben. Sie können Videos auch wiedergeben, indem Sie die URL der HLS-Streamingsitzung in die Adressleiste der Browser [Apple Safari](#) oder Microsoft Edge eingeben. <https://www.microsoft.com/en-us/edge>

- **MPEG-DASH**— [Dynamic Adaptive Streaming over HTTP \(DASH\)](#), auch bekannt als MPEG-DASH, ist ein Streaming-Protokoll mit adaptiver Bitrate, das das qualitativ hochwertige Streaming von Medieninhalten über das Internet ermöglicht, die von herkömmlichen HTTP-Webservern bereitgestellt werden.

Es kann MPEG-DASH für die Live-Wiedergabe verwendet werden. 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 (z. B. [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 die Videowiedergabe und den Track](#)
- [Videowiedergabe mit HLS](#)
- [Videowiedergabe mit MPEG-DASH](#)

Anforderungen an die Videowiedergabe und den Track

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

Themen

- [GetClip Anforderungen](#)
- [GetDASHStreamingSessionURL Anforderungen](#)
- [GetHLSStreamingSessionURL Anforderungen](#)
- [GetImages Anforderungen](#)

GetClip Anforderungen

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

Beschreibung von Track 1	Codec-ID für Track 1	Beschreibung von Track 2	Track-2-Codec-ID
H.264 Video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 Video	V_MPEG4/ISO/AVC	AAC-Audio	A_AAC
H.264 Video	V_MPEG4/ISO/AVC	G.711 Audio (nur) A-Law	A_MS/ACM
H.265 Video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 Video	V_MPEGH/ISO/HEVC	AAC-Audio	A_AAC

⚠ Important

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

⚠ Important

Änderungen nachverfolgen werden nicht unterstützt. Die Titel müssen auf allen abgefragten Medien konsistent bleiben. Ein Fehler wird zurückgegeben, wenn die Fragmente im Stream von nur Video zu Audio und Video wechseln oder wenn eine AAC-Audiospur in eine A-Law Audiospur geändert wird.

GetDASHStreamingSessionURL Anforderungen

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

Beschreibung von Track 1	Codec-ID für Track 1	Beschreibung von Track 2	Track-2-Codec-ID
H.264 Video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 Video	V_MPEG4/ISO/AVC	AAC-Audio	A_AAC
H.264 Video	V_MPEG4/ISO/AVC	G.711 Audio (nur) A-Law	A_MS/ACM
H.264 Video	V_MPEG4/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-Daten (CPD) enthalten codecspezifische Initialisierungsinformationen wie Bildrate, Auflösung und Kodierungsprofil, die für die korrekte Decodierung des Fragments erforderlich sind. CPD-Änderungen werden während einer Streaming-Sitzung nicht unterstützt. Die CPD muss auf allen abgefragten Medien konsistent bleiben.

⚠ Important

Änderungen nachverfolgen wird nicht unterstützt. Die Titel müssen auf allen abgefragten Medien konsistent bleiben. Das Streaming schlägt fehl, wenn die Fragmente im Stream von nur Video zu Audio und Video wechseln oder wenn eine AAC-Audiospur in eine A-Law Audiospur geändert wird.

GetHLSStreamingSessionURL Anforderungen

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

HLS Mp4

Beschreibung von Track 1	Codec-ID für Track 1	Beschreibung von Track 2	Track-2-Codec-ID
H.264 Video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 Video	V_MPEG4/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 für Track 1	Beschreibung von Track 2	Track-2-Codec-ID
H.264 Video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 Video	V_MPEG4/ISO/AVC	AAC-Audio	A_AAC
AAC-Audio	A_AAC	N/A	N/A

 Note

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

 Important

Änderungen nachverfolgen werden nicht unterstützt. Die Titel müssen auf allen abgefragten Medien konsistent bleiben. Das Streaming schlägt fehl, wenn die Fragmente im Stream von nur Video zu Audio und Video wechseln oder wenn eine AAC-Audiospur in eine A-Law Audiospur geändert wird.

GetImages Anforderungen

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

 Note

Das GetImages Medium sollte eine Videospur in Track 1 enthalten.

Videowiedergabe mit HLS

[HTTP Live Streaming \(HLS\)](#) ist ein branchenübliches Kommunikationsprotokoll für HTTP-based 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 Player eines Drittanbieters (z. B. [Google Shaka Player](#)) verwenden, um den Videostream anzuzeigen, indem Sie die URL der HLS-Streamingsitzung entweder programmgesteuert [Video.js](#) oder manuell angeben. Sie können Videos auch wiedergeben, indem Sie die URL der HLS-Streamingsitzung in die Adressleiste der Browser [Apple Safari](#) oder Microsoft Edge eingeben. <https://www.microsoft.com/en-us/edge>

Um einen Kinesis-Videostream mit HLS anzusehen, erstellen Sie zunächst eine Streaming-Sitzung mit [GetHLSStreamingSessionURL](#). 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. Informationen zu [GetHLSStreamingSessionURL](#) den spezifischen Anforderungen für das Hochladen finden Sie unter.

Themen

- [Benutze das AWS CLI um eine URL für eine HLS-Streaming-Sitzung abzurufen](#)
- [Beispiel: Verwenden Sie HLS in HTML und JavaScript](#)
- [Behebung von HLS-Problemen](#)

Benutze das AWS CLI um eine URL für eine HLS-Streaming-Sitzung abzurufen

Das folgende Verfahren zeigt, wie Sie die AWS CLI verwenden, um eine HLS-Streaming-Sitzungs-URL für einen Kinesis-Videostream zu generieren.

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

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

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

1. Geben Sie Folgendes in das Terminal ein:

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

Sie erhalten eine Antwort, die wie folgt aussieht:

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

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

Live

Für die 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 ein Scrubber-Steuerelement zur Auswahl der anzuzeigenden Position im Wiedergabefenster vorhanden ist.

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 an einer bestimmten Startzeit. Die HLS-Medien-Playlist wird ebenfalls kontinuierlich 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 der Erkennung eines Ereignisses zu starten und das Livestreamen von Medien fortzusetzen, die zum Zeitpunkt der Sitzungserstellung noch nicht aufgenommen wurden.

Ermitteln Sie einen Startzeitstempel.

Für dieses Beispiel verwenden wir die Unix-Epoch-Zeit im Sekundenformat. Weitere Informationen zur [Formatierung von Zeitstempeln finden Sie im](#) Abschnitt Zeitstempel im AWS Command Line Interface Benutzerhandbuch.

Ein [UnixTime.org](https://unixtime.org/) Konvertierungstool finden Sie unter.

- 1708471800 entspricht dem 20. Februar 2024 um 15:30 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 Wiedergabe auf Abruf enthält die HLS-Medien-Playlist die vom HLS-Fragmentselektor angegebenen Medien. Wenn diese Art von Sitzung in einem Media Player abgespielt wird,

zeigt die Benutzeroberfläche in der Regel ein Scrubber-Steuerelement an, mit dem Sie die Position im Wiedergabefenster auswählen können, die angezeigt werden soll.

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

Für dieses Beispiel verwenden wir die Unix-Epoch-Zeit im Sekundenformat. Weitere Informationen zur [Formatierung von Zeitstempeln finden Sie im](#) Abschnitt Zeitstempel im AWS Command Line Interface Benutzerhandbuch.

Ein [UnixTime.org](#) Konvertierungstool finden Sie unter.

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

Rufen Sie die `GetHLSStreamingSessionURL` API mit dem angegebenen `ON_DEMAND` Wiedergabemodus und einem angegebenen HLS-Fragment-Selector auf. https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/API_reader_GetHLSStreamingSessionURL.html#KinesisVideo-reader_GetHLSStreamingSessionURL-request-HLSFragmentSelector

```
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 innerhalb von 24 Stunden voneinander entfernt sein, wie in der Dokumentation angegeben. [HLSTimestampRange](#)

Sie erhalten eine Antwort, die wie folgt aussieht:

```
{
```

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

Important

Teilen oder speichern Sie dieses Token nicht an Orten, an denen eine nicht autorisierte Entität darauf zugreifen könnte. Das Token bietet Zugriff auf den Inhalt des Streams. Schützen Sie das Token mit den gleichen Maßnahmen, die Sie 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 Browser Apple Safari oder Microsoft Edge eingeben.

Beispiel: Verwenden Sie HLS in HTML und JavaScript

Das folgende Beispiel zeigt, wie das AWS SDK für JavaScript Version 2 verwendet wird, um eine HLS-Streaming-Sitzung für einen Kinesis-Videostream abzurufen und auf einer Webseite wiederzugeben. 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 der MPEG-DASH Ausgabe aus dem Amazon Kinesis-Videostream. Die Beispielseite enthält Eingabefelder für die folgenden Parameter:

- AWS-Region: Die Region, in der sich Ihr Kinesis-Videostream befindet
- Stream-Name: Der Name Ihres Kinesis-Videostreams
- Wiedergabemodus: Der HLS-Wiedergabemodus (LIVE, LIVE_REPLAY oder ON_DEMAND)

- **Fragmentauswahltyp:** Die Methode, mit der Fragmente ausgewählt werden (SERVER_TIMESTAMP oder PRODUCER_TIMESTAMP)
- **Fragmentnummer:** Die Startnummer des Fragments (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 Anfrage für eine HLS-Streaming-Sitzung zu erstellen, die auf der Seite angezeigt wird.

Erläuterung der Themen im Code:

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

Importiere AWS SDK JavaScript für Browser

Fügen Sie auf der Webseite das folgende Skript-Tag ein, 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 des](#) AWS SDK.

Richten Sie den Kinesis Video Streams-Client ein

Um mit HLS auf Streaming-Videos zuzugreifen, erstellen und konfigurieren Sie zunächst den Kinesis Video Streams-Client. Weitere Authentifizierungsmethoden finden Sie unter Anmeldeinformationen in einem Webbrowser <https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-credentials-browser.html> 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.

Ruft den Endpunkt für die HLS-Wiedergabe ab

Verwenden Sie den Kinesis Video Streams-Client, um die [GetDataEndpoint](#) API zum Abrufen des Endpunkts aufzurufen.

```
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.

Richten Sie den Kinesis Video Streams-Client für archivierte Medien ein

Geben Sie in der Client-Konfiguration für den Kinesis Video Streams-Client für archivierte Medien den Endpunkt an, den Sie im vorherigen Schritt erhalten 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-Client für archivierte Medien, um die [GetHLSStreamingSessionURL](#) API zum Abrufen der HLS-Wiedergabe-URL aufzurufen.

```
const getHLSStreamingSessionURLOptions = {
  StreamName: 'YourStreamName',
  PlaybackMode: 'LIVE'
};
const getHLSStreamingSessionURLResponse = await kinesisisVideoArchivedMediaClient
  .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 die [Video.js](#) zugehörigen CSS-Klassen in unser Browserskript 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 Browserskript 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 hls.js Player, geben Sie ihm die HLS-URL und sagen Sie ihm, 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();  
});
```

Behebung von HLS-Problemen

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

Problembereiche

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

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

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

Um das Problem zu beheben, versuchen Sie Folgendes:

- Stellen Sie fest, ob der Videostream in der Kinesis Video Streams-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, da er zu häufig Anfragen nach Videofragmenten stellt.
- 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-Streamingsitzung angeben. Probieren Sie verschiedene Kombinationen von Werten für die folgenden Parameter aus:
 - `ContainerFormat`
 - `PlaybackMode`
 - `FragmentSelectorType`
 - `DiscontinuityMode`
 - `MaxMediaPlaylistFragmentResults`

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

- Stellen Sie sicher, dass jedes Fragment über eine konsistente Anzahl von Tracks verfügt. Stellen Sie sicher, dass sich die Fragmente im Stream nicht ändern, weil sie sowohl eine Audio- als auch eine Videospur enthalten und nur eine Videospur haben. Stellen Sie außerdem sicher, dass sich die Encoder-Einstellungen (Auflösung und Bildrate) nicht zwischen den 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 anpassen, so 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 Senden. Ein Videostream mit einer niedrigeren Bitrate verfügt in der Regel jedoch über eine geringere Videoqualität.

- **Anzahl der Fragmente in Medien-Playlisten** — Ein Player, der auf Latenz reagiert, sollte nur die neuesten Fragmente einer Medien-Playlist laden. Die meisten Player 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-Größe ist es möglich, dass ein Fragment während der Wiedergabe übersprungen wird, wenn es beim Hinzufügen neuer Fragmente zur Playlist zu einer Verzögerung kommt oder wenn der Player eine aktualisierte Playlist verzögert erhält. Wir empfehlen, 3—5 Fragmente zu verwenden und einen Player zu verwenden, der so konfiguriert ist, dass nur die neuesten Fragmente aus einer Playlist geladen werden.
- **Player-Puffergröße** — Die meisten Videoplayer haben eine konfigurierbare Mindestpufferdauer, in der Regel mit einer Standardeinstellung von 10 Sekunden. Für die niedrigste Latenz können Sie diesen Wert auf 0 Sekunden setzen. Dies bedeutet jedoch, dass der Player bei Verzögerungen einen Rebuffer durchführt und Fragmente erzeugt, da der Player keinen Puffer hat, um die Verzögerung zu absorbieren.
- **„Aufholen“ durch den Player** — Videoplayer fangen die Wiedergabe in der Regel nicht automatisch bis zum Anfang des Videopuffers ab, wenn der Puffer voll ist, z. B. wenn ein verzögertes Fragment dazu führt, dass ein Rückstand von Fragmenten abgespielt wird. 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 anzusehen MPEG-DASH, erstellen Sie zunächst eine Streaming-Sitzung mit [GetDASHStreamingSessionURL](#). Diese Aktion gibt eine URL (die ein Sitzungstoken enthält) 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 “GetDASHStreamingSessionURL”](#).
- Der Datenerhaltzeitraum muss größer als 0 sein.

- Die Videospur jedes Fragments muss private Codec-Daten im Format Advanced Video Coding (AVC) für das H.264 Format und HEVC für das Format enthalten. H.265 [Weitere Informationen finden Sie in der Spezifikation 14496-15MPEG-4. ISO/IEC](#) 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 im MS Wave-Format enthalten.](#)

Beispiel: Verwendung in HTML und MPEG-DASH 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 zur MPEG-DASH Wiedergabe ab](#)
- [Rufen Sie die URL der MPEG-DASH Streaming-Sitzung ab](#)
- [Zeigen Sie das Streaming-Video mit MPEG-DASH Wiedergabe an](#)
- [Beispiel abgeschlossen](#)

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

Um auf Streaming-Video zuzugreifen MPEG-DASH, müssen Sie zunächst den Kinesis Video Streams Streams-Client (zum Abrufen des Service-Endpunkts) und den archivierten Medienclient (zum Abrufen der MPEG-DASH Streaming-Sitzung) erstellen und konfigurieren. 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 zur 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

[GetDASHStreamingSessionURL](#) 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 URL der MPEG-DASH Streaming-Sitzung haben, stellen Sie sie dem Videoplayer zur Verfügung. 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');
}
```

Beispiel abgeschlossen

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 Kunden mithilfe von Amazon Simple Notification Service (Amazon SNS) -Benachrichtigungen.

Note

Amazon Kinesis Video Streams verwendet Amazon SNS-Standardthemen für die Kommunikation. FIFO-Themen werden derzeit nicht unterstützt.

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

Themen

- [Benachrichtigungskonfigurationen verwalten](#)
- [Über die MKV-Tags des Herstellers](#)
- [Amazon SNS-Nachrichten](#)
- [Cross-account Veröffentlichung von Amazon SNS-Benachrichtigungen](#)

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-Entwicklerhandbuch.

 Note

Nach der Aktualisierung der Benachrichtigungskonfiguration dauert es mindestens eine Minute, um die Benachrichtigung zu initiieren. Warten Sie PutMedia nach dem Aktualisierungsauftrag mindestens eine Minute, bevor Sie den Vorgang aufrufen.

DescribeNotificationConfiguration

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

Über die MKV-Tags des Herstellers

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

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

Syntax für Hersteller-MKV-Tags

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

```
| + Name: CUSTOM_KEY_2 // Max 128 bytes  
| + String: CUSTOM_VALUE_2 // Max 256 bytes
```

Grenzwerte für MKV-Tags

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

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

Amazon SNS-Nachrichten

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

Themen

- [Payload zum Thema Amazon SNS](#)
- [Sehen Sie sich Ihre Amazon SNS-Nachrichten an](#)

Payload zum Thema Amazon SNS

Jede Benachrichtigung, die über den vorherigen Workflow initiiert wurde, liefert die Payload für das Amazon SNS-Thema, wie im folgenden Beispiel gezeigt. Bei diesem Beispiel handelt es sich um eine Amazon SNS-Nachricht, die nach dem Verbrauch 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
  }
}

```

Sehen Sie sich Ihre Amazon SNS-Nachrichten an

Sie können Nachrichten nicht direkt aus einem Amazon 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.

Um Ihre Amazon SNS-Nachrichten mithilfe von Amazon SQS anzuzeigen

1. Erstellen Sie eine [Amazon SQS-Warteschlange](#).

2. Öffnen Sie in der das Amazon SNS-Thema AWS-Managementkonsole, das als Ziel unter festgelegt wurde. `NotificationConfiguration`
3. Wählen Sie `Create Subscription` und dann die Amazon 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-Konsole die Amazon SQS-Warteschlange aus und wählen Sie dann `Nachrichten senden und empfangen` für die Amazon SQS-Warteschlange aus.
6. Umfrage nach Nachrichten. Dieser Befehl sollte alle von der `PutMedia` Sitzung generierten Benachrichtigungen anzeigen. Informationen zu Polling finden Sie unter [Amazon SQS short and long polling](#).

Cross-account Veröffentlichung von Amazon SNS-Benachrichtigungen

Um Amazon SNS-Benachrichtigungen zu einem Thema in einem anderen AWS Konto zu veröffentlichen, müssen Sie sowohl identitätsbasierte als auch ressourcenbasierte Richtlinien konfigurieren. Diese Konfiguration ermöglicht es Kinesis Video Streams, Benachrichtigungen von einem Konto zu einem Amazon SNS-Thema in einem anderen Konto zu veröffentlichen.

Identity-based Konfiguration der Richtlinie

Die IAM-Rolle oder der IAM-Benutzer, der die `PutMedia` API aufruft, muss über `sns:Publish` Berechtigungen für das kontoübergreifende Amazon SNS-Thema verfügen. Fügen Sie der identitätsbasierten Richtlinie die folgende Richtlinienerklärung hinzu:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-east-1:123456789012:stream/*"
    },
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-1:123456789012:*"
      ]
    }
  ]
}

```

Resource-based Konfiguration der Richtlinie

Das Amazon SNS-Thema im Zielkonto muss über eine ressourcenbasierte Zugriffsrichtlinie verfügen, die es dem Quellkonto ermöglicht, Nachrichten zu veröffentlichen. Konfigurieren Sie die Zugriffsrichtlinie für Amazon SNS-Themen wie folgt:

JSON

```

{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic-name"
    }
  ]
}

```

<kvs_streams_account_id> Ersetzen Sie es durch die AWS Konto-ID, unter der sich Ihre Kinesis Video Streams-Streams befinden, und <sns_topic_arn> durch den ARN Ihres Amazon SNS-Themas.

Anforderungen und Überlegungen

- Sowohl die identitätsbasierte Richtlinie (im Quellkonto) als auch die ressourcenbasierte Richtlinie (im Zielkonto) müssen konfiguriert werden, damit die kontoübergreifende Veröffentlichung funktioniert.
- Die für PutMedia Operationen verwendete IAM-Rolle muss `sns:Publish` Berechtigungen enthalten, auch wenn IoT-Zertifikate mit Rollenaliasen verwendet werden.

Extrahieren Sie Bilder aus Videostreams

Sie können Amazon Kinesis Video Streams-APIs und -SDKs verwenden, um On-Demand-Bildextraktionen und automatische Bildextraktionen in Echtzeit aus Ihren Kinesis-Videostreams durchzuführen. Sie können diese Bilder für erweiterte Wiedergabeanwendungen wie Miniaturansichten oder verbessertes Scrubbing oder für die Verwendung in Workflows für maschinelles Lernen verwenden.

Kinesis Video Streams unterstützt das Extrahieren von Bildern aus Videostreams auf zwei Arten:

- [On-demand Bilderzeugung](#) — Verwenden Sie die [GetImages](#) API, um ein einzelnes Bild oder mehrere Bilder aus einem in Kinesis Video Streams gespeicherten Video zu extrahieren.
- [the section called “Real-time Generierung von Bildern”](#) - Konfigurieren Sie Kinesis Video Streams so, dass Bilder automatisch in Echtzeit aus Videodaten extrahiert werden, und zwar auf der Grundlage von Fragment-Tags aus dem Video, das aufgenommen wird, und die Bilder an einen S3-Bucket übertragen.

Automatisierte Bildgenerierung in Echtzeit

Amazon Kinesis Video Streams bietet die Möglichkeit, Bilder zu transkodieren und bereitzustellen. Kinesis Video Streams extrahiert automatisch Bilder aus Videodaten in Echtzeit und übermittelt die Bilder an Ihren angegebenen Amazon S3-Bucket. Die Implementierung einer automatisierten Bildextraktion in Echtzeit umfasst die folgenden Schritte:

- Erstellen eines S3-Buckets zum Empfangen der generierten Bilder.
- Konfiguration der [ImageGenerationConfiguration](#) Stream-Eigenschaft, die Kinesis Video Streams mitteilt, wie die Bilder erstellt und wohin sie gesendet werden sollen.
- Tags zur Bildgenerierung hinzufügen — Kinesis Video Streams generiert Bilder nur mithilfe von Fragmenten, die das Bilderzeugungs-Tag tragen. Diese Tags werden beim Hochladen von Videos mithilfe des Kinesis Video Streams Producer SDK zusammen mit der Methode hinzugefügt.
`putKinesisVideoEventMetadata`

Die folgenden Verfahren enthalten Anweisungen zum Ausführen der einzelnen Schritte.

Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, stellen Sie sicher, dass die Rolle, die die `PutMedia` Aufrufe ausführt (Uploader), über die folgenden Berechtigungen verfügt, die

zum Verschlüsseln und Entschlüsseln von Daten sowie für den Zugriff auf den Amazon S3-Bucket erforderlich sind.

- kms:Encrypt
- kms:GenerateDataKey
- kms:Decrypt
- s3:PutObject

Weitere Informationen finden Sie unter [the section called “Wie fange ich mit der serverseitigen Verschlüsselung an?”](#).

Um das Ziel der generierten Bilder zu konfigurieren

1. Erstellen Sie einen S3-Ziel-Bucket, an den die Bilder gesendet werden.

Folgen Sie dem [Amazon S3-Benutzerhandbuch](#), um einen Amazon S3-Bucket zu erstellen.

Notieren Sie sich die URI des Buckets, die Sie im nächsten Schritt benötigen, wenn Sie die Konfiguration für die Image-Generierung des Streams aktualisieren.

2. Stellen Sie sicher, dass Sie das AWS CLI installiert und konfiguriert haben. Weitere Informationen finden Sie im [AWS Command Line Interface Benutzerhandbuch für Version 2](#).
3. Erstellen Sie eine neue Datei `update-image-generation-input.json` mit dem Namen und dem folgenden Inhalt als Eingabe. Aktualisieren Sie die Platzhalterwerte mit den Werten, die Sie verwenden möchten. Die unterstützten Höchst- und Mindestwerte finden Sie in der [UpdateImageGenerationConfiguration](#) API.

```
{
  "StreamName": "demo-stream",
  "ImageGenerationConfiguration": {
    "Status": "ENABLED",
    "DestinationConfig": {
      "DestinationRegion": "us-east-1",
      "Uri": "s3://my-bucket-name"
    },
    "SamplingInterval": 200,
    "ImageSelectorType": "PRODUCER_TIMESTAMP",
    "Format": "JPEG",
    "FormatConfig": {
      "JPEGQuality": "80"
    }
  }
}
```

```
    },  
    "WidthPixels": 320,  
    "HeightPixels": 240  
  }  
}
```

4. Aktualisieren Sie die Konfiguration der Bildgenerierung des Streams mithilfe der [UpdateImageGenerationConfiguration](#) API und hängen Sie die JSON-Datei als Eingabe an, wie im folgenden Befehl gezeigt. Beachten Sie, dass der Dateipfad auf die Datei im aktuellen Verzeichnis verweist.

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

5. Bei Erfolg wird eine leere Antwort zurückgegeben und nichts wird in Ihrem Terminal gedruckt.

 Note

Nach der Aktualisierung der Konfiguration zur Bildgenerierung dauert es mindestens 1 Minute, um den Arbeitsablauf zur Bildgenerierung zu starten. Warten Sie mindestens 1 Minute, bevor Sie das Video in Ihren Stream hochladen.

6. Überprüfen Sie die Konfigurationseinstellungen. Verwenden Sie die AWS CLI , um die [DescribeImageGenerationConfiguration](#) API für Ihren Stream aufzurufen.

```
aws kinesisvideo describe-image-generation-configuration \  
  --stream-name "demo-stream"
```

Kinesis Video Streams generiert und liefert nur Bilder für Fragmente, die das Image Generation Tag tragen. Alle zusätzlichen Fragment-Metadaten, die zusammen mit den Amazon S3-Image-Generierungs-Tags bereitgestellt werden, werden als Amazon S3-Metadaten gespeichert.

 Note

Tags zur Bildgenerierung beziehen sich auf Fragment-Metadaten-Tags und nicht auf Tags auf Stream-Ebene.

⚠ Important

Tags zur Bilderzeugung werden auf das Tag-Limit für Fragment-Metadaten angerechnet. Weitere Informationen finden Sie unter [the section called “Kontingente für Streaming-Metadaten”](#).

Im Folgenden finden Sie ein Beispiel dafür, wie die Struktur der Fragment-Metadaten-Tags mithilfe des `mkvinfo` Dienstprogramms aussieht. Das Tag zur Bilderzeugung ist ein einfaches MKV-Tag mit einem Schlüssel von `AWS_KINESISVIDEO_IMAGE_GENERATION` und ohne Wert. Weitere Informationen finden Sie unter Beispiel für [Video-Tags](#) in der Matroska-Dokumentation.

```
|+ 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

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

Hinzufügen von Tags zur Bildgenerierung zu Fragmenten

Kinesis Video Streams generiert und liefert Bilder nur für Fragmente, die mit dem Tag zur Bildgenerierung versehen sind. Kinesis Video Streams erkennt diese speziellen MKV-Tags und initiiert den Bildgenerierungs-Workflow auf der Grundlage der Bildverarbeitungskonfiguration des Streams.

Wenn Sie das Kinesis Video Streams Producer SDK zum Hochladen von Medien verwenden, verwenden Sie die `putKinesisVideoEventMetadata` Methode, um jedem Fragment, das Sie

taggen möchten, das Tag für die Bildgenerierung hinzuzufügen. Ein neues Fragment beginnt, wenn `putFrame` es mit einem Frame aufgerufen wird, der das `keyframe` Flag enthält.

Wenn Sie ein aufgezeichnetes Video hochladen, wird es je nach Netzwerkgeschwindigkeit möglicherweise mit einer anderen Geschwindigkeit hochgeladen als der, mit der es aufgenommen wurde. Wir empfehlen, dass Sie den Producer-Zeitstempel verwenden, um die Bildgenerierung zu konfigurieren, wenn Sie Bilder in regelmäßigen Abständen auf der Grundlage der ursprünglichen Zeitstempel des Videos generieren möchten und nicht die Serverzeitstempel verwenden möchten, die auf der Geschwindigkeit basieren, mit der Amazon Kinesis Video Streams Ihr Video empfangen hat.

Ein vollständiges Beispiel für diesen Code finden Sie im Codebeispiel unter [VideoOnlyRealtimeStreamingSample](#). [GitHub](#)

```
// 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;

Frame eofr = EOFR_FRAME_INITIALIZER;

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));

    // 1. End the previous fragment
    if (frame.flags == FRAME_FLAG_KEY_FRAME && !firstFrame) {
        putKinesisVideoFrame(streamHandle, &eofr);
    }

    // 2. putFrame call
    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));

    if (frame.flags == FRAME_FLAG_KEY_FRAME) {
        // 3. Adding the image generation tag
```

```

        CHK_STATUS(putKinesisVideoEventMetadata(streamHandle,
        STREAM_EVENT_TYPE_IMAGE_GENERATION, NULL));

        // 4. Adding fragment metadata
        for (n = 1; n <= 5; n++) {
            SNPRINTF(metadataKey, METADATA_MAX_KEY_LENGTH, "SAMPLE_KEY_%d", n);
            SNPRINTF(metadataValue, METADATA_MAX_VALUE_LENGTH, "SAMPLE_VALUE_%d",
            frame.index + n);
            CHK_STATUS(putKinesisVideoFragmentMetadata(streamHandle, metadataKey,
            metadataValue, FALSE));
        }
    }
    defaultThreadSleep(frame.duration);

    frame.decodingTs += frame.duration;
    frame.presentationTs = frame.decodingTs;
    frameIndex++;
    fileIndex++;
    fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
    firstFrame = TRUE;
}

// 5. End the final fragment
putKinesisVideoFrame(streamHandle, &eofr);

```

Die Elemente des Beispielcodes zum Einrichten von Beispielframes werden wie folgt erklärt:

1. Jedes Fragment muss mit einem Ende von fragment (`eofr`) enden. Diese Anweisung besagt, dass jedes Mal, wenn ein neuer Keyframe empfangen wird, der den Beginn des nächsten Frames signalisiert, ein gesetzt wird, `eofr` bevor das nächste Bild zum Stream hinzugefügt wird.
2. Fügt den aktuellen Frame in den Stream ein.
3. Fügen Sie das Tag zur Bilderzeugung hinzu. Die `putKinesisVideoEventMetadata` Methode kann jederzeit nach dem `putFrame(keyFrame)` Aufruf und vor dem aufgerufen werden `putFrame(eofr)`. Sie darf nur maximal einmal pro Fragment aufgerufen werden. Da jedes Fragment nur einen Keyframe haben wird, rufen wir ihn der Einfachheit halber zu diesem Zeitpunkt auf. Der Rückgabewert von `putKinesisVideoEventMetadata` wird auf einen Erfolgscode (0) überprüft.
4. Fügen Sie weitere benutzerdefinierte Fragment-Metadaten hinzu, die Kinesis Video Streams in Amazon S3-Objektmetadaten umwandelt.
5. Beenden Sie das letzte Fragment in dieser Upload-Sitzung.

Verwenden der Beispiele zum Hinzufügen von Tags zur Bildgenerierung

Sie können das `kvs_gstreamer_audio_video_sample` im C++ Producer SDK verwenden, wenn Sie eine Befehlszeilenoption zum Hinzufügen von Tags zur Bildgenerierung benötigen. Aktivieren Sie diese Funktion, indem Sie entweder das `-e both` Argument `-e image` oder hinzufügen, wie im folgenden Beispiel gezeigt.

```
./kvs_gstreamer_audio_video_sample stream-name \  
-f video-to-upload.mp4 \  
-e both
```

Weitere Informationen zu dieser Beispielanwendung finden Sie im [Amazon Kinesis Video Streams CPP Producer](#), im [GStreamer-Plugin](#) und in der [JNI-README-Datei](#) unter [GitHub](#).

Amazon S3-Objektpfad (Bild)

Der S3-Objektpfad beschreibt den Ort auf dem konfigurierten S3-Bucket, an den das generierte Bild geliefert wird. Es verwendet das folgende Format:

```
ImagePrefix_AccountID_StreamName_ImageTimecode_RandomID.file-extension
```

Die Objektpfadelemente sind wie folgt definiert:

- `ImagePrefix`- Wert von `AWS_KINESISVIDEO_IMAGE_PREFIX`, falls vorhanden.
- `AccountID`— Die AWS-Konto ID, unter der der Stream erstellt wird.
- `StreamName`— Name des Streams, aus dem das Bild generiert wird.
- `ImageTimecode`- Epochen-Timecode (in Millisekunden) in dem Fragment, in dem das Bild generiert wird.
- `RandomID`- Zufällige GUID.
- `file-extension`- JPG oder PNG, basierend auf dem angeforderten Bildformat.

In diesem Beispiel sieht der Objektpfad für die generierten Bilder wie folgt aus:

```
111122223333_demo-stream_16907729324_f20f9add-75e7-4399-a30f-fc7aefb1bab7.jpg
```

Bildmetadaten werden abgerufen

Sie können die S3-Konsole oder CLI verwenden, um die Metadaten für die generierten Bilder abzurufen.

Kinesis Video Streams legt die Fragmentnummer, den Produzenten- und Serverzeitstempel sowie die Inhaltstyp-Metadaten des generierten Bilds fest, die alle als Amazon S3-Objektmetadaten formatiert sind. Wenn zusätzliche MKV-Tags vorhanden sind, werden diese Tags ebenfalls als Amazon S3-Objektmetadaten hinzugefügt. Das folgende Beispiel zeigt, wie der Amazon S3-API-Befehl `head-object` verwendet wird, um die Objektmetadaten abzurufen. Die Antwort enthält die von Kinesis Video Streams erstellten Metadaten.

```
aws s3api head-object --bucket my-bucket-name --key 111122223333_demo-  
stream_1690707290324_f20f9add-7e57-4399-a30f-fc7aefb1bab7.jpg  
{  
  "AcceptRanges": "bytes",  
  "LastModified": "2023-07-30T08:54:51+00:00",  
  "ContentLength": 22693,  
  "ETag": "\"63e03cb6d57f77e2db984c1d344b1083\"",  
  "ContentType": "image/jpeg",  
  "ServerSideEncryption": "AES256",  
  "Metadata": {  
    "aws_kinesisvideo_producer_timestamp": "1690707290324",  
    "aws_kinesisvideo_server_timestamp": "1690707289209",  
    "aws_kinesisvideo_fragment_number":  
    "91343852333182036507421233921329142742245756394"  
  }  
}
```

Weitere Hinweise zu S3-Objektmetadaten finden Sie unter <https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMetadata.html>.

Amazon S3-URI-Empfehlungen zum Schutz vor Drosselung

Wenn Sie Tausende von Bildern in Amazon S3 schreiben, besteht das Risiko einer Drosselung. Weitere Informationen finden Sie unter Limits für [S3-Präfix-Put-Anfragen](#).

Ein Amazon 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-Präfixe. Zeitcodierte Daten wirken sich jeweils auf ein Präfix aus und ändern sich auch regelmäßig, sodass frühere Präfix-Skalierungen ungültig werden.

Um eine schnellere, konsistente Amazon S3-Skalierung zu ermöglichen, empfehlen wir, der Amazon S3-Ziel-URI ein zufälliges Präfix hinzuzufügen, z. B. einen Hex-Code oder eine UUID. Beispielsweise teilen Hex-Code-Präfixe Ihre Anfragen auf natürliche Weise nach dem Zufallsprinzip auf 16 verschiedene Präfixe auf (ein Präfix für jedes eindeutige Hex-Zeichen), sodass nach der automatischen Skalierung von Amazon S3 56.000 PUT-Anfragen pro Sekunde möglich sind.

Fehlerbehebung

Bilder wurden nicht an den Amazon S3-Bucket geliefert

Um dieses Problem zu beheben, müssen Sie auf einige Dinge achten:

1. Fehlende Berechtigungen
2. Die Konfiguration der Image-Generierung ist falsch
3. Das Tag wurde dem Fragment nicht hinzugefügt

Fehlende Berechtigungen

Wenn Sie einen vom Kunden verwalteten KMS-Schlüssel verwenden, stellen Sie wie folgt sicher, dass die Rolle, die die PutMedia Aufrufe ausführt (Uploader), über die entsprechenden Verschlüsselungs- und Entschlüsselungsberechtigungen verfügt und Zugriff auf den Amazon S3-Bucket hat:

- kms:Encrypt
- kms:GenerateDataKey
- kms:Decrypt
- s3:PutObject

Weitere Informationen finden Sie unter [the section called “Wie fange ich mit der serverseitigen Verschlüsselung an?”](#).

Überprüfen Sie das Ziel

Verwenden Sie die AWS CLI , um die DescribeImageGenerationConfiguration API für Ihren Stream aufzurufen.

```
aws kinesisisvideo describe-image-generation-configuration \
```

```
--stream-name "demo-stream"
```

Überprüfe das DestinationConfig in der Antwort und bestätige, dass es korrekt aussieht.

Stellen Sie sicher, dass das Tag zur Image-Generierung dem Fragment hinzugefügt wurde

1. Vergewissern Sie sich, dass der putKinesisVideoEventMetadata Aufruf erfolgreich war.

Die putKinesisVideoEventMetadata Methode gibt bei Erfolg den Statuscode 0 zurück.

Wir empfehlen, die Rückgabewerte der Funktionen auf 0 zu überprüfen. Wenn ein Statuscode ungleich Null zurückgegeben wird, konvertieren Sie ihn in Hex und überprüfen Sie den, um [the section called "Fehlercodereferenz"](#) weitere Informationen zu erhalten.

Vergewissern Sie sich, dass die Fehlerprotokolle aktiviert sind, und überprüfen Sie die Protokolle auf weitere Fehler in der Anwendung. Überprüfen und vergleichen Sie das Frame-Submission-Aufrufmuster Ihrer Anwendung mit der empfohlenen Implementierung: [Hinzufügen von Tags zur Bildgenerierung zu Fragmenten](#).

2. Überprüfen Sie die lokal generierte MKV-Datei

Vergewissern Sie sich, dass das Producer SDK oder die Beispielanwendung die Tags korrekt angehängt hat.

- a. Legen Sie die KVS_DEBUG_DUMP_DATA_FILE_DIR-Umgebungsvariable fest. Wenn dieser Wert festgelegt ist, schreibt das Producer SDK die Mediendateien, die es an Kinesis Video Streams gesendet hätte, an den angegebenen Speicherort.

```
export KVS_DEBUG_DUMP_DATA_FILE_DIR=/path/to/output/directory
```

 Note

Das SDK erstellt kein neues Verzeichnis, wenn der Pfad nicht existiert. Erstellen Sie den Ordner, falls erforderlich.

- b. Führen Sie die Anwendung erneut aus. Sie sollten sehen, dass .mkv Dateien in das angegebene Ausgabeverzeichnis geschrieben werden.
- c. Überprüfen Sie den Inhalt mithilfe oder einer anderen Software, um sicherzustellen MKVToolNix, dass das Tag vorhanden ist.
 - i. Installieren Sie MKVToolNix: `brew install mkvtoolnix`

- ii. Führen Sie es MKVToolNix mit einer der .mkv Dateien im Ausgabeverzeichnis aus.

```
mkvinfo -v ./path/to/video/file
```

- iii. Überprüfen Sie die MKVToolNix Ausgabe. Wenn die `KinesisVideoStream::PutFragmentMetadata` Producer-SDK-Methode korrekt aufgerufen wurde, sollte das folgende MKV-Tag angezeigt werden.

```
|+ Tags
| + Tag
|   + Simple
|     + Name: AWS_KINESISVIDEO_IMAGE_GENERATION
```

 Note

Die Tags gehören zum vorherigen Cluster.

- d. Wenn das MKV-Tag nicht vorhanden ist, stellen Sie sicher, dass die `KinesisVideoStream::PutEventMetadata` Producer-SDK-Methode mit dem `STREAM_EVENT_TYPE_IMAGE_GENERATION` Argument aufgerufen wurde und dass sie einen Erfolgscode (0) zurückgegeben 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-Videostreams](#)
- [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 müssen über Folgendes verfügen, 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 unter [the section called “Melden Sie sich an für 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 `StreamingMkvReader`-Objekt erstellen und verwenden können, um `MkvElement`-Objekte aus dem Eingabe-Stream `inputStream` zu extrahieren:

```
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new
InputStreamParserByteSource(inputStream));
    while (mkvStreamReader.mightHaveNext()) {
        Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
        if (mkvElement.isPresent()) {
            mkvElement.get().accept(fragmentVisitor);
            ...
        }
    }
}
```

FragmentMetadataVisitor

Diese Klasse ruft Metadaten für Fragmente (Medienelemente) ab und verfolgt einzelne Datenströme, die Medieninformationen 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, unter der des aktuellen Frames liegt.

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.
- Ruft [PutMedia](#) auf, um Videofragmente in den Kinesis-Videostream zu streamen.
- Ruft [GetMedia](#) auf, um Videofragmente aus dem Kinesis-Videostream zu streamen.
- Verwendet einen [StreamingMkvReader](#) zum Analysieren der zurückgegebenen Fragmente auf dem Stream und verwendet [FragmentMetadataVisitor](#) für die Protokollierung der Fragmente.

Löschen und Neuerstellen des Streams

Das folgende Codebeispiel (aus der `StreamOps.java` Datei) löscht einen bestimmten Kinesis-Videostream:

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

Das folgende Codebeispiel (aus der `StreamOps.java` Datei) erstellt einen Kinesis-Videostream mit dem angegebenen Namen:

```
amazonKinesisVideo.createStream(new CreateStreamRequest().withStreamName(streamName)
    .withDataRetentionInHours(DATA_RETENTION_IN_HOURS)
    .withMediaType("video/h264"));
```

Rufen Sie an PutMedia

Das folgende Codebeispiel (aus der `PutMediaWorker.java` Datei) ruft den Stream [PutMedia](#) auf:

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

Rufen Sie an GetMedia

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) ruft den Stream [GetMedia](#) auf:

```
GetMediaResult result = videoMedia.getMedia(new  
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

Analysieren Sie das Ergebnis `GetMedia`

Dieser Abschnitt beschreibt, wie [StreamingMkvReader](#), [FragmentMetadataVisitor](#) und `CompositeMkvElementVisitor` zur Analyse, zum Speichern in die Datei und zur Protokollierung der von `GetMedia` zurückgegebenen Daten verwendet wird.

Lesen Sie die Ausgabe von `with GetMedia StreamingMkvReader`

Das folgende Codebeispiel (aus der `GetMediaWorker.java` Datei) erstellt eine [StreamingMkvReader](#) und verwendet sie, um das Ergebnis der [GetMedia](#) Operation zu analysieren:

```
StreamingMkvReader mkvStreamReader = StreamingMkvReader.createDefault(new  
    InputStreamParserByteSource(result.getPayload()));  
log.info("StreamingMkvReader created for stream {} ", streamName);  
try {  
    mkvStreamReader.apply(this.elementVisitor);  
} catch (MkvElementVisitException e) {  
    log.error("Exception while accepting visitor {}", e);  
}
```

Im vorigen Codebeispiel ruft der [StreamingMkvReader](#) `MkvElement`-Objekte aus der Nutzlast des Ergebnisses von `GetMedia` ab. Im nächsten Abschnitt werden die Elemente einem [FragmentMetadataVisitor](#) übergeben.

Rufen Sie Fragmente ab mit `FragmentMetadataVisitor`

Die folgenden Codebeispiele (aus den Dateien `KinesisVideoExample.java` und `StreamingMkvReader.java`) erstellen einen [FragmentMetadataVisitor](#). Die `MkvElement`-Objekte, die von dem [StreamingMkvReader](#) durchlaufen werden, werden dann über die Methode `accept` dem Besucher übergeben.

von *KinesisVideoExample.java*:

```
FragmentMetadataVisitor fragmentMetadataVisitor = FragmentMetadataVisitor.create();
```

von *StreamingMkvReader.java*:

```
if (mkvElementOptional.isPresent()) {
    //Apply the MkvElement to the visitor
    mkvElementOptional.get().accept(elementVisitor);
}
```

Die Elemente schreiben und in eine Datei schreiben

Das folgende Code-Beispiel (aus der *KinesisVideoExample.java*-Datei) erstellt die folgenden Objekte und gibt sie als Teil des Rückgabewerts der *GetMediaProcessingArguments*-Funktion zurück:

- Einen *LogVisitor* (eine Erweiterung von *MkvElementVisitor*), die in das Systemprotokoll schreibt.
- Einen *OutputStream*, der die eingehenden Daten in eine MKV-Datei schreibt.
- Einen *BufferedOutputStream*, der die für *OutputStream* gebundenen Daten puffert.
- Einen [the section called “OutputSegmentMerger”](#), der aufeinanderfolgende Elemente in das *GetMedia*-Ergebnis einfügt, mit denselben Track- und EBML-Daten.
- A *CompositeMkvElementVisitor*, das das [FragmentMetadataVisitor](#), und zu einem *LogVisitor* einzigen Element [the section called “OutputSegmentMerger”](#), Besucher, zusammensetzt.

```
//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-Videostreams

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon Kinesis Video Streams und Ihren AWS Lösungen. Wir

empfehlen, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln, um Sie bei der Behebung eines Fehlers an mehreren Stellen zu unterstützen, falls einer auftritt. 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. Wenn Sie Amazon Kinesis Video Streams überwachen, speichern Sie 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 die Amazon Kinesis Video Streams-Metriken mit CloudWatch](#)
- [Überwachen Sie den Amazon Kinesis Video Streams Edge Agent mit CloudWatch](#)
- [Protokollieren Sie Amazon Kinesis Video Streams-API-Aufrufe mit AWS CloudTrail](#)

Überwachen Sie die Amazon Kinesis Video Streams-Metriken mit CloudWatch

Sie können einen Kinesis-Videostream mithilfe von Amazon überwachen. Amazon CloudWatch erfasst und verarbeitet Rohdaten aus Amazon Kinesis Video Streams in lesbare Messwerte, die nahezu in Echtzeit 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-Konsole](#) können Sie CloudWatch Kennzahlen für einen Amazon Kinesis-Videostream auf zwei Arten anzeigen:

- Wählen Sie auf der Dashboard-Seite im Abschnitt Account-level Metriken für die aktuelle Region den Tab Videostreams aus.
- Wählen Sie auf der Detailseite des Videostreams die Registerkarte Monitoring (Überwachung) aus.

Amazon Kinesis Video Streams bietet die folgenden Metriken:

Metrik	Description
ArchivedFragmentsConsumed.Media	Die Anzahl der Quotenpunkte für Fragment-Medien, die von allen APIs verbraucht wurden. Eine Erläuterung des Konzepts der Kontingentpunkte finden Sie unter the section called “Fragment-metadata und Kontingente für fragmentierte Medien”. Einheiten: Anzahl
ArchivedFragmentsConsumed.Metadata	Die Anzahl der Metadaten-Quotenpunkte für Fragmente, die von allen APIs verbraucht wurden. Eine Erläuterung des Konzepts der Kontingentpunkte finden Sie unter the section called “Fragment-metadata 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 des PutMedia Streams empfangen wurden. Einheiten: Byte
PutMedia.IncomingFragments	Die Anzahl der vollständigen Fragmente, die als Teil von PutMedia für den Stream empfangen wurden.

Metrik	Description
	Einheiten: Anzahl
<code>PutMedia.IncomingFrames</code>	Die Anzahl der vollständigen Frames, die als Teil von <code>PutMedia</code> für den Stream empfangen wurden. Einheiten: Anzahl
<code>PutMedia.ActiveConnections</code>	Die Gesamtzahl der Verbindungen zum Service-Host. Einheiten: Anzahl
<code>PutMedia.ConnectionErrors</code>	Die Fehler beim <code>PutMedia</code> Verbindungsaufbau für den Stream. Einheiten: Anzahl
<code>PutMedia.FragmentIngestionLatency</code>	Der Zeitunterschied zwischen dem Zeitpunkt, zu dem das erste und das letzte Byte eines Fragments von Amazon Kinesis Video Streams empfangen werden. Einheiten: Millisekunden
<code>PutMedia.FragmentPersistLatency</code>	Die Zeit ab dem Empfang und der Archivierung der vollständigen Fragmentdaten. Einheiten: Anzahl
<code>PutMedia.Latency</code>	Der Zeitunterschied zwischen der Anfrage und der InletService HTTP-Antwort vom Zeitpunkt des Verbindungsaufbaus. Einheiten: Anzahl

Metrik	Description
<code>PutMedia.BufferingAckLatency</code>	Der Zeitunterschied zwischen dem Zeitpunkt, zu dem das erste Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, zu dem das Buffering ACK für das Fragment gesendet wird. Einheiten: Millisekunden
<code>PutMedia.ReceivedAckLatency</code>	Der Zeitunterschied zwischen dem Zeitpunkt, zu dem das letzte Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, zu dem das empfangene ACK für das Fragment gesendet wird. Einheiten: Millisekunden
<code>PutMedia.PersistedAckLatency</code>	Der Zeitunterschied zwischen dem Zeitpunkt, zu dem das letzte Byte eines neuen Fragments von Amazon Kinesis Video Streams empfangen wird, und dem Zeitpunkt, zu dem das persistente ACK für das Fragment gesendet wird. Einheiten: Millisekunden
<code>PutMedia.ErrorAckCount</code>	Die Anzahl der Fehler-ACKS, die bei der Bearbeitung des Streams <code>PutMedia</code> gesendet wurden. Einheiten: Anzahl
<code>PutMedia.Success</code>	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

Metrik	Description
<code>GetMedia.Requests</code>	Die Anzahl der GetMedia API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl
<code>GetMedia.OutgoingBytes</code>	Die Gesamtzahl der Byte, die vom Dienst als Teil der GetMedia API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
<code>GetMedia.OutgoingFragments</code>	Die Anzahl der Fragmente, die bei der GetMedia Bearbeitung des Streams gesendet wurden. Einheiten: Anzahl
<code>GetMedia.OutgoingFrames</code>	Die Anzahl der Frames, die während des GetMedia angegebenen Streams gesendet wurden. Einheiten: Anzahl
<code>GetMedia.MillisBehindNow</code>	Der Zeitunterschied zwischen dem aktuellen Serverzeitstempel und dem Serverzeitstempel des zuletzt gesendeten Fragments. Einheiten: Millisekunden
<code>GetMedia.ConnectionErrors</code>	Die Anzahl der Verbindungen, die nicht erfolgreich aufgebaut wurden. Einheiten: Anzahl

Metrik	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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 Dienst als Teil der GetMediaForFragmentList API für einen bestimmten Stream gesendet wurden. Einheiten: Anzahl

Metrik	Description
<code>GetMediaForFragmentList.Requests</code>	Die Anzahl der <code>GetMediaForFragmentList</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl
<code>GetMediaForFragmentList.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>ListFragments.Latency</code>	Die Latenz der <code>ListFragments</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>ListFragments.Requests</code>	Die Anzahl der <code>ListFragments</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
<code>ListFragments.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>GetHLSStreamingSessionURL.Latency</code>	Die Latenz der <code>GetHLSStreamingSessionURL</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>GetHLSStreamingSessionURL.Requests</code>	Die Anzahl der <code>GetHLSStreamingSessionURL</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
<code>GetHLSStreamingSessionURL.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>GetHLSMasterPlaylist.Latency</code>	Die Latenz der <code>GetHLSMasterPlaylist</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>GetHLSMasterPlaylist.Requests</code>	Die Anzahl der <code>GetHLSMasterPlaylist</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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	Description
<code>GetHLSMediaPlaylist.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>GetMP4InitFragment.Latency</code>	Die Latenz der <code>GetMP4InitFragment</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>GetMP4InitFragment.Requests</code>	Die Anzahl der <code>GetMP4InitFragment</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
<code>GetMP4InitFragment.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>GetMP4MediaFragment.Latency</code>	Die Latenz der <code>GetMP4MediaFragment</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>GetMP4MediaFragment.Requests</code>	Die Anzahl der <code>GetMP4MediaFragment</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
<code>GetMP4MediaFragment.Success</code>	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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. Einheiten: Anzahl
<code>GetMP4MediaFragment.OutgoingBytes</code>	Die Gesamtzahl der Byte, die vom Dienst als Teil der <code>GetMP4MediaFragment</code> API für einen bestimmten Stream gesendet wurden. Einheiten: Byte
<code>GetTSFragment.Latency</code>	Die Latenz der <code>GetTSFragment</code> API-Aufrufe für den angegebenen Stream-Namen. Einheiten: Millisekunden
<code>GetTSFragment.Requests</code>	Die Anzahl der <code>GetTSFragment</code> API-Anfragen für einen bestimmten Stream. Einheiten: Anzahl

Metrik	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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	Description
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 AWS Anforderungs-IDs, finden Sie unter Request/Response Summary Logging. 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 helfen, Antworten auf die folgenden Fragen zu finden:

Themen

- [Erreichen Daten den Amazon Kinesis Video Streams-Dienst?](#)
- [Warum werden Daten nicht erfolgreich vom Amazon Kinesis Video Streams-Dienst aufgenommen?](#)
- [Warum können die Daten vom Amazon Kinesis Video Streams-Dienst 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-Dienst?

Relevante Metriken:

- `PutMedia.IncomingBytes`
- `PutMedia.IncomingFragments`
- `PutMedia.IncomingFrames`

Aktionselemente:

- Wenn diese Kennzahlen 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 nicht erfolgreich vom Amazon Kinesis Video Streams-Dienst aufgenommen?

Relevante Metriken:

- `PutMedia.Requests`
- `PutMedia.ConnectionErrors`
- `PutMedia.Success`
- `PutMedia.ErrorAckCount`

Aktionselemente:

- Wenn es zu einem Anstieg kommt `PutMedia.ConnectionErrors`, sehen Sie sich die HTTP-Antwort und die Fehlercodes an, die der Producer-Client erhalten hat, um zu sehen, welche Fehler beim Verbindungsaufbau auftreten.
- Wenn es zu einem Rückgang `PutMedia.Success` oder Anstieg kommt `PutMedia.ErrorAckCount`, sehen Sie sich den Ack-Fehlercode in den vom Service gesendeten Ack-Antworten an, um zu sehen, warum die Datenaufnahme fehlschlägt. Weitere Informationen finden Sie unter [AckErrorCode.Values](#).

Warum können die Daten vom Amazon Kinesis Video Streams-Dienst 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. Low-bandwidth Verbindungen könnten dazu führen, dass die Daten den Dienst mit einer niedrigeren Geschwindigkeit erreichen.

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 abnimmt `PutMedia.IncomingFragments`, überprüfen Sie die Netzwerkbandbreite und ob die Daten noch 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 `ListFragments.Latency` zunimmt, liegt höchstwahrscheinlich ein Serviceproblem vor. Wenn der Zustand über einen längeren Zeitraum anhält, erkundigen Sie sich bei Ihrem Kundendienstmitarbeiter, 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 es zu einem Anstieg kommt `GetMedia.ConnectionErrors`, kann es sein, dass der Verbraucher beim Lesen des Streams hinterherhinkt, da er häufig versucht, erneut eine Verbindung zum Stream herzustellen. Sehen Sie sich die `response/error` HTTP-Codes an, die für die `GetMedia` Anfrage zurückgegeben wurden.
- Wenn es zu einem Ausfall kommt `GetMedia.Success`, liegt das wahrscheinlich daran, dass der Dienst die Daten nicht an den Verbraucher senden kann, was dazu führen würde, dass die Verbindung unterbrochen wird und die Verbraucher erneut eine Verbindung herstellen, was dazu führen würde, dass der Verbraucher hinter dem Stream zurückbleibt.
- Wenn es zu einem Anstieg kommt `GetMedia.MillisBehindNow`, schauen Sie sich Ihre Bandbreitenlimits an, um festzustellen, ob Sie die Daten aufgrund der niedrigeren 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 Metriken 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 zu einem Anstieg kommt `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 eingehen, die der Dienst an die Verbraucher senden kann.
- Wenn es zu einem Rückgang kommt `GetMedia.Success` oder `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. Amazon CloudWatch sammelt und verarbeitet Rohdaten zu lesbaren Metriken, die nahezu in Echtzeit sind. Diese Statistiken werden für einen Zeitraum von 15 Monaten aufgezeichnet. Anhand dieser historischen Informationen können Sie sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder Ihres Amazon Kinesis Video Streams Edge Agent-Dienstes verschaffen.

Gehen Sie wie folgt vor, um die Metriken anzuzeigen:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im linken Navigationsbereich unter **Metriken** die Option **Alle Metriken** aus.
3. Wählen Sie den Tab „Durchsuchen“ und dann den **EdgeRuntimeAgent** benutzerdefinierten Namespace aus.

Amazon Kinesis Video Streams Edge Agent veröffentlicht die folgenden Metriken unter dem Namespace: **EdgeRuntimeAgent**

Dimensionen	Status	Description
Name des Streams, RecordJob	In Ausführung	Veröffentlicht kontinuierlich, wenn der RecordJob läuft. Einheiten: Keine. „1“ wird veröffentlicht, solange sie RecordJob sich in diesem Zustand befindet.
	FatalError	Wird 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.
Name des Streams, UploadJob	In Ausführung	Veröffentlicht kontinuierlich, wenn der UploadJob läuft. Einheiten: Keine. „1“ wird veröffentlicht, solange sie UploadJob sich in diesem Zustand befindet.

Dimensionen	Status	Description
	FatalError	Wird veröffentlicht, wenn der Fehler UploadJob schwerwiegend ist. Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.  Note Weitere Informationen finden Sie in den Protokollen.
	Completed	Veröffentlicht, wenn das abgeschlossen UploadJob ist. Einheiten: Keine. „1“ wird einmal veröffentlicht, wenn dieses Ereignis eintritt.
Name des Streams	PercentageSpaceUsed	Dies ist der Prozentsatz des gesamten Speicherplatzes, der in Amazon Kinesis Video Streams Edge Agent-Konfigurationen für Aufnahmemedien zugewiesen ist. Weitere Informationen finden Sie unter LocalSizeConfig. Einheiten: Prozentsatz (Skala 0—1).
Thing-Name	Lebendig	Veröffentlicht jede Minute vom Amazon Kinesis Video Streams Edge Agent aus, unabhängig von den darauf ausgeführten Konfigurationen. Dies kann verwendet werden, um zu ermitteln, ob der Amazon Kinesis Video Streams Edge Agent aktiv ist und Konfigurationen annehmen kann. Einheiten: Keine. „1“ wird jede Minute veröffentlicht.
	RecordJobs.HealthyJobCount	Gesamtzahl der laufenden und geplanten Aufzeichnungsaufträge auf Amazon Kinesis Video Streams Edge Agent. Einheiten: Anzahl.

Dimensionen	Status	Description
	UploadJobsHealthyJobCount	Gesamtzahl der laufenden und geplanten Upload-Jobs auf Amazon Kinesis Video Streams Edge Agent. Einheiten: Anzahl.
	RecordJobsUnhealthyJobCount	Gesamtzahl der aktuell fehlerhaften Rekordaufträge. Einheiten: Anzahl.
	UploadJobsUnhealthyJobCount	Gesamtzahl der aktuell fehlerhaften Upload-Jobs. Einheiten: Anzahl.
	RecordJobsRunningJobCount	Gesamtzahl der aktiv ausgeführten Rekordjobs. Einheiten: Anzahl.
	UploadJobsRunningJobCount	Gesamtzahl der aktiv ausgeführten Upload-Jobs. Einheiten: Anzahl.
	RecordJobsEdgeConfigCount	Gesamtzahl der Datensatzkonfigurationen, die auf Amazon Kinesis Video Streams Edge Agent bearbeitet werden. Einheiten: Anzahl.
	UploadJobsEdgeConfigCount	Gesamtzahl der Upload-Konfigurationen, die auf Amazon Kinesis Video Streams Edge Agent gerade bearbeitet werden. Einheiten: Anzahl.

CloudWatch Leitfaden zu Metriken für Amazon Kinesis Video Streams Edge Agent

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

Themen

- [Verfügt der Amazon Kinesis Video Streams Edge Agent über ausreichend Speicherplatz zum Aufzeichnen?](#)
- [Ist der Amazon Kinesis Video Streams Edge Agent aktiv?](#)
- [Gibt es ungesunde Jobs?](#)
- [Erfordert irgendein Job ein Eingreifen von außen?](#)

Verfügt der Amazon Kinesis Video Streams Edge Agent über ausreichend Speicherplatz zum Aufzeichnen?

Relevante Metriken: `PercentageSpaceUsed`

Aktion: Keine Aktion erforderlich.

Ist der Amazon Kinesis Video Streams Edge Agent aktiv?

Relevante Metriken: `Alive`

Handlung: 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 Situationen gestoßen ist:

- Ein Problem mit der Anwendungslaufzeit: Speicher- oder andere Ressourceneinschränkungen, Fehler usw
- Das AWS IoT Gerät, auf dem der Agent beim Herunterfahren, Abstürzen oder Beenden läuft
- Das AWS IoT Gerät hat keine Netzwerkkonnektivität

Gibt es ungesunde Jobs?

Relevante Metriken:

- `RecordJobs.UnhealthyJobCount`
- `UploadJobs.UnhealthyJobCount`

Handlung: Prüfen 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 `ProblemStartEdgeConfigurationUpdate`, bevor Sie den Job manuell neu starten.

- Wenn die `FatalError` Metrik nicht vorhanden ist, ist ein vorübergehender (nicht schwerwiegender) Fehler aufgetreten und Amazon Kinesis Video Streams Edge Agent versucht erneut, den Job auszuführen.

Note

Damit der Agent einen Auftrag, bei dem ein schwerwiegender Fehler aufgetreten ist, erneut versucht, verwenden Sie. [StartEdgeConfigurationUpdate](#)

Erfordert irgendein Job ein Eingreifen von außen?

Relevante Metriken:

- `PercentageSpaceUsed`— Wenn dieser Wert einen bestimmten Wert überschreitet, wird der Aufzeichnungsvorgang unterbrochen und erst wieder aufgenommen, wenn Speicherplatz verfügbar ist (wenn die Speicherdauer der Medien abgelaufen ist). Sie können eine aktualisierte Konfiguration mit einer höheren 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 fortgesetzt werden kann.

Handlung: Führen Sie einen API-Aufruf mit der Konfiguration durch, um Jobs neu zu starten, bei denen dieses Problem auftritt.

Protokollieren Sie Amazon Kinesis Video Streams-API-Aufrufe mit AWS CloudTrail

Amazon Kinesis Video Streams arbeitet mit einem Dienst zusammen AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon Kinesis Video Streams ausgeführten Aktionen bereitstellt. CloudTrail erfasst alle API-Aufrufe für Amazon Kinesis Video Streams als Ereignisse. Zu den aufgezeichneten Aufrufen gehören Aufrufe von der Amazon Kinesis Video Streams-Konsole und Codeaufrufe an die Amazon Kinesis Video Streams-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übertragung von CloudTrail Ereignissen an einen Amazon 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 Event-Verlauf einsehen. Anhand der gesammelten Informationen können Sie ermitteln CloudTrail, welche Anfrage 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.

Weitere Informationen CloudTrail, einschließlich der Konfiguration und Aktivierung, finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Amazon Kinesis-Videostreams und CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Amazon Kinesis Video Streams eine unterstützte Ereignisaktivität auftritt, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen im CloudTrail Eventverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Eventverlauf [anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für Amazon Kinesis Video Streams, erstellen Sie einen Trail. Ein Trail ermöglicht die CloudTrail Übermittlung von Protokolldateien an einen Amazon S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3-Bucket. Darüber hinaus können Sie andere so konfigurieren, AWS-Services dass die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter analysiert und entsprechend reagiert wird. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon 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: Amazon Kinesis Video Streams-Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon 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 [CreateStream](#) Aktion veranschaulicht.

```
{
  "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": "b7acfc0d-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 Videoaufzeichnung und -speicherung mit 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 aufzeichnen und speichern und Videos nach einem vom Kunden festgelegten Zeitplan zur langfristigen Speicherung, Wiedergabe und analytischen 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 auf Ihren lokalen Edge-Rechengeräten bereitstellen. Sie können sie auch problemlos in Docker-Containern bereitstellen, die auf Amazon EC2-Instances ausgeführt werden. Nach der Bereitstellung können Sie die Amazon Kinesis Video 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-Installation 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](#).
- Am 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-Entwicklerhandbuch](#).
- Bei einer nativen AWS IoT Bereitstellung: Sie können den Amazon Kinesis Video Streams Edge Agent nativ auf jeder Recheninstanz installieren. Das Edge SDK wird [AWS IoT Core](#) für die Verwaltung von Edge über die [Amazon Kinesis Video Streams API Operations verwendet](#).

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

Themen

- [API-Operationen für Amazon Kinesis Video Streams Edge Agent](#)
- [Überwachung des Amazon Kinesis Video Streams Edge Agent](#)
- [Stellen Sie es bereit, wenn nicht AWS IoT Greengrass mode](#)
- [Stellen Sie den Amazon Kinesis Video Streams Edge Agent bereit für AWS IoT Greengrass](#)
- [Häufig gestellte Fragen zum 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:

- [StartEdgeConfigurationUpdate](#)
- [DescribeEdgeConfiguration](#)
- [DeleteEdgeConfiguration](#)
- [ListEdgeAgentConfigurations](#)

Überwachung des Amazon Kinesis Video Streams Edge Agent

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

Stellen Sie es bereit, wenn nicht AWS IoT Greengrass mode

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
- Ausführen der Beispielanwendung

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 die RTSP-URLs Ihrer IP-Kamera](#)
- [Erstellen Sie eine IAM-Berechtigungsrichtlinie](#)
- [Erstellen einer IAM-Rolle](#)
- [Erstellen Sie den AWS IoT Rollen-Alias](#)
- [Erstellen Sie den AWS IoT policy](#)
- [Erstellen Sie eine AWS IoT Ding und hol AWS IoT Core Anmeldedaten](#)
- [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 systemeigenen Prozess aus](#)

Abhängigkeiten installieren

Bevor Sie mit der Verwendung des Amazon Kinesis Video Streams Producer SDK beginnen 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 unter [the 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 entsprechenden Bibliotheken auf Ihrem Gerät:

Ubuntu

Type:

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

Amazon Linux 2

Type:

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

Installieren Sie log4cplus-2.1.0 von der Quelle aus.

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/  
log4cplus-2.1.0.tar.gz  
tar -xzf 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://d1cdn.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 zum [Installieren oder Aktualisieren der neuesten Version der AWS CLI](#) Verfahren finden Sie im AWS Command Line Interface Benutzerhandbuch.

Erstellen Sie Ressourcen für die RTSP-URLs Ihrer IP-Kamera

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

Erstellen Sie Amazon Kinesis-Videostreams

Erstellen Sie Amazon Kinesis-Videostreams mithilfe der AWS-Managementkonsole AWS CLI, oder - API.

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

Weitere Informationen finden Sie unter [the section called "Erstellen Sie einen Amazon Kinesis-Videostream"](#).

Erstelle Geheimnisse in AWS Secrets Manager

Öffnen Sie in der AWS-Managementkonsole die [AWS Secrets Manager Konsole](#). Wählen Sie im linken Navigationsbereich 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 **Key/Value 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. Nenn es, wie du willst.
 - 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“.

Wähle den Namen deines Geheimnisses. Notieren Sie sich den Secret ARN.

3. Wiederhole diesen Vorgang für jeden MediaURI, von dem du streamen möchtest.

 Note

Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der Amazon EC2-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 Dauer des Fragments darf das unter angegebene Limit nicht überschreiten. [the section called “SDK-Kontingente für Produzenten”](#)

Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.

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

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

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 Vorgänge) für eine AWS Ressource. In diesem Fall sind die AWS Ressourcen 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-Managementkonsole 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 die ARNs für die Videostreams und `arn:aws:secretsmanager:*:*:secret:*` ersetzen Sie sie durch die ARNs, die die MediaURI-Geheimnisse enthalten, 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 [im](#) IAM-Benutzerhandbuch unter Neustrukturierung von Richtlinien.

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 von AWS IoT übernommen werden, um temporäre Anmeldeinformationen von AWS -Security-Token-Service (AWS STS) zu erhalten. Dies erfolgt, wenn Autorisierungsanfragen für Anmeldeinformationen vom Amazon Kinesis Video Streams Edge Agent ausgeführt werden.

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

1. Melden Sie sich bei der an AWS-Managementkonsole 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, in [the section called “Erstellen Sie eine IAM-Berechtigungsrichtlinie”](#) der Sie erstellt haben.
5. Wählen Sie Weiter aus.
6. Geben Sie einen Rollennamen oder ein Rollennamensuffix ein, damit Sie den Zweck dieser Rolle leichter identifizieren 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 key/value 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 Rolle erstellen.

Erstellen Sie den AWS IoT Rollen-Alias

Gehen Sie wie folgt vor, um einen AWS IoT Rollenalias für die IAM-Rolle zu erstellen, in der Sie erstellt haben. [the section called “Erstellen einer IAM-Rolle”](#) Ein Rollenalias ist ein alternatives Datenmodell, das auf die IAM-Rolle verweist. Eine Anforderung eines AWS IoT Anmeldeinformationsanbieters muss einen Rollenalias enthalten, der angibt, welche IAM-Rolle übernommen werden muss, 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](#) abzurufen.

Erstellen Sie den AWS IoT Rollen-Alias

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>.
2. Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.
3. Wählen Sie im linken Navigationsbereich Sicherheit und dann Rollenalias 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 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 ARN des Rollenalias. Das benötigen Sie für die AWS IoT Richtlinie im nächsten Schritt.

Erstellen Sie den AWS IoT policy

Gehen Sie wie folgt vor, um eine AWS IoT Richtlinie zu erstellen, die dem Gerätezertifikat beigelegt wird. Dadurch werden Berechtigungen für AWS IoT Funktionen erteilt und die Übernahme des Rollenalias mithilfe des Zertifikats ermöglicht.

Mithilfe von AWS IoT Core Richtlinien können Sie den Zugriff auf die AWS IoT Core Datenebene steuern. Die AWS IoT Core Datenebene besteht aus Vorgängen, mit denen Sie Folgendes ausführen können:

- Stellen Sie eine Verbindung 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 AWS IoT Richtlinien-Editor zum Erstellen eines AWS IoT policy

1. Melden Sie sich bei der an AWS-Managementkonsole 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": [
      "sts:AssumeRoleWithWebIdentity"
    ],
    "Resource": "arn:aws:iot:us-west-2:123456789012:rolealias/your-  
role-alias"
  }
]
}

```

 Note

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

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

Erstellen Sie eine AWS IoT Ding und hol AWS IoT Core Anmeldedaten

Zu diesem Zeitpunkt haben Sie Folgendes erstellt:

- Eine IAM-Berechtigungsrichtlinie. Siehe [the section called “Erstellen Sie eine IAM-Berechtigungsrichtlinie”](#).
- Eine IAM-Rolle, an die die Berechtigungsrichtlinie angehängt ist. 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 zu erstellen und zu registrieren AWS IoT Ding und hol AWS IoT Core Zugangsdaten

1. Registrieren Sie das Gerät als AWS IoT Ding und generieren Sie das X.509 Zertifikat für das Gerät.

- a. Melden Sie sich bei der an AWS-Managementkonsole 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 „Einzelne Sache erstellen“ und dann „Weiter“.

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

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

2. Schritt 2. Gerätezertifikat konfigurieren

Wählen Sie Auto-generate ein neues Zertifikat aus (empfohlen) und wählen Sie dann Weiter.

3. Schritt 3. Hängen Sie Richtlinien an das Zertifikat an

Suchen Sie nach der Berechtigungsrichtlinie, in der Sie erstellt haben [the section called „Erstellen Sie die AWS IoT Richtlinie“](#).

Markieren 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. Das ist das X.509 Zertifikat.
- Öffentliche Schlüsseldatei
- Private Schlüsseldatei
- 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 Fertig aus. Auf der nächsten Seite siehst du einen Hinweis, dass dein Ding erfolgreich erstellt wurde.

- h. Übertrage die oben heruntergeladenen Dateien auf dein AWS IoT Ding, falls es nicht schon da ist.
2. Besorgen Sie sich den Endpunkt des Anmeldeinformationsanbieters für Ihr AWS Konto.

AWS CLI

Führen Sie den folgenden Befehl aus:

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

AWS-Managementkonsole

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

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

Notieren Sie sich diese Informationen für einen späteren Schritt.

3. Rufen Sie den Gerätedaten-Endpunkt für Ihr AWS Konto ab.

AWS CLI

Führen Sie den folgenden Befehl aus:

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

AWS-Managementkonsole

Gehen Sie wie folgt vor:

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>.
2. Wählen Sie im linken Navigationsbereich Einstellungen aus.
3. Suchen Sie den Endpunkt Gerätedaten.

Notieren Sie sich diese Informationen für einen späteren Schritt.

4. (Optional) Stellen Sie sicher, dass Ihre Zertifikate korrekt generiert wurden.

Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Ihre Artikel 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 abzurufen](#).

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 das Jar in 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 Jar mit den Abhängigkeiten.

Note

Das im Lieferumfang 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 Folgendes enthält `pom.xml`.

Geben Sie `mvn clean package` ein.

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.

Geben Sie `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/`
`aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/` ein.

6. Legen Sie Umgebungsvariablen mithilfe der Werte 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 das Vertrauen in den Back-End-Dienst über TLS herzustellen. Beispiel: <i>/file/path/to/AmazonRootCA1.pem</i>
AWS_IOT_CORE_CERT	Ja	Dateipfad zum X.509 Zertifikat. Beispiel: <i>/file/path/to/certificateID-certificate.pem.crt</i>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	Ja	Der Endpunkt des Providers für AWS IoT Core Anmeldeinformationen für Ihr AWS Konto. Beispiel: <i>credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com</i>

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_IOT_CORE_DATA_ATS_ENDPOINT	Ja	Der Endpunkt auf AWS IoT Core Datenebene für Ihr AWS Konto. Beispiel: <i>data-accoun</i> <i>unt-specific-prefix</i> <i>.iot.aws-region</i> <i>.amazonaws.com</i>
AWS_IOT_CORE_PRIVATE_KEY	Ja	Dateipfad zum privaten Schlüssel, der im public/private Schlüsselpaar verwendet wird. Weitere Informationen finden Sie unter Schlüsselverwaltung in AWS IoT. Beispiel: <i>/file/path/to/certificateID-private</i> <i>.pem.key</i>
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: <i>kvs-edge-role-alias</i>

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 IngestorPipelineJNI plattformabhängigen Bibliotheken enthält. Lässt GStreamer diese Plugins laden. Weitere Informationen finden Sie unter the section called “Laden Sie das GStreamer-Element herunter, erstellen Sie es und konfigurieren Sie es”. Beispiel: /<i>download-location</i> /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgentVersion</i> /

Name der Umgebungsvariablen	Erforderlich	Beschreibung
LD_LIBRARY_PATH	Ja	Dateipfad, der auf das Verzeichnis verweist, das die <code>cproducer</code> und <code>KinesisVideoProducer</code> plattformabhängigen Bibliotheken enthält. Beispiel: <code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Nein	Legt fest, ob der Amazon Kinesis Video Streams Edge Agent Kennzahlen zum Jobstatus veröffentlicht. Amazon CloudWatch Zulässige Werte: TRUE/FALSE (Groß- und Kleinschreibung nicht beachten). Ist standardmäßig eingestellt, FALSE falls nicht angegeben. Beispiel: FALSE

Name der Umgebungsvariablen	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_LEVEL	Nein	Der Grad 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, findet ein Rollover statt. • 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. Die Standardeinstellung ist, <code>./log</code> falls nicht angegeben. Beispiel: <code>/file/path/</code>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Nein	Die Anzahl der Rollover-Protokolle, die vor dem Löschen aufbewahrt werden sollen. • Mindestbestellmenge: 1 • Maximal: 100 • Standard: 10, falls nicht angegeben Beispiel: 20
AWS_KVS_EDGE_RECORDING_DIRECTORY	Nein	Der Dateipfad, der auf das Verzeichnis verweist, in das die aufgezeichneten Medien geschrieben werden. Falls nicht angegeben, wird standardmäßig das aktuelle Verzeichnis verwendet. Beispiel: <code>/file/path/</code>

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

7. Löschen Sie den GStreamer-Cache. Type:

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

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

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

Name der Java-Eigenschaft	Erforderlich	Description
java.library.path	Nein	Dateipfad, der auf den Ordner verweist, der die gstkvssink und die IngestorPipelineJN

Name der Java-Eigenschaft	Erforderlich	Description
		I 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 *-Djava-property-name=value* sie dem Java-Befehl hinzu, mit dem das JAR ausgeführt wurde.

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 wie aus *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 (je nach Systemzeit auf Ihrem Gerät) aufzeichnet. AWS IoT Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 Uhr bis 21:59:59 Uhr hochgeladen.

Weitere Informationen finden Sie unter [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 Feld ein: AWS CLI

```
aws kinesisisvideo 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

Seien Sie sich der [CloudWatch Quoten](#) bewusst.

Folgen Sie diesen Verfahren, um den CloudWatch Agenten so zu installieren und zu konfigurieren, dass er die vom Amazon Kinesis Video Streams Edge Agent generierten Protokolle automatisch hochlädt. CloudWatch Dieser Schritt ist optional.

Anweisungen <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html> zur Installation des CloudWatch Agenten auf Ihrem Gerät finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Wenn Sie zur 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 Protokollierung verwendet wird.

Der verwendete Dateipfad setzt voraus, dass Sie den Amazon Kinesis Video Streams Edge Agent von dem 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 des Geräts veröffentlicht, 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 den RAM und die CPU 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 systemeigenen Prozess aus

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

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

Gehen Sie wie folgt vor:

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

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

Fügen Sie Folgendes ein:

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

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

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

Weitere Informationen zu den Parametern, die von der `systemd` Dienstkonfigurationsdatei akzeptiert werden, finden Sie in der [Dokumentation](#).

 Note

Fügen Sie die erforderlichen Umgebungsvariablen am . . . Speicherort hinzu, wie unter [the section called “Erstellen Sie den Edge-Agenten”](#).

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

Geben Sie `sudo systemctl daemon-reload` ein.

3. Starten Sie den Service.

Geben Sie `sudo systemctl start aws.kinesisvideo.edge-runtime-agent.service` ein.

4. Überprüfen Sie den Status des Amazon Kinesis Video Streams Edge Agent-Dienstes, um sicherzustellen, dass er ausgeführt wird.

Geben Sie `sudo systemctl status aws.kinesisvideo.edge-runtime-agent.service` ein.

Das Folgende ist 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.

Geben Sie `journalctl -e -u aws.kinesisvideo.edge-runtime-agent.service` ein.

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

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

- Um den Computer neu zu starten, geben `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service` Sie ein.
- Um den Vorgang zu beenden, geben Sie Folgendes ein `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Um bei jedem Geräteeustart automatisch zu starten, geben Sie Folgendes 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
- Erstellen und Verpacken einer Lambda-Funktion
- Konfiguration des Kinesis Video Streams-Kerngeräts
- Bereitstellung auf dem Kerngerät
- Verifizieren Sie Ihren Stream

Folgen Sie diesen Schritten, um den Amazon Kinesis Video Streams Edge Agent bereitzustellen, AWS IoT Greengrass um Medien von IP-Kameras aufzuzeichnen und hochzuladen.

Themen

- [Erstellen Sie eine Ubuntu Amazon EC2-Instance](#)
- [Richten Sie das ein AWS IoT Greengrass V2 Kerngerät auf dem Gerät](#)

- [Erstellen Sie die Amazon Kinesis Video Streams und AWS Secrets Manager Ressourcen für Ihre IP-Kamera-RTSP-URLs](#)
- [Fügen Sie der Rolle des Tokenaustauschdienstes \(TES\) Berechtigungen hinzu](#)
- [Installieren Sie das AWS IoT Greengrass Secret Manager-Komponente auf dem Gerät](#)
- [Stellen Sie den Amazon Kinesis Video Streams Edge Agent bereit AWS IoT Greengrass Komponente auf dem Gerät](#)
- [Installieren Sie den AWS IoT Greengrass Log-Manager-Komponente auf dem Gerät](#)

Erstellen Sie eine Ubuntu Amazon EC2-Instance

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

Erstellen Sie eine Ubuntu Amazon EC2-Instance

1. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die Amazon EC2-Konsole unter. <https://console.aws.amazon.com/ec2/>

Vergewissern Sie sich, 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-Anmeldung — Erstellen Sie Ihr eigenes Schlüsselpaar.
 - 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 Instanz und rufen Sie sie per SSH auf.

Gehen Sie wie folgt vor:

1. Wählen Sie im linken Navigationsbereich Instanzen und dann die Instanz-ID aus.
2. Wählen Sie oben rechts Verbinden aus.

3. Wählen Sie den 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 von Host (...) 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 ein AWS IoT Greengrass V2 Kerngerät auf dem Gerät

Folgen Sie diesen Verfahren, um die AWS IoT Greengrass Kernsoftware Nucleus auf der Amazon EC2-Instance zu installieren.

Richten Sie das ein AWS IoT Greengrass Kerngerät

1. Melden Sie sich bei AWS-Managementkonsole, an <https://console.aws.amazon.com/iot/>.

Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.

2. Wählen Sie im linken Navigationsbereich Greengrass-Geräte und Core-Geräte aus.

3. Wählen Sie Ein Kerngerät einrichten.

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 wird als Teil von installiert [the 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 Amazon EC2-Instance ein.

- Schritt 3.3: Führen Sie das Installationsprogramm aus
 1. Kopieren Sie die Befehle „Das Installationsprogramm herunterladen“ und „Das Installationsprogramm ausführen“ in der Ubuntu Amazon EC2-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 wird. Sie benötigen sie später.

 Note

Standardmäßig heißt die erstellte Rolle GreengrassV2TokenExchangeRole.

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 für benötigt werden AWS Secrets Manager. Führen Sie diesen Schritt zuerst aus, da Sie die ARNs der erstellten Ressourcen in den Richtlinien benötigen.

Erstellen Sie Amazon Kinesis-Videostreams

Erstellen Sie Amazon Kinesis-Videostreams mithilfe der AWS-Managementkonsole AWS CLI, oder - API.

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

Weitere Informationen finden Sie unter [the section called “Erstellen Sie einen Amazon Kinesis-Videostream”](#).

Erstelle Geheimnisse in AWS Secrets Manager

Öffnen Sie in der AWS-Managementkonsole die [AWS Secrets Manager Konsole](#). Wählen Sie im linken Navigationsbereich 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 Key/Value 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. Nenn es, wie du willst.
 - 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“.

Wähle den Namen deines Geheimnisses. Notieren Sie sich den Secret ARN.

3. Wiederhole diesen Vorgang für jeden MediaURI, von dem du streamen möchtest.

Note

Das AWS Netzwerk blockiert einige öffentliche RTSP-Quellen. Sie können nicht von der Amazon EC2-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 Dauer des Fragments darf das unter angegebene Limit nicht überschreiten. [the section called “SDK-Kontingente für Produzenten”](#)
Amazon Kinesis Video Streams Edge Agent unterstützt nur Video.

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

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

Fügen Sie der Rolle des Tokenaustauschdienstes (TES) Berechtigungen hinzu

Weisen Sie dem Gerät, das die Berechtigungen zum Einsehen der Geheimnisse annimmt, die Rolle des Tokenaustauschdienstes (TES) zu. 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-Managementkonsole und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie in der linken Navigationsleiste 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 die 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. Benennen Sie Ihre Richtlinie und wählen Sie dann Create policy 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 hinzugefügt haben.

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

Markieren Sie das Kontrollkästchen und wählen Sie Richtlinien anhängen aus.

Auf dem nächsten Bildschirm wird ein Hinweis angezeigt, dass die Richtlinie erfolgreich an die Rolle angehängt wurde.

10. Erstellen Sie eine weitere Richtlinie und hängen Sie sie an, diesmal für Ihre Geheimnisse.

Ersetzen Sie die Richtlinie durch die folgende:

 Note

arn:aws:secretsmanager:*:*:secret:* Ersetzen Sie sie durch die ARNs, die die MediaURI-Geheimnisse enthalten, in denen Sie 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, diesmal für Metriken, und hängen Sie sie an. Amazon CloudWatch Ersetzen Sie die Richtlinie durch die folgende:

JSON

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

```
}  
]  
}
```

Installieren Sie das 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-Managementkonsole und öffnen Sie die AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>. Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.
2. Wählen Sie in der linken Navigationsleiste Greengrass-Geräte und Deployments aus.

Wählen Sie das Deployment mit demselben Ziel wie das Ding, in dem wir es erstellt haben. [the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”](#)

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

Wählen Sie im 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 den Schalter Nur ausgewählte Komponenten anzeigen um und suchen Sie nach aws.greengrass.SecretManager.
 - Markieren Sie das Kästchen neben aws.greengrass.SecretManager und wählen Sie dann Weiter.
 - Schritt 3: Komponenten konfigurieren. Konfigurieren Sie die AWS IoT Greengrass Secret Manager-Komponente so, dass die Secrets aus der AWS IoT Greengrass Umgebung heruntergeladen werden.

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

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

 Note

`arn:aws:secretsmanager:*:*:secret:*` Ersetzen Sie die Secrets, die Sie in erstellt haben, durch die ARNs. [the 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 Schlüsselarn. Weitere Informationen finden Sie im [Abschnitt](#) Secret Manager-Konfiguration im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

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

- Schritt 4: Konfigurieren Sie die erweiterten 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 Ubuntu Amazon EC2-Instance Folgendes ein, `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` um zu überprüfen, ob die Komponente die aktualisierte Konfiguration erhalten hat.

6. Überprüfen Sie die AWS IoT Greengrass Kernprotokolle.

Geben Sie `sudo less /greengrass/v2/logs/greengrass.log` ein.

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 `sudo service greengrass restart`, um den AWS IoT Greengrass Kerndienst neu zu starten.

Wenn der Bereitstellungsfehler mit fehlenden Berechtigungen zusammenhängt, 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.

7. 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 das Deployment aktualisiert wird.

Um die Secrets in 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 die Secrets aus der AWS IoT Greengrass Umgebung heruntergeladen werden.

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

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

Weitere Informationen finden Sie unter Updates <https://docs.aws.amazon.com/greengrass/v2/developerguide/update-component-configurations.html#reset-configuration-update> zurücksetzen.

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

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 das Jar in 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 Jar mit den Abhängigkeiten.

Note

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

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

Geben Sie `mvn clean package` ein.

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.

Geben Sie `mv ./target/libs.jar ./KvsEdgeComponent/artifacts/`
`aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/` ein.

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

Name der Umgebungsvariable	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 zur Komponente Umgebungsvariablen im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

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

Name der Umgebungsvariable	Erforderlich	Beschreibung
LD_LIBRARY_PATH	Ja	Dateipfad, der auf das Verzeichnis verweist, das die <code>cproducer</code> und <code>KinesisVideoProducer</code> plattformabhängigen Bibliotheken enthält. Beispiel: <code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Nein	Legt fest, ob der Amazon Kinesis Video Streams Edge Agent Kennzahlen zum Jobstatus veröffentlicht. Amazon CloudWatch Zulässige Werte: TRUE/FALSE (Groß- und Kleinschreibung nicht beachten). Ist standardmäßig eingestellt, FALSE falls nicht angegeben. Beispiel: FALSE

Name der Umgebungsvariable	Erforderlich	Beschreibung
AWS_KVS_EDGE_LOG_LEVEL	Nein	Der Grad 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, findet ein Rollover statt. • Min.: 1 • Maximal: 100 • Standard: 20, falls nicht angegeben • Einheiten: Megabyte (MB) Beispiel: 5

Name der Umgebungsvariable	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. Die Standardeinstellung ist, <code>./log</code> falls nicht angegeben. Beispiel: <code>/file/path/</code>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Nein	Die Anzahl der Rollover-Protokolle, die vor dem Löschen aufbewahrt werden sollen. • Mindestbestellmenge: 1 • Maximal: 100 • Standard: 10, falls nicht angegeben Beispiel: 20
AWS_KVS_EDGE_RECORDING_DIRECTORY	Nein	Der Dateipfad, der auf das Verzeichnis verweist, in das die aufgezeichneten Medien geschrieben werden. Falls nicht angegeben, wird standardmäßig das aktuelle Verzeichnis verwendet. Beispiel: <code>/file/path/</code>

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

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

⚠ Important

Stellen Sie sicher, dass das geänderte Run-Skript 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 (je nach Systemzeit auf Ihrem Gerät) aufzeichnet. AWS IoT Außerdem werden die aufgezeichneten Medien täglich von 19:00:00 Uhr bis 21:59:59 Uhr hochgeladen.

Weitere Informationen finden Sie unter [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 ein, AWS CLI um die Datei an den Amazon Kinesis Video Streams Edge Agent zu senden:

```

aws kinesisisvideo 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 den AWS IoT Greengrass Log-Manager-Komponente auf dem Gerät

Note

Seien Sie sich der [CloudWatch Kontingente bewusst](#).

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 den 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-Managementkonsole und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
 - b. Klicken Sie in der linken Navigationsleiste auf Rollen.
 - c. Wählen Sie den Namen der TES-Rolle, die in erstellt wurde [the 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 überprüfen Sie, ob 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 wenn einige erforderliche Berechtigungen fehlen, erstellen Sie eine neue IAM-Richtlinie mit diesen Berechtigungen und hängen Sie sie an die TES-Rolle an, die in [the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”](#) erstellt wurde.
2. Melden Sie sich bei der an AWS-Managementkonsole und öffnen Sie die AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/>. Vergewissern Sie sich, dass die entsprechende Region ausgewählt ist.
3. Wählen Sie in der linken Navigationsleiste `Greengrass-Geräte` und `Deployments` aus.

Wählen Sie das Deployment mit demselben Ziel wie das Ding, in dem Sie es erstellt haben. [the section called “Richten Sie das AWS IoT Greengrass Kerngerät ein”](#)
4. Wählen Sie in der oberen rechten Ecke `Aktionen` und dann `Überarbeiten` aus.

Wählen Sie im 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: Wählen Sie Komponenten aus.
 - i. Stellen Sie sicher, dass die Komponente `aws.greengrass.cli` und `aws.greengrass.SecretManagerKomponenten` sind immer noch ausgewählt.

 Important

Deinstallieren Sie diese Komponenten nicht.

- ii. Schalten Sie den Schalter `Nur ausgewählte Komponenten anzeigen` um und suchen Sie nach `aws.greengrass.LogManager`.
- iii. Wählen Sie das Feld neben `aws.greengrass` aus. `LogManager` und 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 aus.

Fügen Sie auf dem daraufhin angezeigten Bildschirm die folgende Log-Manager-Konfiguration in das Feld Zusammensetzende 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

logFileDirectoryPathBei der vorherigen Konfiguration wird davon ausgegangen, dass der Standardspeicherort für die Protokollierung 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 die erweiterten Einstellungen. Klicken Sie auf **Weiter**.
- e. Schritt 5: Überprüfen. Wählen Sie **Bereitstellen** aus.
6. Vergewissern Sie sich, dass die AWS Log-Manager-Komponente und die Berechtigungen korrekt installiert wurden.
7. Geben Sie auf der Ubuntu Amazon EC2-Instance Folgendes 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.

Geben Sie `sudo less /greengrass/v2/logs/greengrass.log` ein.

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

Wenn der Bereitstellungsfehler mit fehlenden Berechtigungen zusammenhängt, überprüfen [the section called “Fügen Sie der TES-Rolle Berechtigungen hinzu”](#) Sie, ob die TES-Rolle über die richtigen Berechtigungen verfügt. Wiederholen Sie dann diesen Abschnitt.

Häufig gestellte Fragen zum 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

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

AL 2

- amzn2
 - AMD64 Amazon Linux: 2.0.20210219.0-amd64 (Schneeball)

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

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

Funktioniert der Amazon Kinesis Video Streams Edge Agent in AL2?

Ja.

Wie kann ich mehrere Streams innerhalb von ausführen AWS IoT Ding oder Gerät?

Senden Sie ein anderes [StartEdgeConfigurationUpdate](#) an dieselbenHubDeviceArn, aber anderen Amazon Kinesis Video Streams/ARNs AWS Secrets Manager .

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

Senden Sie eine aktualisierte [StartEdgeConfigurationUpdate](#) Version HubDeviceArn mit demselben Amazon Kinesis Video Streams-ARN. Wenn die Anwendung die Nachricht von Amazon Kinesis Video Streams empfängt, überschreibt sie die vorherige Konfiguration für diesen Stream. Änderungen werden dann vorgenommen.

Haben Sie Beispiele für Gemeinsamkeiten **ScheduleConfigs**?

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

Description	ScheduleExpression	DurationInSeconds
24/7 Aufnahme, stündliches Hochladen	(null ScheduleConfig)	
9:00:00 Uhr — 16:59:59 Uhr täglich	0 0 9-16 * * ? *	3599
09:00:00 Uhr — 16:59:59 Uhr wochentags	0 0 9-16 ? * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599
9:00:00 Uhr — 16:59:59 Uhr am Wochenende	0 0 9-16 ? * SAT,SUN *	3599

Description	ScheduleExpression	DurationInSeconds
22.00 Uhr — 23:59:59 Uhr wochentags	0 0 22,23 ? * MON-FRI *	3599
9:00:00 Uhr — 10:00:00 Uhr jeden Tag	0 0 9 * * ? *	3600
16:00:00 Uhr — 17:59:59 Uhr jeden Tag	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 [DeleteEdgeConfiguration](#) API, um Streams von einem Gerät zu löschen. Wenn Sie eine Konfiguration für denselben Stream mithilfe von aktualisieren, wird die Anzahl der Streams des Geräts [StartEdgeConfigurationUpdate](#) 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 zur Bestimmung, 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 FatalError Metrik unter.

Senden Sie den erneut [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”](#).

Streamen Sie Videos über eine VPC

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

Mit dem Amazon Kinesis Video Streams VPC-Endpunktservice 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 eine E-Mail mit](#) den folgenden Informationen:

- Konto-ID
- ARNs streamen
- VPC-ID

Note

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

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

- [AWS PrivateLink Hintergrund](#)
- [VPC-Handbuch für die ersten Schritte](#)

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 Quotenunterschiede sind:

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

Endpunkt herstellen

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

Erstellen Sie einen [Schnittstellen-VPC-Endpunkt](#) für Amazon Kinesis Video Streams, indem Sie entweder die Amazon VPC-Konsole oder das AWS Command Line Interface () verwenden. AWS CLI

Geben Sie in das Feld 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 weitergeleitet zu werden. 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 der AWS CLI Option finden Sie unter <https://docs.aws.amazon.com/cli/latest/reference/ec2/create-vpc-endpoint.html> create-vpc-endpoint.

Kontrollieren Sie den Zugriff auf Endpunkte

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 Prinzipal, 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 Prinzipalen 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 API arbeiten:

Beispiele: Senden von Daten an Kinesis Video Streams

- [Beispiel: Das SDK, das GStreamer-Plugin des Herstellers von Kinesis Video Streams — kvssink](#): Zeigt, wie das Kinesis Video Streams Producer SDK zur Verwendung als GStreamer-Ziel erstellt wird.
- [Führen Sie das GStreamer-Element in einem Docker-Container aus](#): Zeigt, wie ein vorgefertigtes Docker-Image zum Senden von RTSP-Videos (Real-Time Streaming Protocol) von einer IP-Kamera an Kinesis Video Streams verwendet wird.
- [Beispiel: Streaming von einer RTSP-Quelle](#): Zeigt, wie Sie Ihr eigenes Docker-Image erstellen und ein 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, die bereits in einem Containerformat (MKV) vorliegen. [PutMedia](#)

Beispiele: Abrufen von Daten aus Kinesis Video Streams

- [KinesisVideoExample](#): Zeigt, wie Videofragmente mithilfe der Kinesis Video Streams Parser Library analysiert und protokolliert werden.
- [Beispiel: Parsen und Rendern von Kinesis Video Streams-Fragmenten](#): Zeigt, wie Kinesis-Videostream-Fragmente mithilfe von JCodec und JFrame analysiert und gerendert werden. <http://jcodec.org/> <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>

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 ein, indem Sie ein Profil angeben, das Sie in Ihrer AWS Anmeldeinformationsprofildatei festgelegt haben, oder indem Sie Anmeldeinformationen in den Java-Systemeigenschaften Ihrer integrierten Entwicklungsumgebung (IDE) angeben. Falls Sie dies noch nicht getan haben, richten Sie zunächst Ihre Anmeldeinformationen ein. Weitere Informationen finden Sie unter AWS Anmeldedaten und Region für Entwicklung [einrichten](#).
- Wir empfehlen, eine der folgenden Java-IDEs zum Bearbeiten und Ausführen des Codes zu verwenden:
 - [Eclipse Java Neon](#)
 - [JetBrains IntelliJ IDEA](#)

Beispiel: Das SDK, das GStreamer-Plugin des Herstellers von Kinesis Video Streams — kvssink

In diesem Thema wird beschrieben, wie das Amazon Kinesis Video Streams Producer SDK zur Verwendung als GStreamer-Plugin erstellt wird.

Themen

- [Laden Sie das GStreamer-Element herunter, erstellen Sie es und konfigurieren Sie es](#)
- [Führen Sie das GStreamer-Element aus](#)
- [Beispiel für GStreamer-Startbefehle](#)
- [Führen Sie das GStreamer-Element in einem Docker-Container aus](#)
- [Referenz zu den GStreamer-Elementparametern](#)

[GStreamer](#) ist 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 Streams GStreamer-Plugin optimiert die Integration Ihrer vorhandenen GStreamer-Medienpipeline mit Kinesis Video Streams. Nach der Integration von GStreamer können Sie Videos von einer Webcam oder einer RTSP-Kamera (Real Time Streaming Protocol) auf Kinesis Video Streams streamen, um sie in Echtzeit oder später wiederzugeben, zu speichern und weiter zu analysieren.

Das GStreamer-Plug-In verwaltet automatisch die Übertragung Ihres Videostreams zu Kinesis Video Streams, indem es die vom Kinesis Video Streams Producer SDK bereitgestellten Funktionen in einem GStreamer-Sinkelement kapselt. `kvssink` Das GStreamer-Framework stellt eine verwaltete Standardumgebung für das Erstellen des Medienflusses von einem Gerät wie einer Kamera auf eine andere Videoquelle bereit, wo der Stream dann weiter verarbeitet, gerendert oder gespeichert werden kann.

Die GStreamer-Pipeline besteht in der Regel aus einem Link zwischen einer Quelle (Videokamera) und dem Sink-Element (entweder einem Player zum Rendern des Videos oder einem Speicher für den Offline-Abruf). In diesem Beispiel verwenden Sie das Producer SDK-Element als Sink bzw. Medienziel für die Videoquelle (Webcam oder IP-Kamera). Das Plugin-Element, das das SDK kapselt, sendet dann den Videostream an Kinesis Video Streams.

In diesem Thema wird beschrieben, wie eine GStreamer-Medienpipeline erstellt wird, die in der Lage ist, Videos von einer Videoquelle wie einer Webcam oder einem RTSP-Stream zu streamen, die in der Regel über Zwischenkodierungsstufen (mithilfe H.264 der Codierung) mit Kinesis Video Streams verbunden sind. 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 Beispiel für das GStreamer-Plug-In ist im Producer SDK von Kinesis Video Streams C++ enthalten. Weitere Informationen zu den Voraussetzungen und Downloadoptionen für das SDK finden Sie unter [Laden Sie den C++ Producer-Bibliothekscode herunter und konfigurieren Sie ihn](#).

Sie können das Producer-SDK GStreamer-Sink 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 Videos nur von einer Netzwerkkamera streamen, wenn Sie GStreamer in einem Docker-Container ausführen. Video-Streaming von einer USB-Kamera auf macOS in einem Docker-Container wird nicht unterstützt.

Führen Sie das GStreamer-Element aus

Verwenden Sie den Befehl, um GStreamer mit dem Kinesis Video Streams Producer SDK-Element als Senke auszuführen. `gst-launch-1.0` Verwenden Sie Upstream-Elemente, die für die Verwendung durch das GStreamer-Plugin geeignet sind. Zum Beispiel [v4l2src](#) für v4l2-Geräte auf Linux-Systemen oder [rtspsrc](#) für RTSP-Geräte. Geben Sie `kvssink` als Sink (Endziel der Pipeline) an, um Videos an das Producer SDK zu senden.

Zusätzlich zur [Bereitstellung von Anmeldeinformationen](#) und [einer Region](#) verfügt das `kvssink` Element über den folgenden erforderlichen Parameter:

- `stream-name`— Der Name der Kinesis Video Streams als Ziel.

Weitere Informationen zu optionalen `kvssink`-Parametern finden Sie unter [Referenz zu den GStreamer-Elementparametern](#).

Die neuesten Informationen zu GStreamer-Plugins und -Parametern finden Sie unter [GStreamer-Plugins](#). Sie können auch ein GStreamer-Element oder -Plugin `gst-inspect-1.0` gefolgt vom Namen eines GStreamer-Elements oder -Plug-ins verwenden, um dessen Informationen auszudrucken und zu überprüfen, ob sie auf Ihrem Gerät verfügbar sind:

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

Wenn das Erstellen `kvssink` fehlgeschlagen ist oder `GST_PLUGIN_PATH` nicht richtig eingestellt ist, sieht Ihre Ausgabe ungefähr so aus:

```
No such element or plugin 'kvssink'
```

Beispiel für GStreamer-Startbefehle

Die folgenden Beispiele zeigen, wie Sie das kvssink GStreamer-Plugin verwenden, um Videos von verschiedenen Gerätetypen zu streamen.

Beispiel 1: Streamen Sie Videos von einer RTSP-Kamera auf Ubuntu

Mit dem folgenden Befehl wird eine GStreamer-Pipeline auf Ubuntu erstellt, die von einer RTSP-Netzwerkamera mithilfe des [rtspsrc](#)-GStreamer-Plugins streamt:

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Beispiel 2: Video von einer USB-Kamera auf Ubuntu kodieren und streamen

Mit dem folgenden Befehl wird eine GStreamer-Pipeline auf Ubuntu erstellt, die den Stream von einer USB-Kamera im H.264 Format kodiert und an Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer-Plugin v4l2src [verwendet](#).

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-
raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc bframes=0 key-int-max=45
bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink
stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 3: Streamen Sie vorcodiertes Video von einer USB-Kamera auf Ubuntu

Mit dem folgenden Befehl wird eine GStreamer-Pipeline auf Ubuntu erstellt, die Videos, deren H.264 Format die Kamera bereits codiert hat, in Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer-Plugin v4l2src [verwendet](#).

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-
h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512
access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 4: Streamen Sie Videos von einer Netzwerkkamera auf macOS

Mit dem folgenden Befehl wird eine GStreamer-Pipeline unter macOS erstellt, die Videos von einer Netzwerkkamera an Kinesis Video Streams streamt. In diesem Beispiel wird das [rtspsrc](#)-GStreamer-Plugin verwendet.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! 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: Video von einer Netzwerkkamera unter Windows streamen

Mit dem folgenden Befehl wird unter Windows eine GStreamer-Pipeline erstellt, die Videos von einer Netzwerkkamera an Kinesis Video Streams streamt. In diesem Beispiel wird das [rtspsrc](#)-GStreamer-Plugin verwendet.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-
  name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
  key="YourSecretKey" aws-region="YourAWSRegion"
```

Beispiel 6: Video von einer Kamera auf dem Raspberry Pi streamen

Der folgende Befehl erstellt eine GStreamer-Pipeline auf dem Raspberry Pi, die Videos an Kinesis Video Streams streamt. In diesem Beispiel wird das GStreamer-Plugin <https://gstreamer.freedesktop.org/documentation/video4linux2/v4l2src.html?gi-language=c#v4l2src-page-v4l2src> verwendet.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert !
  video/x-raw,format=I420,width=640,height=480,framerate=30/1 !
  omxh264enc control-rate=1 target-bitrate=5120000 periodicity-
  idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-
  format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink
  stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"
  aws-region="YourAWSRegion"
```

Beispiel 7: Streamen Sie sowohl Audio als auch Video in Raspberry Pi und Ubuntu

Erfahren Sie, wie [Sie den Befehl gst-launch-1.0 ausführen, um das Streamen von Audio und Video in Ubuntu zu starten. Raspberry-Pi](#)

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-kodierten 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, wodurch die Erstellung und Verwendung der Anwendung optimiert wird.

So installieren und konfigurieren Sie Docker:

- [Anleitung zum Herunterladen von Docker](#)
- [Erste Schritte mit Docker](#)

Nach der Installation von Docker können Sie das Kinesis Video Streams C++ Producer SDK (und das GStreamer-Plugin) mithilfe eines der unten angegebenen Befehle aus der Amazon Elastic Container Registry herunterladen. `docker pull`

Gehen Sie wie folgt vor, um GStreamer mit dem Kinesis Video Streams Producer SDK-Element 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 für jede verwendete Registrierung Authentifizierungstoken erhalten. Die Token sind 12 Stunden lang 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 ein, wie in der Abbildung gezeigt.

```
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 die `gst-launch-1.0` zu `kvssink` verwenden, um eine für Ihr Gerät und Ihre Videoquelle geeignete Pipeline auszuführen. Beispiele für Pipelines finden Sie unter [Beispiel für GStreamer-Startbefehle](#).

Referenz zu den GStreamer-Elementparametern

Um Videos an das C++-SDK des Amazon Kinesis Video Streams-Herstellers zu senden, geben Sie `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-Plugin — kvssink”](#).

Topics

- [the section called “Geben Sie Anmeldeinformationen für kvssink ein ”](#)
- [the section called “Geben Sie eine Region für kvssink an”](#)
- [the section called “kvssink, optionale Parameter”](#)

Geben Sie Anmeldeinformationen für kvssink ein

Damit das `kvssink` GStreamer-Element Anfragen an folgende Adresse richten kann AWS, geben Sie AWS Anmeldeinformationen an, die es verwenden kann, wenn es den Amazon Kinesis Video Streams-Dienst aufruft. Die Kette der Anmeldeinformationsanbieter sucht in der folgenden Reihenfolge nach Anmeldeinformationen:

1. AWS IoT Anmeldedaten

Informationen zum Einrichten von AWS IoT Anmeldeinformationen finden Sie unter [the section called “Steuern Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT”](#).

Der `iot-credentials` Parameterwert muss mit einer kommasetrennten Liste der folgenden **key** = **value** -Paare beginnen `iot-certificate`, und darauf folgen.

Key (Schlüssel)	Erforderlich	Beschreibung
<code>ca-path</code>	Ja	Dateipfad zum CA-Zertifikat, das verwendet wird, um das Vertrauen in den Back-End-Dienst über TLS herzustellen. Example Beispiel: <code>/file/path/to/certificate.pem</code>

Key (Schlüssel)	Erforderlich	Beschreibung
cert-path	Ja	Dateipfad zum X.509 Zertifikat. Example Beispiel: <code>/file/path/to/certificateID-certificate.pem.crt</code>
endpoint	Ja	Der Endpunkt des Providers für AWS IoT Core Anmeldeinformationen für Ihr AWS Konto. Weitere Informationen finden Sie im AWS IoT Entwicklerhandbuch. Example Beispiel: <code>credential-account-specific-prefix.credentials.iot.region.amazonaws.com</code>
key-path	Ja	Dateipfad zum privaten Schlüssel, der im public/private Schlüsselpaar verwendet wird. Example Beispiel: <code>/file/path/to/certificateID-private.pem.key</code>

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: <i>KvsCameraIoRoleAlias</i>
iot-thing-name	Nein	Die iot-thing-name ist optional. Wenn nicht angegeben, iot-thing-name wird der stream-name Parameterwert verwendet. Example Beispiel: <i>kvs_example_camera</i>

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 Sitzungstoken-Wert an, wenn Sie temporäre Sicherheitssanmeldeinformationen direkt von AWS STS Operations 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 zukünftige Sitzungen persistent zu machen, legen Sie sie im Startskript Ihrer Shell fest.

3. **Zugriffsschlüssel**, Geheimschlüsselparameter

Um Anmeldeinformationen direkt als Parameter anzugeben, legen Sie die `kvssink` 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 Sitzungstoken-Wert an, wenn Sie temporäre Sicherheitsanmeldeinformationen direkt von AWS STS Operations verwenden.

Example

Verwenden statischer Anmeldeinformationen:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

Example

Verwendung temporärer Anmeldeinformationen:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

4. Datei mit Anmeldeinformationen

Important

Wenn Sie eine der vorherigen Methoden ausgewählt haben, können Sie den `credential-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 es in zu verwendenkvssink, 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 Zukunft mindestens $5 + 30 + 3 = 38$ Sekunden betragen. Die Kulanzzfrist ist als die `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` Variable in definiert [IotCredentialProvider.h](#). Wenn die Anmeldeinformationen zu Beginn zu kurz vor dem Ablaufdatum stehenkvssink, erhalten Sie den Fehlercode `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`.

⚠ Important

kvssink ändert die Datei mit den Anmeldeinformationen nicht. Wenn Sie temporäre Anmeldeinformationen verwenden, muss die Datei mit den Anmeldeinformationen vor Ablauf der Gültigkeitsdauer abzüglich der Kulanzfrist von einer externen Quelle aktualisiert werden.

Geben Sie eine Region für **kvssink** an

Im Folgenden finden Sie die Suchreihenfolge für Regionen:

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	Description	Einheit/Typ	Standard
stream-name	Der Name des Amazon Kinesis-Zielvideostreams.		
	⚠ Important Wenn kein Streamname angegeben ist, wird der Standard-Stream-Name		

Parameter	Description	Einheit/Typ	Standard
	verwendet: „DEFAULT_STREAM“. Wenn ein Stream mit diesem Standardnamen noch nicht existiert, wird er erstellt.		
absolute-fragment-times	Gibt an, ob absolute Fragmentzeiten verwendet werden.	Boolesch	true
access-key	Der AWS Zugriffsschlüssel, der für den Zugriff auf Kinesis Video Streams verwendet wird. Sie müssen entweder AWS Anmeldeinformationen festgelegt haben oder diesen Parameter angeben. Geben Sie Folgendes ein, um diese Informationen bereitzustellen: <pre>export AWS_ACCESS_KEY_ID=</pre>		

Parameter	Description	Einheit/Typ	Standard
avg-bandwidth-bps	Die erwartete durchschnittliche Bandbreite für den Stream.	Bits pro Sekunde	4194304

Parameter	Description	Einheit/Typ	Standard
aws-region	Die AWS-Region zu verwendenden.  Note Sie können die Region auch mit der <code>AWS_DEFAULT_REGION</code> Umgebungsvariablen versehen. Die Umgebungsvariablen haben Vorrang, wenn sowohl die Umgebungsvariable als auch die <code>kvssink-Parameter</code> gesetzt sind.  Important Die Region wird standardmäßig auf <code>us-west-2</code> gesetzt.	Zeichenfolge	"us-west-2"

Parameter	Description	Einheit/Typ	Standard
	sofern nicht anders angegeben.		
buffer-duration	Laufzeit des Streampuffers	Sekunden	120
codec-id	Die Codec-ID des Streams	Zeichenfolge	"V_MPEG4/ISO/AVC"
connection-staleness	Die Zeit, nach der der Stream Staleness Callback aufgerufen wird.	Sekunden	60
content-type	Der Inhaltstyp des Streams	Zeichenfolge	"video/h264"
fragment-acks	Gibt an, ob Fragment-ACKs verwendet werden.	Boolesch	true
fragment-duration	Die gewünschte Fragmentdauer	Millisekunden	2000
framerate	Die erwartete Bildrate.	Frames pro Sekunde	25
frame-timecodes	Verwendung von Frame-Timecodes oder Generieren von Zeitstempeln mit dem aktuellen Zeit-Callback:	Boolesch	true

Parameter	Description	Einheit/Typ	Standard
<code>key-frame-fragmentation</code>	Gibt an, ob Fragmente für einen Keyframe erstellt werden.	Boolesch	true
<code>log-config</code>	Der Protokollkonfigurationspfad	Zeichenfolge	"../kvs_log_configuration"
<code>max-latency</code>	Die maximale Latenz für den Stream	Sekunden	60
<code>recalculate-metrics</code>	Gibt an, ob die Metriken neu berechnet werden.	Boolesch	true
<code>replay-duration</code>	Die Laufzeit, um die der aktuelle Reader bei Auftreten eines Fehlers "zurückgespult" werden muss, um die Wiedergabe zu wiederholen	Sekunden	40
<code>restart-on-error</code>	Gibt an, ob beim Auftreten von Fehlern ein Neustart ausgeführt wird.	Boolesch	true
<code>retention-period</code>	Die Zeitdauer, für die der Stream aufbewahrt wird	Stunden	2

Parameter	Description	Einheit/Typ	Standard
rotation-period	Der Schlüsselrotationszeitraum. Weitere Informationen 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 Anmeldeinformationen festgelegt haben oder diesen Parameter angeben. <pre>export AWS_SECRET_ACCESS_KEY=</pre>		
session-token	Gibt den erforderlichen Sitzungstoken-Wert an, wenn Sie temporäre Sicherheitsanmeldeinformationen direkt aus AWS STS Vorgängen verwenden.		

Parameter	Description	Einheit/Typ	Standard
<code>storage-size</code>	Die Gerätespeichergröße in Mebibyte (MiB). Weitere Informationen zum Konfigurieren von Speicherg eräten finden Sie unter StorageInfo .	Mebibyte (MiB)	128
<code>streaming-type</code>	Der Streaming-Typ. Gültige Werte sind: • 0: Echtzeit • 1: nahezu in Echtzeit (derzeit nicht unterstützt) • 2: offline	Enum <code>GstKvsSinkStreamingType</code>	0: Echtzeit
<code>timecode-scale</code>	Die MKV-Timecode-Skala	Millisekunden	1
<code>track-name</code>	Der Name des MKV-Tracks.	Zeichenfolge	"kinesis_video"

Parameter	Description	Einheit/Typ	Standard
	<pre>h /to/certificate</pre> • <code>key-path=/localdirectorypath /to/private/key</code> • <code>ca-path=/localdirectorypath/to/ca-cert</code> • <code>role-aliases =role-aliases</code> • <code>iot-thing-name=YourIotThingName</code>		

Beispiel: Senden von Daten an Kinesis Video Streams mithilfe der API PutMedia

Dieses Beispiel zeigt, wie die [PutMedia](#) API 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 zu Einzelbildern zusammenfügen), finden Sie weitere Informationen unter [Zu Kinesis Video Streams hochladen](#).

Note

Die PutMedia Operation ist nur in den C++- und Java-SDKs verfügbar. Dies ist auf die Vollduplexverwaltung 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 Sie den Code und untersuchen Sie ihn](#)
- [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 Speicherorte der Bibliotheken zu konfigurieren und den Code so zu konfigurieren, dass er Ihre AWS Anmeldeinformationen verwendet.

1. Erstellen Sie ein Verzeichnis und klonen Sie den Beispiel Quellcode aus dem GitHub Repository. Das PutMedia-Beispiel ist Bestandteil der [Java](#).

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

2. Öffnen Sie die Java-IDE, die Sie verwenden (z. B. [Eclipse](#) oder [IntelliJ IDEA](#)), und importieren Sie das Apache Maven-Projekt, das Sie heruntergeladen haben:
 - In Eclipse: Klicken Sie auf File (Datei), Import (Importieren), Maven, Existing Maven Projects (Vorhandene Maven-Projekte) und navigieren Sie zum Stammverzeichnis des heruntergeladenen Pakets. Wählen Sie die pom.xml-Datei aus.
 - In IntelliJ Idea: Klicken Sie auf Import. Navigieren Sie zur Datei pom.xml im Stammverzeichnis des heruntergeladenen Pakets.

Weitere Informationen finden Sie in der Dokumentation zur IDE.

3. Aktualisieren Sie das Projekt, damit die IDE die importierten Bibliotheken findet.
 - Gehen Sie bei IntelliJ IDEA wie folgt vor:
 - a. Klicken Sie mit der rechten Maustaste auf das Verzeichnis lib des Projekts und wählen Sie Add as library.
 - b. Wählen Sie „Datei“ und dann „Projektstruktur“.
 - 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 Sie den Code und untersuchen Sie ihn

Das PutMedia-API-Beispiel () weist folgendes Programmiermuster auf:PutMediaDemo

Themen

- [Erstellen Sie den PutMediaClient](#)
- [Streamen von Medien und Anhalten des Threads](#)

Die Codebeispiele in diesem Abschnitt stammen aus der Klasse PutMediaDemo.

Erstellen Sie den PutMediaClient

Für das Erstellen 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 [Verwenden Sie die Java-Producer-Bibliothek](#), die Kinesis Video 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-Konsole oder mithilfe der AWS CLI.
2. Ändern Sie Ihr Arbeitsverzeichnis auf das Java Producer SDK-Verzeichnis:

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Kompilieren Sie das Java SDK und die Demoanwendung:

```
mvn package
```

4. Erstellen Sie einen temporären Dateinamen im Verzeichnis `/tmp`:

```
jar_files=$(mktemp)
```

- Erstellen Sie eine classpath-Zeichenfolge mit den Abhängigkeiten zum lokalen Repository in einer Datei:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

- Legen Sie den Wert der Umgebungsvariablen LD_LIBRARY_PATH wie folgt fest:

```
export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-  
video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:  
$LD_LIBRARY_PATH  
$ classpath_values=$(cat $jar_files)
```

- Führen Sie die Demo wie folgt über die Befehlszeile 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
```

- Öffnen Sie die [Kinesis Video Streams-Konsole](#) und wählen Sie auf der Seite „Streams verwalten“ Ihren Stream aus. Das Video wird im Bereich Video Preview wiedergegeben.

Beispiel: Streaming von einer RTSP-Quelle

Der [C++](#) enthält eine Definition für einen <https://www.docker.com/> 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 vereinfacht 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 Amazon Kinesis Video Streams zu senden. Dies ist eine umfassende 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 Kinesis Video Streams RTSP-Beispielanwendung 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-Netzwerkcameraquelle: Weitere Informationen zu empfohlenen Kameras finden Sie unter [Systemanforderungen](#).

Erstellen Sie das Docker-Image

Erstellen Sie zunächst das Docker-Image, in dem die Demoanwendung ausgeführt werden soll.

1. Klonen Sie das Amazon Kinesis Video Streams-Demo-Repository.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Wechseln Sie in das Verzeichnis, das das Dockerfile enthält. In diesem Fall ist es das <https://github.com/aws-samples/amazon-kinesis-video-streams-demos/blob/master/producer-cpp/docker-rtsp/> Docker-RTSP-Verzeichnis.

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. Verwenden Sie den folgenden Befehl, um das Docker-Image zu erstellen. Dieser Befehl erstellt das Image und kennzeichnet es als rtspdockertest.

```
docker build -t rtspdockertest .
```

4. Führen Sie aus `docker images` und suchen Sie nach der Bild-ID, die mit `rtspdockertest` markiert ist.

In der Beispielausgabe unten `IMAGE ID` ist das 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-Videostreams und die URL der RTSP-Netzwerkamera 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-Konsole](#) an AWS-Managementkonsole und öffnen Sie sie.

Sehen Sie sich den Stream an.

4. Um den Docker-Container zu verlassen, schließen Sie das Terminalfenster oder geben Sie `exit`.

Außerhalb des Docker-Containers

Verwenden Sie von 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: Parsen und Rendern von Kinesis Video Streams-Fragmenten

Der [Streamen Sie mithilfe der Parser-Bibliothek](#) enthält eine Demo-Anwendung mit dem Namen `KinesisVideoRendererExample`, die das Analysieren und Rendern von Amazon Kinesis-Videostream-Fragmenten demonstriert. Das Beispiel verwendet [JCodec](#), um die H.264 kodierten [Frames](#) zu dekodieren, die mithilfe der Anwendung aufgenommen werden. [Beispiel: Das SDK, das](#)

[GStreamer-Plugin des Herstellers von Kinesis Video Streams — kvssink](#) Nachdem der Frame mit JCodec decodiert wurde, wird das sichtbare Bild mit [JFrame](#) gerendert.

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.
- Sehen Sie sich den Videoinhalt von Streams in einer benutzerdefinierten Anwendung an, anstatt die Kinesis Video Streams-Konsole zu verwenden.

Sie können die Klassen in diesem Beispiel auch verwenden, um Kinesis-Videostream-Inhalte anzusehen, die nicht codiert sind H.264, 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. Falls 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](#).

Ausführen des Renderer-Beispiels

1. Erstellen Sie ein Verzeichnis und klonen Sie dann den Beispiel Quellcode 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 Neuaufbau 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

Das Beispiel sendet MKV-Beispieldaten aus der Datei

`rendering_example_video.mkv` und sendet mithilfe von Videodaten aus einem Stream mit dem Namen `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 an GetMedia](#) in der Dokumentation zu [Streamen Sie mithilfe der Parser-Bibliothek](#).

Den Frame dekodieren und anzeigen

Der Frame wird dann im Beispiel mittels [JFrame](#) decodiert und angezeigt.

Das folgende Code-Beispiel stammt aus der Klasse KinesisVideoFrameViewer, die JFrame erweitert:

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

Das Bild wird als Instanz von [java.awt.image angezeigt. BufferedImage](#). Beispiele, die zeigen, wie man damit arbeitet BufferedImage, finden Sie unter [Reading/Loading Bild](#).

Sicherheit in Amazon Kinesis-Videostreams

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitsrelevantesten Organisationen 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 für den Schutz der Infrastruktur verantwortlich, auf der AWS Dienste in der AWS Cloud ausgeführt werden. 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 in Scope by Compliance Program](#).
- **Sicherheit in der Cloud** — Ihre Verantwortung hängt von dem AWS Dienst ab, 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. Die folgenden Themen zeigen Ihnen, wie Sie Kinesis Video Streams so konfigurieren, dass Ihre Sicherheits- und Compliance-Ziele erreicht werden. Außerdem erfahren Sie, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Kinesis Video Streams-Ressourcen überwachen und sichern können.

Topics

- [Datenschutz in Kinesis Video Streams](#)
- [Steuerung des Zugriffs auf Kinesis Video Streams-Ressourcen mithilfe von IAM](#)
- [Steuern Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT](#)
- [Überprüfung der Konformität für Amazon Kinesis Video Streams](#)
- [Widerstandsfähigkeit bei Amazon Kinesis-Videostreams](#)
- [Sicherheit der Infrastruktur in Kinesis Video Streams](#)
- [Bewährte Sicherheitsmethoden für Kinesis Video Streams](#)

Datenschutz in Kinesis Video Streams

Sie können serverseitige Verschlüsselung (SSE) mithilfe von AWS Key Management Service (AWS KMS) -Schlüsseln verwenden, um strenge Datenverwaltungsanforderungen zu erfüllen, indem Sie Ihre Daten im Ruhezustand 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?](#)
- [Erstellen und Verwenden eines vom Kunden verwalteten Schlüssels](#)
- [Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels](#)

Was ist serverseitige Verschlüsselung für Kinesis Video Streams?

Server-side Verschlüsselung ist eine Funktion in Kinesis Video Streams, die Daten mithilfe eines von Ihnen angegebenen AWS KMS Schlüssels automatisch verschlüsselt, bevor sie im Ruhezustand gespeichert werden. Die Daten werden verschlüsselt, bevor sie in die Stream-Speicherschicht von Kinesis Video Streams geschrieben werden. Nach Abruf aus dem Speicher werden sie entschlüsselt. Dadurch werden Ihre Daten stets im Ruhezustand im Kinesis-Video-Streams-Service verschlüsselt.

Mit der serverseitigen Verschlüsselung müssen Ihre Kinesis-Videostream-Produzenten und -Verbraucher keine KMS-Schlüssel oder kryptografischen Operationen verwalten. Wenn die Datenspeicherung aktiviert ist, werden Ihre Daten automatisch verschlüsselt, wenn sie Kinesis Video Streams betreten und verlassen, sodass Ihre gespeicherten Daten verschlüsselt werden. 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 Dienst importiert wird. AWS KMS

Überlegungen zu Kosten, Regionen und Leistung

Wenn Sie serverseitige Verschlüsselung anwenden, fallen für Sie die AWS KMS API-Nutzung und die Schlüsselkosten an. Im Gegensatz zu benutzerdefinierten AWS KMS Schlüsseln wird der `aws/kinesisvideo` Standard-KMS-Schlüssel kostenlos angeboten. Sie müssen jedoch weiterhin für die API-Nutzungskosten aufkommen, die Kinesis Video Streams in Ihrem Namen entstehen.

Die API-Nutzungskosten fallen für jeden KMS-Schlüssel an, auch für benutzerdefinierte. Die AWS KMS Kosten skalieren mit der Anzahl der Benutzeranmeldeinformationen, die Sie für Ihre Datenproduzenten und -verbraucher verwenden, da für jede Benutzeranmeldeinformationen ein eindeutiger API-Aufruf von 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.
- User-generated KMS-Schlüssel sind AWS KMS key kostenpflichtig. Weitere Informationen finden Sie unter [AWS Key Management Service – Preise](#).

AWS KMS Nutzung der API

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

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

Verfügbarkeit der serverseitigen Verschlüsselung nach Region

Server-side Die Verschlüsselung von Kinesis-Videostreams ist überall dort verfügbar, AWS-Regionen wo Kinesis Video Streams verfügbar ist.

Wie fange ich mit der serverseitigen Verschlüsselung an?

Server-side Die Verschlüsselung ist in 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 bei der Erstellung zugewiesen werden. Sie können einem Stream später mithilfe der [UpdateStream](#) API keinen anderen Schlüssel zuweisen.

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

- Wenn Sie einen Kinesis-Videostream in der erstellen AWS-Managementkonsole, geben Sie den KMS-Schlüssel auf der Seite **Neuen Videostream erstellen** auf der Registerkarte **Verschlüsselung** an.
- Wenn Sie einen Kinesis-Videostream mithilfe der [CreateStream](#) API erstellen, geben Sie die Schlüssel-ID im `KmsKeyId` Parameter an.

Erstellen und Verwenden eines vom Kunden verwalteten Schlüssels

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

Einen vom Kunden verwalteten Schlüssel erstellen

Informationen zum Erstellen eigener Schlüssel finden Sie [im AWS Key Management Service Entwicklerhandbuch unter](#) **Erstellen von Schlüsseln**. Nachdem Sie Schlüssel für Ihr Konto erstellt haben, gibt der Kinesis Video Streams-Dienst diese Schlüssel in der **Liste der vom Kunden verwalteten Schlüssel** zurück.

Verwenden Sie einen vom Kunden verwalteten Schlüssel

Nachdem Ihren Verbrauchern, Produzenten und Administratoren die richtigen Berechtigungen erteilt wurden, können Sie benutzerdefinierte KMS-Schlüssel in Ihrem eigenen AWS-Konto oder einem anderen 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 zu verwenden, 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 wird. Dies führt zu folgendem Ergebnis:

- Wenn der von Ihnen angegebene Schlüssel nicht existiert, ist der `CreateStream` Vorgang erfolgreich, aber `PutMedia` alle `GetMedia` Operationen im Stream schlagen fehl.
- Wenn Sie den bereitgestellten Schlüssel (`aws/kinesisvideo`) verwenden, ist der Schlüssel erst in Ihrem Konto vorhanden, wenn die erste `PutMedia` `GetMedia` Operation ausgeführt wurde.

Berechtigungen zur Verwendung eines vom Kunden verwalteten Schlüssels

Bevor Sie die serverseitige Verschlüsselung mit einem vom Kunden verwalteten Schlüssel verwenden können, müssen Sie die 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 Standarddienstschlüssels 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-Prinzipale) 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 Herstellerberechtigungen

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

JSON

```
{
```

```

"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": [
      "kinesisvideo:PutMedia"
    ],
    "Resource": "arn:aws:kinesisvideo:*:123456789012:stream/MyStream/*"
  }
]
}

```

Beispiel für Verbraucherberechtigungen

Ihre Kinesis-Videostream-Nutzer müssen über die folgende `kms:Decrypt` Erlaubnis verfügen:

JSON

```

{
  "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": [
        "kinesisvideo:GetMedia"
      ],

```

```
    "Resource": "arn:aws:kinesisvideo:*:123456789012:stream/MyStream/*"
  }
]
}
```

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

Sie können AWS Identity and Access Management (IAM) mit Amazon Kinesis Video Streams verwenden, um zu steuern, 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 mit IAM](#)
- [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 Genehmigung 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, beim Erstellen und Verwalten von IAM-Richtlinien den [IAM Policy Generator](#) und den [IAM Policy Simulator 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-Operationen anzugeben, verwenden Sie das Sternchen (*) als Platzhalter wie folgt:

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

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

Amazon-Ressourcennamen (ARNs) für Kinesis Video Streams

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

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

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

Beispiel:

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

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

Anderen IAM-Konten Zugriff auf einen Kinesis-Videostream gewähren

Möglicherweise müssen Sie anderen IAM-Konten die Erlaubnis erteilen, Operationen an Streams in Kinesis Video Streams auszuführen. 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 9999999999 als ID für Ihre Kinesis Video Streams

2. Erstellen Sie eine IAM-verwaltete Richtlinie in dem Konto, dem der Stream gehört (999999999999), die die Zugriffsebene zulässt, die Sie gewähren möchten.

Beispiel für eine Richtlinie:

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-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, das über die Berechtigung für Operationen wie `DescribeStreamGetDataEndpoint`, 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 ARN der Rolle. Sie benötigen ihn im nächsten Schritt.

- Erstellen Sie im anderen Konto, 111111111111, eine verwaltete Richtlinie, die die `AssumeRole` Aktion für die Rolle zulässt, die Sie im vorherigen Schritt im Konto 999999999999 erstellt haben. Sie müssen den Rollen-ARN aus dem vorherigen Schritt angeben.

Beispiel für eine Richtlinie:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
    }
  ]
}

```

- Hängen Sie die im vorherigen Schritt erstellte Richtlinie an eine IAM-Entität an, z. B. eine Rolle oder einen Benutzer im Konto 111111111111. Dieser Benutzer hat jetzt die Berechtigung, die Rolle im Konto 999999999999 zu übernehmen. `CustomRoleName`

Die Anmeldeinformationen dieses Benutzers rufen die AWS STS AssumeRole API auf, um die Anmeldeinformationen für die Sitzung abzurufen. Diese werden anschließend zum Aufrufen der Kinesis Video Streams-APIs für den Stream verwendet, der im Konto 9999999999 erstellt wurde.

```
aws sts assume-role --role-arn "arn:aws:iam::9999999999: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 zuvor in der Umgebung festgelegten Daten fest.

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=
```

7. Führen Sie die Kinesis Video Streams-APIs aus, um den Datenendpunkt für den Stream im Konto 999999999999 zu beschreiben und abzurufen.

```
aws kinesisisvideo 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 schrittweise Anweisungen zum Gewähren von kontoübergreifendem Zugriff finden Sie [unter Delegieren von Zugriffsrechten mithilfe von IAM-Rollen. AWS-Konten](#)

Beispielrichtlinien für Kinesis Video Streams

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

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

Diese Richtlinie ermöglicht es einem Benutzer oder einer Gruppe `DescribeStream`, die `ListTagsForStream` Operationen, `GetDataEndpoint` `GetMediaListStreams`, und an jedem 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-Ressourcen

Diese Richtlinie ermöglicht es einem Benutzer oder einer Gruppe, jeden Kinesis Video Streams-Vorgang auf einer beliebigen Ressource auszuführen. 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 Sie den Zugriff auf Kinesis Video Streams-Ressourcen mit AWS IoT

In diesem Abschnitt wird beschrieben, wie Sie es 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 AWS IoT Anmeldeinformationsanbieter und eine AWS Identity and Access Management (IAM-) Rolle verwenden.

Geräte können X.509 Zertifikate verwenden, um AWS IoT mithilfe der gegenseitigen TLS-Authentifizierungsprotokolle eine Verbindung herzustellen. Andere AWS-Services (z. B. Kinesis Video Streams) unterstützen keine zertifikatsbasierte Authentifizierung, können aber mithilfe von AWS Anmeldeinformationen im Format AWS Signature Version 4 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 hat einen Anmeldeinformationsanbieter, der es Ihnen ermöglicht, das integrierte X.509 Zertifikat als eindeutige Geräteidentität zur Authentifizierung von AWS Anfragen (z. B. Anfragen an Kinesis Video Streams) zu verwenden. Dadurch entfällt die Notwendigkeit, eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel auf Ihrem Gerät zu speichern.

Der Anmeldeinformationsanbieter authentifiziert einen Client (in diesem Fall ein Kinesis Video Streams-SDK, das auf der Kamera ausgeführt wird, von der 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 einen Anruf an die 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, sodass der AWS IoT Anmeldeinformationsanbieter 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 Stream-Name](#)
- [AWS IoT Certificateld als Stream-Name](#)
- [Verwenden Sie AWS IoT Anmeldeinformationen zum Streamen an einen fest codierten Stream-Namen](#)

AWS IoT ThingName als Stream-Name

Themen

- [Schritt 1: Erstellen Sie eine AWS IoT Dingstyp und ein AWS IoT Objekt](#)
- [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 das AWS IoT Anmeldeinformationen mit Ihrem Kinesis-Videostream](#)
- [Schritt 5: Bereitstellen AWS IoT Zertifikate und Anmeldeinformationen im Dateisystem Ihrer Kamera und Streamen von Daten in Ihren Videostream](#)

Schritt 1: Erstellen Sie eine AWS IoT Dingstyp und ein AWS IoT Objekt

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 zuerst einen Dingtyp erstellen. AWS IoT 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 des Typs `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 sind Benutzern insofern ähnlich, als eine Rolle eine AWS Identität mit Berechtigungsrichtlinien ist, die festlegen, was die Identität tun kann 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 übernommen werden, AWS IoT um temporäre Anmeldeinformationen vom Security Token Service (STS) abzurufen, wenn Autorisierungsanfragen für Anmeldeinformationen vom Client ausgeführt werden. In diesem Fall ist der Client das Kinesis Video Streams SDK, das auf Ihrer Kamera ausgeführt wird.

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. Hängen Sie als Nächstes eine Berechtigungsrichtlinie an die IAM-Rolle an, die Sie zuvor erstellt haben. Diese Berechtigungsrichtlinie ermöglicht die selektive Zugriffskontrolle (eine Teilmenge der unterstützten Vorgänge) 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 das folgende 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 wird. (`${credentials-iot:ThingName}`) Dieser Platzhalter nimmt den Wert des Ding-Attributs an AWS IoT, ThingName wenn der Anbieter der AWS IoT Anmeldeinformationen den Namen des Videostreams in der Anfrage sendet.

3. Erstellen Sie als Nächstes einen Rollenalias für Ihre IAM-Rolle. Rollenalias ist ein alternatives Datenmodell, das auf die IAM-Rolle verweist. Eine Anforderung eines AWS IoT Anmeldeinformationsanbieters muss einen Rollenalias enthalten, der angibt, welche IAM-Rolle übernommen werden muss, 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 called `erstelltKvsCameraIoTPolicy`.

```
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` JSON-Dokument 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 ist 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 AWS IoT Anmeldeinformationsanbieter zu autorisieren, benötigst du den AWS IoT Anmeldeinformationen-Endpunkt, der für deine AWS-Konto ID eindeutig ist. Sie können den folgenden Befehl verwenden, um den Endpunkt für 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 zum Back-End-Dienst herzustellen. 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 das AWS IoT Anmeldeinformationen mit Ihrem Kinesis-Videostream

Jetzt können Sie die AWS IoT Anmeldeinformationen 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 für AWS IoT Anmeldeinformationen auf, 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 JSON-Ausgabe enthält die `accessKey` Zeichen `secretKey`, und die `sessionToken`, mit denen Sie auf die Kinesis-Videostreams zugreifen können.

3. Für Ihren Test können Sie diese Anmeldeinformationen verwenden, um die Kinesis Video `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 AWS IoT Zertifikate 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 in den vorherigen Schritten generierte CA-Zertifikat in das Dateisystem Ihrer Kamera. Geben Sie die Pfade für den Speicherort dieser Dateien, den Rollenaliasnamen und den Endpunkt der AWS IoT Anmeldeinformationen für die Ausführung des `gst-launch-1.0` Befehls oder Ihrer Beispielanwendung an.
2. Der folgende Beispielbefehl verwendet die AWS IoT Zertifikatsautorisierung, um Videos 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 Certificateld als Stream-Name

Um Ihr Gerät (z. B. Ihre Kamera) durch AWS IoT etwas darzustellen, aber einen anderen Streamnamen zu autorisieren, können Sie das AWS IoT `certificateId` Attribut als Ihren Streamnamen verwenden und Kinesis Video Streams-Berechtigungen für den Stream gewähren. AWS IoT Die Schritte, um dies zu erreichen, ähneln den zuvor beschriebenen, mit ein paar Änderungen.

- Ändern Sie die Berechtigungsrichtlinie für Ihre IAM-Rolle (`iam-permission-document.json`) wie folgt:

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 Stream-Namen verwenden. Rufen Sie die Zertifikat-ID aus dem Zertifikat ab, sodass Sie sie als Stream-Namen im folgenden Describe-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 AWS IoT Anmeldeinformationsanbieter 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 die an den `thingname` Anmeldeinformationsanbieter 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 zum Streamen an einen fest codierten Stream-Namen

Um Ihr Gerät (z. B. Ihre Kamera) durch AWS IoT etwas darzustellen, aber das Streaming zu einem bestimmten Amazon Kinesis-Videostream zu autorisieren, müssen Sie Amazon Kinesis Video Streams-Berechtigungen für den verwendeten Stream gewähren. AWS IoT Das Verfahren ist ähnlich wie in den vorherigen Abschnitten, mit einigen Änderungen.

Ändern Sie die Berechtigungsrichtlinie für Ihre IAM-Rolle (`iam-permission-document.json`) wie folgt:

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 in den vorherigen Schritten generierte Zertifikat, den privaten Schlüssel und das CA-Zertifikat in das Dateisystem Ihrer Kamera.

Geben Sie die Pfade für den Speicherort dieser Dateien, den Rollenaliasnamen, den AWS IoT Dingnamen und den Endpunkt der AWS IoT Anmeldeinformationen für die Ausführung des `gst-launch-1.0` Befehls oder Ihrer Beispielanwendung an.

Der folgende Beispielbefehl verwendet eine 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"
```

Überprüfung der Konformität für Amazon Kinesis Video Streams

Unser neues AWS Anmeldeerlebnis ist nicht für regulierte Workloads konzipiert. Wenn Sie unser neues AWS Anmeldeerlebnis nutzen, es aber AWS für regulierte Workloads nutzen möchten, können Sie sich für erweiterte Funktionen [für AWS Ihre Umgebung anmelden oder](#) diese aktivieren. AWS

Um zu erfahren, ob AWS-Service ein Programm [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm](#) und wählen Sie das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme](#).

Sie können Auditberichte von Drittanbietern mit heruntergeladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter Berichte AWS Artifact](#).

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens sowie den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Compliance-Verantwortung bei der Verwendung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Widerstandsfähigkeit bei Amazon Kinesis-Videostreams

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 hochredundanten Netzwerken 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](#).

Sicherheit der Infrastruktur 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 sind](#).

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 Verschlüsselungssammlungen mit Perfect Forward Secrecy (PFS) wie Ephemeral (DHE) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Diffie-Hellman Die meisten modernen Systeme, z. B. Java 7 und höher, unterstützen diese Modi.

Darüber hinaus müssen Anfragen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert werden, der einem IAM-Principal zugeordnet ist. 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-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.

Beispielsweise benötigt ein Producer, der Daten an Kinesis Video Streams sendet `PutMedia`, nur `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

Producer- 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 Anmeldeinformationen, die nicht automatisch rotiert werden und erhebliche Auswirkungen auf das Geschäft haben könnten, wenn sie missbraucht werden.

Stattdessen sollten Sie eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Ihre Producer- 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 Zugangsschlüssel) 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 arbeitet mit AWS CloudTrail, einem Dienst, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service in Kinesis Video Streams ausgeführten Aktionen bereitstellt.

Anhand der gesammelten Informationen können Sie ermitteln, welche Anfrage CloudTrail an 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.

Weitere Informationen finden Sie unter [the section called “API-Aufrufe protokollieren mit CloudTrail”](#).

Problembehandlung bei Kinesis-Videostreams

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme mit Amazon Kinesis Video Streams zu beheben.

Themen

- [Allgemeine Probleme](#)
- [API-Probleme](#)
- [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-Dienst gesendet werden. Eine Möglichkeit, die Latenzzeit zwischen Producer und Service zu reduzieren, besteht darin, die Medien-Pipeline so zu konfigurieren, dass kürzere Fragmentlaufzeiten entstehen.

Um die Anzahl der in jedem Fragment gesendeten Frames zu reduzieren, reduzieren Sie den folgenden Wert in `inkinesis_video_gstreamer_sample_app.cpp`:

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512,
NULL);
```

Note

Die Latenzen sind im Mozilla Firefox-Browser aufgrund der internen Implementierung des Video-Renderings höher.

API-Probleme

In diesem Abschnitt werden API-Probleme beschrieben, die bei der Arbeit mit Kinesis Video Streams auftreten können.

Problembereiche

- [Fehler: „Unbekannte Optionen“](#)
- [Fehler: „Der service/operation Name, der autorisiert werden soll, konnte nicht bestimmt 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 Standardausgabety (none) neu. Informationen zur Konfiguration von finden [Sie AWS CLI unter configure](#) in der AWS CLI Befehlsreferenz.

Fehler: „Der service/operation Name, der autorisiert werden soll, konnte nicht bestimmt 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 abrufen, achten Sie darauf, den folgenden Parameter in den GetDataEndpoint Aufruf aufzunehmen, 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 vergewissern Sie sich, dass die Stream-Informationen abgerufen werden können:

```
aws kinesisisvideo describe-stream --stream-name StreamName --endpoint https://  
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Wenn der Anruf fehlschlägt, finden Sie AWS CLI [weitere Informationen unter](#) Problembehandlung.

Fehler: „Der Dienst hat die Verbindung geschlossen, bevor die endgültige AckEvent Version empfangen wurde“

PutMedia kann mit dem folgenden Fehler fehlschlagen:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was  
received
```

Dieser Fehler kann auftreten, wenn PushbackInputStream nicht korrekt implementiert ist. Stellen Sie sicher, dass die unread() Methoden korrekt implementiert sind.

Fehler: "STATUS_STORE_OUT_OF_MEMORY"

PutMedia kann mit dem folgenden Fehler fehlschlagen:

The content store is out of memory.

Dieser Fehler tritt auf, wenn der Inhaltsspeicher nicht ausreichend groß dimensioniert wurde. Um den Inhaltsspeicher zu vergrößern, erhöhen Sie den Wert von `StorageInfo.storageSize`. Weitere Informationen finden Sie unter [StorageInfo](#).

Fehler: „Die Anmeldeinformationen sollten auf eine gültige Region beschränkt sein.“

Dieser Fehler tritt auf, wenn die Signaturregion nicht mit der Endpunktregion übereinstimmt.

Wenn Sie beispielsweise `us-west-2` als Ihre Signaturregion angeben, aber versuchen, eine Verbindung zum `kinesisvideo.us-east-1.amazonaws.com` (`us-east-1`)-Endpunkt herzustellen, erhalten Sie diesen Fehler.

In einigen Anwendungen, wie [kvssink](#), ist die Region-Fallback-Kette standardmäßig auf eingestellt. `us-west-2` Vergewissern Sie sich, dass Sie Ihre Region entsprechend der von Ihnen verwendeten Anwendung richtig eingestellt haben.

HLS-Probleme

Wenn Ihr Videostream nicht korrekt wiedergegeben wird, finden Sie weitere Informationen unter [the section called “Behebung von HLS-Problemen”](#).

Probleme mit Java

In diesem Abschnitt wird beschrieben, wie häufig auftretende Java-Probleme bei der Arbeit mit Kinesis Video Streams behoben werden.

Problembereiche

- [Aktivieren von Java-Protokollen](#)

Aktivieren von Java-Protokollen

Zur Behebung von Problemen mit Java-Beispielen und -Bibliotheken ist es hilfreich, die Debug-Protokolle zu aktivieren und zu überprüfen. 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: „Security token included in the request is invalid“ \(Das Sicherheits-Token der Anfrage ist ungültig\) beim Streamen von Daten mit der GStreamer-Demo-Anwendung](#)
- [Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“](#)
- [GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt“](#)
- [Fehler: „Failed to allocate heap“ \(Heap konnte nicht zugeordnet werden\) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi](#)
- [Fehler: „Illegal Instruction“ \(Ungültige Anweisung\) beim Ausführen der GStreamer-Demo auf Raspberry Pi](#)
- [Die Kamera kann nicht auf Raspberry Pi geladen werden.](#)
- [Kamera kann auf macOS High Sierra nicht gefunden werden.](#)
- [Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden](#)
- [Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung](#)
- [Timestamp/range Behauptung zur 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

Um Ihren Videostream in der Konsole anzuzeigen, muss er H.264 im AVCC-Format codiert sein. Prüfen Sie Folgendes, wenn der Stream nicht angezeigt wird:

- Sie [NAL-Anpassungsflags](#) sind auf eingestellt, NAL_ADAPTATION_ANNEXB_NALS | NAL_ADAPTATION_ANNEXB_CPD_NALS wenn der ursprüngliche Stream im Annex-B Format ist. Dies ist der Standardwert im `StreamDefinition`-Konstruktor.
- Sie stellen die Code Private Data ordnungsgemäß bereit. Denn H.264 das ist 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 elementare Streams haben das folgende Format, wobei Ab sich der Annex-B Startcode (001 oder 0001) befindet:

```
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), wenn er als SPS und PPS im Stream H.264 ist, kann an das aVCC-Format angepasst werden. Sofern die Medienpipeline die CPD nicht separat bereitstellt, kann die Anwendung die CPD aus dem Frame extrahieren, indem sie nach dem ersten Idr-Frame (der SPS und PPS enthalten sollte) sucht, die beiden NALUs extrahiert (die Ab (Sps) Ab (Pps) sein werden) und sie in die CPD einfügt `StreamDefinition`.

Fehler: „Security token included in the request is invalid“ (Das Sicherheits-Token der Anfrage ist ungültig) beim Streamen von Daten mit der GStreamer-Demo-Anwendung

Wenn dieser Fehler auftritt, liegt ein Problem mit Ihren Anmeldeinformationen vor. Überprüfen Sie Folgendes:

- Wenn Sie temporäre Anmeldeinformationen verwenden, müssen Sie den Sitzungs-Token angeben.
- Überprüfen Sie, ob Ihre temporären Anmeldeinformationen abgelaufen sind.
- Vergewissern Sie sich, dass Sie über die richtigen Rechte verfügen.
- Vergewissern Sie sich unter MacOS, dass Sie keine Anmeldeinformationen im Schlüsselbund zwischengespeichert haben.

Fehler: „Fehler beim Senden des Frames an den Kinesis-Video-Client“

Wenn dieser Fehler auftritt, sind die Zeitstempel im Quell-Stream nicht richtig gesetzt. Gehen Sie wie folgt vor:

- Verwenden Sie die neueste SDK-Version, in der es möglicherweise schon ein Update gibt, das Ihr Problem behebt.
- Stellen Sie für den Stream mit hoher Qualität eine höhere Bitrate ein und korrigieren Sie alle Jitter im Quellstream, sofern die Kamera dies unterstützt.

GStreamer-Anwendung stoppt auf OS X mit der Meldung „Streamen gestoppt, Grund nicht ausgehandelt“

Das Streamen kann unter OS X mit der folgenden Meldung abgebrochen werden:

```
Debugging information: gstbasesrc.c(2939): void gst_base_src_loop(GstPad *) (): /
GstPipeline:test-pipeline/GstAutoVideoSrc:source/GstAVFVideoSrc:source-actual-src-
avfvide:
streaming stopped, reason not-negotiated (-4)
```

Eine mögliche Problemumgehung besteht darin, die Framerate-Parameter aus dem `gst_caps_new_simple` Aufruf zu entfernen in: `kinesis_video_gstreamer_sample_app.cpp`

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

Fehler: „Failed to allocate heap“ (Heap konnte nicht zugeordnet werden) beim Erstellen des Kinesis-Video-Clients in der GStreamer-Demo auf Raspberry Pi

Die GStreamer-Beispielanwendung versucht, 512 MB RAM zuzuweisen, die möglicherweise nicht auf Ihrem System verfügbar sind. Sie können die Größe des zugewiesenen Speicherplatzes verringern, indem Sie den folgenden Wert in `KinesisVideoProducer.cpp` verkleinern:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Fehler: „Illegal Instruction“ (Ungültige Anweisung) beim Ausführen der GStreamer-Demo auf Raspberry Pi

Wenn beim Ausführen der GStreamer-Demo der folgende Fehler auftritt, überprüfen Sie, ob Sie die Anwendung für die richtige Version Ihres Geräts kompiliert haben. (Stellen Sie beispielsweise sicher, dass Sie nicht für Raspberry Pi 3 kompilieren, wenn Sie auf Raspberry Pi 2 arbeiten.)

```
INFO - Initializing curl.  
Illegal instruction
```

Die Kamera kann nicht auf Raspberry Pi geladen werden.

Um zu überprüfen, ob die Kamera geladen wurde, führen Sie die folgenden Schritte aus:

```
ls /dev/video*
```

Ist die Suche ergebnislos, führen Sie die folgenden Schritte aus:

```
vcgencmd get_camera
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
supported=1 detected=1
```

Wenn der Treiber die Kamera nicht erkennt, gehen Sie wie folgt vor:

1. Überprüfen Sie die technische Einrichtung der Kamera und prüfen Sie, ob sie richtig angeschlossen ist.

2. Führen Sie die folgenden Schritte aus, um die Firmware zu aktualisieren:

```
sudo rpi-update
```

3. Starten Sie das Gerät neu.
4. Führen Sie die folgenden Schritte aus, um den Treiber zu laden:

```
sudo modprobe bcm2835-v4l2
```

5. Überprüfen Sie, ob die Kamera erkannt wurde:

```
ls /dev/video*
```

Kamera kann auf macOS High Sierra nicht gefunden werden.

Unter macOS High Sierra kann die Demo-Anwendung die Kamera nicht finden, wenn mehr als eine Kamera verfügbar ist.

Datei jni.h beim Kompilieren auf macOS High Sierra nicht gefunden

Um diesen Fehler zu beheben, aktualisieren Sie Ihre Installation von Xcode auf die neueste Version.

Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung

Um Curl-Fehler beim Ausführen der GStreamer-Demo-Anwendung zu beheben, kopieren Sie [diese Zertifikatsdatei](#) zu `/etc/ssl/cert.pem`.

Timestamp/range Behauptung zur 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/aws-labs/aws-sdk-android-samples/blob/master/
AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/
StreamConfigurationFragment.java#L169](https://github.com/aws-labs/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. Das Limit für die maximale Fragmentdauer [the section called “API-Dienstkontingente für Medien und archivierte Medien”](#) finden Sie im 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 auf Schlüsselbildern basierende Fragmentierung verwenden, stellen Sie den Encoder so ein, dass Schlüsselbilder innerhalb von 10 Sekunden bereitgestellt werden.
 - Wenn Sie keine auf Schlüsselframes basierende Fragmentierung verwenden, legen Sie bei der Definition des Streams die maximale Fragmentdauer auf einen Wert fest [Schreiben und untersuchen Sie den Code](#), der unter 10 Sekunden liegt.
- 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. Setzen Sie zum Beispiel `key-int-max` auf 60 und `fps` auf 30, 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, legen Sie die GOP-Größe auf 60 fest.

"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` identisch ist. Weitere Informationen finden Sie unter [Referenz zu den GStreamer-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

Stellen Sie sicher, dass Ihr Stream die richtigen privaten Codec-Daten enthält, um in Ihrer Consumer-Anwendung auf einen einzelnen Frame von einer Streaming-Quelle zuzugreifen. Informationen über das Format der Daten in einem Stream finden Sie unter [Datenmodell](#).

In der folgenden Testdatei auf der Website erfahren Sie, wie Sie private Codec-Daten für den Zugriff auf einen Frame verwenden: GitHub [KinesisVideoRendererExampleTest.java](#)

Fragmentdekodierungsfehler

Wenn Ihre Fragmente in einem H.264 Format und einer Ebene, die der Browser unterstützt, nicht richtig codiert sind, wird beim Abspielen Ihres Streams in der Konsole möglicherweise der folgende Fehler angezeigt:

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 die Ebene der kodierten Frames entsprechen dem Profil und der Ebene, die in den privaten Codec-Daten angegeben sind.
- Der Browser unterstützt die profile/level Kombination. Die meisten aktuellen Browser unterstützen alle Profil- und Level-Kombinationen.
- Die Zeitstempel sind präzise und in der richtigen Reihenfolge, und es werden keine doppelten Zeitstempel erstellt.
- Ihre Anwendung codiert die Frame-Daten mithilfe des H.264 Formats.

Probleme mit dem Netzwerk

Wenn Sie beim Versuch, eine Verbindung zu Kinesis Video Streams herzustellen, Verbindungsfehler wie „Connection Timeout“ oder „Connection Failed“ (Verbindungs-Timeout) oder „Verbindung fehlgeschlagen“ sehen, kann das an Einschränkungen des IP-Adressbereichs in Ihrer Netzwerkkonfiguration liegen.

Wenn in Ihrem Setup Einschränkungen für den IP-Adressbereich von Kinesis Video Streams gelten, aktualisieren Sie Ihre Netzwerkkonfiguration, sodass die IP-Adressbereiche von Kinesis Video Streams in die Liste aufgenommen werden können. <https://ip-ranges.amazonaws.com/ip-ranges.json>

 Important

Die Liste der IP-Bereiche ist keine vollständige Liste der Kinesis Video Streams-IP-Adressen. Schließen Sie die angezeigten IP-Adressbereiche ein 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

Änderungen	Beschreibung	Date
C++ auf Raspberry Pi	Die Dokumentation für die Verwendung des C++-Producer-SDK auf Raspberry Pi wurde aktualisiert.	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 Senden Sie Daten an einen Amazon Kinesis-Videostream .	21. Januar 2019
Streaming-Metadaten	Sie können das Producer SDK verwenden, um Metadaten in einen Kinesis-Videostream einzubetten. Weitere Informationen finden Sie unter Verwenden von Streaming-	28. September 2018

Änderungen	Beschreibung	Date
	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 mithilfe von HTTP Live Streaming ansehen. Weitere Informationen finden Sie unter Kinesis-Videostreams-Wiedergabe.	13. Juli 2018
Streaming aus einer RTSP-Quelle	Beispielanwendung für Kinesis Video Streams, die in einem Docker-Container ausgeführt wird und Video von einer RTSP-Quelle streamt. Weitere Informationen finden Sie unter RTSP und Docker.	20. Juni 2018
C++ Producer SDK GStreamer-Plugin	Zeigt, wie das C++ erstellt werden muss, um es als GStreamer-Ziel zu verwenden. Weitere Informationen finden Sie unter GStreamer-Plugin — kvssink.	15. Juni 2018

Änderungen	Beschreibung	Date
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 Speicheranforderungen 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 Protokollieren Sie Amazon Kinesis Video Streams-API-Aufrufe 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

Änderungen	Beschreibung	Date
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: Parsen und Rendern von Kinesis Video 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-Videostreams .	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

Änderungen	Beschreibung	Date
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.