



Guida per gli sviluppatori

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: Guida per gli sviluppatori

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è Amazon Kinesis Video Streams?	1
Disponibilità nelle regioni	1
Come funziona	3
API e librerie per produttori	4
Modello di dati	8
Requisiti di sistema	16
Requisiti della fotocamera	17
Sistemi operativi testati	18
Requisiti di archiviazione SDK	18
Quote	18
Quote di servizio API del piano di controllo	18
Quote dei servizi API multimediali e archiviati	22
Fragment-metadata e quote di frammenti multimediali	28
Quote dei servizi di metadati di streaming	30
Quote SDK di Producer	31
Storage su più livelli	35
Panoramica di	35
Concetti chiave	35
Configurazione dello storage su più livelli	36
Best practice	36
Configurazione di un account	36
Registrati per un Account AWS	37
Crea un Account AWS Chiave	37
Vantaggi	37
Nozioni di base	39
Crea uno streaming video Amazon Kinesis	39
Crea uno streaming video usando la console	40
Crea uno streaming video utilizzando AWS CLI	40
Invia dati a un flusso video Amazon Kinesis	40
Crea l'SDK e gli esempi	41
Esegui gli esempi per caricare contenuti multimediali su Kinesis Video Streams	44
Rivedi gli oggetti di riconoscimento	46
Consuma dati multimediali	46
Visualizza i contenuti multimediali nella console	47

Consuma dati multimediali utilizzando HLS	47
Caricamento su Kinesis Video Streams	48
Cliente produttore di Kinesis Video Streams	48
Libreria per produttori di Kinesis Video Streams	49
Scopri cosa sono le librerie per produttori	50
Java	50
Procedura: utilizzate il Java producer SDK	50
Prerequisiti	51
Scarica e configura il codice	52
Scrivi ed esamina il codice	53
Esegui la pulizia delle risorse	54
Esegui e verifica il codice	55
Android	56
Procedura: utilizza l'Android producer SDK	56
Prerequisiti	56
Scarica e configura il codice	60
Esamina il codice	61
Esegui e verifica il codice	63
C++	65
Modello a oggetti	65
Inserisci contenuti multimediali nello stream	65
Interfacce di callback	66
Procedura: utilizzare l'SDK per produttori C++	66
Prerequisiti	66
Scarica e configura il codice	67
Scrivi ed esamina il codice	72
Esegui e verifica il codice	79
Usa l'SDK come plugin GStreamer	79
Usa il C++ producer SDK come plugin GStreamer in un contenitore Docker	80
Usa la registrazione	80
Produttore C	81
Modello a oggetti	81
Inserisci contenuti multimediali nello stream	82
Procedura: utilizzare C Producer SDK	82
Prerequisiti	83
Scarica il codice	84

Scrivi ed esamina il codice	84
Esegui e verifica il codice	87
C++ su Raspberry Pi	88
Prerequisiti	89
Creazione di un utente	89
Unisciti alla tua rete	91
Connettiti da remoto	92
Configura la videocamera	93
Prerequisiti per l'installazione del software	95
Scaricate e compilate l'SDK per produttori C++	96
Trasmetti in streaming un live streaming	98
Riproduci contenuti multimediali	110
Risoluzione dei problemi	113
Riferimento al codice di errore	118
Errori e codici di stato restituiti da PutFrame Callbacks - Platform Independent Code (PIC) .	118
Errori e codici di stato restituiti dai PutFrame callback - C producer library	168
Bandiere di adattamento NAL	176
Strutture dei produttori	177
DeviceInfo/DefaultDeviceInfoProvider	177
StorageInfo	178
Strutture di flusso	179
StreamDefinition/StreamInfo	180
ClientMetrics	196
StreamMetrics	197
Callback	199
ClientCallbackProvider	200
StreamCallbackProvider	200
ClientCallbacks	201
Implementazioni di callback per riprovare lo streaming	206
Utilizzo di metadati in streaming	207
Aggiungere metadati a un flusso video Kinesis	207
Usa IPv6 con Kinesis Video Streams	209
Configura il AWS SDK per IPv6	210
Configurazione dell'SDK di Kinesis Video Streams Producer per IPv6	211
Configura il AWS CLI per IPv6	213
Esempi di configurazione	214

Considerazioni	214
Clienti interessati dall'aggiornamento per l'inclusione di IPv6	215
Risoluzione dei problemi	217
Riproduzione video	218
Requisiti di riproduzione	219
GetClip	219
GetDASHStreamingSessionURL	220
GetHLSStreamingSessionURL	221
GetImages	222
Riproduzione con HLS	223
Usa il AWS CLI per recuperare l'URL di una sessione di streaming HLS	223
Esempio: usa HLS in HTML e JavaScript	227
Risoluzione dei problemi relativi all'HLS	231
Riproduzione con MPEG-DASH	234
Esempio: utilizzo in HTML e MPEG-DASH JavaScript	234
Configurare le notifiche	239
Gestisci le configurazioni delle notifiche	239
UpdateNotificationConfiguration	239
DescribeNotificationConfiguration	240
Informazioni sui tag MKV del produttore	240
Sintassi per i tag MKV del produttore	240
Limiti dei tag MKV	240
Messaggi Amazon SNS	241
Payload tematico di Amazon SNS	241
Visualizza i tuoi messaggi Amazon SNS	242
Cross-account Pubblicazione delle notifiche Amazon SNS	243
Identity-based configurazione delle politiche	243
Resource-based configurazione della politica	244
Requisiti e considerazioni	244
Estrai immagini dai flussi video	245
Real-time generazione di immagini	245
Aggiungere tag di generazione di immagini ai frammenti	248
Percorso dell'oggetto Amazon S3 (immagine)	251
Recupero dei metadati delle immagini	251
Raccomandazioni relative agli URI di Amazon S3 per la protezione dalla limitazione	252
Risoluzione dei problemi	253

Accedi all'analisi video	256
Consuma i metadati	256
Trasmetti in streaming utilizzando la libreria parser	257
Prerequisiti	258
Scarica il codice	258
Esamina il codice	259
Esegui il codice	266
Monitoraggio	266
Monitora le metriche con CloudWatch	267
Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch	286
Registra le chiamate API con CloudTrail	291
Limiti dei metadati di streaming	296
Pianifica la registrazione e l'archiviazione video	297
Operazioni dell'API Amazon Kinesis Video Streams Edge Agent	298
Monitoraggio di Amazon Kinesis Video Streams Edge Agent	298
Distribuisci in modalità non AWS IoT Greengrass	298
Installare le dipendenze	299
Crea risorse per gli URL RTSP della tua telecamera IP	301
Crea una politica di autorizzazioni IAM	303
Creazione di un ruolo IAM	305
Crea l'alias del AWS IoT ruolo	306
Crea la AWS IoT politica	307
Crea un AWS IoT oggetto e ottieni le AWS IoT Core credenziali	309
Crea l'Edge Agent	311
Installa l' CloudWatch agente	323
Esegui Edge Agent come processo nativo	326
Implementa su AWS IoT Greengrass	328
Crea un'istanza di Ubuntu	329
Configura il dispositivo AWS IoT Greengrass principale	331
Crea risorse per gli URL RTSP della tua telecamera IP	332
Aggiungi autorizzazioni al ruolo TES	334
Installa il componente Secret Manager	337
Distribuisci Edge Agent sul dispositivo	339
Installa il componente di gestione dei AWS IoT Greengrass log	348
Domande frequenti	352
Quali sistemi operativi sono supportati da Amazon Kinesis Video Streams Edge Agent?	353

Amazon Kinesis Video Streams Edge Agent supporta i contenuti multimediali? H.265	353
Amazon Kinesis Video Streams Edge Agent funziona in AL2?	353
Come posso eseguire più stream all'interno di AWS IoT cosa o dispositivo?	353
Come posso modificare un messaggio StartEdgeConfigurationUpdate dopo averlo inviato?	353
Hai qualche esempio comune ScheduleConfigs?	354
Esiste un limite massimo di streaming?	355
Come posso riavviare un processo che ha ricevuto un errore?	355
Come posso monitorare lo stato del mio Amazon Kinesis Video Streams Edge Agent?	355
Trasmetti video tramite un VPC	356
Informazioni aggiuntive	356
Procedure per gli endpoint VPC	356
Esempi	359
Esempi: invio di dati a Kinesis Video Streams	359
Esempi: recupero di dati da Kinesis Video Streams	359
Esempi: riproduzione di dati video	359
Prerequisiti	359
Plugin GStreamer - kvssink	360
Scarica, crea e configura l'elemento GStreamer	361
Esegui l'elemento GStreamer	361
Comandi di avvio	362
Esegui l'elemento GStreamer in un contenitore Docker	364
Documentazione di riferimento dei parametri	367
PutMedia API	381
Scarica e configura il codice	382
Scrivi ed esamina il codice	383
Esegui e verifica il codice	385
RTSP e Docker	386
Tutorial video	387
Prerequisiti	387
Create l'immagine Docker	387
Esegui l'applicazione di esempio RTSP	388
Renderer	389
Prerequisiti	390
Esempio di esecuzione del renderer	390
Come funziona	391

Sicurezza	393
Protezione dei dati	394
Cos'è la crittografia lato server per Kinesis Video Streams?	394
Costi, regioni e considerazioni sulle prestazioni	394
Come posso iniziare a usare la crittografia lato server?	395
Creazione e utilizzo di una chiave gestita dal cliente	396
Autorizzazioni per l'utilizzo di una chiave gestita dal cliente	396
Controllo dell'accesso alle risorse di Kinesis Video Streams tramite IAM	398
Sintassi delle policy	399
Azioni per Kinesis Video Streams	400
Amazon Resource Names (ARN) per Kinesis Video Streams	400
Concedere ad altri account IAM l'accesso a uno streaming video Kinesis	401
Policy di esempio	404
Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT	407
AWS IoT ThingName come nome dello stream	408
AWS IoT CertificateId come nome dello stream	414
Utilizzo AWS IoT credenziali per lo streaming verso un nome di streaming codificato	416
Convalida della conformità	417
Resilienza	418
Sicurezza dell'infrastruttura	418
Best practice di sicurezza	418
Implementazione dell'accesso con privilegi minimi	419
Uso di ruoli IAM	419
CloudTrail Usalo per monitorare le chiamate API	419
Risoluzione dei problemi	421
Problemi generali	421
Latenza troppo elevata	421
Problemi relativi alle API	422
Errore: "Opzioni sconosciute"	422
Errore: «Impossibile determinare service/operation il nome da autorizzare»	422
Errore: "Failed to put a frame in the stream" (Impossibile inserire un fotogramma nel flusso)	423
Errore: «Il servizio ha chiuso la connessione prima della ricezione della versione definitiva AckEvent »	423
Errore "STATUS_STORE_OUT_OF_MEMORY"	424
Errore: «La credenziale deve essere associata a una regione valida».	424

Problemi HLS	424
Problemi con Java	424
Abilitazione dei log di Java	424
Problemi con la libreria del produttore	425
Impossibile compilare l'SDK del produttore	426
Il flusso video non viene visualizzato nella console	426
Errore "Security token included in the request is invalid" durante lo streaming dei dati con l'applicazione demo GStreamer	427
Errore "Failed to submit frame to Kinesis Video client"	427
L'applicazione GStreamer viene arrestata con il messaggio "streaming stopped, reason not- negotiated" su OS X	428
Errore "Failed to allocate heap" durante la creazione del client Kinesis Video nella demo GStreamer su Raspberry Pi	428
Errore "Illegal Instruction" durante l'esecuzione della demo GStreamer su Raspberry Pi	428
La videocamera non viene caricata su Raspberry Pi	429
Impossibile trovare la videocamera su macOS High Sierra	429
File jni.h non trovato durante la compilazione su macOS High Sierra	430
Errori di curl durante l'esecuzione dell'applicazione demo GStreamer	430
Timestamp/range asserzione in fase di esecuzione su Raspberry Pi	430
Asserzione su <code>gst_value_set_fraction_range_full</code> su Raspberry Pi	430
STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) error on Android	430
È stata raggiunta la durata massima del frammento, errore	431
Errore "Invalid thing name passed (Passato nome di cosa non valido)" quando si utilizza l'autorizzazione IoT	431
Problemi con la libreria Stream Parser	432
Impossibile accedere a un singolo fotogramma del flusso	432
Errore di decodifica frammento	432
Problemi di rete	433
Cronologia dei documenti	434
.....	cdxxxix

Che cos'è Amazon Kinesis Video Streams?

Puoi utilizzare Amazon Kinesis Video Streams, una soluzione completamente gestita Servizio AWS, per trasmettere video in diretta dai dispositivi a Internet o creare applicazioni per l' Cloud AWS elaborazione video in tempo reale o l'analisi video orientata ai batch.

Kinesis Video Streams non è solo uno storage per dati video. ma anche a visualizzare i flussi video in tempo reale nel momento stesso in cui vengono ricevuti nel cloud. Puoi monitorare i tuoi live streaming in Console di gestione AWS, oppure sviluppare una tua applicazione di monitoraggio che utilizzi la libreria API Kinesis Video Streams per visualizzare video in diretta.

Puoi usare Kinesis Video Streams per acquisire enormi quantità di dati video in diretta da milioni di fonti, tra cui smartphone, telecamere di sicurezza, webcam, telecamere integrate in auto, droni e altre fonti. Puoi anche inviare dati non video e serializzati nel tempo, come dati audio, immagini termiche, dati di profondità e dati RADAR. Man mano che i video vengono trasmessi in diretta da queste sorgenti a un flusso video Kinesis, potete creare applicazioni per accedere ai dati, fotogramma per fotogramma, in tempo reale per un'elaborazione a bassa latenza. Kinesis Video Streams è indipendente dalla fonte. È possibile eseguire lo streaming di video dalla webcam di un computer utilizzando la [Plugin GStreamer - kvssink](#) libreria o da una telecamera in rete utilizzando il protocollo di streaming in tempo reale (RTSP).

È inoltre possibile configurare lo streaming video Kinesis per archiviare in modo duraturo i dati multimediali per il periodo di conservazione specificato. Kinesis Video Streams archivia automaticamente questi dati e li crittografa quando sono inattivi. Inoltre, Kinesis Video Streams indicizza i dati archiviati in base sia ai timestamp del produttore che ai timestamp di ingestion. È possibile creare applicazioni che elaborano periodicamente in batch i dati video oppure applicazioni che richiedono l'accesso una tantum ai dati storici per diversi casi d'uso.

Le tue applicazioni personalizzate, in tempo reale o orientate ai batch, possono essere eseguite su istanze Amazon EC2. Queste applicazioni potrebbero elaborare dati utilizzando algoritmi open source di deep learning o utilizzare applicazioni di terze parti che si integrano con Kinesis Video Streams.

Disponibilità nelle regioni

Amazon Kinesis Video Streams è disponibile nelle seguenti regioni:

Nome della regione	AWS Codice regionale
Stati Uniti orientali (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
Stati Uniti occidentali (Oregon)	us-west-2
AWS GovCloud (US-East)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1
Africa (Città del Capo)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Malesia)	ap-southeast-3
Asia Pacifico (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
Europa (Francoforte)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Spagna)	eu-south-2

Nome della regione	AWS Codice regionale
Medio Oriente (Bahrein)	me-south-1
Sud America (San Paolo)	sa-east-1

Kinesis Video Streams: come funziona

Argomenti

- [Supporto per le API e le librerie di produzione di Kinesis Video Streams](#)
- [Modello di dati Kinesis Video Streams](#)

Puoi usare Amazon Kinesis Video Streams, una soluzione completamente gestita Servizio AWS, per trasmettere video in diretta dai dispositivi ai dispositivi Cloud AWS e archivarli in modo duraturo. Puoi quindi creare le tue applicazioni per l'elaborazione di video in tempo reale o eseguire analisi dei video in batch.

Il diagramma seguente fornisce una panoramica del funzionamento di Kinesis Video Streams.

Il diagramma mostra l'interazione tra i componenti seguenti:

- **Produttore:** qualsiasi fonte che inserisce dati in un flusso video Kinesis. Un produttore può essere qualsiasi dispositivo per la generazione di video, come una telecamera di sicurezza, una videocamera indossabile, una fotocamera per smartphone o una telecamera da cruscotto. Un producer può anche inviare dati non video, come feed audio, immagini o dati RADAR.

Un producer singolo può generare uno o più flussi video. Ad esempio, una videocamera può inviare dati video a un flusso video Kinesis e dati audio a un altro.

- **Librerie per produttori di Kinesis Video Streams:** un set di software e librerie che puoi installare e configurare sui tuoi dispositivi. Puoi utilizzare queste librerie per connettere in modo sicuro e trasmettere video in streaming in modo affidabile in diversi modi, ad esempio in tempo reale, dopo averli memorizzati nel buffer per alcuni secondi o come carichi multimediali a posteriori.

- **Kinesis video stream:** una risorsa che puoi utilizzare per trasportare dati video in diretta, archivarli facoltativamente e renderli disponibili per il consumo sia in tempo reale che in batch o una tantum. In una configurazione tipica, un flusso video Kinesis prevede un solo produttore che pubblica i dati.

Il flusso è in grado di trasportare audio e flussi di dati simili codificati temporalmente, ad esempio feed di rilevamento profondità, feed RADAR e molto altro. È possibile creare un flusso video Kinesis utilizzando Console di gestione AWS o utilizzando programmaticamente gli SDK. AWS

Più applicazioni indipendenti possono utilizzare un flusso video Kinesis in parallelo.

- **Consumer:** ottiene dati, come frammenti e frame, da un flusso video Kinesis per visualizzarli, elaborarli o analizzarli. Generalmente questi utenti sono denominati applicazioni Kinesis Video Streams. È possibile scrivere applicazioni che consumano ed elaborano dati in Kinesis Video Streams in tempo reale o dopo che i dati sono stati archiviati e indicizzati nel tempo, quando non è richiesta un'elaborazione a bassa latenza. Puoi creare queste applicazioni consumer da eseguire su istanze Amazon EC2.
- [Guarda l'output delle telecamere usando la libreria parser](#)— Consente alle applicazioni Kinesis Video Streams di ottenere in modo affidabile contenuti multimediali dallo streaming video Kinesis in modo a bassa latenza. Inoltre, analizza i limiti del frame nel contenuto multimediale in modo che le applicazioni possano concentrarsi sull'elaborazione e l'analisi dei frame stesso.

Supporto per le API e le librerie di produzione di Kinesis Video Streams

Kinesis Video Streams fornisce API per creare e gestire stream e leggere o scrivere dati multimediali da e verso uno stream. La console Kinesis Video Streams, oltre alla funzionalità di amministrazione, supporta anche la riproduzione live e di video on demand. Kinesis Video Streams fornisce anche una serie di librerie di produttori che potete utilizzare nel codice dell'applicazione per estrarre dati dalle vostre sorgenti multimediali e caricarli nel vostro streaming video Kinesis.

Argomenti

- [API Kinesis Video Streams](#)
- [Modello di individuazione degli endpoint](#)
- [Librerie di produttori](#)

API Kinesis Video Streams

Kinesis Video Streams fornisce API per creare e gestire Kinesis Video Streams. Fornisce inoltre API per la lettura e la scrittura di dati multimediali su un flusso, nel modo seguente:

- API per produttori: Kinesis Video Streams fornisce un'PutMediaAPI per scrivere dati multimediali su uno streaming video Kinesis. In una richiesta PutMedia, il producer invia un flusso di frammenti multimediali. Per frammento si intende una sequenza autonoma di frame. I frame appartenenti a un frammento non devono avere alcun tipo di dipendenza dai frame di altri frammenti. Per ulteriori informazioni, consulta [PutMedia](#).

Man mano che arrivano i frammenti, Kinesis Video Streams assegna un numero di frammento univoco, in ordine crescente. Inoltre, memorizza i timestamp lato produttore e lato server per ogni frammento, come metadati Kinesis Video. Streams-specific

- API consumer: i consumatori possono utilizzare le seguenti API per ottenere dati da uno stream:
 - GetMedia: quando si utilizza questa API, i consumer devono identificare il frammento di partenza. L'API restituisce i frammenti nell'ordine in cui sono stati aggiunti nel flusso (in ordine crescente per numero di frammento). I dati multimediali nel frammento vengono compressi in un formato strutturato, ad esempio [Matroska \(MKV\)](#). Per ulteriori informazioni, consulta [GetMedia](#).

Note

GetMedia sa dove sono i frammenti (archiviati nello datastore o disponibili in tempo reale). Ad esempio, se GetMedia determina che i frammenti di partenza sono archiviati, ne avvia la restituzione dal datastore. Quando deve restituire frammenti più recenti che non sono ancora stati archiviati, GetMedia passa alla lettura di frammenti da un buffer di flusso in memoria.

Questo è un esempio di un consumer continuo, che elabora i frammenti nell'ordine in cui vengono acquisiti dal flusso.

GetMedia permette che le applicazioni di elaborazione video falliscano o accumulino ritardo, recuperando senza alcuno sforzo aggiuntivo. Tramite GetMedia, le applicazioni sono in grado di elaborare i dati archiviati nel datastore e man mano che l'applicazione si aggiorna, GetMedia continua ad aggiornare i dati multimediali in tempo reale al loro arrivo.

- `GetMediaFromFragmentList` (e `ListFragments`): le applicazioni per l'elaborazione in batch sono considerate consumer offline. I consumatori offline potrebbero scegliere di recuperare in modo esplicito particolari frammenti di file multimediali o intervalli di video combinando le API `ListFragments` e `GetMediaFromFragmentList`. `ListFragments` e `GetMediaFromFragmentList` consentono a un'applicazione di identificare i segmenti di video per un determinato intervallo di tempo o intervallo di frammenti, quindi di recuperare i frammenti in modo sequenziale o in parallelo per l'elaborazione. Questo approccio è idoneo per suite di applicazioni MapReduce, che devono elaborare rapidamente grandi quantità di dati in parallelo.

Ad esempio, supponiamo che un consumer voglia elaborare i frammenti di video di un intero giorno. Il consumer procederebbe nel modo seguente:

1. Ottenere un elenco di frammenti chiamando l'API `ListFragments` e specificando un intervallo di tempo per selezionare la raccolta di frammenti desiderata.

L'API restituisce i metadati da tutti i frammenti nell'intervallo di tempo specificato. I metadati forniscono informazioni come il numero dei frammenti, i timestamp sul lato produttore e sul lato server e così via.

2. Prendere l'elenco dei metadati dei frammenti e recuperare i frammenti, in qualsiasi ordine. Ad esempio, per elaborare tutti i frammenti del giorno, il consumatore potrebbe scegliere di suddividere l'elenco in sottoelenchi e chiedere ai lavoratori (ad esempio, più istanze di Amazon EC2) di recuperare i frammenti in parallelo utilizzando e di elaborarli in parallelo. `GetMediaFromFragmentList`

Il seguente diagramma mostra il flusso di dati per frammenti e blocchi durante queste chiamate API.

Quando un producer invia una richiesta `PutMedia`, invia metadati multimediali nel payload nonché una sequenza di frammenti di dati multimediali. Una volta ricevuti i dati, Kinesis Video Streams memorizza i dati multimediali in entrata come blocchi di Kinesis Video Streams. Ogni blocco è costituito dagli elementi seguenti:

- Una copia dei metadati multimediali
- Un frammento
- Streams-specific Metadati di Kinesis Video, ad esempio il numero del frammento e i timestamp lato server e lato produttore

Quando un utente richiede metadati multimediali, Kinesis Video Streams restituisce un flusso di blocchi, a partire dal numero di frammento specificato nella richiesta.

Se abiliti la persistenza dei dati per lo stream, dopo aver ricevuto un frammento sullo stream, Kinesis Video Streams salva anche una copia del frammento nel data store.

Modello di individuazione degli endpoint

API REST del piano di controllo

Per accedere alle API REST di [Kinesis Video Streams Control Plane](#), utilizzate gli endpoint del servizio [Kinesis Video Streams](#).

API REST Data Plane

Kinesis Video Streams è costruito utilizzando un'architettura [cellulare](#) per garantire migliori proprietà di scalabilità e isolamento del traffico. Poiché ogni stream è mappato su una cella specifica in una regione, l'applicazione deve utilizzare gli endpoint specifici della cella corretti su cui è stato mappato lo stream. Quando accedi alle API REST di Data Plane, dovrai gestire e mappare tu stesso gli endpoint corretti. Questo processo, il modello di rilevamento degli endpoint, è descritto di seguito:

1. Il pattern di rilevamento degli endpoint inizia con una chiamata a una delle `GetEndpoints` azioni. Queste azioni appartengono al piano di controllo.
 1. Se stai recuperando gli endpoint per i servizi [Amazon Kinesis Video Streams Media API](#) o [Amazon Kinesis Video Streams Archived Media API](#), utilizza [GetDataEndpoint](#)
 2. Se stai recuperando gli endpoint per l'API Amazon Kinesis Video Signaling Channels, l'API di [storage Amazon Kinesis Video WebRTC](#) o l'API di [storage Kinesis Video Signaling](#), utilizza <https://docs.aws.amazon.com/kinesisvideostreams-webrtc-dg/latest/devguide/kvswebrtc-websocket-apis.html> [GetSignalingChannelEndpoint](#)
2. Memorizza nella cache e riutilizza l'endpoint.
3. Se l'endpoint memorizzato nella cache non funziona più, effettua una nuova chiamata `GetEndpoints` a per aggiornare l'endpoint.

Librerie di produttori

Dopo aver creato uno stream video Kinesis, puoi iniziare a inviare dati allo stream. Nel codice dell'applicazione, puoi utilizzare queste librerie per estrarre dati dalle tue fonti multimediali e caricarli

nel tuo streaming video Kinesis. Per ulteriori informazioni sulle librerie di producer disponibili, consulta [Caricamento su Kinesis Video Streams](#).

Modello di dati Kinesis Video Streams

[Caricamento su Kinesis Video Streams](#) e [the section called “Trasmetti in streaming utilizzando la libreria parser”](#) inviano e ricevono dati video in un formato che supporta l'incorporamento di informazioni insieme ai dati video. Questo formato si basa sulla specifica Matroska (MKV).

Il [formato MKV](#) è una specifica aperta per i dati multimediali. Tutte le librerie e gli esempi di codice contenuti nella [Amazon Kinesis Video Streams Developer Guide](#) inviano o ricevono dati in formato MKV.

[Caricamento su Kinesis Video Streams](#) Utilizza i Frame tipi `StreamDefinition` e per produrre intestazioni di stream MKV, intestazioni di frame e dati di frame.

Per informazioni sulle specifiche complete di MKV, consulta l'articolo sulle [specifiche Matroska](#).

Le sezioni seguenti descrivono i componenti dei MKV-formatted dati prodotti da [C++](#)

Argomenti

- [Elementi di intestazione dello stream](#)
- [Trasmetti i dati della traccia](#)
- [Elementi dell'intestazione del frame](#)
- [Dati del frame MKV](#)

Elementi di intestazione dello stream

I seguenti elementi di intestazione MKV vengono utilizzati da `StreamDefinition` (definiti in `StreamDefinition.h`).

Elemento	Description	Valori tipici
<code>stream_name</code>	Corrisponde al nome del flusso video Kinesis.	<code>my-stream</code>
<code>retention_period</code>	La durata, in ore, della persistenza dei dati dello	24

Elemento	Description	Valori tipici
	streaming da parte di Kinesis Video Streams. Specifica 0 uno stream che non conserva dati.	
tags	Una raccolta chiave-valore di dati utente. Questi dati vengono visualizzati nella Console di gestione AWS e possono essere letti dalle applicazioni client per filtrare o ottenere informazioni su un flusso.	
kms_key_id	Se presente, la AWS KMS chiave definita dall'utente viene utilizzata per crittografare i dati sullo stream. Se assente, i dati vengono crittografati dalla Kinesis-s applied chiave (). aws/kinesisvideo	01234567-89ab-cdef-0123-456789ab
streaming_type	Attualmente, l'unico tipo di streaming valido è STREAMING_TYPE_REALTIME .	STREAMING_TYPE_REALTIME
content_type	Il tipo di contenuto definito dall'utente. Per lo streaming di dati video da riprodurre nella console, il tipo di contenuto deve essere video/h264 .	video/h264

Elemento	Description	Valori tipici
max_latency	Questo valore non è attualmente utilizzato e deve essere impostato su 0.	0
fragment_duration	La stima della durata dei tuoi frammenti, utile per l'ottimizzazione. La durata effettiva è determinata dai dati di streaming.	2
timecode_scale	Indica la scala utilizzata dai timestamp dei frame. Il valore predefinito è 1 millisecondo. Se si specifica 0, si assegna anche il valore predefinito di 1 millisecondo. Questo valore può essere compreso tra 100 nanosecondi e 1 secondo. Per ulteriori informazioni, consultate la documentazione TimecodeScale di Matroska.	
key_frame_fragmentation	Se true, il flusso avvia un nuovo cluster quando riceve un keyframe.	true

Elemento	Description	Valori tipici
frame_timecodes	Se <code>true</code> , Kinesis Video Streams utilizza i valori del timestamp di presentazione (pts) e del timestamp di decodifica (dts) dei frame ricevuti. Se <code>false</code> , Kinesis Video Streams contrassegna i frame quando vengono ricevuti con valori temporali generati dal sistema.	<code>true</code>
absolute_fragment_time	Se <code>true</code> , i timecode del cluster vengono interpretati come se utilizzassero tempo assoluto (ad esempio, dall'orologio di sistema del producer). Se <code>false</code> , i timecode del cluster sono interpretati come relativi all'ora di inizio del flusso.	<code>true</code>
fragment_acks	Se <code>true</code> , i riconoscimenti (ACK) vengono inviati quando Kinesis Video Streams riceve i dati. Le ACK possono essere ricevute utilizzando le callback <code>KinesisVideoStreamFragmentAck</code> o <code>KinesisVideoStreamParseFragmentAck</code> .	<code>true</code>
restart_on_error	Indica se il flusso deve riprendere la trasmissione dopo la segnalazione di un errore di flusso.	<code>true</code>

Elemento	Description	Valori tipici
<code>nal_adaptation_flags</code>	Indica se i dati privati del codec o l'adattamento NAL (Network Abstraction Layer) sono presenti nel contenuto. I flag validi includono <code>NAL_ADAPTATION_ANNEXB_NALS</code> e <code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code> .	<code>NAL_ADAPTATION_ANNEXB_NALS</code>
<code>frame_rate</code>	Una stima della frequenza di frame del contenuto. Tale valore viene utilizzato per l'ottimizzazione. La frequenza di frame effettiva è determinata dalla velocità dei dati in entrata. Se si specifica <code>0</code> , si assegna il valore predefinito di <code>24</code> .	<code>24</code>
<code>avg_bandwidth_bps</code>	Una stima della larghezza di banda dei contenuti, in Mbps. Tale valore viene utilizzato per l'ottimizzazione. La frequenza effettiva è determinata dalla larghezza di banda dei dati in entrata. Ad esempio, per un flusso video con risoluzione di 720 p in esecuzione a 25 FPS, puoi aspettarti una larghezza di banda media di 5 Mbps.	<code>5</code>

Elemento	Description	Valori tipici
buffer_duration	La durata del buffering del contenuto sul producer. Se la latenza di rete è bassa, questo valore può essere ridotto. Se la latenza di rete è elevata, l'aumento di questo valore impedisce che i frame vengano eliminati prima di poter essere inviati, a causa del mancato inserimento dei frame nel buffer più piccolo nell'allocazione.	
replay_duration	Periodo di tempo in cui il flusso di dati video viene «riavvolto» in caso di interruzione della connessione. Questo valore può essere zero se la perdita di frame dovuta alla perdita della connessione non è un problema. Il valore può essere aumentato se l'applicazione che lo utilizza è in grado di rimuovere i frame ridondanti. Questo valore deve essere inferiore alla durata del buffer, altrimenti viene utilizzato a la durata del buffer.	
connection_staleness	Il periodo durante il quale una connessione viene mantenuta quando non vengono ricevuti dati.	

Elemento	Description	Valori tipici
codec_id	Il codec utilizzato per il contenuto. Per ulteriori informazioni, consulta CodecID nelle specifiche Matroska.	V_MPEG2
track_name	Il nome definito dall'utente della traccia.	my_track
codec PrivateData	I dati forniti dall'encoder utilizzato per decodificare i dati dei frame, ad esempio la larghezza di banda e l'altezza in pixel degli stessi, un'informazione richiesta da numerosi consumer a valle. Nella libreria di produzione C++ , l'gMkvTrackVideoBits array MkvStatics.cpp include la larghezza e l'altezza dei pixel per il frame.	
codec PrivateDataSize	La dimensione dei dati nel parametro codecPrivateData .	
track_type	Il tipo di traccia per il flusso.	MKV_TRACK_INFO_TYPE_AUDIO or MKV_TRACK_INFO_TYPE_VIDEO
segment_uuid	User-defined uuid del segmento (16 byte).	
default_track_id	Numero univoco diverso da zero per la traccia.	1

Trasmetti i dati della traccia

I seguenti elementi della traccia MKV vengono utilizzati da `StreamDefinition` (definiti in `StreamDefinition.h`).

Elemento	Description	Valori tipici
track_name	User-defined nome della traccia. Ad esempio, "audio" per la traccia audio.	audio
codec_id	Id codec della traccia. Ad esempio, "A_AAC" per la traccia audio.	A_AAC
cpd	I dati forniti dall'encoder utilizzato per decodificare i dati del frame. Queste informazioni possono includere larghezza e altezza del frame in pixel, necessarie per numerosi consumatori downstream. Nella libreria di produzione C++ , l'MkvTrackVideoBits array g MkvStatics.cpp include la larghezza e l'altezza dei pixel per il frame.	
cpd_size	La dimensione dei dati nel parametro codecPrivateData .	
track_type	Il tipo di traccia. Ad esempio, è possibile utilizzare il valore di enum MKV_TRACK_INFO_TYPE_AUDIO per l'audio.	MKV_TRACK_INFO_TYPE_AUDIO

Elementi dell'intestazione del frame

I seguenti elementi di intestazione MKV vengono utilizzati da `Frame` (definiti nel pacchetto `KinesisVideoPic`, in `mkvgen/Include.h`):

- **Frame Index (Indice del frame):** un valore a incremento monotono.
- **Flags (Flag):** il tipo di frame. I valori validi includono quanto segue:
 - `FRAME_FLAGS_NONE`
 - `FRAME_FLAG_KEY_FRAME`: se `key_frame_fragmentation` è impostato su flusso, i key frame avviano un nuovo frammento.
 - `FRAME_FLAG_DISCARDABLE_FRAME`: indica al decoder che è possibile ignorare questo frame se la decodifica è lenta.
 - `FRAME_FLAG_INVISIBLE_FRAME`: la durata di questo blocco è 0.
- **Timestamp di decodifica:** il timestamp di quando questo frame è stato decodificato. Se i frame precedenti dipendono da questo frame per la decodifica, questo timestamp potrebbe essere precedente a quello dei frame precedenti. Questo valore è relativo all'inizio del frammento.
- **Timestamp di presentazione:** il timestamp di quando viene visualizzato questo frame. Questo valore è relativo all'inizio del frammento.
- **Duration (Durata):** la durata di riproduzione del frame.
- **Size (Dimensione):** la dimensione dei dati del frame in byte

Dati del frame MKV

I dati in `frame.frameData` potrebbero contenere solo i dati multimediali per il frame o ulteriori informazioni di intestazione annidate, a seconda dello schema di codifica utilizzato. Per essere visualizzati in Console di gestione AWS, i dati devono essere codificati nel [H.264](#) codec, ma Kinesis Video Streams può ricevere flussi di dati serializzati nel tempo in qualsiasi formato.

Requisiti di sistema di Amazon Kinesis Video Streams

Le seguenti sezioni contengono i requisiti hardware, software e di storage per Amazon Kinesis Video Streams.

Argomenti

- [Requisiti della fotocamera](#)

- [Sistemi operativi testati](#)
- [Requisiti di archiviazione SDK](#)

Requisiti della fotocamera

Le videocamere utilizzate per eseguire l'SDK e gli esempi del produttore di Kinesis Video Streams hanno i seguenti requisiti di memoria:

- La visualizzazione dei contenuti SDK richiede 16 MB di memoria.
- La configurazione di default dell'applicazione di esempio è di 512 MB. Si tratta di un valore appropriato per i producer con una buona connettività di rete e senza requisiti di buffering aggiuntivo. Se la connessione di rete è scadente ed è necessario ulteriore buffering, è possibile calcolare la memoria richiesta per ogni secondo di buffering moltiplicando il frame rate al secondo per le dimensioni di memoria del fotogramma. Per ulteriori informazioni sull'allocazione della memoria, consultare [StorageInfo](#).

Consigliamo di utilizzare telecamere USB o RTSP (Real Time Streaming Protocol) che codificano i dati in modo da rimuovere il carico di lavoro H.264 di codifica dalla CPU.

Attualmente, l'applicazione demo non supporta lo User Datagram Protocol (UDP) per lo streaming RTSP. Questa funzionalità verrà aggiunta in futuro.

L'SDK del produttore supporta i seguenti tipi di telecamere:

- Videocamere Web
- Videocamere USB
- Fotocamere con H.264 codifica (preferibile).
- Fotocamere senza H.264 codifica.
- Modulo telecamera Raspberry Pi. Si tratta dell'opzione preferita per i dispositivi Raspberry Pi perché si connette alla GPU per il trasferimento di dati video senza includere l'elaborazione da parte della CPU.
- Videocamere RTSP (di rete). Queste telecamere sono preferite perché i flussi video sono già codificati con H.264

Sistemi operativi testati

Le videocamere Web e RTSP sono state testate con i seguenti dispositivi e sistemi operativi:

- Mac mini
 - High Sierra
- MacBook Computer portatili Pro
 - Sierra (10.12)
 - El Capitan (10.11)
- Laptop HP con Ubuntu 16.04
- Ubuntu 17.10 (container Docker)
- Raspberry Pi 3

Requisiti di archiviazione SDK

Per l'installazione delle [Caricamento su Kinesis Video Streams](#) è richiesta una capacità di storage minima di 170 MB e sono raccomandati 512 MB.

Quote del servizio Amazon Kinesis Video Streams

Kinesis Video Streams ha le seguenti quote di servizio:

Important

Le seguenti quote di servizio sono o soft [s], che possono essere aggiornate inviando un ticket di supporto, o rigide [h], che non possono essere aumentate. Nelle tabelle seguenti verranno visualizzati [s] e [h] accanto alle quote di servizio individuali.

Quote di servizio API del piano di controllo

La sezione seguente descrive le quote di servizio per le API del piano di controllo. TPS è l'acronimo di Transactions per second.

Quando viene raggiunto un limite di richieste a livello di account o di risorsa, viene generato un `ClientLimitExceededException`

"Hello, World!"	Limite dell'account: Richiesta	Limite dell'account: Streams	Stream-level limite	Eccezioni e note pertinenti
CreateStream	50 TPS	10.000 stream per account in tutte le regioni supportate.		I dispositivi, le interfacce della riga di comando, gli SDK-driven accessi e la console possono tutti richiamare questa API. Se il flusso non è già esistente, va a buon fine una sola chiamata API.
		 Note Questo limite può essere aumentato fino a 100.000 (o più) flussi per account. Accedi Console di gestione AWS all'indirizzo https://console.aws.amazon.com/ e		

"Hello, World!"	Limite dell'account: Richiesta	Limite dell'account: Streams	Stream-level limite	Eccezioni e note pertinenti
		richiedi un aumento di questo limite.		
DeleteEdgeConfiguration	10 TPS [r]	N/A	1 TPS [h]	
DeleteStream	50 TPS [r]	N/A	5 TPS [r]	
DescribeEdgeConfiguration	50 TPS [r]	N/A	5 TPS [r]	
DescribeImageGenerationConfiguration	50 TPS [r]	N/A	5 TPS [r]	
DescribeMappedResourceConfiguration	50 TPS [r]	N/A	5 TPS [r]	
DescribeNotificationConfiguration	50 TPS [r]	N/A	5 TPS [r]	
DescribeStream	300 TPS [r]	N/A	5 TPS [r]	

"Hello, World!"	Limite dell'account: Richiesta	Limite dell'account: Streams	Stream-level limite	Eccezioni e note pertinenti
GetDataEndpoint	300 TPS [r]	N/A	5 TPS [r]	Viene chiamata ogni 45 minuti per aggiornare il token di streaming per la maggior parte dei casi d'uso PutMedia/GetMedia. Il caching dei dati dell'endpoint è un'operazione sicura se l'applicazione ricarica i dati in caso di errore.
ListEdgeAgentConfigurations	50 TPS [r]	N/A	N/A	
ListStreams	50 TPS [r]	N/A		
ListTagsForStream	50 TPS [r]	N/A	5 TPS [r]	
StartEdgeConfigurationUpdate	10 TPS [r]	N/A	1 TPS [h]	
TagStream	50 TPS [r]	N/A	5 TPS [r]	
UntagStream	50 TPS [r]	N/A	5 TPS [r]	
UpdateDataRetention	50 TPS [r]	N/A	5 TPS [r]	
UpdateImageGenerationConfiguration	50 TPS [r]	N/A	5 TPS [r]	

"Hello, World!"	Limite dell'account: Richiesta	Limite dell'account: Streams	Stream-level limite	Eccezioni e note pertinenti
UpdateNotificationConfiguration	50 TPS [r]	N/A	5 TPS [r]	
UpdateStream	50 TPS [r]	N/A	5 TPS [r]	
UpdateStreamStorageConfiguration	50 TPS [r]	N/A	5 TPS [r]	
DescribeStreamStorageConfiguration	50 TPS [r]	N/A	5 TPS [r]	

Quote dei servizi API multimediali e archiviati

La sezione seguente descrive le quote di servizio per le API multimediali e archiviate.

Quando viene raggiunto un limite di richieste a livello di account o di risorsa, viene generato un `ClientLimitExceededException`.

Quando viene raggiunto un limite a livello di connessione, viene generata un'eccezione `ConnectionLimitExceededException`.

Quando viene raggiunto un limite a livello di frammento, vengono generati i seguenti errori o messaggi ACK:

- Se un frammento è inferiore alla durata minima, viene restituito il messaggio ACK `MIN_FRAGMENT_DURATION_REACHED`.
- Se un frammento è superiore alla durata massima, viene restituito il messaggio ACK `MAX_FRAGMENT_DURATION_REACHED`.

- Se un frammento supera le dimensioni massime dei dati, viene restituito il messaggio ACK `MAX_FRAGMENT_SIZE`.
- Viene generata un'eccezione `FragmentLimitExceeded` se viene raggiunto un limite a livello di frammento in un'operazione `GetMediaForFragmentList`.

Quote dei servizi API Data Plane

"Hello, World!"	Stream-level limite	Connection-level limite	Limite larghezza di banda	Fragment-level limite	Eccezioni e note pertinenti
PutMedia	5 TPS [r]	1 [h]	12,5 MB/secondo o 100 Mbps [s] per stream	• Durata minima frammento : 1 secondo [r] • Durata massima del frammento : 20 secondi [h] • Dimensione massima frammento : 50 MB [r] • Numero massimo di brani: 3 [s]	PutMedia le richieste sono connessioni in streaming e di lunga durata. Non è necessario aprire una nuova connessione per ogni dato perché è possibile inviare più frammenti in un'unica connessione persistente. Se tenti più di una <code>PutMedia</code> connessione simultanea, Kinesis Video Streams limita le connessioni più recenti con un messaggio di errore. <code>ConnectionLimitExceededException</code>

"Hello, World!"	Stream-level limite	Connection-level limite	Limite larghezza di banda	Fragment-level limite	Eccezioni e note pertinenti
				• Numero massimo di frammenti inviati al secondo: 5 [h] • Limite massimo di metadati per frammenti: 10 tag [h]	
GetClip	N/A	N/A	Limite di dimensione 100 MB [h]	Numero massimo di frammenti: 200 [h]	
GetDASHStreamingSessionURL	25 SUGGERIMENTI [h]	N/A	N/A	N/A	
GetHLSStreamingSessionURL	25 TPS [h]	N/A	N/A	N/A	

"Hello, World!"	Stream-level limite	Connection-level limite	Limite larghezza di banda	Fragment-level limite	Eccezioni e note pertinenti
GetImages	N/A	N/A	100 MB [h]	N/A	Il numero massimo di immagini per richiesta è 100 [h].  Note Il valore minimo per SamplingInterval è 200 millisecondi (ms), ovvero 5 immagini al secondo.

"Hello, World!"	Stream-level limite	Connection-level limite	Limite larghezza di banda	Fragment-level limite	Eccezioni e note pertinenti
GetMedia	5 TPS [r]	3 [h]	25 MB/s o 200 Mbps [s]	Massimo 5 frammenti inviati al secondo [h]	GetMedia richiede sono connessioni in streaming e di lunga durata. Non è necessario aprire una nuova connessione per ogni dato perché è possibile inviare più frammenti in un'unica connessione persistente. Se si tentano più di tre GetMedia connessioni simultanee, Kinesis Video Streams limita le connessioni più recenti con un messaggio di errore. <code>ConnectionLimitExceededException</code> Se un frammento tipico è di circa 5 MB, questo limite equivale a ~75 MBps per flusso video Kinesis. Un tale flusso avrebbe un bitrate in uscita doppio rispetto al massimo bitrate in entrata per i flussi.  Note GetMedia non viene utilizzato

"Hello, World!"	Stream-level limite	Connection-level limite	Limite larghezza di banda	Fragment-level limite	Eccezioni e note pertinenti
					per la riproduzione. HLS/DASH
GetMediaForFragmentList	N/A	5 [s]	25 MB/s o 200 Mbps [s]	Numero massimo di frammenti: 1.000 [h]	È possibile richiamare contemporaneamente cinque applicazioni che utilizzano frammenti. GetMediaForFragmentList Eventuali altre connessioni vengono rifiutate.

Quote dei servizi API del protocollo di riproduzione video

"Hello, World!"	Session-level limite	Fragment-level limite
GetDASHManifestPlaylist	5 TPS [r]	Numero massimo di frammenti per playlist: 5.000 [h]
GetHLSMasterPlaylist	5 TPS [r]	N/A
GetHLSMediaPlaylist	5 TPS [r]	Numero massimo di frammenti per playlist: 5.000 [h]
GetMP4InitFragment	5 TPS [r]	N/A
GetMP4MediaFragment	20 TPS [h]	N/A
GetTSFragment	20 TPS [h]	N/A

Fragment-metadata e quote di frammenti multimediali

[Le API Kinesis Video Streams per l'accesso ai](#) contenuti multimediali archiviati vengono limitate in base al numero di frammenti richiesti anziché al numero di chiamate API. Le API sono limitate in base alla velocità sia dal numero di metadati dei frammenti che dal numero di frammenti multimediali richiesti. I metadati dei frammenti e le quote multimediali dei frammenti vengono applicati per stream. In altre parole, le richieste di metadati o contenuti multimediali di frammento in uno stream non si applicano alle quote di un altro stream. Tuttavia, all'interno di un determinato flusso, ogni quota è condivisa tra più API. Ciò significa che, per un determinato flusso, le richieste di frammenti tra API diverse utilizzano la stessa quota. Quando viene superata la quota dei metadati dei frammenti o dei frammenti multimediali per uno stream, l'API restituisce un `ClientLimitExceededException`. Le tabelle seguenti mostrano come le API utilizzano ciascuno dei due tipi di quota. Per la seconda colonna di queste tabelle, supponiamo che se uno stream ha una quota di N, ciò significa che le API hanno N punti da utilizzare da quel tipo di quota per quel flusso. L'GetClipAPI appare in entrambe le tabelle.

Fragment-metadata quota di consumo

"Hello, World!"	Numero di punti di quota consumati per richiesta	Quota condivisa (N)
ListFragments	Valore del MaxResults parametro	10.000 punti quota al secondo, per stream [h]
GetClip	Numero di frammenti nella clip risultante	
GetHLSMediaPlaylist	Valore del parametro MaxMediaPlaylistFragmentResults	
GetDASHManifest	Valore del MaxManifestFragmentResults parametro	
GetImages	Valore di oltre 400 (numero massimo di immagini richieste)	

Fragment-media quota di consumo

"Hello, World!"	Numero di punti di quota consumati per richiesta	Quota condivisa (N)
GetMediaForFragmentList	Numero di frammenti nel parametro Fragments	500 punti di quota al secondo, per stream [h]
GetClip	Numero di frammenti nella clip risultante	
GetMP4MediaFragment	1	
GetTSFragment	1	
GetImages	Numero massimo di immagini richieste	

Ad esempio, con una quota di 500 frammenti multimediali al secondo, sono supportati i seguenti pattern di chiamata per un determinato flusso:

- 5 richieste al secondo GetClip con 100 frammenti in ogni clip.
- 100 richieste al secondo GetClip con 5 frammenti in ogni clip.
- 2 richieste al secondo per GetClip 100 frammenti in ogni clip e 3 richieste al secondo per GetMediaForFragmentList ogni clip.
- 400 richieste al secondo a GetMP4MediaFragment e 100 richieste al secondo a GetTSFragment.

Queste quote hanno un'importante implicazione per quanto riguarda il numero di HLS e di MPEG-DASH sessioni che possono essere supportate per stream. Non c'è limite al numero di sessioni HLS e DASH che possono essere utilizzate dai lettori multimediali in un determinato momento. Pertanto, è importante che l'applicazione di riproduzione non consenta di utilizzare troppe sessioni contemporaneamente. I due esempi seguenti descrivono come determinare il numero di sessioni di riproduzione simultanee che possono essere supportate:

Esempio 1: streaming live

In uno scenario di live streaming con HLS con frammenti della durata di 1 secondo, una traccia audio e video e `MaxMediaPlaylistFragmentResults` impostato su cinque, un lettore multimediale effettua in genere due chiamate al `GetHLSMediaPlaylist` secondo. Una chiamata è per i metadati video più recenti e un'altra per i metadati audio corrispondenti. Le due chiamate utilizzano ciascuna cinque punti di quota di metadati frammentari. Inoltre, effettua due chiamate al secondo: una `GetMP4MediaFragment` per il video più recente e l'altra per l'audio corrispondente. Ogni chiamata utilizza un token multimediale a frammento singolo, quindi vengono consumati due token in totale.

In questo scenario, possono essere supportate fino a 250 sessioni di riproduzione simultanee. Con 250 sessioni, questo scenario consuma 2.500 punti di quota di metadati di frammenti al secondo (ben al di sotto della quota di 10.000) e 500 punti di quota multimediale di frammenti al secondo.

Esempio 2: riproduzione On-demand

In uno scenario di riproduzione su richiesta di un evento passato con MPEG-DASH una traccia audio e video `MaxManifestFragmentResults` impostata su 1.000, un lettore multimediale in genere chiama una `GetDASHManifest` volta all'inizio della sessione (consumando 1.000 punti di quota di metadati di frammenti) e chiama `GetMP4MediaFragment` a una velocità massima di 5 volte al secondo (consumando 5 punti di quota multimediale a frammenti) fino al caricamento di tutti i frammenti. In questo scenario, è possibile avviare fino a 10 nuove sessioni al secondo (proprio alla quota di 10.000 frammenti di metadati al secondo) e fino a 100 sessioni possono caricare attivamente file multimediali frammentati a una velocità di 5 al secondo (proprio alla quota di 500 frammenti di file multimediali al secondo).

È possibile utilizzare `ArchivedFragmentsConsumed.Metadata` e `ArchivedFragmentsConsumed.Media` monitorare rispettivamente il consumo di metadati di frammenti e punti di quota dei frammenti multimediali. Per informazioni sul monitoraggio, vedere [the section called "Monitoraggio"](#)

Quote dei servizi di metadati di streaming

Le seguenti quote di servizio si applicano all'aggiunta di metadati di streaming a un flusso video Kinesis:

- Puoi anteporre a un frammento fino a 10 elementi metadati.
- La lunghezza del nome dei metadati del frammento può essere al massimo di 128 byte.
- La lunghezza del valore dei metadati del frammento può essere al massimo di 256 byte.
- Il nome di metadati di un frammento non può iniziare con la stringa "». AWS Se un elemento metadati viene aggiunto, il metodo `putFragmentMetadata` nel PIC restituisce un errore

STATUS_INVALID_METADATA_NAME (codice errore 0x52000077). La tua applicazione può ignorare l'errore (il PIC non aggiunge l'elemento metadati) oppure correggere l'errore.

Quote SDK di Producer

La tabella seguente contiene le quote correnti per i valori nell'SDK. Per ulteriori informazioni, consulta [Caricamento su Kinesis Video Streams](#).

Note

Prima di impostare questi valori, è necessario convalidare gli input. L'SDK non convalida questi limiti e si verifica un errore di runtime se vengono superati i limiti.

Valore	Limite	Note
Numero massimo di flussi	128	Il numero massimo di flussi che un oggetto producer può creare. Si tratta di un limite flessibile, per il quale puoi richiedere un aumento. Garantisce che il produttore e non crei accidentalmente stream in modo ricorsivo.
Lunghezza massima del nome del dispositivo	128 caratteri	
Numero massimo di tag	50 per flusso	
Lunghezza massima del nome del flusso	256 caratteri	
Dimensione minima di storage	10 MiB = 10 * 1024 * 1024 byte	
Dimensione massima di storage	10 GiB = 10 * 1024 * 1024 * 1024 byte	

Valore	Limite	Note
Lunghezza massima del percorso della directory root	4,096 caratteri	
Lunghezza massima delle informazioni di autorizzazione	10,000 byte	
Lunghezza massima della stringa URI	10,000 caratteri	
Lunghezza massima del nome del tag	128 caratteri	
Lunghezza massima del valore del tag	1,024 caratteri	
Periodo minimo del token di sicurezza	30 secondi	
Periodo di tolleranza del token di sicurezza	40 minuti	Se la durata specificata è più lunga, è limitata a questo valore.
Periodo di conservazione	0 o superiore a un'ora	Il valore 0 indica nessuna conservazione.
Durata minima del cluster	1 secondo	Il valore è specificato in unità di 100 ns, ovvero lo standard dell'SDK.
Durata massima del cluster	30 secondi	Il valore è specificato in unità di 100 ns, ovvero lo standard dell'SDK. L'API di backend può imporre una durata del cluster più breve.

Valore	Limite	Note
Dimensione massima del frammento	50 MB	Per ulteriori informazioni, consulta Quote del servizio Amazon Kinesis Video Streams .
Durata massima del frammento	20 secondi	Per ulteriori informazioni, consulta Quote del servizio Amazon Kinesis Video Streams .
Durata massima della connessione	45 minuti	Il back-end chiude la connessione dopo questo intervallo di tempo. L'SDK ruota il token e stabilisce una nuova connessione entro questo intervallo di tempo.
Lunghezza massima del segmento ACK	1,024 caratteri	Lunghezza massima del segmento di conferma inviato alla funzione del parser di ACK.
Lunghezza massima della stringa del tipo di contenuto	128 caratteri	
Lunghezza massima della stringa dell'ID codec	32 caratteri	
Lunghezza massima della stringa del nome della traccia	32 caratteri	
Lunghezza massima dei dati privati del codec	1 MiB = 1 * 1024 * 1024 byte	

Valore	Limite	Note
Lunghezza minima del valore di scala del timecode	100 ns	Il valore di scala minimo del timecode per rappresentare i timestamp del fotogramma nel cluster MKV risultante. Il valore è specificato in incrementi di 100 ns, ovvero lo standard dell'SDK.
Lunghezza massima del valore di scala del timecode	1 secondo	Il valore di scala massimo del timecode per rappresentare i timestamp del fotogramma nel cluster MKV risultante. Il valore è specificato in incrementi di 100 ns, ovvero lo standard dell'SDK.
Numero minimo degli elementi di visualizzazione dei contenuti	10	
Durata minima del buffer	20 secondi	Il valore è specificato in incrementi di 100 ns, ovvero lo standard dell'SDK.
Lunghezza massima della versione di aggiornamento	128 caratteri	
Lunghezza massima dell'ARN	1.024 caratteri	
Lunghezza massima della sequenza dei frammenti	128 caratteri	
Periodo massimo di conservazione	10 anni	

Amazon Kinesis Video Streams Storage su più livelli

Lo storage su più livelli di Amazon Kinesis Video Streams consente di ottimizzare i costi associati all'archiviazione e al consumo di contenuti multimediali in base alle esigenze di consumo previste, alla frequenza di accesso ai dati e alla durata di conservazione. KVS offre due livelli di storage per aiutarti a bilanciare costi e requisiti prestazionali.

Argomenti

- [Panoramica di](#)
- [Concetti chiave](#)
- [Configurazione dello storage su più livelli](#)
- [Best practice](#)

Panoramica di

Kinesis Video Streams offre 2 livelli di archiviazione:

- Kinesis Video Streams Standard (Hot) Tier: fornisce un accesso immediato con la latenza più bassa per i contenuti multimediali a cui si accede di frequente.
- Kinesis Video Streams Warm Tier: Cost-effective per i casi d'uso in cui i contenuti multimediali vengono archiviati per un periodo più lungo e vi si accede meno frequentemente, ma richiedono un accesso rapido quando necessario.

Concetti chiave

La comprensione di questi concetti ti aiuterà a configurare politiche di archiviazione ottimali:

- Costo dello storage: il livello Warm è più economico del livello Hot.
- Costo di acquisizione: il livello Warm viene addebitato per 1000 frammenti mentre il livello Hot viene addebitato per 1 GB di file multimediali ingeriti.
- Durata minima di archiviazione: lo storage Warm tier viene addebitato per una durata minima di archiviazione di 30 giorni. Se i file multimediali vengono eliminati o spostati prima di questo periodo di 30 giorni, ti verrà addebitato il normale costo di utilizzo dello spazio di archiviazione più un addebito proporzionale per il resto della durata minima di archiviazione. I contenuti multimediali archiviati per un periodo superiore alla durata minima di archiviazione non comporteranno un costo minimo.

- Latenza: leggero aumento (millisecondi) della latenza di acquisizione e consumo quando i contenuti multimediali vengono archiviati nel livello Warm.

Configurazione dello storage su più livelli

Crea un nuovo stream con Warm Tier Ingestion

1. Apri la console all'indirizzo. <https://console.aws.amazon.com/kinesisvideo/home>
2. Nella pagina Video streams (Flussi video) scegliere Create video stream (Crea flusso video).
3. Nella pagina Crea un nuovo streaming video, inserisci *YourStreamName* il nome dello stream.
4. Scegli Configurazione personalizzata. Imposta una conservazione dei dati (ad es.). *31 days*
Imposta il livello di archiviazione come. *Warm tier*
5. Scegliere Create video stream (Crea flusso video).
6. Dopo che Amazon Kinesis Video Streams ha creato lo stream, consulta i dettagli nella *YourStreamName* pagina.

Il tuo stream ora inserirà i contenuti multimediali direttamente nel Warm Tier, fornendo un accesso immediato con la latenza più bassa.

Best practice

- Analizza i modelli di accesso ai file multimediali prima di selezionare i livelli di archiviazione
- Usa Hot tier per i contenuti multimediali ad accesso frequente che richiedono un accesso immediato
- Configura il livello Warm per i contenuti multimediali a cui si accede occasionalmente con una leggera latenza accettabile
- Considerate il costo totale di proprietà, inclusi i costi di archiviazione e recupero

Configurazione di un account

Prima di utilizzare Amazon Kinesis Video Streams per la prima volta, completa le seguenti attività.

Argomenti

- [Registrati per un Account AWS](#)
- [Crea un Account AWS Chiave](#)

Registrati per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedere Guida [introduttiva a Account AWS](#) nella Guida Gestione dell'account AWS di riferimento.

Crea un Account AWS Chiave

Avrai bisogno di una Account AWS chiave per accedere ad Amazon Kinesis Video Streams a livello di programmazione.

Per creare una Account AWS chiave, procedi come segue:

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Scegli Utenti nella barra di navigazione e scegli l'utente Amministratore.
3. Seleziona la scheda Security credentials (Credenziali di sicurezza), quindi Create access key (Crea chiave di accesso).
4. Registra l'Access key ID (ID chiave di accesso) Scegli Show (Mostra) in Secret access key (Chiave di accesso segreta). Registra la Secret access key (Chiave di accesso segreta).

Vantaggi

I vantaggi dell'utilizzo di Kinesis Video Streams includono:

- Connessione e streaming da milioni di dispositivi: puoi usare Kinesis Video Streams per connettere e trasmettere video, audio e altri dati da milioni di dispositivi, tra cui smartphone di consumo, droni e dash cam. Puoi utilizzare le librerie di produzione di Kinesis Video Streams per configurare i tuoi dispositivi e trasmettere in streaming in modo affidabile in tempo reale o come upload multimediale a posteriori.
- Archivia, crittografa e indicizza i dati in modo duraturo: puoi configurare il tuo streaming video Kinesis per archiviare in modo duraturo i dati multimediali per periodi di conservazione personalizzati. Kinesis Video Streams genera anche un indice dei dati archiviati in base ai timestamp generati dal produttore o dal servizio. Le applicazioni possono recuperare dati specifici in uno stream utilizzando il time-index.
- Concentrati sulla gestione delle applicazioni anziché sull'infrastruttura: Kinesis Video Streams è serverless, quindi non c'è alcuna infrastruttura da configurare o gestire. Non devi preoccuparti

dell'implementazione, della configurazione o della scalabilità elastica dell'infrastruttura sottostante perché i flussi di dati e il numero di applicazioni utilizzate crescono e si riducono. Kinesis Video Streams esegue automaticamente tutta l'amministrazione e la manutenzione necessarie per gestire gli stream, così puoi concentrarti sulle applicazioni, non sull'infrastruttura.

- Crea applicazioni in tempo reale e in batch su flussi di dati: puoi usare Kinesis Video Streams per creare applicazioni personalizzate in tempo reale che operano su flussi di dati in tempo reale e creare applicazioni batch o monouso che operano su dati persistenti in modo duraturo senza rigidi requisiti di latenza. Puoi creare, distribuire e gestire applicazioni personalizzate: soluzioni open source (Apache MXNet, OpenCV), sviluppate in casa o di terze parti utilizzando Marketplace AWS per elaborare e analizzare i tuoi stream. È possibile utilizzare le Get API Kinesis Video Streams per creare più applicazioni simultanee che elaborano dati in tempo reale o orientati ai batch.
- Trasmetti i dati in modo più sicuro: Kinesis Video Streams crittografa tutti i dati mentre fluiscono nel servizio e quando persistono. Kinesis Video Streams applica la crittografia basata su Transport Layer Security (TLS) sullo streaming di dati dai dispositivi e crittografa tutti i dati inattivi utilizzando (). AWS Key Management Service AWS KMS Inoltre, puoi gestire l'accesso ai tuoi dati utilizzando (IAM). AWS Identity and Access Management
- Paga in base al consumo: per ulteriori informazioni, consulta [Calcolatore dei prezzi AWS](#).

Guida introduttiva ad Amazon Kinesis Video Streams

Questa sezione descrive come eseguire le seguenti attività in Amazon Kinesis Video Streams:

- Configura Account AWS e crea un amministratore, se non l'hai già fatto.
- Crea un flusso video Kinesis.
- Inviare i dati allo streaming video Kinesis dalla videocamera e visualizzare i file multimediali nella console.

Se non conosci Amazon Kinesis Video Streams, ti consigliamo di leggere prima. [Kinesis Video Streams: come funziona](#)

Note

Seguire l'esempio Guida introduttiva non comporterà alcun addebito a tuo carico. Account AWS Per i costi dei dati nella tua regione, consulta i prezzi di [Amazon Kinesis Video Streams](#).

Argomenti

- [Crea uno streaming video Amazon Kinesis](#)
- [Invia dati a uno streaming video di Amazon Kinesis](#)
- [Consuma dati multimediali](#)

Crea uno streaming video Amazon Kinesis

Questa sezione descrive come creare uno streaming video Kinesis.

Questa sezione contiene le procedure seguenti:

- [the section called “Crea uno streaming video usando la console”](#)
- [the section called “Crea uno streaming video utilizzando AWS CLI”](#)

Crea uno streaming video usando la console

1. Apri la console all'indirizzo <https://console.aws.amazon.com/kinesisvideo/home>.
2. Nella pagina Video streams (Flussi video) scegliere Create video stream (Crea flusso video).
3. Nella pagina Crea un nuovo streaming video, inserisci *YourStreamName* il nome dello stream. Lascia selezionato il pulsante di configurazione predefinita.
4. Scegliere Create video stream (Crea flusso video).
5. Dopo che Amazon Kinesis Video Streams ha creato lo stream, consulta i dettagli nella *YourStreamName* pagina.

Crea uno streaming video utilizzando AWS CLI

1. Verifica di aver AWS CLI installato e configurato. Per ulteriori informazioni, consulta la documentazione [AWS Command Line Interface](#).
2. Esegui il seguente comando Create-Stream nella AWS CLI:

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

La risposta sarà simile alla seguente:

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

Invia dati a uno streaming video di Amazon Kinesis

Questa sezione descrive come inviare dati multimediali da una videocamera al flusso video Kinesis creato nella sezione precedente. Questa sezione usa il [Usa la libreria di produzione C++](#) come [Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink](#) plugin.

Per inviare contenuti multimediali da una varietà di dispositivi su diversi sistemi operativi, questo tutorial utilizza la libreria di produzione C++ Kinesis Video Streams e [GStreamer](#), un framework multimediale open source che standardizza l'accesso alle telecamere e ad altre fonti multimediali.

Argomenti

- [Crea l'SDK e gli esempi](#)
- [Esegui gli esempi per caricare contenuti multimediali su Kinesis Video Streams](#)
- [Rivedi gli oggetti di riconoscimento](#)

Crea l'SDK e gli esempi

Puoi creare l'SDK e gli esempi sul tuo computer o in AWS Cloud9. Segui le procedure appropriate riportate di seguito.

Build on your computer

Utilizzate le istruzioni nel file [readme](#) per creare la libreria di produzione e l'applicazione di esempio.

Questo include:

- Installazione delle dipendenze
- Clonazione del repository
- Usare CMake per generare makefile
- Creazione dei file binari usando make

Build in AWS Cloud9

Segui queste procedure per caricarli su Kinesis Video Streams in AWS Cloud9. Non dovrai scaricare nulla sul tuo computer.

1. In Console di gestione AWS, apri [AWS Cloud9](#).

Seleziona Crea ambiente.

2. Nella schermata Crea ambiente, completa quanto segue:

- Nome: digita un nome per il nuovo ambiente.
- Piattaforma: seleziona Ubuntu Server 22.04 LTS.

Puoi lasciare gli altri campi con le selezioni predefinite.

3. Una volta creato l'ambiente, seleziona Apri nella colonna IDE Cloud9.

Nella parte medio-bassa dello schermo, vedi. Admin:~/environment \$ Questo è il terminale AWS Cloud9 (Amazon EC2).

 Note

Se chiudi accidentalmente il terminale, seleziona Finestra, Nuovo terminale.

Esegui i seguenti comandi nel terminale per modificare il volume a 20 GiB.

- a. Scarica lo script .

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

- b. Assegna le autorizzazioni di esecuzione dello script.

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

- c. Eseguire lo script.

```
./resize_volume.sh
```

4. Recupera le informazioni più recenti su tutto il software che puoi installare o aggiornare tramite l'Advanced Packaging Tool (APT).

Questo comando non aggiorna il software stesso, ma assicura che il sistema sappia quali sono le ultime versioni disponibili.

```
sudo apt-get update
```

5. Installa le dipendenze del produttore SDK di C++.

```
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. Usa git per clonare il C++ producer SDK.

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

7. Prepara una cartella di compilazione.

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

8. Usa CMake per generare makefile.

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

La fine dell'output previsto è la seguente:

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

9. Usa make per compilare l'SDK e le applicazioni di esempio, oltre a creare gli eseguibili finali.

```
make
```

La fine dell'output previsto è la seguente:

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

10. Conferma che i file di esempio sono stati creati. Elenca i file nella directory corrente:

```
ls
```

Conferma che siano presenti i seguenti file:

- kvs_gstreamer_sample
- libgstkvssink.so

11. (Facoltativo) Puoi aggiungere l'impostazione della variabile di ambiente GST_PLUGIN_PATH allo script di avvio della shell. Ciò garantisce che GST_PLUGIN_PATH sia impostato

correttamente durante una nuova sessione di terminale. In AWS Cloud9, lo script di avvio della shell è: `~/ .bashrc`

Esegui il comando seguente per aggiungere il comando alla fine dello script di avvio della shell.

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

Digitate quanto segue per eseguire lo script di avvio della shell:

```
source ~/.bashrc
```

Conferma che `GST_PLUGIN_PATH` sia impostato.

```
echo $GST_PLUGIN_PATH
```

Se imposti l'output correttamente, vedrai il seguente output. Se l'output è vuoto, la variabile di ambiente non è impostata correttamente.

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

Eseguite gli esempi per caricare contenuti multimediali su Kinesis Video Streams

L'applicazione di esempio non supporta le credenziali IMDS. Nel tuo terminale, esporta AWS le credenziali per il tuo utente o ruolo IAM e la regione in cui si trova il tuo stream.

```
export AWS_ACCESS_KEY_ID=YourAccessKey  
export AWS_SECRET_ACCESS_KEY=YourSecretKey  
export AWS_DEFAULT_REGION=YourAWSRegion
```

Se utilizzi AWS credenziali temporanee, esporta anche il token di sessione:

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Scarica un video .mp4 di esempio da caricare su Kinesis Video Streams.

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

Specifiche video:

- Risoluzione: 1280 x 720 pixel
- Frequenza fotogrammi: 30 fotogrammi al secondo
- Durata: 14,0 secondi
- Codifica video - H.264, nella traccia 1
- Fotogrammi chiave: ogni 3 secondi, con una durata del frammento (nota anche come dimensione di un gruppo di immagini (GoP)) di 3 secondi, con il frammento finale di 2 secondi.

Esegui il comando seguente con il nome dello stream che hai creato in precedenza. Se non hai ancora creato uno stream, consulta [the section called “Crea uno streaming video Amazon Kinesis”](#).

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

Sample video from GStreamer

Usa il seguente comando per generare un video usando GStreamer.

Indica a GStreamer dove localizzare il plugin GStreamer. kvssink Nella tua cartella di compilazione, specifica il percorso della cartella contenente il file. libgstkvssink.so

Dalla cartella di compilazione, esegui il comando seguente:

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

Questa pipeline GStreamer genera un flusso video di test live con uno schema di test standard che viene eseguito a 10 fotogrammi al secondo con una risoluzione di 640x480 pixel. Viene aggiunto un overlay che mostra l'ora e la data correnti del sistema. Il video viene quindi codificato in H.264 formato e i fotogrammi chiave vengono generati al massimo ogni 10 fotogrammi, con una durata del frammento (nota anche come dimensione di un gruppo di immagini (GoP)) di 1

secondo. `kvssink` prende il flusso H.264-encoded video, lo impacchetta nel formato contenitore Matroska (MKV) e lo carica nel flusso video Kinesis.

Esegui il comando seguente:

```
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"
```

Per interrompere la pipeline GStreamer, selezionate la finestra del terminale e premete CTRL+C.

Note

Per ulteriori informazioni sull'utilizzo del plugin GStreamer per lo streaming di video da un flusso RTSP da una videocamera o da una videocamera USB, consulta. [Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink](#)

Rivedi gli oggetti di riconoscimento

Durante il caricamento, Kinesis Video Streams invierà gli oggetti di riconoscimento al client che esegue il caricamento. Dovresti vederli stampati nell'output del comando. Un esempio è simile al seguente:

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

Se la conferma `EventType` è `PERSISTED`, significa che Kinesis Video Streams ha archiviato e crittografato in modo duraturo questo blocco di contenuti multimediali per il recupero, l'analisi e l'archiviazione a lungo termine.

Per ulteriori informazioni sui riconoscimenti, vedere. [PutMedia](#)

Consuma dati multimediali

Puoi consumare dati multimediali visualizzandoli nella console o creando un'applicazione che legge i dati multimediali da uno stream utilizzando Hypertext Live Streaming (HLS).

Visualizza i contenuti multimediali nella console

In un'altra scheda del browser, apri il Console di gestione AWS. Nella dashboard di Kinesis Video Streams, seleziona Streams [video](#).

Seleziona il nome del tuo stream nell'elenco degli stream. Usa la barra di ricerca, se necessario.

Espandi la sezione Riproduzione multimediale. Se il video è ancora in fase di caricamento, verrà mostrato. Se il caricamento è terminato, seleziona la doppia freccia sinistra.

Consuma dati multimediali utilizzando HLS

È possibile creare un'applicazione client che utilizza dati da un flusso video Kinesis utilizzando HLS. Per ulteriori informazioni su come creare un'applicazione che consuma i dati multimediali utilizzando HLS, consulta [Riproduzione video](#).

Caricamento su Kinesis Video Streams

Le librerie per produttori di Amazon Kinesis Video Streams sono un insieme di librerie nell'SDK per produttori di Kinesis Video Streams. Il client utilizza le librerie e l'SDK per creare l'applicazione sul dispositivo per la connessione sicura a Kinesis Video Streams e lo streaming di dati multimediali da visualizzare nella console o nelle applicazioni client in tempo reale.

I dati multimediali possono essere diffusi nei seguenti modi:

- In tempo reale
- Dopo averlo bufferizzato per alcuni secondi
- Dopo il caricamento dei file multimediali

Dopo aver creato uno stream Kinesis Video Streams, puoi iniziare a inviargli dati. Puoi utilizzare l'SDK per creare codice applicativo che estrae i dati video, noti come frame, dalla sorgente multimediale e li carica su Kinesis Video Streams. Queste applicazioni vengono anche definite applicazioni producer.

Le librerie Producer contengono i seguenti componenti:

- [Cliente produttore di Kinesis Video Streams](#)
- [Libreria per produttori di Kinesis Video Streams](#)

Cliente produttore di Kinesis Video Streams

Il client di produzione Kinesis Video Streams include `KinesisVideoClient` un'unica classe. Questa classe gestisce le sorgenti multimediali, riceve dati dalle fonti e gestisce il ciclo di vita dello stream mentre i dati fluiscono da una sorgente multimediale a Kinesis Video Streams. Fornisce inoltre un'`MediaSource` interfaccia per definire l'interazione tra Kinesis Video Streams e l'hardware e il software proprietari.

Un'origine multimediale può essere costituita da pressoché qualsiasi elemento. Ad esempio, è possibile utilizzare un'origine multimediale sotto forma di fotocamera o microfono. Le origini multimediali sorgenti non si limitano a sorgenti audio e video. Ad esempio, i log dei dati potrebbero essere file di testo, che tuttavia possono sempre essere inviati come flusso di dati. Puoi inoltre disporre di più telecamere sul tuo telefono che trasmettono in streaming i dati simultaneamente.

Per ottenere i dati da qualsiasi di queste origini, puoi implementare l'interfaccia `MediaSource`. Questa interfaccia consente ulteriori scenari per cui non forniamo il supporto integrato. Ad esempio, puoi scegliere di inviare quanto segue a Kinesis Video Streams:

- Un flusso di dati diagnostici (ad esempio, i log dell'applicazione e gli eventi)
- I dati provenienti da telecamere a infrarossi, RADAR o telecamere di profondità

Kinesis Video Streams non fornisce implementazioni integrate per dispositivi di produzione multimediale come le fotocamere. Per estrarre i dati provenienti da questi dispositivi, è necessario implementare il codice, creando la tua implementazione di origine multimediale personalizzate. È quindi possibile registrare in modo esplicito le sorgenti multimediali personalizzate con `KinesisVideoClient`, che carica i dati su Kinesis Video Streams.

Il client per produttori Kinesis Video Streams è disponibile per applicazioni Java e Android. Per ulteriori informazioni, consultare [Usa la libreria del produttore Java](#) e [Usa la libreria del produttore Android](#).

Libreria per produttori di Kinesis Video Streams

La libreria per produttori di Kinesis Video Streams è contenuta nel client per produttori Kinesis Video Streams. La libreria è disponibile anche per l'uso diretto per coloro che desiderano un'integrazione più profonda con Kinesis Video Streams. Ciò consente l'integrazione da dispositivi con sistemi operativi proprietari, stack di rete oppure risorse sul dispositivo limitate.

La libreria per produttori Kinesis Video Streams implementa la macchina a stati per lo streaming su Kinesis Video Streams. Offre hook di callback, che ti richiedono di fornire la tua implementazione di trasporto e di gestire in modo esplicito ogni messaggio da e verso il servizio.

Potresti scegliere di utilizzare direttamente la libreria per produttori di Kinesis Video Streams per i seguenti motivi:

- Il dispositivo in cui vuoi eseguire l'applicazione non ha una macchina virtuale Java.
- Vuoi scrivere il codice dell'applicazione in altri linguaggi diversi da Java.
- Desiderate ridurre il sovraccarico del codice e limitarlo al minimo livello di astrazione, a causa di limitazioni come la memoria e la potenza di elaborazione.

Attualmente, la libreria per produttori Kinesis Video Streams è disponibile per applicazioni Android, C, C++ e Java. Per ulteriori informazioni, consulta le lingue supportate nei seguenti Argomenti correlati.

Scopri cosa sono le librerie per produttori

[Usa la libreria del produttore Java](#)

[Usa la libreria del produttore Android](#)

[Usa la libreria di produzione C++](#)

[Usa la libreria C producer](#)

[Usa l'SDK per produttori C++ su Raspberry Pi](#)

Usa la libreria del produttore Java

Puoi utilizzare la libreria di produzione Java fornita da Amazon Kinesis Video Streams per scrivere codice applicativo con una configurazione minima e inviare dati multimediali da un dispositivo a un flusso video Kinesis.

Esegui i passaggi seguenti per integrare il codice con Kinesis Video Streams in modo che l'applicazione possa avviare lo streaming di dati verso il tuo streaming video Kinesis:

1. Crea un'istanza dell'oggetto `KinesisVideoClient`.
2. Crea un oggetto `MediaSource` fornendo informazioni sull'origine multimediale. Ad esempio, durante la creazione di un'origine multimediale fotocamera, devi fornire informazioni quali l'identificazione della fotocamera e specificare la codifica utilizzata dalla fotocamera.

Quando vuoi avviare lo streaming, devi creare un'origine multimediale personalizzata.

3. Registra l'origine multimediale con `KinesisVideoClient`.

Dopo che hai registrato l'origine multimediale con `KinesisVideoClient`, ogni volta che i dati diventano disponibili con tale origine, viene chiamato `KinesisVideoClient` con i dati.

Procedura: utilizzate il Java producer SDK

Questa procedura dimostra come utilizzare il client di produzione Java Kinesis Video Streams nell'applicazione Java per inviare dati al flusso video Kinesis.

Per questa procedura non è necessario disporre di un'origine multimediale, come una videocamera o un microfono. Al contrario, a scopo di test, il codice genera frame di esempio composti da una serie di byte. Puoi utilizzare lo stesso modello di codifica quando invii dati multimediali da origini reali, come videocamere e microfoni.

La procedura include le seguenti fasi:

- [Scarica e configura il codice](#)
- [Scrivi ed esamina il codice](#)
- [Esegui e verifica il codice](#)

Prerequisiti

Prima di configurare Java producer SDK, assicurati di avere i seguenti prerequisiti:

- Nel codice di esempio, fornisci le credenziali specificando un profilo che hai impostato nel file del profilo delle credenziali AWS . Se non lo hai già fatto, configura prima il profilo delle credenziali. Per ulteriori informazioni, consulta [Configurare AWS le credenziali e l'area per lo sviluppo in](#). AWS SDK per Java

Note

L'esempio Java utilizza un `SystemPropertiesCredentialsProvider` oggetto per ottenere le credenziali. Il provider recupera queste credenziali dalle proprietà del sistema Java `aws.accessKeyId` e `aws.secretKey`. Puoi impostare queste proprietà nel tuo ambiente di sviluppo Java. Per informazioni su come impostare le proprietà del sistema Java, consulta la documentazione per il tuo ambiente di sviluppo integrato (IDE) specifico.

- `NativeLibraryPathDevi` contenere il tuo `KinesisVideoProducerJNI` file, disponibile all'indirizzo <https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp>. L'estensione del nome di questo file varia a seconda del sistema operativo:
 - `KinesisVideoProducerJNI.so` per Linux
 - `KinesisVideoProducerJNI.dylib` per macOS
 - `KinesisVideoProducerJNI.dll` per Windows

 Note

Pre-built le librerie per macOS, Ubuntu, Windows e Raspbian sono disponibili all'`src/main/resources/lib` indirizzo. <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java> Per altri ambienti, compila [C++](#).

Scarica e configura il codice della libreria del produttore Java

In questa sezione della procedura della libreria Java producer, scaricate il codice di esempio Java, importate il progetto nel vostro IDE Java e configurate le posizioni delle librerie.

Per i prerequisiti e altri dettagli su questo esempio, vedete [Utilizzo della libreria](#) Java producer.

1. Create una directory, quindi clonate il codice sorgente di esempio dal GitHub repository.

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

2. Apri l'ambiente di sviluppo integrato (IDE) Java che usi (ad esempio, [Eclipse](#) o [JetBrains IntelliJ IDEA](#)) e importa il progetto Apache Maven che hai scaricato:
 - In IntelliJ IDEA: scegli Import (Importa). Andare al file `pom.xml` nella radice del pacchetto scaricato.
 - In Eclipse: scegliere File, Import (Importa), Maven, Existing Maven Projects (Progetti Maven esistenti). Passa quindi alla directory `kinesis-video-java-demo`.

Per ulteriori informazioni, consulta la documentazione relativa al tuo IDE.

3. Il codice di esempio Java utilizza le credenziali correnti. AWS Per usare un profilo di credenziali differenti, individua il seguente codice in `DemoAppMain.java`:

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Modifica il codice come indicato di seguito:

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        new ProfileCredentialsProvider("credentials-profile-name"));
```

Per ulteriori informazioni, consulta [ProfileCredentialsProvider](#) nella referenza per AWS SDK per Java.

Scrivi ed esamina il codice

In questa sezione della procedura della libreria [Java producer](#), scrivi ed esamini il codice di esempio Java scaricato nella sezione precedente.

L'applicazione per i test di Java ([DemoAppMain](#)) mostra il seguente modello di codifica:

- Creare un'istanza di `KinesisVideoClient`.
- Creare un'istanza di `MediaSource`.
- Registra `MediaSource` con il client.
- Avvia lo streaming. Avvia `MediaSource` e inizia a inviare dati al client.

Nelle seguenti sezioni sono fornite maggiori informazioni.

Crea un'istanza di `KinesisVideoClient`

Per creare l'oggetto `KinesisVideoClient`, richiama l'operazione `createKinesisVideoClient`.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Per eseguire chiamate alla rete, `KinesisVideoClient` ha bisogno di credenziali da autenticare. Puoi trasferire un'istanza di `SystemPropertiesCredentialsProvider`, che legge `AWSCredentials` per il profilo predefinito nel file di credenziali:

```
[default]
```

```
aws_access_key_id = ABCDEFGHIJKLMOPQRSTU  
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

Crea un'istanza di MediaSource

Per inviare byte al tuo flusso video Kinesis, devi produrre i dati. Amazon Kinesis Video Streams fornisce l'`MediaSource` interfaccia, che rappresenta l'origine dei dati.

Ad esempio, la libreria Java Kinesis Video Streams fornisce l'`ImageFileMediaSource` implementazione dell'interfaccia. `MediaSource` Questa classe legge solo i dati da una serie di file multimediali anziché da un flusso video Kinesis, ma è possibile utilizzarla per testare il codice.

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

Registrali MediaSource con il client

Registra l'origine multimediale creata con `KinesisVideoClient` in modo che sia a conoscenza dell'esistenza del client e possa quindi inviargli i dati.

```
kinesisVideoClient.registerMediaSource(mediaSource);
```

Avvia la fonte multimediale

Avviate la fonte multimediale in modo che possa iniziare a generare dati e inviarli al client.

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

Eseguire la pulizia delle risorse

Per evitare perdite di memoria, procedi come segue per annullare la registrazione di una fonte multimediale dal client e liberare il client.

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

```
    throw new RuntimeException(e);  
}
```

Se hai aggiunto degli elementi alla cache utilizzando [CachedInfoMultiAuthServiceCallbacks](#), ad esempio:

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

Svuota la cache quando hai finito:

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

Esegui e verifica il codice

Per eseguire il test harness Java per la libreria [Java Producer](#), effettuate le seguenti operazioni.

1. Scegli DemoAppMain.
2. Scegliete Esegui, Esegui 'DemoAppMain'.
3. Aggiungi le tue credenziali per gli argomenti JVM per l'applicazione:
 - Per AWS credenziali non temporanee: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
 - Per le credenziali temporanee AWS: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"
4. Accedi Console di gestione AWS e apri la console [Kinesis Video Streams](#).

Nella pagina Manage Streams (Gestisci flussi), scegli il tuo flusso.

5. Il video di esempio viene riprodotto nel lettore integrato. La visualizzazione del video potrebbe richiedere una breve attesa (fino a 10 secondi con la larghezza di banda e le condizioni del processore abituali) affinché vengano caricati i fotogrammi.

L'esempio di codice crea un flusso. Appena MediaSource nel codice si avvia, inizia a inviare frame di esempio a KinesisVideoClient. Il client invia quindi i dati al tuo streaming video Kinesis.

Usa la libreria del produttore Android

Puoi utilizzare la libreria per produttori Android fornita da Amazon Kinesis Video Streams per scrivere codice applicativo, con una configurazione minima, per inviare dati multimediali da un dispositivo Android a un flusso video Kinesis.

Esegui i passaggi seguenti per integrare il codice con Kinesis Video Streams in modo che l'applicazione possa avviare lo streaming di dati nel tuo streaming video Kinesis:

1. Crea un'istanza dell'oggetto `KinesisVideoClient`.
2. Crea un oggetto `MediaSource` fornendo informazioni sull'origine multimediale. Ad esempio, durante la creazione di un'origine multimediale fotocamera, devi fornire informazioni quali l'identificazione della fotocamera e specificare la codifica utilizzata dalla fotocamera.

Quando vuoi avviare lo streaming, devi creare un'origine multimediale personalizzata.

Procedura: utilizza l'Android producer SDK

Questa procedura dimostra come utilizzare il client di produzione Android Kinesis Video Streams nell'applicazione Android per inviare dati al flusso video Kinesis.

La procedura include le seguenti fasi:

- [the section called “Prerequisiti”](#)
- [the section called “Scarica e configura il codice”](#)
- [the section called “Esamina il codice”](#)
- [the section called “Esegui e verifica il codice”](#)

Prerequisiti

Ti consigliamo di utilizzare [Android Studio](#) per esaminare, modificare ed eseguire il codice dell'applicazione. Si consiglia di utilizzare l'ultima versione stabile.

Nel codice di esempio, fornisci le credenziali di Amazon Cognito.

Segui queste procedure per configurare un pool di utenti e un pool di identità di Amazon Cognito.

- [Configura un pool di utenti](#)
- [Configura un pool di identità](#)

Configura un pool di utenti

Per configurare un pool di utenti

1. Accedi alla console [Amazon Cognito](#) e verifica che la regione sia corretta.
2. Nella barra di navigazione a sinistra, scegli Pool di utenti.
3. Nella sezione Pool di utenti, scegli Crea pool di utenti.
4. Completa le seguenti sezioni:
 - a. Passaggio 1: Configurazione dell'esperienza di accesso: nella sezione delle opzioni di accesso del pool di utenti Cognito, seleziona le opzioni appropriate.

Seleziona Avanti.
 - b. Passaggio 2: Configurare i requisiti di sicurezza: selezionare le opzioni appropriate.

Seleziona Avanti.
 - c. Passaggio 3: Configura l'esperienza di registrazione: seleziona le opzioni appropriate.

Seleziona Avanti.
 - d. Passaggio 4: Configurare la consegna dei messaggi: selezionare le opzioni appropriate.

Nel campo di selezione del ruolo IAM, seleziona un ruolo esistente o crea un nuovo ruolo.

Seleziona Avanti.
 - e. Passaggio 5: integra la tua app: seleziona le opzioni appropriate.

Nel campo Client iniziale dell'app, scegli Client riservato.

Seleziona Avanti.
 - f. Passaggio 6: Rivedi e crea: rivedi le selezioni effettuate nelle sezioni precedenti, quindi scegli Crea pool di utenti.
5. Nella pagina Pool di utenti, seleziona il pool che hai appena creato.

Copia l'ID del pool di utenti e prendine nota per dopo. Nel `awsconfiguration.json` file, questo è `CognitoUserPool.Default.PoolId`.
6. Seleziona la scheda Integrazione dell'app e vai in fondo alla pagina.
7. Nella sezione Elenco dei client dell'app, scegli il nome del client dell'app che hai appena creato.

Copia l'ID cliente e prendine nota per dopo. Nel `awsconfiguration.json` file, questo è `CognitoUserPool.Default.AppClientId`.

8. Mostra il segreto del Cliente e prendine nota per dopo. Nel `awsconfiguration.json` file, questo è `CognitoUserPool.Default.AppClientSecret`.

Configura un pool di identità

Per configurare un pool di identità

1. Accedi alla console [Amazon Cognito](#) e verifica che la regione sia corretta.
2. Nella barra di navigazione a sinistra, scegli Identity pool.
3. Scegli Crea pool di identità.
4. Configura il pool di identità.
 - a. Fase 1: Configurare l'attendibilità del pool di identità - Completa le seguenti sezioni:
 - Accesso utente: seleziona Accesso autenticato
 - Fonti di identità autenticate: seleziona il pool di utenti di Amazon Cognito

Seleziona Avanti.

- b. Fase 2: Configurazione delle autorizzazioni - Nella sezione Ruolo autenticato, completa i seguenti campi:
 - Ruolo IAM: seleziona Crea un nuovo ruolo IAM
 - Nome del ruolo IAM: inserisci un nome e prendine nota per un passaggio successivo.

Seleziona Avanti.

- c. Fase 3: Connessione dei provider di identità - Nella sezione dei dettagli del pool di utenti, completa i seguenti campi:
 - ID del pool di utenti: seleziona il pool di utenti che hai creato in precedenza.
 - ID client dell'app: seleziona l'ID client dell'app che hai creato in precedenza.

Seleziona Avanti.

- d. Passaggio 4: Configurazione delle proprietà: digita un nome nel campo Nome del pool di identità.

Seleziona Avanti.

- e. Passaggio 5: Rivedi e crea: rivedi le selezioni in ciascuna sezione, quindi seleziona Crea pool di identità.

5. Nella pagina dei pool di identità, seleziona il tuo nuovo pool di identità.

Copia l'ID del pool di identità e prendine nota per dopo. Nel `awsconfiguration.json` file, questo è `CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Aggiorna le autorizzazioni per il ruolo IAM.

- a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
- b. Nella barra di navigazione a sinistra, scegli Ruoli.
- c. Trova e seleziona il ruolo che hai creato sopra.

 Note

Usa la barra di ricerca, se necessario.

- d. Seleziona la politica di autorizzazione allegata.

Seleziona Edit (Modifica).

- e. Seleziona la scheda JSON e sostituisci la policy con la seguente:

JSON

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

```
        " *"  
    }  
  }  
}
```

Seleziona Avanti.

- f. Seleziona la casella accanto a **Imposta questa nuova versione come predefinita** se non è già selezionata.

Seleziona **Salva modifiche**.

Scarica e configura il codice della libreria del produttore Android

In questa sezione della procedura della libreria di produttore Android, scarichi il codice di esempio Android e apri il progetto in Android Studio.

Per i prerequisiti e altri dettagli su questo esempio, vedi [Utilizzo della libreria](#) di produttori Android.

1. Crea una directory, quindi clonala AWS Mobile SDK per Android dal repository. GitHub

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

2. Apri [Android Studio](#).
3. Nella schermata iniziale, scegli **Apri un progetto di Android Studio esistente**.
4. Passa alla directory `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` e scegli **OK**.
5. Apri il file `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json`.

Nel `CredentialsProvider` nodo, fornisci l'ID del pool di identità indicato nella procedura [Per impostare un pool di identità nella sezione Prerequisiti](#) e fornisci il tuo Regione AWS (ad esempio, **us-west-2**).

Nel `CognitoUserPool` nodo, fornisci il segreto del client dell'app, l'ID del client dell'app e l'ID del pool dalla procedura [Per configurare un pool di utenti nella sezione Prerequisiti](#) e fornisci il tuo Regione AWS (ad esempio, **us-west-2**).

6. Il file `awsconfiguration.json` sarà simile a quanto riportato di seguito:

```
{
  "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. Aggiorna `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` con la tua regione (nell'esempio seguente, è impostata su `US_WEST_2`):

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

Per informazioni sulle Regione AWS costanti, consulta Regioni. <https://aws-amplify.github.io/aws-sdk-android/docs/reference/com/amazonaws/regions/Regions.html>

Esamina il codice

In questa sezione della procedura della libreria di produttori [Android](#), esamini il codice di esempio.

L'applicazione per i test di Android (AmazonKinesisVideoDemoApp) mostra il seguente modello di codifica:

- Creare un'istanza di `KinesisVideoClient`.
- Creare un'istanza di `MediaSource`.
- Avvia lo streaming. `MediaSource` avvia e inizia a inviare dati al client.

Nelle seguenti sezioni sono fornite maggiori informazioni.

Crea un'istanza di `KinesisVideoClient`

Per creare l'oggetto [KinesisVideoClient](#), richiama l'operazione [createKinesisVideoClient](#).

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(  
    getActivity(),  
    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,  
    KinesisVideoDemoApp.getCredentialsProvider());
```

Per eseguire chiamate alla rete, `KinesisVideoClient` ha bisogno di credenziali da autenticare. Inserisci un'istanza di `AWSCredentialsProvider`, che legge le tue credenziali Amazon Cognito dal `awsconfiguration.json` file che hai modificato nella sezione precedente.

Crea un'istanza di `MediaSource`

Per inviare byte al tuo flusso video Kinesis, devi produrre i dati. Amazon Kinesis Video Streams fornisce l'[MediaSource](#) interfaccia, che rappresenta l'origine dei dati.

Ad esempio, la libreria Android Kinesis Video Streams fornisce l'[AndroidCameraMediaSource](#) implementazione dell'interfaccia. `MediaSource` Questa classe legge i dati da una delle fotocamere del dispositivo.

Nel seguente esempio di codice (dal file [fragment/StreamConfigurationFragment.java](#)), viene creata la configurazione per l'origine multimediale:

```
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));
}

```

Nel seguente esempio di codice (dal file [fragment/StreamingFragment.java](#)), viene creata la configurazione per l'origine multimediale:

```

mCameraMediaSource = (AndroidCameraMediaSource) mKinesisVideoClient
    .createMediaSource(mStreamName, mConfiguration);

```

Avvia la fonte multimediale

Avvia l'origine multimediale in modo che inizi a generare dati e a inviarli al client. I seguenti esempi di codice sono tratti dal file [fragment/StreamingFragment.java](#):

```

mCameraMediaSource.start();

```

Esegui e verifica il codice

Per eseguire l'applicazione di esempio Android per la libreria di produttori [Android](#), procedi come segue.

1. Esegui la connessione a un dispositivo Android.
2. Scegli Run (Esegui), Run... (Esegui...), quindi scegli Edit configurations... (Modifica configurazioni...).

3. Scegli l'icona più (+), App per Android. Nel campo Name (Nome), inserire **AmazonKinesisVideoDemoApp**. Nel menu a tendina Modulo, scegli AmazonKinesisVideoDemoApp. Scegli OK.
4. Scegli Run (Esegui), Run (Esegui).
5. Nella schermata Select Deployment Target (Seleziona destinazione distribuzione) scegli il dispositivo connesso, quindi scegli OK.
6. Nell'AWSKinesisVideoDemoAppapplicazione sul dispositivo, scegli Crea nuovo account.
7. Inserisci i valori per USERNAME (NOME UTENTE), Password, Given name (Nome specificato), Email address (Indirizzo e-mail) e Phone number (Numero di telefono), quindi scegli Sign up (Iscrizione).

Note

Questi valori hanno le seguenti limitazioni:

- Password: deve contenere lettere maiuscole e minuscole, numeri e caratteri speciali. Puoi modificare questi vincoli nella pagina del pool di utenti sulla console [Amazon Cognito](#).
- Email address (Indirizzo e-mail): deve essere un indirizzo valido a cui ricevere un codice di conferma.
- Phone number (Numero di telefono): deve avere il seguente formato: **+<Country code><Number>**, ad esempio, **+12065551212**.

8. Inserisci il codice che ricevi via e-mail e scegli Conferma. Scegli OK.
9. Nella pagina successiva, mantieni i valori predefiniti e scegli Stream.
10. Accedi Console di gestione AWS e apri la console [Kinesis Video Streams](#) nella regione Stati Uniti occidentali (Oregon).

Nella pagina Manage Streams (Gestisci flussi), scegli demo-stream.

11. Il video streaming viene riprodotto nel lettore integrato. La visualizzazione del video potrebbe richiedere una breve attesa (fino a 10 secondi con la larghezza di banda e le condizioni del processore abituali) affinché vengano caricati i fotogrammi.

 Note

Se lo schermo del dispositivo ruota (ad esempio, da verticale a orizzontale), l'applicazione interrompe lo streaming del video.

L'esempio di codice crea un flusso. Appena `MediaSource` nel codice viene avviato, inizia a inviare fotogrammi dalla fotocamera a `KinesisVideoClient`. Il client invia quindi i dati a un flusso video Kinesis denominato `demo-stream`.

Usa la libreria di produzione C++

Puoi utilizzare la libreria di produzione C++ fornita da Amazon Kinesis Video Streams per scrivere codice applicativo per inviare dati multimediali da un dispositivo a un flusso video Kinesis.

Modello a oggetti

La libreria C++ fornisce i seguenti oggetti per gestire l'invio di dati a un flusso video Kinesis:

- `KinesisVideoProducer`: contiene informazioni sulla fonte multimediale e sulle AWS credenziali e mantiene i callback per segnalare gli eventi di Kinesis Video Streams.
- `KinesisVideoStream`: rappresenta il flusso video Kinesis. Contiene informazioni sui parametri del flusso video, come nome, periodo di conservazione dei dati e tipo di contenuto multimediale.

Inserisci contenuti multimediali nello stream

È possibile utilizzare i metodi forniti dalla libreria C++ (ad esempio `PutFrame`) per inserire dati nell'`KinesisVideoStream` oggetto. La libreria gestisce quindi lo stato interno dei dati, che può includere le seguenti attività:

- Esecuzione dell'autenticazione.
- Controllo della latenza di rete. Se la latenza è troppo elevata, la libreria potrebbe scegliere di eliminare i fotogrammi.
- Monitoraggio dello stato dello streaming in corso.

Interfacce di callback

Questo livello mostra un set di interfacce di richiamata, che consentono la comunicazione con il livello dell'applicazione. Queste interfacce di richiamata includono:

- Service callbacks interface (**CallbackProvider**): la libreria richiama gli eventi ottenuti tramite questa interfaccia quando crea un flusso, ottiene una descrizione del flusso ed elimina un flusso.
- Client-ready state or low storage events interface (**ClientCallbackProvider**): la libreria richiama gli eventi su questa interfaccia quando il client è pronto o quando rileva che potrebbe esaurirsi lo spazio di archiviazione o la memoria disponibile.
- Stream events callback interface (**StreamCallbackProvider**): la libreria richiama eventi su questa interfaccia quando si verificano eventi di streaming, come lo stream che entra nello stato pronto, i frame persi o gli errori di streaming.

Kinesis Video Streams fornisce implementazioni predefinite per queste interfacce. È inoltre possibile fornire un'implementazione personalizzata, ad esempio se è necessaria una logica di rete personalizzata o se si desidera esporre una condizione di archiviazione insufficiente all'interfaccia utente.

Per ulteriori informazioni sui callback nelle librerie dei produttori, vedere [richiamate SDK di Producer](#)

Procedura: utilizzare l'SDK per produttori C++

Questa procedura dimostra come utilizzare il client Kinesis Video Streams e le sorgenti multimediali in un'applicazione C++ per inviare dati al flusso video Kinesis.

La procedura include le seguenti fasi:

Prerequisiti

Prima di configurare il C++ producer SDK, assicurati di avere i seguenti prerequisiti:

- Credenziali: nel codice di esempio, fornisci le credenziali specificando un profilo che hai impostato nel file del profilo delle credenziali. AWS Se non lo hai già fatto, configura prima il profilo delle credenziali.

Per ulteriori informazioni, consulta [Configurare le AWS credenziali e l'area per lo sviluppo](#).

- Integrazione con l'archivio dei certificati: la libreria dei produttori di Kinesis Video Streams deve stabilire un rapporto di fiducia con il servizio che chiama. Ciò avviene tramite la convalida delle autorità di certificazione (CA) nell'archivio pubblico dei certificati. Linux-based Sui modelli, questo archivio si trova nella directory `/etc/ssl/`.

Scarica il certificato dal percorso seguente per lo store dei certificati:

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

- Installa le seguenti dipendenze di build per macOS:
 - [Autoconf 2.69](#) (Licenza GPLv3+/Autoconf: GNU GPL versione 3 o successive)
 - [CMake 3.7 o 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select versione 2347)
 - Java Development Kit (JDK) (per la compilazione di Java JNI)
 - [Lib-Pkg](#)
- Installa le seguenti dipendenze di compilazione per Ubuntu:
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`
 - configurazione pkg: `sudo apt install pkg-config`
 - Apri JDK: `sudo apt install openjdk-8-jdk`

Note

Questo è richiesto solo se stai creando Java Native Interface (JNI).

- Imposta la variabile di ambiente JAVA_HOME: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Scarica e configura il codice della libreria di produzione C++

Per informazioni su come scaricare e configurare la libreria di produzione C++, consulta [Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin e JNI](#).

Per i prerequisiti e ulteriori informazioni su questo esempio, consulta [the section called "C++"](#)

Argomenti CMake

Di seguito è riportata una tabella di riferimento per gli argomenti CMake di C++ Producer SDK-specific . Puoi anche passare le opzioni [standard di CMake](#) a CMake.

Important

Sono tutte opzionali.

Contrassegni per l'inclusione o l'esclusione di determinate funzionalità

Argomento CMake	Tipo	Predefinita	Spiegazione
BUILD_DEPENDENCIES	Booleano	ATTIVATO	Crea dipendenze dalla fonte. Altrimenti, usa le dipendenze già installate nel sistema. Se non è possibile trovare una delle dipendenze richieste , verrà restituito un errore.
BUILD_GSTREAMER_PLUGIN	Booleano	OFF	Crea il plugin kvssink GStreamer.
BUILD_JNI	Booleano	OFF	Crea la Java Native Interface (JNI) per poter chiamare questo codice da un ambiente di runtime Java.
ALIGNED_MEMORY_MODEL	Booleano	OFF	Se le allocazioni di memoria devono essere allineate ai limiti di 8 byte. Alcune

Argomento CMake	Tipo	Predefinita	Spiegazione
			architetture non consentono l'accesso non allineato alla memoria.
CONSTRAIN ED_DEVICE	Booleano	OFF	Non-Windows solo. Quando è attiva, imposta la dimensione dello stack di thread su 0.5 MiB. Necessario per le build di Alpine Linux. Altrimenti, viene utilizzata l'impostazione predefinita del sistema operativo.
BUILD_STATIC	Booleano	OFF	Crea librerie ed eseguibili come condivisi (OFF) o statici (ON).
ADD_MUCLIBC	Booleano	OFF	Collegati a uCLibc anziché alla libreria C standard, che è una libreria standard C più piccola progettata per sistemi integrati.
OPEN_SRC_ INSTALL_PREFIX	Stringa	../aprire- source/local	Posizione in cui installare le dipendenze open source, se create dal codice sorgente.

Bandiere per la compilazione incrociata

Important

Impostali se le architetture della CPU della macchina target e della macchina host sono diverse.

Argomento CMake	Tipo	Predefinita	Spiegazione
BUILD_LOG4CPLUS_HOST	Stringa	""	Crea la log4cplus dipendenza per l'architettura della CPU specificata. Se non è impostata, log4cplus rileverà automaticamente e utilizzerà l'architettura della CPU del computer host.
BUILD_OPENSSL_PLATFORM	Stringa	""	Crea la OpenSSL dipendenza per l'architettura della CPU specificata. Se non è impostata, OpenSSL rileverà automaticamente e utilizzerà l'architettura della CPU del computer host.

Bandiere relative ai test

Argomento CMake	Tipo	Predefinita	Spiegazione
BUILD_TEST	Booleano	OFF	Costruisci i test di unità e integrazione. Per eseguire tutti i test, esegui <code>./tst/producerTest</code> dalla directory build. AWS Le credenziali sono necessarie per eseguire i test.
CODE_COVERAGE	Booleano	OFF	Disponibile solo per i GNU/Clang compilatori. Abilita la raccolta della copertura del codice con gcov e la generazione di report.
COMPILER_WARNINGS	Booleano	OFF	Disponibile solo per GNU/Clang i compilatori. Abilita tutti gli avvisi del compilatore.
ADDRESS_SANITIZER	Booleano	OFF	Disponibile solo per GNU/Clang i compilatori. Costruisci con AddressSanitizer .
MEMORY_SANITIZER	Booleano	OFF	Disponibile solo per i GNU/Clang compilatori. Costruisci con MemorySanitizer .
THREAD_SANITIZER	Booleano	OFF	Disponibile solo per i GNU/Clang compilatori.

Argomento CMake	Tipo	Predefinita	Spiegazione
			ri. Costruisci con ThreadSanitizer .
UNDEFINED _BEHAVIOR _SANITIZER	Booleano	OFF	Disponibile solo per i GNU/Clang compilato ri. Costruisci con UndefinedBehaviorSanitizer .

Per utilizzare questi argomenti CMake, passali come elenco di `-Dkey=value` coppie separate da spazi dopo il comando. `cmake ..` Ad esempio:

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

CMake cercherà la toolchain del compilatore seguendo la variabile. `$PATH` Prima di eseguire CMake, imposta le variabili di CXX ambiente CC e per impostare esplicitamente la toolchain da utilizzare per la compilazione incrociata.

Scrivi ed esamina il codice

In questa sezione del [the section called "C++"](#), si esamina il codice contenuto nel test harness C++ (`tst/ProducerTestFixture.h` in altri file). Hai scaricato questo codice nella sezione precedente.

L'esempio di C++ indipendente dalla piattaforma mostra i seguenti modelli di codifica:

- Create un'istanza di `KinesisVideoProducer` per accedere a Kinesis Video Streams.
- Creare un'istanza di `KinesisVideoStream`. Questo crea un flusso video Kinesis nel tuo Account AWS se non esiste già uno stream con lo stesso nome.
- Chiama `putFrame` in `KinesisVideoStream` per ogni fotogramma di dati, non appena diventa disponibile, per inviarlo al flusso.

Le sezioni seguenti forniscono ulteriori informazioni su questo modello di codifica.

Create un'istanza di KinesisVideoProducer

Per creare l'oggetto `KinesisVideoProducer`, richiama il metodo `KinesisVideoProducer::createSync`. L'esempio seguente crea `KinesisVideoProducer` nel file `ProducerTestFixture.h`:

```
kinesis_video_producer_ = KinesisVideoProducer::createSync(move(device_provider_),
    move(client_callback_provider_),
    move(stream_callback_provider_),
    move(credential_provider_),
    defaultRegion_);
```

Il metodo `createSync` accetta i parametri seguenti:

- Un oggetto `DeviceInfoProvider`, che restituisce un oggetto `DeviceInfo` contenente informazioni sul dispositivo o sulla configurazione dello storage.

Note

Puoi configurare la dimensione dello store dei contenuti utilizzando il parametro `deviceInfo.storageInfo.storageSize`. I flussi di contenuti condividono lo store dei contenuti. Per determinare i requisiti delle dimensioni di storage, moltiplica la dimensione media dei fotogrammi per il numero di fotogrammi archiviati per la durata massima di tutti i flussi. Quindi moltiplica per 1,2 per prendere in considerazione la deframmentazione.

Supponi, ad esempio, che la tua applicazione abbia la seguente configurazione:

- Tre flussi
- 3 minuti di durata massima
- Ogni flusso è di 30 fotogrammi al secondo (FPS)
- Ogni fotogramma ha una dimensione di 10.000 KB

Il requisito del content store per questa applicazione è $3 \text{ (stream)} * 3 \text{ (minuti)} * 60 \text{ (secondi in un minuto)} * 10000 \text{ (kb)} * 1,2 \text{ (limite di deframmentazione)} = 194,4 \text{ Mb} \sim 200 \text{ Mb}$.

- Un oggetto `ClientCallbackProvider`, che restituisce puntatori delle funzioni che segnalano eventi specifici del client.
- Un oggetto `StreamCallbackProvider`, che restituisce puntatori delle funzioni richiamati quando si verificano eventi specifici del flusso.

- Un `CredentialProvider` oggetto che fornisce l'accesso alle variabili di ambiente relative alle credenziali. AWS
- Il Regione AWS («us-west-2»). L'endpoint del servizio è determinato dalla regione.

Crea un'istanza di `KinesisVideoStream`

Per creare l'oggetto `KinesisVideoStream`, richiama il metodo `KinesisVideoProducer::CreateStream` con un parametro `StreamDefinition`. L'esempio crea `KinesisVideoStream` nel file `ProducerTestFixture.h` con il tipo di traccia video e con id traccia pari a 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));
```

L'oggetto `StreamDefinition` dispone dei campi seguenti:

- Nome del flusso.
- Periodo di retention dei dati.
- Tag per il flusso. Questi tag possono essere utilizzati dalle applicazioni dei consumatori per trovare il flusso corretto o per avere maggiori informazioni sul flusso. I tag possono anche essere visualizzati in Console di gestione AWS.
- AWS KMS chiave di crittografia per lo stream. Per ulteriori informazioni, consulta [the section called “Protezione dei dati”](#).
- Tipo di streaming. Attualmente, l'unico valore valido è `STREAMING_TYPE_REALTIME`.
- Tipo di contenuti multimediali.
- Latenza file multimediali. Questo valore non è attualmente utilizzato e deve essere impostato su 0.

- Durata della riproduzione di ogni frammento.
- Scala del timecode dei file multimediali.
- Se i file multimediali usano la frammentazione del fotogramma chiave.
- Se i file multimediali utilizzano i timecode.
- Se i file multimediali usano i tempi assoluti del frammento.

Aggiungete una traccia audio allo stream video di Kinesis

È possibile aggiungere i dettagli della traccia audio alla definizione dello streaming di una traccia video utilizzando il `addTrack` metodo `StreamDefinition`:

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

Il `addTrack` metodo richiede i seguenti parametri:

- ID della traccia (come uno per l'audio). Questo deve essere unico e un valore diverso da zero.
- User-defined nome della traccia (ad esempio, «audio» per la traccia audio).
- ID del codec per questa traccia (ad esempio, per la traccia audio «A_AAC»).
- Tipo di traccia (ad esempio, usa il valore enum di `MKV_TRACK_INFO_TYPE_AUDIO` per l'audio).

Se si dispone di dati provati codec per la traccia audio, è possibile passarli quando si chiama la funzione `addTrack`. È inoltre possibile inviare i dati privati del codec dopo aver creato l'oggetto mentre si chiama il metodo `start` in `KinesisVideoStream`.

Inserite un fotogramma nel flusso video di Kinesis

Inserite contenuti multimediali nel flusso video Kinesis

utilizzando `KinesisVideoStream::putFrame`, passando un `Frame` oggetto che contiene l'intestazione e i dati multimediali. L'esempio richiama `putFrame` nel file `ProducerApiTest.cpp`:

```
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

L'esempio precedente di produttore in C++ invia un buffer di dati di test. In un'applicazione reale, è consigliabile ottenere il buffer e la dimensione del fotogramma dai dati del fotogramma da un'origine multimediale (ad esempio una fotocamera).

L'oggetto Frame dispone dei campi seguenti:

- Indice dei fotogrammi. Deve essere un valore a incremento monotono.
- Flag associati al fotogramma. Ad esempio, se il codificatore è stato configurato per ottenere un fotogramma chiave, a questo fotogramma verrà assegnato il contrassegno `FRAME_FLAG_KEY_FRAME`.
- Timestamp di decodifica.
- Timestamp della presentazione.
- Durata del fotogramma (in unità di 100 ns).
- Dimensione del fotogramma in byte.
- Dati del fotogramma.

Per ulteriori informazioni sul formato della cornice, vedere [the section called “Modello di dati”](#)

Inserire un KinesisVideoFrame in una traccia specifica di KinesisVideoStream

Puoi usare la `PutFrameHelper` classe per inserire i dati dei frame in una traccia specifica. Innanzitutto, chiama `getFrameDataBuffer` per ottenere un puntatore a uno dei buffer preallocati per inserire i dati. `KinesisVideoFrame` Quindi, puoi chiamare `putFrameMultiTrack` per inviare `KinesisVideoFrame` insieme al valore booleano per indicare il tipo di dati del frame. Utilizza "true" se si tratta di dati video o "false" se il frame contiene dati audio. Il `putFrameMultiTrack` metodo utilizza un meccanismo di accodamento per garantire che i frammenti MKV mantengano i timestamp dei frame crescenti in modo monotono e che due frammenti qualsiasi non si sovrappongano. Ad esempio, il timestamp MKV del primo fotogramma di un frammento deve essere sempre maggiore del timestamp MKV dell'ultimo fotogramma del frammento precedente.

Ha i seguenti campi: `PutFrameHelper`

- Numero massimo di frame audio in coda.
- Numero massimo di fotogrammi video in coda.
- Dimensione da allocare per un singolo frame audio.
- Dimensioni da assegnare a un singolo fotogramma video.

Metriche di accesso e registrazione delle metriche

L'SDK per produttori C++ include funzionalità per le metriche e la registrazione delle metriche.

Puoi utilizzare le operazioni `getKinesisVideoMetrics` e `getKinesisVideoStreamMetrics` API per recuperare informazioni su Kinesis Video Streams e sui tuoi stream attivi.

Il seguente codice è tratto dal file `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);
```

L'oggetto `PClientMetrics` compilato da `getKinesisVideoMetrics` contiene le seguenti informazioni:

- `contenutoStoreSize`: la dimensione complessiva in byte del content store (la memoria utilizzata per archiviare i dati in streaming).
- `contenutoStoreAvailableSize`: la memoria disponibile nel content store, in byte.
- `contentStoreAllocatedSize`: la memoria allocata nel content store.
- `totaleContentViewsSize`: la memoria totale utilizzata per la visualizzazione del contenuto. La visualizzazione del contenuto è una serie di indici di informazioni nel content store.
- `totaleFrameRate`: il numero aggregato di fotogrammi al secondo in tutti gli stream attivi.
- `totaleTransferRate`: il totale dei bit al secondo (bps) inviati in tutti gli stream.

L'oggetto `PStreamMetrics` compilato da `getKinesisVideoStreamMetrics` contiene le seguenti informazioni:

- `attualeViewDuration`: la differenza in unità da 100 ns tra l'instanziazione della visualizzazione del contenuto (quando i frame sono codificati) e la posizione corrente (quando i dati dei frame vengono inviati a Kinesis Video Streams).
- `complessivoViewDuration`: la differenza in unità da 100 ns tra l'inizio della visualizzazione del contenuto (quando i frame sono codificati) e la coda (quando i frame vengono svuotati dalla memoria, o perché viene superato lo spazio totale allocato per la visualizzazione del contenuto o perché viene ricevuto un `PersistedAck` messaggio da Kinesis Video Streams e i frame noti per essere persistenti vengono svuotati).
- `currentViewSize`: la dimensione in byte del contenuto visualizzato dalla testa (quando i frame sono codificati) alla posizione corrente (quando i frame vengono inviati a Kinesis Video Streams).
- `complessivaViewSize`: la dimensione totale in byte della visualizzazione del contenuto.

- `attualeFrameRate`: l'ultima frequenza misurata dello stream, in fotogrammi al secondo.
- `attualeTransferRate`: l'ultima frequenza misurata dello stream, in byte al secondo.

Teardown:

Se desideri inviare i byte rimanenti in un buffer e attendere il ACK, puoi utilizzare `stopSync`:

```
kinesis_video_stream->stopSync();
```

In alternativa, puoi chiamare `stop` per terminare lo streaming:

```
kinesis_video_stream->stop();
```

Dopo aver arrestato il flusso, puoi liberarlo richiamando la seguente API:

```
kinesis_video_producer_->freeStream(kinesis_video_stream);
```

Esegui e verifica il codice

Per eseguire e verificare il codice per [the section called “C++”](#), consulta le seguenti OS-specific istruzioni:

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [Sistema operativo Raspberry Pi](#)

Puoi monitorare il traffico sul tuo stream osservando le metriche associate allo streaming nella CloudWatch console Amazon, ad esempio. `PutMedia.IncomingBytes`

Usa il C++ producer SDK come plugin GStreamer

[GStreamer](#) è un popolare framework multimediale utilizzato da più fotocamere e sorgenti video per creare pipeline multimediali personalizzate combinando plugin modulari. Il plugin Kinesis Video Streams GStreamer semplifica l'integrazione della pipeline multimediale GStreamer esistente con Kinesis Video Streams.

Per informazioni sull'utilizzo del C++ producer SDK come plug-in GStreamer, consultate. [Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink](#)

Usa il C++ producer SDK come plugin GStreamer in un contenitore Docker

[GStreamer](#) è un popolare framework multimediale utilizzato da più fotocamere e sorgenti video per creare pipeline multimediali personalizzate combinando plugin modulari. Il plugin Kinesis Video Streams GStreamer semplifica l'integrazione della pipeline multimediale GStreamer esistente con Kinesis Video Streams.

Inoltre, l'utilizzo di [Docker](#) per creare la pipeline GStreamer standardizza l'ambiente operativo per Kinesis Video Streams, semplificando la creazione e l'esecuzione dell'applicazione.

Per informazioni sull'utilizzo del C++ producer SDK come plug-in GStreamer in un contenitore Docker, consulta. [Esegui l'elemento GStreamer in un contenitore Docker](#)

Usa la registrazione con il C++ producer SDK

È possibile configurare la registrazione per le applicazioni C++ producer SDK nel `kvs_log_configuration` file nella cartella `kinesis-video-native-build`

L'esempio seguente mostra la prima riga del file di configurazione predefinito, che configura l'applicazione per scrivere voci di livello DEBUG nella Console di gestione AWS:

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

È possibile impostare il livello di logging su INFO per ottenere un log meno dettagliato.

Per configurare l'applicazione per scrivere le voci di registro in un file di registro, aggiorna la prima riga del file con la seguente:

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

Ciò consente di configurare l'applicazione per scrivere voci di log nel file `kvs.log` nella cartella `kinesis-video-native-build/log`.

Per modificare la posizione del file di log, aggiorna la riga seguente con il nuovo percorso:

```
log4cplus.appender.KvsFileAppender.File=../log/kvs.log
```

 Note

Se il logging di livello DEBUG viene scritto in un file, il file di log può occupare rapidamente tutto lo spazio di storage disponibile nel dispositivo.

Usa la libreria C producer

Puoi utilizzare la libreria C producer fornita da Amazon Kinesis Video Streams per scrivere codice applicativo per inviare dati multimediali da un dispositivo a un flusso video Kinesis.

Modello a oggetti

La libreria di produttori Kinesis Video Streams C si basa su un componente comune chiamato Platform Independent Codebase (PIC), disponibile all'indirizzo. GitHub <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/> Il PIC contiene una logica di business indipendente dalla piattaforma per i componenti fondamentali. La libreria di produttori Kinesis Video Streams C include PIC con un livello di API aggiuntivo che consente callback ed eventi specifici per scenari e piattaforme. La libreria di produttori Kinesis Video Streams C ha i seguenti componenti basati su PIC:

- Provider di informazioni sui dispositivi: espone la `DeviceInfo` struttura che può essere fornita direttamente all'API PIC. È possibile configurare un set di provider, incluso un provider ottimizzato per gli scenari applicativi in grado di ottimizzare l'archivio di contenuti in base al numero e ai tipi di flussi gestiti dall'applicazione e alla quantità di buffering richiesta configurata in base alla quantità di RAM disponibile.
- Stream info provider: espone la `StreamInfo` struttura che può essere fornita direttamente all'API PIC. Esiste una serie di provider specifici per i tipi di applicazioni e i tipi più comuni di scenari di streaming. Questi includono provider come video, audio e multitraccia audio e video. Ciascuno di questi scenari presenta impostazioni predefinite che è possibile personalizzare in base ai requisiti dell'applicazione.
- Provider di callback: espone la `ClientCallbacks` struttura che può essere fornita direttamente all'API PIC. Ciò include un set di provider di callback per il networking (callback CURL-based API), l'autorizzazione (API delle AWS credenziali) e la ripetizione dello streaming in caso di errore. L'API Callback Provider richiede una serie di argomenti per la configurazione, come le informazioni di autorizzazione e. Regione AWS Questa operazione viene eseguita utilizzando certificati IoT o utilizzando AWS AccessKeyId SecretKey, o SessionToken. È possibile potenziare i provider di

callback con callback personalizzati se l'applicazione ha bisogno di ulteriori elaborazioni di un determinato callback per raggiungere la logica specifica dell'applicazione.

- **FrameOrderCoordinator**— Aiuta a gestire la sincronizzazione audio e video per scenari multitraccia. Ha un comportamento predefinito, che è possibile personalizzare per gestire la logica specifica dell'applicazione. Inoltre, semplifica l'imballaggio dei metadati dei frame nella struttura PIC Frame prima di inviarli all'API PIC di livello inferiore. Per gli scenari non multitraccia, questo componente è un punto di passaggio all'API putFrame PIC.

La libreria C fornisce i seguenti oggetti per gestire l'invio di dati a un flusso video di Kinesis:

- **KinesisVideoClient**— Contiene informazioni sul dispositivo e mantiene i callback per segnalare gli eventi di Kinesis Video Streams.
- **KinesisVideoStream**— Rappresenta informazioni sui parametri del flusso video, come nome, periodo di conservazione dei dati e tipo di contenuto multimediale.

Inserisci contenuti multimediali nello stream

È possibile utilizzare i metodi forniti dalla libreria C (ad esempio `PutKinesisVideoFrame`) per inserire dati nell'`KinesisVideoStream` oggetto. La libreria gestisce quindi lo stato interno dei dati, che può includere le seguenti attività:

- Esecuzione dell'autenticazione.
- Controllo della latenza di rete. Se la latenza è troppo elevata, la libreria potrebbe scegliere di eliminare i fotogrammi.
- Monitoraggio dello stato dello streaming in corso.

Procedura: utilizzare C Producer SDK

Questa procedura dimostra come utilizzare il client Kinesis Video Streams e le sorgenti multimediali in un'applicazione C per H.264-encoded inviare frame video al flusso video Kinesis.

La procedura include le seguenti fasi:

- [Scarica il codice della libreria C Producer](#)
- [Scrivi ed esamina il codice](#)
- [Esegui e verifica il codice](#)

Prerequisiti

Prima di configurare C Producer SDK, assicurati di avere i seguenti prerequisiti:

- **Credenziali:** nel codice di esempio, fornite le credenziali specificando un profilo che avete impostato nel file di profilo delle credenziali. AWS Se non lo hai già fatto, configura prima il profilo delle credenziali.

Per ulteriori informazioni, consulta [Configurare le AWS credenziali](#) e la regione per lo sviluppo.

- **Integrazione dell'archivio certificati:** la libreria dei produttori di Kinesis Video Streams deve stabilire un rapporto di fiducia con il servizio che chiama. Ciò avviene tramite la convalida delle autorità di certificazione (CA) nell'archivio di certificati pubblico. Linux-based Sui modelli, questo archivio si trova nella directory `/etc/ssl /`.

Scarica il certificato dal percorso seguente per lo store dei certificati:

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

- Installa le seguenti dipendenze di build per macOS:
 - [Autoconf 2.69](#) (Licenza GPLv3+/Autoconf: GNU GPL versione 3 o successive)
 - [CMake 3.7 o 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select versione 2347)
 - Java Development Kit (JDK) (per la compilazione di Java JNI)
 - [Lib-Pkg](#)
- Installa le seguenti dipendenze di build per Ubuntu:
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`
 - configurazione pkg: `sudo apt install pkg-config`
 - OpenJDK: `sudo apt install openjdk-8-jdk`
 - Imposta la variabile di ambiente JAVA_HOME: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Scarica il codice della libreria C Producer

In questa sezione sarà possibile scaricare le librerie di basso livello. Per i prerequisiti e altri dettagli su questo esempio, vedere [the section called “C++”](#).

1. Create una directory, quindi clonate il codice sorgente di esempio dal GitHub repository.

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

Note

Se non si esegue Git clone con `--recursive`, eseguire `git submodule update --init` nella directory `amazon-kinesis-video-streams-producer-c/open-source`. È inoltre necessario installare `pkg-config`, `CMake` e un ambiente di compilazione.

Per ulteriori informazioni, consulta `README.md` in <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git>.

2. Apri il codice nell'ambiente di sviluppo integrato (IDE) di tua scelta (ad esempio, [Eclipse](#)).

Scrivi ed esamina il codice

In questa sezione, si esamina il codice dell'applicazione `KvsVideoOnlyStreamingSample.c` di esempio `samples` nella cartella del <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c> repository on GitHub. Hai scaricato questo codice nella fase precedente. Questo esempio dimostra come utilizzare la libreria C producer per inviare frame H.264-encoded video all'interno della cartella `samples/h264SampleFrames` al flusso video Kinesis.

Questa applicazione di esempio ha tre sezioni:

- Inizializzazione e configurazione:
 - Inizializzazione e configurazione della pipeline dei supporti specifici della piattaforma.
 - Inizializzazione e configurazione `KinesisVideoClient` e `KinesisVideoStream` per la pipeline, impostazione dei callback, integrazione dell'autenticazione specifica dello scenario, estrazione e invio di dati privati del codec e trasferimento dello stream allo stato `READY`.

- Loop principale:
 - Ottenere il frame dalla pipeline dei supporti con timestamp e flag.
 - KinesisVideoStreamInvio del frame a.
- Teardown:
 - Arresto (sincronizzazione) KinesisVideoStream, liberazione, liberazione KinesisVideoStream.
 - KinesisVideoClient

Questa applicazione di esempio completa le attività seguenti:

- Chiama l'API `createDefaultDeviceInfo` per creare l'oggetto `deviceInfo` che contiene informazioni sul dispositivo o sulla configurazione dello storage.

```
// 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;
```

- Chiama l'API `createRealtimeVideoStreamInfoProvider` per creare l'oggetto `StreamInfo`.

```
CHK_STATUS(createRealtimeVideoStreamInfoProvider(streamName,
    DEFAULT_RETENTION_PERIOD, DEFAULT_BUFFER_DURATION, &pStreamInfo));
// adjust members of pStreamInfo here if needed
```

- Chiama l'`createDefaultCallbacksProviderWithAwsCredentialsAPI` per creare il provider di callback predefinito basato su credenziali statiche. AWS

```
CHK_STATUS(createDefaultCallbacksProviderWithAwsCredentials(accessKey,
    secretKey,
    sessionToken,
    MAX_UINT64,
    region,
    cacertPath,
    NULL,
    NULL,
    FALSE,
    &pClientCallbacks));
```

- Chiama l'`createKinesisVideoClientAPI` per creare l'`KinesisVideoClient` oggetto che contiene informazioni sull'archiviazione del dispositivo e mantiene i callback per segnalare gli eventi di Kinesis Video Streams.

```
CHK_STATUS(createKinesisVideoClient(pDeviceInfo, pClientCallbacks, &clientHandle));
```

- Chiama l'API `createKinesisVideoStreamSync` per creare l' oggetto `KinesisVideoStream`.

```
CHK_STATUS(createKinesisVideoStreamSync(clientHandle, pStreamInfo, &streamHandle));
```

- Configura un frame di esempio e chiama l'API `PutKinesisVideoFrame` per l'invio di frame all'oggetto `KinesisVideoStream`.

```
// 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;
}
```

- Teardown:

```
CHK_STATUS(stopKinesisVideoStreamSync(streamHandle));  
CHK_STATUS(freeKinesisVideoStream(&streamHandle));  
CHK_STATUS(freeKinesisVideoClient(&clientHandle));
```

Esegui e verifica il codice

Per eseguire e verificare il codice per [ilthe section called “C++”](#), procedi come segue:

1. Esegui i seguenti comandi per creare una build directory nell'[SDK C scaricato](#) e avviala cmake da essa:

```
mkdir -p amazon-kinesis-video-streams-producer-c/build;  
cd amazon-kinesis-video-streams-producer-c/build;  
cmake ..
```

È possibile inviare le seguenti opzioni a cmake ..

- -DBUILD_DEPENDENCIES- se creare librerie dipendenti dai sorgenti.
- -DBUILD_TEST=TRUE- build unit e test di integrazione. Potrebbe essere utile per confermare il supporto per il tuo dispositivo.

./tst/webrtc_client_test

- -DCODE_COVERAGE- abilitare la segnalazione della copertura.
- -DCOMPILER_WARNINGS- abilita tutti gli avvisi del compilatore.
- -DADDRESS_SANITIZER- costruisci con AddressSanitizer
- -DMEMORY_SANITIZER- costruisci con MemorySanitizer.
- -DTHREAD_SANITIZER- costruisci con ThreadSanitizer.
- -DUNDEFINED_BEHAVIOR_SANITIZER- costruisci con UndefinedBehaviorSanitizer.
- -DALIGNED_MEMORY_MODEL: compilazione per dispositivi solo con modello di memoria allineato Il valore predefinito è “OFF”.

2. Passa alla build directory che hai appena creato con il passaggio precedente ed esegui make per creare l'SDK WebRTC C e gli esempi forniti.

```
make
```

3. L'applicazione di esempio `kinesis_video_cproducer_video_only_sample` invia fotogrammi video con codifica h.264 all'interno della cartella a Kinesis `samples/h264SampleFrames` Video Streams. Il seguente comando invia i fotogrammi video in loop per dieci secondi a Kinesis Video Streams:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Se desideri inviare H.264-encoded frame da un'altra cartella (ad esempio, `MyH264FramesFolder`), esegui l'esempio con i seguenti argomenti:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10 MyH264FramesFolder
```

4. Per abilitare i log dettagliati, definite `HEAP_DEBUG` e `LOG_STREAMING` C-defines decommentando le righe appropriate. `CMakeList.txt`

Puoi monitorare l'avanzamento della suite di test nell'output di debug nell'IDE. Puoi anche monitorare il traffico sul tuo stream guardando le metriche associate al tuo stream nella CloudWatch console Amazon, ad esempio `PutMedia.IncomingBytes`.

Note

Poiché il test harness invia solo fotogrammi di byte vuoti, la console non visualizza i dati come un flusso di video.

Usa l'SDK per produttori C++ su Raspberry Pi

Raspberry Pi è un piccolo computer economico che può essere utilizzato per insegnare e apprendere competenze di programmazione di base. Questo tutorial descrive come configurare e utilizzare l'SDK per produttori C++ di Amazon Kinesis Video Streams su un dispositivo Raspberry Pi. Le fasi includono anche come verificare l'installazione utilizzando l'applicazione demo `GStreamer`.

Argomenti

- [Prerequisiti](#)
- [Crea un utente IAM con il permesso di scrivere su Kinesis Video Streams](#)

- [Collega il tuo Raspberry Pi alla tua rete Wi-Fi](#)
- [Connettiti in remoto al tuo Raspberry Pi](#)
- [Configura la videocamera Raspberry Pi](#)
- [Prerequisiti per l'installazione del software](#)
- [Scarica e crea l'SDK per produttori C++ di Kinesis Video Streams](#)
- [Trasmetti i video sul tuo stream video Kinesis](#)
- [Riproduci contenuti multimediali dal vostro streaming video Kinesis](#)
- [Risoluzione dei problemi di compilazione su C++ producer SDK per Raspberry Pi](#)

Prerequisiti

Prima di configurare il C++ producer SDK sul tuo Raspberry Pi, assicurati di avere i seguenti prerequisiti:

- Un dispositivo Raspberry Pi con la seguente configurazione:
 - Versione della scheda: 3 Model B o versioni successive.
 - Un modulo [telecamera collegato](#) o una videocamera USB connessa (web cam).
 - Una scheda SD con una capacità minima di 8 GB.
 - Sistema operativo Raspbian (kernel 4.9 o versioni successive) installato. È possibile scaricare l'ultima immagine del sistema operativo Raspberry Pi (precedentemente chiamata Raspbian) dal sito Web [Raspberry Pi](#). Segui le istruzioni di Raspberry Pi per [installare l'immagine scaricata su una scheda SD](#).
- E Account AWS con uno streaming video Kinesis. Per ulteriori informazioni, consulta [Nozioni di base su Kinesis Video Streams](#).

Crea un utente IAM con il permesso di scrivere su Kinesis Video Streams

Se non l'hai già fatto, configura un utente AWS Identity and Access Management (IAM) con le autorizzazioni per scrivere su uno streaming video Kinesis.

Queste procedure hanno lo scopo di aiutarti a iniziare rapidamente a utilizzare una coppia di chiavi di AWS accesso. I dispositivi possono utilizzare X.509 i certificati a cui connettersi AWS IoT. [the section called “Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT”](#) Per ulteriori

informazioni su come configurare il dispositivo per utilizzare l'autenticazione basata su certificati, consulta.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel menu di navigazione a sinistra, scegli Users (Utenti).
3. Per creare un nuovo utente, scegli Add user (Aggiungi utente).
4. Fornire un User name (Nome utente) descrittivo per l'utente, ad esempio **kinesis-video-raspberry-pi-producer**.
5. Sotto Access type (Tipo di accesso), scegli Programmatic access (Accesso programmatico).
6. Scegli Successivo: autorizzazioni.
7. Sotto Set permissions for kinesis-video-raspberry-pi-producer (Imposta autorizzazioni per kinesis-video-raspberry-pi-producer), scegli Attach existing policies directly (Collega direttamente policy esistenti).
8. Scegli Crea policy. La pagina Create policy (Crea policy) viene visualizzata in una nuova scheda del browser Web.
9. Scegli la scheda JSON.
10. Copia la seguente policy JSON e incollala nell'area del testo. Questa policy consente all'utente di creare e scrivere dati nei flussi video Kinesis.

JSON

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

11. Scegliere Esamina policy.
12. Fornire un Name (Nome) alla policy, ad esempio **kinesis-video-stream-write-policy**.
13. Scegli Crea policy.
14. Torna alla scheda Add user (Aggiungi utente) nel browser e scegli Refresh (Aggiorna).
15. Nella casella di ricerca digita il nome della policy creata.
16. Seleziona la casella di controllo accanto alla nuova policy nell'elenco.
17. Scegliere Next:Review (Successivo:Rivedi).
18. Selezionare Create user (Crea utente).
19. Nella console viene visualizzato l'ID chiave di accesso per il nuovo utente. Scegli Show (Mostra) per visualizzare la Secret Access Key (Chiave di accesso segreta). Registra questi valori perché saranno necessari per la configurazione dell'applicazione.

Collega il tuo Raspberry Pi alla tua rete Wi-Fi

Se usi un monitor e una tastiera collegati, passa a [Configura la videocamera Raspberry Pi](#).

Queste istruzioni sono state scritte per aiutarti a configurare il tuo Raspberry Pi quando viene eseguito in modalità headless, ovvero senza tastiera, monitor o cavo di rete collegati. Segui le istruzioni seguenti per configurare il tuo Raspberry Pi in modo che tenti automaticamente di connettersi alla rete specificata, permettendo alla macchina host di accedervi tramite SSH.

1. Sul tuo computer, crea un file denominato `wpa_supplicant.conf`.
2. Copia il testo seguente e incollalo nel `wpa_supplicant.conf` file:

```
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"
}
```

Sostituisci i psk valori ssid e con le informazioni per la tua Wi-Fi rete.

3. Copia il file `wpa_supplicant.conf` nella scheda SD. Devi copiarlo nella directory principale del volume boot.
4. Inserisci la scheda SD nel Raspberry Pi e accendi il dispositivo. Si unisce alla Wi-Fi rete e SSH è abilitato.

Connettiti in remoto al tuo Raspberry Pi

Se usi un monitor e una tastiera collegati, passa a [Configura la videocamera Raspberry Pi](#).

Queste istruzioni sono state scritte per aiutarti a configurare il tuo Raspberry Pi quando viene eseguito in modalità headless, ovvero senza tastiera, monitor o cavo di rete collegati. Segui le istruzioni riportate di seguito per localizzare il tuo Raspberry Pi sulla tua rete e accedervi tramite SSH dal tuo computer host.

1. Prima di connetterti al dispositivo Raspberry Pi in remoto, esegui una delle operazioni seguenti per determinarne l'indirizzo IP:
 - Se hai accesso al Wi-Fi router della tua rete, guarda i dispositivi collegati Wi-Fi . Trova il dispositivo denominato Raspberry Pi per trovare l'indirizzo IP del dispositivo.
 - Se non hai accesso al Wi-Fi router della tua rete, puoi utilizzare altri software per trovare i dispositivi sulla rete. [Fing](#) è un'applicazione molto diffusa disponibile per dispositivi sia Android che iOS. Puoi utilizzare la versione gratuita di questa applicazione per trovare gli indirizzi IP dei dispositivi in rete.
2. Una volta trovato l'indirizzo IP del dispositivo Raspberry Pi, puoi usare qualsiasi applicazione terminale per connetterti.
 - Su macOS o Linux, usa `ssh`:

```
ssh pi@<IP address>
```

- Su Windows, usa [PuTTY](#), un client SSH gratuito per Windows.

Per una nuova installazione di Raspbian, il nome utente è **pi** e la password è **raspberry**. Ti consigliamo di [modificare la password predefinita](#).

Configura la videocamera Raspberry Pi

Segui questi passaggi per configurare il modulo videocamera [Raspberry Pi](#) per inviare video dal dispositivo a un flusso video Kinesis.

Note

Se utilizzi una webcam USB, passa a [the section called “Prerequisiti per l'installazione del software”](#)

Camera module 1

Segui queste istruzioni per aggiornare il file dei moduli, abilitare l'interfaccia della fotocamera e verificare la funzionalità della videocamera. L'aggiornamento del file dei moduli indica al Raspberry Pi quali moduli del kernel caricare all'avvio. Il driver della fotocamera non viene caricato di default per conservare le risorse di sistema sui dispositivi Raspberry Pi che non utilizzano una fotocamera.

1. Apri un editor per apportare modifiche al file dei moduli. Apri il terminale e usa il seguente comando per modificare il file usando l'nanoeditor:

```
sudo nano /etc/modules
```

2. Aggiungi la riga seguente alla fine del file, se non è già presente:

```
bcm2835-v4l2
```

3. Salvare il file e uscire dall'editor. Per salvare e uscire usando l'nanoeditor, usa Ctrl+X.
4. Riavvia il dispositivo Raspberry Pi:

```
sudo reboot
```

5. Quando il dispositivo si riavvia, esegui di nuovo la connessione utilizzando l'applicazione terminale se ti connetti in remoto.
6. Aprire raspi-config:

```
sudo raspi-config
```

7. Scegli Opzioni di interfaccia, Legacy Camera. Nelle versioni precedenti del sistema operativo Raspbian, questa opzione di menu potrebbe trovarsi in Opzioni di interfaccia, Fotocamera.

Abilita la fotocamera se non è già abilitata, quindi riavvia se viene richiesto.

8. Verifica che la fotocamera funzioni digitando il comando seguente:

```
raspistill -v -o test.jpg
```

Se la fotocamera è configurata correttamente, questo comando acquisisce un'immagine dalla fotocamera, la salva in un file denominato e visualizza messaggi `test.jpg` informativi.

Camera module 2 or 3

Se si utilizza un modulo Camera 2, si utilizza `bcm2835-v4l2` (versione precedente) o `libcamera` (moderna). Tuttavia, lo `libcamera` stack è consigliato per un supporto e funzionalità migliori. Segui i passaggi seguenti per confermare che `libcamera` sia aggiornato sul tuo sistema.

1. [libcamera](#) dovrebbe essere preinstallata sul tuo Raspberry Pi. Verifica la presenza di eventuali aggiornamenti e aggiorna alla versione più recente per correzioni di bug e aggiornamenti di sicurezza. Apri il terminale e digita i seguenti comandi:

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

2. Riavvia il sistema per rendere effettivi gli aggiornamenti.

```
sudo reboot
```

3. Metti alla prova la tua fotocamera. Questa applicazione avvia un flusso di anteprima della fotocamera e lo visualizza sullo schermo.

```
libcamera-hello
```

Se hai problemi con il modulo della fotocamera, consulta la documentazione del [Raspberry Pi](#) per la risoluzione dei problemi.

Prerequisiti per l'installazione del software

Il C++ producer SDK richiede l'installazione dei seguenti prerequisiti software su Raspberry Pi.

1. Aggiorna l'elenco dei pacchetti e installa le librerie necessarie per creare l'SDK. Apri il terminale e digita i seguenti comandi:

```
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. Se stai usando lo `libcamera` stack, installa anche il plugin `libcamerasrc` GStreamer. Questo plugin GStreamer non viene installato di default.

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

3. Copia il file PEM seguente in `/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
```

Scarica e crea l'SDK per produttori C++ di Kinesis Video Streams

Segui le procedure seguenti per scaricare e creare l'SDK per produttori C++ di [Kinesis Video Streams](#). Assicurati di aver installato i prerequisiti del software; consulta questi passaggi. [the section called “Prerequisiti per l'installazione del software”](#)

1. Vai alla directory di download. Apri un terminale e passa alla tua directory di download preferita.

Ad esempio:

```
cd ~/Downloads
```

2. Clona il repository SDK. Usa il `git clone` comando per scaricare l'SDK dal repository. GitHub Type:

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

Questo comando clona solo un ramo (il `master` ramo), riducendo le dimensioni e il tempo di download. Inoltre, inserisce i contenuti scaricati in una cartella denominata `kvs-producer-sdk-cpp` all'interno della directory corrente.

3. Verifica il download. Una volta completato il processo di clonazione, elenca il contenuto della `kvs-producer-sdk-cpp` cartella per verificare che l'SDK sia stato scaricato.

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

4. Prepara una directory di compilazione. Type:

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

5. Configura la build. Esegui il `cmake` comando seguente per configurare l'ambiente di compilazione con opzioni specifiche:

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

[CMake](#) utilizza le seguenti opzioni per generare le opzioni appropriate `Makefiles`:

- Usando la cartella del progetto (. .) come directory di origine.
- Utilizzo della directory corrente (.) (build/) per l'output della compilazione.
- -DBUILD_GSTREAMER_PLUGIN=0Nabilita la creazione del plugin GStreamer kvssink.
- -DBUILD_DEPENDENCIES=0FFdisabilita la creazione di dipendenze esterne dai sorgenti. Il progetto troverà e utilizzerà le dipendenze esterne installate in una fase precedente.
- -DALIGNED_MEMORY_MODEL=0Ndisabilita il modello di memoria non allineato. L'accesso non allineato alla memoria non è supportato da alcuni dispositivi Raspberry Pi.

Note

Per un elenco completo degli argomenti di CMake, consulta. [the section called “Scarica e configura il codice”](#)

6. Compilare il progetto. Dopo aver configurato la build, usa il make comando per compilare usando il Makefile generato da. cmake

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

L'- jargomento to make consente di eseguire più lavori di compilazione in parallelo. Per ridurre i tempi di compilazione, utilizzate il nproc comando per calcolare dinamicamente il numero di core della CPU sul vostro Raspberry Pi.

7. Conferma che è presente libgstkvssink.so.

Elenca i file nella directory corrente.

Richiesta:

```
ls
```

Risposta:

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

8. Conferma che GStreamer può caricare. kvssink

Imposta la variabile di GST_PLUGIN_PATH ambiente sulla directory che contiene. `libgstkvssink.so`

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

Fai caricare GStreamer: kvssink

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

Dovresti vedere della documentazione su. kvssink Usa i tasti freccia per navigare e premi q per uscire.

9. (Facoltativo) Aggiorna lo script di avvio della shell per includere l'impostazione della variabile di GST_PLUGIN_PATH ambiente. Ciò garantisce che GST_PLUGIN_PATH sia impostato correttamente durante una nuova sessione di terminale. Sui dispositivi Raspberry Pi, lo script di avvio della shell è. `~/ .bashrc`

Esegui il comando seguente per aggiungere il comando alla fine dello script di avvio della shell.

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

Digita quanto segue per eseguire lo script di avvio della shell, oppure chiudi la shell corrente e aprirne una nuova.

```
source ~/.bashrc
```

Conferma GST_PLUGIN_PATH che è impostato e puoi caricare kvssink.

```
echo $GST_PLUGIN_PATH
```

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

Trasmetti i video sul tuo stream video Kinesis

Per eseguire l'applicazione di esempio sono necessarie le seguenti informazioni:

Trasmetti in streaming un live streaming

- Il nome del flusso creato nella sezione [Prerequisiti](#).
- Le credenziali dell'account (ID chiave di accesso e chiave di accesso segreta) create in [Crea un utente IAM con il permesso di scrivere su Kinesis Video Streams](#).
- GStreamer è in grado di localizzare il plugin. kvssink Per ulteriori informazioni, consulta [the section called “Scaricate e compilate l'SDK per produttori C++”](#).

1. Imposta le credenziali e la regione.

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

Per altri metodi di autenticazione, vedere [the section called “Fornisci le credenziali a kvssink”](#).

 Note

L'SDK per produttori di C++ utilizza la regione Stati Uniti occidentali (Oregon) (us-west-2) per impostazione predefinita. Per utilizzare l'impostazione predefinita, Regione AWS crea il tuo streaming video Kinesis nella regione Stati Uniti occidentali (Oregon). Per utilizzare una regione diversa per il tuo streaming video Kinesis, imposta la seguente variabile di ambiente sulla tua regione (ad esempio,): *us-east-1*

```
export AWS_DEFAULT_REGION=us-east-1
```

2. A seconda del contenuto multimediale di input, scegliete una delle seguenti opzioni:

Sample GStreamer video

Questa pipeline GStreamer genera un flusso video di test live con uno schema di test standard che viene eseguito a 10 fotogrammi al secondo con una risoluzione di 640x480 pixel. Viene aggiunto un overlay che mostra l'ora e la data correnti del sistema. Il video viene quindi codificato in un H.264 formato e i fotogrammi chiave vengono generati al massimo ogni 10 fotogrammi, ottenendo una durata del frammento (nota anche come dimensione di un gruppo di immagini (GoP)) di 1 secondo. kvssink prende il flusso H.264-encoded video, lo impacchetta nel formato contenitore Matroska (MKV) e lo carica nel flusso video Kinesis.

Esegui il comando seguente:

```
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"
```

Per interrompere la pipeline GStreamer, seleziona la finestra del terminale e premi CTRL+C.

La pipeline video di esempio di GStreamer ha il seguente aspetto:

USB web cam

Esegui il seguente comando per fare in modo che GStreamer rilevi automaticamente la tua videocamera USB:

```
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"
```

Per interrompere la pipeline di GStreamer, seleziona la finestra del terminale e premi CTRL+C.

Invece di lasciare che GStreamer si rilevi automaticamente, puoi utilizzarlo con un identificatore specifico del dispositivo. `v4l2src` Esegui il comando seguente:

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

Nell'output, vedrai alcuni dispositivi e l'avvio di una pipeline GStreamer su come utilizzare il dispositivo:

```
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 ! ...

```

Per interrompere la pipeline GStreamer, seleziona la finestra del terminale e premi CTRL+C.

Raspberry Pi camera module 1

Se stai usando il modulo telecamera Pi 1 o il modulo telecamera Pi 2 con, usa quanto segue:
bcm2835-v4l2

```

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"

```

Per interrompere la pipeline GStreamer, seleziona la finestra del terminale e premi CTRL+C.

Raspberry Pi camera module 2 or 3

Se stai usando lo libcamera stack moderno, usa la seguente pipeline GStreamer:

```

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"
```

Per interrompere la pipeline GStreamer, seleziona la finestra del terminale e premi CTRL+C.

Sample RTSP camera

Per questo esempio, utilizziamo per ospitare localmente un feed dimostrativo Gst-Rtsp-Server di una telecamera RTSP. Quindi costruiamo una pipeline GStreamer per caricare il feed della telecamera RTSP nel flusso video Kinesis specificato.

Per configurare sul tuo Raspberry Pi Gst-Rtsp-Server

1. Installa le librerie di dipendenze necessarie per creare il Gst-Rtsp-Server progetto. Assicurati di aver installato anche i prerequisiti del software. Digita quanto segue nel tuo terminale:

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

2. Scarica la versione 1.22 di GStreamer sul tuo Raspberry Pi.

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

3. Cambia le directory nella directory degli esempi nel gst-rtsp-server.

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

4. Compila test-launch.c in un eseguibile chiamato test-launch usando gcc.

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

5. Esegui l'eseguibile con i seguenti argomenti. Nota: GStreamer potrebbe richiedere del tempo per caricarsi per la prima volta.

```
./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"
```

Verrà visualizzato l'output seguente:

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

6. Verifica lo streaming video RTSP. Puoi usare qualsiasi visualizzatore RTSP. Ad esempio, VLC media player. Per utilizzare VLC media player per visualizzare il tuo live streaming, apri un nuovo terminale e digita:

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

per installare VLC media player. Quindi digita:

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

Dovrebbe apparire una finestra VLC con il live streaming. In caso contrario, controlla che l'eseguibile test-launch sia ancora in esecuzione e controlla l'output per eventuali errori.

Un altro modo per verificare il flusso RTSP è utilizzare l'utilità gst-discoverer-1.0. Type:

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

L'output previsto è il seguente:

```
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
```

Per inviare il feed RTSP al tuo Kinesis Video Stream usando kvssink

Questa pipeline GStreamer utilizza la connessione rtspsrc al server RTSP per recuperare il flusso video RTP. Passa i frame alrtph264depay, che estrae i frame video dai pacchetti RTP. H.264-encoded h264parseraggruppa i fotogrammi video nel formato kvssink comprensibile. kvssinkprende il flusso H.264-encoded video, lo impacchetta nel formato contenitore Matroska (MKV) e lo carica nel flusso video Kinesis.

Esegui il comando seguente:

```
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"
```

Per interrompere la pipeline GStreamer, selezionate la finestra del terminale e premete CTRL +C.

Utilizza l'hardware

Alcuni modelli Raspberry Pi sono dotati di encoder con accelerazione hardware H.264 . Puoi usarli al posto di, che è un codificatore x264enc software.

1. Assicurati che i plugin GStreamer siano installati:

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

2. Type:

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

Determina se sono disponibili i seguenti elementi:

- omxh264enc
- v4l2h264enc

Se sono disponibili, puoi usarli. Ecco alcuni esempi di pipeline che utilizzano questi elementi:

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"
```

v4l2h264encev4l2convert:

```
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"
```

Problemi di runtime

Di seguito sono riportati alcuni problemi di runtime riscontrati di frequente e come risolverli.

Nessun elemento di questo tipo «xxxxxxxxx»

Se ricevi un errore come il seguente, significa che ti manca un plugin GStreamer:

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

Risoluzione:

In base all'elemento mancante, stabilisci l'azione appropriata:

- **kvssink:** Vedi [the section called “Scaricate e compilate l'SDK per produttori C++”](#).
- **libcamerasrc:** Vedi [the section called “Errore «Attivazione del buffer pool non riuscita»”](#) per installare l'elemento libcamerasrc GStreamer.
- **omxh264encoppure: v4l2h264enc**

Segui [the section called “Prerequisiti per l'installazione del software”](#) per installare tutte le librerie GStreamer. Se le hai installate tutte e questi elementi non vengono visualizzati, significa che il tuo Raspberry Pi non ha l'hardware. Usa invece l'encoder x264enc software.

- **Altro:** Segui [the section called “Prerequisiti per l'installazione del software”](#) per installare tutte le librerie GStreamer. Diversi elementi di GStreamer si trovano nei vari gruppi di plugin GStreamer (buoni, cattivi, brutti), quindi assicurati di installarli tutti.

Errore «Attivazione del buffer pool non riuscita»

Se ricevi un errore come il seguente, significa che la pipeline utilizzata sta utilizzando v4l2src, ma dovrebbe invece libcamerasrc utilizzarla.

```
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)
```

Ad esempio, se state usando la seguente pipeline con il modulo telecamera 2 senza `libcamerasrc` installazione, potreste riscontrare questo errore quando GStreamer sta cercando di rilevare automaticamente quali elementi usare.

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

Risoluzione:

Assicuratevi che `libcamerasrc` sia installato e usalo come elemento sorgente, anziché `v4l2src`. Digita quanto segue per installare l'elemento `libcamerasrc` GStreamer:

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

Una volta `libcamerasrc` installato, se stai usando l'autovideosrcelemento, GStreamer dovrebbe passare automaticamente all'uso della fonte corretta anziché `libcamerasrc v4l2src`.

Errore del bus

Se ricevete un errore del bus poco dopo l'avvio `kvssink` (in genere, all'incirca nel momento in cui `PutMedia` viene completata la chiamata HTTP), significa che il vostro Raspberry Pi non supporta l'accesso non allineato alla memoria. I log avranno il seguente aspetto:

```
INFO Camera camera.cpp:1197 configuring streams: (0) 640x480-YUV420
INFO RPI pisp.cpp:1450 Sensor: /base/axi/pcie@1200000/rp1/i2c@880000/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 utilizza un accesso non allineato alla memoria per ottimizzare l'utilizzo della memoria, che non è supportato da tutti i dispositivi.

Risoluzione:

Per utilizzare l'SDK in modalità di accesso alla memoria allineato, è necessario impostare esplicitamente il flag `ALIGNED_MEMORY_MODEL` CMake su `ON` durante la compilazione, poiché per impostazione predefinita è `kvssink OFF`. Vedi per istruzioni più dettagliate. [the section called “Scaricate e compilate l'SDK per produttori C++”](#)

Il timestamp si blocca e la pipeline si blocca

Quando si utilizza `x264enc` in una pipeline `GStreamer`, è possibile che si verifichino situazioni in cui la timeline della pipeline rallenta in modo significativo o si blocca completamente nel giro di pochi secondi.

Ciò si verifica perché le impostazioni `x264enc` predefinite possono introdurre un'elevata latenza di codifica, che supera la capacità del buffer di input predefinito. Di conseguenza, il buffer di input si riempie, provocando il blocco degli elementi a monte e il blocco della pipeline.

Per ulteriori informazioni, consulta la documentazione di `GStreamer`. <https://gstreamer.freedesktop.org/documentation/x264/index.html?gi-language=c>

Risoluzione:

Configura `x264enc` con l'opzione di `zerolatency` ottimizzazione. Ciò riduce significativamente la latenza di codifica ottimizzando gli scenari in tempo reale, garantendo che i frame vengano elaborati e generati più rapidamente.

Configurazione di esempio:

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

Note

Sebbene questa soluzione prevenga efficacemente lo stallo della pipeline, può influire sull'efficienza e sulla qualità della codifica. Per gli scenari che richiedono bassa latenza e alta qualità, prendi in considerazione approcci alternativi, come l'utilizzo di ottimizzazioni hardware o la ricerca di una web cam che emetta H.264 direttamente l'output, saltando questa fase di codifica.

Per ulteriori informazioni, consulta [the section called “Utilizza l'hardware”](#).

Non è possibile eseguire più pipeline dallo stesso dispositivo v4l2 contemporaneamente

A dispositivi come questo `/dev/video0` è possibile accedere solo da un processo alla volta. Se più processi tentano di accedervi contemporaneamente, il secondo attende il completamento del primo.

Risoluzione:

Crea un dispositivo di loopback, che consenta a più processi di utilizzare l'interfaccia di loopback contemporaneamente. Per ulteriori informazioni, vedere Stack Exchange. <https://raspberrypi.stackexchange.com/questions/19630/take-picam-image-while-motion-is-running/19897#19897>

Errore interno del flusso di dati

Quando crei una pipeline GStreamer, colleghi gli elementi collegando il pad sorgente di un elemento al sink pad di un altro elemento. Questo processo di collegamento consente il flusso di dati dall'elemento sorgente all'elemento sink, formando una pipeline di dati.

Il messaggio di errore «Pad link failed» nel log indica che GStreamer ha riscontrato un problema nel tentativo di stabilire una connessione (link) tra i pad di due elementi nella pipeline.

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

Risoluzione:

Determina quali elementi non riescono a collegarsi tra loro. Per restringere l'ambito della pipeline, rimuovete gli elementi dalla pipeline. Sostituite l'elemento più a destra con `fakesink` e rimuovete gli elementi uno alla volta.

Potrebbe essere necessario modificare gli elementi [capsfilter, and/or modificare](#) gli elementi utilizzati dalla pipeline.

I casi più comuni sono richiedere un framerate o resolution che la fotocamera non supporta. Utilizzate `gst-device-monitor-1.0` nel terminale per ottenere i file `supportatiframerates,resolutions, eformats`. È possibile utilizzare l'elemento [Videoscale](#) GStreamer per regolare la risoluzione video e [videorate](#) per regolare la frequenza dei fotogrammi del video.

Per controllare i formati supportati per un singolo elemento GStreamer, digita nel terminale. `gst-inspect-1.0 element-name`

Riproducere contenuti multimediali dal vostro streaming video Kinesis

Apri la console [Kinesis Video Streams](#) e seleziona il nome dello stream che hai creato.

Il flusso di video inviato da Raspberry Pi viene visualizzato nella console.

Note

Potrebbero essere necessari alcuni secondi prima che il video appaia nella console.

Una volta avviato lo streaming, puoi provare le seguenti funzionalità nella console:

- Nella sezione Video preview (Anteprima video) usa i controlli di navigazione per riavvolgere o far avanzare rapidamente il flusso.
- Nella sezione Informazioni sullo streaming, controlla il codec, la risoluzione e la velocità in bit dello stream. I valori di risoluzione e bit rate sono volutamente bassi sul Raspberry Pi per ridurre al minimo l'utilizzo della larghezza di banda per questo tutorial.

Per visualizzare le CloudWatch metriche Amazon che vengono create per il tuo stream, seleziona Visualizza le metriche dello streaming in. CloudWatch

- Sotto Data retention period (Periodo di retention dei dati), nota che il flusso di video viene conservato per un giorno. Puoi modificare questo valore e impostarlo su No data retention (Nessuna retention dei dati) oppure impostare un valore compreso tra un giorno e diversi anni.
- In Server-side crittografia, nota che i tuoi dati vengono crittografati mentre sono inattivi utilizzando una chiave gestita da AWS Key Management Service (AWS KMS).

Problemi di riproduzione

Di seguito sono riportati alcuni problemi di riproduzione riscontrati di frequente e come risolverli.

Nessun file multimediale, ma nei log sono presenti degli ACK PERSISTED

Se vedi PERSISTED Acks nei log, significa che Kinesis Video Streams ha correttamente assimilato e archiviato i file multimediali caricati da. kvssink Gli ack ricevuti da Kinesis Video Streams hanno il seguente aspetto. Nel JSON, controllate il valore della chiave. "EventType"

[illegible]

Risoluzione:

Attendi uno o due minuti nella console Kinesis Video Streams, quindi usa la doppia freccia destra. Se non viene visualizzato alcun file multimediale, verifica che lo stream venga inviato nella regione corretta e controlla l'ortografia del nome dello stream. Puoi trovare queste informazioni nei log.

Vedi [the section called “Fornisci una regione a kvssink”](#) per maggiori informazioni su come kvssink determina quale regione usare.

Il caricamento del file multimediale richiede molto tempo in Console di gestione AWS

 Important

L'esperienza di riproduzione sulla console è diversa dall'esperienza di riproduzione HLS e DASH. Usa anche la pagina web [ospitata](#) nel lettore multimediale di esempio GitHub per testare la riproduzione. Il codice sorgente della pagina web è disponibile [qui](#).

I file multimediali possono essere caricati lentamente nella console a causa della scarsa larghezza di banda della rete o di un dispositivo limitato, ma possono anche essere correlati alla codifica e alla frammentazione video.

Nozioni di base sulla codifica video:

- H.264 e H.265 i codificatori utilizzano i fotogrammi chiave (I-Frames) e i frame previsti (P-Frames) per una compressione efficiente.
- Key-frames contengono dati di immagine completi, mentre contengono P-frames solo modifiche rispetto ai fotogrammi precedenti.
- L' "intervallo tra fotogrammi chiave» determina la frequenza con cui i fotogrammi chiave vengono visualizzati nel flusso video.

Frammentazione nello streaming:

- In Kinesis Video Streams, i nuovi frammenti iniziano con ogni fotogramma chiave. Per ulteriori informazioni, consulta [the section called “Modello di dati”](#).
- La lunghezza del frammento (in secondi) può essere stimata come: intervallo tra fotogrammi chiave ÷ frequenza fotogrammi

Esempio:

Per uno streaming con un intervallo tra fotogrammi chiave di 30 e una frequenza fotogrammi di 15 fps:

Lunghezza del frammento = $30 \div 15 = 2$ secondi

A causa degli intervalli di fotogrammi chiave più lunghi, i frammenti più lunghi aumentano la latenza nei contenuti multimediali in streaming.

Risoluzione:

Per migliorare i tempi di caricamento, valuta la possibilità di ridurre l'intervallo tra fotogrammi chiave. Ciò creerà frammenti più corti, diminuirà la latenza, ma aumenterà anche le dimensioni del file video.

Per l'elemento x264enc GStreamer, puoi impostare esplicitamente l'intervallo tra fotogrammi chiave tramite la proprietà: `key-int-max`

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

Quando esaminate l'output del log, notate la frequenza con cui il client di caricamento riceve gli ACK da Kinesis Video Streams. Più fotogrammi chiave vengono generati, più ACK vengono restituiti.

Il file multimediale è distorto o presenta artefatti

Per risolvere il problema, assicurati che tutti i cavi siano collegati saldamente. Controlla l'output di `libcamera-hello` (o `raspistill` per le fotocamere Pi precedenti) per i moduli delle telecamere.

Nella tua pipeline GStreamer, sostituisci `kvssink` con `autovideosink` o `and.matroskamux filesink`. Ad esempio:

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

Controlla il file di output `filesink` o il lettore multimediale che si apre durante l'uso `autovideosink` per vedere se ci sono anche degli artefatti.

Esamina anche l'output della seguente pipeline:

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

L'aggiunta di elementi alla pipeline, come [dewarp](#), può correggere le uscite delle telecamere fish eye.

Esamina i codec di output supportati per la tua fotocamera e regola gli elementi secondo necessità.

Ad esempio, se la tua fotocamera USB supporta solo l'output JPEG, dovrai utilizzare `jpegdec` gli elementi `jpegparse` e `and` per trasformare il file multimediale prima di codificarlo in uso. H.264 x264enc. Cerca assistenza nei forum di GStreamer per altri utenti con pipeline `and/or` simili e configurazioni di webcam.

Risoluzione dei problemi di compilazione su C++ producer SDK per Raspberry Pi

Se riscontri un problema di compilazione e desideri provare diversi argomenti CMake, assicurati di eseguire una build pulita. Elimina `open-source dependency` le `build` cartelle e prima di riprovare.

Crea problemi con OpenSSL

Se ricevi un risultato simile al seguente, indica che OpenSSL ha rilevato erroneamente l'architettura del tuo sistema.

```
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 questo esempio, sta tentando di creare una versione a 64 bit (linux-aarch64) quando questo Raspberry Pi è effettivamente a 32 bit. Alcuni dispositivi Raspberry Pi hanno un kernel a 64 bit, ma uno spazio utente a 32 bit.

Determina per quale architettura OpenSSL sta cercando di creare. Puoi trovare la riga di registro durante la configure fase di 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
```

Verifica l'architettura del tuo sistema:

- Esamina il bit-ness del kernel: esegui `uname -m`
- Controlla lo spazio utente bit-ness: run `getconf LONG_BIT`

Puoi anche rivedere le informazioni sulla CPU con `cat /proc/cpuinfo` i nostri comandi. `lscpu`

Risoluzione:

Per risolvere questo problema, aggiungi il seguente argomento CMake durante la compilazione, per assicurarti che OpenSSL venga compilato correttamente per l'architettura ARM a 32 bit:

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

Risolvi i problemi **di caricamento di kvssink in GStreamer**

Conferma `GST_PLUGIN_PATH`

Assicurati che la variabile di ambiente `GST_PLUGIN_PATH` nella sessione di shell corrente punti alla directory che contiene `kvssink`. Le variabili di ambiente sono specifiche della sessione, quindi dovrai impostarle per ogni nuova sessione. Per rendere permanente questa modifica, vedi «Aggiorna lo script di avvio della shell per includere l'impostazione della variabile di ambiente `GST_PLUGIN_PATH`».

Errore: impossibile aprire il file oggetto condiviso: nessun file o directory di questo tipo

Se riscontri l'errore `Cannot open shared object file: No such file or directory`, esegui il comando seguente:

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

Se ricevete il seguente output, indica che il linker dinamico non è in grado di individuare le librerie richieste per `kvssink`. Ciò si verifica in genere a causa di:

- `kvssink` Trasferimento in un luogo diverso da quello in cui è stato costruito.
- Cross-compiling per l'architettura della CPU sbagliata.
- Manca una dipendenza richiesta.

Output:

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

Risoluzione:

Per le librerie spostate, aggiungete la cartella contenente le librerie mancanti a `LD_LIBRARY_PATH`.

Dalla directory principale del repository originale, è possibile individuare la libreria mancante utilizzando l'`find` utilità. Nel terminale, digita:

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

Output:

```
./build/dependency/libkvsproducer/kvsproducer-src/libcproducer.so
```

Il separatore del percorso dei file sui dispositivi Linux è: . Il comando seguente aggiunge un nuovo percorso di cartella alla variabile di LD_LIBRARY_PATH ambiente esistente, preservando i valori precedenti.

Nel terminale, digita:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/build/dependency/libkvsproducer/
kvsproducer-src
```

Important

Le variabili di ambiente sono specifiche della sessione. Per rendere permanenti le modifiche tra le sessioni, modifica lo script di avvio della shell.

Potrebbe essere necessario aggiungerlo anche open-source/local/lib al tuo\$LD_LIBRARY_PATH.

Errore: ./path/to/libcproducer.so.1: intestazione ELF non valida

Se ricevete questo errore durante il caricamento delle librerie condivise, potrebbe essere dovuto a collegamenti simbolici non funzionanti (). symlinks I collegamenti simbolici possono interrompersi se il sistema operativo del computer host non corrisponde a quello del computer di destinazione. Ad esempio, la compilazione incrociata su un Raspberry MacBook Pi.

Un'altra possibile causa è che i file binari compilati erano per l'architettura sbagliata. Ad esempio, se i file binari sono stati compilati per x86 (Raspberry Pi utilizza CPU ARM).

Vai alla posizione della libreria specificata nell'errore e digita: ls -la per ispezionare la libreria. symlinks

Risposta:

```
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.1.1.1.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.1.1.1.so
drwxr-xr-x  6 me  staff    192 Sep 10 17:16 pkgconfig

```

Nell'output di esempio sopra riportato, non symlinks sono interrotti. Broken symlinks non avrà frecce puntate verso i bersagli.

Risoluzione:

Ci sono due opzioni per correggere i collegamenti simbolici:

- Consigliato: ricreare il comando `symlink` with the `ln`. Type:

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

- Copia il file di libreria effettivo e rinominalo in modo che corrisponda a. `symlink`

Note

Questa opzione comporta un maggiore utilizzo dello spazio di archiviazione.

Come best practice, è consigliabile compilare sullo stesso sistema operativo utilizzando strumenti come Docker per evitare problemi di compilazione incrociata.

Dipendenze mancanti:

Se il nome della libreria mancante inizia con `libkvs`, consulta la sezione «librerie spostate» sopra per installare le librerie Kinesis Video Streams dal dispositivo host al dispositivo di destinazione.

Altrimenti, segui questa procedura [the section called “Prerequisiti per l'installazione del software”](#) per assicurarti che tutti i prerequisiti del software open source siano installati sul dispositivo di destinazione.

Riferimento al codice di errore

Questa sezione contiene informazioni sui codici di errore e stato delle [Caricamento su Kinesis Video Streams](#).

Per informazioni sulle soluzioni ai problemi comuni, consulta [Risoluzione dei problemi](#).

Argomenti

- [Errori e codici di stato restituiti da PutFrame Callbacks - Platform Independent Code \(PIC\)](#)
- [Errori e codici di stato restituiti dai PutFrame callback - C producer library](#)

Errori e codici di stato restituiti da PutFrame Callbacks - Platform Independent Code (PIC)

Le seguenti sezioni contengono informazioni sugli errori e sullo stato restituite dai callback per l'PutFrameoperazione all'interno del Platform Independent Code (PIC).

Argomenti

- [Codici di errore e di stato restituiti dalla libreria client](#)
- [Codici di errore e di stato restituiti dalla libreria di durata](#)
- [Codici di errore e di stato restituiti dalla libreria comune](#)
- [Codici di errore e di stato restituiti dalla libreria heap](#)
- [Codici di errore e di stato restituiti dalla libreria MKVGen](#)
- [Codici di errore e di stato restituiti dalla libreria Trace](#)
- [Codici di errore e di stato restituiti dalla libreria Utils](#)
- [Codici di errore e di stato restituiti dalla libreria View](#)

Codici di errore e di stato restituiti dalla libreria client

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della libreria Kinesis Client Video Streams.

Codice	Messaggio	Description	Azione consigliata
0x52000001	STATUS_MAX_STREAM_COUNT	È stato raggiunto il numero massimo di flussi.	Specificare un numero massimo di flussi più elevato in DeviceInfo come specificato in Quote SDK di Producer .
0x52000002	STATUS_MIN_STREAM_COUNT	Errore numero minimo di flussi.	Specificate il numero massimo di stream superiore a zero. DeviceInfo
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	Lunghezza del nome del dispositivo non valida.	Fai riferimento alla lunghezza massima del nome del dispositivo in caratteri specificata in Quote SDK di Producer .
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	Versione della struttura DeviceInfo non valida.	Specificare la versione corrente corretta della struttura.
0x52000005	STATUS_MAX_TAG_COUNT	È stato raggiunto il numero massimo di tag.	Fai riferimento al numero massimo di tag corrente specificato in Quote SDK di Producer .
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	Versione della struttura Callbacks non valida.	Specificare la versione corrente corretta della struttura.

Codice	Messaggio	Description	Azione consigliata
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	Versione della struttura StreamInfo non valida.	Specificare la versione corrente corretta della struttura.
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	Lunghezza del nome del flusso non valida.	Fai riferimento alla lunghezza massima del nome dello stream in caratteri specificata in Quote SDK di Producer .
0x5200000a	STATUS_INVALID_STORAGE_SIZE	È stata specificata una dimensione di storage non valida.	La dimensione di storage in byte non deve superare i limiti specificati in Quote SDK di Producer .
0x5200000b	STATUS_INVALID_DIRECTORY_LENGTH	Lunghezza della stringa della directory root non valida.	Fate riferimento alla lunghezza massima del percorso della directory principale specificata in Quote SDK di Producer .
0x5200000c	STATUS_INVALID_SPILL_RATIO	Rapporto di spill non valido.	Esprimete il rapporto di fuoriuscita in percentuale da 0 a 100.
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Versione della struttura StorageInfo non valida.	Specificare la versione corrente corretta della struttura.

Codice	Messaggio	Description	Azione consigliata
0x5200000e	STATUS_INVALID_STREAM_STATE	Lo stato attuale del flusso non consente l'operazione corrente.	In genere, questo errore si verifica quando l'SDK non riesce a raggiungere lo stato richiesto per eseguire l'operazione richiesta. Ad esempio, si verifica se la chiamata API <code>GetStreamingEndpoint</code> non riesce e se l'applicazione client la ignora e continua a inserire i fotogrammi nel flusso.
0x5200000f	STATUS_SERVICE_CALLBACK_MISSING	Nella struttura <code>Callbacks</code> non sono presenti dei punti di accesso della funzione di alcune funzioni obbligatorie.	Verificate che i callback obbligatori siano implementati nell'applicazione client. Questo errore è esposto solo ai client Platform Independent Code (PIC). C++ e altri wrapper di livello più elevato sono in grado di soddisfare queste chiamate.

Codice	Messaggio	Description	Azione consigliata
0x52000010	STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR	Non autorizzato.	Verifica il token di sicurezza, il certificato, l'integrazione del token di sicurezza e la scadenza. Verifica che al token siano associati i diritti corretti. Per le applicazioni di esempio Kinesis Video Streams, verifica che la variabile di ambiente sia impostata correttamente.
0x52000011	STATUS_DESCRIBE_STREAM_CALL_FAILED	Errore dell'API DescribeStream .	Questo errore viene restituito dopo che il nuovo tentativo dell'API DescribeStream ha restituito un errore. Il client PIC restituisce questo errore dopo aver smesso di riprovare.
0x52000012	STATUS_INVALID_DESCRIBE_STREAM_RESPONSE	Struttura DescribeStreamResponse non valida.	La struttura inviata al DescribeStreamResultEvent è nulla o contiene elementi non validi, come un Amazon Resource Name (ARN) nullo.

Codice	Messaggio	Description	Azione consigliata
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	Il flusso è in fase di eliminazione.	Il flusso in fase di eliminazione ha causato un errore dell'API. Verifica che nessun altro processo stia tentando di eliminare lo stream mentre lo stream è in uso.
0x52000014	STATUS_SERVICE_CALL_INVALID_ARG_ERROR	Sono stati specificati degli argomenti non validi per la chiamata di servizio.	Il backend restituisce questo errore quando un argomento della chiamata di servizio non è valido o quando l'SDK rileva un errore che non può interpretare.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	Il dispositivo non è stato trovato.	Verifica che il dispositivo non venga eliminato mentre è in uso.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	Non è stato effettuato il provisioning del dispositivo.	Verifica che il dispositivo sia stato fornito.

Codice	Messaggio	Description	Azione consigliata
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	Risorsa generica non trovata restituita dal servizio.	Questo errore si verifica quando il servizio non è in grado di individuare la risorsa (ad esempio, un flusso). Tale errore potrebbe avere più significati in contesti diversi, ma la sua causa più probabile è l'utilizzo delle API prima della creazione del flusso. L'utilizzo dell'SDK conferma che lo stream viene creato per primo.
0x52000018	STATUS_INVALID_AUTH_LEN	Lunghezza delle informazioni di autorizzazione non valida.	Fare riferimento ai valori correnti specificati in Quote SDK di Producer .
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	La chiamata dell'API <code>CreateStream</code> non è riuscita.	Fare riferimento alla stringa di errore per ulteriori informazioni dettagliate sul motivo della mancata riuscita dell'operazione.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	La chiamata <code>GetStreamingToken</code> non è riuscita.	Fare riferimento alla stringa di errore per ulteriori informazioni dettagliate sul motivo della mancata riuscita dell'operazione.

Codice	Messaggio	Description	Azione consigliata
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	La chiamata dell'API GetStreamingEndpoint non è riuscita.	Fare riferimento alla stringa di errore per ulteriori informazioni dettagliate sul motivo della mancata riuscita dell'operazione.
0x5200002c	STATUS_INVALID_URI_LEN	L'API GetStreamingEndpoint ha restituito una lunghezza della stringa URI non valida.	Fare riferimento ai valori massimi correnti specificati in Quote SDK di Producer .
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	La chiamata dell'API PutMedia non è riuscita.	Fare riferimento alla stringa di errore per ulteriori informazioni dettagliate sul motivo della mancata riuscita dell'operazione.

Codice	Messaggio	Description	Azione consigliata
0x5200002e	STATUS_STORE_OUT_OF_MEMORY	La memoria dello store dei contenuti è esaurita.	Lo store dei contenuti viene condiviso tra i flussi e deve disporre di capacità sufficiente per l'archiviazione della durata massima di tutti i flussi più di un altro 20% circa di spazio (per la deframmentazione). È importante non superare la capacità di storage. Scegliere i valori di durata massima per flusso corrispondenti alla dimensione di storage cumulativa e alle tolleranze di latenza. Consigliamo di eliminare i frame non appena escono dalla finestra di visualizzazione dei contenuti anziché semplicemente inserirli (pressione della memoria di archiviazione dei contenuti). Questo perché l'eliminazione dei frame avvia i richiami di notifica della pressione del flusso. L'applicazione è in grado di regolare i componenti multimediali di upstream (come il codificatore) per comprimere il bitrate, eliminare i fotogrammi

Codice	Messaggio	Description	Azione consigliata
			i o adottare le misure necessarie.
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	Al momento non sono disponibili altri dati per un flusso.	Questo è un risultato potenzialmente valido quando la pipeline multimediale produce i frame da inviare al servizio più lentamente di quanto il thread di rete consuma. Higher-level i client (ad esempio, C++, Java o Android) non visualizzano questo avviso perché è gestito internamente.
0x52000030	STATUS_INVALID_TAG_VERSION	Versione della struttura Tag non valida.	Specificare la versione corrente corretta della struttura.
0x52000031	STATUS_SERVICE_CALL_UNKNOWN_ERROR	È stato restituito un errore sconosciuto o generico dallo stack di rete.	Consultare i log per ulteriori informazioni dettagliate.
0x52000032	STATUS_SERVICE_CALL_RESOURCE_IN_USE_ERROR	Risorsa in uso.	Restituito dal servizio. Per ulteriori informazioni, consulta la Guida di riferimento all'API Kinesis Video Streams.

Codice	Messaggio	Description	Azione consigliata
0x52000033	STATUS_SERVICE_CLIENT_LIMIT_ERROR	Limite del client.	Restituito dal servizio. Per ulteriori informazioni, consulta la Guida di riferimento all'API Kinesis Video Streams.
0x52000034	STATUS_SERVICE_DEVICE_LIMIT_ERROR	Limite del dispositivo.	Restituito dal servizio. Per ulteriori informazioni, consulta la Guida di riferimento all'API Kinesis Video Streams.
0x52000035	STATUS_SERVICE_STREAM_LIMIT_ERROR	Limite del flusso.	Restituito dal servizio. Per ulteriori informazioni, consulta la Guida di riferimento all'API Kinesis Video Streams.
0x52000036	STATUS_SERVICE_RESOURCE_DELETED_ERROR	La risorsa è stata eliminata o è in fase di eliminazione.	Restituito dal servizio. Per ulteriori informazioni, consulta la Guida di riferimento all'API Kinesis Video Streams.
0x52000037	STATUS_SERVICE_TIMEOUT_ERROR	La chiamata di servizio è scaduta.	La chiamata a una determinata API del servizio è risultata in un timeout. Verifica di disporre di una connessione di rete valida. Il PIC ritenterà l'operazione automaticamente.

Codice	Messaggio	Description	Azione consigliata
0x52000038	STATUS_STREAM_READ_Y_CALLBACK_FAILED	Notifica di flusso pronto all'uso.	Questa notifica viene inviata dal PIC al client per indicare che il flusso asincrono è stato creato.
0x52000039	STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	Sono stati specificati dei tag non validi.	Il numero di tag non è zero, ma i tag sono vuoti. Verifica che i tag siano specificati o che il conteggio sia zero.
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	Versione della struttura StreamDescription non valida.	Specificare la versione corrente corretta della struttura.
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	Lunghezza del nome del tag non valida.	Fare riferimento ai limiti relativi al nome del tag specificati in Quote SDK di Producer .
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	Lunghezza del valore del tag non valida.	Fare riferimento ai limiti relativi al valore del tag specificati in Quote SDK di Producer .
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	Errore dell'API TagResource .	La chiamata dell'API TagResource non è riuscita. Controllare che la connessione di rete sia attiva. Consultare i log per ulteriori informazioni sull'errore.

Codice	Messaggio	Description	Azione consigliata
0x5200003e	STATUS_INVALID_CUSTOM_DATA	Dati personalizzati non validi durante la chiamata alle API del PIC.	Sono stati specificati dei dati personalizzati non validi in una chiamata alle API del PIC. Questo errore può verificarsi soltanto per i client che utilizzano direttamente il PIC.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	Struttura CreateStreamResponse non valida.	La struttura o i relativi campi dei membri non sono validi (ovvero, l'ARN è nullo o di dimensioni superiori rispetto a quelle specificate in Quote SDK di Producer).
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	Autenticazione del client non riuscita.	Il PIC non è riuscito a ottenere le informazioni di autenticazione corrette (AccessKeyId o SecretAccessKey) dopo diversi tentativi. Verificare l'integrazione dell'autenticazione. Le applicazioni di esempio utilizzano le variabili di ambiente per inviare le informazioni sulle credenziali alla libreria del producer C++.

Codice	Messaggio	Description	Azione consigliata
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	Errore di ricezione della chiamata al token di sicurezza.	Questa situazione può verificarsi per i client che utilizzano direttamente il PIC. Dopo un determinato numero di tentativi, la chiamata non riesce con questo errore.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	Errore di provisioning.	Il provisioning non è implementato.
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	Impossibile creare il client del producer.	Un errore generico restituito dal PIC dopo un determinato numero di tentativi quando la creazione del client non riesce.
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	Non è stato possibile impostare il client del producer sullo stato READY.	Restituito dal computer di stato PIC se il PIC non riesce a passare allo stato READY. Consultare i log per ulteriori informazioni sulla causa principale.
0x52000045	STATUS_TAG_CLIENT_CALL_FAILED	Errore di TagResource per il client del producer.	Errore della chiamata API TagResource per il client del producer. Consultare i log per ulteriori informazioni sulla causa principale.

Codice	Messaggio	Description	Azione consigliata
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	Device/Producer creazione non riuscita.	Gli SDK di livello superiore (ad esempio, C++ o Java) non implementano ancora l'API di creazione del dispositivo o del produttore. I client che utilizzano o direttamente il PIC possono indicare un errore tramite la notifica sui risultati.
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	Il timestamp dell'ACK di ricezione non si trova nella visualizzazione.	Questo errore si verifica se il fotogramma corrispondente all'ACK di ricezione esce dalla finestra della visualizzazione dei contenuti. In genere, ciò si verifica se la consegna dell'ACK è lenta. Può essere interpretato come un avviso e un'indicazione delle prestazioni lente del downlink.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	Versione della struttura FragmentAck non valida.	Specificare la versione corrente corretta della struttura FragmentAck.

Codice	Messaggio	Description	Azione consigliata
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	Scadenza del token di sicurezza non valida.	La scadenza del token di sicurezza dovrebbe avere un timestamp assoluto nel futuro superiore a quello attuale, con un periodo di grazia. Per i limiti relativi al periodo di tolleranza, consulta Quote SDK di Producer .
0x5200004a	STATUS_END_OF_STREAM	Indicatore di fine del flusso (EOS).	Nel chiamata dell'API <code>GetStreamData</code> , indica che la sessione di gestione dell'upload attuale è terminata. Ciò si verifica se la sessione termina o produce degli errori oppure se il token della sessione è scaduto e la sessione viene ruotata.
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	Nome di flusso duplicato.	Più flussi non possono avere lo stesso nome. Scegliere un nome univoco per il flusso.
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	Periodo di conservazione non valido.	Un periodo di conservazione non valido è stato specificato nella struttura <code>StreamInfo</code> . Per informazioni sull'intervallo valido di valori per il periodo di conservazione, consulta Quote SDK di Producer .

Codice	Messaggio	Description	Azione consigliata
0x5200004d	STATUS_IN VALID_ACK _KEY_START	FragmentAck non valido.	Impossibile analizzare e la stringa ACK del frammento. Indicatore di inizio della chiave non valido. La stringa ACK del frammento potrebbe essere danneggiata. Può correggersi autonomam ente e questo errore può essere trattato come un avviso.
0x5200004e	STATUS_IN VALID_ACK _DUPLICAT E_KEY_NAME	FragmentAck non valido.	Impossibile analizzare e la stringa ACK del frammento. Più chiavi hanno lo stesso nome. La stringa ACK del frammento potrebbe essere danneggiata. Può correggersi autonomam ente e questo errore può essere trattato come un avviso.

Codice	Messaggio	Description	Azione consigliata
0x5200004f	STATUS_INVALID_ACK_VALUE_START	FragmentAck non valido.	Impossibile analizzare e la stringa ACK del frammento a causa di un indicatore di inizio del valore di chiave non valido. La stringa ACK del frammento potrebbe essere danneggiata. Può correggersi autonomamente e questo errore può essere trattato come un avviso.
0x52000050	STATUS_INVALID_ACK_VALUE_END	FragmentAck non valido.	Impossibile analizzare e la stringa ACK del frammento a causa di un indicatore di fine del valore di chiave non valido. La stringa ACK del frammento potrebbe essere danneggiata. Può correggersi autonomamente e questo errore può essere trattato come un avviso.
0x52000051	STATUS_INVALID_ACK_TYPE	FragmentAck non valido.	Impossibile analizzare e la stringa ACK del frammento perché è stato specificato un tipo di ACK non valido.

Codice	Messaggio	Description	Azione consigliata
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	Il flusso è stato interrotto.	Il flusso è stato interrotto, ma un fotogramma viene comunque inserito nel flusso.
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Versione della struttura <code>StreamMetrics</code> non valida.	Specificare la versione corrente corretta della struttura <code>StreamMetrics</code> .
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Versione della struttura <code>ClientMetrics</code> non valida.	Specificare la versione corrente corretta della struttura <code>ClientMetrics</code> .
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	L'inizializzazione del producer non ha raggiunto uno stato <code>READY</code> .	Impossibile raggiungere lo stato <code>READY</code> durante l'inizializzazione del client del producer. Consultare i log per ulteriori informazioni.
0x52000056	STATUS_MACHINE_STATE_NOT_FOUND	Errore interno del computer di stato.	Non è un errore visibile pubblicamente.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	Un tipo di <code>ACK</code> non valido è stato specificato nella struttura <code>FragmentAck</code> .	La struttura <code>FragmentAck</code> deve contenere i tipi di <code>ACK</code> definiti nell'istanza pubblica.
0x52000058	STATUS_INVALID_STREAM_READY_STATE	Errore di transizione interno del computer di stato.	Non è un errore visibile pubblicamente.

Codice	Messaggio	Description	Azione consigliata
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	L'oggetto di flusso è stato liberato in seguito alla liberazione del producer.	Si è tentato di liberare un oggetto di flusso dopo la liberazione dell'oggetto del producer. Ciò può verificarsi soltanto nei client che utilizzano direttamente il PIC.
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	Errore interno di storage.	Un errore interno che indica che la dimensione e di allocazione effettiva dall'archivio di contenuti è inferiore alla dimensione del frame e del frammento impacchettati.
0x5200005b	STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION	Errore interno di storage.	La dimensione archiviata dell'allocazione nella vista dei contenuti è superiore alla dimensione di allocazione nello store dei contenuti.
0x5200005c	STATUS_ACK_ERROR_STREAM_READ_ERROR	ACK di errore di lettura del flusso.	Un errore restituito dall'ACK dal backend che indica un errore di lettura o analisi dello stream. Ciò si verifica generalmente quando il backend non riesce a recuperare lo stream. Auto-restreaming di solito può correggere questo errore.

Codice	Messaggio	Description	Azione consigliata
0x5200005d	STATUS_ACK_ERR_FRAGMENT_SIZE_REACHED	La dimensione e massima del frammento è stata raggiunta.	La dimensione massima del frammento in byte è definita in Quote SDK di Producer . Questo errore indica che sono presenti fotogrammi di dimensioni molto grandi o che non è presente nessun fotogramma chiave per la creazione di frammenti di dimensione gestibile. Controllate le impostazioni dell'encoder e verificate che i keyframe vengano prodotti correttamente. Per i flussi con densità molto elevata, configurare il codificatore sulla creazione di frammenti in durate più ridotte per gestire la dimensione massima.

Codice	Messaggio	Description	Azione consigliata
0x5200005e	STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED	La durata massima del frammento è stata raggiunta.	La durata massima del frammento viene definita in Quote SDK di Producer . Questo errore indica che sono presenti fotogrammi molto lenti al secondo o che non è presente nessun fotogramma chiave per la creazione di frammenti di durata gestibile. Controllate le impostazioni dell'encoder e verificate che i keyframe vengano prodotti correttamente a intervalli regolari.
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	La durata massima della connessione è stata raggiunta.	Kinesis Video Streams impone la durata massima della connessione come specificato in Quote SDK di Producer . L'SDK Producer ruota automaticamente lo stream o il token prima che venga raggiunto il massimo. I client che utilizzano l'SDK non dovrebbero ricevere questo errore.
0x52000060	STATUS_ACK_ERR_FRAME_TIMESTAMP_NOT_MONOTONIC	I timecode non aumentano in maniera monotona.	L'SDK Producer impone i timestamp, quindi i client che utilizzano l'SDK non dovrebbero ricevere questo errore.

Codice	Messaggio	Description	Azione consigliata
0x52000061	STATUS_AC K_ERR_MUL TI_TRACK_MKV	Tracce multiple in MKV.	L'SDK Producer impone flussi a traccia singola, quindi i client che utilizzano l'SDK non dovrebbero ricevere questo errore.
0x52000062	STATUS_AC K_ERR_INV ALID_MKV_DATA	Dati MKV non validi.	Il parser MKV di back-end ha rilevato un errore durante l'analisi del flusso. I client che utilizzano l'SDK potrebbero riscontrare questo errore se lo stream viene danneggiato durante la transizione. Ciò può verificarsi anche se le pressioni del buffer costringono l'SDK a eliminare i frame di coda che vengono parzialmente trasmessi. In quest'ultimo caso, ti consigliamo di ridurre l'FPS e la risoluzione, aumentare il rapporto di compressione o (in caso di rete «a intermittenza») aumentare la durata dell'archivio dei contenuti e del buffer per far fronte alle pressioni temporanee.

Codice	Messaggio	Description	Azione consigliata
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	Timestamp del producer non valido.	Il servizio restituisce questo errore ACK se l'orologio del produttore ha una grande deviazione verso il futuro. Higher-level Gli SDK (ad esempio, Java o C++) utilizzano alcune versioni dell'orologio di sistema per soddisfare la richiamata dell'ora corrente da PIC. Verifica che l'orologio di sistema sia impostato correttamente. I client che utilizzano direttamente il PIC devono verificare che le loro funzioni di callback restituiscano il timestamp corretto.
0x52000064	STATUS_ACK_STREAM_NOT_ACTIVE	Flusso non attivo.	È stata effettuata una chiamata all'API di back-end mentre il flusso non era in stato "Attivo". Ciò si verifica quando il client crea il flusso e continua immediatamente a inserirvi i fotogrammi. L'SDK gestisce questo scenario attraverso il computer di stato e il meccanismo di recupero.

Codice	Messaggio	Description	Azione consigliata
0x52000065	STATUS_AK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS errore di accesso negato.	Restituito quando l'account non dispone dell'accesso alla chiave specificata.
0x52000066	STATUS_AK_ERR_KMS_KEY_DISABLED	AWS KMS la chiave è disabilitata.	La chiave specificata è stata disabilitata.
0x52000067	STATUS_AK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS errore di convalida della chiave.	Errore di convalida generico. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Key Management Service .
0x52000068	STATUS_AK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key non disponibile.	La chiave non è disponibile. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Key Management Service .
0x52000069	STATUS_AK_ERR_KMS_KEY_INVALID_USAGE	Uso non valido della chiave KMS.	Non AWS KMS key è configurato per essere utilizzato in questo contesto. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Key Management Service .

Codice	Messaggio	Description	Azione consigliata
0x5200006a	STATUS_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS stato non valido.	Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Key Management Service .
0x5200006b	STATUS_ACK_ERR_KMS_KEY_NOT_FOUND	Chiave KMS non trovata.	La chiave non è stata trovata. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API di AWS Key Management Service .
0x5200006c	STATUS_ACK_ERR_STREAM_DELETED	Il flusso è stato eliminato o è in fase di eliminazione.	Il flusso è in fase di eliminazione da un'altra applicazione o tramite Console di gestione AWS.
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	Errore interno.	Errore interno generico del servizio.
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR	Errore di archiviazione del frammento.	Restituito quando il servizio non riesce a mantenere in modo durevole e indicizzare il frammento. Anche se è raro, questo errore può verificarsi per diversi motivi. Per impostazione predefinita, l'SDK riprova a inviare il frammento.

Codice	Messaggio	Description	Azione consigliata
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	Errore sconosciuto.	Il servizio ha restituito un errore sconosciuto.
0x52000070	STATUS_MISSING_ERR_ACK_ID	Informazioni ACK mancanti.	Il parser ACK ha completato l'analisi, ma l'informazione FragmentAck è mancante.
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	Lunghezza del segmento ACK non valida.	Nel parser ACK è stata specificata una stringa di segmento ACK con lunghezza non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x52000074	STATUS_MAX_FRAGMENT_METADATA_COUNT	Il numero massimo di elementi metadati è stato aggiunto a un frammento.	Uno stream video Kinesis può aggiungere fino a 10 elementi di metadati a un frammento, aggiungendo un elemento non persistente a un frammento o aggiungendo un elemento persistente alla coda dei metadati. Per ulteriori informazioni, consulta Utilizzo dei metadati di streaming con Kinesis Video Streams .

Codice	Messaggio	Description	Azione consigliata
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	È stato raggiunto un limite (numero massimo di metadati, lunghezza del nome o del valore dei metadati) .	Producer SDK limita il numero e la dimensione degli elementi metadati. Questo errore non si verifica a meno che i limiti nel codice Producer SDK non vengano modificati. Per ulteriori informazioni, consulta Utilizzo dei metadati di streaming con Kinesis Video Streams .
0x52000076	STATUS_BLOCKING_PUT_INTERRUPTED_STREAM_TERMINATED	Non ancora disponibile.	
0x52000077	STATUS_INVALID_METADATA_NAME	Il nome dei metadati non è valido.	Il nome dei metadati non può iniziare con la stringa "»AWS. Se si verifica questo errore, l'elemento di metadati non viene aggiunto al frammento o alla coda dei metadati. Per ulteriori informazioni, consulta Utilizzo dei metadati di streaming con Kinesis Video Streams .
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	Il termine di un frammento frame è in uno stato non valido.	La fine del frammento non deve essere inviata in un flusso frammentato non basato su fotogrammi chiave.

Codice	Messaggio	Description	Azione consigliata
0x52000079	STATUS_TRACK_INFO_MISSING	Le informazioni sulle tracce sono mancanti.	Il numero della traccia deve essere maggiore di zero e corrispondere all'ID della traccia.
0x5200007a	STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED	È stato superato il conteggio massimo di tracce.	Puoi avere un massimo di tre tracce per stream.
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	Il tempo di conservazione della modalità di streaming offline è impostato su zero.	Il tempo di conservazione della modalità streaming offline non deve essere impostato su zero.
0x5200007c	STATUS_ACK_TRACK_NUMBER_MISMATCH	Il numero di traccia dell'errore ACK non corrisponde.	
0x5200007d	STATUS_ACK_FRAME_MISSING_FOR_TRACK	Frame mancanti per una traccia.	
0x5200007e	STATUS_ACK_ERROR_THAN_ALLOWED_TRACKS_FOUND	Il numero massimo di tracce consentito è stato superato.	
0x5200007f	STATUS_UPLOAD_HANDLE_ABORTED	L'handle di caricamento è stata interrotta.	

Codice	Messaggio	Description	Azione consigliata
0x52000080	STATUS_INVALID_CERT_PATH_LENGTH	Lunghezza percorso certificato non valida.	
0x52000081	STATUS_DUPLICATE_TRACK_ID_FOUND	Trovato ID traccia duplicato.	
0x52000082	STATUS_INVALID_CLIENT_INFO_VERSION		
0x52000083	STATUS_INVALID_ID_STRING_LENGTH		
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		

Codice	Messaggio	Description	Azione consigliata
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_EOF		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_IMAGE_METADATA_KEY_LENGTH		

Codice	Messaggio	Description	Azione consigliata
0x5200008f	STATUS_IN VALID_IMA GE_METADA TA_VALUE_LENGTH		

Codici di errore e di stato restituiti dalla libreria di durata

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della `Duration` libreria.

Codice	Messaggio
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

Codici di errore e di stato restituiti dalla libreria comune

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della `Common` libreria.

Note

Le informazioni sui codici di errore e di stato sono comuni a molte API.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x00000001	0x1	STATUS_NULL_ARG	Valore NULL inviato per un argomento obbligatorio.
0x00000002	0x2	STATUS_IN VALID_ARG	È stato specificato un valore non valido per un argomento.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x00000003	0x3	STATUS_INVALID_ARG_LEN	È stata specificata una lunghezza di argomento non valida.
0x00000004	0x4	STATUS_NOT_ENOUGH_MEMORY	Impossibile allocare memoria sufficiente.
0x00000005	0x5	STATUS_BUFFER_TOO_SMALL	La dimensione specificata del buffer è troppo piccola.
0x00000006	0x6	STATUS_UNEXPECTED_EOF	È stata raggiunta una fine imprevista del file.
0x00000007	0x7	STATUS_FORMAT_ERROR	È stato rilevato un formato non valido.
0x00000008	0x8	STATUS_INVALID_HANDLE_ERROR	Valore di gestione non valido.
0x00000009	0x9	STATUS_OPEN_FILE_FAILED	Impossibile aprire un file.
0x0000000a	0xa	STATUS_READ_FILE_FAILED	Impossibile leggere da un file.
0x0000000b	0xb	STATUS_WRITE_TO_FILE_FAILED	Impossibile scrivere su un file.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x0000000c	0xc	STATUS_INTERNAL_ERROR	Un errore interno che in genere non si verifica e che può indicare un bug dell'SDK o dell'API del servizio.
0x0000000d	0xd	STATUS_INVALID_OPERATION	Si è verificata un'operazione non valida o l'operazione non è consentita.
0x0000000e	0xe	STATUS_NOT_IMPLEMENTED	La funzione non viene implementata.
0x0000000f	0xf	STATUS_OPERATION_TIMED_OUT	Timeout dell'operazione.
0x00000010	0x10	STATUS_NOT_FOUND	Una risorsa obbligatoria non è stata trovata.
0x00000011	0x11	STATUS_CREATE_THREAD_FAILED	Impossibile creare un thread.
0x00000012	0x12	STATUS_THREAD_NOT_ENOUGH_RESOURCES	Risorse insufficienti per creare un altro thread oppure è stato rilevato un limite imposto dal sistema al numero di thread.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x00000013	0x13	STATUS_THREAD_READ_INVALID_ARG	Sono stati specificati attributi del thread non validi o un altro thread è già in attesa di unirsi a questo thread.
0x00000014	0x14	STATUS_THREAD_READ_PERMISSIONS	Nessuna autorizzazione per impostare la politica di pianificazione e i parametri specificati negli attributi del thread.
0x00000015	0x15	STATUS_THREAD_READ_DEADLOCKED	È stato rilevato un deadlock o il thread di unione specifica il thread chiamante.
0x00000016	0x16	STATUS_THREAD_READ_DOES_NOT_EXIST	Impossibile trovare il thread con l'ID del thread specificato.
0x00000017	0x17	STATUS_JOIN_THREAD_FAILED	È stato restituito un errore sconosciuto o generico dall'operazione di unione del thread.
0x00000018	0x18	STATUS_WAIT_FAILED	È stato superato il tempo massimo di attesa della variabile condizionale.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x00000019	0x19	STATUS_CANCEL_THREAD_FAILED	È stato restituito un errore sconosciuto o generico dall'operazione di annullamento del thread.
0x0000001a	0x1a	STATUS_THREAD_IS_NOT_JOINABLE	L'operazione di unione dei thread è richiesta su un thread non unibile.
0x0000001b	0x1b	STATUS_DETACH_THREAD_FAILED	È stato restituito un errore sconosciuto o generico dall'operazione di distacco del thread.
0x0000001c	0x1c	STATUS_THREAD_ATTR_INIT_FAILED	Impossibile inizializzare l'oggetto degli attributi del thread.
0x0000001d	0x1d	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	Impossibile impostare la dimensione dello stack per l'oggetto degli attributi del thread.
0x0000001e	0x1e	STATUS_MEMORY_NOT_FREED	Usato solo nei test. Indica che non tutta la memoria richiesta è stata liberata.

Codice	Codice senza 0 iniziali	Messaggio	Description
0x0000001f	0x1f	STATUS_INVALID_THREAD_PARAMS_VERSION	Versione della struttura "" non valida. ThreadParams Specificare la versione corrente corretta della struttura .

Codici di errore e di stato restituiti dalla libreria heap

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della Heap libreria.

Codice	Messaggio	Description
0x10000001	STATUS_HEAP_FLAGS_ERROR	È stata specificata una combinazione di flag non valida.
0x10000002	STATUS_HEAP_NOT_INITIALIZED	È stata tentata un'operazione prima dell'inizializzazione dell'heap.
0x10000003	STATUS_HEAP_CORRUPTED	L'heap era danneggiato o la banda di guardia (in modalità debug) è stata sovrascritta. Il danneggiamento dell'heap potrebbe essere causato da un overflow del buffer nel codice del client.
0x10000004	STATUS_HEAP_VRAM_LIB_MISSING	La libreria in modalità utente o kernel VRAM (RAM video) non può essere caricata o è mancante. Verificare che

Codice	Messaggio	Description
		la piattaforma sottostante supporti le allocazioni della VRAM.
0x10000005	STATUS_HEAP_VRAM_L IB_REOPEN	Impossibile aprire la libreria della VRAM.
0x10000006	STATUS_HEAP_VRAM_I NIT_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione INIT.
0x10000007	STATUS_HEAP_VRAM_A LLOC_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione ALLLOC.
0x10000008	STATUS_HEAP_VRAM_F REE_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione FREE.
0x10000009	STATUS_HEAP_VRAM_L OCK_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione LOCK.
0x1000000a	STATUS_HEAP_VRAM_U NLOCK_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione UNLOCK.
0x1000000b	STATUS_HEAP_VRAM_U NINIT_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione UNINIT.
0x1000000c	STATUS_HEAP_VRAM_G ETMAX_FUNC_SYMBOL	Impossibile caricare l'esportazione della funzione GETMAX.
0x1000000d	STATUS_HEAP_DIRECT _MEM_INIT	Impossibile inizializzare il pool di heap principale nell'heap ibrido.
0x1000000e	STATUS_HEAP_VRAM_I NIT_FAILED	L'inizializzazione dinamica della VRAM non è riuscita.
0x1000000f	STATUS_HEAP_LIBRAR Y_FREE_FAILED	Impossibile annullare l'allocazione e liberare la libreria della VRAM.

Codice	Messaggio	Description
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	L'allocazione della VRAM non è riuscita.
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	La liberazione della VRAM non è riuscita.
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	La mappatura della VRAM non è riuscita.
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	L'annullamento della mappatura della VRAM non è riuscito.
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	La deinizializzazione della VRAM non è riuscita.
0x10000015	STATUS_INVALID_ALLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

Codici di errore e di stato restituiti dalla libreria MKVGen

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della MKVGen libreria.

Codice	Messaggio	Descrizione/Azione consigliata
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Membri della struttura di dati Frame non validi. Assicurati che la durata, la dimensione e i dati del frame siano validi

Codice	Messaggio	Descrizione/Azione consigliata
		e rientrano nei limiti specificati in Quote SDK di Producer .
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	Timestamp del fotogramma non valido. I valori del timestamp di presentazione (PTS) e del timestamp di decodifica (DTS) calcolati sono maggiori o uguali al timestamp del fotogramma iniziale del frammento. Ciò indica un potenziale riavvio della pipeline multimediale o un problema di stabilità del codificatore. Per informazioni sulla risoluzione dei problemi, consulta Errore "Failed to submit frame to Kinesis Video client"
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	È stata specificata una durata non valida del frammento. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	Lunghezza della stringa del tipo di contenuto non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .

Codice	Messaggio	Descrizione/Azione consigliata
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	Si è verificato un tentativo di codifica di un numero troppo grande per essere rappresentato nel formato EBML (Extensible Binary Meta Language). Questo errore non dovrebbe essere esposto ai client dell'SDK.
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	Lunghezza della stringa dell'ID codec non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	Lunghezza della stringa del nome della traccia non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	Lunghezza dei dati privati del codec non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	I dati privati del codec (CPD) sono NULL, mentre la dimensione CPD è maggiore di zero.
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	Valore di scala del timecode non valido. Per ulteriori informazioni, consulta Quote SDK di Producer .

Codice	Messaggio	Descrizione/Azione consigliata
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	Il timecode del fotogramma è superiore al valore massimo. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	Il valore massimo del timecode del fotogramma è stato raggiunto. Il formato MKV utilizza 16-bit con firma per rappresentare il timecode relativo del fotogramma all'inizio del cluster. L'errore viene generato se il timecode del frame non può essere rappresentato. Questo errore indica una selezione di scala del timecode non valida o che la durata del cluster è troppo lunga. Di conseguenza, il timecode del fotogramma supera lo spazio di 16-bit con firma.

Codice	Messaggio	Descrizione/Azione consigliata
0x3200000d	STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA	È stato rilevato un codice di Annex-B avvio non valido. Ad esempio, è stato specificato il flag di Annex-B adattamento e il codice rileva una sequenza iniziale non valida di più di tre zeri. Un Annex-B formato valido dovrebbe avere una sequenza di «prevenzione dell'emulazione» per evitare una sequenza di tre o più zeri nel bytestream. Per ulteriori informazioni, consulta la specifica MPEG. Per ulteriori informazioni su questo errore su Android, consulta STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) error on Android .
0x3200000e	STATUS_MKV_INVALID _AVCC_NALU_IN_FRAME DATA	Pacchetto AVCC NALU non valido quando viene specificato il flag AVCC di adattamento. Verificate che il bytestream sia in un formato AVCC valido. Per ulteriori informazioni, consulta la specifica MPEG.
0x3200000f	STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED	Sono stati specificati sia l'adattamento di AVCC che quello di NALU. Annex-B Specificare soltanto un formato o nessuno dei due.

Codice	Messaggio	Descrizione/Azione consigliata
0x32000010	STATUS_MKV_INVALID _ANNEXB_NALU_IN_CPD	Annex-B Formato di CPD non valido quando viene specificato il flag di adattamento. Annex-B Verificate che il CPD sia in formato valido. Annex-B Se non lo è, rimuovi il flag di Annex-B adattamento CPD.
0x32000011	STATUS_MKV_PTS_DTS _ARE_NOT_SAME	Kinesis Video Streams impone che il PTS (timestamp di presentazione) e il DTS (timestamp di decodifica) siano gli stessi per i frame iniziali del frammento. Si tratta dei fotogrammi chiave che avviano il frammento.
0x32000012	STATUS_MKV_INVALID _H264_H265_CPD	Impossibile H264/H265 analizzare i dati privati del codec.
0x32000013	STATUS_MKV_INVALID _H264_H265_SPS_WIDTH	Impossibile estrarre la larghezza dai dati privati del codec.
0x32000014	STATUS_MKV_INVALID _H264_H265_SPS_HEIGHT	Impossibile estrarre l'altezza dai dati privati del codec.
0x32000015	STATUS_MKV_INVALID _H264_H265_SPS_NALU	SPS NALU non valido H264/H265 .
0x32000016	STATUS_MKV_INVALID _BIH_CPD	Formato di intestazione delle informazioni bitmap non valido nei dati privati del codec.

Codice	Messaggio	Descrizione/Azione consigliata
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	Conteggio unità High Efficiency Video Coding (HEVC) Network Abstraction Layer (NALU) non valido.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	Formato HEVC non valido.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	HEVC NALUs mancanti in Sequence Parameter Set (SPS).
0x3200001a	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	Dimensione HEVC SPS NALU non valida.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	IDC formato Chroma non valido.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	HEVC SPS riservato non valido.
0x3200001d	STATUS_MKV_MIN_ANNEXB_CPD_SIZE	Dimensione minima del valore beta privato del AnnexBb codec. Per H264, questo valore deve essere maggiore o uguale a 11. Per H265, questo valore deve essere maggiore o uguale a 15.
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Dati privati del codec mancanti in Annex-B NALUs.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Codec privato beta non valido in NALUs. Annex-B

Codice	Messaggio	Descrizione/Azione consigliata
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	Lunghezza del nome del tag non valida. Il valore valido è maggiore di zero e inferiore a 128.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	Lunghezza del valore del tag non valida. Il valore valido è maggiore di zero e minore di 256.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	Tag stato generatore non validi.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	Indice non valido di frequenza di campionamento dati privati codec AAC.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	Configurazione canale dati privati codec AAC non valida.
0x32000025	STATUS_MKV_INVALID_AAC_CPD	Dati privati codec AAC non validi.
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	Informazioni sulle tracce non trovate.
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	UUID segmento non valido.
0x32000028	STATUS_MKV_INVALID_TRACK_UID	UID traccia non valido.
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	

Codice	Messaggio	Descrizione/Azione consigliata
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	

Codici di errore e di stato restituiti dalla libreria Trace

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della `Trace` libreria.

Codice	Messaggio
0x10100001	STATUS_MIN_PROFILER_BUFFER

Codici di errore e di stato restituiti dalla libreria Utils

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della `Utils` libreria.

Codice	Messaggio
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW

Codice	Messaggio
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED
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

Codice	Messaggio
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
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

Codice	Messaggio
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

Codici di errore e di stato restituiti dalla libreria View

La tabella seguente contiene informazioni sugli errori e sullo stato restituite dai metodi della View libreria.

Codice	Messaggio	Description
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	È stato specificato un numero di elementi di visualizzazione dei contenuti non valido. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	È stata specificata una durata di visualizzazione dei contenuti non valida. Per ulteriori informazioni, consulta Quote SDK di Producer .
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	Si è tentato di superare la posizione principale.
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	È stato specificato un indice non valido.
0x30000005	STATUS_CONTENT_VIEW_INVALID_TIMESTAMP	Timestamp non valido o sovrapposto. Il timestamp di decodifica del frame deve essere maggiore o uguale al timestamp del frame precedente, più la durata del frame precedent

Codice	Messaggio	Description
		e.: <code>`DTS(n) >= DTS(n-1) + Duration(n-1)`</code> Questo errore spesso indica un codificatore "instabile". Il codificatore produce un incremento dei fotogrammi codificati e i relativi timestamp sono inferiori rispetto alle durate intra-frame. Oppure, il flusso è configurato sull'uso dei timestamp dell'SDK e i fotogrammi vengono inviati più velocemente rispetto alla loro durata. Per evitare qualche «tremolio» nell'encoder, puoi regolare la durata del frame nella struttura <code>Frame</code> quando chiami <code>putFrame ()</code> o <code>put ()</code>. <code>KinesisVideoFrame</code> Alcuni flussi richiedono un controllo più preciso sul tempo per il rilevamento degli errori.
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	È stata specificata una lunghezza dei dati degli elementi di visualizzazione dei contenuti non valida.

Errori e codici di stato restituiti dai PutFrame callback - C producer library

La sezione seguente contiene informazioni sugli errori e sullo stato restituite dai callback per l'PutFrameoperazione all'interno della libreria C producer.

Codice	Messaggio	Description	Azione consigliata
0x15000001	STATUS_STOP_CALLBACK_CHAIN	La catena di callback è stata interrotta.	
0x15000002	STATUS_MAX_CALLBACK_CHAIN	Il limite massimo della catena callback è stato raggiunto.	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	Versione della struttura PlatformCallbacks non valida.	Specificare la versione corrente corretta della struttura.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Versione della struttura ProducerCallbacks non valida.	Specificare la versione corrente corretta della struttura.
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Versione della struttura StreamCallbacks non valida.	Specificare la versione corrente corretta della struttura.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Versione della struttura AuthCallbacks non valida.	Specificare la versione corrente corretta della struttura.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	Versione della struttura ApiCallbacks non valida.	Specificare la versione corrente corretta della struttura.

Codice	Messaggio	Description	Azione consigliata
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Versione della struttura <code>AwsCredentials</code> non valida.	Specificare la versione corrente corretta della struttura.
0x15000009	STATUS_MAX_REQUEST_HEADER_COUNT	Il numero massimo dei conteggi dell'intestazione della richiesta è stato raggiunto.	
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	La lunghezza massima del nome dell'intestazione della richiesta è stata raggiunta.	
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	La lunghezza massima del valore dell'intestazione della richiesta è stata raggiunta.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	Restituzione JSON non valida per una chiamata API.	
0x1500000d	STATUS_CURL_INIT_FAILED	Inizializzazione curl non riuscita.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Inizializzazione lib curl non riuscita.	

Codice	Messaggio	Description	Azione consigliata
0x1500000f	STATUS_INVALID_DESCRIBE_STREAM_RETURN_JSON	JSON di ritorno non valido per. DescribeStream	
0x15000010	STATUS_HMAC_GENERATION_ERROR	Errore di generazione HMAC.	
0x15000011	STATUS_IOT_FAILED	Autorizzazione IoT non riuscita.	
0x15000012	STATUS_MAX_ROLE_ALIAS_LEN_EXCEEDED	La lunghezza massima dell'alias del ruolo è stata raggiunta.	Specificare una lunghezza dell'alias minore.
0x15000013	STATUS_MAX_USER_AGENT_POSTFIX_LEN_EXCEEDED	La lunghezza massima del suffisso del nome agente è stata raggiunta.	
0x15000014	STATUS_MAX_CUSTOM_USER_AGENT_LEN_EXCEEDED	La lunghezza massima dell'agente utente cliente è stata raggiunta.	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	Lunghezza agente utente non valida.	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	Periodo di caching dell'endpoint non valido.	Specificare un periodo di caching inferiore a 24 ore.

Codice	Messaggio	Description	Azione consigliata
0x15000017	STATUS_IOT_EXPIRATION_OCCURS_IN_PAST	Il timestamp di scadenza dell'IoT risale al passato.	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	L'analisi della scadenza dell'IoT non è riuscita.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FREE_FUNC		
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		

Codice	Messaggio	Description	Azione consigliata
0x1500001e	STATUS_MAX_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_LWS_CONTENT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEING_SHUTDOWN		

Codice	Messaggio	Description	Azione consigliata
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RETRY_FAILED		
0x16000001	STATUS_CURL_PERFORM_FAILED	CURL ha restituito un codice non riuscito.	Esamina i log per ulteriori informazioni. Un errore CURL comune è «Impossibile risolvere il nome host. », controlla la connettività Internet del dispositivo. Un altro errore comune è il codice di errore 403. Ciò indica che i certificati IoT non sono stati creati o specificati correttamente. Verifica che i percorsi dei file ai certificati IoT e le autorizzazioni siano impostate correttamente. Per ulteriori informazioni, consulta the section called “Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT”.

Codice	Messaggio	Description	Azione consigliata
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH	Ha ricevuto una risposta di lunghezza 0 durante il recupero delle credenziali IoT.	Controlla il pannello di controllo dello AWS stato e riprova più tardi.
0x16000003	STATUS_IOT_NULL_AWS_CREDS	Il codice JSON restituito dall'endpoint delle credenziali IoT non conteneva l'oggetto credenziali.	Controlla l'elemento «messaggio» in JSON per ulteriori informazioni.
0x16000004	STATUS_IOT_INVALID_URI_LEN	L'URL passato alla funzione fetch IoT credenziali non ha una lunghezza compresa tra 1 e 10.000.	Controlla l'URL passato a questa funzione.
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT	L'elemento di «scadenza» nel JSON relativo al recupero delle credenziali IoT non è nel formato: YYYY-MM-DDTHH:mm:ssZ	Controlla la dashboard dello AWS stato di salute e riprova più tardi.

Documentazione di riferimento per i flag di adattamento del Network Abstraction Layer (NAL)

Questa sezione contiene informazioni sui flag disponibili per l'enumerazione `StreamInfo.NalAdaptationFlags`.

Lo [stream elementare](#) di un'applicazione può essere in formato AVCC Annex-Bo in uno dei seguenti formati:

- Il Annex-B formato delimita [NALus \(Network Abstraction Layer units\)](#) con due byte di zeri, seguiti da uno o tre byte di zeri, seguiti dal numero 1 (chiamato codice iniziale, ad esempio 00000001).
- Anche il formato AVCC esegue il wrapping delle NALU, ma ogni NALU è preceduta da un valore che ne indica la dimensione (di solito quattro byte).

Molti Annex-B codificatori producono il formato bitstream. Alcuni processori bitstream di livello superiore (come un motore di riproduzione o il lettore [Media Source Extensions \(MSE\) Console di gestione AWS](#)) utilizzano il formato AVCC per i propri frame.

Il codec private data (CPD), che corrisponde al codec SPS/PPS (Sequence Set/Picture Parameter Set), può anche essere in formato H.264 AVCC. Annex-B Tuttavia, per i CPD, i formati sono diversi da quelli descritti in precedenza.

I flag indicano all'SDK di adattare il NALU ad AVCC o Annex-B per i dati dei frame e il CPD come segue:

Flag	Adattamento	
NAL_ADAPTATION_FLAG_NONE	Nessun adattamento.	
NAL_ADAPTATION_ANNEXB_NALS	Adattare i Annex-B NALU ai NALU AVCC.	
NAL_ADAPTATION_AVCC_NALS	Adattare i NALU AVCC ai NALU. Annex-B	

Flag	Adattamento	
NAL_ADAPTATION_ANN EXB_CPD_NALS	Adatta i Annex-B NALU per i dati privati del codec ai NALU in formato AVCC.	
NAL_ADAPTATION_ANN EXB_CPD_AND_FRAME_ NALS	Adatta Annex-B NALus per il codec e inquadra i dati privati ai NALU in formato AVCC.	

Per ulteriori informazioni sui tipi di NALU, consulta la sezione 1.3 relativa ai tipi di Network Abstraction Layer Unit in [RFC 3984](#).

Strutture SDK di Producer

Questa sezione include informazioni sulle strutture che è possibile utilizzare per fornire dati all'oggetto Kinesis Video Streams Producer.

Argomenti

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

DeviceInfo/DefaultDeviceInfoProvider

DefaultDeviceInfoProvider Gli oggetti DeviceInfo and controllano il comportamento dell'oggetto Kinesis Video Streams Producer.

Campi membri

- **versione:** un valore intero utilizzato per assicurarsi che la versione corretta della struttura venga utilizzata con la versione corrente del codice base. La versione corrente viene specificata con la macro `DEVICE_INFO_CURRENT_VERSION`.
- **name:** il nome leggibile dall'uomo per il dispositivo.
- **tagCount/tags**— Non utilizzato attualmente.
- **StreamCount:** il numero massimo di stream che il dispositivo può gestire. Questo campo effettua inizialmente la pre-allocazione dello storage per i puntatori verso gli oggetti di flusso, ma gli oggetti

di flusso effettivi vengono creati in un secondo momento. Il valore predefinito è 16 flussi, ma questo valore può essere modificato nel file `DefaultDeviceInfoProvider.cpp`.

- `storageInfo`: un oggetto che descrive la configurazione principale dello storage. Per ulteriori informazioni, consulta [StorageInfo](#).

StorageInfo

Specifica la configurazione dello storage principale per Kinesis Video Streams.

L'implementazione predefinita si basa su un'implementazione di heap rapida a bassa frammentazione, ottimizzata per lo streaming. Utilizza l'allocatore `MEMALLOC`, che è possibile sovrascrivere su determinate piattaforme. Alcune piattaforme dispongono di un'allocazione di memoria virtuale senza il supporto dell'allocazione con pagine fisiche. Man mano che la memoria viene utilizzata, le pagine virtuali vengono supportate dalle pagine fisiche. Di conseguenza, quando lo storage è sottoutilizzato, la memoria viene utilizzata in modo ridotto in tutto il sistema.

Calcolare la dimensione predefinita dello storage in base alla seguente formula.

`DefragmentationFactor` deve essere impostato su 1.2 (20%).

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

Nell'esempio seguente, un dispositivo dispone di flussi video e audio. Il flusso audio contiene 512 campioni al secondo, con una media di campionamento di 100 byte. Il flusso video è composto da 25 fotogrammi al secondo, con una media di campionamento di 10.000 byte. Ogni flusso ha 3 minuti di durata di buffer.

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

Se il dispositivo ha più memoria disponibile, ti consigliamo di aggiungere altra memoria allo storage per evitare una grave frammentazione.

Verifica che le dimensioni di archiviazione siano adeguate a contenere tutti i buffer per tutti i flussi a elevata complessità di codifica (quando la dimensione del frame è maggiore a causa dell'elevato movimento) o quando la larghezza di banda è ridotta. Se il produttore raggiunge la pressione della memoria, emette callback della pressione di overflow dello storage ().

`StorageOverflowPressureFunc` Tuttavia, quando non è disponibile alcuna memoria nell'archivio dei contenuti, il frame che viene inviato a Kinesis Video Streams viene eliminato con un errore (`STATUS_STORE_OUT_OF_MEMORY = 0x5200002e`). Per ulteriori informazioni, consulta [Codici di errore e di stato restituiti dalla libreria client](#). Ciò può verificarsi anche se le ACK dell'applicazione non sono disponibili o se le ACK persistenti vengono ritardate. In questo caso, i buffer raggiungono la capacità di «durata del buffer» prima che i frame precedenti inizino a cadere.

Campi per i membri

- `versione`: un valore intero utilizzato per assicurarsi che la versione corretta della struttura venga utilizzata con la versione corrente del codice base.
- `StorageType`: enumerazione che specifica il `DEVICE_STORAGE_TYPE` supporto e l'implementazione sottostanti dello storage. Attualmente, l'unico valore supportato è `DEVICE_STORAGE_TYPE_IN_MEM`. `DEVICE_STORAGE_TYPE_HYBRID_FILE` sarà supportato in un'implementazione futura per indicare che lo storage utilizza lo store dei contenuti supportato da file.
- `StorageSize`: la dimensione di archiviazione in byte da preallocare. L'allocazione minima è 10 MB, mentre quella massima è 10 GB (questo valore cambierà nell'implementazione futura dello store dei contenuti supportato da file).
- `SpillRatio` — Un valore intero che rappresenta la percentuale di storage da allocare dal tipo di storage a memoria diretta (RAM), rispetto allo storage overflow secondario (archiviazione di file). Attualmente non utilizzato.
- `rootDirectory`: il percorso alla directory in cui è si trova lo store dei contenuti supportato da file. Attualmente non utilizzato.

Strutture dei flussi video Kinesis

È possibile utilizzare le seguenti strutture per fornire dati a un'istanza di un flusso video Kinesis.

Argomenti

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

StreamDefinition/StreamInfo

L'oggetto `StreamDefinition` nel livello C++ racchiude l'oggetto `StreamInfo` nel Platform Independent Code (PIC) e fornisce alcuni valori predefiniti nel costruttore.

Campi membri

Campo	Tipo di dati	Description	Valore predefinito
<code>stream_name</code>	<code>string</code>	Nome di flusso facoltativo. Consulta Quote SDK di Producer per ulteriori informazioni sulla lunghezza del nome del flusso. Ogni flusso deve avere un nome univoco.	Se non viene specificato alcun nome, ne verrà generato uno casualmente.
<code>retention_period</code>	<code>duration<uint64_t, ratio<3600>></code>	Il periodo di conservazione del flusso in secondi. La specifica di <code>0</code> indica nessuna conservazione.	3600 (un'ora)
<code>tags</code>	<code>const map<string, string>*</code>	Una mappa di coppie chiave-valore contenente informazioni sugli utenti. Se il flusso dispone già un set di tag, i nuovi tag verranno aggiunti al set di tag esistente.	Nessun tag
<code>kms_key_id</code>	<code>string</code>	L'ID della AWS KMS chiave da utilizzare per crittografare lo	La chiave di KMS predefinita (<code>aws/kinesisvideo</code>).

Campo	Tipo di dati	Description	Valore predefinito
		stream. Per ulteriori informazioni, consulta Protezione dei dati in Kinesis Video Streams .	
streaming_type	Enumerazione STREAMING_TYPE	L'unico valore supportato è STREAMING_TYPE_REALTIME .	
content_type	string	Il formato del contenuto del flusso. La console Kinesis Video Streams è in grado di riprodurre contenuti video/h264 in questo formato.	video/h264
max_latency	duration< uint64_t, milli>	La latenza massima del flusso in milliseconds. Il callback di utilizzo della latenza del flusso (se specificato) verrà richiamato se la durata del buffer supera questo intervallo di tempo. Un valore 0 indica che non verrà richiamato o nessun callback di utilizzo della latenza del flusso.	milliseconds::zero()

Campo	Tipo di dati	Description	Valore predefinito
fragment_duration	duration<uint64_t>	La durata del frammento desiderata, in secondi. Tale valore viene utilizzato in combinazione con il valore key_frame_fragmentation . Se questo valore è false, Kinesis Video Streams genera frammenti su un fotogramma chiave allo scadere di tale periodo. Ad esempio, ciascun fotogramma del flusso audio Advanced Audio Coding (AAC) è un fotogramma chiave. Se viene specificato il valore key_frame_fragmentation = false , la frammentazione verrà effettuata su un fotogramma chiave in seguito a questa durata, generando frammenti di 2 secondi.	2

Campo	Tipo di dati	Description	Valore predefinito
timecode_scale	duration<uint64_t, milli>	La dimensione del timecode MKV in millisecondi, che specifica la granularità dei timecode dei fotogrammi all'interno del cluster MKV. Il timecode del frammento MKV è sempre relativo all'inizio del cluster. MKV utilizza un valore a 16-bit con firma (0-32767) per rappresentare il timecode all'interno del cluster (frammento). Verifica che il codice temporale del frame possa essere rappresentato con la scala del codice di tempo specificata. Il valore di scala predefinito del timecode di 1 ms assicura che il fotogramma di maggiori dimensioni che può essere rappresentato sia pari a 32767 ms ~ = 32 secondi. Questo valore è superiore alla	1

Campo	Tipo di dati	Description	Valore predefinito
		durata massima del frammento specificata in Quote del servizio Amazon Kinesis Video Streams , pari a 10 secondi.	
key_frame_fragmentation	bool	Per scegliere se creare frammenti su un fotogramma chiave. Se true, l'SDK crea un inizio del frammento ogni volta che è presente un fotogramma chiave. Se false, Kinesis Video Streams attende fragment_duration almeno e produce un nuovo frammento sul fotogramma chiave successivo.	true

Campo	Tipo di dati	Description	Valore predefinito
frame_timecodes	bool	Per scegliere se utilizzare i timecode del fotogramma o generare timestamp tramite il callback dell'ora corrente. Molti codificatori non creano timestamp con i fotogrammi. Quindi, specificando false questo parametro, si assicura che i fotogrammi abbiano il timestamp quando vengono inseriti in Kinesis Video Streams.	true

Campo	Tipo di dati	Description	Valore predefinito
<code>absolute_fragment_times</code>	<code>bool</code>	Kinesis Video Streams utilizza MKV come meccanismo di packaging di base. La specifica MKV è rigorosa per quanto riguarda l'impostazione dei timecode del fotogramma come relativi all'inizio del cluster (frammento), ma i timecode del cluster possono essere assoluti o relativi all'ora di inizio del flusso. Se i timestamp sono relativi, la chiamata API del servizio PutMedia utilizza il timestamp di inizio facoltativo del flusso e modifica i timestamp del cluster. Il servizio archivia sempre i frammenti con i relativi timestamp assoluti.	<code>true</code>
<code>fragment_acks</code>	<code>bool</code>	Se ricevere ACK (riconoscimenti) con frammenti a livello di applicazione.	<code>true</code> , per indicare che l'SDK riceverà le ACK e agirà di conseguenza.

Campo	Tipo di dati	Description	Valore predefinito
restart_on_error	bool	Per scegliere se riavviare in seguito a errori specifici.	true, per indicare che l'SDK tenta di riavviare lo streaming in caso di errori.
recalculate_metrics	bool	Per scegliere se calcolare nuovamente i parametri. Ogni chiamata effettuata per recuperare i parametri è in grado di ricalcolarli per ottenere il valore "in esecuzione" più recente (ciò potrebbe avere un leggero impatto sulla CPU). Potrebbe essere necessario impostarlo a false su power/footprint dispositivi con dimensioni estremamente ridotte per risparmiare i cicli della CPU. Altrimenti, non è consigliabile utilizzarlo a false per questo valore.	true

Campo	Tipo di dati	Description	Valore predefinito
<code>nal_adaptation_flags</code>	<code>uint32_t</code>	Specifica i flag di adattamento dell'unità a Network Abstraction Layer (NALU). Se il bitstream è H.264 codificato, può essere elaborato come grezzo o impacchettato in NALus. Questi sono in formato o AVCC. Annex-B La maggior parte dei produttori e consumatori di stream elementari (codificatori e decodificatori di lettura) utilizza il Annex-B formato perché presenta vantaggi, come il ripristino degli errori. Higher-leveli sistemi utilizzano il formato AVCC, che è il formato predefinito per MPEG, HLS, DASH e così via. La funzione di riproduzione della console utilizza le estensioni di sorgente multimediali (MSE) del browser per decodificare e riprodurre i flussi	L'impostazione predefinita consiste nell'adattare il Annex-B formato al formato AVCC sia per i dati del frame che per i dati privati del codec.

Campo	Tipo di dati	Description	Valore predefinito
		in formato AVCC. Per H.264 (e per M-JPEG e H.265), l'SDK offre funzionalità di adattamento. Molti flussi semplici sono nel formato seguente. In questo esempio, Ab è il codice Annex-B iniziale (001 o 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> Nel caso di H.264, i dati privati del codec (CPD) sono nei parametri SPS (sequence parameter s set) e PPS (picture parameters set) e possono essere adattati al formato AVCC. A meno che la pipeline multimediale non fornisca i CPD separatamente,	

Campo	Tipo di dati	Description	Valore predefinito
		l'applicazione è in grado di estrarli dal fotogramma. Può farlo cercando il primo frame IDR (che dovrebbe contenere SPS e PPS), estrarre i due NALU (che lo sono) e impostarli nel CPD in. StreamDefinition Per ulteriori informazioni, consulta the section called “Bandiere di adattamento NAL”.	
frame_rate	uint32_t	La frequenza fotogrammi prevista. Tale valore viene utilizzato per calcolare meglio le esigenze di buffering.	25
avg_bandwidth_bps	uint32_t	La larghezza di banda media prevista del flusso. Tale valore viene utilizzato per calcolare meglio le esigenze di buffering.	4 * 1024 * 1024

Campo	Tipo di dati	Description	Valore predefinito
buffer_duration	duration<uint64_t>	La durata del buffer del flusso in secondi. L'SDK mantiene i frame nell'archivio dei contenuti per un massimo di buffer_duration , dopodiché i frame precedenti i vengono eliminati man mano che la finestra avanza. Se il frame che viene eliminato non è stato inviato al backend, viene chiamato il callback del frame eliminato. Se la durata del buffer corrente è maggiore di max_latency , viene richiamato il callback di utilizzo della latenza del flusso. Alla ricezione dell'ACK relativa al frammento persistente, il buffer verrà tagliato in corrispondenza dell'inizio del frammento successivo o per indicare che i contenuti sono stati resi persistenti in	120

Campo	Tipo di dati	Description	Valore predefinito
		modo durevole nel cloud e che quindi non è più necessario archiviare i contenuti sul dispositivo locale.	

Campo	Tipo di dati	Description	Valore predefinito
replay_duration	duration<uint64_t>	La durata, in secondi, necessaria per riportare il lettore corrente all'indietro per riprodurlo durante un errore, se il riavvio è abilitato. Il rollback viene interrotto all'inizio del buffer (se ha appena avviato lo streaming o se viene visualizzata l'ACK persistente). Il rollback tenta di raggiungere un fotogramma chiave che indica l'inizio di un frammento. Se l'errore che causa il riavvio non è indicativo di un host morto (l'host è ancora attivo e contiene i dati del frame nei suoi buffer interni), il rollback si interrompe all'ultimo frame ACK ricevuto. e verrà spostato al fotogramma chiave successivo, poiché l'intero frammento è già memorizzato nella memoria dell'host.	40

Campo	Tipo di dati	Description	Valore predefinito
connection_staleness	duration<uint64_t>	Il tempo, in secondi, dopo il quale viene chiamata la callback stream staleness se l'SDK non riceve l'ACK di buffering. Indica che i frame vengono inviati dal dispositivo, ma il backend non li riconosce. indicativ o di una connessio ne interrotta nel passaggio intermedi o o sul sistema di bilanciamento del carico.	30
codec_id	string	L'ID codec della traccia MKV.	V_MPEG4/ISO/AVC
track_name	string	Il nome della traccia MKV.	kinesis_video

Campo	Tipo di dati	Description	Valore predefinito
codec PrivateData	unsigned char*	Il buffer dei dati privati del codec (CPD). Se sulla pipeline multimediale sono presenti informazioni sui CPD prima dell'inizio del flusso, sarà possibile impostarlo su <code>StreamDefinition.codecPrivateData</code> . I bit vengono copiati e il buffer può essere riutilizzato o liberato in seguito alla chiamata per la creazione del flusso. Tuttavia, se i dati non sono disponibili al momento della creazione dello stream, possono essere impostati in uno degli overload della <code>KinesisVideoStream.start(cpd)</code> funzione.	null
codec PrivateDataSize	uint32_t	La dimensione del buffer dei dati privati del codec.	0

ClientMetrics

L'`ClientMetrics` oggetto viene riempito `getKinesisVideoMetrics` chiamando.

Campi membri

Campo	Tipo di dati	Description
versione	UINT32	La versione della struttura , definita nella macro <code>CLIENT_METRICS_CURRENT_VERSION</code> .
contenutoStoreSize	UINT64	La dimensione totale dello store dei contenuti in byte. Si tratta del valore specificato in <code>DeviceInfo.StorageInfo.storageSize</code> .
contenuto StoreAvailableSize	UINT64	dimensione di archiviazione attualmente disponibile in byte.
contenuto StoreAllocatedSize	UINT64	Dimensione attualmente allocata. La dimensione allocata e quella disponibile devono essere leggermente inferiori alla dimensione totale di storage, per via della contabilità interna e dell'implementazione dello store dei contenuti.
totale ContentViewsSize	UINT64	La dimensione della memoria allocata per tutte le visualizzazioni dei contenuti per tutti i flussi. Questo non viene conteggiato nella dimension

Campo	Tipo di dati	Description
		e dello spazio di archiviazione. Questa memoria viene allocata tramite la macro <code>MEMALLOC</code> , che è possibile sovrascrivere per fornire un allocatore personalizzato.
totale FrameRate	UINT64	La frequenza fotogrammi totale osservata su tutti i flussi.
totale TransferRate	UINT64	La velocità di streaming totale osservata in byte al secondo su tutti i flussi.

StreamMetrics

L'`StreamMetrics` oggetto viene riempito chiamando `getKinesisVideoMetrics`.

Campi membri

Campo	Tipo di dati	Description
versione	UINT32	La versione della struttura, definita nella macro <code>STREAM_METRICS_CURRENT_VERSION</code> .
attuale ViewDuration	UINT64	La durata dei fotogrammi accumulati. Nel caso di rete veloce, questa durata è pari a zero o alla durata del frame (mentre il frame viene trasmesso). Se la durata diventa più lunga di quella <code>max_latency</code> specificata in <code>StreamDefinition</code> ,

Campo	Tipo di dati	Description
		viene chiamata la richiamat a della latenza del flusso, se specificata. La durata viene specificata in unità di 100ns, ovvero l'unità di tempo predefinita del livello PIC.
complessivamente ViewDuration	UINT64	La durata di visualizzazione generale. Se lo stream è configurato senza ACK o persistenza, questo valore aumenta man mano che i frame vengono inseriti nel flusso video Kinesis e diventa uguale a quello <code>buffer_duration</code> in <code>StreamDefinition</code> . Quando gli ACK sono abilitati e viene ricevuto l'ACK persistente, il buffer viene ridotto al keyframe successivo. Questo perché il timestamp ACK indica l'inizio dell'intero frammento. La durata viene specificata in unità di 100 ns, ovvero l'unità di tempo predefinita del livello PIC.
attuale ViewSize	UINT64	La dimensione in byte del buffer corrente.
complessivamente ViewSize	UINT64	La dimensione totale della visualizzazione in byte.
attuale FrameRate	UINT64	La frequenza fotogramma osservata del flusso corrente.

Campo	Tipo di dati	Description
attuale TransferRate	UINT64	La velocità di trasferimento osservata in byte al secondo del flusso corrente.

richiamate SDK di Producer

Le classi e i metodi dell'SDK Amazon Kinesis Video Streams Producer non mantengono processi propri. Vengono, invece, utilizzate le chiamate di funzione in ingresso e gli eventi per programmare callback e comunicare con l'applicazione.

Sono disponibili due modelli di callback che l'applicazione può utilizzare per interagire con l'SDK:

- **CallbackProvider**— Questo oggetto espone ogni callback dal componente PIC (Platform-Independent Code) all'applicazione. Questo modello consente di eseguire questa funzionalità, ma anche significa che l'implementazione deve gestire tutti i metodi API pubblici e le firme nel livello C++.
- [StreamCallbackProvider](#) e [ClientCallbackProvider](#) — Questi oggetti espongono i callback specifici dello stream e del client, mentre il livello C++ dell'SDK espone il resto dei callback. Questo è il modello di callback preferito per interagire con l'SDK Producer.

Il seguente diagramma mostra il modello di oggetti callback:

Nel diagramma precedente, `DefaultCallbackProvider` deriva da `CallbackProvider` (che espone tutti i callback nel PIC) e contiene `StreamCallbackProvider` e `ClientCallbackProvider`.

Questo argomento contiene le sezioni seguenti:

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks struttura](#)
- [Implementazioni di callback per riprovare lo streaming](#)

ClientCallbackProvider

L'oggetto `ClientCallbackProvider` espone le funzioni di callback a livello di client. I dettagli delle funzioni sono descritti nella sezione [the section called “ClientCallbacks”](#).

Metodi di callback:

- `getClientReadyCallback`— Indica lo stato di disponibilità per il cliente.
- `getStorageOverflowPressureCallback`— Segnala il sovraccarico o la pressione dello storage. Questo callback viene chiamato quando l'utilizzo dello storage è inferiore al valore `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, pari al 5% delle dimensioni dello spazio di archiviazione generale. Per ulteriori informazioni, consulta [StorageInfo](#).

StreamCallbackProvider

L'oggetto `StreamCallbackProvider` espone le funzioni di callback a livello di flusso.

Metodi di callback:

- `getDroppedFragmentReportCallback`: segnala un frammento interrotto.
- `getDroppedFrameReportCallback`— Segnala un frame perso.
- `getFragmentAckReceivedCallback`— Segnala che è stato ricevuto un frammento ACK per lo stream.
- `getStreamClosedCallback`— Segnala una condizione di chiusura dello stream.
- `getStreamConnectionStaleCallback`— Segnala una condizione di connessione obsoleta. In questa condizione, il produttore invia dati al servizio ma non riceve riconoscimenti.
- `getStreamDataAvailableCallback`— Segnala che i dati sono disponibili nello stream.
- `getStreamErrorReportCallback`— Segnala una condizione di errore dello stream.
- `getStreamLatencyPressureCallback`— Segnala una condizione di latenza del flusso, ovvero quando la dimensione del buffer accumulato è maggiore del valore. `max_latency` Per ulteriori informazioni, consulta [StreamDefinition/StreamInfo](#).
- `getStreamReadyCallback`: —Segnala una condizione di disponibilità dello stream.
- `getStreamUnderflowReportCallback`— Segnala una condizione di sottoflusso del flusso. Questa funzione non è attualmente utilizzata ed è riservata per utilizzi futuri.

Per il codice sorgente di `StreamCallbackProvider`, vedi [StreamCallbackProvider.h](#).

ClientCallbacks struttura

La struttura `ClientCallbacks` contiene i punti di accesso della funzione di callback che il PIC chiama quando si verificano determinati eventi. La struttura, inoltre, contiene le informazioni sulla versione nel campo `CALLBACKS_CURRENT_VERSION` e un campo `customData` per i dati definiti dall'utente che vengono restituiti con le singole funzioni di callback.

L'applicazione client può utilizzare un puntatore `this` per il campo `custom_data` per mappare le funzioni membro per le funzioni statiche `ClientCallback` in fase di runtime, come mostra il seguente codice di esempio:

```
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(...);
```

Eventi

Funzione	Description	Tipo
CreateDeviceFunc	Non attualmente implementato nel back-end. Questa chiamata ha esito negativo quando viene effettuata da Java o C++. Altri client eseguono l'inizializzazione specifica per piattaforma.	API di back-end
CreateStreamFunc	Chiamata al momento della creazione del flusso	API di back-end
DescribeStreamFunc	Chiamata con <code>DescribeStream</code> .	API di back-end
GetStreamingEndpointFunc	Chiamata con <code>GetStreamingEndpoint</code> .	API di back-end

Funzione	Description	Tipo
GetStreamingTokenFunc	Chiamata con GetStreamingToken .	API di back-end
PutStreamFunc	Chiamata con PutStream .	API di back-end
TagResourceFunc	Chiamata con TagResource .	API di back-end
CreateMutexFunc	Crea un mutex per la sincronizzazione.	Sincronizzazione
FreeMutexFunc	Libera il mutex.	Sincronizzazione
LockMutexFunc	Blocca il mutex di sincronizzazione.	Sincronizzazione
TryLockMutexFunc	Tenta di bloccare il mutex. Non attualmente implementato.	Sincronizzazione
UnlockMutexFunc	Sblocca il mutex.	Sincronizzazione
ClientReadyFunc	Viene chiamato quando il client entra nello stato pronto.	Notification
DroppedFrameReportFunc	Segnala quando un fotogramma viene interrotto.	Notification
DroppedFragmentReportFunc	Segnala quando un frammento viene interrotto. Questa funzione non è attualmente utilizzata ed è riservata per utilizzi futuri.	Notification

Funzione	Description	Tipo
FragmentAckReceiveFunc	Viene chiamato quando viene ricevuto un frammento ACK (buffering, ricevuto, persistente e di errore).	Notification
StorageOverflowPressureFunc	Chiamato quando l'utilizzo dello storage è al di sotto del valore STORAGE_PRESSURE_NOTIFICATION_THRESHOLD, pari al 5% delle dimensioni dello spazio di archiviazione generale.	Notification
StreamClosedFunc	Chiamato quando vengono trasmessi gli ultimi bit dei fotogrammi rimanenti.	Notification
StreamConnectionStaleFunc	Chiamato quando il flusso entra in uno stato di connessione obsoleto. In questa condizione, il produttore invia dati al servizio, ma non riceve conferme.	Notification
StreamDataAvailableFunc	Chiamato quando sono disponibili i dati di flusso.	Notification
StreamErrorReportFunc	Chiamato quando si verifica un errore di flusso. Il PIC chiude automaticamente il flusso in questa condizione.	Notification

Funzione	Description	Tipo
StreamLatencyPressureFunc	Chiamato quando il flusso entra in una condizione di latenza, quando la dimensione del buffer accumulata è superiore al valore <code>max_latency</code> . Per ulteriori informazioni, consulta StreamDefinition/StreamInfo .	Notification
StreamReadyFunc	Viene chiamato quando il flusso entra nello stato pronto.	Notification
StreamUnderflowReportFunc	Questa funzione non è attualmente utilizzata ed è riservata per utilizzi futuri.	Notification
DeviceCertToTokenFunc	Restituisce il certificato di connessione come token.	Integrazione con la piattaforma
GetCurrentTimeFunc	Restituisce l'orario attuale.	Integrazione con la piattaforma
GetDeviceCertificateFunc	Restituisce il certificato del dispositivo. Questa funzione non è attualmente utilizzata ed è riservata per utilizzi futuri.	Integrazione con la piattaforma
GetDeviceFingerprintFunc	Restituisce l'impronta del dispositivo. Questa funzione non è attualmente utilizzata ed è riservata per utilizzi futuri.	Integrazione con la piattaforma
GetRandomNumberFunc	Restituisce un numero casuale compreso tra 0 e <code>RAND_MAX</code> .	Integrazione con la piattaforma

Funzione	Description	Tipo
<code>GetSecurityTokenFunc</code>	Restituisce il token di sicurezza passato alle funzioni che comunicano con l'API di backend. L'implementazione è in grado di specificare i <code>AccessKeyId</code> e <code>SecretKeyId</code> serializzati e il token della sessione.	Integrazione con la piattaforma
<code>LogPrintFunc</code>	Registra una riga di testo con il tag e il livello di log. Per ulteriori informazioni, consulta <code>PlatformUtils.h</code> .	Integrazione con la piattaforma

Per le funzioni di integrazione della piattaforma nella tabella precedente, l'ultimo parametro è una struttura `ServiceCallContext`, che include i campi riportati di seguito:

- `version`: la versione della strut.
- `callAfter`: un tempo assoluto dopo il quale per chiamare la funzione.
- `timeout`: il timeout dell'operazione in unità di 100 nanosecondi.
- `customData`: un valore definito dall'utente da passare al client.
- `pAuthInfo`: le credenziali per la chiamata. Per maggiori informazioni, consultare la seguente struttura (`__AuthInfo`).

Le informazioni di autorizzazione vengono fornite utilizzando la struttura `__AuthInfo` che può essere credenziali serializzate o un token di autenticazione specifico del provider. Questa struttura include i campi riportati di seguito:

- `version`: la versione della struttura `__AuthInfo`.
- `type`: un valore `AUTH_INFO_TYPE` che definisce il tipo di credenziale (certificato o token di sicurezza).
- `data`: un array di byte contenente le informazioni di autenticazione.
- `size`: la dimensione del parametro `data`.

- `expiration`: la scadenza delle credenziali in unità di 100 nanosecondi.

Implementazioni di callback per riprovare lo streaming

Kinesis Video Producer SDK fornisce lo stato dello streaming tramite le funzioni di callback. Ti consigliamo di implementare i seguenti meccanismi di callback per risolvere eventuali problemi di rete momentanei riscontrati durante lo streaming.

- Richiamata della pressione della latenza dello stream: questo meccanismo di callback viene avviato quando l'SDK rileva una condizione di latenza del flusso. Ciò accade quando la dimensione del buffer accumulato è maggiore del valore `MAX_LATENCY`. Quando il flusso viene creato, l'applicazione di streaming imposta `MAX_LATENCY` sul valore predefinito di 60 secondi. L'implementazione tipica per questo callback è il ripristino della connessione. È possibile utilizzare l'implementazione di esempio in base alle esigenze. <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.c> Tieni presente che non è possibile archiviare i frame non consegnati a causa di un'interruzione della rete in uno storage secondario per il riempimento.
- Stream staleness callback: questo callback viene avviato quando il produttore può inviare dati al servizio Amazon Kinesis Data Streams (uplink) ma non è in grado di recuperare i riconoscimenti (ACK bufferizzato) indietro nel tempo (l'impostazione predefinita è 60 secondi). A seconda delle impostazioni di rete, è possibile avviare lo stream latency pressure callback o lo stream staleness callback, oppure entrambi. Analogamente all'implementazione del nuovo tentativo di callback di utilizzo della latenza del flusso, l'implementazione tipica consiste nel ripristinare la connessione e avviare una nuova connessione per lo streaming. È possibile utilizzare l'implementazione di esempio in base alle esigenze. <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c/blob/master/src/source/ConnectionStateStateMachine.c>
- Stream error callback: questo callback viene avviato quando l'SDK rileva un timeout sulla connessione di rete o altri errori durante la chiamata alle chiamate al servizio API KVS.
- Dropped frame callback: questo callback viene avviato quando lo spazio di archiviazione è pieno, a causa della bassa velocità di rete o di un errore di streaming. Se la velocità della rete comporta la perdita di fotogrammi, è possibile aumentare le dimensioni di archiviazione, ridurre le dimensioni dei fotogrammi video o la frequenza dei fotogrammi in modo che corrisponda alla velocità della rete.

Utilizzo dei metadati di streaming con Kinesis Video Streams

Puoi utilizzare l'SDK per produttori di Amazon Kinesis Video Streams per incorporare i metadati a livello di singolo frammento in un flusso video Kinesis. I metadati in Kinesis Video Streams sono una coppia chiave-valore mutabile. Puoi utilizzarla per descrivere il contenuto del frammento, incorporare le letture dei sensori associate che devono essere trasferite insieme al frammento effettivo o soddisfare altre esigenze personalizzate. I metadati sono disponibili come parte delle operazioni delle API [GetMedia](#) o [GetMediaForFragmentList](#). Viene archiviato insieme ai frammenti per l'intera durata del periodo di conservazione dello stream. Le applicazioni che consumano possono leggere, elaborare e reagire in base ai metadati utilizzando il [Guarda l'output delle telecamere usando la libreria parser](#)

Esistono due modalità in cui i metadati possono essere incorporati con i frammenti in un flusso:

- Non persistente: puoi applicare i metadati una tantum o ad hoc ai frammenti di un flusso, in base a criteri aziendali specifici che si sono verificati. Un esempio è una videocamera intelligente che rileva il movimento e aggiunge metadati ai frammenti corrispondenti che contengono il movimento prima di inviarli al relativo flusso video Kinesis. Puoi applicare i metadati al frammento nel formato seguente: `Motion = true`.
- Persistente: puoi aggiungere metadati a frammenti successivi e consecutivi in un flusso in base a un'esigenza continua. Un esempio è una videocamera intelligente che invia le coordinate correnti di latitudine e longitudine associate a tutti i frammenti che invia al flusso video Kinesis. Puoi applicare i metadati a tutti i frammenti nel formato seguente: `Lat = 47.608013N , Long = -122.335167W`.

Puoi applicare i metadati in entrambe queste modalità allo stesso frammento contemporaneamente in base alle esigenze dell'applicazione. I metadati incorporati potrebbero includere oggetti rilevati, attività tracciate, coordinate GPS o altri dati personalizzati che vuoi associare ai frammenti nel flusso. I metadati sono codificati come coppie di stringhe chiave-valore.

Aggiungere metadati a un flusso video Kinesis

I metadati aggiunti a uno stream video Kinesis sono modellati come tag MKV, implementati come coppie chiave-valore.

I metadati possono essere transitori, ad esempio per contrassegnare un evento all'interno del flusso, o persistenti, ad esempio per identificare frammenti in cui si verifica un determinato evento. Un

elemento di metadati persistente rimane e viene applicato a ogni frammento consecutivo fino a quando non viene annullato.

Note

Gli elementi metadati aggiunti utilizzando le [Caricamento su Kinesis Video Streams](#) differiscono dalle API di tag a livello di flusso implementato con [TagStream](#), [UntagStream](#) e [ListTagsForStream](#).

API per lo streaming di metadati

Puoi utilizzare le seguenti operazioni nell'SDK di Producer per implementare i metadati di streaming.

Argomenti

- [PIC](#)
- [SDK per produttori C++](#)
- [SDK per produttori Java](#)
- [Metadati persistenti e non persistenti](#)

PIC

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

SDK per produttori C++

```
/**
 * 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 per produttori Java

Puoi utilizzare Java Producer SDK, per aggiungere metadati a un utilizzando: `MediaSource` `MediaSourceSink.onCodecPrivateData`

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

Metadati persistenti e non persistenti

Per i metadati non persistenti, puoi aggiungere più elementi metadati con lo stesso nome. Il producer SDK raccoglie gli elementi di metadati nella coda dei metadati fino a quando non vengono aggiunti al frammento successivo. La coda dei metadati viene cancellata quando gli elementi metadati vengono applicati al flusso. Per ripetere i metadati, chiama di nuovo `putKinesisVideoFragmentMetadata` o `putFragmentMetadata`.

Per i metadati persistenti, l'SDK del produttore raccoglie gli elementi di metadati nella coda dei metadati allo stesso modo dei metadati non persistenti. Tuttavia, gli elementi di metadati non vengono rimossi dalla coda quando vengono aggiunti al frammento successivo.

La chiamata `putKinesisVideoFragmentMetadata` o `putFragmentMetadata` con `persistent` impostato su `true` ha il seguente comportamento:

- La chiamata dell'API inserisce l'elemento metadati nella coda. I metadati vengono aggiunti come tag MKV per ogni frammento mentre l'elemento è in coda.
- Se si chiama l'API con lo stesso nome e un valore diverso come elemento metadati aggiunto in precedenza, l'elemento viene sovrascritto.
- Se si chiama l'API con un valore vuoto, l'elemento metadati viene rimosso (annullato) dalla coda dei metadati.

Usa IPv6 con Kinesis Video Streams

È possibile configurare Kinesis Video Streams per utilizzare IPv6 sia per le operazioni sul piano di controllo che sul piano dati. Ciò consente alle applicazioni di comunicare con i servizi Kinesis Video Streams utilizzando indirizzi IPv6 tramite endpoint dual-stack.

Note

Il supporto IPv6 richiede versioni SDK e impostazioni di configurazione specifiche. Assicurati che le versioni SDK e AWS SDK di Kinesis Video Streams supportino gli endpoint dual-stack

IPv6. Dual-stack gli endpoint supportano il traffico IPv4 e IPv6 e sono disponibili per alcuni servizi in alcune regioni.

Kinesis Video Streams supporta IPv6 tramite endpoint dual-stack per applicazioni di produzione e consumo. Puoi configurare le tue applicazioni per utilizzare IPv6 per le chiamate API del piano di controllo e le operazioni di streaming del piano dati.

Configura il AWS SDK per IPv6

Se utilizzi l' AWS SDK per richiamare le API del piano di controllo di Kinesis Video Streams nella configurazione di produzione, puoi abilitare IPv6 configurando gli endpoint dual-stack. L'SDK fornisce diversi metodi standardizzati per abilitare gli endpoint dual-stack. AWS

Important

Quando gli endpoint dual-stack sono abilitati, l'SDK tenta di utilizzare gli endpoint dual-stack per effettuare richieste di rete. Se non esiste un endpoint dual-stack per il servizio o la regione, la richiesta ha esito negativo.

Usa variabili di ambiente

Imposta la seguente variabile di ambiente per abilitare gli endpoint dual-stack IPv6:

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Usa il AWS file di configurazione

Aggiungi la seguente impostazione al tuo file AWS di configurazione (~/.aws/config):

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

Usa le proprietà del sistema JVM (solo SDK Java e Kotlin)

Per le applicazioni Java e Kotlin, imposta la seguente proprietà di sistema JVM:

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

O a livello di codice nel tuo codice Java:

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

Supporto SDK

I seguenti AWS SDK supportano la configurazione degli endpoint dual-stack:

SDK	Supportata	Metodi di configurazione
AWS CLI v2	Sì	Variabile di ambiente, file di configurazione
SDK per C++	Sì	Variabile di ambiente, file di configurazione
SDK per Go V2 (1.x)	Sì	Variabile di ambiente, file di configurazione
SDK per Java 2.x	Sì	Variabile di ambiente, file di configurazione, proprietà JVM
SDK per Java 1.x	No	Non supportata
SDK per 3.x JavaScript	Sì	Variabile di ambiente, file di configurazione
SDK per Python (Boto3)	Sì	Variabile di ambiente, file di configurazione

Dopo aver configurato gli endpoint dual-stack, l' AWS SDK utilizza automaticamente gli endpoint IPv6 quando chiama le API del piano di controllo di Kinesis Video Streams.

Configurazione dell'SDK di Kinesis Video Streams Producer per IPv6

L'SDK Kinesis Video Streams Producer offre opzioni di configurazione IPv6 per le operazioni sul piano di controllo e sul piano dati. Queste impostazioni funzionano con la configurazione degli endpoint dual-stack SDK. AWS

Configura l'SDK ++ Producer C/C

Gli endpoint e la catena di risoluzione DNS predefiniti sono implementati dalla versione 1.6.0 di KVS SDK Producer-C . Controlla in sequenza ogni posizione in cui è possibile impostare la configurazione per questi parametri, quindi seleziona la prima che imposti. La sequenza predefinita è la seguente:

[Per ulteriori informazioni sugli SDK di Producer, consulta gli argomenti correlati a Producer SDK for C, Producer SDK for C++ e Producer SDK.](#)

Configurazione degli endpoint

1. Il `controlPlaneUrl` parametro per `createAbstractDefaultCallbacksProvider`.
2. Parametri CMake di configurazione dell'endpoint: (-
DAWS_KVS_USE_LEGACY_ENDPOINT_ONLY=TRUE,) -
DAWS_KVS_USE_DUAL_STACK_ENDPOINT_ONLY=TRUE
3. Variabili di ambiente: () `export AWS_USE_DUALSTACK_ENDPOINT=TRUE`
 - Se `AWS_USE_DUALSTACK_ENDPOINT` è TRUE (senza distinzione tra maiuscole e minuscole), verrà utilizzato l'endpoint dual-stack.
4. In caso contrario, l'endpoint precedente verrà costruito e utilizzato.

Con 2, 3 e 4, l'endpoint verrà costruito in base alla regione fornita a.

`createAbstractDefaultCallbacksProvider`

Filtraggio DNS

KVS Producer SDK imposterà il `CURLOPT_IPRESOLVE` parametro appropriato in base alla configurazione:

1. Parametri CMake per la risoluzione DNS: (,,) -DAWS_KVS_IPV4_ONLY=TRUE -
DAWS_KVS_IPV6_ONLY=TRUE -DAWS_KVS_IPV4_AND_IPV6_ONLY=TRUE
2. Variabili di ambiente (`export AWS_KVS_USE_IPV4=TRUE,`) `export AWS_KVS_USE_IPV6=TRUE`
3. In caso contrario, non verrà eseguito alcun filtraggio. È possibile utilizzare sia gli indirizzi IP IPv4 che IPv6, se restituiti dal DNS.

 Note

Se le impostazioni del filtro DNS sono impostate per filtrare gli indirizzi IP IPV6, ma la configurazione SDK prevede l'utilizzo dell'endpoint legacy (restituisce gli indirizzi), la richiesta avrà esito negativo. IPV4-only

La versione 3.5.0 di C++ Producer SDK utilizza l'SDK 1.6.0 per le Producer-C chiamate API KVS.

Configura il plugin GStreamer

Il plugin GStreamer utilizza l'SDK C Producer sottostante, quindi la configurazione IPv6 viene gestita automaticamente quando si configura il C SDK per IPv6 come descritto in precedenza.

Non è necessario modificare il codice, è sufficiente creare l'SDK con i parametri CMake o impostare le variabili di ambiente appropriate come descritto nella sezione precedente.

Risoluzione degli endpoint del piano dati

Per le operazioni sul piano dati, utilizza l'GetDataEndpointAPI per recuperare l'endpoint del piano dati dual-stack appropriato. Il servizio restituisce l'endpoint corrispondente in base all'URL della richiesta.

Esempio:

- Se la richiesta GetDataEndpoint API viene effettuata all'endpoint legacy che termina con `.amazonaws.com`, Kinesis Video Streams restituirà gli endpoint del piano dati legacy che terminano con `.amazonaws.com`
- Se la richiesta GetDataEndpoint API viene effettuata all'endpoint dual-stack che termina con `.api.aws`, Kinesis Video Streams restituirà l'endpoint del piano dati dual-stack che termina con `.api.aws`

Configura il AWS CLI per IPv6

Se utilizzi le operazioni di Kinesis Video Streams (in genere AWS CLI per lavori di prototipazione), puoi abilitare IPv6 configurando endpoint dual-stack.

Usa una variabile di ambiente

```
export AWS_USE_DUALSTACK_ENDPOINT=true
```

Usa il AWS file di configurazione

Aggiungi quanto segue al tuo file AWS CLI di configurazione (~/.aws/config):

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

Esempi di configurazione

Esempio di SDK C

Per creare l' Producer-C SDK KVS in IPV6-only modalità e ignorare la configurazione delle variabili di ambiente, usa i seguenti comandi per creare l'SDK:

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

Note

Se hai già creato l'SDK, dovrai eseguire una build pulita. Elimina le cartelle di build, open source e dependency esistenti prima di eseguire i comandi di build.

Considerazioni

Requisiti di rete

- Assicurati che la tua infrastruttura di rete supporti la connettività IPv6
- Verifica che i gruppi di sicurezza e gli ACL di rete consentano il traffico IPv6
- Verifica la connettività agli endpoint AWS IPv6 dal tuo ambiente di distribuzione
- Dual-stack gli endpoint sono disponibili per alcuni servizi in alcune regioni: verifica la disponibilità per le aree di destinazione

Compatibilità con SDK

- Assicurati di utilizzare una versione AWS SDK supportata
- L' AWS SDK for Java 1.x non supporta la configurazione degli endpoint dual-stack
- Per l'SDK for Go 1.x (V1), è necessario abilitare il caricamento dal file di configurazione per utilizzare le impostazioni condivise del file di configurazione

Test e convalida

Prima di distribuire le applicazioni IPv6-enabled Kinesis Video Streams in produzione:

- Verifica le operazioni del piano di controllo (creazione, eliminazione, elenco dei flussi)
- Verifica il funzionamento del piano dati (ingestione e consumo di video)
- Convalida le prestazioni e la connettività nel tuo ambiente di rete
- Esegui test Canary per garantire una funzionalità IPv6 coerente
- Verifica il comportamento di failover quando gli endpoint dual-stack non sono disponibili

Clienti interessati dall'aggiornamento per l'inclusione di IPv6

Quando abiliti IPv6 per Kinesis Video Streams, ci sono diverse aree in cui potresti dover aggiornare le configurazioni e le politiche esistenti per garantire la continuità della funzionalità.

Policy IAM e filtraggio degli indirizzi IP

Se utilizzi il filtraggio degli indirizzi IP di origine nelle politiche degli utenti, delle politiche dei ruoli o delle politiche basate sulle risorse IAM, devi aggiornare queste politiche per includere gli intervalli di indirizzi IPv6.

Important

Le policy IAM esistenti che utilizzano blocchi CIDR IPv4 in `IpAddress` o `NotIpAddress` condizioni non funzioneranno automaticamente con gli indirizzi IPv6. È necessario aggiungere esplicitamente intervalli IPv6 per mantenere il controllo degli accessi.

Esempio di aggiornamento della policy IAM per 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"
          ]
        }
      }
    }
  ]
}
```

Registrazione di log e monitoraggio

Gli indirizzi IPv6 hanno un formato diverso dagli indirizzi IPv4, il che può influire sui sistemi di registrazione, monitoraggio e analisi.

log

I registri conterranno gli indirizzi IPv6 nel campo quando le richieste vengono effettuate tramite IPv6. `sourceIpAddress` Aggiorna gli strumenti e gli script di analisi dei log per gestire i formati di indirizzi IPv6.

Esempio di indirizzo IPv6 nei log:

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

Risoluzione dei problemi

Problemi comuni

- Errori di connessione: verifica della connettività di rete IPv6 e della risoluzione DNS
- Errori SDK: assicurati di utilizzare versioni SDK compatibili che supportano endpoint dual-stack
- Problemi di autenticazione: conferma che le policy e le credenziali IAM funzionino con gli endpoint IPv6
- Endpoint non disponibile: se non esiste un endpoint dual-stack per il servizio o la regione, le richieste hanno esito negativo

Fasi di verifica

- Verifica che `AWS_USE_DUALSTACK_ENDPOINT=true` sia impostato o `use_dualstack_endpoint = true` sia presente nel tuo file di configurazione
- Verifica che i flag di configurazione IPv6 di Kinesis Video Streams SDK siano impostati correttamente
- Verifica la connettività di rete agli endpoint IPv6 AWS
- Esamina i registri delle applicazioni per verificare la presenza di messaggi di errore IPv6-specific
- Verifica che la tua regione supporti gli endpoint dual-stack per Kinesis Video Streams

Convalida della configurazione

Puoi verificare la configurazione dell'endpoint dual-stack controllando:

- Variabili di ambiente: `echo $AWS_USE_DUALSTACK_ENDPOINT`
- AWS file di configurazione: `cat ~/.aws/config | grep use_dualstack_endpoint`
- Proprietà JVM (Java): verifica le proprietà del sistema nei registri delle applicazioni

Riproduzione di Kinesis Video Streams

È possibile visualizzare uno streaming video Kinesis utilizzando i seguenti metodi:

- **GetMedia**— È possibile utilizzare l'GetMediaAPI per creare applicazioni personalizzate per elaborare Kinesis Video Streams. GetMedia è un'API in tempo reale con bassa latenza. Per creare un player che usi GetMedia, devi crearlo tu stesso. Per informazioni su come sviluppare un'applicazione che visualizzi un flusso video Kinesis utilizzando GetMedia, consulta [Trasmetti in streaming utilizzando la libreria parser](#).
- **HLS** — [HTTP Live Streaming \(HLS\)](#) è un protocollo di comunicazione per lo streaming HTTP-based multimediale standard del settore. È possibile utilizzare HLS per visualizzare uno streaming video Kinesis, per la riproduzione dal vivo o per visualizzare i video archiviati.

È possibile usare HLS per la riproduzione in tempo reale. La latenza è in genere compresa tra 3 e 5 secondi, ma può essere compresa tra 1 e 10 secondi, a seconda del caso d'uso, del lettore e delle condizioni della rete. Puoi utilizzare un player di terze parti (ad esempio [Google Shaka Player](#)) per visualizzare lo streaming video fornendo l'URL della sessione di streaming HLS, programmaticamente [Video.js](#) o manualmente. Puoi anche riprodurre il video inserendo l'URL della sessione di streaming HLS nella barra degli indirizzi del browser [Apple Safari](#) o Microsoft Edge. <https://www.microsoft.com/en-us/edge>

- **MPEG-DASH**— [Dynamic Adaptive Streaming over HTTP \(DASH\)](#), noto anche come MPEG-DASH, è un protocollo di streaming con bitrate adattivo che consente lo streaming di alta qualità di contenuti multimediali su Internet forniti da server Web HTTP convenzionali.

È possibile MPEG-DASH utilizzarlo per la riproduzione dal vivo. La latenza è in genere compresa tra 3 e 5 secondi, ma può essere compresa tra 1 e 10 secondi, a seconda del caso d'uso, del lettore e delle condizioni della rete. Puoi utilizzare un player di terze parti (come [dash.js](#) o [Google Shaka Player](#)) per visualizzare lo streaming video fornendo l'URL della sessione di MPEG-DASH streaming, programmaticamente o manualmente.

- **GetClip**— È possibile utilizzare l'GetClipAPI per scaricare una clip (in un file MP4) contenente i file multimediali su richiesta archiviati dal flusso video specificato nell'intervallo di tempo specificato. Per ulteriori informazioni, consulta la documentazione di riferimento delle API [GetClip](#).

Argomenti

- [Requisiti della traccia di riproduzione video](#)

- [Riproduzione video con HLS](#)
- [Riproduzione video con MPEG-DASH](#)

Requisiti della traccia di riproduzione video

Amazon Kinesis Video Streams supporta contenuti multimediali codificati in più formati. Se il tuo streaming video Kinesis utilizza un formato non supportato da una delle quattro API elencate di seguito, utilizza [GetMedia](#) o [GetMediaForFragmentList](#), poiché non hanno limitazioni relative al tipo di traccia.

Argomenti

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

GetClip requisiti

Per ulteriori informazioni sull'API, consulta [GetClip](#).

Descrizione della traccia 1	ID del codec della traccia 1	Descrizione della traccia 2	ID del codec della traccia 2
H.264 video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 video	V_MPEG4/ISO/AVC	Audio AAC	A_AAC
H.264 video	V_MPEG4/ISO/AVC	G.711 audio (solo) A-Law	A_MS/ACM
H.265 video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

⚠ Important

I codec private data (CPD) contenuti in ogni frammento contengono informazioni di inizializzazione specifiche del codec, come frame rate, risoluzione e profilo di codifica, necessarie per decodificare correttamente il frammento. Le modifiche CPD non sono supportate tra i frammenti di destinazione della clip risultante. Il CPD deve rimanere coerente su tutti i supporti interrogati, altrimenti verrà restituito un errore.

⚠ Important

Le modifiche alla traccia non sono supportate. Le tracce devono rimanere coerenti in tutti i file multimediali richiesti. Viene restituito un errore se i frammenti dello stream passano dall'avere solo video a contenere sia audio che video o se una traccia audio AAC viene modificata in una A-Law traccia audio.

GetDASHStreamingSessionURL requisiti

Per ulteriori informazioni sull'API, consulta [GetDASHStreamingSessionURL](#).

Descrizione della traccia 1	ID del codec della traccia 1	Descrizione della traccia 2	ID del codec della traccia 2
H.264 video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 video	V_MPEG4/ISO/AVC	Audio AAC	A_AAC
H.264 video	V_MPEG4/ISO/AVC	G.711 audio (solo) A-Law	A_MS/ACM
H.264 video	V_MPEG4/ISO/AVC	G.711 audio (solo) U-Law	A_MS/ACM
Audio AAC	A_AAC	N/A	N/A
H.265 video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

⚠ Important

I codec private data (CPD) contenuti in ogni frammento contengono informazioni di inizializzazione specifiche del codec, come frame rate, risoluzione e profilo di codifica, necessarie per decodificare correttamente il frammento. Le modifiche al CPD non sono supportate durante una sessione di streaming. Il CPD deve rimanere coerente su tutti i supporti interrogati.

⚠ Important

Le modifiche alla traccia non sono supportate. Le tracce devono rimanere coerenti in tutti i file multimediali richiesti. Lo streaming avrà esito negativo se i frammenti dello stream passano dall'avere solo video a contenuti audio e video o se una traccia audio AAC viene modificata in una A-Law traccia audio.

GetHLSStreamingSessionURL requisiti

Per ulteriori informazioni sull'API, consulta [GetHLSStreamingSessionURL](#).

HLS Mp4

Descrizione della traccia 1	ID del codec della traccia 1	Descrizione della traccia 2	ID del codec della traccia 2
H.264 video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 video	V_MPEG4/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	N/A	N/A
H.265 video	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

HLS TS

Descrizione della traccia 1	ID del codec della traccia 1	Descrizione della traccia 2	ID del codec della traccia 2
H.264 video	V_MPEG4/ISO/AVC	N/A	N/A
H.264 video	V_MPEG4/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	N/A	N/A

Note

I codec private data (CPD) contenuti in ogni frammento contengono informazioni di inizializzazione specifiche del codec, come frame rate, risoluzione e profilo di codifica, necessarie per decodificare correttamente il frammento. Sia per TS che per MP4, le modifiche al CPD sono supportate durante una sessione di streaming. Pertanto, i frammenti di una sessione possono avere informazioni diverse nel CPD senza interrompere la riproduzione. Per ogni sessione di streaming, sono consentite solo 500 modifiche al CPD.

Important

Le modifiche alla traccia non sono supportate. Le tracce devono rimanere coerenti in tutti i file multimediali richiesti. Lo streaming avrà esito negativo se i frammenti dello stream passano dall'avere solo video a contenuti audio e video o se una traccia audio AAC viene modificata in una A-Law traccia audio.

GetImages requisiti

Per ulteriori informazioni sull'API, consulta [GetImages](#).

Note

Il GetImages supporto deve contenere una traccia video nella traccia 1.

Riproduzione video con HLS

[HTTP Live Streaming \(HLS\)](#) è un protocollo di comunicazione per lo streaming HTTP-based multimediale standard del settore. È possibile utilizzare HLS per visualizzare uno streaming video Kinesis, per la riproduzione dal vivo o per visualizzare i video archiviati.

È possibile usare HLS per la riproduzione in tempo reale. La latenza è in genere compresa tra 3 e 5 secondi, ma può essere compresa tra 1 e 10 secondi, a seconda del caso d'uso, del lettore e delle condizioni della rete. Puoi utilizzare un player di terze parti (ad esempio [Google Shaka Player](#)) per visualizzare lo streaming video fornendo l'URL della sessione di streaming HLS, programmaticamente [Video.js](#) o manualmente. Puoi anche riprodurre il video inserendo l'URL della sessione di streaming HLS nella barra degli indirizzi del browser [Apple Safari](#) o Microsoft Edge. <https://www.microsoft.com/en-us/edge>

Per visualizzare uno streaming video Kinesis utilizzando HLS, create innanzitutto una sessione di streaming utilizzando [GetHLSStreamingSessionURL](#). Questa operazione restituisce un URL (contenente un token di sessione) per l'accesso alla sessione HLS. Puoi quindi usare l'URL in un lettore multimediale o in un'applicazione autonoma per visualizzare il flusso.

Important

Non tutti i file multimediali inviati a Kinesis Video Streams possono essere riprodotti tramite HLS. Consulta i [GetHLSStreamingSessionURL](#) requisiti di caricamento specifici.

Argomenti

- [Usa il AWS CLI per recuperare l'URL di una sessione di streaming HLS](#)
- [Esempio: usa HLS in HTML e JavaScript](#)
- [Risoluzione dei problemi relativi all'HLS](#)

Usa il AWS CLI per recuperare l'URL di una sessione di streaming HLS

La procedura seguente illustra come utilizzare l'URL di una sessione AWS CLI di streaming HLS per uno streaming video Kinesis.

Per le istruzioni di installazione, consulta la Guida per l'utente. [AWS Command Line Interface](#) Dopo l'installazione, [configura le AWS CLI](#) credenziali e l'area geografica.

In alternativa, apri il AWS CloudShell terminale, che ha AWS CLI installato e configurato. Per ulteriori informazioni, consultare la Guida [per l'AWS CloudShell utente](#).

Recupera l'endpoint URL HLS per il tuo streaming video Kinesis.

1. Digitate quanto segue nel terminale:

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

Riceverai una risposta simile alla seguente:

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

2. Effettua la richiesta dell'URL della sessione di streaming HLS all'endpoint restituito.

Live

Per la riproduzione dal vivo, la playlist multimediale HLS viene continuamente aggiornata con i contenuti multimediali più recenti non appena disponibili. Quando si riproduce questo tipo di sessione in un lettore multimediale, l'interfaccia utente in genere visualizza una notifica "live", senza alcun controllo di scorrimento per scegliere la posizione nella finestra di riproduzione da visualizzare.

Assicurati di caricare contenuti multimediali su questo stream quando esegui questo comando.

```
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

Per la riproduzione dal vivo, la riproduzione inizia da un orario di inizio specificato. La playlist multimediale HLS viene inoltre continuamente aggiornata con i contenuti multimediali più recenti non appena disponibili. La sessione continuerà a includere i file multimediali

appena inseriti fino alla scadenza della sessione o fino all'ora di fine specificata, a seconda dell'evento che si verifica per primo. Questa modalità è utile per poter avviare la riproduzione dal momento in cui viene rilevato un evento e continuare lo streaming live di contenuti multimediali che non sono ancora stati inseriti al momento della creazione della sessione.

Determina un timestamp di inizio.

Per questo esempio, utilizziamo il formato Unix Epoch time in secondi. Fai riferimento alla [sezione](#) Timestamp nella Guida per l' AWS Command Line Interface utente per ulteriori informazioni sulla formattazione dei timestamp.

Vedi per [UnixTime.org](https://unixtime.org/) uno strumento di conversione.

- 1708471800 è uguale al 20 febbraio 2024 15:30:00 GMT-08:00

In questo esempio, non specifichiamo un timestamp di fine, il che significa che la sessione continuerà a includere i contenuti multimediali appena inseriti fino alla scadenza della sessione.

Richiama l'GetHLSStreamingSessionURLAPI con la modalità di LIVE_REPLAY riproduzione e un selettore di frammenti HLS specificato. 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 LIVE_REPLAY \
  --hls-fragment-selector \
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800}"
```

On-demand

Per la riproduzione su richiesta, la playlist multimediale HLS contiene i file multimediali specificati dal selettore di frammenti HLS. Quando questo tipo di sessione viene riprodotto in un lettore multimediale, l'interfaccia utente in genere visualizza un comando di scorrimento per scegliere la posizione nella finestra di riproduzione da visualizzare.

Per creare un URL per una determinata sezione dello stream, stabilisci innanzitutto i timestamp di inizio e fine.

Per questo esempio, utilizziamo il formato Unix Epoch time in secondi. Fai riferimento alla [sezione](#) Timestamp nella Guida per l' AWS Command Line Interface utente per ulteriori informazioni sulla formattazione dei timestamp.

Vedi per [UnixTime.org](https://unixtime.org) uno strumento di conversione.

- 1708471800 è uguale al 20 febbraio 2024 15:30:00 GMT-08:00
- 1708471860 è uguale a 20 febbraio 2024 15:31:00 GMT-08:00

Richiama l'GetHLSStreamingSessionURLAPI con la modalità di ON_DEMAND riproduzione e un selettore di frammenti HLS specificato. 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, EndTimeStamp=1708471860}"
```

 Note

I timestamp devono trovarsi entro 24 ore l'uno dall'altro, come indicato nella documentazione. [HLSTimestampRange](#)

Riceverai una risposta simile alla seguente:

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

⚠ Important

Non condividere o archiviare questo token dove un'entità non autorizzata potrebbe accedervi. Il token fornisce l'accesso al contenuto dello stream. Proteggi il token con le stesse misure che useresti con le tue AWS credenziali.

Puoi utilizzare questo URL e qualsiasi player HLS per visualizzare lo stream HLS.

Ad esempio, usa il lettore multimediale VLC.

Puoi anche riprodurre HLS Stream inserendo l'URL della sessione di streaming HLS nella barra degli indirizzi del browser Apple Safari o Microsoft Edge.

Esempio: usa HLS in HTML e JavaScript

L'esempio seguente mostra come utilizzare l' AWS SDK for JavaScript v2 per recuperare una sessione di streaming HLS per uno streaming video Kinesis e riprodurla in una pagina web.

L'esempio mostra come riprodurre video con i lettori seguenti:

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

Visualizza il codice di esempio [completo](#) e [la pagina web ospitata in](#) GitHub. Questa pagina web statica semplifica i test e la sperimentazione con HLS e l' MPEG-DASH output dello streaming video di Amazon Kinesis. La pagina di esempio fornisce campi di input per i seguenti parametri:

- Regione AWS: la regione in cui si trova il tuo streaming video Kinesis
- Nome dello stream: il nome del tuo stream video Kinesis
- Modalità di riproduzione: la modalità di riproduzione HLS (LIVE, LIVE_REPLAY o ON_DEMAND)
- Tipo di selettore di frammenti: il metodo utilizzato per selezionare i frammenti (SERVER_TIMESTAMP o PRODUCER_TIMESTAMP)
- Numero del frammento: il numero del frammento iniziale (se applicabile)

- Formato del contenitore: il formato del contenitore multimediale (FRAGMENTED_MP4 o MPEG_TS)

L'applicazione recupera questi valori dalle caselle di input sulla pagina HTML e li utilizza per creare la richiesta di una sessione di streaming HLS che viene visualizzata sulla pagina.

Descrizione dettagliata degli argomenti:

- [Importa il AWS SDK JavaScript per browser](#)
- [Configura il client Kinesis Video Streams](#)
- [Recupera l'endpoint per la riproduzione HLS](#)
- [Configura il client multimediale archiviato di Kinesis Video Streams](#)
- [Recupera l'URL della sessione di streaming HLS](#)
- [Visualizza lo stream HLS sulla pagina web](#)

Importa il AWS SDK JavaScript per browser

Nella pagina web, includi il seguente tag di script per importare l' AWS SDK per JavaScript v2 nel progetto.

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

Per ulteriori informazioni, consulta l'[AWS SDK per la documentazione. JavaScript](#)

Configura il client Kinesis Video Streams

Per accedere allo streaming video con HLS, create e configurate innanzitutto il client Kinesis Video Streams. Per altri metodi di autenticazione, consulta [Impostazione delle credenziali in un browser Web](#).

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

L'applicazione recupera i valori necessari dalle caselle di input nella pagina HTML.

Recupera l'endpoint per la riproduzione HLS

Utilizzate il client Kinesis Video Streams per richiamare l'API per recuperare l'[GetDataEndpoint](#) endpoint.

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

Questo codice memorizza l'endpoint nella variabile. `hlsDataEndpoint`

Configura il client multimediale archiviato di Kinesis Video Streams

Nella configurazione del client per il client multimediale archiviato Kinesis Video Streams, specificate l'endpoint ottenuto nel passaggio precedente.

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

Recupera l'URL della sessione di streaming HLS

Utilizzate il client multimediale archiviato Kinesis Video Streams per richiamare l'[GetHLSStreamingSessionURL](#) API per recuperare l'URL di riproduzione HLS.

```
const getHLSStreamingSessionURLOptions = {
  StreamName: 'YourStreamName',
  PlaybackMode: 'LIVE'
};
```

```
const getHLSStreamingSessionURLResponse = await kinesisVideoArchivedMediaClient
    .getHLSStreamingSessionURL(getHLSStreamingSessionURLOptions)
    .promise();
const hlsUrl = getHLSStreamingSessionURLResponse.HLSStreamingSessionURL;
```

Visualizza lo stream HLS sulla pagina web

Una volta ottenuto l'URL della sessione di streaming HLS, forniscilo al lettore video. Il metodo per fornire l'URL al lettore video dipende dal lettore usato.

Video.js

Effettuate le seguenti operazioni per importare [Video.js](#) e inserire le relative classi CSS nello script del nostro browser:

```
<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>
```

Create un elemento video HTML per visualizzare il video:

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

Imposta l'URL HLS come sorgente dell'elemento video HTML:

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

Shaka

Per importare il player di [Google Shaka nello script del nostro browser](#), procedi come segue:

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

Crea un elemento video HTML per visualizzare il video:

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

Create un player Shaka specificando l'elemento video e chiamate il metodo load:

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

hls.js

Fate quanto segue per importare [hls.js](#) nel nostro script del browser:

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

Create un elemento video HTML per visualizzare il video:

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

Crea un player hls.js, assegnagli l'URL HLS e digli di riprodurlo:

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

Risoluzione dei problemi relativi all'HLS

Questa sezione descrive i problemi che potresti riscontrare quando utilizzi HTTP Live Streaming (HLS) con Kinesis Video Streams.

Problemi

- [Il recupero dell'URL della sessione di streaming HLS riesce, ma si verifica un problema di riproduzione nel lettore video](#)

- [Latenza troppo elevata tra il producer e il lettore](#)

Il recupero dell'URL della sessione di streaming HLS riesce, ma si verifica un problema di riproduzione nel lettore video

Questa situazione si verifica quando è possibile recuperare correttamente l'URL di una sessione di streaming HLS usando `GetHLSStreamingSessionURL`, ma non è possibile riprodurre il video fornendo l'URL a un lettore video.

Per risolvere questo problema, prova le soluzioni seguenti:

- Determina se lo streaming video viene riprodotto nella console Kinesis Video Streams. Esamina gli errori visualizzati nella console.
- Se la durata del frammento è meno di un secondo, aumentala a un secondo. Se la durata del frammento è troppo breve, il servizio potrebbe rallentare il lettore perché richiede frammenti video troppo frequentemente.
- Verificare che ogni URL di una sessione di streaming HLS venga usato da un solo lettore. Se più di un lettore usa un singolo URL di una sessione di streaming HLS, il servizio potrebbe ricevere troppe richieste e limitarle.
- Verifica che il tuo lettore supporti tutte le opzioni che stai specificando per la sessione di streaming HLS. Prova combinazioni diverse di valori per i parametri seguenti:
 - `ContainerFormat`
 - `PlaybackMode`
 - `FragmentSelectorType`
 - `DiscontinuityMode`
 - `MaxMediaPlaylistFragmentResults`

Alcuni lettori multimediali (ad esempio HTML5 e lettori mobile) in genere supportano solo HLS con il formato container fMP4. Altri lettori multimediali (come Flash e lettori personalizzati) potrebbero supportare HLS solo con il formato contenitore MPEG TS. Consigliamo di sperimentare con il `ContainerFormat` parametro per iniziare la risoluzione dei problemi.

- Verificare che ogni frammento abbia un numero consistente di tracce. Verifica che i frammenti nello stream non cambino tra una traccia audio e video e una sola traccia video. Verifica inoltre che le impostazioni dell'encoder (risoluzione e frequenza dei fotogrammi) non cambino tra i frammenti di ciascuna traccia.

Latenza troppo elevata tra il producer e il lettore

Questa situazione si verifica quando vi è una latenza troppo elevata dal momento dell'acquisizione del video alla sua riproduzione nel lettore.

Il video viene riprodotto tramite HLS frammento per frammento. Pertanto, la latenza non può essere inferiore alla durata dei frammenti. La latenza include anche il tempo necessario per il buffering e il trasferimento dei dati. Se la soluzione richiede una latenza inferiore a un secondo, usa l'API `GetMedia`.

È possibile modificare i parametri seguenti per ridurre la latenza complessiva, ma la modifica di questi parametri potrebbe comportare anche la riduzione della qualità video o l'aumento della frequenza di ripetizione del buffering.

- **Durata del frammento:** la durata del frammento è la quantità di video tra le divisioni dello stream, controllata dalla frequenza dei fotogrammi chiave generati dal codificatore video. Il valore consigliato è di un secondo. Una durata inferiore dei frammenti significa meno tempo di attesa per il completamento del frammento prima della trasmissione dei dati video al servizio. I frammenti più brevi vengono anche elaborati più velocemente dal servizio. Tuttavia, se la durata di frammento è troppo breve, aumentano le probabilità che il lettore esaurisca i contenuti e debba arrestarsi ed eseguire il buffering di altri contenuti. Se la durata del frammento è meno di 500 millisecondi, il producer potrebbe creare troppe richieste, che potrebbero pertanto essere limitate dal servizio.
- **Bitrate:** uno streaming video con un bitrate inferiore richiede meno tempo per leggere, scrivere e trasmettere. Tuttavia, un flusso video con una velocità in bit inferiore ha in genere una qualità video inferiore.
- **Numero di frammenti nelle playlist multimediali:** un lettore sensibile alla latenza dovrebbe caricare solo i frammenti più recenti in una playlist multimediale. La maggior parte dei giocatori inizia invece dal primo frammento. Riducendo il numero di frammenti nella playlist, riduci la distanza temporale tra i frammenti precedenti e quelli nuovi. Con una playlist di dimensioni inferiori, è possibile che un frammento venga saltato durante la riproduzione se si verifica un ritardo nell'aggiunta di nuovi frammenti alla playlist o se il lettore riceve una playlist aggiornata in ritardo. Ti consigliamo di utilizzare 3-5 frammenti e di utilizzare un lettore configurato per caricare solo i frammenti più recenti da una playlist.
- **Dimensione del buffer del lettore:** la maggior parte dei lettori video ha una durata minima del buffer configurabile, in genere con un'impostazione predefinita di 10 secondi. Per ottenere la latenza più bassa, è possibile impostare questo valore su 0 secondi. Tuttavia, ciò significa che il giocatore

esegue il rebuffer in caso di ritardo nella produzione di frammenti, poiché il giocatore non avrà alcun buffer per assorbire il ritardo.

- Il lettore «recupera il ritardo»: i lettori video in genere non recuperano automaticamente la riproduzione fino alla parte anteriore del buffer video se il buffer si riempie, ad esempio quando un frammento ritardato causa un ritardo nella riproduzione di frammenti. Un lettore personalizzato consente di evitare questa situazione, eliminando i fotogrammi o aumentando la velocità di riproduzione (ad esempio a 1,1x) per raggiungere la parte iniziale del buffer. Ciò causa una riproduzione irregolare o un aumento della velocità mentre il lettore recupera e la ripetizione del buffering potrebbe avvenire con maggiore frequenza se le dimensioni del buffer sono contenute.

Riproduzione video con MPEG-DASH

Per visualizzare uno streaming video Kinesis utilizzando MPEG-DASH, devi prima creare una sessione di streaming utilizzando [GetDASHStreamingSessionURL](#). Questa azione restituisce un URL (contenente un token di sessione) per accedere alla MPEG-DASH sessione. Puoi quindi usare l'URL in un lettore multimediale o in un'applicazione autonoma per visualizzare il flusso.

Uno streaming video di Amazon Kinesis presenta i seguenti requisiti per la fornitura di video tramite: MPEG-DASH

- Per i requisiti di riproduzione dei brani video in streaming, consulta [the section called “GetDASHStreamingSessionURL”](#)
- La conservazione dei dati deve essere superiore a 0.
- La traccia video di ogni frammento deve contenere dati privati del codec in Advanced Video Coding (AVC) per il formato e HEVC per il H.264 formato. H.265 [Per ulteriori informazioni, vedere la specifica 14496-15. MPEG-4 ISO/IEC](#) Per informazioni sull'adattamento dei dati di flusso a un dato formato, vedi i [contrassegni di adattamento NAL](#).
- [La traccia audio \(se presente\) di ogni frammento deve contenere dati privati del codec in formato AAC \(specifica AAC ISO/IEC 13818-7\) o in formato MS Wave.](#)

Esempio: utilizzo in HTML e MPEG-DASH JavaScript

L'esempio seguente mostra come recuperare una sessione di MPEG-DASH streaming per uno streaming video Kinesis e riprodurla in una pagina Web. L'esempio mostra come riprodurre video con i lettori seguenti:

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

Argomenti

- [Configurare il client Kinesis Video MPEG-DASH Streams per la riproduzione](#)
- [Recupera l'endpoint dei contenuti archiviati di Kinesis Video Streams per la riproduzione MPEG-DASH](#)
- [Recupera l'URL della sessione di streaming MPEG-DASH](#)
- [Visualizza il video in streaming con riproduzione MPEG-DASH](#)
- [Esempio completato](#)

Configurare il client Kinesis Video MPEG-DASH Streams per la riproduzione

Per accedere ai video in streaming con MPEG-DASH, crea e configura prima il client Kinesis Video Streams (per recuperare l'endpoint del servizio) e il client multimediale archiviato (per recuperare la sessione di streaming). MPEG-DASH L'applicazione recupera i valori necessari dalle caselle di input nella pagina HTML.

```
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);
```

Recupera l'endpoint dei contenuti archiviati di Kinesis Video Streams per la riproduzione MPEG-DASH

Dopo l'avvio dei client, recupera l'endpoint dei contenuti archiviati di Kinesis Video Streams in modo da poter recuperare l'URL della sessione di streaming come segue: MPEG-DASH

```
// 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);
});
```

Recupera l'URL della sessione di streaming MPEG-DASH

Quando disponi dell'endpoint dei contenuti archiviati, chiama l'[GetDASHStreamingSessionURL](#) API per recuperare l'URL della sessione di MPEG-DASH streaming nel modo seguente:

```
// 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);
  });
}
```

Visualizza il video in streaming con riproduzione MPEG-DASH

Quando hai l'URL della sessione di MPEG-DASH streaming, forniscilo al lettore video. Il metodo per fornire l'URL al lettore video dipende dal lettore usato.

L'esempio di codice seguente illustra come fornire l'URL della sessione di streaming a un lettore

[Google Shaka Player](#):

```
// 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');
}
```

L'esempio di codice seguente illustra come fornire l'URL della sessione di streaming a un lettore

[dash.js](#):

```
<!-- 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');
}
```

Esempio completato

È possibile [scaricare o visualizzare il codice di esempio completato](#) su GitHub.

Configurare le notifiche in Kinesis Video Streams

Quando un frammento multimediale è disponibile per il consumo, Kinesis Video Streams avvisa i clienti tramite Amazon Simple Notification Service notifiche (Amazon SNS).

Note

Amazon Kinesis Video Streams utilizza Amazon SNS Standard Topics per la comunicazione. Gli argomenti FIFO non sono attualmente supportati.

I seguenti argomenti spiegano come iniziare con le notifiche.

Argomenti

- [Gestisci le configurazioni delle notifiche](#)
- [Informazioni sui tag MKV del produttore](#)
- [Messaggi Amazon SNS](#)
- [Cross-account Pubblicazione delle notifiche Amazon SNS](#)

Gestisci le configurazioni delle notifiche

Per gestire le configurazioni delle notifiche, usa `UpdateNotificationConfiguration` e `DescribeNotificationConfiguration`. Per ulteriori informazioni, vedere di seguito.

UpdateNotificationConfiguration

Usa questa operazione API per aggiornare le informazioni di notifica per uno stream. Per ulteriori informazioni sulla `UpdateNotificationConfiguration` funzionalità, consulta [UpdateNotificationConfiguration](#) la Amazon Kinesis Video Streams Developer Guide.

Note

È necessario almeno un minuto per avviare la notifica dopo l'aggiornamento della configurazione della notifica. Attendi almeno un minuto prima di richiamare `PutMedia` dopo la chiamata di aggiornamento.

DescribeNotificationConfiguration

Usa questa API per descrivere una configurazione di notifica associata a uno stream. Per ulteriori informazioni sulla `DescribeNotificationConfiguration` funzionalità, consulta [DescribeNotificationConfiguration](#) la Amazon Kinesis Video Streams Developer Guide.

Informazioni sui tag MKV del produttore

È possibile utilizzare l'SDK per produttori di Kinesis Video Streams per etichettare specifici frammenti di interesse esponendo un'operazione API nell'SDK. Guarda un esempio di come funziona [in questa sezione di codice](#). Dopo aver chiamato questa API, l'SDK aggiungerà una serie di tag MKV predefiniti insieme ai dati del frammento. Kinesis Video Streams riconoscerà questi tag MKV speciali e avvierà notifiche per i frammenti etichettati.

Tutti i metadati dei frammenti forniti insieme ai tag MKV di notifica verranno pubblicati come parte del payload degli argomenti di Amazon SNS.

Sintassi per i tag MKV del produttore

```
|+ 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
```

Limiti dei tag MKV

La tabella seguente elenca le limitazioni associate ai tag dei metadati. Se il limite dei tag di metadati è regolabile, puoi richiedere un aumento tramite il tuo account manager.

Limite	Valore massimo	Regolabile
Lunghezza opzionale della chiave di metadati	128	No
Lunghezza opzionale del valore dei metadati	256	No
Numero massimo di metadati opzionali	10	Sì

Messaggi Amazon SNS

Questo argomento contiene ulteriori informazioni sui messaggi Amazon SNS e sui payload tematici.

Argomenti

- [Payload tematico di Amazon SNS](#)
- [Visualizza i tuoi messaggi Amazon SNS](#)

Payload tematico di Amazon SNS

Qualsiasi notifica avviata tramite il flusso di lavoro precedente fornirà il payload dell'argomento Amazon SNS, come mostrato nell'esempio seguente. Questo esempio è un messaggio Amazon SNS che si verifica dopo aver consumato i dati di notifica da una coda Amazon Simple Queue Service (Amazon SQS).

```
{
  "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  
  }  
}
```

Visualizza i tuoi messaggi Amazon SNS

Non puoi leggere i messaggi direttamente da un argomento di Amazon SNS perché non esiste un'API per farlo. Per visualizzare i messaggi, iscriviti a una coda SQS all'argomento SNS o scegli qualsiasi altra [destinazione supportata da Amazon SNS](#). Tuttavia, l'opzione più efficiente per visualizzare i messaggi è utilizzare Amazon SQS.

Per visualizzare i tuoi messaggi Amazon SNS utilizzando Amazon SQS

1. Crea una coda [Amazon SQS](#).
2. Da Console di gestione AWS, apri l'argomento Amazon SNS impostato come destinazione in. NotificationConfiguration
3. Scegli Crea abbonamento, quindi scegli la coda Amazon SQS creata nel primo passaggio.
4. Esegui una PutMedia sessione con la configurazione delle notifiche abilitata e con i tag MKV di notifica aggiunti ai frammenti.
5. Scegli la coda Amazon SQS nella console Amazon SQS, quindi seleziona Invia e ricevi messaggi per la coda Amazon SQS.

6. Effettua un sondaggio per i messaggi. Questo comando dovrebbe mostrare tutte le notifiche generate dalla PutMedia sessione. Per informazioni sul polling, consulta [Amazon SQS short and long polling](#).

Cross-account Pubblicazione delle notifiche Amazon SNS

Per pubblicare notifiche Amazon SNS su un argomento in un AWS account diverso, devi configurare policy basate sull'identità e sulle risorse. Questa configurazione consente a Kinesis Video Streams di pubblicare notifiche da un account a un argomento di Amazon SNS in un altro account.

Identity-based configurazione delle politiche

Il ruolo o l'utente IAM che chiama l'PutMediaAPI deve disporre sns:Publish delle autorizzazioni per l'argomento Amazon SNS con più account. Aggiungi la seguente dichiarazione politica alla policy basata sull'identità:

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 configurazione della politica

L'argomento Amazon SNS nell'account di destinazione deve avere una politica di accesso basata sulle risorse che consenta all'account di origine di pubblicare messaggi. Configura la politica di accesso agli argomenti di Amazon SNS come segue:

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"
    }
  ]
}
```

Sostituiscila <kvs_streams_account_id> con l'ID dell' AWS account in cui si trovano i tuoi stream Kinesis Video Streams e <sns_topic_arn> con l'ARN del tuo argomento Amazon SNS.

Requisiti e considerazioni

- Affinché la pubblicazione su più account funzioni, è necessario configurare sia la policy basata sull'identità (nell'account di origine) che la policy basata sulle risorse (nell'account di destinazione).
- Il ruolo IAM utilizzato per PutMedia le operazioni deve includere le `sns:Publish` autorizzazioni, anche quando si utilizzano certificati IoT con alias di ruolo.

Estrarre immagini dai flussi video

Puoi utilizzare le API e gli SDK di Amazon Kinesis Video Streams per eseguire l'estrazione di immagini su richiesta e l'estrazione automatica delle immagini in tempo reale dai tuoi stream video Kinesis. Puoi utilizzare queste immagini per applicazioni di riproduzione avanzate come miniature o scrubbing avanzato o per l'uso in flussi di lavoro di apprendimento automatico.

Kinesis Video Streams supporta l'estrazione di immagini dai flussi video in due modi:

- [On-demand Generazione di immagini](#) : utilizza l'[GetImages](#) API per estrarre una o più immagini dai video archiviati in Kinesis Video Streams.
- [the section called “Real-time generazione di immagini”](#) - Configura Kinesis Video Streams per estrarre automaticamente le immagini dai dati video in tempo reale in base ai tag di frammento del video mentre viene inserito e invia le immagini a un bucket S3.

Generazione automatica di immagini in tempo reale

Amazon Kinesis Video Streams offre la capacità di transcodificare e distribuire immagini. Kinesis Video Streams estrae automaticamente le immagini dai dati video in tempo reale e le invia al bucket Amazon S3 specificato. L'implementazione dell'estrazione automatica delle immagini in tempo reale prevede i seguenti passaggi:

- Creazione di un bucket S3 per ricevere le immagini generate.
- Configurazione della proprietà [ImageGenerationConfiguration](#) stream che indica a Kinesis Video Streams come creare le immagini e dove inviarle.
- Aggiungere tag di generazione delle immagini: Kinesis Video Streams genera immagini solo utilizzando frammenti che hanno il tag di generazione delle immagini. Questi tag vengono aggiunti quando si carica un video utilizzando il Kinesis Video Streams Producer SDK insieme al metodo `putKinesisVideoEventMetadata`

Le seguenti procedure forniscono le istruzioni per completare ognuno di questi passaggi.

Se utilizzi una chiave gestita dal cliente, assicurati che il ruolo che esegue le `PutMedia` chiamate (uploader) disponga delle seguenti autorizzazioni necessarie per crittografare e decrittografare i dati e per accedere al bucket Amazon S3.

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

Per ulteriori informazioni, consulta [the section called “Come posso iniziare a usare la crittografia lato server?”](#).

Per configurare la destinazione delle immagini generate

1. Crea un bucket di destinazione S3 in cui verranno inviate le immagini.

Segui la Guida per l'utente di [Amazon S3](#) per creare un bucket Amazon S3.

Annota l'URI del bucket, che ti servirà nella fase successiva dell'aggiornamento della configurazione di generazione delle immagini dello stream.

2. Verifica di aver AWS CLI installato e configurato. Per ulteriori informazioni, consulta la Guida per [AWS Command Line Interface l'utente per la versione 2](#).
3. Crea un nuovo file chiamato `update-image-generation-input.json` con il seguente contenuto come input. Aggiorna i valori segnaposto con i valori che desideri utilizzare. Per i valori massimi e minimi supportati, consulta l'[UpdateImageGenerationConfigurationAPI](#).

```
{
  "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. Aggiorna la configurazione di generazione delle immagini dello stream utilizzando l'[UpdateImageGenerationConfiguration](#) API e allegando il file JSON come input, come mostrato nel comando seguente. Nota che il percorso del file punta al file nella directory corrente.

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

5. In caso di successo, viene restituita una risposta vuota e non viene stampato nulla nel terminale.

Note

È necessario almeno 1 minuto per avviare il flusso di lavoro di generazione dell'immagine dopo l'aggiornamento della configurazione di generazione dell'immagine. Attendi almeno 1 minuto prima di caricare il video nel tuo stream.

6. Verifica le impostazioni di configurazione. Usa AWS CLI per chiamare l'[DescribeImageGenerationConfiguration](#) API per il tuo stream.

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

Kinesis Video Streams genererà e distribuirà immagini solo per i frammenti che hanno il tag di generazione delle immagini. Tutti i metadati aggiuntivi dei frammenti forniti insieme ai tag di generazione delle immagini di Amazon S3 verranno salvati come metadati Amazon S3.

Note

I tag di generazione di immagini si riferiscono ai tag di metadati dei frammenti e non ai tag a livello di stream.

Important

I tag di generazione delle immagini vengono conteggiati ai fini del limite dei tag framment-metadati. Per ulteriori informazioni, consulta [the section called “Quote dei servizi di metadati di streaming”](#).

Di seguito è riportato un esempio di come appare la struttura dei tag di metadati dei frammenti utilizzando l'utilità. `mkvinfo` Il tag di generazione dell'immagine è un semplice tag MKV con una chiave di `AWS_KINESISVIDEO_IMAGE_GENERATION` e nessun valore. Per ulteriori informazioni, consulta Esempio di tag [video](#) nella documentazione di Matroska.

```
|+ 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
```

Aggiungere tag di generazione di immagini ai frammenti

Kinesis Video Streams genera e fornisce immagini solo per i frammenti che hanno il tag di generazione delle immagini. Kinesis Video Streams riconosce questi tag MKV speciali e avvia il flusso di lavoro di generazione delle immagini in base alla configurazione di elaborazione delle immagini dello stream.

Quando si utilizza Kinesis Video Streams Producer SDK per caricare contenuti multimediali, si utilizza il `putKinesisVideoEventMetadata` metodo per aggiungere il tag di generazione dell'immagine a ciascun frammento a cui si desidera applicare il tag. Un nuovo frammento inizia quando `putFrame` viene richiamato con un frame contenente il flag. `keyframe`

Se stai caricando un video preregistrato, potrebbe essere caricato a una velocità diversa da quella con cui è stato registrato, a seconda della velocità della tua rete. Ti consigliamo di utilizzare il timestamp di Producer per configurare la generazione di immagini se desideri generare immagini a intervalli regolari in base ai timestamp originali del video e non utilizzare i timestamp del server generati in base alla frequenza con cui Amazon Kinesis Video Streams ha ricevuto il tuo video.

Per visualizzare un esempio completo di questo codice, consulta l'esempio di codice in.

[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);
```

Gli elementi del codice di esempio per l'impostazione dei frame di esempio sono spiegati come segue:

1. Ogni frammento deve terminare con la fine di fragment (`eofr`). Questa dichiarazione dice che ogni volta che si riceve un nuovo fotogramma chiave, che indica l'inizio del fotogramma successivo, si inserisce un elemento `eofr` prima di aggiungere il fotogramma successivo allo stream.
2. Inserisci il frame corrente nello stream.
3. Aggiungi il tag di generazione dell'immagine. Il `putKinesisVideoEventMetadata` metodo può essere richiamato in qualsiasi momento dopo la `putFrame(keyFrame)` chiamata e prima del `putFrame(eofr)`. Deve essere chiamato al massimo una volta per frammento. Poiché ogni frammento avrà un solo fotogramma chiave, in questo momento lo chiamiamo per semplicità. Il valore restituito per `putKinesisVideoEventMetadata` viene controllato per un codice di successo (0).
4. Aggiungi altri metadati personalizzati dei frammenti, che Kinesis Video Streams trasformerà in metadati oggetto Amazon S3.
5. Termina il frammento finale in questa sessione di caricamento.

Utilizzo degli esempi per aggiungere tag di generazione di immagini

È possibile utilizzare l'SDK `kvs_gstreamer_audio_video_sample` di C++ Producer se si desidera un'opzione della riga di comando per aggiungere tag di generazione di immagini. Attivate questa funzionalità aggiungendo l'-e `both` argomento -e `image or`, come mostrato nell'esempio seguente.

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

Per ulteriori informazioni su questa applicazione di esempio, consulta [Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin e JNI README in.](#) GitHub

Percorso dell'oggetto Amazon S3 (immagine)

Il percorso dell'oggetto S3 descrive la posizione nel bucket S3 configurato in cui verrà consegnata l'immagine generata. Utilizza il seguente formato:

```
ImagePrefix_AccountID_StreamName_ImageTimecode_RandomID.file-extension
```

Gli elementi del percorso dell'oggetto sono definiti come segue:

- *ImagePrefix*- Valore di `AWS_KINESISVIDEO_IMAGE_PREFIX` se presente.
- *AccountID*- L' Account AWS ID con cui viene creato lo stream.
- *StreamName*- Nome dello stream da cui viene generata l'immagine.
- *ImageTimecode*- Codice temporale dell'epoca (in millisecondi) nel frammento in cui viene generata l'immagine.
- *RandomID*- GUID casuale.
- *file-extension*- JPG o PNG in base al formato di immagine richiesto.

In questo esempio, il percorso dell'oggetto per le immagini generate sarà il seguente:

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

Recupero dei metadati delle immagini

È possibile utilizzare la console S3 o la CLI per recuperare i metadati per le immagini generate.

Kinesis Video Streams imposta il numero del frammento, il timestamp del produttore e del server e i metadati del tipo di contenuto dell'immagine generata, tutti formattati come metadati degli oggetti Amazon S3. Se sono presenti tag MKV aggiuntivi, tali tag verranno aggiunti anche come metadati

degli oggetti Amazon S3. L'esempio seguente mostra come utilizzare il comando API `head-object` di Amazon S3 per recuperare i metadati dell'oggetto. La risposta include i metadati creati da Kinesis Video Streams.

```
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"  
  }  
}
```

Per ulteriori informazioni sui metadati degli oggetti S3, vedere <https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMetadata.html>

Raccomandazioni relative agli URI di Amazon S3 per la protezione dalla limitazione

Se scrivi migliaia di immagini su Amazon S3, c'è il rischio di limitazione. Per ulteriori informazioni, consulta [S3 Prefix Put Request Limits](#).

Un prefisso Amazon S3 inizia con un limite PUT di 3.500 richieste PUT al secondo e aumenterà gradualmente nel tempo per ottenere prefissi univoci. Evita di utilizzare date e ore come prefissi di Amazon S3. I dati con codifica temporale influiranno su un prefisso alla volta e cambieranno regolarmente, invalidando le precedenti scalabilità dei prefissi.

Per consentire una scalabilità più rapida e coerente di Amazon S3, consigliamo di aggiungere un prefisso casuale, ad esempio un codice esadecimale o un UUID, all'URI di destinazione di Amazon S3. Ad esempio, i prefissi in codice esadecimale suddivideranno naturalmente le richieste in modo casuale tra 16 prefissi diversi (un prefisso per ogni carattere esadecimale univoco), il che consentirà 56.000 richieste PUT al secondo dopo il ridimensionamento automatico di Amazon S3.

Risoluzione dei problemi

Immagini non inviate al bucket Amazon S3

Per risolvere questo problema, ci sono alcune cose a cui prestare attenzione:

1. Autorizzazioni mancanti
2. La configurazione per la generazione dell'immagine non è corretta
3. Il tag non è stato aggiunto al frammento

Autorizzazioni mancanti

Se utilizzi una chiave KMS gestita dal cliente, assicurati che il ruolo che esegue le PutMedia chiamate (uploader) disponga delle autorizzazioni di crittografia e decrittografia appropriate e abbia accesso al bucket Amazon S3, come segue:

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

Per ulteriori informazioni, consulta [the section called “Come posso iniziare a usare la crittografia lato server?”](#).

Verifica la destinazione

Usa il AWS CLI per chiamare l' DescribeImageGenerationConfiguration API per il tuo stream.

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

Controlla DestinationConfig la risposta e conferma che sia corretta.

Verifica che il tag di generazione dell'immagine sia stato aggiunto al frammento

1. Verifica che la putKinesisVideoEventMetadata chiamata sia riuscita.

Il putKinesisVideoEventMetadata metodo restituisce un codice di stato 0 in caso di successo. Si consiglia di verificare i valori restituiti dalle funzioni per 0. Se viene restituito

un codice di stato diverso da zero, convertilo in esadecimale e controlla [the section called “Riferimento al codice di errore”](#) per ulteriori informazioni.

Assicurati di aver attivato i log degli errori e rivedi i log per eventuali altri errori nell'applicazione. Rivedi e confronta il pattern di chiamata di invio dei frame della tua applicazione con l'implementazione consigliata: [Aggiungere tag di generazione di immagini ai frammenti](#).

2. Verifica il file MKV generato localmente

Verifica che il Producer SDK o l'applicazione di esempio abbiano aggiunto correttamente i tag.

- a. Imposta la variabile di ambiente `KVS_DEBUG_DUMP_DATA_FILE_DIR`. Se questo valore è impostato, Producer SDK scrive i file multimediali che avrebbe inviato a Kinesis Video Streams nella posizione specificata.

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

Note

L'SDK non creerà una nuova directory se il percorso non esiste. Se necessario, crea la cartella.

- b. Esegui nuovamente l'applicazione. Dovresti vedere `.mkv` i file che vengono scritti nella directory di output specificata.
- c. Verifica il contenuto per assicurarti che il tag sia presente utilizzando MKVToolNix o altro software.
 - i. Installa MKVToolNix: `brew install mkvtoolnix`
 - ii. Esegui MKVToolNix con uno dei `.mkv` file nella directory di output.

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

- iii. Rivedi l' MKVToolNix output. Se il metodo `KinesisVideoStream::PutFragmentMetadata` producer SDK è stato richiamato correttamente, dovresti vedere il seguente tag MKV.

```
|+ Tags
| + Tag
| + Simple
```

| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION

 Note

I tag appartengono al Cluster precedente.

- d. Se il tag MKV non è presente, assicuratevi che il metodo `KinesisVideoStream::PutEventMetadata` producer SDK sia stato chiamato con l'`STREAM_EVENT_TYPE_IMAGE_GENERATION` argomento e che abbia restituito un codice di successo (0).

Accedi all'analisi video

Questa sezione contiene informazioni su come accedere all'analisi video utilizzando la libreria parser e Amazon CloudWatch.

Argomenti

- [Utilizza i metadati incorporati in un flusso video Kinesis](#)
- [Guarda l'output delle telecamere usando la libreria parser](#)
- [Monitoraggio dei flussi video di Amazon Kinesis](#)
- [Limiti dei metadati di streaming](#)

Utilizza i metadati incorporati in un flusso video Kinesis

Per utilizzare i metadati in un flusso video Kinesis, utilizza un'implementazione di: `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");
    }
}
```

Questa interfaccia si trova nella classe [FragmentMetadataVisitor](#) nella [Guarda l'output delle telecamere usando la libreria parser](#).

La classe `FragmentMetadataVisitor` contiene un'implementazione di `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();
}
}
```

La classe `KinesisVideoRendererExample` contiene un esempio di come usare un `BasicMkvTagProcessor`. Nel seguente esempio, un `BasicMkvTagProcessor` viene aggiunto a `MediaProcessingArguments` di un'applicazione:

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

Il metodo `BasicMkvTagProcessor.process` viene chiamato quando arrivano i metadati del frammento. Puoi recuperare i metadati accumulati con `GetTags`. Per recuperare un singolo elemento di metadati, prima chiama `clear` per cancellare i metadati raccolti, quindi recupera nuovamente gli elementi di metadati.

Guarda l'output delle telecamere usando la libreria parser

La libreria `Kinesis Video Stream Parser` è un set di strumenti che potete utilizzare nelle applicazioni Java per utilizzare i dati MKV in un flusso video Kinesis.

La libreria include gli strumenti seguenti:

- [StreamingMkvReader](#): questa classe legge gli elementi MKV specificati da un flusso video.
- [FragmentMetadataVisitor](#): questa classe recupera i metadati relativi ai frammenti (elementi multimediali) e alle tracce (singoli flussi di dati che contengono informazioni multimediali, ad esempio audio o sottotitoli).
- [OutputSegmentMerger](#): questa classe unisce frammenti consecutivi o blocchi in un flusso video.

- [KinesisVideoExample](#): Questa è un'applicazione di esempio che mostra come utilizzare la libreria di parser di streaming video Kinesis.

La libreria include inoltre dei test che illustrano come vengono utilizzati gli strumenti.

Prerequisiti

È necessario disporre di quanto segue per esaminare e utilizzare la libreria di parser per flussi video Kinesis:

- Un account Amazon Web Services (AWS). Se non ne hai già uno Account AWS, consulta [the section called “Registrati per un Account AWS”](#).
- Un ambiente di sviluppo integrato (IDE) Java, come [Eclipse Java Neon](#) o [JetBrains IntelliJ Idea](#).
- Java 11, ad esempio [Amazon Corretto 11](#).

Scarica il codice

In questa sezione, potrai scaricare la libreria Java, testare il codice e importare il progetto nell'IDE Java.

Per i prerequisiti e altri dettagli su questa procedura, consulta [the section called “Trasmetti in streaming utilizzando la libreria parser”](#).

1. Create una directory e clonate il codice sorgente della libreria dal 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. Apri l'IDE Java che stai utilizzando (ad esempio, [Eclipse](#) o [IntelliJ IDEA](#)) e importa il progetto Apache Maven che hai scaricato:
 - In Eclipse: scegliere File, Import (Importa), Maven, Existing Maven Projects (Progetti Maven esistenti) e accedere alla cartella `kinesis-video-streams-parser-lib`.
 - In IntelliJ Idea: scegliere Import (Importa). Andare al file `pom.xml` nella radice del pacchetto scaricato.

Per ulteriori informazioni, consulta la documentazione IDE correlata.

Esamina il codice

In questa sezione, puoi esaminare la libreria Java, testare il codice e scoprire come utilizzare gli strumenti della libreria nel codice.

La libreria Kinesis Video Stream Parser contiene i seguenti strumenti:

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

StreamingMkvReader

Questa classe legge gli elementi MKV specificati da un flusso in modalità senza blocchi.

L'esempio di codice seguente (da `FragmentMetadataVisitorTest`) mostra come creare e utilizzare un `StreamingMkvReader` per recuperare gli oggetti `MkvElement` da un flusso di input chiamato `inputStream`:

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

FragmentMetadataVisitor

Questa classe recupera i metadati per i frammenti (elementi multimediali) e traccia i singoli flussi di dati contenenti informazioni multimediali, come i dati privati del codec, la larghezza o l'altezza dei pixel.

L'esempio di codice seguente (dal file `FragmentMetadataVisitorTest`) illustra come utilizzare `FragmentMetadataVisitor` per recuperare i dati da un oggetto `MkvElement`:

```

FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputSteamParserByteSource(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++;
            }
        }
    }
}

```

L'esempio precedente mostra i seguenti modelli di codifica:

- Creare un `FragmentMetadataVisitor` per analizzare i dati e un [StreamingMkvReader](#) per inserire i dati.
- Per ogni `MkvElement` nel flusso, verificare se i relativi metadati sono di tipo `SIMPLEBLOCK`.
- Se sì, recuperare il `MkvDataElement` dal `MkvElement`.

- Recuperate il `Frame` (dati multimediali) dal `MkvDataElement`.
- Recuperare il `MkvTrackMetadata` relativo al `Frame` dal `FragmentMetadataVisitor`.
- Recuperare e verificare i dati seguenti dagli oggetti `Frame` e `MkvTrackMetadata`:
 - Il numero di traccia.
 - L'altezza dei pixel del fotogramma.
 - La larghezza dei pixel del fotogramma.
 - L'ID codec relativo al codec utilizzato per la codifica del fotogramma.
 - L'arrivo del fotogramma nell'ordine corretto. Verificate che il numero di traccia del frame precedente, se presente, sia inferiore a quello del frame corrente.

Per utilizzare `FragmentMetadataVisitor` nel progetto, inviare gli oggetti `MkvElement` al visitatore tramite il relativo metodo `accept`:

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

Questa classe unisce i metadati di diverse tracce del flusso in un flusso con un singolo segmento.

L'esempio di codice seguente (dal file `FragmentMetadataVisitorTest`) illustra come utilizzare `OutputSegmentMerger` per unire i metadati della traccia da un array di byte chiamato `inputBytes`:

```
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);
    }
}
```

L'esempio precedente mostra i seguenti modelli di codifica:

- Creare un [FragmentMetadataVisitor](#) per recuperare i metadati dal flusso.
- Creare un flusso di output per ricevere i metadati uniti.
- Creare un `OutputSegmentMerger`, tramite l'invio del `ByteArrayOutputStream`.
- Creare un `CompositeMkvElementVisitor` contenente i due visitatori.
- Creare un `InputStream` che punti al file specificato.
- Unire ciascun elemento nei dati di input nel flusso di output.

KinesisVideoExample

Questa è un'applicazione di esempio che mostra come utilizzare la libreria Kinesis Video Stream Parser.

Questa classe effettua le seguenti operazioni:

- Crea un flusso video Kinesis. Se un flusso con il nome specificato è già esistente, tale flusso verrà eliminato e ricreato.
- Chiamate [PutMedia](#) per lo streaming di frammenti video nel flusso video Kinesis.
- Chiamate [GetMedia](#) per lo streaming di frammenti video dal flusso video Kinesis.
- Utilizza un [StreamingMkvReader](#) per analizzare i frammenti restituiti sul flusso e un [FragmentMetadataVisitor](#) per registrare i frammenti.

Eliminazione e nuova creazione del flusso

Il seguente esempio di codice (dal `StreamOps.java` file) elimina un determinato flusso video Kinesis:

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

Il seguente esempio di codice (tratto dal `StreamOps.java` file) crea un flusso video Kinesis con il nome specificato:

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

Chiama PutMedia

Il seguente esempio di codice (tratto dal `PutMediaWorker.java` file) chiama [PutMedia](#) lo stream:

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

Chiama GetMedia

Il seguente esempio di codice (tratto dal `GetMediaWorker.java` file) chiama [GetMedia](#) lo stream:

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

Analizza il risultato GetMedia

In questa sezione viene descritto come utilizzare [StreamingMkvReader](#), [FragmentMetadataVisitor](#) e [CompositeMkvElementVisitor](#) per analizzare, salvare sul file e registrare i dati restituiti da `GetMedia`.

Leggi l'output di with GetMedia StreamingMkvReader

Il seguente esempio di codice (tratto dal `GetMediaWorker.java` file) crea un [StreamingMkvReader](#) e lo utilizza per analizzare il risultato dell'[GetMedia](#) operazione:

```
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);
}
```

Nell'esempio di codice precedente, il [StreamingMkvReader](#) recupera gli oggetti `MKVElement` dal payload dei risultati `GetMedia`. Nella sezione successiva, gli elementi vengono inviati a un [FragmentMetadataVisitor](#).

Recupera i frammenti con FragmentMetadataVisitor

Gli esempi di codice seguenti (dai file `KinesisVideoExample.java` e `StreamingMkvReader.java`) creano un [FragmentMetadataVisitor](#). Gli oggetti `MkvElement` iterati dal [StreamingMkvReader](#) vengono quindi inviati al visitatore tramite il metodo `accept`.

da *KinesisVideoExample.java*:

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

da *StreamingMkvReader.java*:

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

Registrazione degli elementi e scrittura in un file

L'esempio di codice seguente (dal file `KinesisVideoExample.java`) crea gli oggetti seguenti e li restituisce come parte del valore restituito della funzione `GetMediaProcessingArguments`:

- Un `LogVisitor` (un'estensione di `MkvElementVisitor`) che effettua la scrittura nel log di sistema.
- Un `OutputStream` che scrive i dati in entrata su un file MKV.
- Un `BufferedOutputStream` che effettua il buffer dei dati associati all'`OutputStream`.
- Un [the section called “OutputSegmentMerger”](#) che unisce gli elementi consecutivi nel risultato `GetMedia` con la stessa traccia e gli stessi dati EBML.
- A `CompositeMkvElementVisitor` che compone il [FragmentMetadataVisitor](#), [the section called “OutputSegmentMerger”](#), e `LogVisitor` in un unico elemento visitatore.

```
//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);
```

Gli argomenti di elaborazione multimediale vengono quindi passati a `GetMediaWorker`, che a sua volta viene passato a `theExecutorService`, che esegue il worker su un thread separato:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getMkvElementVisitor());
```

```
executorService.submit(getMediaWorker);
```

Esegui il codice

La libreria Kinesis Video Stream Parser contiene strumenti destinati all'uso nei tuoi progetti. Il progetto contiene test di unità degli strumenti che puoi eseguire per verificare l'installazione.

Nella libreria sono inclusi i seguenti test di unità:

- mkv
 - ElementSizeAndOffsetVisitorTest
 - MkvValueTest
 - StreamingMkvReaderTest
- utilities
 - FragmentMetadataVisitorTest
 - OutputSegmentMergerTest

Monitoraggio dei flussi video di Amazon Kinesis

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Kinesis Video Streams e delle tue soluzioni. AWS Ti consigliamo di raccogliere dati di monitoraggio da tutte le parti della AWS soluzione per aiutarti a eseguire il debug di un errore multipunto, se si verifica. Prima di iniziare a monitorare Amazon Kinesis Video Streams, ti consigliamo di creare un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno utilizzati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

Dopo aver definito gli obiettivi di monitoraggio e creato il piano di monitoraggio, il passaggio successivo consiste nello stabilire una linea di base per le normali prestazioni di Amazon Kinesis

Video Streams nel tuo ambiente. È necessario misurare le prestazioni di Amazon Kinesis Video Streams in momenti diversi e in diverse condizioni di carico. Mentre monitori Amazon Kinesis Video Streams, archivia una cronologia dei dati di monitoraggio che hai raccolto. Puoi confrontare le prestazioni attuali di Amazon Kinesis Video Streams con questi dati storici per aiutarti a identificare i normali modelli di prestazioni e le anomalie prestazionali e a ideare metodi per risolvere i problemi che potrebbero sorgere.

Argomenti

- [Monitora le metriche di Amazon Kinesis Video Streams con CloudWatch](#)
- [Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch](#)
- [Registra le chiamate API Amazon Kinesis Video Streams con AWS CloudTrail](#)

Monitora le metriche di Amazon Kinesis Video Streams con CloudWatch

Puoi monitorare uno streaming video Kinesis utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi da Amazon Kinesis Video Streams in metriche leggibili e quasi in tempo reale. Queste statistiche vengono registrate per un periodo di 15 mesi in modo da poter accedere alle informazioni cronologiche e avere una prospettiva migliore sulle prestazioni della tua applicazione o del tuo servizio web.

Nella console [Amazon Kinesis Video Streams](#), puoi visualizzare le CloudWatch metriche per uno streaming video Amazon Kinesis in due modi:

- Nella pagina Dashboard, scegli la scheda Streams video nella sezione Account-level Metriche per la regione corrente.
- Scegliere la scheda Monitoring (Monitoraggio) nella pagina dei dettagli del flusso video.

Amazon Kinesis Video Streams fornisce le seguenti metriche:

Metrica	Description
ArchivedFragmentsConsumed.Media	Il numero di punti di quota multimediali per frammenti utilizzati da tutte le API. Per una spiegazione del concetto di punti di quota, vedi. the section called “Fragment-metadata e quote di frammenti multimediali”

Metrica	Description
	Unità: numero
ArchivedFragmentsConsumed.Metadata	Il numero di punti di quota dei metadati dei frammenti utilizzati da tutte le API. Per una spiegazione del concetto di punti di quota, vedere. the section called “Fragment-metadata e quote di frammenti multimediali” Unità: numero
PutMedia.Requests	Il numero di richieste PutMedia API per un determinato stream. Unità: numero
PutMedia.IncomingBytes	Il numero di byte ricevuti come parte dello PutMedia stream. Unità: byte
PutMedia.IncomingFragments	Il numero di frammenti completi ricevuti come parte dello PutMedia stream. Unità: numero
PutMedia.IncomingFrames	Il numero di frame completi ricevuti come parte dello PutMedia stream. Unità: numero
PutMedia.ActiveConnections	Il numero totale di connessioni al service host. Unità: numero
PutMedia.ConnectionErrors	Gli errori durante la creazione della PutMedia connessione per lo stream. Unità: numero

Metrica	Description
<code>PutMedia.FragmentIngestionLatency</code>	La differenza di tempo tra il momento in cui il primo e l'ultimo byte di un frammento vengono ricevuti da Amazon Kinesis Video Streams. Unità: millisecondi
<code>PutMedia.FragmentPersistLatency</code>	Il tempo impiegato dal momento in cui i dati completi del frammento vengono ricevuti e archiviati. Unità: numero
<code>PutMedia.Latency</code>	La differenza di tempo tra la richiesta e la risposta HTTP InletService durante la creazione della connessione. Unità: numero
<code>PutMedia.BufferingAckLatency</code>	La differenza di tempo tra il momento in cui il primo byte di un nuovo frammento viene ricevuto da Amazon Kinesis Video Streams e il momento in cui viene inviato il Buffering ACK per il frammento. Unità: millisecondi
<code>PutMedia.ReceivedAckLatency</code>	La differenza di tempo tra il momento in cui l'ultimo byte di un nuovo frammento viene ricevuto da Amazon Kinesis Video Streams e il momento in cui viene inviato l'ACK ricevuto per il frammento. Unità: millisecondi
<code>PutMedia.PersistedAckLatency</code>	La differenza di tempo tra il momento in cui l'ultimo byte di un nuovo frammento viene ricevuto da Amazon Kinesis Video Streams e il momento in cui viene inviato l'ACK persistente per il frammento. Unità: millisecondi

Metrica	Description
<code>PutMedia.ErrorAckCount</code>	Il numero di Error ACK inviati durante l'invio dello stream. <code>PutMedia</code> Unità: numero
<code>PutMedia.Success</code>	1 per ogni frammento scritto senza errori; 0 per ogni frammento con errori. Il valore medio di questo parametro indica il numero di frammenti completi e validi inviati. Unità: numero
<code>GetMedia.Requests</code>	Il numero di richieste <code>GetMedia</code> API per un determinato stream. Unità: numero
<code>GetMedia.OutgoingBytes</code>	Il numero totale di byte inviati dal servizio come parte dell'<code>GetMediaAPI</code> per un determinato flusso. Unità: byte
<code>GetMedia.OutgoingFragments</code>	Il numero di frammenti inviati durante lo <code>GetMedia</code> streaming. Unità: numero
<code>GetMedia.OutgoingFrames</code>	Il numero di frame inviati durante <code>GetMedia</code> un determinato stream. Unità: numero
<code>GetMedia.MillisBehindNow</code>	La differenza di tempo tra il timestamp corrente del server e il timestamp del server dell'ultimo frammento inviato. Unità: millisecondi

Metrica	Description
<code>GetMedia.ConnectionErrors</code>	Il numero di connessioni che non sono state stabilite con successo. Unità: numero
<code>GetMedia.Success</code>	1 per ogni frammento inviato senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle richieste, vedere Summary AWS Logging. Request/Response Unità: numero
<code>GetMediaForFragmentList.OutgoingBytes</code>	Il numero totale di byte inviati dal servizio come parte dell'<code>GetMediaForFragmentList</code> API per un determinato stream. Unità: byte
<code>GetMediaForFragmentList.OutgoingFragments</code>	Il numero totale di frammenti inviati dal servizio come parte dell'<code>GetMediaForFragmentList</code> API per un determinato flusso. Unità: numero
<code>GetMediaForFragmentList.OutgoingFrames</code>	Il numero totale di frame inviati dal servizio come parte dell'<code>GetMediaForFragmentList</code> API per un determinato flusso. Unità: numero

Metrica	Description
<code>GetMediaForFragmentList.Requests</code>	Il numero di richieste <code>GetMediaForFragmentList</code> API per un determinato stream. Unità: numero
<code>GetMediaForFragmentList.Success</code>	1 per ogni frammento inviato senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>ListFragments.Latency</code>	La latenza delle chiamate <code>ListFragments</code> API per il nome dello stream specificato. Unità: millisecondi
<code>ListFragments.Requests</code>	Il numero di richieste <code>ListFragments</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>ListFragments.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetHLSStreamingSessionURL.Latency</code>	La latenza delle chiamate <code>GetHLSStreamingSessionURL</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetHLSStreamingSessionURL.Requests</code>	Il numero di richieste <code>GetHLSStreamingSessionURL</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>GetHLSStreamingSessionURL.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetHLSMasterPlaylist.Latency</code>	La latenza delle chiamate <code>GetHLSMasterPlaylist</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetHLSMasterPlaylist.Requests</code>	Il numero di richieste <code>GetHLSMasterPlaylist</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>GetHLSMasterPlaylist.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetHLSMediaPlaylist.Latency</code>	La latenza delle chiamate <code>GetHLSMediaPlaylist</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetHLSMediaPlaylist.Requests</code>	Il numero di richieste <code>GetHLSMediaPlaylist</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>GetHLSMediaPlaylist.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetMP4InitFragment.Latency</code>	La latenza delle chiamate <code>GetMP4InitFragment</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetMP4InitFragment.Requests</code>	Il numero di richieste <code>GetMP4InitFragment</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>GetMP4InitFragment.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetMP4MediaFragment.Latency</code>	La latenza delle chiamate <code>GetMP4MediaFragment</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetMP4MediaFragment.Requests</code>	Il numero di richieste <code>GetMP4MediaFragment</code> API per un determinato stream. Unità: numero

Metrica	Description
<code>GetMP4MediaFragment.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetMP4MediaFragment.OutgoingBytes</code>	Il numero totale di byte inviati dal servizio come parte dell'<code>GetMP4MediaFragment</code> API per un determinato stream. Unità: byte
<code>GetTSFragment.Latency</code>	La latenza delle chiamate <code>GetTSFragment</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetTSFragment.Requests</code>	Il numero di richieste <code>GetTSFragment</code> API per un determinato stream. Unità: numero

Metrica	Description
GetTSFragment.Success	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
GetTSFragment.OutgoingBytes	Il numero totale di byte inviati dal servizio come parte dell'GetTSFragment API per un determinato stream. Unità: byte
GetDASHStreamingSessionURL.Latency	La latenza delle chiamate GetDASHStreamingSessionURL API per il nome dello stream specificato. Unità: millisecondi
GetDASHStreamingSessionURL.Requests	Il numero di richieste GetDASHStreamingSessionURL API per un determinato stream. Unità: numero

Metrica	Description
<code>GetDASHStreamingSessionURL.Success</code>	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
<code>GetDASHManifest.Latency</code>	La latenza delle chiamate <code>GetDASHManifest</code> API per il nome dello stream specificato. Unità: millisecondi
<code>GetDASHManifest.Requests</code>	Il numero di richieste <code>GetDASHManifest</code> API per un determinato stream. Unità: numero

Metrica	Description
GetDASHManifest.Success	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Request/Response Summary Logging. Unità: numero
GetClip.Latency	La latenza delle chiamate GetClip API per il nome del flusso video specificato. Unità: millisecondi
GetClip.Requests	Il numero di richieste GetClip API per un determinato flusso video. Unità: numero

Metrica	Description
GetClip.Success	1 per ogni richiesta senza errori; 0 per ogni errore. Il valore medio indica la percentuale di successo.  Note Gli errori includono errori 400 (utente) e 500 (sistema). Per ulteriori informazioni sull'attivazione di un riepilogo delle richieste e delle risposte, inclusi gli ID delle AWS richieste, consulta Registrazione Request/Response riassuntiva. Unità: numero
GetClip.OutgoingBytes	Il numero totale di byte inviati dal servizio come parte dell' GetClip API per un determinato flusso video. Unità: byte

CloudWatch indicazioni sulle metriche

CloudWatch le metriche possono aiutare a trovare le risposte alle seguenti domande:

Argomenti

- [I dati raggiungono il servizio Amazon Kinesis Video Streams?](#)
- [Perché i dati non vengono inseriti correttamente dal servizio Amazon Kinesis Video Streams?](#)
- [Perché i dati non possono essere letti dal servizio Amazon Kinesis Video Streams alla stessa velocità con cui vengono inviati dal produttore?](#)
- [Perché non c'è video nella console oppure perché il video viene riprodotto in ritardo?](#)
- [Con quanto ritardo vengono letti i dati in tempo reale e perché il client è in ritardo rispetto all'inizio del flusso?](#)
- [Il client sta leggendo i dati dallo streaming video Kinesis e a quale velocità?](#)
- [Perché il cliente non riesce a leggere i dati dal flusso video Kinesis?](#)

I dati raggiungono il servizio Amazon Kinesis Video Streams?

Parametri pertinenti:

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

Cosa fare:

- Se si verifica un calo di queste metriche, controlla se la tua applicazione sta ancora inviando dati al servizio.
- Verificare la larghezza di banda della rete. Se è insufficiente, potrebbe rallentare la velocità con cui il servizio riceve i dati.

Perché i dati non vengono inseriti correttamente dal servizio Amazon Kinesis Video Streams?

Parametri pertinenti:

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

Cosa fare:

- In caso di aumento `PutMedia.ConnectionErrors`, guarda la risposta HTTP e i codici di errore ricevuti dal client produttore per vedere quali errori si verificano durante la creazione della connessione.
- Se si verifica un calo `PutMedia.Success` o un aumento `PutMedia.ErrorAckCount`, guarda il codice di errore ack nelle risposte ack inviate dal servizio per capire perché l'inserimento dei dati non riesce. Per ulteriori informazioni, consulta [AckErrorCode.Values](#).

Perché i dati non possono essere letti dal servizio Amazon Kinesis Video Streams alla stessa velocità con cui vengono inviati dal produttore?

Parametri pertinenti:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Cosa fare:

- Se si verifica un calo di queste metriche, controlla la larghezza di banda di rete delle tue connessioni. Low-bandwidth le connessioni potrebbero far sì che i dati raggiungano il servizio a una velocità inferiore.

Perché non c'è video nella console oppure perché il video viene riprodotto in ritardo?

Parametri pertinenti:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Cosa fare:

- Se si verifica un aumento `PutMedia.FragmentIngestionLatency` o un calo `PutMedia.IncomingFragments`, controlla la larghezza di banda della rete e se i dati vengono ancora inviati.
- Se si verifica un calo `PutMedia.Success`, controlla i codici di errore del pacchetto. Per ulteriori informazioni, consulta [AckErrorCode.Values](#).
- Se c'è un aumento di `PutMedia.FragmentPersistLatency` o `ListFragments.Latency`, molto probabilmente stai riscontrando un problema di servizio. Se la condizione persiste per un periodo di tempo prolungato, verifica con il tuo contatto del servizio clienti se c'è un problema con il tuo servizio.

Con quanto ritardo vengono letti i dati in tempo reale e perché il client è in ritardo rispetto all'inizio del flusso?

Parametri pertinenti:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Cosa fare:

- Se si verifica un aumento `GetMedia.ConnectionErrors`, il consumatore potrebbe rimanere indietro nella lettura dello streaming, a causa dei frequenti tentativi di riconnettersi allo streaming. Guarda i response/error codici HTTP restituiti per la `GetMedia` richiesta.
- Se si verifica un calo `GetMedia.Success`, è probabile che il servizio non sia in grado di inviare i dati al consumatore, con conseguente interruzione della connessione e riconnessioni da parte del consumatore, con conseguente ritardo dell'utente rispetto allo stream.
- In caso di aumento `GetMedia.MillisBehindNow`, controlla i limiti di larghezza di banda per vedere se stai ricevendo i dati a una velocità inferiore a causa della larghezza di banda inferiore.

Il client sta leggendo i dati dallo streaming video Kinesis e a quale velocità?

Parametri pertinenti:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

Cosa fare:

- Queste metriche indicano la velocità con cui vengono letti i dati archiviati e in tempo reale.

Perché il cliente non riesce a leggere i dati dal flusso video Kinesis?

Parametri pertinenti:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Cosa fare:

- Se il numero aumenta `GetMedia.ConnectionErrors`, controllate la risposta HTTP e i codici di errore restituiti dalla `GetMedia` richiesta. Per ulteriori informazioni, consulta [AckErrorCode.Values](#).
- Se stai cercando di leggere i dati più recenti o in tempo reale, controlla `PutMedia.IncomingBytes` se ci sono dati in arrivo nello stream che il servizio invia ai consumatori.
- Se si verifica un calo `GetMedia.Success` o `GetMediaForFragmentList.Success`, è probabile che il servizio non sia in grado di inviare i dati al consumatore. Se la condizione persiste per un periodo di tempo prolungato, contatta il tuo contatto del servizio clienti per vedere se c'è un problema con il tuo servizio.

Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch

Puoi monitorare Amazon Kinesis Video Streams Edge Agent utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi in metriche leggibili e quasi in tempo reale. Queste statistiche vengono registrate per un periodo di 15 mesi. Con queste informazioni cronologiche, puoi avere una prospettiva migliore sulle prestazioni della tua applicazione web o del servizio Amazon Kinesis Video Streams Edge Agent.

Per visualizzare le metriche, procedi come segue:

1. Accedi a Console di gestione AWS e apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nella barra di navigazione a sinistra, in **Metriche**, seleziona **Tutte le metriche**.
3. Scegli la scheda **Sfoglia**, quindi seleziona il namespace **EdgeRuntimeAgent** personalizzato.

Amazon Kinesis Video Streams Edge Agent pubblica le seguenti metriche sotto il namespace: `EdgeRuntimeAgent`

Dimensioni	Stato	Description
Nome dello stream, <code>RecordJob</code>	In esecuzione	Pubblica continuamente quando <code>RecordJob</code> è in esecuzione. Unità: nessuna. «1" viene pubblicato per tutto il tempo in cui <code>RecordJob</code> rimane in questo stato.
	FatalError	Pubblica in caso di errore <code>RecordJob</code> irreversibile. Unità: nessuna. «1" viene pubblicato una volta, quando si verifica questo evento.  Note Per ulteriori informazioni, consulta i log.
	Completato	Pubblica quando a <code>RecordJob</code> è completato. Unità: nessuna. «1" viene pubblicato una volta, quando si verifica questo evento.
Nome dello stream, <code>UploadJob</code>	In esecuzione	Pubblica continuamente quando <code>UploadJob</code> è in esecuzione. Unità: nessuna. «1" viene pubblicato per tutto il tempo in cui <code>UploadJob</code> rimane in questo stato.
	FatalError	Pubblica in caso di errori <code>UploadJob</code> irreversibili. Unità: nessuna. «1" viene pubblicato una volta, quando si verifica questo evento.  Note Per ulteriori informazioni, consulta i log.

Dimensioni	Stato	Description
	Completato	Pubblica quando <code>UploadJob</code> è completato. Unità: nessuna. «1" viene pubblicato una volta, quando si verifica questo evento.
Nome dello stream	PercentageSpaceUsed	Questa è la percentuale utilizzata sullo spazio totale allocato nelle configurazioni di Amazon Kinesis Video Streams Edge Agent per la registrazione di contenuti multimediali. Per ulteriori informazioni, consulta LocalSizeConfig. Unità: percentuale (scala 0—1).
Nome oggetto	Vivo	Pubblica ogni minuto da Amazon Kinesis Video Streams Edge Agent, indipendentemente dalle configurazioni in esecuzione su di esso. Questo può essere utilizzato per capire se Amazon Kinesis Video Streams Edge Agent è attivo e pronto ad accettare le configurazioni. Unità: nessuna. «1" viene pubblicato ogni minuto.
	RecordJobsHealthyJobCount	Numero totale di processi di registrazione in esecuzione e pianificati su Amazon Kinesis Video Streams Edge Agent. Unità: conteggio.
	UploadJobsHealthyJobCount	Numero totale di processi di caricamento in esecuzione e pianificati su Amazon Kinesis Video Streams Edge Agent. Unità: conteggio.
	RecordJobsUnhealthyJobCount	Numero totale dei processi di registrazione attualmente errati. Unità: conteggio.

Dimensioni	Stato	Description
	UploadJobs.UnhealthyJobCount	Numero totale dei lavori di caricamento attualmente errati. Unità: conteggio.
	RecordJobs.RunningJobCount	Numero totale di processi di registrazione in esecuzione attiva. Unità: conteggio.
	UploadJobs.RunningJobCount	Numero totale di processi di caricamento in esecuzione attiva. Unità: conteggio.
	RecordJobs.EdgeConfigCount	Numero totale di configurazioni di record in corso su Amazon Kinesis Video Streams Edge Agent. Unità: conteggio.
	UploadJobs.EdgeConfigCount	Numero totale di configurazioni di caricamento in corso su Amazon Kinesis Video Streams Edge Agent. Unità: conteggio.

CloudWatch indicazioni sulle metriche per Amazon Kinesis Video Streams Edge Agent

CloudWatch le metriche possono essere utili per trovare risposte alle seguenti domande:

Argomenti

- [Amazon Kinesis Video Streams Edge Agent dispone di spazio sufficiente per la registrazione?](#)
- [Amazon Kinesis Video Streams Edge Agent è attivo?](#)
- [Ci sono lavori non salutar?](#)
- [Qualche lavoro richiede un intervento esterno?](#)

Amazon Kinesis Video Streams Edge Agent dispone di spazio sufficiente per la registrazione?

Metriche pertinenti: PercentageSpaceUsed

Azione: nessuna azione richiesta.

Amazon Kinesis Video Streams Edge Agent è attivo?

Metriche pertinenti: `Alive`

Azione: se in qualsiasi momento smetti di ricevere questa metrica, significa che Amazon Kinesis Video Streams Edge Agent ha rilevato una o più delle seguenti situazioni:

- Un problema di runtime dell'applicazione: vincolo di memoria o di altre risorse, bug e così via
- Il AWS IoT dispositivo su cui è in esecuzione l'agente al momento dell'arresto, si è bloccato o è terminato
- Il AWS IoT dispositivo non dispone di connettività di rete

Ci sono lavori non salutari?

Parametri pertinenti:

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

Azione: ispeziona i log e cerca la `FatalError` metrica.

- Se la `FatalError` metrica è presente, si è verificato un errore irreversibile ed è necessario riavviare manualmente il processo. Ispeziona i log e risolvi il problema prima di utilizzare `StartEdgeConfigurationUpdate` per riavviare manualmente il processo.
- Se la `FatalError` metrica non è presente, si è verificato un errore transitorio (non irreversibile) e Amazon Kinesis Video Streams Edge Agent sta riprovando il processo.

 Note

Per fare in modo che l'agente ritenti un job con un errore irreversibile, usa.

[StartEdgeConfigurationUpdate](#)

Qualche lavoro richiede un intervento esterno?

Parametri pertinenti:

- `PercentageSpaceUsed`— Se questo valore supera un determinato valore, il processo di registrazione viene messo in pausa e ripreso solo quando è disponibile spazio (quando i file multimediali non sono più conservati). È possibile inviare una configurazione aggiornata con un valore superiore `MaxLocalMediaSizeInMB` per aggiornare immediatamente il job.
- `RecordJob.FatalError/UploadJob.FatalError`— Esamina i log dell'agente e invia nuovamente la configurazione per riprendere il lavoro.

Azione: Effettua una chiamata API con la configurazione per riavviare i lavori che riscontrano questo problema.

Registra le chiamate API Amazon Kinesis Video Streams con AWS CloudTrail

Amazon Kinesis Video Streams funziona con Amazon Kinesis Video Streams AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, da un ruolo o da un utente Servizio AWS in Amazon Kinesis Video Streams. CloudTrail acquisisce tutte le chiamate API per Amazon Kinesis Video Streams come eventi. Le chiamate acquisite includono chiamate dalla console Amazon Kinesis Video Streams e chiamate in codice alle operazioni dell'API Amazon Kinesis Video Streams. Se crei un percorso, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon Kinesis Video Streams. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon Kinesis Video Streams, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per [l'AWS CloudTrail utente](#).

Amazon Kinesis Video Streams e CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in Amazon Kinesis Video Streams, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Kinesis Video Streams, crea un percorso. Un percorso consente di CloudTrail inviare file di registro

a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il percorso registra gli eventi provenienti da tutte le regioni della AWS partizione e invia i file di registro al bucket Amazon S3 specificato. Inoltre, puoi configurarne altri Servizi AWS per analizzare ulteriormente e agire in base ai dati degli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un trail](#)
- [CloudTrail Servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Amazon Kinesis Video Streams supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Esempio: voci nei file di registro di Amazon Kinesis Video Streams

Un trail è una configurazione che consente l'invio di eventi come file di registro a un bucket Amazon S3 specificato. CloudTrail i file di log contengono una o più voci di registro. Un evento rappresenta una singola richiesta da qualsiasi sorgente e include informazioni sull'azione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. I file di log di CloudTrail non sono una traccia di stack ordinata delle chiamate API pubbliche, pertanto non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'[CreateStream](#)azione.

```
{
  "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": "b7acfcfd0-6ca9-4ee1-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"
  }
]
```

Limiti dei metadati di streaming

[the section called “Quote dei servizi di metadati di streaming”](#) Per ulteriori informazioni sui limiti che si applicano all'aggiunta di metadati di streaming a un flusso video Kinesis, consulta.

Pianifica la registrazione e l'archiviazione video con Amazon Kinesis Video Streams Edge Agent

Amazon Kinesis Video Streams offre un modo efficiente e conveniente per connettersi alle telecamere IP presso la sede del cliente. Con Amazon Kinesis Video Streams Edge Agent, puoi registrare e archiviare localmente i video dalle telecamere e trasmetterli in streaming sul cloud secondo una pianificazione definita dal cliente per l'archiviazione, la riproduzione e l'elaborazione analitica a lungo termine.

Note

Per accedere ad Amazon Kinesis Video Streams Edge Agent, completa questo breve modulo. <https://pages.awscloud.com/GLOBAL-launch-DL-KVS-Edge-2023-learn.html>

Puoi scaricare Amazon Kinesis Video Streams Edge Agent e distribuirlo sui tuoi dispositivi di edge computing locali. Puoi anche distribuirli facilmente in contenitori Docker in esecuzione su istanze Amazon EC2. Dopo la distribuzione, puoi utilizzare l'API Amazon Kinesis Video Streams per aggiornare le configurazioni di registrazione video e caricamento su cloud. La funzionalità funziona con qualsiasi telecamera IP in grado di eseguire lo streaming tramite protocollo RTSP. Non richiede alcuna installazione aggiuntiva del firmware per le telecamere.

Offriamo le seguenti installazioni per Amazon Kinesis Video Streams Edge Agent:

- Come AWS IoT Greengrass V2 componente: puoi installare Amazon Kinesis Video Streams Edge Agent come AWS IoT Greengrass componente su qualsiasi dispositivo certificato. AWS IoT Greengrass Per ulteriori informazioni AWS IoT Greengrass, consulta la Guida per gli [AWS IoT Greengrass Version 2 sviluppatori](#).
- Attivo AWS Snowball Edge: puoi eseguire Amazon Kinesis Video Streams Edge Agent su dispositivi Snowball Edge. Per ulteriori informazioni, consulta la [AWS Snowball Edge Edge Developer Guide](#).
- In una AWS IoT distribuzione nativa: puoi installare Amazon Kinesis Video Streams Edge Agent in modo nativo su qualsiasi istanza di calcolo. L'SDK Edge viene utilizzato [AWS IoT Core](#) per gestire Edge tramite le operazioni API di [Amazon Kinesis Video Streams](#).

Per iniziare a usare Amazon Kinesis Video Streams Edge Agent, continua con le procedure appropriate riportate di seguito.

Argomenti

- [Operazioni dell'API Amazon Kinesis Video Streams Edge Agent](#)
- [Monitoraggio di Amazon Kinesis Video Streams Edge Agent](#)
- [Implementa in modalità non-AWS IoT Greengrass mode](#)
- [Distribuisci Amazon Kinesis Video Streams Edge Agent su AWS IoT Greengrass](#)
- [Domande frequenti su Amazon Kinesis Video Streams Edge Agent](#)

Operazioni dell'API Amazon Kinesis Video Streams Edge Agent

Utilizza le seguenti operazioni API per configurare Amazon Kinesis Video Streams Edge Agent:

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

Monitoraggio di Amazon Kinesis Video Streams Edge Agent

Per monitorare il tuo Amazon Kinesis Video Streams Edge Agent, consulta. [the section called “Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch”](#)

Implementa in modalità non-AWS IoT Greengrass mode

Questa sezione fornisce una guida completa per utilizzare Amazon Kinesis Video Streams al di fuori dell'ambiente AWS IoT Greengrass. Che tu stia lavorando con dispositivi edge o altre piattaforme, queste informazioni ti aiuteranno a configurare e utilizzare Kinesis Video Streams in modo efficace.

Troverai informazioni dettagliate su:

- Configurazione dell'ambiente di sviluppo
- Creazione di uno streaming video Kinesis
- Scaricamento e compilazione del Kinesis Video Streams Producer SDK

- Scrittura ed esame di un'applicazione di esempio
- Esecuzione dell'applicazione di esempio

Continua con i passaggi seguenti per eseguire Amazon Kinesis Video Streams Edge Agent con AWS IoT MQTT come distribuzione autonoma.

Argomenti

- [Installare le dipendenze](#)
- [Crea risorse per gli URL RTSP della tua telecamera IP](#)
- [Crea una politica di autorizzazioni IAM](#)
- [Creazione di un ruolo IAM](#)
- [Crea il AWS IoT alias di ruolo](#)
- [Crea il AWS IoT policy](#)
- [Crea un AWS IoT cosa e ottieni AWS IoT Core credenziali](#)
- [Crea Amazon Kinesis Video Streams Edge Agent](#)
- [Installa l' CloudWatch agente sul dispositivo](#)
- [Esegui Amazon Kinesis Video Streams Edge Agent come processo nativo](#)

Installare le dipendenze

Prima di poter iniziare a utilizzare l'SDK per produttori di Amazon Kinesis Video Streams, devi configurare il tuo ambiente di sviluppo con le dipendenze necessarie. Questa pagina ti guida attraverso il processo di installazione dei componenti e delle librerie software richiesti sul tuo sistema.

Note

Per un elenco dei sistemi operativi supportati, vedere [the section called “Quali sistemi operativi sono supportati da Amazon Kinesis Video Streams Edge Agent?”](#).

Installa le dipendenze sul dispositivo

1. Per eseguire Amazon Kinesis Video Streams Edge Agent, installa le seguenti librerie appropriate sul tuo dispositivo:

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
```

Installa log4cplus-2.1.0 dal codice sorgente.

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

apache-maven-3.9.2 Effettua l'installazione dal codice sorgente.

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

⚠ Important

Se viene visualizzata una schermata che indica che è necessario riavviare alcuni servizi, premi Invio per selezionare Ok.

Per ulteriori informazioni, consulta la Guida per l'utente di [Amazon Corretto 11](#).

2. Installa il AWS Command Line Interface. Consulta la sezione [Installazione o aggiornamento della versione più recente delle AWS CLI](#) procedure nella Guida per l'AWS Command Line Interface utente.

Crea risorse per gli URL RTSP della tua telecamera IP

Segui queste procedure per creare gli stream e i segreti necessari in AWS Secrets Manager. Fai prima questo passaggio, perché hai bisogno degli ARN delle risorse create nelle policy.

Crea Amazon Kinesis Video Streams

Crea Amazon Kinesis Video Streams utilizzando l'API Console di gestione AWS,, AWS CLI o.

In Console di gestione AWS, apri la console [Amazon Kinesis Video Streams](#). Scegli Streams video nella barra di navigazione a sinistra.

Per ulteriori informazioni, consulta [the section called “Crea uno streaming video Amazon Kinesis”](#).

Crea segreti in AWS Secrets Manager

In Console di gestione AWS, apri la [AWS Secrets Manager console](#). Scegli Secrets nella barra di navigazione a sinistra.

Verifica che sia selezionata la regione appropriata.

1. Scegli Archivia un nuovo segreto.
 - a. Passaggio 1: scegli il tipo di segreto
 - Selezionare Other type of secret (Altro tipo di segreti).
 - Nella sezione Key/Value Coppie, aggiungi una coppia chiave-valore.

Chiave: MediaURI

 Note

La chiave deve essere. MediaURI Fa distinzione tra maiuscole e minuscole. Se lo inserisci in modo errato, l'applicazione non funziona.

Valore: *Your MediaURI*.

Example

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

- b. Passo 2: Configura il segreto. Dai un nome a questo segreto. Dai un nome come preferisci.
 - c. Fase 3: Configurare la rotazione - opzionale. Scegli Next (Successivo).
 - d. Fase 4: Revisione. Scegli Store.
2. Se il tuo segreto non viene visualizzato immediatamente, seleziona il pulsante di aggiornamento.

Scegli il nome del tuo segreto. Prendi nota dell'ARN segreto.

3. Ripeti questa procedura per ogni MediaURI da cui desideri eseguire lo streaming.

 Note

La AWS rete blocca alcune fonti RTSP pubbliche. Non è possibile accedervi dall'interno dell'istanza Amazon EC2 o se la connessione alla VPN è in modalità non gestita.

 Important

L'URL RTSP della videocamera deve trasmettere video in formato h.264. La durata del frammento non deve superare il limite indicato in. [the section called “Quote SDK di Producer”](#)

Amazon Kinesis Video Streams Edge Agent supporta solo video.

Corri `gst-discoverer-1.0 Your RtspUrl` per assicurarti che la videocamera sia raggiungibile dal tuo dispositivo.

Salva gli ARN per tutti gli stream e i segreti che hai creato. Ti servono per il passaggio successivo.

Crea una politica di autorizzazioni IAM

Segui queste procedure per creare una policy IAM. Questa politica di autorizzazioni consente il controllo selettivo degli accessi (un sottoinsieme delle operazioni supportate) per una risorsa. AWS In questo caso, AWS le risorse sono i flussi video su cui desideri che Amazon Kinesis Video Streams Edge Agent esegua lo streaming. Le risorse includono anche i AWS Secrets Manager segreti che Amazon Kinesis Video Streams Edge Agent può recuperare. Per ulteriori informazioni, consulta [le politiche IAM](#).

Crea una policy utilizzando l'editor di policy JSON

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione sinistro, scegli Policy.

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

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

Sostituisci `arn:aws:kinesisvideo:*:*:stream/streamName1/*` e `arn:aws:kinesisvideo:*:*:stream/streamName2/*` con gli ARN per i flussi video e sostituiscili `arn:aws:secretsmanager:*:*:secret:*` con gli ARN che contengono i segreti MediaURI che hai creato in [the section called “Crea risorse per gli URL RTSP della tua telecamera IP”](#) Usa gli ARN per i segreti a cui vuoi che Amazon Kinesis Video Streams Edge Agent acceda.

6. Scegli Next (Successivo).

Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta la sezione Ristrutturazione [delle politiche](#) nella IAM User Guide.

7. Nella pagina Rivedi e crea, inserisci il nome della policy e una descrizione opzionale per la policy che stai creando. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Creazione di un ruolo IAM

Il ruolo creato in questo passaggio può essere assunto per ottenere credenziali temporanee da (). AWS IoT Servizio di token di sicurezza AWS AWS STS Questa operazione viene eseguita quando si eseguono richieste di autorizzazione delle credenziali da Amazon Kinesis Video Streams Edge Agent.

Crea il ruolo di servizio per Amazon Kinesis Video Streams (console IAM)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Scegli il tipo di ruolo Custom trust policy e incolla la seguente policy:

JSON

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

```
}
```

4. Seleziona la casella accanto alla policy IAM in cui hai creato [the section called “Crea una politica di autorizzazioni IAM”](#).
5. Scegli Next (Successivo).
6. Inserisci il nome o il suffisso del nome del ruolo per aiutarti a identificare lo scopo di questo ruolo.

Example

Esempio: KvsEdgeAgentRole

7. (Facoltativo) In Descrizione, inserisci una descrizione per il nuovo ruolo.
8. (Facoltativo) Aggiungi metadati al ruolo allegando i tag come coppie. key/value

Per ulteriori informazioni sull'uso dei tag in IAM, consulta la sezione [Tagging IAM resources](#) nella IAM User Guide.

9. Rivedere il ruolo e scegliere Crea ruolo.

Crea il AWS IoT alias di ruolo

Segui queste procedure per creare un alias di AWS IoT ruolo per il ruolo IAM in cui hai creato. [the section called “Creazione di un ruolo IAM”](#) Un alias di ruolo è un modello di dati alternativo che rimanda al ruolo IAM. Una richiesta di fornitore di AWS IoT credenziali deve includere un alias di ruolo per indicare quale ruolo IAM assumere per ottenere credenziali temporanee da (). Servizio di token di sicurezza AWS AWS STS Per ulteriori informazioni, consulta [Come utilizzare un certificato per ottenere un token di sicurezza](#).

Crea il AWS IoT alias di ruolo

1. Accedi a Console di gestione AWS e apri la AWS IoT Core console all'indirizzo <https://console.aws.amazon.com/iot/>.
2. Verifica che sia selezionata la regione appropriata.
3. Nella barra di navigazione a sinistra, seleziona Sicurezza, quindi scegli Alias di ruolo.
4. Scegli Crea alias di ruolo.
5. Inserisci un nome per l'alias del tuo ruolo.

Example

Esempio: KvsEdgeAgentRoleAlias

6. Nel menu a discesa Ruolo, seleziona il ruolo IAM in cui hai creato. [the section called “Creazione di un ruolo IAM”](#)
7. Scegli Create (Crea). Nella pagina successiva, vedrai una nota che indica che l'alias del tuo ruolo è stato creato con successo.
8. Cerca e seleziona l'alias di ruolo appena creato. Prendi nota dell'alias di ruolo ARN. Ne avrai bisogno per la AWS IoT policy nella fase successiva.

Crea il AWS IoT policy

Segui queste procedure per creare una AWS IoT policy da allegare al certificato del dispositivo. Ciò conferisce le autorizzazioni alle AWS IoT funzionalità e consente l'assunzione dell'alias di ruolo utilizzando il certificato.

Con AWS IoT Core le policy, puoi controllare l'accesso al piano dati. AWS IoT Core Il piano AWS IoT Core dati è costituito da operazioni che è possibile utilizzare per effettuare le seguenti operazioni:

- Connettiti al broker di AWS IoT Core messaggi
- Inviare e ricevere messaggi MQTT
- Ottieni o aggiorna l'ombra del dispositivo di un oggetto

Per ulteriori informazioni, consulta [policy AWS IoT Core](#).

Utilizzo AWS IoT editor di politiche per creare un AWS IoT policy

1. Accedi Console di gestione AWS e apri la AWS IoT Core console all'indirizzo <https://console.aws.amazon.com/iot/>.
2. Nella barra di navigazione a sinistra, seleziona Sicurezza, quindi scegli Politiche.
3. Scegli Crea policy.
4. Immetti un nome da assegnare alla policy.

Example

Un esempio di nome di policy è KvsEdgeAccessIoTPolicy.

5. (Facoltativo) Aggiungere metadati alla policy collegando i tag come coppie chiave-valore.

Per ulteriori informazioni sull'uso dei tag in IAM, consulta [Tagging your AWS IoT resources](#) nella AWS IoT Core Developer Guide.

6. Scegli la scheda JSON.
7. Incolla il seguente documento sulla policy JSON:

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

Sostituiscilo `your-role-alias-arn` con l'ARN dell'alias di ruolo in cui hai creato. [the section called “Crea l'alias del AWS IoT ruolo”](#)

8. Scegli Crea per salvare il tuo lavoro.

Crea un AWS IoT cosa e ottieni AWS IoT Core credenziali

A questo punto hai creato:

- Una politica di autorizzazioni IAM. Per informazioni, consulta [the section called “Crea una politica di autorizzazioni IAM”](#).
- Un ruolo IAM, con la politica delle autorizzazioni allegata. Per informazioni, consulta [the section called “Creazione di un ruolo IAM”](#).
- Un alias di AWS IoT ruolo per il ruolo IAM. Per informazioni, consulta [the section called “Crea l'alias del AWS IoT ruolo”](#).
- Una AWS IoT policy, attualmente non associata a nessuna AWS risorsa. Per informazioni, consulta [the section called “Crea la AWS IoT politica”](#).

Per creare e registrare un AWS IoT cosa e ottieni AWS IoT Core credenziali di accesso

1. Registra il dispositivo come AWS IoT oggetto e genera il X.509 certificato per il dispositivo.
 - a. Accedi a Console di gestione AWS e apri la AWS IoT Core console all'indirizzo <https://console.aws.amazon.com/iot/>.
 - b. Seleziona la regione appropriata.
 - c. Nella barra di navigazione a sinistra, seleziona Tutti i dispositivi, quindi scegli Cose.
 - d. Scegli Crea oggetti.
 - e. Seleziona Crea una singola cosa, quindi scegli Avanti.

1. Fase 1: Specifica le proprietà degli oggetti

Digita un nome per l'oggetto, quindi scegli Avanti.

2. Fase 2. Configura il certificato del dispositivo

Seleziona Auto-generate un nuovo certificato (consigliato), quindi scegli Avanti.

3. Fase 3. Allega le politiche al certificato

Cerca la politica di autorizzazione che hai creato in [the section called “Crea la AWS IoT politica”](#).

Seleziona la casella di controllo accanto alla tua politica e scegli **Crea oggetto**.

f. Nella finestra che appare, scarica i seguenti file:

- Certificato del dispositivo. Questo è il X.509 certificato.
- File con chiave pubblica
- File con chiave privata
- Endpoint Amazon Trust Services (chiave RSA a 2048 bit: Amazon Root CA 1)

Annota la posizione di ciascuno di questi file per un passaggio successivo.

g. Seleziona **Fatto**. Nella pagina successiva, vedrai una nota che indica che il tuo oggetto è stato creato con successo.

h. Trasferisci i file scaricati sopra sul tuo AWS IoT oggetto, se non è già presente.

2. Ottieni l'endpoint del fornitore di credenziali per il tuo AWS account.

AWS CLI

Esegui il comando seguente:

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

Console di gestione AWS

In [AWS CloudShell](#), esegui il seguente comando:

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

Prendi nota di queste informazioni per un passaggio successivo.

3. Ottieni l'endpoint dei dati del dispositivo per il tuo AWS account.

AWS CLI

Esegui il comando seguente:

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

Console di gestione AWS

Esegui questa operazione:

1. Accedi Console di gestione AWS e apri la AWS IoT Core console all'indirizzo <https://console.aws.amazon.com/iot/>.
2. Nella barra di navigazione a sinistra, seleziona Impostazioni.
3. Individua l'endpoint dei dati del dispositivo.

Prendi nota di queste informazioni per un passaggio successivo.

4. (Facoltativo) Verifica che i certificati siano stati generati correttamente.

Esegui il comando seguente per verificare che i tuoi articoli siano stati generati correttamente.

```
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
```

Per ulteriori informazioni, vedi [Come usare un certificato per ottenere un token](#) di sicurezza.

Crea Amazon Kinesis Video Streams Edge Agent

Crea Amazon Kinesis Video Streams Edge Agent

1. Scarica il tar file utilizzando il link che ti è stato fornito.

Se hai completato il modulo di interesse per Amazon Kinesis Video Streams Edge Agent, controlla la tua email per trovare il link per il download. Se non hai completato il modulo, compilalo [qui](#).

2. Verifica il checksum.
3. Estrai i file binari e il jar nel tuo dispositivo.

Tipo:tar -xvf kvs-edge-agent.tar.gz.

Dopo l'estrazione, la struttura delle cartelle sarà la seguente:

```
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

Il nome della cartella di rilascio deve essere impostato in modo da riflettere l'ultimo numero binario della versione. Ad esempio, una versione 1.0.0 avrà il nome della cartella impostato come 1.0.0.

4. Crea il jar delle dipendenze.

Note

Il barattolo incluso con `kvs-edge-agent.tar.gz` non contiene le dipendenze. Usa i seguenti passaggi per creare quelle librerie.

Accedere alla `kvs-edge-agent` cartella che contiene `pom.xml`.

Tipo `mvn clean package`.

Questo genera un file `jar` contenente le dipendenze richieste da Amazon Kinesis Video Streams Edge Agent. `kvs-edge-agent/target/libs.jar`

5. Inserisci `libs.jar` nella cartella che contiene gli artefatti del componente.

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

6. Imposta le variabili di ambiente utilizzando i valori dei passaggi precedenti. La tabella seguente fornisce le descrizioni delle variabili.

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_REGION	Sì	La regione utilizzata. Esempio: <code>us-west-2</code>
AWS_IOT_CA_CERT	Sì	Percorso del file al certificato CA utilizzato per stabilire l'attendibilità con il servizio di backend tramite TLS. Esempio: <code>/file/path/to/AmazonRootCA1.pem</code>
AWS_IOT_CORE_CERT	Sì	Percorso del file del certificato. X.509 Esempio: <code>/file/path/to/certificateID-certificate.pem.crt</code>

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	Sì	L'endpoint del fornitore di AWS IoT Core credenziali per il tuo account. AWS Esempio: <i>credential-account-specific-prefix</i>.credentials.iot.<i>aws-region</i>.amazonaws.com
AWS_IOT_CORE_DATA_ENDPOINT	Sì	L'endpoint del piano AWS IoT Core dati per il tuo account. AWS Esempio: <i>data-account-specific-prefix</i>.iot.<i>aws-region</i>.amazonaws.com
AWS_IOT_CORE_PRIVATE_KEY	Sì	Percorso del file alla chiave privata utilizzata nella coppia di public/private chiavi. Per ulteriori informazioni, vedere Gestione delle chiavi in AWS IoT. Esempio: <i>/file/path/to/certificateID-private</i>.pem.key

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_IOT_CORE_ROLE_ALIAS	Sì	Il nome dell'alias del ruolo che rimanda al ruolo AWS IAM da utilizzare per la connessione a. AWS IoT Core Esempio: kvs-edge-role-alias
AWS_IOT_CORE_THING_NAME	Sì	Il nome dell' AWS IoT oggetto su cui viene eseguita l'applicazione. Esempio: my-edge-device-thing

Nome della variabile di ambiente	Richiesto	Descrizione
GST_PLUGIN_PATH	Sì	Percorso del file che punta alla cartella che contiene le librerie gstreamer e quelle dipendenti dalla IngestorPipelineJNI piattaforma. Consente a GStreamer di caricare questi plugin. Per ulteriori informazioni, consulta the section called “Scarica, crea e configura l'elemento GStreamer”. Esempio: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i>

Nome della variabile di ambiente	Richiesto	Descrizione
LD_LIBRARY_PATH	Sì	Percorso del file che punta alla directory contenente le librerie e quelle dipendenti dalla cproducer piattaforma. KinesisVideoProducer Esempio: <i>/download-location</i> /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/<i>EdgeAgentVersion</i> /lib/
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	No	Determina se Amazon Kinesis Video Streams Edge Agent pubblicherà le metriche sullo stato del lavoro su Amazon CloudWatch Valori accettati: TRUE/FALSE (senza distinzione tra maiuscole e minuscole). Il valore predefinito è FALSE se non fornito. Esempio: FALSE

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_KVS_EDGE_LOG_LEVEL	No	Il livello di registrazione degli output di Amazon Kinesis Video Streams Edge Agent. Valori accettati: • OFF • ALL • FATALE • ERRORE • WARN • INFO, impostazione predefinita, se non fornita • DEBUG • TRACE Esempio: INFO
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	No	Una volta che il file di registro raggiunge questa dimensione, si verificherà un rollover. • Minimo: 0 • Massimo: 10000 • Valore predefinito: 20, se non fornito • Unità: Megabyte (MB) Esempio: 5

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	No	Il percorso del file che punta alla directory in cui vengono generati i log di Amazon Kinesis Video Streams Edge Agent. Il valore predefinito è se non fornito. <code>./log</code> Esempio: <code>/file/path/</code>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	No	Il numero di registri trasferiti da conservare prima dell'eliminazione. • Minimo: 1 • Massimo: 100 • Valore predefinito: 10, se non fornito Esempio: 20
AWS_KVS_EDGE_RECORDING_DIRECTORY	No	Verrà scritto il percorso del file che punta alla directory in cui è stato registrato il file multimediale. Il valore predefinito è la directory corrente se non viene fornita. Esempio: <code>/file/path/</code>

Nome della variabile di ambiente	Richiesto	Descrizione
GST_DEBUG	No	Specifica il livello di log di GStreamer da emettere. Per maggiori informazioni, consulta la documentazione di GStreamer . Esempio: 0
GST_DEBUG_FILE	No	Specifica il file di output dei log di debug di GStreamer. Se non è impostato, i log di debug restituiscono un errore standard. Per maggiori informazioni, consulta la documentazione di GStreamer . Esempio: <i>/tmp/gstreamer-logging.log</i>

7. Cancella la cache di GStreamer. Type:

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

Per ulteriori informazioni, consulta la documentazione del registro di [GStreamer](#).

8. Prepara ed esegui il comando java. Amazon Kinesis Video Streams Edge Agent accetta i seguenti argomenti:

Nome della proprietà Java	Richiesto	Description
java.library.path	No	Percorso del file che punta alla cartella contenente gstkvssink le librerie IngstorPipelineJN I dipendenti. Se non viene

Nome della proprietà Java	Richiesto	Description
		fornito, Amazon Kinesis Video Streams Edge Agent li cercherà nella directory corrente.

 **Important**
Amazon Kinesis Video Streams Edge Agent non funzionerà correttamente se non riesce a individuare questi file.

Esempio: */file/path/*

Per impostarli, aggiungili `-Djava-property-name=value` al comando java usato per eseguire il jar.

Ad esempio:

```
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

Esegui il comando java precedente dalla stessa directory di */download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*.

9. Invia le configurazioni all'applicazione utilizzando. AWS CLI

- a. Crea un nuovo file, *example-edge-configuration.json*.

Incolla il codice seguente nel file. Questa è una configurazione di esempio che registra ogni giorno dalle 9:00:00 AM alle 16:59:59 PM (in base all'ora di sistema sul dispositivo). AWS IoT Inoltre, carica i file multimediali registrati ogni giorno dalle 19:00:00 alle 21:59:59.

Per ulteriori informazioni, consulta [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. Per inviare il file ad Amazon Kinesis Video Streams Edge Agent, digita quanto segue nel: AWS CLI

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

10. Ripeti il passaggio precedente per ogni stream per Amazon Kinesis Video Streams Edge Agent.

Installa l' CloudWatch agente sul dispositivo

Note

Fai attenzione alle quote. [CloudWatch](#)

Segui queste procedure per installare e configurare l' CloudWatch agente su cui caricare automaticamente i log generati da Amazon Kinesis Video Streams Edge Agent. CloudWatch Questa fase è opzionale.

[Per](#) le procedure di installazione dell' CloudWatch agente sul tuo dispositivo, consulta la Amazon CloudWatch User Guide.

Quando viene richiesta la configurazione, seleziona una delle seguenti configurazioni.

Important

`file_path`Nelle seguenti configurazioni si presuppone che venga utilizzata la posizione di output di registrazione predefinita.

Il percorso del file utilizzato presuppone che si stia eseguendo Amazon Kinesis Video Streams Edge Agent dalla posizione: `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version`

- Per configurare l' CloudWatch agente per caricare i log e pubblicare le metriche della RAM e della CPU del dispositivo, incolla quanto segue nel file di configurazione.

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

```

- Per caricare solo i log e non raccogliere RAM e CPU del dispositivo, utilizza la seguente configurazione:

```

{
  "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"
    }
  ]
}
}
}
}
}

```

Esegui Amazon Kinesis Video Streams Edge Agent come processo nativo

Configura Amazon Kinesis Video Streams Edge Agent come servizio systemd. Questa fase è opzionale.

systemd è un gestore di sistemi e servizi su dispositivi Linux. systemd è il metodo consigliato per gestire il processo, in quanto riavvierà Amazon Kinesis Video Streams Edge Agent nel caso in cui l'applicazione riscontri un errore o il dispositivo su cui è in esecuzione l'applicazione perda l'alimentazione.

Esegui questa operazione:

Esegui Amazon Kinesis Video Streams Edge Agent come processo nativo

1. Crea un nuovo file `/etc/systemd/system` e assegnagli un nome.
`aws.kinesisvideo.edge-runtime-agent.service`

Incolla quanto segue:

```

[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
```

Per ulteriori informazioni sui parametri accettati dal file `systemd` di configurazione del servizio, consultate la [documentazione](#).

Note

Aggiungete le variabili di ambiente obbligatorie nella . . . posizione, come specificato in [the section called “Crea l'Edge Agent”](#).

2. Ricarica i file di servizio per includere il nuovo servizio.

Tipo `sudo systemctl daemon-reload`.

3. Avviare il servizio.

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

4. Controlla lo stato del servizio Amazon Kinesis Video Streams Edge Agent per verificare che sia in esecuzione.

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

Di seguito è riportato un esempio dell'output che vedrai.

```
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. Ispeziona i log per eventuali errori.

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

6. Digita `systemctl --help` per visualizzare l'elenco completo delle opzioni con cui gestire il processo. `systemctl`

Di seguito sono riportati alcuni comandi comuni per gestire Amazon Kinesis Video Streams Edge Agent:

- Per riavviare, digita. `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`
- Per interrompere, digita `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Per avviarsi automaticamente a ogni riavvio del dispositivo, digita `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Distribuisci Amazon Kinesis Video Streams Edge Agent su AWS IoT Greengrass

Questa sezione fornisce una guida completa all'uso di Amazon Kinesis Video Streams con AWS IoT Greengrass. Combinando questi servizi, puoi trasmettere in modo efficiente i video dai dispositivi periferici al cloud, abilitando un'ampia gamma di applicazioni nei settori dell'IoT, della sorveglianza e altro ancora.

Troverai informazioni dettagliate su:

- Configurazione dell'ambiente di sviluppo
- Creazione di uno streaming video Kinesis
- Creazione e impacchettamento di una funzione Lambda

- Configurazione del dispositivo principale Kinesis Video Streams
- Implementazione sul dispositivo principale
- Verifica del tuo stream

Segui questi passaggi per distribuire Amazon Kinesis Video Streams Edge Agent per registrare e caricare contenuti multimediali AWS IoT Greengrass da telecamere IP.

Argomenti

- [Crea un'istanza Amazon EC2 di Ubuntu](#)
- [Configura AWS IoT Greengrass V2 dispositivo principale sul dispositivo](#)
- [Crea Amazon Kinesis Video Streams e AWS Secrets Manager risorse per gli URL RTSP della tua telecamera IP](#)
- [Aggiungi le autorizzazioni al ruolo del servizio di scambio di token \(TES\)](#)
- [Installa il AWS IoT Greengrass Componente Secret Manager sul dispositivo](#)
- [Implementa Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass componente sul dispositivo](#)
- [Installa AWS IoT Greengrass componente di gestione dei log sul dispositivo](#)

Crea un'istanza Amazon EC2 di Ubuntu

Fai quanto segue per creare un'istanza Amazon EC2 di Ubuntu.

Crea un'istanza Ubuntu Amazon EC2

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. <https://console.aws.amazon.com/ec2/>

Verifica che sia selezionata la regione appropriata.

2. Scegliere Launch Instance (Avvia istanza).

Completare i seguenti campi:

- Nome: digitare un nome per l'istanza.
- Immagini dell'applicazione e del sistema operativo (Amazon Machine Image): seleziona Ubuntu.

- Tipo di istanza: seleziona `t2.large`.
 - Accesso con coppia di chiavi: crea la tua coppia di chiavi.
 - Impostazioni di rete: mantieni le impostazioni predefinite.
 - Configurazione dello spazio di archiviazione: aumenta il volume a 256 GiB.
 - Impostazioni avanzate: mantieni l'impostazione predefinita.
3. Avvia l'istanza e accedi tramite SSH.

Esegui questa operazione:

1. Seleziona `Istanze` nella barra di navigazione a sinistra, quindi seleziona l'ID dell'istanza.
 2. Scegli `Connetti` in alto a destra.
 3. Scegli il client `SSH` e segui le istruzioni sullo schermo.
 4. Apri un terminale e vai al `.pem` file scaricato (probabilmente in `~/Downloads`).
 5. La prima volta che segui queste procedure, riceverai il messaggio «Impossibile stabilire l'autenticità dell'host (...)». Digita `sì`.
4. Installa le librerie di sistema per creare Amazon Kinesis Video Streams Edge Agent sull'istanza.

```
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

Se viene visualizzata una schermata che indica che è necessario riavviare alcuni servizi, premi Invio per selezionare Ok.

Per ulteriori informazioni, consulta la Guida per l'utente di [Amazon Corretto 11](#).

Configura AWS IoT Greengrass V2 dispositivo principale sul dispositivo

Segui queste procedure per installare il software AWS IoT Greengrass core nucleus sull'istanza Amazon EC2.

Configura il AWS IoT Greengrass dispositivo principale

1. Accedi a Console di gestione AWS, <https://console.aws.amazon.com/iot/>.

Verifica che sia selezionata la regione appropriata.

2. Nella barra di navigazione a sinistra, seleziona Dispositivi Greengrass, Dispositivi Core.
3. Scegli Configura un dispositivo principale.
4. Completa i passaggi visualizzati sullo schermo.

- Fase 1: Registrare un dispositivo Greengrass core. Digita un nome per il dispositivo.
- Passaggio 2: aggiungi a un gruppo di oggetti per applicare una distribuzione continua. Seleziona Nessun gruppo.
- Fase 3: Installare il software Greengrass Core. Seleziona Linux.
 - Fase 3.1: Installare Java sul dispositivo

Java è installato come parte di [the section called “Crea un'istanza di Ubuntu”](#). Torna a questo passaggio se non hai ancora installato Java.

- Passaggio 3.2: Copia AWS le credenziali sul dispositivo

Apri l'bash/zshopzione e incolla i comandi di esportazione nell'istanza Amazon EC2.

- Passaggio 3.3: Esegui il programma di installazione
 1. Copia ed esegui i comandi Scarica il programma di installazione ed esegui il programma di installazione nell'istanza Amazon EC2 di Ubuntu.

Note

Il comando Run the installer si aggiornerà automaticamente in base al nome scelto in un passaggio precedente.

2. Prendi nota del ruolo del servizio di scambio di token (TES) che viene creato. perché sarà necessaria in seguito.

 Note

Per impostazione predefinita, viene chiamato il ruolo creato GreengrassV2TokenExchangeRole.

Crea Amazon Kinesis Video Streams e AWS Secrets Manager risorse per gli URL RTSP della tua telecamera IP

Segui queste procedure per creare i flussi e i segreti necessari in AWS Secrets Manager. Fai prima questo passaggio, perché hai bisogno degli ARN delle risorse create nelle policy.

Crea Amazon Kinesis Video Streams

Crea Amazon Kinesis Video Streams utilizzando l'API Console di gestione AWS,, AWS CLI o.

In Console di gestione AWS, apri la console [Amazon Kinesis Video Streams](#). Scegli Streams video nella barra di navigazione a sinistra.

Per ulteriori informazioni, consulta [the section called “Crea uno streaming video Amazon Kinesis”](#).

Crea segreti in AWS Secrets Manager

In Console di gestione AWS, apri la [AWS Secrets Manager console](#). Scegli Secrets nella barra di navigazione a sinistra.

Verifica che sia selezionata la regione appropriata.

1. Scegli Archivia un nuovo segreto.

a. Passaggio 1: scegli il tipo di segreto

- Selezionare Other type of secret (Altro tipo di segreti).
- Nella sezione Key/Value Coppie, aggiungi una coppia chiave-valore.

Chiave: MediaURI

 Note

La chiave deve essere. MediaURI Fa distinzione tra maiuscole e minuscole. Se lo inserisci in modo errato, l'applicazione non funziona.

Valore:*Your MediaURI*.

Example

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

- b. Passo 2: Configura il segreto. Dai un nome a questo segreto. Dai un nome come preferisci.
 - c. Fase 3: Configurare la rotazione - opzionale. Scegli Next (Successivo).
 - d. Fase 4: Revisione. Scegli Store.
2. Se il tuo segreto non viene visualizzato immediatamente, seleziona il pulsante di aggiornamento.

Scegli il nome del tuo segreto. Prendi nota dell'ARN segreto.

3. Ripeti questa procedura per ogni MediaURI da cui desideri eseguire lo streaming.

 Note

La AWS rete blocca alcune fonti RTSP pubbliche. Non è possibile accedervi dall'interno dell'istanza Amazon EC2 o se la connessione alla VPN è in modalità non gestita.

 Important

L'URL RTSP della videocamera deve trasmettere video in formato h.264. La durata del frammento non deve superare il limite indicato in. [the section called “Quote SDK di Producer”](#)

Amazon Kinesis Video Streams Edge Agent supporta solo video.

Corri `gst-discoverer-1.0 Your RtspUrl` per assicurarti che la videocamera sia raggiungibile dal tuo dispositivo.

Salva gli ARN per tutti gli stream e i segreti che hai creato. Ti servono per il passaggio successivo.

Aggiungi le autorizzazioni al ruolo del servizio di scambio di token (TES)

Assegna il ruolo di servizio di scambio di token (TES) al dispositivo che si assume le autorizzazioni per esaminare i segreti. Ciò è necessario affinché il AWS Secrets Manager AWS IoT Greengrass componente funzioni correttamente.

Aggiungere autorizzazioni al ruolo TES

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Scegli Ruoli nella barra di navigazione a sinistra e cerca il ruolo TES che hai creato in precedenza nel processo.
3. Nel menu a discesa Aggiungi autorizzazioni, seleziona Allega politiche.
4. Scegli Crea policy.
5. Scorri verso il basso e seleziona Modifica.
6. Nell'editor delle policy, scegli JSON e modifica la policy.

Sostituisci la policy con quanto segue:

Note

Sostituisci `arn:aws:kinesisvideo:*:*:stream/streamName1/*` e `arn:aws:kinesisvideo:*:*:stream/streamName2/*` con gli ARN per gli stream che hai creato in un passaggio precedente.

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. Nella pagina Add tags (Aggiungi tag), scegliere Next: Review (Successivo: rivedi).
8. Assegna un nome alla tua policy, quindi scegli Crea policy.

Un esempio di nome di politica è KvsEdgeAccessPolicy.

9. Chiudi la scheda e torna alla scheda in cui stavi allegando una policy al ruolo TES.

Scegli il pulsante di aggiornamento, quindi cerca la policy appena creata.

Seleziona la casella di controllo e scegli Allega politiche.

Nella schermata successiva, viene visualizzata una nota che dice che la politica è stata collegata correttamente al ruolo.

10. Crea e allega un'altra polizza, questa volta per i tuoi segreti.

Sostituisci la polizza con la seguente:

Note

Sostituiscila `arn:aws:secretsmanager:*:*:secret:*` con gli ARN contenenti i segreti MediaURI che hai creato in [the section called “Crea risorse per gli URL RTSP della tua telecamera IP”](#)

JSON

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

11. Crea e allega un'altra policy, questa volta per le metriche. Amazon CloudWatch Sostituisci la policy con la seguente:

JSON

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

Installa il AWS IoT Greengrass Componente Secret Manager sul dispositivo

Amazon Kinesis Video Streams Edge Agent richiede innanzitutto l'installazione del componente AWS IoT Greengrass Secret Manager sul dispositivo.

Installa il componente Secret Manager

1. Accedi Console di gestione AWS e apri la AWS IoT Core console all'indirizzo <https://console.aws.amazon.com/iot/>. Verifica che sia selezionata la regione appropriata.
2. Nella barra di navigazione a sinistra, scegli Dispositivi Greengrass, Deployments.

Scegli la distribuzione con lo stesso obiettivo dell'oggetto in cui abbiamo creato. [the section called "Configura il dispositivo AWS IoT Greengrass principale"](#)

3. Nel menu a discesa Azioni nell'angolo in alto a destra, scegli Revisione.

Nel pop-up che appare, scegli Revisione della distribuzione.

4. Completa le seguenti sezioni:
 - Fase 1: Specificare l'obiettivo. Scegli Next (Successivo).
 - Fase 2: Seleziona i componenti.
 - Verifica che il componente AWS.Greengrass.cli sia selezionato. Non disinstallare questo componente.
 - Attiva l'opzione Mostra solo i componenti selezionati e cerca aws.greengrass.SecretManager.
 - Seleziona la casella accanto a aws.greengrass.SecretManager, quindi scegli Avanti.
 - Fase 3: Configurazione dei componenti. Configura il componente AWS IoT Greengrass Secret Manager per scaricare i segreti dall'interno dell' AWS IoT Greengrass ambiente.

Seleziona aws.greengrass.SecretManagercomponente, quindi scegli Configura componente.

Nella schermata che appare, aggiorna gli AWS Secrets Manager ARN nella casella Configurazione da unire.

Note

Sostituisci `arn:aws:secretsmanager:*:*:secret:*` con gli ARN i segreti in cui hai creato. [the section called "Crea risorse per gli URL RTSP della tua telecamera IP"](#)

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

 Note

`cloudSecrets` è un elenco di oggetti con la chiave `arn`. Per ulteriori informazioni, [consulta la](#) sezione sulla configurazione di Secret Manager nella AWS IoT Greengrass Version 2 Developer Guide.

Al termine, seleziona **Conferma**, quindi scegli **Avanti**.

- Passaggio 4: configura le impostazioni avanzate. Seleziona **Avanti**.
 - Fase 5: Revisione. Seleziona **Implementa**.
5. Verifica che il AWS Secrets Manager componente e le autorizzazioni siano stati installati correttamente.

Nell'istanza Amazon EC2 di Ubuntu, digita `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` per verificare che il componente abbia ricevuto la configurazione aggiornata.

6. Ispeziona i log AWS IoT Greengrass principali.

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

Verifica la presenza di errori di distribuzione.

Se si è verificato un errore, rivedi la distribuzione per rimuovere il componente.

```
aws.greengrass.SecretManager
```

Digita `sudo service greengrass restart` per riavviare il servizio AWS IoT Greengrass principale.

Se l'errore di distribuzione era correlato alla mancanza di autorizzazioni, consulta la [the section called "Aggiungi autorizzazioni al ruolo TES"](#) sezione per assicurarti che il ruolo TES disponga delle autorizzazioni appropriate. Quindi, ripeti questa sezione.

7. Aggiorna i segreti sul componente AWS IoT Greengrass Secret Manager

Important

Il componente AWS IoT Greengrass Secret Manager recupera e memorizza nella cache i segreti solo quando la distribuzione viene aggiornata.

Per aggiornare i segreti sul componente AWS IoT Greengrass Secret Manager, seguite i passaggi precedenti da 1 a 6, con le seguenti modifiche.

Fase 3: Configurazione dei componenti. Configura il componente AWS IoT Greengrass Secret Manager per scaricare i segreti dall'interno dell' AWS IoT Greengrass ambiente.

Seleziona `aws.greengrass.SecretManagercomponente`, quindi scegli `Configura componente`.

Nella schermata che appare, incolla `[""]` la casella `Reimposta percorsi` e aggiorna gli AWS Secrets Manager ARN nella casella `Configurazione da unire`.

Per ulteriori informazioni, consulta [Reimposta gli aggiornamenti](#).

Implementa Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass componente sul dispositivo

Effettua le seguenti operazioni per distribuire il componente Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass sul dispositivo:

Implementa il componente

1. Scarica il `tar` file utilizzando il link fornito.

Se hai completato il modulo di interesse per Amazon Kinesis Video Streams Edge Agent, controlla la tua email per trovare il link per il download. Se non hai completato il modulo, compilalo [qui](#).

2. Verifica il checksum.
3. Estrai i file binari e il jar nel tuo dispositivo.

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

Dopo l'estrazione, la struttura delle cartelle sarà la seguente:

```
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

Il nome della cartella di rilascio deve essere impostato in modo da riflettere l'ultimo numero binario della versione. Ad esempio, una versione 1.0.0 avrà il nome della cartella impostato come 1.0.0.

4. Crea il jar delle dipendenze.

Note

Il jar incluso nel file `kvs-edge-agent.tar.gz` non contiene le dipendenze. Usa i seguenti passaggi per creare quelle librerie.

Accedere alla `kvs-edge-agent` cartella che contiene `pom.xml`.

Tipo `mvn clean package`.

Questo genererà un file jar contenente le dipendenze richieste da Amazon Kinesis Video Streams Edge Agent. `kvs-edge-agent/target/libs.jar`

5. Inserisci il file `libs.jar` nella cartella che contiene gli artefatti del componente.

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

6. Facoltativo. Configura le proprietà. Amazon Kinesis Video Streams Edge Agent accetta le seguenti variabili di ambiente in AWS IoT Greengrass modalità:

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_REGION	Sì	La regione utilizzata. Esempio: us-west-2 AWS IoT Greengrass Il software principale imposta automaticamente questo

Nome della variabile di ambiente	Richiesto	Descrizione
		valore per te. Per ulteriori informazioni, consulta l'argomento di riferimento sulle variabili di ambiente dei componenti nella Guida per AWS IoT Greengrass Version 2 gli sviluppatori.
GST_PLUGIN_PATH	Sì	Percorso del file che punta alla cartella contenente le librerie gstreamer e quelle dipendenti dalla IngestorPipelineJNI piattaforma. Questo permette a GStreamer di caricare questi plugin. Per ulteriori informazioni, consulta the section called “Scarica, crea e configura l'elemento GStreamer”. Esempio: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i>

Nome della variabile di ambiente	Richiesto	Descrizione
LD_LIBRARY_PATH	Sì	Percorso del file che punta alla directory contenente le librerie e quelle dipendenti dalla cproducer piattaforma. KinesisVideoProducer Esempio: <i>/download-location</i> /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgentVersion</i> /lib/
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	No	Determina se Amazon Kinesis Video Streams Edge Agent pubblicherà le metriche sullo stato del lavoro su Amazon CloudWatch Valori accettati: TRUE/FALSE (senza distinzione tra maiuscole e minuscole). Il valore predefinito è FALSE se non fornito. Esempio: FALSE

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_KVS_EDGE_LOG_LEVEL	No	Il livello di registrazione degli output di Amazon Kinesis Video Streams Edge Agent. Valori accettati: • OFF • ALL • FATALE • ERRORE • WARN • INFO, impostazione predefinita, se non fornita • DEBUG • TRACE Esempio: INFO
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	No	Una volta che il file di registro raggiunge questa dimensione, si verificherà un rollover. • Minimo: 1 • Massimo: 100 • Valore predefinito: 20, se non fornito • Unità: Megabyte (MB) Esempio: 5

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	No	Il percorso del file che punta alla directory in cui vengono generati i log di Amazon Kinesis Video Streams Edge Agent. Il valore predefinito è se non fornito. <code>./log</code> Esempio: <code>/file/path/</code>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	No	Il numero di registri trasferiti da conservare prima dell'eliminazione. • Minimo: 1 • Massimo: 100 • Valore predefinito: 10, se non fornito Esempio: 20
AWS_KVS_EDGE_RECORDING_DIRECTORY	No	Verrà scritto il percorso del file che punta alla directory in cui è stato registrato il file multimediale. Il valore predefinito è la directory corrente se non viene fornita. Esempio: <code>/file/path/</code>

Nome della variabile di ambiente	Richiesto	Descrizione
GREENGRASS_ROOT_DIRECTORY	No	Il percorso del file alla AWS IoT Greengrass directory principale. Il valore predefinito è /greengrass/v2/ se non viene fornito. Esempio: <i>/file/path/</i>
GST_DEBUG	No	Specifica il livello dei log di GStreamer da emettere. Per maggiori informazioni, consulta la documentazione di GStreamer. Esempio: 0
GST_DEBUG_FILE	No	Specifica il file di output dei log di debug di GStreamer. Se non è impostato, i log di debug restituiscono un errore standard. Per maggiori informazioni, consulta la documentazione di GStreamer. Esempio: <i>/tmp/gstreamer-logging .log</i>

Apri `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` e modifica lo script di esecuzione per aggiungere una qualsiasi delle variabili di ambiente precedenti.

⚠ Important

Assicuratevi che lo script di esecuzione modificato non contenga caratteri di tabulazione. Il software AWS IoT Greengrass principale non sarà in grado di leggere la ricetta.

7. Distribuisci il componente Amazon Kinesis Video Streams Edge Agent. AWS IoT Greengrass

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"
```

Per ulteriori informazioni, consulta le seguenti sezioni nella Guida per gli AWS IoT Greengrass Version 2 sviluppatori:

- [AWS IoT Greengrass Comandi CLI](#)
- [Distribuisci AWS IoT Greengrass i componenti sui dispositivi](#)

8. Invia le configurazioni all'applicazione utilizzando. AWS CLI

- a. Crea un nuovo file, *example-edge-configuration.json*.

Incolla il codice seguente nel file. Questa è una configurazione di esempio che registra ogni giorno dalle 9:00:00 AM alle 16:59:59 PM (in base all'ora di sistema sul dispositivo). AWS IoT Inoltre, carica i file multimediali registrati ogni giorno dalle 19:00:00 alle 21:59:59.

Per ulteriori informazioni, consulta [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. Digita quanto segue AWS CLI per inviare il file ad Amazon Kinesis Video Streams Edge Agent:

```

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

```

9. Ripeti il passaggio precedente per ogni stream per Amazon Kinesis Video Streams Edge Agent.

Installa AWS IoT Greengrass componente di gestione dei log sul dispositivo

Note

Fai attenzione alle [CloudWatch quote](#).

Segui queste procedure per configurare i log di Amazon Kinesis Video Streams Edge Agent in modo che vengano caricati automaticamente CloudWatch utilizzando il componente di gestione dei log di AWS IoT Greengrass. Questa fase è opzionale.

Installa AWS IoT Greengrass componente di gestione dei log

1. Verificare che il ruolo AWS IoT Greengrass del dispositivo disponga delle autorizzazioni <https://docs.aws.amazon.com/greengrass/v2/developerguide/log-manager-component.html#log-manager-component-requirements> appropriate.
 - a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - b. Fai clic su **Ruoli** nella barra di navigazione a sinistra.
 - c. Scegli il nome del ruolo TES creato in [the section called “Configura il dispositivo AWS IoT Greengrass principale”](#). Usa la barra di ricerca se necessario.
 - d. Seleziona la `GreengrassV2TokenExchangeRoleAccess` politica.
 - e. Seleziona la scheda JSON e verifica che la policy sia simile alla seguente:

JSON

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

- f. Se la `GreengrassV2TokenExchangeRoleAccess` policy non esiste o se mancano alcune autorizzazioni richieste, crea una nuova policy IAM con queste autorizzazioni e allegala

al ruolo TES creato in. [the section called “Configura il dispositivo AWS IoT Greengrass principale”](#)

2. Accedi a Console di gestione AWS e apri la AWS IoT Core console all'indirizzo. <https://console.aws.amazon.com/iot/> Verifica che sia selezionata la regione appropriata.
3. Nella barra di navigazione a sinistra, scegli Dispositivi Greengrass, Deployments.

Scegli la distribuzione con lo stesso obiettivo dell'oggetto in cui hai creato. [the section called “Configura il dispositivo AWS IoT Greengrass principale”](#)

4. Nell'angolo in alto a destra, seleziona Azioni, quindi scegli Revisione.

Nel pop-up che appare, scegli Revisione della distribuzione.

5. Completa le seguenti sezioni:
 - a. Fase 1: Specificare l'obiettivo. Scegli Next (Successivo).
 - b. Fase 2: Seleziona i componenti.
 - i. Verifica che il componente `aws.greengrass.cli` e `aws.greengrass.SecretManager` componenti sono ancora selezionati.

 Important

Non disinstallare questi componenti.

- ii. Attiva l'opzione Mostra solo i componenti selezionati e cerca `aws.greengrass.LogManager`.
 - iii. Seleziona la casella accanto a `aws.greengrass.LogManager`, quindi scegli Avanti.
 - c. Fase 3: Configurazione dei componenti. Configura il componente di gestione dei AWS IoT Greengrass log per caricare i log generati da Amazon Kinesis Video Streams Edge Agent.

Seleziona `aws.greengrass.LogManager` componente, quindi scegli Configura componente.

Nella schermata che appare, incolla la seguente configurazione del gestore di log nella casella Configurazione da unire.

```
{
  "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

La `logFileDirectoryPath` configurazione precedente presuppone che venga utilizzata la posizione di output di registrazione predefinita.

 Note

Per ulteriori informazioni su ciascuno dei parametri per la configurazione del log manager, vedere la [sezione](#) Log manager della Developer Guide. AWS IoT Greengrass Version 2

Al termine, seleziona Conferma, quindi scegli Avanti.

- d. Passaggio 4: configura le impostazioni avanzate. Seleziona Avanti.
 - e. Passaggio 5: revisione. Seleziona Implementa.
6. Verificare che il componente e le autorizzazioni del gestore dei AWS log siano stati installati correttamente.
 7. Nell'istanza Amazon EC2 di Ubuntu, digita `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` per verificare che il componente abbia ricevuto la configurazione aggiornata.
 8. Ispeziona i log AWS IoT Greengrass principali.

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

Verifica la presenza di errori di distribuzione.

Se si è verificato un errore, rivedi la distribuzione per rimuovere il componente.

`aws.greengrass.LogManager`

Digita `sudo service greengrass restart` per riavviare il servizio AWS IoT Greengrass principale.

Se l'errore di distribuzione era correlato alla mancanza di autorizzazioni, verifica [the section called "Aggiungi autorizzazioni al ruolo TES"](#) che il ruolo TES disponga delle autorizzazioni appropriate. Quindi, ripeti questa sezione.

Domande frequenti su Amazon Kinesis Video Streams Edge Agent

Di seguito sono riportate alcune domande comuni relative al servizio Amazon Kinesis Video Streams Edge Agent.

Quali sistemi operativi sono supportati da Amazon Kinesis Video Streams Edge Agent?

Amazon Kinesis Video Streams Edge Agent attualmente supporta i seguenti sistemi operativi:

Ubuntu

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

AL 2

- amzn2
 - AMD64 amazonlinux:2.0.20210219.0-amd64 (Palla di neve)

Amazon Kinesis Video Streams Edge Agent supporta i contenuti multimediali? H.265

Amazon Kinesis Video Streams Edge Agent supporta solo stream elementari. H.264

Amazon Kinesis Video Streams Edge Agent funziona in AL2?

Sì.

Come posso eseguire più stream all'interno di AWS IoT cosa o dispositivo?

Inviare un altro [StartEdgeConfigurationUpdate](#) allo stesso HubDeviceArn Amazon Kinesis Video AWS Secrets Manager Streams/ARN ma diverso.

Come posso modificare un messaggio

StartEdgeConfigurationUpdate dopo averlo inviato?

Invia un aggiornamento [StartEdgeConfigurationUpdate](#) allo stesso HubDeviceArn con lo stesso ARN di Amazon Kinesis Video Streams. Quando l'applicazione riceve il messaggio da Amazon

Kinesis Video Streams, sovrascrive la configurazione precedente per quel flusso. Le modifiche avverranno allora.

Hai qualche esempio comune **ScheduleConfigs**?

Amazon Kinesis Video Streams Edge Agent utilizza l'ora di sistema del dispositivo su cui è in esecuzione.

Description	ScheduleExpression	DurationInSeconds
24/7 registrazione, caricamento ogni ora	(null ScheduleConfig)	
9:00:00 AM - 16:59:59 PM tutti i giorni	0 0 9-16 * * ? *	3599
9:00:00 AM - 16:59:59 PM nei giorni feriali	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 AM - 16:59:59 PM nei fine settimana	0 0 9-16 ? * SAT,SUN *	3599
10:00:00 PM - 23:59:59 PM nei giorni feriali	0 0 22,23 ? * MON-FRI *	3599
9:00:00 AM - 10:00:00 AM tutti i giorni	0 0 9 * * ? *	3600
16:00:00 - 17:59:59 PM tutti i giorni	0 0 16-17 * * ? *	3599

Per altri esempi, consulta la documentazione di [Quartz](#).

Esiste un limite massimo di streaming?

Amazon Kinesis Video Streams Edge Agent ha attualmente un limite fisso di 16 stream per dispositivo. Utilizza l'[DeleteEdgeConfiguration](#) API per eliminare gli stream da un dispositivo. L'aggiornamento di una configurazione per lo stesso stream utilizzando il [StartEdgeConfigurationUpdate](#) non aumenta il numero di stream del dispositivo.

Come posso riavviare un processo che ha ricevuto un errore?

Se si verifica un errore, Amazon Kinesis Video Streams Edge Agent tenterà di riavviare il processo. Tuttavia, in caso di alcuni errori (ad esempio errori di configurazione), è necessario riavviare manualmente il processo.

Per determinare quali processi devono essere riavviati manualmente, consulta la `FatalError` metrica in [the section called “Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch”](#)

Invia nuovamente [StartEdgeConfigurationUpdate](#) per riavviare il processo per lo stream.

Come posso monitorare lo stato del mio Amazon Kinesis Video Streams Edge Agent?

Per ulteriori informazioni, consulta [the section called “Monitora Amazon Kinesis Video Streams Edge Agent con CloudWatch”](#).

Trasmetti video tramite un VPC

Questa versione beta è disponibile in anteprima nella regione Europa (Parigi), eu-west-3. Per accedere a questi componenti e alla nostra guida introduttiva, [inviaci un'e-mail](#).

Il servizio endpoint VPC di Amazon Kinesis Video Streams consente di trasmettere e consumare video attraverso la rete Amazon senza che i dati passino attraverso la rete Internet pubblica.

Per richiedere l'accesso, [inviaci](#) un'e-mail con le seguenti informazioni:

- ID account
- Streaming ARN
- ID VPC

Note

Potrebbe essere necessaria fino a una settimana per aggiungerti al servizio.

Se non hai mai lavorato con gli endpoint VPC in passato, consulta le seguenti informazioni per acquisire familiarità con il concetto:

- [AWS PrivateLink sfondo](#)
- [Guida introduttiva a VPC](#)

Informazioni aggiuntive

Una volta aggiunto alla versione beta, ti invieremo un'email con un link a ulteriori informazioni su questa funzionalità.

Procedure per gli endpoint VPC

Quote

Le principali differenze di quota sono:

- Quota inferiore per tutte le API con larghezza di banda (2 mbps):
 - PutMedia
 - GetMedia
 - GetMediaForFragmentList
- 10 stream consentiti per cliente

Creare un endpoint

Una volta elencati i dati consentiti, riceverai il nome del servizio di endpoint VPC per Amazon Kinesis Video Streams. Assomiglierà a: `com.amazonaws.region.kinesisvideo`

Crea un endpoint VPC di [interfaccia](#) per Amazon Kinesis Video Streams utilizzando la console Amazon VPC o (). AWS Command Line Interface AWS CLI

Nel, digita quanto segue: AWS CLI

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

Important

Il traffico all'interno del tuo VPC utilizzerà un DNS privato per il routing sull'endpoint. Se non lo abiliti, dovrai implementare la tua logica DNS. Per ulteriori informazioni sul DNS privato, consulta [AWS PrivateLink la documentazione](#).

Per ulteriori informazioni sull' AWS CLI opzione, consulta <https://docs.aws.amazon.com/cli/latest/reference/ec2/create-vpc-endpoint.html> create-vpc-endpoint.

Controlla l'accesso agli endpoint

Puoi allegare una policy sugli endpoint al tuo endpoint VPC che controlli l'accesso ad Amazon Kinesis Video Streams. Questa policy specifica le informazioni riportate di seguito:

- il principale che può eseguire azioni,
- le azioni che possono essere eseguite e

- le risorse su cui è possibile eseguire le azioni.

Per ulteriori informazioni, consulta [Controllare l'accesso ai servizi con gli endpoint VPC utilizzando le politiche degli endpoint](#) nella Guida. AWS PrivateLink

Di seguito è riportato un esempio di policy sugli endpoint per Amazon Kinesis Video Streams. Se collegata a un endpoint, questa policy nega l'accesso alle PutMedia azioni elencate a tutti i principali su tutte le risorse.

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

Esempi di Amazon Kinesis Video Streams

I seguenti esempi di codice dimostrano come lavorare con l'API Kinesis Video Streams:

Esempi: invio di dati a Kinesis Video Streams

- [Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink](#): mostra come creare l'SDK per produttori di Kinesis Video Streams da utilizzare come destinazione GStreamer.
- [Esegui l'elemento GStreamer in un contenitore Docker](#): mostra come utilizzare un'immagine Docker predefinita per inviare video Real-Time Streaming Protocol (RTSP) da una telecamera IP a Kinesis Video Streams.
- [Esempio: streaming da una fonte RTSP](#): mostra come creare un'immagine Docker personalizzata e inviare video RTSP da una telecamera IP a Kinesis Video Streams.
- [Esempio: invio di dati a Kinesis Video Streams tramite l'API PutMedia](#): mostra come inviare dati a Kinesis Video Streams che sono già in formato contenitore (MKV) utilizzando l'API. [Usa la libreria del produttore Java PutMedia](#)

Esempi: recupero di dati da Kinesis Video Streams

- [KinesisVideoExample](#): mostra come analizzare e registrare frammenti video utilizzando la libreria di parser Kinesis Video Streams.
- [Esempio: analisi e rendering dei frammenti di Kinesis Video Streams](#): mostra come analizzare e renderizzare i frammenti del flusso video Kinesis utilizzando JCodec e JFrame. <http://jcodec.org/> <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>

Esempi: riproduzione di dati video

- [Esempio: usa HLS in HTML e JavaScript](#): mostra come recuperare una sessione di streaming HLS per uno streaming video Kinesis e riprodurla in una pagina web.

Prerequisiti

- Nel codice di esempio, si forniscono le credenziali specificando un profilo impostato nel file del profilo delle AWS credenziali o fornendo le credenziali nelle proprietà di sistema Java del proprio

ambiente di sviluppo integrato (IDE). Se non l'hai già fatto, configura prima le tue credenziali. Per ulteriori informazioni, consulta [Configurare AWS le credenziali e l'area per lo sviluppo](#).

- Per visualizzare ed eseguire il codice, ti consigliamo di utilizzare un IDE Java, come uno dei seguenti:
 - [Eclipse Java Neon](#)
 - [JetBrains IntelliJ IDEA](#)

Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink

Questo argomento descrive come creare l'SDK per produttori di Amazon Kinesis Video Streams da utilizzare come plug-in GStreamer.

Argomenti

- [Scarica, crea e configura l'elemento GStreamer](#)
- [Esegui l'elemento GStreamer](#)
- [Esempio di comandi di avvio di GStreamer](#)
- [Esegui l'elemento GStreamer in un contenitore Docker](#)
- [Riferimento ai parametri degli elementi GStreamer](#)

[GStreamer](#) è un popolare framework multimediale utilizzato da più fotocamere e sorgenti video per creare pipeline multimediali personalizzate combinando plugin modulari. Il plugin Kinesis Video Streams GStreamer semplifica l'integrazione della pipeline multimediale GStreamer esistente con Kinesis Video Streams. Dopo aver integrato GStreamer, puoi trasmettere video da una webcam o da una videocamera RTSP (Real Time Streaming Protocol) a Kinesis Video Streams per la riproduzione, l'archiviazione e ulteriori analisi in tempo reale o in un secondo momento.

Il plug-in GStreamer gestisce automaticamente il trasferimento del flusso video su Kinesis Video Streams incapsulando la funzionalità fornita dal produttore di Kinesis Video Streams SDK in un elemento sink GStreamer. `kvssink` Il framework GStreamer fornisce un ambiente gestito standard per la costruzione di un flusso multimediale da un dispositivo, ad esempio una videocamera o altre sorgente video per ulteriori elaborazioni, il rendering o lo storage.

La pipeline GStreamer in genere contiene il link tra una sorgente (videocamera) e l'elemento sink (un lettore per il rendering del video o lo storage per il recupero offline). In questo esempio, si utilizza

l'elemento SDK Producer come sink o destinazione multimediale, per la fonte video (webcam o videocamera IP). L'elemento plug-in che incapsula l'SDK invia quindi il flusso video a Kinesis Video Streams.

Questo argomento descrive come costruire una pipeline multimediale GStreamer in grado di trasmettere video da una sorgente video, come una webcam o un flusso RTSP, in genere connessa tramite fasi di codifica intermedie (utilizzando la codifica) a Kinesis Video Streams. H.264 Quando lo streaming video è disponibile come streaming video Kinesis, è possibile utilizzarlo [the section called “Trasmetti in streaming utilizzando la libreria parser”](#) per l'ulteriore elaborazione, riproduzione, archiviazione o analisi del flusso video.

Scarica, crea e configura l'elemento GStreamer

L'esempio del plugin GStreamer è incluso nell'SDK per produttori C++ di Kinesis Video Streams. Per informazioni sui prerequisiti e il download di SDK consulta [Scarica e configura il codice della libreria di produzione C++](#).

Puoi creare il sink GStreamer del produttore SDK come libreria dinamica su macOS, Ubuntu, Raspberry Pi o Windows. Il plugin GStreamer si trova nella directory `build`. Per caricare questo plugin, deve essere nel tuo. `GST_PLUGIN_PATH` Esegui il comando seguente:

```
export GST_PLUGIN_PATH=`pwd`/build
```

Note

Su macOS, è possibile solo eseguire lo streaming di video da una videocamera di rete durante l'esecuzione di GStreamer in un container Docker. Lo streaming video da una videocamera USB su macOS in un container Docker non è supportato.

Esegui l'elemento GStreamer

Per eseguire GStreamer con l'elemento Kinesis Video Streams producer SDK come sink, usate il comando. `gst-launch-1.0` Utilizzate elementi upstream appropriati per l'uso del plugin GStreamer. Ad esempio, [v4l2src](#) per dispositivi v4l2 su sistemi Linux o [rtspsrc](#) per dispositivi RTSP. Specificare `kvssink` come sink (destinazione finale della pipeline) per inviare video all'SDK Producer.

Oltre a [fornire le credenziali](#) e [fornire una regione](#), l'kvssinkelemento ha il seguente parametro obbligatorio:

- `stream-name`— Il nome della destinazione Kinesis Video Streams.

Per informazioni sui kvssink parametri opzionali consulta [Riferimento ai parametri degli elementi GStreamer](#).

Per le informazioni più recenti sui plugin e sui parametri di GStreamer, consulta GStreamer Plugins. https://gstreamer.freedesktop.org/documentation/plugins_doc.html?gi-language=c Puoi anche utilizzare il comando «`gst-inspect-1.0` seguito dal nome» di un elemento o plug-in di GStreamer per stamparne le informazioni e verificare che sia disponibile sul tuo dispositivo:

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

Se la compilazione kvssink non è riuscita o GST_PLUGIN_PATH non è impostato correttamente, l'output sarà simile a questo:

```
No such element or plugin 'kvssink'
```

Esempio di comandi di avvio di GStreamer

I seguenti esempi dimostrano come utilizzare il plugin kvssink GStreamer per lo streaming di video da diversi tipi di dispositivi.

Esempio 1: streaming di video da una videocamera RTSP su Ubuntu

Il comando seguente crea una pipeline GStreamer su Ubuntu che esegue flussi in streaming da una videocamera di rete RTSP, utilizzando il plugin [rtspsrc](#) GStreamer:

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !  
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Esempio 2: codifica e trasmetti video da una videocamera USB su Ubuntu

Il comando seguente crea una pipeline GStreamer su Ubuntu che codifica il flusso da una videocamera USB in H.264 formato e lo trasmette a Kinesis Video Streams. Questo esempio

utilizza il plugin gStreamer v4l2src. <https://gstreamer.freedesktop.org/documentation/video4linux2/v4l2src.html?gi-language=c#v4l2src-page>

```
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"
```

Esempio 3: Trasmetti video precodificato da una videocamera USB su Ubuntu

Il comando seguente crea una pipeline GStreamer su Ubuntu che trasmette video che la videocamera ha già codificato in formato Kinesis Video Streams. H.264 Questo esempio utilizza il plugin gStreamer v4l2src. <https://gstreamer.freedesktop.org/documentation/video4linux2/v4l2src.html?gi-language=c#v4l2src-page>

```
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"
```

Esempio 4: streaming di video da una telecamera di rete su macOS

Il comando seguente crea una pipeline GStreamer su macOS che trasmette video a Kinesis Video Streams da una telecamera di rete. In questo esempio viene utilizzato il plugin [rtspsrc](#) GStreamer.

```
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"
```

Esempio 5: streaming di video da una telecamera di rete su Windows

Il comando seguente crea una pipeline GStreamer su Windows che trasmette video a Kinesis Video Streams da una telecamera di rete. In questo esempio viene utilizzato il plugin [rtspsrc](#) GStreamer.

```
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"
```

Esempio 6: streaming di video da una videocamera su Raspberry Pi

Il comando seguente crea una pipeline GStreamer su Raspberry Pi che trasmette video a Kinesis Video Streams. Questo esempio utilizza il plugin gStreamer v4l2src. <https://gstreamer.freedesktop.org/documentation/video4linux2/v4l2src.html?gi-language=c#v4l2src-page>

```
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"
```

Esempio 7: Trasmetti audio e video in Raspberry Pi e Ubuntu

Scopri come [eseguire il comando gst-launch-1.0 per avviare lo streaming di audio e video su Ubuntu. Raspberry-Pi](#)

Esempio 8: trasmetti in streaming audio e video da sorgenti di dispositivi in macOS

Scopri come [eseguire il comando gst-launch-1.0 per avviare lo streaming audio e video in MacOS.](#)

Esempio 9: carica un file MKV che contiene sia audio che video

Scopri come [eseguire il comando gst-launch-1.0 per caricare il file MKV che contiene sia audio che video](#). Avrai bisogno di un file di test MKV con contenuti multimediali codificati h.264 e AAC.

Esegui l'elemento GStreamer in un contenitore Docker

Docker è una piattaforma per sviluppare, distribuire ed eseguire le applicazioni utilizzando i contenitori. L'uso di Docker per creare la pipeline GStreamer standardizza l'ambiente operativo per Kinesis Video Streams, il che semplifica la creazione e l'uso dell'applicazione.

Per installare e configurare Docker, consulta le seguenti informazioni:

- [Istruzioni di download di Docker](#)
- [Nozioni di base su Docker](#)

Dopo aver installato Docker, puoi scaricare Kinesis Video Streams C++ Producer SDK (e il plugin GStreamer) da Amazon Elastic Container Registry utilizzando uno dei comandi forniti di seguito.

```
docker pull
```

Per eseguire GStreamer con l'elemento Kinesis Video Streams producer SDK come sink in un contenitore Docker, procedi come segue:

Argomenti

- [Autentica il tuo client Docker](#)
- [Scaricare l'immagine Docker per Ubuntu, macOS, Windows o Raspberry Pi](#)
- [Esegui l'immagine Docker](#)

Autentica il tuo client Docker

Autentica il tuo client Docker nel registro Amazon ECR dal quale desideri estrarre la tua immagine. È necessario ottenere i token di autenticazione per ogni registro utilizzato. I token sono validi per 12 ore. Per maggiori informazioni, consulta [Autorizzazioni del registro](#) nella Guida per l'utente di Amazon Elastic Container Registry.

Example: autenticazione con Amazon ECR

Per autenticarti con Amazon ECR, copia e incolla il seguente comando come mostrato.

```
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
```

In caso di esito positivo, l'output stampa `Login Succeeded`.

Scaricare l'immagine Docker per Ubuntu, macOS, Windows o Raspberry Pi

Scaricare l'immagine Docker per l'ambiente Docker utilizzando uno dei comandi riportati di seguito, a seconda del sistema operativo in uso:

Scaricare l'immagine Docker per Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Scaricare l'immagine Docker per macOS

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Scaricare l'immagine Docker per Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

Scaricare l'immagine Docker per Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

Per verificare che l'immagine sia stata aggiunta, eseguire il comando riportato di seguito:

```
docker images
```

Esegui l'immagine Docker

Utilizzare uno dei seguenti comandi per eseguire l'immagine Docker, a seconda del sistema operativo in uso:

Esegui l'immagine Docker su Ubuntu

```
sudo docker run -it --network="host" --device=/dev/video0 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Esegui l'immagine Docker su macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Esegui l'immagine Docker su Windows

```
docker run -it 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-windows AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY RTSP_URL STREAM_NAME
```

Esegui l'immagine Docker su Raspberry Pi

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc
546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-
pi /bin/bash
```

Docker avvia il contenitore e ti presenta un prompt dei comandi per l'utilizzo dei comandi all'interno del contenitore.

Nel container, impostare le variabili di ambiente utilizzando il comando seguente:

```
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
```

Inizia lo streaming kvssink utilizzando per `gst-launch-1.0` eseguire una pipeline appropriata per il tuo dispositivo e la tua sorgente video. Ad esempio pipeline, vedi. [Esempio di comandi di avvio di GStreamer](#)

Riferimento ai parametri degli elementi GStreamer

Per inviare video all'SDK C++ del produttore di Amazon Kinesis Video Streams, devi specificare kvssink come sink o destinazione finale della pipeline. La documentazione di riferimento fornisce informazioni sui parametri kvssink obbligatori e facoltativi. Per ulteriori informazioni, consulta [the section called "Plugin GStreamer - kvssink"](#).

Argomenti

- [the section called "Fornisci le credenziali a kvssink"](#)
- [the section called "Fornisci una regione a kvssink "](#)
- [the section called "parametri opzionali kvssink"](#)

Fornisci le credenziali a kvssink

Per consentire all'elemento kvssink GStreamer di effettuare richieste a AWS, fornisci le AWS credenziali da utilizzare quando chiama il servizio Amazon Kinesis Video Streams. La catena di fornitori di credenziali cerca le credenziali nel seguente ordine:

1. AWS IoT credenziali

Per configurare le AWS IoT credenziali, vedere. [the section called “Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT”](#)

Il valore del `iot-credentials` parametro deve iniziare con `iot-certificate`, ed essere seguito da un elenco separato da virgole delle seguenti coppie =. *key value*

Chiave	Richiesto	Descrizione
<code>ca-path</code>	Sì	Percorso del file al certificato CA utilizzato per stabilire l'attendibilità con il servizio di backend tramite TLS. Example Esempio: <i>/file/path/to/certificate.pem</i>
<code>cert-path</code>	Sì	Percorso del file del certificato. X.509 Example Esempio: <i>/file/path/to/certificateID -certificate.pem.crt</i>
<code>endpoint</code>	Sì	L'endpoint del fornitore di AWS IoT Core credenziali per il tuo account. AWS Consulta la Guida per gli sviluppatori. AWS IoT Example Esempio: <i>credential-account-specific</i>

Chiave	Richiesto	Descrizione
		<i>-prefix</i> .credenti als.iot. <i>aws-regio</i> <i>n</i> .amazonaws.com
key-path	Sì	Percorso del file alla chiave privata utilizzata nella coppia di public/private chiavi. Example Esempio: <i>/file/path/to/certifica</i> <i>teID</i> -private.pem.key
role-aliases	Sì	Il nome dell'alias del ruolo che rimanda al ruolo AWS IAM da utilizzare per la connessione. AWS IoT Core Example Esempio: <i>KvsCamera</i> <i>IoTRoleAlias</i>
iot-thing-name	No	Il iot-thing-name è facoltativo. Se non iot-thing-name viene fornito, viene stream-name utilizzato o il valore del parametro. Example Esempio: <i>kvs_examp</i> <i>le_camera</i>

Example

Esempio:

```
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. Variabili di ambiente

Per kvssink utilizzare le credenziali dell'ambiente, imposta le seguenti variabili di ambiente:

Nome della variabile di ambiente	Richiesto	Descrizione
AWS_ACCESS_KEY_ID	Sì	La chiave di AWS accesso utilizzata per accedere ad Amazon Kinesis Video Streams.
AWS_SECRET_ACCESS_KEY	Sì	La chiave AWS segreta associata alla chiave di accesso.
AWS_SESSION_TOKEN	No	Specifica il valore del token di sessione richiesto se si utilizzano credenziali di sicurezza temporanee direttamente dalle AWS STS operazioni.

L'impostazione della variabile di ambiente modifica il valore utilizzato fino al termine della sessione della shell o finché non imposti la variabile su un valore diverso. Per rendere persistenti le variabili nelle sessioni future, impostale nello script di avvio della shell.

3. chiave di accesso, parametri a chiave segreta

Per specificare le credenziali direttamente come kvssink parametro, impostate i seguenti parametri:

kvssinkNome del parametro	Richiesto	Descrizione
access-key	Sì	La chiave di AWS accesso utilizzata per accedere ad Amazon Kinesis Video Streams.
secret-key	Sì	La chiave AWS segreta associata alla chiave di accesso.
session-token	No	Specifica il valore del token di sessione richiesto se si utilizzano credenziali di sicurezza temporanee direttamente dalle AWS STS operazioni.

Example

Utilizzo di credenziali statiche:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE"
```

Example

Utilizzo di credenziali temporanee:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE" session-token="STEXAMPLE"
```

4. File delle credenziali

Important

Se hai selezionato uno dei metodi precedenti, non puoi utilizzare il `credential-path` `kvssink` parametro.

kvssinkNome del parametro	Richiesto	Descrizione
<code>credential-path</code>	Sì	Percorso del file di testo contenente le credenziali in un formato specifico.

Il file di testo deve contenere le credenziali in uno dei seguenti formati:

- CREDENZIALI *YourAccessKey YourSecretKey*
- CREDENZIALI *YourAccessKey Expiration YourSecretKey SessionToken*

Example

Esempio: il *credentials.txt* file si trova in `/home/ubuntu` e contiene quanto segue:

CREDENTIALS *AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE*

Per utilizzarlo `kvssink`, digita:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"  
credential-path="/home/ubuntu/credentials.txt"
```

Note

Il tempo di scadenza dovrebbe essere di almeno $5 + 30 + 3 = 38$ secondi nel futuro. Il periodo di tolleranza è definito come la `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` variabile in [IotCredentialProvider.h](#). Se le credenziali sono troppo vicine alla scadenza all'avvio `kvssink`, viene visualizzato il codice di errore. `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`

⚠ Important

kvssink non modifica il file delle credenziali. Se utilizzi credenziali temporanee, il file delle credenziali deve essere aggiornato da una fonte esterna prima della scadenza meno il periodo di tolleranza.

Fornisci una regione a kvssink

Di seguito è riportato l'ordine di ricerca della regione:

1. `AWS_DEFAULT_REGION` La variabile di ambiente viene esaminata per prima. Se è impostata, tale regione viene utilizzata per configurare il client.
2. `aws-region` Il parametro viene esaminato successivamente. Se è impostato, tale regione viene utilizzata per configurare il client.
3. Se non è stato utilizzato nessuno dei metodi precedenti, il kvssink valore predefinito è. `us-west-2`

parametri opzionali kvssink

L'elemento kvssink ha i seguenti parametri opzionali. Per ulteriori informazioni su questi parametri, consultare [Strutture dei flussi video Kinesis](#).

Parametro	Description	Unità/Tipo	Predefinita
stream-name	Il nome dello streaming video Amazon Kinesis di destinazione. ⚠ Important Se non viene specificato alcun nome di stream, verrà utilizzat		

Parametro	Description	Unità/Tipo	Predefinita
	o il nome dello stream predefinito: «DEFAULT_STREAM». Se uno stream con quel nome predefinito non esiste già, verrà creato.		
absolute-fragment-times	Per scegliere se usare i tempi assoluti del frammento.	Booleano	true
access-key	La chiave di AWS accesso utilizzata per accedere a Kinesis Video Streams. È necessario avere AWS le credenziali impostate o fornire questo parametro . Per fornire queste informazioni, digitare quanto segue: <pre>export AWS_ACCESS_KEY_ID=</pre>		
avg-bandwidth-bps	La larghezza di banda media prevista del flusso.	Bit al secondo	4194304

Parametro	Description	Unità/Tipo	Predefinita
aws-region	I Regione AWS da usare.  Note Puoi anche fornire alla regione la variabile di AWS_DEFAULT_REGION ambiente. Le variabili di ambiente hanno la precedenza se sono impostati sia la variabile di ambiente che i parametri kvssink.  Important La regione verrà impostata per impostazione predefinita us-west-2 se non	Stringa	"us-west-2"

Parametro	Description	Unità/Tipo	Predefinita
	diversamente specificato.		
buffer-duration	Durata del buffer del flusso.	Secondi	120
codec-id	L'ID del codec del flusso.	Stringa	"V_MPEG4/ISO/AVC"
connection-staleness	Il tempo dopo il quale viene chiamato il callback dello stream staleness.	Secondi	60
content-type	Il tipo di contenuto del flusso.	Stringa	"video/h264"
fragment-acks	Per scegliere se utilizzare il frammento ACK.	Booleano	true
fragment-duration	La durata del frammento desiderato.	Millisecondi	2000
framerate	La frequenza fotogrammi prevista.	Fotogrammi al secondo	25
frame-timecodes	Per scegliere se utilizzare i timecode del fotogramma o generare timestamp tramite il callback dell'ora corrente.	Booleano	true

Parametro	Description	Unità/Tipo	Predefinita
key-frame-fragmentation	Per scegliere se creare frammenti su un fotogramma chiave.	Booleano	true
log-config	Percorso di configurazione del log.	Stringa	"../kvs_log_configuration"
max-latency	La latenza massima del flusso.	Secondi	60
recalculate-metrics	Per scegliere se calcolare nuovamente i parametri.	Booleano	true
replay-duration	La durata per spostare all'indietro il lettore corrente per la riproduzione quando si verifica un errore se il riavvio è abilitato.	Secondi	40
restart-on-error	Per scegliere se riavviare quando si verifica un errore.	Booleano	true
retention-period	Il periodo di tempo in cui il flusso viene conservato.	Ore	2
rotation-period	Il periodo di rotazione delle chiavi. Per ulteriori informazioni, vedere AWS KMS Rotating Keys .	Secondi	3600

Parametro	Description	Unità/Tipo	Predefinita
secret-key	La chiave AWS segreta utilizzata per accedere a Kinesis Video Streams. È necessario avere AWS le credenziali impostate o fornire questo parametro. <pre>export AWS_SECRET_ACCESS_KEY=</pre>		
session-token	Specifica il valore del token di sessione richiesto se si utilizzano credenziali di sicurezza temporanee e direttamente dalle operazioni. AWS STS		
storage-size	La dimensione di archiviazione del dispositivo in mebibyte (MiB). Per informazioni sulla configurazione dello storage del dispositivo, consulta StorageInfo.	Mebibyte (MiB)	128

Parametro	Description	Unità/Tipo	Predefinita
streaming-type	Tipo di streaming. I valori validi includono: • 0: tempo reale • 1: quasi in tempo reale (non attualmente supportato) • 2: non in linea	Enum GstKvsSinkStreamingType	0: tempo reale
timecode-scale	La dimensione del timecode MKV.	Millisecondi	1
track-name	Il nome della traccia MKV.	Stringa	"kinesis_video"

Parametro	Description	Unità/Tipo	Predefinita
iot-certificate	AWS IoT credenziali da utilizzare nell'kvssink elemento. iot-certificate accetta le seguenti chiavi e valori:  Note iot-thing-name È facoltativo. Se non iot-thing-name viene fornito, viene utilizzato il valore del stream-name parametro. • endpoint=iotcredentialsproviderendpoint • cert-path=/localdirectorypath/to/certificate • key-path=/localdir	Stringa	Nessuno

Parametro	Description	Unità/Tipo	Predefinita
	ectorypath / to/private/ key • ca-path=/ localdir ectorypath/ to/ca-cert • role-alia ses =role-alia ses • iot-thing- name=YourIotTh ingName		

Esempio: invio di dati a Kinesis Video Streams tramite l'API PutMedia

Questo esempio dimostra come utilizzare l'[PutMedia](#) API. Mostra come inviare dati già in formato contenitore (MKV). Se i tuoi dati devono essere assemblati in un formato contenitore prima dell'invio (ad esempio, se stai assemblando i dati video della telecamera in frame), consulta. [Caricamento su Kinesis Video Streams](#)

Note

L'PutMedia operazione è disponibile solo negli SDK C++ e Java. Ciò è dovuto alla gestione completa delle connessioni, del flusso di dati e delle conferme. Non è supportato in altre lingue.

Questo esempio include le fasi seguenti:

- [Scarica e configura il codice](#)
- [Scrivi ed esamina il codice](#)
- [Esegui e verifica il codice](#)

Scarica e configura il codice

Segui i passaggi per scaricare il codice di esempio Java, importare il progetto nel tuo IDE Java, configurare le posizioni delle librerie e configurare il codice per utilizzare AWS le tue credenziali.

1. Create una directory e clonate il codice sorgente di esempio dal GitHub repository. L'esempio PutMedia è parte della [Java](#).

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

2. Apri l'IDE Java che stai utilizzando (ad esempio, [Eclipse](#) o [IntelliJ IDEA](#)) e importa il progetto Apache Maven che hai scaricato:
 - In Eclipse: scegliere File, Import (Importa), Maven, Existing Maven Projects (Progetti Maven esistenti) e andare alla radice del pacchetto scaricato. Selezionare il file pom.xml.
 - In IntelliJ Idea: scegliere Import (Importa). Andare al file pom.xml nella radice del pacchetto scaricato.

Per ulteriori informazioni, consulta la documentazione IDE correlata.

3. Aggiornare il progetto per fare in modo che l'IDE possa individuare le librerie importate.
 - Per IntelliJ IDEA, procedere come segue:
 - a. Facendo clic con il pulsante destro del mouse, aprire il menu contestuale della directory lib del progetto e scegliere Add as library (Aggiungi come libreria).
 - b. Scegli File, quindi scegli Struttura del progetto.
 - c. In Project Settings (Impostazioni progetto), scegliere Modules (Moduli).
 - d. Nella scheda Sources (Origini), impostare Language Level (Livello linguaggio) su **7** o superiore.
 - Per Eclipse, procedere come segue:
 - a. Facendo clic con il pulsante destro del mouse, aprire il menu contestuale del progetto e scegliere Properties (Proprietà), Java Build Path (Percorso di compilazione Java), Source (Origine). Quindi, esegui queste operazioni:
 1. Nella scheda Source (Origine), fare doppio clic su Native library location (Percorso libreria nativa).

2. Nella procedura guidata Native Library Folder Configuration (Configurazione cartella libreria nativa), scegliere Workspace (Area di lavoro).
 3. Nel menu di selezione Native Library Folder (Cartella libreria nativa), scegliere la directory `lib` nel progetto.
- b. Facendo clic con il pulsante destro del mouse, aprire il menu contestuale del progetto e scegliere Properties (Proprietà). Quindi, esegui queste operazioni:
1. Nella scheda Libraries (Librerie), scegliere Add Jars (Aggiungi Jar).
 2. Nella procedura guidata JAR selection (Selezione JAR), scegliere tutti i file `.jar` nella directory `lib` del progetto.

Scrivi ed esamina il codice

L'esempio dell'API PutMedia (PutMediaDemo) mostra i seguenti modelli di codifica:

Argomenti

- [Crea il PutMediaClient](#)
- [Trasmetti contenuti multimediali e metti in pausa il thread](#)

Gli esempi di codice riportati in questa sezione sono presi dalla classe PutMediaDemo.

Crea il PutMediaClient

La creazione PutMediaClient dell'oggetto richiede i seguenti parametri:

- L'URI dell'endpoint PutMedia.
- Un InputStream diretto al file MKV di cui effettuare lo streaming.
- Il nome del flusso. In questo esempio viene utilizzato il flusso creato in [Usa la libreria del produttore Java](#) (`my-stream`). Per utilizzare un flusso diverso, modificare il parametro seguente:

```
private static final String STREAM_NAME="my-stream";
```

 Note

L'esempio di PutMedia API non crea uno stream. È necessario creare uno stream utilizzando l'applicazione di test per [Usa la libreria del produttore Java](#) la console Kinesis Video Streams o. AWS CLI

- Il timestamp corrente.
- Il tipo di timecode. L'esempio utilizza il codice RELATIVE, che indica che il timestamp è relativo all'inizio del container.
- Un oggetto AWSKinesisVideoV4Signer che verifica che i pacchetti ricevuti siano stati inviati da un mittente autorizzato.
- La larghezza di banda di upstream massima in Kbps.
- Un oggetto AckConsumer per la ricezione delle conferme di ricezione dei pacchetti.

Il codice seguente crea l'oggetto PutMediaClient:

```
/* 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();
```

Trasmetti contenuti multimediali e metti in pausa il thread

In seguito alla creazione del client, il campione avvia lo streaming asincrono con `putMediaInBackground`. Il thread principale viene quindi messo in pausa tramite `latch.await` fino al ritorno di `AckConsumer` e a questo punto il client viene chiuso.

```
/* start streaming video in a background thread */
    client.putMediaInBackground();

    /* wait for request/response to complete */
    latch.await();

    /* close the client */
    client.close();
```

Esegui e verifica il codice

Per eseguire l'esempio dell'API `PutMedia`, procedere come segue:

1. Crea uno stream denominato `my-stream` nella console Kinesis Video Streams o utilizzando AWS CLI
2. Per modificare la directory di lavoro nella directory SDK del producer Java:

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Per compilare l'SDK Java e l'applicazione di demo:

```
mvn package
```

4. Creare un filename temporaneo nella directory `/tmp`:

```
jar_files=$(mktemp)
```

5. Creare una stringa di classpath di dipendenze dall'archivio locale a un file:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. Impostare il valore della variabile di ambiente LD_LIBRARY_PATH come segue:

```
export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-  
video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:  
$LD_LIBRARY_PATH  
$ classpath_values=$(cat $jar_files)
```

7. Eseguite la demo dalla riga di comando nel modo seguente, fornendo le vostre AWS credenziali:

```
java -classpath target/kinesisvideo-java-demo-1.0-SNAPSHOT.jar:$classpath_values -  
Daws.accessKeyId=${ACCESS_KEY} -Daws.secretKey=${SECRET_KEY} -Djava.library.path=/  
opt/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build  
com.amazonaws.kinesisvideo.demoapp.DemoAppMain
```

8. Apri la console <https://console.aws.amazon.com/kinesisvideo/home/> Kinesis Video Streams e scegli il tuo stream nella pagina Manage Streams. Viene avviata la riproduzione del video nel riquadro Video Preview (Anteprima video).

Esempio: streaming da una fonte RTSP

[C++](#) Contiene una definizione per un [contenitore](#) Docker che si connette a una telecamera di rete Real-Time Streaming Protocol (RTSP). L'uso di Docker standardizza l'ambiente operativo per Kinesis Video Streams, semplificando la creazione e l'utilizzo dell'applicazione.

La procedura seguente illustra come configurare e utilizzare l'applicazione demo di RTSP.

Argomenti

- [Tutorial video](#)
- [Prerequisiti](#)
- [Create l'immagine Docker](#)
- [Esegui l'applicazione di esempio RTSP](#)

Tutorial video

Questo video mostra come configurare un Raspberry Pi per inviare feed RTSP al cloud e ad Amazon Kinesis Video Streams. AWS Questa è una dimostrazione end-to-end.

Questo video dimostra come acquisire immagini da un feed per utilizzare la visione artificiale e Amazon Rekognition per elaborare le immagini e inviare avvisi.

Prerequisiti

Per eseguire l'applicazione di esempio Kinesis Video Streams RTSP, è necessario disporre di quanto segue:

- Docker: per informazioni sull'installazione e l'utilizzo di Docker, consulta i seguenti collegamenti:
 - [Istruzioni di download di Docker](#)
 - [Nozioni di base su Docker](#)
- Origine videocamera di rete RTSP: per informazioni sulle videocamere consigliate consulta [Requisiti di sistema](#).

Create l'immagine Docker

Innanzitutto, crea l'immagine Docker all'interno della quale verrà eseguita l'applicazione demo.

1. Clona il repository dimostrativo di Amazon Kinesis Video Streams.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Passa alla directory contenente il Dockerfile. In questo caso, è la directory <https://github.com/aws-samples/amazon-kinesis-video-streams-demos/blob/master/producer-cpp/docker-rtsp/docker-rtsp>.

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. Usa il seguente comando per creare l'immagine Docker. Questo comando crea l'immagine e la contrassegna come rtspdockertest.

```
docker build -t rtspdockertest .
```

4. Esegui `docker images` e cerca l'ID dell'immagine contrassegnato con `rtspdockertest`

Ad esempio, nell'output di esempio riportato di seguito, `IMAGE ID` è `54f0d65f69b2`.

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
	BLOB SIZE				
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB
	292.4 MiB				

Ne avrai bisogno in un passaggio successivo.

Esegui l'applicazione di esempio RTSP

È possibile eseguire l'applicazione di esempio RTSP dall'interno o dall'esterno del contenitore Docker. Segui le istruzioni appropriate riportate di seguito.

Argomenti

- [All'interno del contenitore Docker](#)
- [All'esterno del contenitore Docker](#)

All'interno del contenitore Docker

Esegui l'applicazione di esempio RTSP

1. Avvia il contenitore Docker di Amazon Kinesis Video Streams utilizzando il seguente comando:

```
docker run -it YourImageId /bin/bash
```

2. Per avviare l'applicazione di esempio, fornisci AWS le tue credenziali, il nome del flusso video Amazon Kinesis e l'URL della telecamera di rete RTSP.

Important

Se utilizzi credenziali temporanee, dovrai fornire anche le tue. `AWS_SESSION_TOKEN`
Vedi il secondo esempio qui sotto.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
```

```
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Credenziali temporanee:

```
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. Accedi Console di gestione AWS e apri la console [Kinesis Video Streams](#).

Visualizza lo stream.

4. Per uscire dal contenitore Docker, chiudi la finestra del terminale o digita `exit`.

All'esterno del contenitore Docker

Dall'esterno del contenitore Docker, usa il seguente comando:

```
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"
```

Esempio: analisi e rendering dei frammenti di Kinesis Video Streams

[Trasmetti in streaming utilizzando la libreria parser](#) Contiene un'applicazione demo denominata `KinesisVideoRendererExample` che dimostra l'analisi e il rendering di frammenti di streaming video di Amazon Kinesis. L'esempio utilizza [jCodec](#) per decodificare i frame H.264 codificati che vengono inseriti utilizzando l'applicazione. [Esempio: plug-in GStreamer SDK del produttore di Kinesis Video Streams - kvssink](#) In seguito alla decodifica del fotogramma tramite JCodec, viene effettuato il rendering dell'immagine visibile mediante [JFrame](#).

In questo esempio viene mostrato come effettuare le operazioni seguenti:

- Recupera i frame da un flusso video Kinesis utilizzando l'API ed esegui il rendering dello stream per la visualizzazione `GetMedia`.
- Visualizza il contenuto video degli stream in un'applicazione personalizzata anziché utilizzare la console Kinesis Video Streams.

È inoltre possibile utilizzare le classi in questo esempio per visualizzare contenuti in streaming video Kinesis non codificati come H.264, ad esempio un flusso di file JPEG che non richiede la decodifica prima di essere visualizzati.

La procedura seguente illustra come configurare e utilizzare l'applicazione demo di `Renderer`.

Prerequisiti

Per esaminare e utilizzare la libreria di esempio di `Renderer`, occorre quanto segue:

- Un account Amazon Web Services (AWS). Se non disponi già di un AWS account, consulta [Guida introduttiva a Kinesis Video Streams](#).
- Un ambiente di sviluppo integrato (IDE) Java, come [Eclipse Java Neon](#) o IntelliJ Idea. [JetBrains](#)

Esempio di esecuzione del `renderer`

1. Create una directory, quindi clonate il codice sorgente di esempio dal repository. GitHub

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Aprire l'IDE Java in uso (ad esempio, [Eclipse](#) o [IntelliJ IDEA](#)) e importare il progetto Apache Maven scaricato:
 - In Eclipse: scegliere File, Import (Importa), Maven, Existing Maven Projects (Progetti Maven esistenti). Passa alla directory `kinesis-video-streams-parser-lib`.
 - In IntelliJ Idea: scegliere Import (Importa). Andare al file `pom.xml` nella radice del pacchetto scaricato.

Note

Se IntelliJ non è in grado di trovare le dipendenze, potrebbe essere necessario eseguire quanto segue:

- **Build clean:** scegliere File (File), Settings (Impostazioni), Build, Execution, Deployment (Creazione, esecuzione, distribuzione), Compiler (Compilatore). Verificate che sia selezionata l'opzione **Cancella directory di output** alla ricostruzione, quindi scegliete **Build**, **Build Project**.
- **Reimport the project** (Reimporta il progetto): aprire il menu di scelta rapida (fare clic con il pulsante destro del mouse) del progetto e scegliere **Maven**, **Reimport** (Reimporta).

Per ulteriori informazioni, consulta la documentazione IDE correlata.

3. Dall'IDE Java, aprire `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest`.
4. Rimuovere la direttiva `@Ignore` dal file.
5. Aggiorna il `.stream` parametro con il nome del tuo streaming video Kinesis.
6. Eseguire il test `KinesisVideoRendererExample`.

Come funziona

L'applicazione di esempio dimostra quanto segue:

- [Invio dei dati MKV](#)
- [Analisi dei frammenti MKV in frame](#)
- [Decodifica e visualizzazione della cornice](#)

Invio dei dati MKV

L'esempio invia i dati MKV di esempio dal file `rendering_example_video.mkv`, tramite `PutMedia`, per inviare i dati video a un flusso denominato `render-example-stream`.

L'applicazione crea un `PutMediaWorker`:

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
```

```
executorService.submit(putMediaWorker);
```

Per informazioni sulle classi `PutMediaWorker`, consulta [Chiama PutMedia](#) nella documentazione della [Trasmetti in streaming utilizzando la libreria parser](#).

Analisi dei frammenti MKV in frame

L'esempio quindi recupera e analizza i frammenti MKV dal flusso tramite un `GetMediaWorker`:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getFrameVisitor());
executorService.submit(getMediaWorker);
```

Per ulteriori informazioni sulle classi `GetMediaWorker`, consulta [Chiama GetMedia](#) nella documentazione della [Trasmetti in streaming utilizzando la libreria parser](#).

Decodifica e visualizzazione della cornice

L'esempio quindi decodifica e visualizza il fotogramma tramite [JFrame](#).

Il seguente esempio di codice è tratto dalla classe `KinesisVideoFrameViewer`, che estende `JFrame`:

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

L'immagine viene visualizzata come istanza di [java.awt.image. BufferedImage](#). Per esempi che mostrano come lavorare con `BufferedImage`, vedi [Reading/Loading un'immagine](#).

Sicurezza in Amazon Kinesis Video Streams

La sicurezza del cloud AWS è la massima priorità. Come AWS cliente, trarrai vantaggio da un data center e da un'architettura di rete creati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS voi. Il [modello di responsabilità condivisa](#) descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in sicurezza. L'efficacia della nostra sicurezza è regolarmente testata e verificata da revisori di terze parti come parte dei [programmi di conformitàAWS](#). Per saperne di più sui programmi di conformità che si applicano a Kinesis Video Streams, consulta [AWS Services in Scope by Compliance Program](#).
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e leggi e normative applicabili.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza Kinesis Video Streams. I seguenti argomenti mostrano come configurare Kinesis Video Streams per soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerete anche a utilizzare altri AWS servizi che possono aiutarvi a monitorare e proteggere le vostre risorse Kinesis Video Streams.

Argomenti

- [Protezione dei dati in Kinesis Video Streams](#)
- [Controllo dell'accesso alle risorse di Kinesis Video Streams tramite IAM](#)
- [Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT](#)
- [Convalida della conformità per Amazon Kinesis Video Streams](#)
- [Resilienza nei flussi video di Amazon Kinesis](#)
- [Sicurezza dell'infrastruttura in Kinesis Video Streams](#)
- [Le migliori pratiche di sicurezza per Kinesis Video Streams](#)

Protezione dei dati in Kinesis Video Streams

Puoi utilizzare la crittografia lato server (SSE) utilizzando AWS Key Management Service le chiavi () per soddisfare severi requisiti di gestione dei dati crittografando i dati inattivi in Amazon Kinesis Video Streams.AWS KMS

Argomenti

- [Cos'è la crittografia lato server per Kinesis Video Streams?](#)
- [Costi, regioni e considerazioni sulle prestazioni](#)
- [Come posso iniziare a usare la crittografia lato server?](#)
- [Creazione e utilizzo di una chiave gestita dal cliente](#)
- [Autorizzazioni per l'utilizzo di una chiave gestita dal cliente](#)

Cos'è la crittografia lato server per Kinesis Video Streams?

Server-side la crittografia è una funzionalità di Kinesis Video Streams che crittografa automaticamente i dati prima che vengano archiviati a riposo utilizzando una chiave specificata dall'utente. AWS KMS I dati vengono crittografati prima della scrittura nel livello di archiviazione di Kinesis Video Streams e vengono decrittografati dopo essere stati recuperati dall'archiviazione. Di conseguenza, i dati vengono sempre crittografati a riposo all'interno del servizio Kinesis Video Streams.

Grazie alla crittografia lato server, i produttori e i consumatori di streaming video Kinesis non devono gestire le chiavi KMS o le operazioni crittografiche. Se la conservazione dei dati è abilitata, i dati vengono crittografati automaticamente quando entrano ed escono da Kinesis Video Streams, quindi i dati inattivi sono crittografati. AWS KMS fornisce tutte le chiavi utilizzate dalla funzionalità di crittografia lato server. AWS KMS semplifica l'uso di una chiave KMS per Kinesis Video Streams gestita da AWS, una chiave specificata dall'utente AWS KMS importata nel servizio. AWS KMS

Costi, regioni e considerazioni sulle prestazioni

Quando applichi la crittografia lato server, sei soggetto all'utilizzo delle AWS KMS API e ai costi delle chiavi. A differenza delle AWS KMS chiavi personalizzate, la chiave `aws/kinesisvideo` KMS predefinita viene offerta gratuitamente. Tuttavia, devi comunque pagare i costi di utilizzo delle API sostenuti da Kinesis Video Streams per tuo conto.

I costi di utilizzo delle API si applicano a ogni chiave KMS, comprese quelle personalizzate. I AWS KMS costi variano in base al numero di credenziali utente utilizzate sui produttori e sui consumatori di dati, poiché ogni credenziale utente richiede una chiamata API univoca a. AWS KMS

Di seguito vengono descritti i costi per risorsa:

Chiavi

- La chiave KMS per Kinesis Video Streams gestita da AWS (alias =) è gratuita. `aws/kinesisvideo`
- User-generated Le chiavi KMS sono soggette a costi. AWS KMS key Per ulteriori informazioni, consultare [AWS Key Management Service Prezzi](#).

AWS KMS Utilizzo delle API

Le richieste API per generare nuove chiavi di crittografia dei dati o per recuperare le chiavi di crittografia esistenti aumentano con l'aumentare del traffico e sono soggette a costi di AWS KMS utilizzo. Per ulteriori informazioni, consulta [AWS Key Management Service Prezzi: Utilizzo](#).

Kinesis Video Streams genera richieste chiave anche quando la conservazione è impostata su 0 (nessuna conservazione).

Disponibilità della crittografia lato server per regione

Server-side la crittografia dei flussi video Kinesis è disponibile in tutti i paesi in Regioni AWS cui Kinesis Video Streams è disponibile.

Come posso iniziare a usare la crittografia lato server?

Server-side la crittografia è sempre abilitata su Kinesis Video Streams. Se non viene specificata una chiave fornita dall'utente al momento della creazione dello stream, viene utilizzata la Chiave gestita da AWS (fornita da Kinesis Video Streams).

Una chiave KMS fornita dall'utente deve essere assegnata a un flusso video Kinesis al momento della creazione. Non è possibile assegnare una chiave diversa a uno stream utilizzando l'[UpdateStream](#) API in un secondo momento.

È possibile assegnare una chiave KMS fornita dall'utente a uno streaming video Kinesis in due modi:

- Quando si crea un flusso video Kinesis in Console di gestione AWS, specificate la chiave KMS nella scheda Crittografia della pagina Crea un nuovo flusso video.

- Quando create un flusso video Kinesis utilizzando l'[CreateStream](#) API, specificate l'ID della chiave nel parametro. `KmsKeyId`

Creazione e utilizzo di una chiave gestita dal cliente

Questa sezione descrive come creare e utilizzare le proprie chiavi KMS anziché utilizzare la chiave amministrata da Amazon Kinesis Video Streams.

Creazione di una chiave gestita dal cliente

Per informazioni su come creare chiavi personalizzate, consulta [Creating Keys](#) nella AWS Key Management Service Developer Guide. Dopo aver creato le chiavi per l'account, il servizio Kinesis Video Streams le restituisce nell'elenco delle chiavi gestite dal cliente.

Utilizzo di una chiave gestita dal cliente

Dopo aver applicato le autorizzazioni corrette a consumatori, produttori e amministratori, puoi utilizzare chiavi KMS personalizzate, una tua Account AWS o un'altra. Account AWS Tutte le chiavi KMS del tuo account vengono visualizzate nell'elenco delle chiavi gestite dal cliente sulla console.

Per utilizzare chiavi KMS personalizzate che si trovano in un altro account, è necessario disporre delle autorizzazioni per utilizzare tali chiavi. Inoltre, occorre creare il flusso tramite l'API `CreateStream`. Non puoi utilizzare chiavi KMS di account diversi negli stream creati nella console.

Note

Non si accede alla chiave KMS finché non viene GetMedia eseguita l'operazione PutMedia or. Tali operazioni producono i risultati seguenti:

- Se la chiave specificata non esiste, l'CreateStream operazione ha esito positivo, ma PutMedia GetMedia le operazioni sullo stream hanno esito negativo.
- Se utilizzi la chiave fornita (aws/kinesisvideo), la chiave non è presente nel tuo account fino a quando non viene eseguita la prima PutMedia GetMedia operazione.

Autorizzazioni per l'utilizzo di una chiave gestita dal cliente

Prima di poter utilizzare la crittografia lato server con una chiave gestita dal cliente, è necessario configurare le politiche delle chiavi KMS per consentire la crittografia degli stream e la crittografia

e la decrittografia dei record di streaming. Per esempi e ulteriori informazioni sulle AWS KMS autorizzazioni, consulta Autorizzazioni [AWS KMS API: riferimento alle azioni e alle risorse](#).

Note

L'uso della chiave di servizio predefinita per la crittografia non richiede l'applicazione di autorizzazioni IAM personalizzate.

Prima di utilizzare una chiave gestita dal cliente, verificate che i produttori e i consumatori dello streaming video Kinesis (principali IAM) siano utenti nella policy delle chiavi AWS KMS predefinita. In caso contrario, le operazioni di lettura e scrittura dai flussi non riusciranno, causando perdite di dati, ritardi delle elaborazioni o blocchi delle applicazioni. Puoi gestire le autorizzazioni relative alle chiavi KMS tramite le policy IAM. Per ulteriori informazioni, consulta [Using IAM Policies with. AWS KMS](#)

Esempi di autorizzazioni di produttore

I produttori di streaming video Kinesis devono avere l'`kms:GenerateDataKey` autorizzazione:

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/*"
    }
  ]
}
```

```
}
```

Esempio di autorizzazioni per i consumatori

I tuoi utenti dello streaming video Kinesis devono avere l'`kms:Decrypt` autorizzazione:

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/*"
    }
  ]
}
```

Controllo dell'accesso alle risorse di Kinesis Video Streams tramite IAM

Puoi utilizzare AWS Identity and Access Management (IAM) con Amazon Kinesis Video Streams per controllare se gli utenti della tua organizzazione possono eseguire un'attività utilizzando specifiche operazioni API Kinesis Video Streams e se possono utilizzare risorse specifiche. AWS

Per ulteriori informazioni su IAM, consulta:

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

- [Guida introduttiva a IAM](#)
- [Guida per l'utente di IAM](#)

Indice

- [Sintassi delle policy](#)
- [Azioni per Kinesis Video Streams](#)
- [Amazon Resource Names \(ARN\) per Kinesis Video Streams](#)
- [Concedere ad altri account IAM l'accesso a uno streaming video Kinesis](#)
- [Esempi di politiche per Kinesis Video Streams](#)

Sintassi delle policy

Una policy IAM è un documento JSON costituito da una o più dichiarazioni. Ogni dichiarazione è strutturata come segue:

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

Una dichiarazione è costituita da diversi elementi:

- **Effetto:** l'effetto può essere Allow o Deny. Per impostazione predefinita, gli utenti non dispongono dell'autorizzazione per l'utilizzo di risorse e operazioni API, pertanto tutte le richieste vengono rifiutate. Un permesso esplicito sostituisce l'impostazione predefinita. Un rifiuto esplicito sovrascrive tutti i consensi.
- **Azione:** l'azione è l'azione API specifica per la quale si concede o si nega l'autorizzazione.

- **Risorsa:** la risorsa interessata dall'azione. Per specificare una risorsa nella dichiarazione, devi utilizzare il suo ARN (Amazon Resource Name).
- **Condizione:** le condizioni sono opzionali. Possono essere utilizzate per controllare quando è in vigore una policy.

Quando crei e gestisci le policy IAM, ti consigliamo di utilizzare [IAM Policy Generator](#) e [IAM Policy Simulator](#).

Azioni per Kinesis Video Streams

In una dichiarazione di policy IAM, è possibile specificare qualsiasi operazione API per qualsiasi servizio che supporta IAM. Per Kinesis Video Streams, usa il seguente prefisso con il nome dell'azione API: `kinesisvideo:`. For example: `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams` e `kinesisvideo:DescribeStream`.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

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

Puoi anche specificare più operazioni tramite caratteri jolly. Ad esempio, puoi specificare tutte le operazioni il cui nome inizia con la parola "Get" come segue:

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

Per specificare tutte le operazioni di Kinesis Video Streams, utilizzate il carattere jolly asterisco (*) come segue:

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

Per l'elenco completo delle azioni dell'API Kinesis Video Streams, consultate il riferimento all'API Kinesis Video Streams.

Amazon Resource Names (ARN) per Kinesis Video Streams

Ogni dichiarazione di policy IAM si applica alle risorse specificate utilizzando i relativi ARN.

Utilizza il seguente formato di risorse ARN per Kinesis Video Streams:

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

Ad esempio:

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

Puoi ottenere l'ARN di uno stream usando. [DescribeStream](#)

Concedere ad altri account IAM l'accesso a uno streaming video Kinesis

Potrebbe essere necessario concedere l'autorizzazione ad altri account IAM per eseguire operazioni sugli stream in Kinesis Video Streams. La seguente panoramica descrive i passi per concedere l'accesso ai flussi video su tutti gli account:

1. Ottieni l'ID di 12 cifre dell'account a cui desideri concedere le autorizzazioni per eseguire operazioni sulla risorsa di streaming creata nel tuo account.

Esempio: nei passaggi seguenti, utilizzeremo 1111 come ID dell'account a cui desideri concedere l'autorizzazione e 9999 come ID per i tuoi Kinesis Video Streams

2. Crea una policy gestita da IAM nell'account proprietario dello stream (6269999) che consenta il livello di accesso che desideri concedere.

Esempio di politica:

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"
    }
  ]
}
```

```
}
```

Per altri esempi di policy per le risorse Kinesis Video Streams, [Policy di esempio](#) consultate la sezione successiva.

3. Crea un ruolo nell'account proprietario dello stream (9999) e specifica l'account per il quale desideri concedere le autorizzazioni (1111). Ciò aggiungerà un'entità attendibile al ruolo.

Esempio di politica affidabile:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Allega a questo ruolo la policy che hai creato nel passaggio precedente.

Ora hai creato un ruolo nell'account 9999 che dispone dell'autorizzazione per operazioni come `DescribeStreamGetDataEndpoint`, e `PutMedia` su un ARN di risorse di streaming nella policy gestita. Questo nuovo ruolo prevede che anche l'altro account, 1111, assuma questo ruolo.

 Important

Prendi nota del ruolo ARN, ti servirà nel passaggio successivo.

4. Crea una policy gestita nell'altro account, 1111, che consenta l'`AssumeRole`azione sul ruolo che hai creato nell'account 9999 nel passaggio precedente. Dovrai menzionare il ruolo ARN del passaggio precedente.

Esempio di politica:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
    }
  ]
}
```

5. Allega la policy creata nel passaggio precedente a un'entità IAM, ad esempio un ruolo o un utente nell'account 1111. Questo utente ha ora il permesso di assumere il ruolo CustomRoleName nell'account 9999.

Le credenziali di questo utente chiamano l' AWS STS AssumeRoleAPI per ottenere le credenziali di sessione, che vengono successivamente utilizzate per chiamare le API di Kinesis Video Streams sullo stream creato nell'account 9999.

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/CustomRoleName" --
role-session-name "kvs-cross-account-assume-role"
{
  "Credentials": {
    "AccessKeyId": "",
    "SecretAccessKey": "",
    "SessionToken": "",
    "Expiration": ""
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "",
    "Arn": ""
  }
}
```

6. Imposta la chiave di accesso, la chiave segreta e le credenziali di sessione in base al set precedente nell'ambiente.

```
set AWS_ACCESS_KEY_ID=
```

```
set AWS_SECRET_ACCESS_KEY=  
set AWS_SESSION_TOKEN=
```

7. Esegui le API Kinesis Video Streams per descrivere e ottenere l'endpoint di dati per lo streaming nell'account 9999.

```
aws kinesisvideo describe-stream --stream-arn "arn:aws:kinesisvideo:us-  
west-2:999999999999:stream/custom-stream-name/1613732218179"  
{  
  "StreamInfo": {  
    "StreamName": "custom-stream-name",  
    "StreamARN": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-  
stream-name/1613732218179",  
    "KmsKeyId": "arn:aws:kms:us-west-2:999999999999:alias/aws/kinesisvideo",  
    "Version": "abcd",  
    "Status": "ACTIVE",  
    "CreationTime": "2018-02-19T10:56:58.179000+00:00",  
    "DataRetentionInHours": 24  
  }  
}  
  
aws kinesisvideo get-data-endpoint --stream-arn "arn:aws:kinesisvideo:us-  
west-2:999999999999:stream/custom-stream-name/1613732218179" --api-name "PUT_MEDIA"  
{  
  "DataEndpoint": "https://s-b12345.kinesisvideo.us-west-2.amazonaws.com"  
}
```

Per istruzioni dettagliate generiche sulla concessione dell'accesso tra più account, consulta [Delegate Access Across Using IAM Roles](#). [Account AWS](#)

Esempi di politiche per Kinesis Video Streams

Le seguenti politiche di esempio dimostrano come è possibile controllare l'accesso degli utenti ai propri Kinesis Video Streams

Example 1: Consenti agli utenti di ottenere dati da qualsiasi flusso video Kinesis

Questa policy consente a un utente o a un gruppo di eseguire `ListTagsForStream` operazioni `DescribeStream` `GetDataEndpoint` `GetMedia`, `ListStreams`, e su qualsiasi flusso video Kinesis. Questa policy è appropriata per gli utenti che possono ottenere i dati da qualsiasi flusso di video.

JSON

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

Example 2: Consenti a un utente di creare un flusso video Kinesis e di scrivervi dati

Questa policy consente a un utente o a un gruppo di eseguire le operazioni `CreateStream` e `PutMedia`. Questa policy è appropriata per una fotocamera di sicurezza su cui è possibile creare un flusso di video e inviare dati.

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

Example 3: Consenti a un utente l'accesso completo a tutte le risorse di Kinesis Video Streams

Questa policy consente a un utente o a un gruppo di eseguire qualsiasi operazione di Kinesis Video Streams su qualsiasi risorsa. Questa policy è appropriata per gli amministratori.

JSON

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

Example 4: Consenti a un utente di scrivere dati su uno specifico flusso video Kinesis

Questa policy consente a un utente o a un gruppo di scrivere dati su un determinato flusso di video. Questa policy è appropriata per un dispositivo in grado di inviare dati a un solo flusso.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/your_stream/0123456789012"
    }
  ]
}
```

Controllo dell'accesso alle risorse di Kinesis Video Streams utilizzando AWS IoT

Questa sezione descrive come abilitare un dispositivo (ad esempio una videocamera) a inviare dati audio e video solo a un particolare flusso video Kinesis. È possibile farlo utilizzando il fornitore di AWS IoT credenziali e un ruolo AWS Identity and Access Management (IAM).

I dispositivi possono utilizzare X.509 i certificati per connettersi AWS IoT utilizzando i protocolli di autenticazione reciproca TLS. Altri Servizi AWS (ad esempio, Kinesis Video Streams) non supportano l'autenticazione basata su certificati, ma possono essere richiamati utilizzando le AWS credenziali nel formato Signature Version 4. AWS L'algoritmo Signature Version 4 richiede in genere che il chiamante disponga di un ID della chiave di accesso e di una chiave di accesso segreta. AWS IoT dispone di un provider di credenziali che consente di utilizzare il X.509 certificato integrato come identità univoca del dispositivo per autenticare AWS le richieste (ad esempio, le richieste a Kinesis Video Streams). Ciò elimina la necessità di memorizzare un ID della chiave di accesso e una chiave di accesso segreta sul dispositivo.

Il provider delle credenziali autentica un client (in questo caso, un Kinesis Video Streams SDK in esecuzione sulla telecamera a cui si desidera inviare i dati a un flusso video) utilizzando un X.509 certificato ed emette un token di sicurezza temporaneo con privilegi limitati. Puoi utilizzare il token per firmare e autenticare qualsiasi AWS richiesta (in questo caso, una chiamata a Kinesis Video Streams). Per ulteriori informazioni, consulta [Autorizzazione delle chiamate dirette ai servizi. AWS](#)

Questo modo di autenticare le richieste della videocamera su Kinesis Video Streams richiede la creazione e la configurazione di un ruolo IAM e l'aggiunta di politiche IAM appropriate al ruolo in modo che il fornitore di AWS IoT credenziali possa assumere il ruolo per conto dell'utente.

Per ulteriori informazioni, consulta la Documentazione. AWS IoT [AWS IoT Core](#) Per ulteriori informazioni su IAM, consulta [AWS Identity and Access Management \(IAM\)](#).

Argomenti

- [AWS IoT ThingName come nome dello stream](#)
- [AWS IoT CertificateId come nome dello stream](#)
- [Utilizzo AWS IoT credenziali per lo streaming verso un nome di streaming codificato](#)

AWS IoT ThingName come nome dello stream

Argomenti

- [Passaggio 1: creare un AWS IoT tipo di cosa e un AWS IoT thing](#)
- [Fase 2: Creare un ruolo IAM che verrà assunto da AWS IoT](#)
- [Fase 3: Creare e configurare il certificato X.509](#)
- [Fase 4: Testare il AWS IoT credenziali con il tuo streaming video Kinesis](#)
- [Fase 5: distribuzione AWS IoT certificati e credenziali sul file system della videocamera e streaming di dati nel flusso video](#)

Passaggio 1: creare un AWS IoT tipo di cosa e un AWS IoT thing

In AWS IoT, una cosa è una rappresentazione di un dispositivo o di un'entità logica specifica. In questo caso, qualsiasi AWS IoT oggetto rappresenta il flusso video Kinesis e desiderate configurare il controllo degli accessi a livello di risorsa. Per creare un oggetto, è necessario innanzitutto creare un tipo di oggetto. AWS IoT È possibile utilizzare i tipi di AWS IoT oggetto per memorizzare informazioni di descrizione e configurazione comuni a tutti gli oggetti associati allo stesso tipo di oggetto.

1. Il comando di esempio seguente crea un tipo di oggetto `kvs_example_camera`:

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

2. Questo comando di esempio crea l'`kvs_example_camera_stream` oggetto del tipo di `kvs_example_camera` oggetto:

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

Fase 2: Creare un ruolo IAM che verrà assunto da AWS IoT

I ruoli IAM sono simili agli utenti, in quanto un ruolo è un' AWS identità con politiche di autorizzazione che determinano cosa può e non può fare l'identità. AWS Un ruolo può essere assunto da chiunque ne abbia bisogno. Tuttavia, quando assumi un ruolo, vengono fornite le credenziali di sicurezza provvisorie per la sessione del ruolo.

Il ruolo creato in questa fase può essere assunto AWS IoT per ottenere credenziali temporanee dal servizio di token di sicurezza (STS) durante l'esecuzione delle richieste di autorizzazione delle credenziali dal client. In questo caso, il client è l'SDK Kinesis Video Streams in esecuzione sulla videocamera.

Eseguire la procedura seguente per creare e configurare questo ruolo IAM:

1. Crea un ruolo IAM.

Il comando di esempio seguente crea un ruolo IAM denominato `KVSCameraCertificateBasedIAMRole`:

```
aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
--assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
```

Puoi utilizzare il seguente JSON della policy di attendibilità per `iam-policy-document.json`:

JSON

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

2. Quindi, allega una policy di autorizzazioni al ruolo IAM che hai creato in precedenza. Questa politica di autorizzazioni consente il controllo selettivo degli accessi (un sottoinsieme delle operazioni supportate) per una risorsa. AWS In questo caso, la AWS risorsa è il flusso video a cui desideri che la videocamera invii i dati. In altre parole, una volta completate tutte le fasi di configurazione, questa telecamera sarà in grado di inviare dati solo a questo flusso video.

```
aws --profile default iam put-role-policy --role-name
KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
document 'file://iam-permission-document.json'
```

Puoi utilizzare la seguente policy IAM JSON per: `iam-permission-document.json`

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:ThingName}/*"
    }
  ]
}
```

Tieni presente che questa policy autorizza le azioni specificate solo su un flusso video (AWS risorsa) specificato dal segnaposto. (`${credentials-iot:ThingName}`) Questo segnaposto assume il valore dell'attributo AWS IoT thing `ThingName` quando il fornitore delle AWS IoT credenziali invia il nome del flusso video nella richiesta.

3. Quindi, crea un alias di ruolo per il tuo ruolo IAM. L'alias del ruolo è un modello di dati alternativo che punta al ruolo IAM. Una richiesta di fornitore di AWS IoT credenziali deve includere un alias di ruolo per indicare quale ruolo IAM assumere per ottenere le credenziali temporanee dal STS.

Il comando di esempio seguente crea un alias del ruolo denominato `KvsCameraIoTRoleAlias`.

```
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. Ora puoi creare la policy che consentirà di AWS IoT assumere il ruolo con il certificato (una volta collegato) utilizzando l'alias di ruolo.

Il seguente comando di esempio crea una policy per call AWS IoT . KvsCameraIoTPolicy

```
aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-  
document 'file://iot-policy-document.json'
```

È possibile utilizzare il comando seguente per creare il `iot-policy-document.json` documento JSON:

```
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
```

Fase 3: Creare e configurare il certificato X.509

La comunicazione tra un dispositivo (il tuo streaming video) AWS IoT è protetta tramite l'uso di X.509 certificati.

1. Crea il certificato a cui devi allegare la policy AWS IoT che hai creato in precedenza.

```
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. Allega la policy per AWS IoT (KvsCameraIoTPolicy creata in precedenza) a questo certificato.

```
aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target  
$(jq --raw-output '.certificateArn' certificate)
```

3. Allega la tua AWS IoT cosa (kvs_example_camera_stream) al certificato che hai appena creato:

```
aws --profile default iot attach-thing-principal --thing-name  
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'  
certificate)
```

4. Per autorizzare le richieste tramite il fornitore di AWS IoT credenziali, è necessario l'endpoint delle AWS IoT credenziali, che è univoco per il tuo ID. Account AWS. È possibile utilizzare il comando seguente per ottenere l'endpoint delle credenziali. AWS IoT

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider  
--output text > iot-credential-provider.txt
```

5. Oltre al X.509 certificato creato in precedenza, è necessario disporre anche di un certificato CA per stabilire l'attendibilità con il servizio di backend tramite TLS. Puoi ottenere il certificato CA utilizzando il comando seguente:

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output  
cacert.pem
```

Fase 4: Testare il AWS IoT credenziali con il tuo streaming video Kinesis

Ora puoi testare AWS IoT le credenziali che hai impostato finora.

1. In primo luogo, creare un Kinesis Video Stream con cui testare questa configurazione.

 Important

Crea uno streaming video con un nome identico al nome dell' AWS IoT oggetto che hai creato nel passaggio precedente (kvs_example_camera_stream).

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

2. Quindi, chiama il fornitore delle AWS IoT credenziali per ottenere le credenziali temporanee:

```
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

Puoi usare il seguente comando per ottenere: IOT_GET_CREDENTIAL_ENDPOINT

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

Il JSON di output contiene i accessKey valorisecretKey, esessionToken, che potete usare per accedere a Kinesis Video Streams.

3. Per il test, puoi utilizzare queste credenziali per richiamare l'API Kinesis Video Streams per il flusso DescribeStream video di esempio. kvs_example_camera_stream

```
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
```

Fase 5: distribuzione AWS IoT certificati e credenziali sul file system della videocamera e streaming di dati nel flusso video

Note

I passaggi di questa sezione descrivono l'invio di file multimediali a un flusso video Kinesis da una videocamera che utilizza il. [the section called "C++"](#)

1. Copia il X.509 certificato, la chiave privata e il certificato CA generati nei passaggi precedenti nel file system della videocamera. Specificate i percorsi in cui sono archiviati questi file, il nome dell'alias del ruolo e l'endpoint delle AWS IoT credenziali per l'esecuzione del `gst-launch-1.0` comando o dell'applicazione di esempio.
2. Il seguente comando di esempio utilizza l'autorizzazione del AWS IoT certificato per inviare video a Kinesis Video Streams:

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
  rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink stream-
  name="kvs_example_camera_stream" aws-region="YourAWSRegion" iot-certificate="iot-
  certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
  region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
  private.pem,key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias"
```

AWS IoT CertificateId come nome dello stream

Per rappresentare il dispositivo (ad esempio, la videocamera) tramite un AWS IoT oggetto, ma autorizzare un nome di streaming diverso, è possibile utilizzare l' `AWS IoT certificateId` attributo come nome dello stream e fornire le autorizzazioni Kinesis Video Streams sullo stream utilizzando. AWS IoT I passaggi per eseguire questa operazione sono simili a quelli descritti in precedenza, con alcune modifiche.

- Modifica la politica delle autorizzazioni in base al tuo ruolo IAM (`iam-permission-document.json`) come segue:

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

L'ARN risorsa utilizza l'ID certificato come segnaposto per il nome del flusso. L'autorizzazione IAM funzionerà quando utilizzi l'ID del certificato come nome dello stream. Ottieni l'ID del certificato dal certificato in modo da poterlo utilizzare come nome dello stream nella seguente descrizione della chiamata API dello stream.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Verificare questa modifica utilizzando il comando CLI describe-stream di 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}

```

- Passate il certificateID al fornitore delle AWS IoT credenziali nell'applicazione [di](#) esempio nell'SDK C++ di Kinesis Video Streams:

```

credential_provider =
    make_unique<IoTCertCredentialProvider>(iot_get_credential_endpoint,
        cert_path,
        private_key_path,
        role_alias,
        ca_cert_path,

```

```
certificateId);
```

Note

Nota che stai passando il codice al fornitore delle credenziali. thingname AWS IoT Puoi usarlo getenv per passare il thingname all'applicazione demo in modo simile a quando passi gli altri AWS IoT attributi. Utilizzare l'ID certificato come nome del flusso nei parametri della riga di comando quando si esegue l'applicazione di esempio.

Utilizzo AWS IoT credenziali per lo streaming verso un nome di streaming codificato

Per rappresentare il tuo dispositivo (ad esempio, la tua videocamera) attraverso un AWS IoT oggetto, ma autorizzare lo streaming su uno specifico streaming video Amazon Kinesis, fornisci le autorizzazioni Amazon Kinesis Video Streams sullo streaming utilizzando. AWS IoT Il processo è simile alle sezioni precedenti, con alcune modifiche.

Modifica la politica delle autorizzazioni in base al tuo ruolo IAM (iam-permission-document.json) come segue:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
    }
  ]
}
```

Copia il X.509 certificato, la chiave privata e il certificato CA generati nei passaggi precedenti nel file system della videocamera.

Specifica i percorsi in cui sono archiviati questi file, il nome dell'alias del ruolo, il nome dell' AWS IoT oggetto e l'endpoint delle AWS IoT credenziali per l'esecuzione del `gst-launch-1.0` comando o dell'applicazione di esempio.

Il seguente comando di esempio utilizza l'autorizzazione AWS IoT del certificato per inviare video ad Amazon Kinesis Video Streams:

```
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"
```

Convalida della conformità per Amazon Kinesis Video Streams

La nostra nuova esperienza AWS di registrazione non è progettata per carichi di lavoro regolamentati. Se stai utilizzando la nostra nuova esperienza di AWS registrazione, ma desideri utilizzarla AWS per carichi di lavoro regolamentati, puoi [iscriverti a AWS \(avanzato\)](#) o [attivare funzionalità](#) avanzate per il tuo ambiente. AWS

Per sapere se un Servizio AWS rientra nell'ambito di specifici programmi di conformità, consulta [Servizi AWS la sezione Scope by Compliance Program](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, vedere Programmi di [AWS conformità Programmi di](#)

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, vedere [Scaricamento dei report in AWS Artifact](#).

La responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS, consulta [AWS la Documentazione sulla sicurezza](#).

Resilienza nei flussi video di Amazon Kinesis

L'infrastruttura AWS globale è costruita attorno AWS alle regioni e alle zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate a reti a bassa latenza, alta produttività e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure. [AWS](#)

Sicurezza dell'infrastruttura in Kinesis Video Streams

In quanto servizio gestito, Amazon Kinesis Video Streams è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper [Amazon Web Services: Overview of Security Processes](#).

Utilizzi le chiamate API AWS pubblicate per accedere a Kinesis Video Streams attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. I clienti devono inoltre supportare suite di crittografia con Perfect Forward Secrecy (PFS) come Ephemeral (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un responsabile IAM. In alternativa è possibile utilizzare [Servizio di token di sicurezza AWS](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Le migliori pratiche di sicurezza per Kinesis Video Streams

Amazon Kinesis Video Streams offre una serie di funzionalità di sicurezza da considerare durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, sono da considerare come considerazioni utili anziché prescrizioni.

Per le best practice di sicurezza per i tuoi dispositivi remoti, consulta [Best practice per la sicurezza degli agenti dei dispositivi](#).

Implementazione dell'accesso con privilegi minimi

Quando concedi le autorizzazioni, sei tu a decidere chi concede quali autorizzazioni e a quali risorse di Kinesis Video Streams. È possibile abilitare operazioni specifiche che si desidera consentire su tali risorse. Pertanto è necessario concedere solo le autorizzazioni necessarie per eseguire un'attività. L'implementazione dell'accesso con privilegi minimi è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Ad esempio, un produttore che invia dati a Kinesis Video Streams richiede solo, e. PutMedia e GetStreamingEndpoint. Non concedere autorizzazioni delle applicazioni del producer per tutte le operazioni (*) o per altre operazioni, ad esempio GetMedia.

Per ulteriori informazioni, consulta l'argomento che descrive [qual è il privilegio minimo e perché è necessario](#).

Uso di ruoli IAM

Le applicazioni del produttore e del cliente devono disporre di credenziali valide per accedere a Kinesis Video Streams. Non devi archiviare credenziali AWS direttamente in un'applicazione client o in un bucket Amazon S3. Si tratta di credenziali a lungo termine che non vengono ruotate automaticamente e potrebbero avere un impatto aziendale significativo se compromesse.

Dovreste invece utilizzare un ruolo IAM per gestire le credenziali temporanee per consentire alle applicazioni del produttore e del cliente di accedere a Kinesis Video Streams. Quando si utilizza un ruolo, non è necessario utilizzare credenziali a lungo termine (come nome utente e password o chiavi di accesso) per accedere ad altre risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente IAM:

- [Ruoli IAM](#)
- [Scenari comuni per ruoli: utenti, applicazioni e servizi](#)

CloudTrail Usalo per monitorare le chiamate API

Kinesis Video Streams funziona con AWS CloudTrail un servizio che fornisce una registrazione delle azioni intraprese da un utente, da un ruolo o da un utente Servizio AWS in Kinesis Video Streams.

È possibile utilizzare le informazioni raccolte da CloudTrail per determinare la richiesta effettuata a Kinesis Video Streams, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni, consulta [the section called “Registra le chiamate API con CloudTrail”](#).

Risoluzione dei problemi relativi a Kinesis Video Streams

Utilizza le seguenti informazioni per risolvere i problemi più comuni riscontrati con Amazon Kinesis Video Streams.

Argomenti

- [Problemi generali](#)
- [Problemi relativi alle API](#)
- [Problemi HLS](#)
- [Problemi con Java](#)
- [Problemi con la libreria del produttore](#)
- [Problemi con la libreria Stream Parser](#)
- [Problemi di rete](#)

Problemi generali

Questa sezione descrive i problemi generali che potresti riscontrare quando lavori con Kinesis Video Streams.

Problemi

- [Latenza troppo elevata](#)

Latenza troppo elevata

La latenza potrebbe essere causata dalla durata dei frammenti inviati al servizio Kinesis Video Streams. Uno dei modi per ridurre la latenza tra producer e servizio è configurare la pipeline dei file multimediali per ottenere frammenti di più breve durata.

Per ridurre il numero di frame inviati in ogni frammento, riduci il seguente valore in:

kinesis_video_gstreamer_sample_app.cpp

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512,
NULL);
```

 Note

Le latenze sono più elevate nel browser Mozilla Firefox a causa dell'implementazione interna del rendering video.

Problemi relativi alle API

Questa sezione descrive i problemi relativi alle API che potreste incontrare quando lavorate con Kinesis Video Streams.

Problemi

- [Errore: "Opzioni sconosciute"](#)
- [Errore: «Impossibile determinare service/operation il nome da autorizzare»](#)
- [Errore: "Failed to put a frame in the stream" \(Impossibile inserire un fotogramma nel flusso\)](#)
- [Errore: «Il servizio ha chiuso la connessione prima della ricezione della versione definitiva AckEvent »](#)
- [Errore "STATUS_STORE_OUT_OF_MEMORY"](#)
- [Errore: «La credenziale deve essere associata a una regione valida».](#)

Errore: "Opzioni sconosciute"

Se le richieste `GetMedia` e `GetMediaForFragmentList` non riescono, può essere generato questo errore:

```
Unknown options: <filename>.mkv
```

Questo errore si verifica se è stato configurato AWS CLI con un output tipo di `json`. Riconfigurare il AWS CLI con il tipo di output predefinito (`none`). Per informazioni sulla configurazione di AWS CLI, vedere [configure](#) nel AWS CLI Command Reference.

Errore: «Impossibile determinare service/operation il nome da autorizzare»

Se la richiesta `GetMedia` non riesce, può essere generato questo errore:

```
Unable to determine service/operation name to be authorized
```

Questo errore può verificarsi se l'endpoint non è specificato correttamente. Quando ricevi l'endpoint, assicurati di includere il seguente parametro nella `GetDataEndpoint` chiamata, a seconda dell'API da chiamare:

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

Errore: "Failed to put a frame in the stream" (Impossibile inserire un fotogramma nel flusso)

Se la richiesta `PutMedia` non riesce, può essere generato questo errore:

```
Failed to put a frame in the stream
```

Questo errore può verificarsi se la connettività o le autorizzazioni non sono disponibili per il servizio. Esegui quanto segue nel AWS CLI e verifica che le informazioni sullo stream possano essere recuperate:

```
aws kinesisvideo describe-stream --stream-name StreamName --endpoint https://  
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Se la chiamata ha esito negativo, vedi [Risoluzione dei problemi degli AWS CLI errori](#) per ulteriori informazioni.

Errore: «Il servizio ha chiuso la connessione prima della ricezione della versione definitiva AckEvent »

Se la richiesta `PutMedia` non riesce, può essere generato questo errore:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was  
received
```

Questo errore può verificarsi se l'elemento `PushbackInputStream` non è implementato correttamente. Verifica che i `unread()` metodi siano implementati correttamente.

Errore "STATUS_STORE_OUT_OF_MEMORY"

Se la richiesta PutMedia non riesce, può essere generato questo errore:

```
The content store is out of memory.
```

Questo errore si verifica quando allo store dei contenuti non è allocato spazio a sufficienza. Per ovviare, aumentare il valore di `StorageInfo.storageSize`. Per ulteriori informazioni, consulta [StorageInfo](#).

Errore: «La credenziale deve essere associata a una regione valida».

Questo errore si verifica se l'area di firma non corrisponde alla regione dell'endpoint.

Ad esempio, se specifichi `us-west-2` come area di firma, ma provi a connetterti all'endpoint `kinesisvideo.us-east-1.amazonaws.com` (`us-east-1`), riceverai questo errore.

In alcune applicazioni, come [kvssink](#), il valore predefinito della catena di fallback della regione è `us-west-2`. Verifica di aver impostato correttamente la regione in base all'applicazione che stai utilizzando.

Problemi HLS

Se il tuo streaming video non viene riprodotto correttamente, consulta [the section called “Risoluzione dei problemi relativi all'HLS”](#).

Problemi con Java

Questa sezione descrive come risolvere i problemi Java più comuni riscontrati quando si lavora con Kinesis Video Streams.

Problemi

- [Abilitazione dei log di Java](#)

Abilitazione dei log di Java

Per risolvere i problemi relativi agli esempi e alle librerie Java, è utile abilitare ed esaminare i log di debug. Per abilitarli, procedere come segue:

1. Aggiungere log4j al file pom.xml nel nodo dependencies:

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. Nella directory target/classes, creare un file denominato log4j.properties con il seguente contenuto:

```
# 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
```

I log di debug vengono quindi stampati nella console IDE.

Problemi con la libreria del produttore

Questa sezione descrive i problemi che possono verificarsi durante l'utilizzo della [Caricamento su Kinesis Video Streams](#).

Problemi

- [Impossibile compilare l'SDK del produttore](#)
- [Il flusso video non viene visualizzato nella console](#)
- [Errore "Security token included in the request is invalid" durante lo streaming dei dati con l'applicazione demo GStreamer](#)
- [Errore "Failed to submit frame to Kinesis Video client"](#)
- [L'applicazione GStreamer viene arrestata con il messaggio "streaming stopped, reason not-negotiated" su OS X](#)

- [Errore "Failed to allocate heap" durante la creazione del client Kinesis Video nella demo GStreamer su Raspberry Pi](#)
- [Errore "Illegal Instruction" durante l'esecuzione della demo GStreamer su Raspberry Pi](#)
- [La videocamera non viene caricata su Raspberry Pi](#)
- [Impossibile trovare la videocamera su macOS High Sierra](#)
- [File jni.h non trovato durante la compilazione su macOS High Sierra](#)
- [Errori di curl durante l'esecuzione dell'applicazione demo GStreamer](#)
- [Timestamp/range asserzione in fase di esecuzione su Raspberry Pi](#)
- [Asserzione su `gst\_value\_set\_fraction\_range\_full` su Raspberry Pi](#)
- [STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA \(0x3200000d\) error on Android](#)
- [È stata raggiunta la durata massima del frammento, errore](#)
- [Errore "Invalid thing name passed \(Passato nome di cosa non valido\)" quando si utilizza l'autorizzazione IoT](#)

Impossibile compilare l'SDK del produttore

Controllare che le librerie richieste siano presenti nel percorso. Per eseguire una verifica, utilizzare il comando seguente:

```
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
```

Il flusso video non viene visualizzato nella console

Per visualizzare il flusso video nella console, è necessario codificarlo utilizzando H.264 il formato AVCC. Se il flusso non viene visualizzato, verificare quanto segue:

- [Bandiere di adattamento NAL](#) Sei impostato su `NAL_ADAPTATION_ANNEXB_NALS` | `NAL_ADAPTATION_ANNEXB_CPD_NALS` se lo stream originale è in Annex-B formato. Questo è il valore di default nel costruttore `StreamDefinition`.
- I dati privati del codec sono forniti correttamente. Per H.264, si tratta del set di parametri di sequenza (SPS) e del set di parametri dell'immagine (PPS). A seconda dell'origine del contenuto multimediale, questi dati potrebbero essere recuperati dall'origine separatamente o codificati nel fotogramma.

Molti stream elementari sono nel seguente formato, Ab dov'è il codice Annex-B iniziale (001 o 0001):

```
Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)
```

Il CPD (Codec Private Data), se H.264 è presente nello stream come SPS e PPS, può essere adattato al formato AVCC. A meno che la pipeline multimediale non fornisca il CPD separatamente, l'applicazione può estrarre il CPD dal frame cercando il primo frame Idr (che dovrebbe contenere SPS e PPS), estrarre i due NALU (che saranno Ab (Sps) Ab (Pps)) e impostarli nel CPD `inStreamDefinition`.

Errore "Security token included in the request is invalid" durante lo streaming dei dati con l'applicazione demo GStreamer

Questo errore indica che è presente un problema con le credenziali. Verificare quanto segue:

- Se si utilizzano credenziali di sicurezza temporanee, è necessario specificare il token di sessione.
- Verificare che le credenziali temporanee non siano scadute.
- Verificare che siano impostati gli opportuni diritti.
- Su macOS, verificare che non vi siano credenziali memorizzate nella cache in Keychain.

Errore "Failed to submit frame to Kinesis Video client"

Questo errore indica che i timestamp non sono impostati in modo corretto nel flusso sorgente. Esegui quanto segue:

- Utilizzare l'SDK di esempio più recente, che potrebbe includere un aggiornamento per la risoluzione del problema.
- Imposta lo streaming di alta qualità su un bitrate più elevato e correggi eventuali jitter nel flusso sorgente se la fotocamera lo supporta.

L'applicazione GStreamer viene arrestata con il messaggio "streaming stopped, reason not-negotiated" su OS X

Lo streaming potrebbe essere arrestato su OS X con il seguente messaggio:

```
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)
```

Una possibile soluzione è rimuovere i parametri di frequenza dei fotogrammi dalla `gst_caps_new_simple` chiamata 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);
```

Errore "Failed to allocate heap" durante la creazione del client Kinesis Video nella demo GStreamer su Raspberry Pi

L'applicazione di esempio GStreamer cerca di allocare 512 MB di RAM, che potrebbero non essere disponibili nel sistema. È possibile diminuire tale allocazione riducendo il seguente valore in `KinesisVideoProducer.cpp`:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Errore "Illegal Instruction" durante l'esecuzione della demo GStreamer su Raspberry Pi

Se riscontri il seguente errore durante l'esecuzione della demo di GStreamer, verifica di aver compilato l'applicazione per la versione corretta del tuo dispositivo. (Ad esempio, verifica di non eseguire la compilazione per Raspberry Pi 3 quando utilizzi Raspberry Pi 2.)

```
INFO - Initializing curl.  
Illegal instruction
```

La videocamera non viene caricata su Raspberry Pi

Per controllare se la videocamera viene caricata, eseguire quanto segue:

```
ls /dev/video*
```

Se l'operazione non ha esito, eseguire quanto segue:

```
vcgencmd get_camera
```

L'output visualizzato dovrebbe essere simile al seguente:

```
supported=1 detected=1
```

Se il driver non rileva la videocamera, procedere come segue:

1. Controllare la configurazione fisica della videocamera e verificare che sia collegata correttamente.
2. Eseguire quanto segue per aggiornare il firmware:

```
sudo rpi-update
```

3. Riavviare il dispositivo.
4. Eseguire quanto segue per caricare il driver:

```
sudo modprobe bcm2835-v4l2
```

5. Verificare che la videocamera sia stata rilevata:

```
ls /dev/video*
```

Impossibile trovare la videocamera su macOS High Sierra

Su macOS High Sierra, l'applicazione demo non è in grado di trovare la videocamera se ne è disponibile più di una.

File jni.h non trovato durante la compilazione su macOS High Sierra

Per risolvere questo errore, aggiornare l'installazione di Xcode alla versione più recente.

Errori di curl durante l'esecuzione dell'applicazione demo GStreamer

Per risolvere gli errori di curl durante l'esecuzione dell'applicazione demo GStreamer, copiare [questo file di certificato](#) in `/etc/ssl/cert.pem`.

Timestamp/range asserzione in fase di esecuzione su Raspberry Pi

Se si verifica un'asserzione di intervallo timestamp in fase di runtime, aggiornare il firmware e riavviare il dispositivo:

```
sudo rpi-update
$ sudo reboot
```

Asserzione su `gst_value_set_fraction_range_full` su Raspberry Pi

La seguente asserzione viene visualizzata se è in esecuzione il servizio `uv4l`:

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
denominator_end) < 0' failed
```

In questo caso, interrompere il servizio `uv4l` e riavviare l'applicazione.

STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) error on Android

Il seguente errore viene visualizzato se [Bandiere di adattamento NAL](#) non sono corretti per il flusso di file multimediali:

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Se si verifica questo errore, specificare il flag `.withNalAdaptationFlags` corretto per i file multimediali (ad esempio, `NAL_ADAPTATION_ANNEXB_CPD_NALS`). Fornire questo flag nella seguente riga di [Android](#):

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/StreamConfigurationFragment.java#L169>

È stata raggiunta la durata massima del frammento, errore

Questo errore si verifica quando un frammento di file multimediale in un flusso supera il limite massimo di durata. Vedi il limite massimo di durata del frammento nella [the section called “Quote dei servizi API multimediali e archiviati”](#) sezione.

Per risolvere il problema, prova a eseguire queste operazioni:

- Se stai usando una webcam/USB fotocamera, esegui una delle seguenti operazioni:
 - Se utilizzi la frammentazione basata su fotogrammi chiave, imposta l'encoder in modo che fornisca i fotogrammi chiave entro 10 secondi.
 - Se non utilizzi la frammentazione basata su key frame, quando definisci lo stream in [Scrivi ed esamina il codice](#), imposta il limite massimo di durata del frammento su un valore inferiore a 10 secondi.
 - Se stai usando codificatori software (come x264) nella pipeline GStreamer, puoi impostare l'attributo key-int-max su un valore entro 10 secondi. Ad esempio, imposta key-int-max su 60, con fps impostato su 30, per abilitare i fotogrammi chiave ogni 2 secondi.
- Se utilizzi una fotocamera RPI, imposta l'attributo keyframe-interval su un valore inferiore a 10 secondi.
- Se utilizzi una telecamera IP (RTSP), imposta la dimensione GOP su 60.

Errore "Invalid thing name passed (Passato nome di cosa non valido)" quando si utilizza l'autorizzazione IoT

Per evitare questo errore (HTTP Error 403: Response: {"message":"Invalid thing name passed"}) quando utilizzi le credenziali IoT per l'autorizzazione, assicurati che il valore di stream-name (un parametro obbligatorio dell'kvssink elemento) sia identico al valore di iot-thingname Per ulteriori informazioni, consulta [Riferimento ai parametri degli elementi GStreamer](#).

Problemi con la libreria Stream Parser

Questa sezione descrive i problemi che possono verificarsi durante l'utilizzo della [Trasmetti in streaming utilizzando la libreria parser](#).

Problemi

- [Impossibile accedere a un singolo fotogramma del flusso](#)
- [Errore di decodifica frammento](#)

Impossibile accedere a un singolo fotogramma del flusso

Per accedere a un singolo frame da una fonte di streaming nell'applicazione consumer, verifica che lo stream contenga i dati privati del codec corretti. Per informazioni sul formato dei dati in un flusso, consultare [Modello di dati](#).

Per informazioni su come utilizzare i dati privati del codec per accedere a un frame, consultate il seguente file di test sul GitHub sito Web: [KinesisVideoRendererExampleTest.java](#)

Errore di decodifica frammento

Se i frammenti non sono codificati correttamente in un H.264 formato e livello supportati dal browser, potresti visualizzare il seguente errore durante la riproduzione dello streaming nella console:

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264
content
```

In questo caso, verificare quanto segue:

- La risoluzione dei fotogrammi corrisponde alla risoluzione specificata nei dati privati del codec (Codec Private Data).
- Il H.264 profilo e il livello dei frame codificati corrispondono al profilo e al livello specificati nei dati privati del codec.
- Il browser supporta la combinazione. profile/level La maggior parte dei browser attuali supporta tutte le combinazioni di profilo e livello.
- I timestamp sono accurati e nell'ordine corretto, senza creazione di duplicati in corso.
- L'applicazione sta codificando i dati del frame utilizzando il H.264 formato.

Problemi di rete

Se si verificano errori di connessione, ad esempio «Connection Timeout» o «Connection Failed», quando si tenta di connettersi a Kinesis Video Streams, ciò potrebbe essere dovuto a limitazioni dell'intervallo di indirizzi IP nella configurazione di rete.

Se la configurazione prevede restrizioni sull'intervallo di indirizzi IP per Kinesis Video Streams, aggiorna la configurazione di rete per inserire nella lista consentita gli intervalli di indirizzi IP di Kinesis Video Streams. <https://ip-ranges.amazonaws.com/ip-ranges.json>

Important

L'elenco degli intervalli IP non è un elenco esaustivo degli indirizzi IP di Kinesis Video Streams. Includi gli intervalli di indirizzi IP visualizzati e tieni presente che gli indirizzi IP possono cambiare nel tempo.

Per ulteriori informazioni, consulta Intervalli [AWS IP](#). Per ricevere una notifica quando gli intervalli IP cambiano, segui la procedura di [abbonamento](#).

Cronologia dei documenti per Amazon Kinesis Video Streams

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di Amazon Kinesis Video Streams.

- Versione API più recente: 29-11-2017
- Ultimo aggiornamento della documentazione: 6 gennaio 2025

Modifica	Descrizione	Data
C++ su Raspberry Pi	Documentazione aggiornata per l'utilizzo del C++ Producer SDK su Raspberry Pi.	6 gennaio 2025
Connessione con Amazon Kinesis Video Edge-to-Cloud Streams Edge Agent	Rilascio di una nuova funzionalità. Per ulteriori informazioni, consulta Pianifica la registrazione e l'archiviazione video .	27 giugno 2023
Guida introduttiva: invio di dati a un flusso video Kinesis	Tutorial di base per l'invio di dati multimediali da una videocamera a un flusso video Kinesis. Per ulteriori informazioni, consulta Invia dati a uno streaming video di Amazon Kinesis .	21 gennaio 2019
Streaming di metadati	Puoi utilizzare Producer SDK per incorporare i metadati in un flusso video Kinesis. Per ulteriori informazioni, consulta Utilizzo dei metadati	28 settembre 2018

Modifica	Descrizione	Data
	di streaming con Kinesis Video Streams.	
Logging di C++ Producer SDK	Puoi configurare il logging per le applicazioni C++ Producer SDK. Per ulteriori informazioni, consulta Usa la registrazione con il C++ producer SDK.	18 luglio 2018
Streaming di video HLS	Ora puoi visualizzare uno streaming video Kinesis utilizzando HTTP Live Streaming. Per ulteriori informazioni, consulta Riproduzione di Kinesis Video Streams.	13 luglio 2018
Streaming da un'origine RTSP	Applicazione di esempio per Kinesis Video Streams che viene eseguita in un contenitore Docker e trasmette video da una sorgente RTSP. Per ulteriori informazioni, consulta RTSP e Docker.	20 giugno 2018
Plugin C++ Producer SDK GStreamer	Mostra come creare un C++ da utilizzare come destinazione GStreamer. Per ulteriori informazioni, consulta Plugin GStreamer - kvssink.	15 giugno 2018

Modifica	Descrizione	Data
Documentazione di riferimento sui callback dell'SDK producer	Documentazione di riferimen to per i callback utilizzati da Caricamento su Kinesis Video Streams . Per ulteriori informazioni, consulta richiamate SDK di Producer .	12 giugno 2018
Requisiti di sistema	Documentazione relativa ai requisiti di memoria e di storage per i dispositi vi producer e SDK. Per ulteriori informazioni, consulta Requisiti di sistema di Amazon Kinesis Video Streams .	30 maggio 2018
CloudTrail supporto	Documentazione CloudTrai l da utilizzare per monitorare l'utilizzo delle API. Per ulteriori informazioni, consulta Registra le chiamate API Amazon Kinesis Video Streams con AWS CloudTrail .	24 maggio 2018
Documentazione di riferimento sulle strutture SDK producer	Documentazione di riferimen to per le strutture utilizzate da Caricamento su Kinesis Video Streams . Per ulteriori informazioni, consultare Strutture SDK di Producer e Strutture dei flussi video Kinesis .	7 maggio 2018

Modifica	Descrizione	Data
Documentazione di esempio di Renderer	Documentazione per l'applicazione di esempio Renderer, che mostra come decodificare e visualizzare i frame da un flusso video Kinesis. Per ulteriori informazioni, consulta Esempio: analisi e rendering dei frammenti di Kinesis Video Streams .	15 marzo 2018
Documentazione di riferimento sui limiti dell'SDK producer	Le informazioni sui limiti per le operazioni in C++ . Per ulteriori informazioni, consulta Quote SDK di Producer .	13 marzo 2018
Monitoraggio	Informazioni sul monitoraggio delle metriche e delle chiamate API di Kinesis Video Streams tramite Amazon e CloudWatch AWS CloudTrail. Per ulteriori informazioni, consulta Monitoraggio dei flussi video di Amazon Kinesis .	5 febbraio 2018
Documentazione di riferimento per i flag di adattamento del Network Abstraction Layer (NAL)	Informazioni sull'impostazione dei flag di adattamento NAL nell'utilizzo di streaming di video. Per ulteriori informazioni, consulta Bandiere di adattamento NAL .	15 gennaio 2018

Modifica	Descrizione	Data
Supporto Android per lo streaming di video	Kinesis Video Streams ora supporta lo streaming di video da dispositivi Android. Per ulteriori informazioni, consulta Android .	12 gennaio 2018
Documentazione di esempio di Kinesis Video	Documentazione per l'applicazione di esempio Kinesis Video, che mostra come utilizzarla Guarda l'output delle telecamere usando la libreria parser in un'applicazione. Per ulteriori informazioni, consulta KinesisVideoExample .	09 gennaio 2018
Rilasciata la documentazione di Kinesis Video Streams	Rilascio iniziale della Developer Guide di Amazon Kinesis Video Streams.	29 novembre 2017

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.