Guía para desarrolladores

Amazon Kinesis Video Streams

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

Amazon Kinesis Video Streams: Guía para desarrolladores

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon Kinesis Video Streams? ... 1
Disponibilidad por región ... 1
Funcionamiento ... 3
Bibliotecas de API y productores ... 4
Modelo de datos ... 8
Requisitos del sistema ... 17
Requisitos de cámara ... 18
Sistemas operativos probados ... 19
Requisitos de almacenamiento del SDK ... 19
Cuotas ... 19
Cuotas de servicio de la API del plano de control ... 19
Cuotas del servicio de API multimedia y multimedia archivada ... 23
Cuotas de fragmentos de metadatos y fragmentos de medios ... 29
Cuotas del servicio de metadatos de streaming ... 32
Cuotas de SDK para productores ... 32
Configurar una cuenta de ... 36
Inscríbase en una Cuenta de AWS ... 36
Creación de un usuario con acceso administrativo ... 37
Cree una clave Cuenta de AWS ... 38
Ventajas ... 39
Introducción ... 41
Creación de una transmisión de vídeo de Amazon Kinesis ... 41
Cree una transmisión de vídeo mediante la consola ... 42
Cree una transmisión de vídeo mediante el AWS CLI ... 42
Enviar datos a una transmisión de vídeo de Amazon Kinesis ... 42
Cree el SDK y los ejemplos ... 43
Ejecute las muestras para cargar contenido multimedia en Kinesis Video Streams ... 46
Revise los objetos de reconocimiento ... 48
Consume datos multimedia ... 49
Vea contenido multimedia en la consola ... 49
Consume datos multimedia mediante HLS ... 49
Cargar a Kinesis Video Streams ... 50
Cliente productor de Kinesis Video Streams ... 50
Biblioteca de productores de Kinesis Video Streams ... 51

Comprenda qué son las bibliotecas de productores 52
Java 52
Procedimiento: utilice el SDK para productores de Java 53
Requisitos previos 53
Descargue y configure el código 54
Escribe y examina el código 55
Eliminar recursos 57
Ejecuta y verifica el código 57
Android 58
Procedimiento: utilice el SDK para productores de Android 58
Requisitos previos 59
Descarga y configura el código 62
Examine el código 64
Ejecute y verifique el código 66
C++ 67
Modelo de objetos 68
Coloca contenido multimedia en la transmisión 68
Interfaces de devolución de llamada 68
Procedimiento: utilice el SDK para productores de C++ 69
Requisitos previos 69
Descarga y configura el código 70
Escribe y examina el código 76
Ejecute y verifique el código 83
Usa el SDK como GStreamer complemento 83
Utilice el SDK del productor de C++ como GStreamer complemento en un contenedor de Docker 83
Usa el registro 84
Productor de C 84
Modelo de objetos 85
Coloca contenido multimedia en la transmisión 86
Procedimiento: utilice el SDK del productor de C 86
Requisitos previos 86
Descarga el código 87
Escribe y examina el código 88
Ejecute y verifique el código 91
C++ en Raspberry Pi 92

Requisitos previos ... 93
Creación de un usuario ... 93
Únete a tu red ... 95
Conéctese de forma remota ... 96
Configura la cámara ... 97
Requisitos previos de instalación del software ... 99
Descargue y cree el SDK para productores de C++ ... 100
Transmita una transmisión en directo ... 103
Reproduzca contenido multimedia ... 111
Solución de problemas ... 115
Referencia de códigos de error ... 119
Errores y códigos de estado devueltos por PutFrame Callbacks: código independiente de la plataforma (PIC) ... 120
Errores y códigos de estado devueltos por PutFrame llamadas de retorno: biblioteca de productores de C ... 174
Banderas de adaptación NAL ... 181
Estructuras de productores ... 183
DeviceInfo/DefaultDeviceInfoProvider ... 183
StorageInfo ... 183
Estructuras de transmisión ... 185
StreamDefinition/StreamInfo ... 185
ClientMetrics ... 203
StreamMetrics ... 204
Devoluciones de llamada ... 206
ClientCallbackProvider ... 207
StreamCallbackProvider ... 207
ClientCallbacks ... 208
Implementaciones de callback para volver a intentar la transmisión ... 213
Uso de metadatos de streaming ... 214
Añadir metadatos a una transmisión de vídeo de Kinesis ... 215
Reproducción de vídeo ... 218
Requisitos de reproducción ... 219
GetClip ... 219
Obtenga la URL de la sesión DASHStreaming ... 220
Obtenga la URL de la sesión HLSStreaming ... 221
GetImages ... 223

Reproducción con HLS ... 223
Utilícela AWS CLI para recuperar la URL de una sesión de streaming de HLS ... 224
Ejemplo: utilice HLS en HTML y JavaScript ... 228
Solución de problemas de HLS ... 232
Reproducción con MPEG-DASH ... 234
Ejemplo: usar MPEG-DASH en HTML y JavaScript ... 235
Configure las notificaciones ... 240
Administra las configuraciones de notificaciones ... 240
UpdateNotificationConfiguration ... 240
DescribeNotificationConfiguration ... 241
Acerca de las etiquetas MKV del productor ... 241
Sintaxis de las etiquetas MKV del productor ... 241
Límites de etiquetas MKV ... 241
Mensajes de Amazon SNS ... 242
Carga útil del tema Amazon SNS ... 242
Ver tus mensajes de Amazon SNS ... 243
Extraiga imágenes de transmisiones de vídeo ... 245
Generación de imágenes en tiempo real ... 245
Añadir etiquetas de generación de imágenes a los fragmentos ... 248
Ruta de objetos de Amazon S3 (imagen) ... 251
Recuperando los metadatos de la imagen ... 252
Recomendaciones de URI de Amazon S3 para protegerse contra la limitación ... 252
Solución de problemas ... 253
Acceda a la analítica de vídeo ... 256
Consume metadatos ... 256
Transmita usando la biblioteca de analizadores ... 257
Requisitos previos ... 258
Descarga el código ... 258
Examine el código ... 259
Ejecuta el código ... 266
Monitorización ... 266
Supervise las métricas con CloudWatch ... 267
Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch ... 287
Registre las llamadas a la API con CloudTrail ... 292
Límites de metadatos de transmisión ... 297
Programe la grabación y el almacenamiento de vídeo ... 298

Operaciones de la API de Amazon Kinesis Video Streams Edge Agent 299
Supervisión de Amazon Kinesis Video Streams Edge Agent 299
Implemente en AWS IoT Greengrass modo no funcional 299
Cómo instalar las dependencias 300
Cree recursos para su cámara IP RTSP URLs 302
Creación de una política de permisos de IAM 304
Creación de un rol de IAM 306
Cree el alias del AWS IoT rol 307
Cree la AWS IoT política 308
Crea cualquier AWS IoT cosa y obtén AWS IoT Core credenciales 310
Cree el agente Edge 312
Instale el CloudWatch agente 324
Ejecute Edge Agent como un proceso nativo 327
Implemente en AWS IoT Greengrass 330
Crea una instancia de Ubuntu 330
Configure el dispositivo AWS IoT Greengrass principal 332
Cree recursos para el RTSP de su cámara IP URLs 333
Agregue permisos a la función TES 335
Instale el componente Secret Manager 338
Implemente Edge Agent en el dispositivo 341
Instale el componente del administrador de AWS IoT Greengrass registros 351
Preguntas frecuentes 355
¿Qué sistemas operativos admite Amazon Kinesis Video Streams Edge Agent? 355
¿El agente Edge de Amazon Kinesis Video Streams es compatible con contenido multimedia H.265? 355
¿Funciona el agente Edge de Amazon Kinesis Video Streams en? AL2 355
¿Cómo puedo ejecutar varias transmisiones dentro de un objeto AWS IoT o dispositivo? 356
¿Cómo puedo editar un mensaje `StartEdgeConfigurationUpdate` después de haberlo enviado? 356
¿Tiene algún ejemplo común`ScheduleConfigs`? 356
¿Hay un límite máximo de transmisión? 357
¿Cómo puedo reiniciar un trabajo que se ha producido un error? 357
¿Cómo superviso el estado de mi agente Edge de Amazon Kinesis Video Streams? 357
Transmita vídeo a través de una VPC 358
Información adicional 358
Procedimientos de punto final de VPC 359

Ejemplos ... 361
Ejemplos: Envío de datos a Kinesis Video Streams ... 361
Ejemplos: Recuperación de datos de Kinesis Video Streams ... 361
Ejemplos: reproducción de datos de vídeo ... 361
Requisitos previos ... 361
GStreamer Complemento: kvssink ... 362
Descargue, cree y configure el GStreamer elemento ... 363
Ejecuta el elemento GStreamer ... 363
Lanzar comandos ... 364
Ejecute el GStreamer elemento en un contenedor de Docker ... 366
Referencia de parámetros ... 369
PutMedia API ... 384
Descarga y configura el código ... 385
Escribe y examina el código ... 386
Ejecute y verifique el código ... 388
RTSP y Docker ... 389
Tutoriales de vídeo ... 390
Requisitos previos ... 390
Cree la imagen de Docker ... 390
Ejecute la aplicación de ejemplo de RTSP ... 391
Renderer ... 392
Requisitos previos ... 393
Ejemplo de ejecución del renderizador ... 393
Cómo funciona ... 394
referencia de la API ... 396
Acciones ... 396
Amazon Kinesis Video Streams ... 398
Contenido multimedia de Amazon Kinesis Video Streams ... 524
Contenido multimedia archivado de Amazon Kinesis Video Streams ... 542
Canales de señalización de vídeo de Amazon Kinesis ... 593
Almacenamiento WebRTC de vídeo en Amazon Kinesis ... 602
Data Types ... 612
Amazon Kinesis Video Streams ... 613
Contenido multimedia de Amazon Kinesis Video Streams ... 654
Contenido multimedia archivado de Amazon Kinesis Video Streams ... 657
Canales de señalización de vídeo de Amazon Kinesis ... 676

Almacenamiento WebRTC de vídeo en Amazon Kinesis 678
Errores comunes 678
Parámetros comunes 680
Seguridad 683
Protección de los datos 684
¿Qué es el cifrado del lado del servidor para Kinesis Video Streams? 684
Cuestiones sobre costos, regiones y rendimiento 684
¿Cómo puedo comenzar a usar el cifrado en el servidor? 685
Creación y uso de una clave gestionada por el cliente 686
Permisos para usar una clave administrada por el cliente 687
Control del acceso a los recursos de Kinesis Video Streams mediante IAM 688
Sintaxis de la política 689
Acciones para Kinesis Video Streams 690
Nombres de recursos de Amazon (ARNs) para Kinesis Video Streams 690
Otorgar acceso a otras cuentas de IAM a una transmisión de vídeo de Kinesis 691
Ejemplos de políticas 694
Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT 697
AWS IoT ThingName como nombre de flujo 698
AWS IoT CertificateId como nombre de transmisión 704
Utilice AWS IoT las credenciales para transmitir a un nombre de transmisión codificado 706
Validación de la conformidad 707
Resiliencia 708
Seguridad de infraestructuras 708
Prácticas recomendadas de seguridad 709
Implementación del acceso a los privilegios mínimos 709
Uso de roles de IAM 709
Se usa CloudTrail para monitorear las llamadas a la API 710
Solución de problemas 711
Problemas generales 711
Latencia demasiado alta 711
Problemas con la API 712
Error: "Opciones desconocidas" 712
Error: "Unable to determine service/operation name to be authorized" (No se puede determinar el nombre del servicio o la operación que se debe autorizar) 712
Error: "Failed to put a frame in the stream" (Error al colocar un fotograma en la transmisión) 713

Error: «El servicio cerró la conexión antes de recibir AckEvent la conexión definitiva» 713
Error: "STATUS_STORE_OUT_OF_MEMORY" 714
Error: «La credencial debe estar limitada a una región válida». 714
Problemas de HLS 714
Problemas con Java 714
Habilitar los logs de Java 714
Problemas con la biblioteca del productor 715
No se puede compilar el SDK del productor 716
La transmisión de vídeo no aparece en la consola 716
Error: «El token de seguridad incluido en la solicitud no es válido» al transmitir datos mediante la aplicación de demostración GStreamer 717
Error: "Failed to submit frame to Kinesis Video client" (No se pudo enviar el fotograma al cliente de Kinesis Video) 717
GStreamer la aplicación se detiene con el mensaje «la transmisión se ha detenido, no se ha negociado el motivo» en OS X 718
Error: «No se pudo asignar el montón» al crear el Kinesis Video Client GStreamer en una demostración en Raspberry Pi 718
Error: «Instrucción ilegal» al ejecutar la GStreamer demostración en Raspberry Pi 718
La cámara no se carga en un dispositivo Raspberry Pi 719
No se puede encontrar la cámara en macOS High Sierra 719
No se ha encontrado el archivo jni.h al compilar en macOS High Sierra 720
Errores de Curl al ejecutar la aplicación de demostración GStreamer 720
Aserción de rango/marca temporal en el tiempo de ejecución en un dispositivo Raspberry Pi 720
Aserción en gst_value_set_fraction_range_full en un dispositivo Raspberry Pi 720
Error STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) en Android 720
Error al alcanzar la duración máxima del fragmento 721
Error «Nombre de objeto no válido pasado» al utilizar la autorización de IoT 721
Problemas con la biblioteca de analizadores de flujo 721
No se puede tener acceso a un solo fotograma de la transmisión 722
Error de descodificación de fragmentos 722
Problemas de red 723
Historial de documentos 724
..... dccxxix

¿Qué es Amazon Kinesis Video Streams?

Puede utilizar Amazon Kinesis Video Streams, un servicio totalmente Servicio de AWS gestionado, para transmitir vídeo en directo desde los dispositivos al dispositivo o crear aplicaciones para Nube de AWS el procesamiento de vídeo en tiempo real o el análisis de vídeo orientado a lotes.

Kinesis Video Streams no es solo almacenamiento de datos de vídeo. Puede utilizarlo para ver sus transmisiones de vídeo en tiempo real a medida que se reciben en la nube. Puede monitorizar sus transmisiones en directo en o desarrollar su propia aplicación de supervisión que utilice la biblioteca de API de Kinesis Video Streams para mostrar vídeo en directo. AWS Management Console

Puede utilizar Kinesis Video Streams para capturar enormes cantidades de datos de vídeo en directo de millones de fuentes, incluidos smartphones, cámaras de seguridad, cámaras web, cámaras integradas en automóviles, drones y otras fuentes. También puede enviar datos serializados en el tiempo que no sean de vídeo, como datos de audio, imágenes térmicas, datos de profundidad y datos de RADAR. A medida que el vídeo en directo se transmite desde estas fuentes a una transmisión de vídeo de Kinesis, puede crear aplicaciones para acceder a los datos en tiempo real para un procesamiento de baja latencia. frame-by-frame Kinesis Video Streams es independiente de la fuente. Puede transmitir vídeo desde la cámara web de un ordenador mediante la GStreamer Complemento: kvssink biblioteca o desde una cámara de su red mediante el protocolo de transmisión en tiempo real (RTSP).

También puede configurar la transmisión de vídeo de Kinesis para almacenar datos multimedia de forma duradera durante el período de retención especificado. Kinesis Video Streams almacena automáticamente estos datos y los cifra en reposo. Además, Kinesis Video Streams indexa los datos almacenados en función de las marcas de tiempo del productor y de las marcas de tiempo de ingesta. Puede crear aplicaciones que procesen por lotes los datos de vídeo de forma periódica, o puede crear aplicaciones que requieran un acceso único a los datos históricos para distintos casos de uso.

Sus aplicaciones personalizadas, en tiempo real o orientadas a lotes, pueden ejecutarse en instancias de Amazon EC2 . Estas aplicaciones pueden procesar datos mediante algoritmos de aprendizaje profundo de código abierto o aplicaciones de terceros que se integran con Kinesis Video Streams.

Disponibilidad por región

Amazon Kinesis Video Streams está disponible en las siguientes regiones:

Nombre de la región	AWS Código de región
Este de EE. UU. (Ohio)	us-east-2
Este de EE. UU. (Norte de Virginia)	us-east-1
Oeste de EE. UU. (Oregón)	us-west-2
AWS GovCloud (Este de EE. UU.)	us-gov-east-1
AWS GovCloud (Estados Unidos-Oeste)	us-gov-west-1
África (Ciudad del Cabo)	af-south-1
Asia-Pacífico (Hong Kong)	ap-east-1
Asia-Pacífico (Bombay)	ap-south-1
Asia-Pacífico (Seúl)	ap-northeast-2
Asia-Pacífico (Singapur)	ap-southeast-1
Asia-Pacífico (Sídney)	ap-southeast-2
Asia-Pacífico (Tokio)	ap-northeast-1
Canadá (centro)	ca-central-1
China (Beijing)	cn-north-1
Europa (Fráncfort)	eu-central-1
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (París)	eu-west-3
América del Sur (São Paulo)	sa-east-1

Kinesis Video Streams: cómo funciona

Temas

- Compatibilidad con la API de Kinesis Video Streams y las bibliotecas de productores
- Modelo de datos de Kinesis Video Streams

Puede utilizar Amazon Kinesis Video Streams, un servicio totalmente Servicio de AWS gestionado, para transmitir vídeo en directo desde los dispositivos Nube de AWS al dispositivo y almacenarlo de forma duradera. A continuación, puede crear sus propias aplicaciones de procesamiento de vídeo en tiempo real o realizar análisis de vídeo orientado a lotes.

El siguiente diagrama proporciona una descripción general del funcionamiento de Kinesis Video Streams.

El diagrama ilustra la interacción entre los siguientes componentes:

- Productor: cualquier fuente que coloque datos en una transmisión de vídeo de Kinesis. Un productor puede ser cualquier dispositivo generador de vídeo, como una cámara de seguridad, una cámara corporal, la cámara de un smartphone o la cámara de un salpicadero. Un productor también puede enviar datos que no sean de vídeo, como, por ejemplo, fuentes de audio, imágenes o datos de radar.

  Un único productor puede generar una o varias transmisiones de vídeo. Por ejemplo, una cámara de vídeo puede enviar datos de vídeo a una transmisión de vídeo de Kinesis y datos de audio a otra.

  - Bibliotecas de productores de Kinesis Video Streams: conjunto de software y bibliotecas que puede instalar y configurar en sus dispositivos. Puede utilizar estas bibliotecas para conectar y transmitir vídeo de forma segura y fiable de diferentes maneras, por ejemplo, en tiempo real, después de guardarlo en el búfer durante unos segundos, o mientras after-the-fact se carga contenido multimedia.
- Transmisión de vídeo de Kinesis: un recurso que puede utilizar para transportar datos de vídeo en directo, almacenarlos opcionalmente y hacer que los datos estén disponibles para su consumo tanto en tiempo real como por lotes o una sola vez. En una configuración típica, una transmisión de vídeo de Kinesis solo tiene un productor que publica datos en ella.

La transmisión puede constar de audio, vídeo y transmisiones de datos codificadas temporalmente similares, como fuentes de detección de profundidad, fuentes de radar y mucho más. Puede crear una transmisión de vídeo de Kinesis mediante el AWS Management Console o mediante programación mediante el. AWS SDKs

Varias aplicaciones independientes pueden consumir una transmisión de vídeo de Kinesis en paralelo.

- Consumidor: obtiene datos, como fragmentos y fotogramas, de una transmisión de vídeo de Kinesis para verlos, procesarlos o analizarlos. Por lo general, estos consumidores se denominan aplicaciones de Kinesis Video Streams. Puede escribir aplicaciones que consuman y procesen datos en Kinesis Video Streams en tiempo real o después de que los datos se hayan almacenado e indexado en el tiempo cuando no se requiera un procesamiento de baja latencia. Puede crear estas aplicaciones de consumo para que se ejecuten en EC2 instancias de Amazon.
  - Vea la salida de las cámaras mediante la biblioteca de analizadores— Permite a las aplicaciones de Kinesis Video Streams obtener contenido multimedia de la transmisión de vídeo de Kinesis de forma fiable y con baja latencia. Además, analiza los límites de los fotogramas en los medios, de modo que las aplicaciones puedan centrarse en el procesamiento y el análisis de los propios fotogramas.

Compatibilidad con la API de Kinesis Video Streams y las bibliotecas de productores

Kinesis Video Streams le APIs permite crear y administrar transmisiones y leer o escribir datos multimedia en y desde una transmisión. La consola Kinesis Video Streams, además de la funcionalidad de administración, también admite la transmisión en directo video-on-demand y la reproducción. Kinesis Video Streams también proporciona un conjunto de bibliotecas de productores que puede usar en el código de su aplicación para extraer datos de sus fuentes multimedia y cargarlos en su transmisión de vídeo de Kinesis.

Temas

- API de Kinesis Video Streams
- Patrón de detección de terminales
- Bibliotecas de productores

API de Kinesis Video Streams

Kinesis Video Streams APIs permite crear y administrar Kinesis Video Streams. También permite APIs leer y escribir datos multimedia en una transmisión, de la siguiente manera:

- API de productor: Kinesis Video Streams proporciona `PutMedia` una API para escribir datos multimedia en una transmisión de vídeo de Kinesis. En una solicitud `PutMedia`, el productor envía una transmisión de fragmentos de medios. Un fragmento es una secuencia de fotogramas autónoma. Los fotogramas que pertenecen a un fragmento no deben tener ningún tipo de dependencia de fotogramas de otros fragmentos. Para obtener más información, consulte PutMedia.

  A medida que llegan los fragmentos, Kinesis Video Streams asigna un número de fragmento único, en orden creciente. También almacena las marcas de tiempo del lado del productor y del lado del servidor para cada fragmento, como metadatos específicos de Kinesis Video Streams.

- Consumidor APIs: los consumidores pueden usar lo siguiente para obtener datos de una transmisión: APIs

  - `GetMedia`: al utilizar esta API, los consumidores deben identificar el fragmento de inicio. Posteriormente, la API devuelve los fragmentos en el orden en el que se agregaron a la transmisión (en orden ascendente por número de fragmento). Los datos de medios en los fragmentos se empaquetan en un formato estructurado, como Matroska (MKV). Para obtener más información, consulte GetMedia.

    > **Note**
    >
    > `GetMedia` sabe dónde están los fragmentos (archivados en el almacén de datos o disponibles en tiempo real). Por ejemplo, si `GetMedia` determina que el fragmento de inicio está archivado, comienza a devolver fragmentos desde el almacén de datos. Cuando debe devolver fragmentos más recientes que aún no están archivados, `GetMedia` pasa a leer fragmentos de un búfer de flujo en memoria.

    Este es un ejemplo de un consumidor continuo, que procesa fragmentos en el orden en el que se reciben en la transmisión.

    `GetMedia` permite que las aplicaciones de procesamiento de vídeo fallen o se retrasen y después se pongan al día sin esfuerzo adicional. Al usar `GetMedia`, las aplicaciones pueden

procesar datos archivados en el almacén de datos, y a medida que la aplicación se pone al día, `GetMedia` sigue enviando datos de medios en tiempo real a medida que llegan.

- `GetMediaFromFragmentList` (y `ListFragments`): las aplicaciones de procesamiento por lotes se consideran consumidores sin conexión. Los consumidores que no estén conectados a Internet pueden optar por buscar de forma explícita determinados fragmentos multimedia o gamas de vídeo combinando las teclas y. `ListFragments` `GetMediaFromFragmentList` APIs `ListFragments`y `GetMediaFromFragmentList` permitir que una aplicación identifique segmentos de vídeo para un intervalo de tiempo o un intervalo de fragmentos determinado y, a continuación, extraiga esos fragmentos secuencialmente o en paralelo para su procesamiento. Este enfoque es adecuado para conjuntos de aplicaciones `MapReduce`, que deben procesar rápidamente grandes cantidades de datos en paralelo.

  Por ejemplo, supongamos que un consumidor quiere procesar los fragmentos de vídeo de todo un día. El consumidor haría lo siguiente:

  1. Obtener una lista de fragmentos llamando a la API `ListFragments` y especificar un intervalo de tiempo para seleccionar la colección de fragmentos deseada.

     La API devuelve los metadatos de todos los fragmentos en el intervalo de tiempo especificado. Los metadatos proporcionan información como el número de fragmentos, las marcas horarias del lado del productor y del lado del servidor, etc.
  2. Tomar la lista de metadatos de fragmentos y recuperar fragmentos, en cualquier orden. Por ejemplo, para procesar todos los fragmentos del día, el consumidor podría optar por dividir la lista en sublistas y hacer que los trabajadores (por ejemplo, varias EC2 instancias de Amazon) busquen los fragmentos en paralelo utilizando el `GetMediaFromFragmentList` y los procesen en paralelo.

En el siguiente diagrama se muestra el flujo de datos de fragmentos durante estas llamadas a la API.

Cuando un productor envía una solicitud `PutMedia`, envía los metadatos de medios en la carga y, a continuación, envía una secuencia de fragmentos de datos de medios. Al recibir los datos, Kinesis Video Streams almacena los datos multimedia entrantes como fragmentos de Kinesis Video Streams. Cada fragmento consta de los elementos siguientes:

- Una copia de los metadatos de medios
- Un fragmento

- Metadatos específicos de Kinesis Video Streams; por ejemplo, el número de fragmento y las marcas horarias del lado del servidor y del productor

Cuando un consumidor solicita metadatos multimedia, Kinesis Video Streams devuelve una secuencia de fragmentos, empezando por el número de fragmento que especifique en la solicitud.

Si habilita la persistencia de datos para la transmisión, después de recibir un fragmento en la transmisión, Kinesis Video Streams también guarda una copia del fragmento en el banco de datos.

Patrón de detección de terminales

Plano de control REST APIs

Para acceder al plano de control APIs REST de Kinesis Video Streams, utilice los puntos finales del servicio Kinesis Video Streams.

Plano de datos: REST APIs

Kinesis Video Streams se ha creado con una arquitectura celular para garantizar mejores propiedades de escalado y aislamiento del tráfico. Como cada transmisión se asigna a una celda específica de una región, la aplicación debe usar los puntos finales correctos específicos de la celda a los que se ha mapeado la transmisión. Al acceder al plano de datos REST APIs, tendrá que gestionar y mapear usted mismo los puntos finales correctos. Este proceso, el patrón de detección de puntos finales, se describe a continuación:

1. El patrón de detección de puntos finales comienza con una llamada a una de las `GetEndpoints` acciones. Estas acciones pertenecen al plano de control.
    1. Si va a recuperar los puntos finales de los the section called “Contenido multimedia archivado de Amazon Kinesis Video Streams” servicios the section called “Contenido multimedia de Amazon Kinesis Video Streams” o servicios, utilice. the section called “GetDataEndpoint”
    2. Si va a recuperar los puntos finales para Kinesis Video Signaling the section called “Canales de señalización de vídeo de Amazon Kinesis”the section called “Almacenamiento WebRTC de vídeo en Amazon Kinesis”, utilice. the section called “GetSignalingChannelEndpoint”
2. Guarde en caché y reutilice el punto final.
3. Si el punto final almacenado en caché ya no funciona, realice una nueva llamada `GetEndpoints` a para actualizar el punto final.

Bibliotecas de productores

Tras crear una transmisión de vídeo de Kinesis, puede empezar a enviar datos a la transmisión. En el código de su aplicación, puede utilizar estas bibliotecas para extraer datos de sus fuentes multimedia y cargarlos en la transmisión de vídeo de Kinesis. Para obtener más información acerca de las producer libraries, consulte Cargar a Kinesis Video Streams.

Modelo de datos de Kinesis Video Streams

El Cargar a Kinesis Video Streams y el the section called “Transmita usando la biblioteca de analizadores” envían y reciben los datos de vídeo en un formato compatible con la integración de información junto a los datos de vídeo. Este formato se basa en la especificación Matroska (MKV).

El formato MKV es una especificación abierta para los datos de medios. Todas las bibliotecas y ejemplos de código de la Guía para desarrolladores de Amazon Kinesis Video Streams envían o reciben datos en formato MKV.

Cargar a Kinesis Video StreamsUtiliza los `Frame` tipos `StreamDefinition` y para producir encabezados de transmisiones MKV, encabezados de fotogramas y datos de fotogramas.

Para obtener información sobre la especificación completa de MKV, consulte Matroska Specifications.

En las siguientes secciones se describen los componentes de datos con formato MKV producidos por la C++.

Temas

- Elementos de encabezado de transmisión
- Transmita los datos de las pistas
- Elementos de encabezado de fotogramas
- Datos de fotogramas MKV

Elementos de encabezado de transmisión

Los siguientes elementos de encabezado MKV los utiliza `StreamDefinition` (definidos en `StreamDefinition.h`).

Elemento	Descripción	Valores típicos
stream_name	Corresponde al nombre de la transmisión de vídeo de Kinesis.	my-stream
retention_period	Kinesis Video Streams conserva la duración, en horas, de los datos de la transmisión. Especifique `0` una transmisión que no retenga datos.	24
etiquetas	colección clave-valor de datos de usuarios. Estos datos se muestra en la AWS Management Console y pueden leerlos las aplicacio nes cliente para filtrar u obtener información sobre una transmisión.	
kms_key_id	Si está presente, la AWS KMS clave definida por el usuario se utiliza para cifrar los datos de la transmisión. Si no está presente, los datos se cifran con la clave proporcionada por Kinesis (). `aws/kines isvideo`	01234567-89ab-cdef -0123-456789ab
streaming_type	Actualmente, el único tipo de streaming válido es `STREAMING_TYPE_REA LTIME` .	STREAMING_TYPE_REA LTIME
content_type	El tipo de contenido definido por el usuario. Para que los	vídeo/h264

Elemento	Descripción	Valores típicos
	datos de streaming de vídeo se reproduzcan en la consola, el tipo de contenido debe ser `video/h264` .	
max_latency	Este valor no se usa actualmente y debe establecerse en 0.	0
fragment_duration	Este valor es una estimación de la duración recomendada de sus fragmentos y se utiliza con fines de optimización. La duración real del fragmento se determina mediante los datos de streaming.	2
timecode_scale	Indica la escala utilizada por las marcas de tiempo de los fotogramas. El valor predeterminado es 1 milisegundo. Si se especifica a `0`, se asigna también el valor predeterminado de 1 milisegundo. Este valor puede estar comprendido entre 100 nanosegundos y 1 segundo. Para obtener más información, consulte la documentación TimecodeScalede Matroska.	
key_frame_fragmentation	Si se establece en `true`, la transmisión inicia un nuevo clúster cuando recibe un fotograma clave.	true

Elemento	Descripción	Valores típicos
frame_timecodes	Si`true`, Kinesis Video Streams utiliza los valores de marca de tiempo de presentación (pts) y de marca de tiempo de decodificación (dts) de los fotogramas recibidos. Si`false`, Kinesis Video Streams estampa los fotogramas cuando se reciben con valores de tiempo generados por el sistema.	true
absolute_fragment_time	Si es `true`, los códigos temporales del clúster se interpretan como si se usara un tiempo absoluto (por ejemplo, el del reloj del sistema del productor). Si es `false`, los códigos temporale s del clúster se interpretan como relativos a la hora de inicio de la transmisión.	true
fragment_acks	Si`true`, los acuses de recibo (ACKs) se envían cuando Kinesis Video Streams recibe los datos. Se ACKs pueden recibir mediante llamadas `KinesisVideoStream FragmentAck` o `KinesisVideoStream ParseFragmentAck` llamadas.	true

Elemento	Descripción	Valores típicos
restart_on_error	Indica si la transmisión debe reanudarse después de que se presente un error en la misma.	true
nal_adaptation_flags	Indica si los datos privados del códec o adaptación NAL (Network Abstraction Layer) están presentes en el contenido. Los indicadores válidos son `NAL_ADAPTATION_ANNEXB_NALS` y `NAL_ADAPTATION_ANNEXB_CPD_NALS` .	NAL_ADAPTATION_ANNEXB_NALS
frame_rate	Una estimación de la velocidad de fotogramas del contenido. Este valor se utiliza para la optimización; la velocidad de fotogramas real se determina en función de la velocidad de los datos entrantes. Si se especifica `0`, se asigna el valor predeterminado de `24`.	24

Elemento	Descripción	Valores típicos
avg_bandwidth_bps	Una estimación del ancho de banda del contenido, en Mbps. Este valor se utiliza para la optimización; la velocidad real se determina en función del ancho de banda de los datos entrantes. Por ejemplo, para una transmisión de vídeo con una resolución de 720p que se ejecute a 25 FPS, podría esperar un ancho de banda medio de 5 Mbps.	5
buffer_duration	periodo que ha de almacenar se en búfer el contenido en el productor. Si la latencia de red es baja, este valor se puede reducir. Si la latencia de la red es alta, al aumentar este valor se evita que las tramas se descarten antes de que puedan enviarse, ya que la asignación no puede colocar las tramas en el búfer más pequeño.	

Elemento	Descripción	Valores típicos
replay_duration	Cantidad de tiempo que se «rebobina» la transmisión de datos de vídeo si se pierde la conexión. Este valor puede ser cero si la pérdida de fotogramas debido a la pérdida de conexión no es motivo de preocupación. El valor se puede aumentar si la aplicación consumidora puede eliminar las tramas redundantes. Este valor debe ser inferior a la duración del búfer; de lo contrario, se utilizará la duración del búfer.	
connection_staleness	periodo que se mantiene una conexión cuando no se reciben datos.	
codec_id	El códec utilizado por el contenido. Para obtener más información, consulte CodecID en la especificación de Matroska.	V_ MPEG2
track_name	El nombre de la pista definido por el usuario.	my_track

Elemento	Descripción	Valores típicos
codecPrivateData	datos facilitados por el codificador empleado para descodificar los datos de los fotogramas, por ejemplo, la anchura y altura de los mismos en píxeles, ya que es una información necesaria para muchos consumidores a posteriori. En la bibliotec a del productor de C++, la `gMkvTrackVideoBits` matriz `MkvStatics.cpp` incluye el ancho y el alto de los píxeles del marco.	
codecPrivateDataSize (Tamaño)	El tamaño de los datos en el parámetro `codecPriv ateData` .	
track_type	El tipo de pista para el flujo.	MKV_TRACK_INFO_TYP E_AUDIO o MKV_TRACK _INFO_TYPE_VIDEO
segment_uuid	Uuid de segmento definido por el usuario (16 bytes).	
default_track_id	Número único distinto de cero de la pista.	1

Transmita los datos de las pistas

Los siguientes elementos de pista MKV los utiliza `StreamDefinition` (definidos en `StreamDefinition.h`).

Elemento	Descripción	Valores típicos
track_name	Nombre de pista definido por el usuario. Por ejemplo, "audio" para la pista de audio.	audio
codec_id	ID del códec para la pista. Por ejemplo, "A_AAC" para una pista de audio.	A_AAC
cpd	Datos que proporciona el codificador usado para descodificar los datos del fotograma. Estos datos pueden incluir la anchura y la altura del fotograma en píxeles, lo cual es necesario para muchos consumidores a posteriori. En la biblioteca de productores de C++, la gMkvTrack VideoBits matriz de MkvStatics .cpp incluye el ancho y el alto de los píxeles del marco.	
cpd_size	El tamaño de los datos del parámetro. codecPrivateData	
track_type	El tipo de la pista. Por ejemplo, puede utilizar el valor de enumeración de MKV_TRACK_INFO_TYPE_AUDIO para el audio.	MKV_TRACK_INFO_TYPE_AUDIO

Elementos de encabezado de fotogramas

Los siguientes elementos de encabezado MKV los utiliza `Frame` (definidos en el paquete `KinesisVideoPic`, en `mkvgen/Include.h`):

- Frame Index: valor con aumento monotónico.
- Flags: tipo de fotograma. Entre los valores válidos se incluyen los siguientes:
  - `FRAME_FLAGS_NONE`
  - `FRAME_FLAG_KEY_FRAME`: si `key_frame_fragmentation` está configurado en la transmisión, los fotogramas claves inician un nuevo fragmento.
  - `FRAME_FLAG_DISCARDABLE_FRAME`: indica al descodificador que puede descartar este fotograma si la descodificación es lenta.
  - `FRAME_FLAG_INVISIBLE_FRAME`: la duración de este bloque es 0.
- Marca de tiempo de decodificación: la marca de tiempo del momento en que se decodificó este fotograma. Si los fotogramas anteriores dependen de este fotograma para la decodificación, esta marca de tiempo puede ser anterior a la de los fotogramas anteriores. Este valor es relativo al inicio del fragmento.
- Marca temporal de la presentación: marca temporal en la que se muestra este fotograma. Este valor es relativo al inicio del fragmento.
- Duration: duración de reproducción del fotograma.
- Size: tamaño en bytes de los datos del fotograma

Datos de fotogramas MKV

Los datos de `frame.frameData` podrían contener solamente datos multimedia del fotograma o también información anidada adicional de encabezados, en función del esquema de codificación empleado. Para mostrarse en el AWS Management Console, los datos deben estar codificados en el códec H.264, pero Kinesis Video Streams puede recibir flujos de datos serializados en el tiempo en cualquier formato.

Requisitos del sistema para Amazon Kinesis Video Streams

Las siguientes secciones contienen los requisitos de hardware, software y almacenamiento para Amazon Kinesis Video Streams.

Temas

- Requisitos de cámara
- Sistemas operativos probados
- Requisitos de almacenamiento del SDK

Requisitos de cámara

Las cámaras que se utilizan para ejecutar el SDK y las muestras del productor de Kinesis Video Streams tienen los siguientes requisitos de memoria:

- La vista de contenido de SDK requiere 16 MB de memoria.
- La configuración predeterminada de la aplicación de muestra es 512 MB. Este valor es adecuado para los productores que tengan buena conectividad de red y no tengan requisitos de almacenamiento en búfer adicional. Si la conectividad de red es deficiente y se requiere más almacenamiento en búfer, puede calcular el requisito de memoria por segundo de buffering multiplicando la velocidad de fotogramas por segundo por el tamaño de memoria del fotograma. Para obtener más información acerca de la asignación de memoria, consulte StorageInfo.

Le recomendamos que utilice cámaras USB o RTSP (Real Time Streaming Protocol) que codifican datos utilizando H.264 ya que esto elimina la carga de trabajo de codificación de la CPU.

Actualmente, la aplicación de demostración no admite el Protocolo de datagramas de usuario (UDP) para la transmisión por RTSP. Esta capacidad se añadirá en el futuro.

El SDK para productores es compatible con los siguientes tipos de cámaras:

- Cámaras web.
- Cámaras USB.
- Cámaras con codificación H.264 (opción preferida).
- Cámaras sin codificación H.264.
- Módulo de cámara Raspberry Pi. Esta es la opción preferida para dispositivos Raspberry Pi, ya que se conecta a la GPU para transferencia de datos de vídeo, por lo que no hay ninguna capacidad adicional para el procesamiento de CPU.
- Cámaras (red) RTSP. Estas cámaras son preferibles porque las secuencias de vídeo ya están codificadas en H.264.

Sistemas operativos probados

Hemos probado cámaras web y cámaras RTSP con los siguientes dispositivos y sistemas operativos:

- Mac mini
  - High Sierra
- MacBook Portátiles profesionales
  - Sierra (10.12)
  - El Capitán (10,11)
- Portátiles HP con Ubuntu 16.04
- Ubuntu 17.10 (contenedor Docker)
- Raspberry Pi 3

Requisitos de almacenamiento del SDK

La instalación del Cargar a Kinesis Video Streams tiene un requisito de almacenamiento mínimo de 170 MB y un requisito de almacenamiento recomendado de 512 MB.

Cuotas de servicio de Amazon Kinesis Video Streams

Kinesis Video Streams tiene las siguientes cuotas de servicio:

> **Important**
>
> Las siguientes cuotas de servicio son flexibles, que se pueden ampliar mediante la presentación de un ticket de soporte, o fijas, que no se pueden aumentar. En las tablas siguientes, verás [s] y [h] junto a la cuota de servicio individual.

Cuotas de servicio de la API del plano de control

En la siguiente sección, se describen las cuotas de servicio para el plano de control APIs. TPS son las siglas de Transactions per Second (transacciones por segundo).

Cuando se alcanza un límite de solicitudes a nivel de cuenta o de recurso, se lanza a. `ClientLimitExceededException`

API	Límite de cuenta: solicitud	Límite de cuenta: Streams	Límite de nivel de transmisión	Excepciones y notas relevantes
CreateStream	50 TPS [f]	10 000 transmisiones por cuenta (s) en todas las regiones compatibles. Note Este límite se puede aumentar hasta 100 000 (o más) secuencias por cuenta. Inicia sesión en AWS Management Console at https://console.aws.amazon.com/		Todos los dispositivos CLIs, el acceso controlado por el SDK y la consola pueden invocar esta API. Una llamada a la API se ejecutará correctamente solo si aún no existe la transmisión.

API	Límite de cuenta: solicitud	Límite de cuenta: Streams	Límite de nivel de transmisión	Excepciones y notas relevantes
		y solicita un aumento de este límite.		
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]	

API	Límite de cuenta: solicitud	Límite de cuenta: Streams	Límite de nivel de transmisión	Excepciones y notas relevantes
DescribeStream	300 TPS [r]	N/A	5 TPS [r]	
GetDataEndpoint	300 TPS [r]	N/A	5 TPS [r]	Se ejecuta cada 45 minutos para actualizar el token de streaming para la mayoría de los casos de uso de `PutMedia`/`GetMedia`. El almacenamiento de puntos de enlace de datos en caché es seguro si la aplicación los vuelve a cargar en caso de error.
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]	

API	Límite de cuenta: solicitud	Límite de cuenta: Streams	Límite de nivel de transmisión	Excepciones y notas relevante s
UpdateImageGenerationConfiguration	50 TPS [r]	N/A	5 TPS [r]	
UpdateNotificationConfiguration	50 TPS [r]	N/A	5 TPS [r]	
UpdateStream	50 TPS [r]	N/A	5 TPS [r]	

Cuotas del servicio de API multimedia y multimedia archivada

En la siguiente sección, se describen las cuotas de servicio para los medios multimedia y archivados. APIs

Cuando se alcanza un límite de solicitudes a nivel de cuenta o de recursos, se lanza un. `ClientLimitExceededException`

Cuando se alcanza un límite de nivel de conexión, se produce una excepción `ConnectionLimitExceededException`.

Los siguientes errores o acks se producen cuando se alcanza un límite de nivel de fragmento:

- Se devuelve un ack `MIN_FRAGMENT_DURATION_REACHED` para un fragmento inferior a la duración mínima.
- Se devuelve un ack `MAX_FRAGMENT_DURATION_REACHED` para un fragmento superior a la duración máxima.
- Se devuelve un ack `MAX_FRAGMENT_SIZE` para un fragmento superior al tamaño máximo de datos.
- Se produce una excepción `FragmentLimitExceeded` si se alcanza un límite de fragmentos en una operación `GetMediaForFragmentList`.

Cuotas de servicio de la API de Data Plane

API	Límite de nivel de transmisión	Límite de nivel de conexión	Límite de ancho de banda	Límite de nivel de fragmento	Excepciones y notas relevantes
PutMedia	5 TPS [r]	1 [h]	12,5 MB/segundo o 100 Mbps [s] por transmisión	• Duración mínima del fragmento: 1 segundo [r] • Duración máxima del fragmento: 20 segundos [h] • Tamaño máximo del fragmento: 50 MB [r] • Número máximo de pistas: 3 [f] • Número máximo de fragmentos	PutMedialas solicitudes son conexiones en streaming y de larga duración. No necesitas abrir una conexión nueva para cada dato, ya que puedes enviar varios fragmentos en una única conexión persistente. Si intenta realizar más de una `PutMedia` conexión simultánea, Kinesis Video Streams limita las conexiones más recientes con un mensaje de error. `ConnectionLimitExceededException`

API	Límite de nivel de transmisión	Límite de nivel de conexión	Límite de ancho de banda	Límite de nivel de fragmento	Excepciones y notas relevantes
				enviados por segundo: 5 [h] • Límite máximo de metadatos de fragmentos: 10 etiquetas [h]	
GetClip	N/A	N/A	Límite de tamaño de 100 MB [h]	Número máximo de fragmentos: 200 [r]	
Obtenga la URL de la sesión DASHStreaming	25 TPS [h]	N/A	N/A	N/A	
Obtenga HLSStreaming la URL de la sesión	25 TPS [h]	N/A	N/A	N/A	

API	Límite de nivel de transmisión	Límite de nivel de conexión	Límite de ancho de banda	Límite de nivel de fragmento	Excepciones y notas relevantes
GetImages	N/A	N/A	100 MB [h]	N/A	El número máximo de imágenes por solicitud es de 100 [h]. **Note** El valor mínimo `SamplingInterval` es de 200 milisegundos (ms), es decir, 5 imágenes por segundo.

API	Límite de nivel de transmisión	Límite de nivel de conexión	Límite de ancho de banda	Límite de nivel de fragmento	Excepciones y notas relevantes
GetMedia	5 TPS [r]	3 [h]	25 MB/s o 200 Mbps [s]	Se envían un máximo de 5 fragmentos por segundo [h]	GetMedialas solicitudes son conexiones en streaming y de larga duración. No necesitas abrir una conexión nueva para cada dato, ya que puedes enviar varios fragmentos en una única conexión persistente. Si intenta realizar más de tres `GetMedia` conexiones simultáneas, Kinesis Video Streams limita las últimas conexiones con un mensaje de error. `ConnectionLimitExceededException` Si un fragmento típico es de aproximadamente 5 MB, este límite significa aproximadamente 75 MBps por transmisión de vídeo de Kinesis. En un flujo como este, la velocidad de bits de salida sería el doble de la velocidad de bits de entrada máxima de los flujos.

API	Límite de nivel de transmisión	Límite de nivel de conexión	Límite de ancho de banda	Límite de nivel de fragmento	Excepciones y notas relevantes
					Note `GetMediano` se utiliza para HLS/DASH la reproducción.
GetMediaForFragmentList	N/A	5 [f]	25 MB/s o 200 Mbps [s]	Número máximo de fragmentos: 1000 [h]	Se pueden invocar simultáneamente cinco aplicaciones consumidoras basadas en fragmentos. `GetMediaForFragmentList` Las conexiones adicionales se rechazarán.

Cuotas de servicio de protocolo de reproducción de vídeo (API)

API	Límite de nivel de sesión	Límite de nivel de fragmento
Obtenga la DASHManifest lista de reproducción	5 TPS [r]	Número máximo de fragmentos por lista de reproducción: 5.000 [h]
Obtener HLSMaster lista de reproducción	5 TPS [r]	N/A
Obtener HLSMedia lista de reproducción	5 TPS [r]	Número máximo de fragmentos por lista de reproducción: 5.000 [h]
¡Obtén MP4 InitFragment	5 TPS [r]	N/A

API	Límite de nivel de sesión	Límite de nivel de fragmento
Consigue MP4 MediaFragment	20 TPS [h]	N/A
¡Obtén TSFragment	20 TPS [h]	N/A

Cuotas de fragmentos de metadatos y fragmentos de medios

Las APIs transmisiones de Kinesis Video Streams para acceder a los medios archivados se limitan en función del número de fragmentos solicitados y no del número de llamadas a la API. APIs tienen una velocidad limitada tanto por el número de fragmentos de metadatos como por el número de fragmentos de medios que se solicitan. Las cuotas de metadatos de fragmentos y medios de fragmentos se aplican por transmisión. En otras palabras, las solicitudes de fragmentos de metadatos o contenido multimedia en una transmisión no se aplican a las cuotas de otra transmisión. Sin embargo, dentro de una transmisión determinada, cada cuota se comparte entre varias APIs. Esto significa que, para una transmisión determinada, las solicitudes de fragmentos de diferentes canales APIs consumen la misma cuota. Cuando se supera la cuota de metadatos o fragmentos de contenido multimedia de una transmisión, la API devuelve un`ClientLimitExceededException`. En las siguientes tablas, se muestra el APIs consumo de cada uno de los dos tipos de cuota. Para la segunda columna de estas tablas, supongamos que si una transmisión tiene una cuota de N, significa que APIs tiene N puntos para consumir de ese tipo de cuota para esa transmisión. La `GetClip` API aparece en ambas tablas.

Consumo de cuotas de metadatos fragmentados

API	Número de puntos de cuota consumidos por solicitud	Cuota compartida (N)
`ListFragments`	Valor del `MaxResults` parámetro	10 000 puntos de cuota por segundo, por transmisión [h]
`GetClip`	Número de fragmentos del clip resultante	

API	Número de puntos de cuota consumidos por solicitud	Cuota compartida (N)
`GetHLSMediaPlaylist`	Valor del `MaxMediaPlaylistFragmentResults` parámetro	
`GetDASHManifest`	Valor del `MaxManifestFragmentResults` parámetro	
`GetImages`	Valor de 400 + número máximo de imágenes solicitadas	

Consumo de cuotas de medios fragmentados

API	Número de puntos de cuota consumidos por solicitud	Cuota compartida (N)
`GetMediaForFragmentList`	Número de fragmentos en el parámetro Fragments	500 puntos de cuota por segundo, por transmisión [h]
`GetClip`	Número de fragmentos del clip resultante	
`GetMP4MediaFragment`	1	
`GetTSFragment`	1	
`GetImages`	Número máximo de imágenes solicitadas	

Por ejemplo, con una cuota de 500 fragmentos de contenido multimedia por segundo, se admiten los siguientes patrones de llamadas para una transmisión concreta:

- 5 solicitudes por segundo hasta `GetClip` 100 fragmentos en cada clip.

- 100 solicitudes por segundo o 5 fragmentos en cada clip. `GetClip`
- 2 solicitudes por segundo `GetClip` con 100 fragmentos en cada clip y 3 solicitudes por segundo `GetMediaForFragmentList` en cada clip.
- 400 solicitudes por segundo para `GetMP4MediaFragment` y 100 solicitudes por segundo para`GetTSFragment`.

Estas cuotas tienen una implicación importante en cuanto al número de sesiones HLS y MPEG-DASH que se pueden admitir por transmisión. No hay límite en cuanto al número de sesiones HLS y DASH que pueden utilizar los reproductores multimedia en un momento dado. Por lo tanto, es importante que la aplicación de reproducción no permita utilizar demasiadas sesiones simultáneamente. Los dos ejemplos siguientes describen cómo determinar el número de sesiones de reproducción simultánea que se pueden admitir:

Ejemplo 1: Transmisión en directo

En un escenario de transmisión en directo con HLS con fragmentos de 1 segundo de duración, una pista de audio y vídeo y `MaxMediaPlaylistFragmentResults` configurado en cinco, un reproductor multimedia suele realizar dos llamadas `GetHLSMediaPlaylist` por segundo. Una llamada es para obtener los metadatos de vídeo más recientes y otra para los metadatos de audio correspondientes. Las dos llamadas consumen cinco puntos de cuota de metadatos de fragmentos cada una. También realiza dos llamadas `GetMP4MediaFragment` por segundo: una llamada para el vídeo más reciente y otra para el audio correspondiente. Cada llamada consume un fragmento de token multimedia, por lo que se consumen dos tokens en total.

En este escenario, se pueden admitir hasta 250 sesiones de reproducción simultáneas. Con 250 sesiones, este escenario consume 2500 puntos de cuota de metadatos de fragmentos por segundo (muy por debajo de la cuota de 10 000) y 500 puntos de cuota de fragmentos de medios por segundo.

Ejemplo 2: reproducción bajo demanda

En un escenario de reproducción bajo demanda de un evento pasado con MPEG-DASH, una pista de audio y vídeo `MaxManifestFragmentResults` configurada en 1000, un reproductor multimedia normalmente llama una `GetDASHManifest` vez al inicio de la sesión (consume 1000 puntos de cuota de metadatos de fragmentos) y llama a una `GetMP4MediaFragment` velocidad de hasta 5 veces por segundo (consume 5 puntos de cuota de fragmentos de media) hasta que se carguen todos los fragmentos. En este escenario, se pueden iniciar hasta 10 sesiones nuevas por segundo (justo con la cuota de 10 000 fragmentos de metadatos por segundo) y hasta 100 sesiones pueden

cargar activamente fragmentos multimedia a una velocidad de 5 por segundo (justo con la cuota de 500 fragmentos multimedia por segundo).

Puede utilizar `ArchivedFragmentsConsumed.Metadata` y `ArchivedFragmentsConsumed.Media` supervisar el consumo de metadatos fragmentados y los puntos de cuota de medios fragmentados, respectivamente. Para obtener información sobre la supervisión, consultethe section called “Monitorización”.

Cuotas del servicio de metadatos de streaming

Las siguientes cuotas de servicio se aplican a la adición de metadatos de streaming a una transmisión de vídeo de Kinesis:

- Puede anexar hasta 10 elementos de metadatos a un fragmento.
- Un nombre de metadatos de fragmento puede tener hasta 128 bytes de longitud.
- Un valor de metadatos de fragmento puede tener hasta 256 bytes de longitud.
- El nombre de un fragmento de metadatos no puede empezar por la cadena "`AWS`». Si se añade este tipo de elemento de metadatos, el método `putFragmentMetadata` del PIC devuelve un error `STATUS_INVALID_METADATA_NAME` (código de error `0x52000077`). La aplicación puede hacer caso omiso del error (el PIC no añade el elemento de metadatos) o responder a él.

Cuotas de SDK para productores

La siguiente tabla contiene las cuotas actuales para los valores del SDK. Para obtener más información, consulte Cargar a Kinesis Video Streams.

> **Note**
>
> Antes de configurar estos valores, debe validar sus entradas. El SDK no valida estos límites y se produce un error en tiempo de ejecución si se superen los límites.

Valor	Límite	Notas
Recuento máximo de secuencias	128	El número máximo de secuencias que puede crear un objeto de Producer. Se

Valor	Límite	Notas
		trata de un límite flexible (puede solicitar un aumento). Garantiza que el productor no cree accidentalmente transmisiones de forma recursiva.
Longitud máxima de nombre de dispositivo	128 caracteres	
Recuento máximo de etiquetas	50 por secuencia	
Longitud máxima de nombre de secuencia	256 caracteres	
Tamaño mínimo de almacenamiento	10 MiB = 10 * 1024 * 1024 bytes	
Tamaño máximo de almacenamiento	10 GiB = 10 * 1024 * 1024 * 1024 bytes	
Longitud máxima de ruta de directorio raíz	4,096 caracteres	
Longitud máxima de informaci ón de autenticación	10 000 bytes	
Longitud máxima de cadena URI	10,000 caracteres	
Longitud máxima de nombre de etiqueta	128 caracteres	
Longitud máxima de valor de etiqueta	1,024 caracteres	

Valor	Límite	Notas
Período mínimo de token de seguridad	30 segundos	
Período de gracia de token de seguridad	40 minutos	Si la duración especificada es mayor, se limita a este valor.
Periodo de retención	0 o mayor que una hora	0 indica que no hay ninguna retención.
Duración mínima de clúster	1 segundo	El valor está especificado en unidades de 100 ms, que es el estándar de SDK.
Duración máxima de clúster	30 segundos	El valor está especificado en unidades de 100 ms, que es el estándar de SDK. La API de backend puede imponer una duración de clúster más corta.
Tamaño máximo de fragmento	50 MB	Para obtener más información, consulte Cuotas de servicio de Amazon Kinesis Video Streams.
Duración máxima de fragmento	20 segundos	Para obtener más información, consulte Cuotas de servicio de Amazon Kinesis Video Streams.
Duración máxima de conexión	45 minutos	El backend cierra la conexión después de este tiempo. El SDK gira el token y establece una nueva conexión dentro de este tiempo.

Valor	Límite	Notas
Longitud máxima de segmento ACK	1,024 caracteres	Longitud máxima del segmento del reconocim iento enviado a la función de analizador ACK.
Longitud máxima de cadena de tipo de contenido	128 caracteres	
Longitud máxima de cadena de ID de códec	32 caracteres	
Longitud máxima de cadena de nombre de pista	32 caracteres	
Longitud máxima de datos privados de códec	1 MiB = 1 * 1024 * 1024 bytes	
Longitud mínima de valor de escala de código temporal	100 ns	El valor de escala mínimo de código temporal para representar las marcas temporales del fotograma en el clúster de MKV resultante. El valor está especificado en incrementos de 100 ns, que es el estándar de SDK.
Longitud máxima de valor de escala de código temporal	1 segundo	El valor de escala máximo de código temporal para representar las marcas temporales del fotograma en el clúster de MKV resultante. El valor está especificado en incrementos de 100 ns, que es el estándar de SDK.

Valor	Límite	Notas
Recuento mínimo de elemento de vista de contenido	10	
Duración mínima de búfer	20 segundos	El valor está especificado en incrementos de 100 ns, que es el estándar de SDK.
Longitud máxima de versión de actualización	128 caracteres	
Longitud máxima de ARN	1024 caracteres	
Longitud máxima de secuencia de fragmento	128 caracteres	
Periodo de retención máximo	10 años	

Configurar una cuenta de

Antes de utilizar Amazon Kinesis Video Streams por primera vez, lleve a cabo las siguientes tareas.

Temas

- Inscríbase en una Cuenta de AWS
- Creación de un usuario con acceso administrativo
- Cree una clave Cuenta de AWS

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Cree una clave Cuenta de AWS

Necesitará una Cuenta de AWS clave para acceder a Amazon Kinesis Video Streams mediante programación.

Para crear una Cuenta de AWS clave, haga lo siguiente:

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. Elija Usuarios en la barra de navegación y, a continuación, elija el usuario Administrador.
3. Elija la pestaña Security credentials (Credenciales de seguridad) y, a continuación, elija Create access key (Crear clave de acceso).

4. Anote el valor de Access key ID (ID de clave de acceso). Elija Show (Mostrar) bajo Secret access key (Clave de acceso secreta). Anote el valor de Secret access key (Clave de acceso secreta).

Ventajas

Entre las ventajas de utilizar Kinesis Video Streams se incluyen las siguientes:

- Conéctese y transmita desde millones de dispositivos: puede usar Kinesis Video Streams para conectar y transmitir vídeo, audio y otros datos desde millones de dispositivos, desde teléfonos inteligentes de consumo, drones y cámaras de salpicadero. Puede utilizar las bibliotecas de productores de Kinesis Video Streams para configurar sus dispositivos y transmitir de forma fiable en tiempo real o after-the-fact como cargas multimedia.
- Almacene, cifre e indexe datos de forma duradera: puede configurar su transmisión de vídeo de Kinesis para almacenar datos multimedia de forma duradera durante períodos de retención personalizados. Kinesis Video Streams también genera un índice de los datos almacenados en función de las marcas horarias generadas por el productor o por el lado del servicio. Sus aplicaciones pueden recuperar datos específicos de una transmisión mediante el índice de tiempo.
- Céntrese en administrar las aplicaciones en lugar de la infraestructura: Kinesis Video Streams no tiene servidor, por lo que no es necesario configurar ni administrar ninguna infraestructura. No tiene que preocuparse por el despliegue, la configuración o la escalabilidad elástica de la infraestructura subyacente, ya que sus flujos de datos y la cantidad de aplicaciones que consumen más recursos aumentan o se reducen. Kinesis Video Streams realiza automáticamente toda la administración y el mantenimiento necesarios para gestionar las transmisiones, de modo que usted puede centrarse en las aplicaciones y no en la infraestructura.
- Cree aplicaciones en tiempo real y por lotes a partir de flujos de datos: puede utilizar Kinesis Video Streams para crear aplicaciones personalizadas en tiempo real que funcionen en transmisiones de datos en directo y crear aplicaciones por lotes o únicas que funcionen con datos de persistencia duradera sin requisitos estrictos de latencia. Puede crear, implementar y administrar aplicaciones personalizadas: soluciones de código abierto (Apache MXNet, OpenCV), propias o de terceros que las utilizan para procesar y analizar sus AWS Marketplace transmisiones. Puede usar Kinesis Video `Get` APIs Streams para crear varias aplicaciones simultáneas que procesen datos en tiempo real o por lotes.
- Transmita datos de forma más segura: Kinesis Video Streams cifra todos los datos a medida que fluyen por el servicio y cuando los conservan. Kinesis Video Streams aplica el cifrado basado en la seguridad de la capa de transporte (TLS) en la transmisión de datos desde los dispositivos y cifra

todos los datos en reposo mediante (). AWS Key Management Service AWS KMS Además, puede administrar el acceso a sus datos mediante AWS Identity and Access Management (IAM).

- Pague por uso: para obtener más información, consulte Calculadora de precios de AWS.

Introducción a Amazon Kinesis Video Streams

En esta sección se describe cómo realizar las siguientes tareas en Amazon Kinesis Video Streams:

- Configure Cuenta de AWS y cree un administrador, si aún no lo ha hecho.
- Crea una transmisión de vídeo de Kinesis.
- Envíe datos a la transmisión de vídeo de Kinesis desde su cámara y visualice el contenido multimedia en la consola.

Si es la primera vez que utiliza Amazon Kinesis Video Streams, le recomendamos que Kinesis Video Streams: cómo funciona lea primero.

> **Note**
>
> Si sigue el ejemplo de introducción, no se le cobrará ningún cargo. Cuenta de AWS Para conocer los costes de datos en su región, consulte los precios de Amazon Kinesis Video Streams.

Temas

- Creación de una transmisión de vídeo de Amazon Kinesis
- Enviar datos a una transmisión de vídeo de Amazon Kinesis
- Consume datos multimedia

Creación de una transmisión de vídeo de Amazon Kinesis

En esta sección se describe cómo crear una transmisión de vídeo de Kinesis.

Esta sección contiene los siguientes procedimientos:

- the section called “Cree una transmisión de vídeo mediante la consola”
- the section called “Cree una transmisión de vídeo mediante el AWS CLI”

Cree una transmisión de vídeo mediante la consola

1. Abre la consola enhttps://console.aws.amazon.com/kinesisvideo/home.
2. En la página Video streams (Transmisiones de vídeo), elija Create video stream (Crear transmisión de vídeo).
3. En la página Crear una nueva transmisión de vídeo, introduzca *YourStreamName* el nombre de la transmisión. Deje seleccionado el botón de configuración predeterminada.
4. Elija Crear transmisión de vídeo.
5. Una vez que Amazon Kinesis Video Streams haya creado la transmisión, revise los detalles de YourStreamNamela página.

Cree una transmisión de vídeo mediante el AWS CLI

1. Compruebe que lo tiene AWS CLI instalado y configurado. Para obtener más información, consulte la documentación de AWS Command Line Interface.
2. Ejecute el comando `Create-Stream` siguiente en la AWS CLI:

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

La respuesta tendrá un aspecto similar al siguiente:

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

Enviar datos a una transmisión de vídeo de Amazon Kinesis

En esta sección se describe cómo enviar datos multimedia desde una cámara a la transmisión de vídeo de Kinesis que creó en la sección anterior. En esta sección se usa la Utilice la biblioteca de productores de C++ como complemento de Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink.

Para enviar contenido multimedia desde diversos dispositivos y sistemas operativos, este tutorial utiliza la biblioteca de productores de C++ de Kinesis Video Streams GStreamery un marco multimedia de código abierto que estandariza el acceso a las cámaras y otras fuentes multimedia.

Temas

- Cree el SDK y los ejemplos
- Ejecute las muestras para cargar contenido multimedia en Kinesis Video Streams
- Revise los objetos de reconocimiento

Cree el SDK y los ejemplos

Puede crear el SDK y las muestras en su ordenador o en AWS Cloud9. Siga los procedimientos correspondientes que se indican a continuación.

Build on your computer

Siga las instrucciones del archivo readme para crear la biblioteca del productor y la aplicación de ejemplo.

Esto incluye:

- Instalación de dependencias
- Clonar el repositorio
- Se utiliza CMakepara generar archivos MAKE
- Construir los archivos binarios usando make

Build in AWS Cloud9

Siga estos procedimientos para cargar en Kinesis Video Streams AWS Cloud9 en. No necesitará descargar nada a su ordenador.

1. En el AWS Management Console, abierto AWS Cloud9.

   Seleccione Crear entorno.
2. En la pantalla Crear entorno, complete lo siguiente:

   - Nombre: escriba un nombre para el nuevo entorno.
   - Plataforma: seleccione Ubuntu Server 22.04 LTS.

Puede dejar los demás campos con las selecciones predeterminadas.

3. Cuando se haya creado el entorno, seleccione Abrir en la columna Cloud9 IDE.

   En la zona media inferior de la pantalla, verá. `Admin:~/environment $` Esta es la terminal AWS Cloud9 (Amazon EC2).

   > **Note**
   >
   > Si cierra accidentalmente la terminal, seleccione Ventana, Nueva terminal.

   Ejecute los siguientes comandos en el terminal para cambiar el volumen a 20 GiB.

   a. Descargue el script .

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

   b. Otorgue permisos de ejecución al script.

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

   c. Ejecute el script.

   ```
   ./resize_volume.sh
   ```

4. Obtenga la información más reciente sobre todo el software que puede instalar o actualizar mediante la herramienta avanzada de empaquetado (APT).

   Este comando no actualiza el software en sí, pero garantiza que el sistema sepa cuáles son las últimas versiones disponibles.

   ```
   sudo apt-get update
   ```

5. Instale las dependencias del SDK del productor de 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 para clonar el SDK del productor de C++.

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

7. Prepara un directorio de compilación.

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

8. Se usa CMake para generar archivos MAKE.

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

El final del resultado esperado tiene el siguiente aspecto:

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

9. Utilice make para compilar el SDK y las aplicaciones de muestra, así como para crear los ejecutables finales.

```
make
```

El final del resultado esperado tiene el siguiente aspecto:

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

10. Confirme que se hayan creado los archivos de muestra. Enumere los archivos del directorio actual:

```
ls
```

Confirme que estén presentes los siguientes archivos:

- kvs_gstreamer_sample
- libgstkvssink.so

11. (Opcional) Puede añadir la configuración de la variable de entorno GST_PLUGIN_PATH al script de inicio del shell. Esto garantiza que GST_PLUGIN_PATH esté configurado correctamente durante una nueva sesión de terminal. En AWS Cloud9, el script de inicio del shell es:. `~/.bashrc`

    Ejecute el siguiente comando para agregar el comando al final del script de inicio del shell.

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

    Escriba lo siguiente para ejecutar el script de inicio del shell:

    ```
    source ~/.bashrc
    ```

    Confirme que GST_PLUGIN_PATH esté configurado.

    ```
    echo $GST_PLUGIN_PATH
    ```

    Si configuras la salida correctamente, verás la siguiente salida. Si la salida está en blanco, la variable de entorno no está configurada correctamente.

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

Ejecute las muestras para cargar contenido multimedia en Kinesis Video Streams

La aplicación de muestra no admite las credenciales de IMDS. En tu terminal, exporta AWS las credenciales de tu usuario o rol de IAM y de la región en la que se encuentra tu transmisión.

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

Si utilizas AWS credenciales temporales, exporta también tu token de sesión:

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Descargue un ejemplo de vídeo en formato MP4 para subirlo a Kinesis Video Streams.

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

Especificaciones del vídeo:

- Resolución: 1280 x 720 píxeles
- Velocidad de fotogramas: 30 fotogramas por segundo
- Duración: 14,0 segundos
- Codificación de vídeo: H.264, en la pista 1
- Fotogramas clave: cada 3 segundos, lo que da como resultado una duración del fragmento (también conocida como tamaño de grupo de imágenes (GoP)) de 3 segundos, y el fragmento final dura 2 segundos.

Ejecute el siguiente comando con el nombre de la secuencia que creó anteriormente. Si aún no ha creado una transmisión, consultethe section called “Creación de una transmisión de vídeo de Amazon Kinesis”.

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

Sample video from GStreamer

Utilice el siguiente comando para generar un vídeo utilizando GStreamer.

Indique GStreamer dónde ubicar el `kvssink` GStreamer complemento. En tu directorio de compilación, especifica la ruta a la carpeta que contiene el `libgstkvssink.so` archivo.

Desde tu directorio de compilación, ejecuta el siguiente comando:

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

Esta GStreamer canalización genera una transmisión de vídeo de prueba en directo con un patrón de prueba estándar que se ejecuta a 10 fotogramas por segundo con una resolución

de 640 x 480 píxeles. Se añade una superposición que muestra la fecha y hora actuales del sistema. A continuación, el vídeo se codifica en formato H.264 y los fotogramas clave se generan como máximo cada 10 fotogramas, lo que da como resultado una duración del fragmento (también conocida como tamaño de grupo de imágenes (GoP)) de 1 segundo. `kvssink`toma la transmisión de vídeo codificada en H.264, la empaqueta en el formato contenedor Matroska (MKV) y la carga en la transmisión de vídeo de Kinesis.

Ejecuta el siguiente comando:

```
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"
```

Para detener la GStreamer canalización, seleccione la ventana del terminal y pulse CTRL+C.

> **Note**
>
> Para obtener más información sobre el uso del GStreamer complemento para transmitir vídeo desde una transmisión RTSP desde una cámara o desde una cámara USB, consulte. Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink

Revise los objetos de reconocimiento

Durante la carga, Kinesis Video Streams devolverá los objetos de confirmación al cliente que realiza la carga. Debería verlos impresos en el resultado del comando. Un ejemplo tiene el siguiente aspecto:

```
{"EventType":"PERSISTED","FragmentTimecode":1711124585823,"FragmentNumber":"12345678901234567890
```

Si el acuse de recibo lo `EventType` es`PERSISTED`, significa que Kinesis Video Streams ha almacenado y cifrado de forma duradera este fragmento de contenido multimedia para su recuperación, análisis y almacenamiento a largo plazo.

Para obtener más información sobre los reconocimientos, consulte. the section called “PutMedia”

Consume datos multimedia

Puede consumir datos multimedia visualizándolos en la consola o creando una aplicación que lea los datos multimedia de una transmisión mediante la transmisión en directo con hipertexto (HLS).

Vea contenido multimedia en la consola

En otra pestaña del navegador, abra AWS Management Console. En el panel de Kinesis Video Streams, seleccione Secuencias de vídeo.

Seleccione el nombre de la transmisión en la lista de transmisiones. Usa la barra de búsqueda, si es necesario.

Amplíe la sección de reproducción multimedia. Si el vídeo sigue cargándose, se mostrará. Si la carga ha finalizado, selecciona la flecha doble a la izquierda.

Consume datos multimedia mediante HLS

Puede crear una aplicación cliente que consuma datos de una transmisión de vídeo de Kinesis mediante HLS. Para obtener más información acerca de cómo crear una aplicación que consuma datos multimedia con HLS, consulte Reproducción de vídeo.

Cargar a Kinesis Video Streams

Las bibliotecas para productores de Amazon Kinesis Video Streams son un conjunto de bibliotecas del SDK para productores de Kinesis Video Streams. El cliente usa las bibliotecas y el SDK para crear la aplicación en el dispositivo para conectarse de forma segura a Kinesis Video Streams y transmitir datos multimedia para verlos en la consola o en las aplicaciones cliente en tiempo real.

Se puede hacer streaming de los datos de medios de las siguientes maneras:

- En tiempo real
- Después de almacenarlo en búfer durante unos segundos
- Después de que se carguen los archivos multimedia

Tras crear una transmisión de Kinesis Video Streams, puede empezar a enviarle datos. Puede usar el SDK para crear un código de aplicación que extraiga los datos de vídeo, conocidos como fotogramas, de la fuente multimedia y los cargue en Kinesis Video Streams. Estas aplicaciones también se denominan aplicaciones productoras.

Las bibliotecas de productores contienen los siguientes componentes:

- Cliente productor de Kinesis Video Streams
- Biblioteca de productores de Kinesis Video Streams

Cliente productor de Kinesis Video Streams

El cliente productor de Kinesis Video Streams incluye una `KinesisVideoClient` sola clase. Esta clase administra las fuentes de medios, recibe datos de las fuentes y administra el ciclo de vida de la transmisión a medida que los datos fluyen desde una fuente de medios a Kinesis Video Streams. También proporciona una `MediaSource` interfaz para definir la interacción entre Kinesis Video Streams y su hardware y software patentados.

Un origen de medios puede ser casi cualquier cosa. Por ejemplo, puede utilizar un origen de medios que sea una cámara o un micrófono. Los orígenes de medios no se limitan exclusivamente a fuentes de audio y vídeo. Por ejemplo, los logs de datos pueden ser archivos de texto, pero aun así pueden enviarse como flujos de datos. También puede disponer de varias cámaras en un teléfono que transmitan datos simultáneamente.

Para obtener datos de cualquiera de estas fuentes, puede implementar la interfaz `MediaSource`. Esta interfaz habilita situaciones adicionales para las que no facilitamos soporte integrado. Por ejemplo, puede optar por enviar lo siguiente a Kinesis Video Streams:

- Una transmisión de datos de diagnóstico (por ejemplo, registros y eventos de aplicaciones)
- Datos de cámaras de infrarrojos o cámaras de profundidad RADARs

Kinesis Video Streams no proporciona implementaciones integradas para dispositivos de producción multimedia, como cámaras. Para extraer datos de estos dispositivos, debe implementar código, por lo que estará creando su propia implementación de origen de medios personalizada. A continuación, puede registrar de forma explícita sus fuentes multimedia personalizadas con `KinesisVideoClient` las que cargar los datos en Kinesis Video Streams.

El cliente de producción de Kinesis Video Streams está disponible para aplicaciones Java y Android. Para obtener más información, consulte Utilice la biblioteca de productores de Java y Usa la biblioteca de productores de Android.

Biblioteca de productores de Kinesis Video Streams

La biblioteca de productores de Kinesis Video Streams se encuentra en el cliente de productores de Kinesis Video Streams. La biblioteca también está disponible para que la utilicen directamente aquellos que deseen una integración más profunda con Kinesis Video Streams. Permite la integración desde dispositivos con sistemas operativos propios, pilas de red o recursos en el dispositivo limitados.

La biblioteca de productores de Kinesis Video Streams implementa la máquina de estados para la transmisión a Kinesis Video Streams. Proporciona enlaces de devolución de llamada, que requieren que proporcione su propia implementación de transporte y administre de forma explícita cada mensaje entrante y saliente desde el servicio.

Puede optar por utilizar directamente la biblioteca de productores de Kinesis Video Streams por los siguientes motivos:

- El dispositivo en el que desea ejecutar la aplicación no cuenta con una máquina virtual Java.
- Quiere escribir el código de la aplicación en un lenguaje que no sea Java.
- Desea reducir la sobrecarga del código y limitarlo al mínimo nivel de abstracción, debido a limitaciones como la memoria y la potencia de procesamiento.

Actualmente, la biblioteca de productores de Kinesis Video Streams está disponible para aplicaciones Android, C, C++ y Java. Para obtener más información, consulte los idiomas compatibles en los siguientes temas relacionados.

Comprenda qué son las bibliotecas de productores

Utilice la biblioteca de productores de Java

Usa la biblioteca de productores de Android

Utilice la biblioteca de productores de C++

Utilice la biblioteca de productores de C

Utilice el SDK para productores de C++ en Raspberry Pi

Utilice la biblioteca de productores de Java

Puede utilizar la biblioteca Java Producer proporcionada por Amazon Kinesis Video Streams para escribir código de aplicación con una configuración mínima y enviar datos multimedia desde un dispositivo a una transmisión de vídeo de Kinesis.

Realice los siguientes pasos para integrar el código con Kinesis Video Streams para que su aplicación pueda empezar a transmitir datos a su transmisión de vídeo de Kinesis:

1. Crear una instancia del objeto `KinesisVideoClient`.
2. Puede crear un objeto `MediaSource` proporcionando información sobre el origen de medios. Por ejemplo, al crear un origen de medios de cámara, puede proporcionar información como, por ejemplo, la identificación de la cámara y la especificación del método de cifrado que utilice.

   Cuando quiera iniciar el streaming, debe crear un origen de medios personalizado.
3. Registre el origen multimedia con `KinesisVideoClient`.

   Después de registrar el origen de medios con `KinesisVideoClient`, siempre que los datos estén disponibles en el origen de medios, llama a `KinesisVideoClient` con los datos.

Procedimiento: utilice el SDK para productores de Java

Este procedimiento muestra cómo utilizar el cliente productor Java de Kinesis Video Streams en su aplicación Java para enviar datos a su transmisión de vídeo de Kinesis.

Estos pasos no requieren que tenga un origen de medios, como, por ejemplo, una cámara o micrófono. Para los fines de prueba, el código genera fotogramas de ejemplo consistentes en una serie de bytes. Puede utilizar el mismo patrón de codificación al enviar datos de medios desde fuentes reales como, por ejemplo, cámaras y micrófonos.

El procedimiento incluye los pasos siguientes:

- Descargue y configure el código
- Escribe y examina el código
- Ejecute y verifique el código

Requisitos previos

Antes de configurar el SDK para productores de Java, asegúrese de cumplir los siguientes requisitos previos:

- En el código de ejemplo, proporciona las credenciales especificando un perfil que ha configurado en su archivo de perfil de AWS credenciales. Si aún no lo ha hecho, configure en primer lugar su perfil de credenciales. Para obtener más información, consulte Configurar AWS credenciales y regiones para el desarrollo en AWS SDK para Java.

  > **Note**
  >
  > El ejemplo de Java usa un `SystemPropertiesCredentialsProvider` objeto para obtener sus credenciales. El proveedor recupera estas credenciales de las propiedades del sistema Java `aws.accessKeyId` y `aws.secretKey`. Puede definir estas propiedades del sistema en su entorno de desarrollo de Java. Para obtener información sobre cómo configurar las propiedades del sistema de Java, consulte la documentación de su entorno de desarrollo integrado (IDE) específico.

- `NativeLibraryPath`Debe contener el `KinesisVideoProducerJNI` archivo, disponible en https://github.com/awslabs/amazon-kinesis-video-streams- producer-sdk-cpp. La extensión de este archivo dependerá de su sistema operativo:

- KinesisVideoProducerJNI.so para Linux
- KinesisVideoProducerJNI.dylib para macOS
- KinesisVideoProducerJNI.dll para Windows

> **Note**
>
> Las bibliotecas prediseñadas para macOS, Ubuntu, Windows y Raspbian están disponibles `src/main/resources/lib` en https://github.com/awslabs/amazon-kinesis-video-streams- producer-sdk-java .git. Para otros entornos, compile la C++.

Descarga y configura el código de la biblioteca del productor de Java

En esta sección del procedimiento de la biblioteca del productor de Java, se descarga el código de ejemplo de Java, se importa el proyecto a su IDE de Java y se configuran las ubicaciones de las bibliotecas.

Para ver los requisitos previos y otros detalles sobre este ejemplo, consulte Uso de la biblioteca del productor de Java.

1. Cree un directorio y, a continuación, clone el código fuente del ejemplo desde el GitHub repositorio.

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

2. Abra el entorno de desarrollo integrado (IDE) de Java que utilice (por ejemplo, Eclipse o JetBrains IntelliJ IDEA) e importe el proyecto Apache Maven que ha descargado:

   - En IntelliJ IDEA: seleccione Import. Vaya al archivo `pom.xml` en la raíz del paquete descargado.
   - En Eclipse: seleccione File, Import, Maven, Existing Maven Projects. A continuación, vaya al directorio `kinesis-video-java-demo`.

   Para obtener más información, consulte la documentación de su IDE.

3. El código de ejemplo de Java usa las credenciales actuales. AWS Para utilizar un perfil de credenciales diferente, localice el siguiente código en `DemoAppMain.java`:

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

Cambie ese código a lo siguiente:

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

Para obtener más información consulte ProfileCredentialsProvider en la Referencia AWS SDK para Java.

Escribe y examina el código

En esta sección del procedimiento de la biblioteca productora de Java, escribirá y examinará el código de ejemplo de Java que descargó en la sección anterior.

La aplicación de prueba de Java (`DemoAppMain`) muestra el siguiente patrón de codificación:

- Cree una instancia de `KinesisVideoClient`.
- Cree una instancia de `MediaSource`.
- Registre el `MediaSource` con el cliente.
- Inicie el streaming. Inicie `MediaSource` y empezará a enviar datos al cliente.

En las siguientes secciones presentamos más detalles.

Crea una instancia de KinesisVideoClient

Puede crear el objeto `KinesisVideoClient` llamando a la operación `createKinesisVideoClient`.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
```

```
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

Para que `KinesisVideoClient` realice llamadas de red, necesita credenciales de autenticación. Puede transferir en una instancia de `SystemPropertiesCredentialsProvider`, que lee `AWSCredentials` para el perfil predeterminado en el archivo de credenciales:

```
[default]
aws_access_key_id = ABCDEFGHIJKLMOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

Crea una instancia de MediaSource

Para enviar bytes a la transmisión de vídeo de Kinesis, debe generar los datos. Amazon Kinesis Video Streams proporciona `MediaSource` la interfaz, que representa la fuente de datos.

Por ejemplo, la biblioteca Java de Kinesis Video Streams proporciona `ImageFileMediaSource` la implementación de `MediaSource` la interfaz. Esta clase solo lee los datos de una serie de archivos multimedia en lugar de una transmisión de vídeo de Kinesis, pero puede utilizarla para probar el código.

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

Regístrese en MediaSource el cliente

Registre el origen de medios que haya creado con el `KinesisVideoClient`, de modo que sea consciente de la existencia del cliente y pueda enviarle los datos.

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

Inicie la fuente multimedia

Inicie la fuente multimedia para que pueda empezar a generar datos y enviarlos al cliente.

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

Eliminar recursos

Para evitar pérdidas de memoria, haga lo siguiente para anular el registro de una fuente multimedia del cliente y liberar el cliente.

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

Si ha añadido algún elemento a la memoria caché mediante `CachedInfoMultiAuthServiceCallbacks`, por ejemplo:

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

Borra la caché cuando hayas terminado:

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

Ejecuta y verifica el código

Para ejecutar el arnés de pruebas de Java para la biblioteca productora de Java, haga lo siguiente.

1. Elija DemoAppMain.
2. Elija Ejecutar, Ejecutar 'DemoAppMain'.
3. Añada sus credenciales a los argumentos de JVM de la aplicación:
   - Para AWS credenciales no temporales: `"-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"`
   - Para AWS credenciales temporales: `"-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"`

4. Inicie sesión en la consola de Kinesis Video Streams AWS Management Console y ábrala.

   En la página Manage Streams, elija su transmisión.
5. El vídeo de ejemplo se reproducirá en el reproductor incrustado. Es posible que necesite esperar un poco (hasta diez segundos en condiciones típicas de ancho de banda y procesador) mientras se acumulan los fotogramas hasta que aparece el vídeo.

El código de ejemplo crea una transmisión. Cuando se inicia el `MediaSource` en el código, comienza a enviar fotogramas de muestra al `KinesisVideoClient`. A continuación, el cliente envía los datos a la transmisión de vídeo de Kinesis.

Usa la biblioteca de productores de Android

Puede utilizar la biblioteca de productores de Android proporcionada por Amazon Kinesis Video Streams para escribir código de aplicación, con una configuración mínima, para enviar datos multimedia desde un dispositivo Android a una transmisión de vídeo de Kinesis.

Realice los siguientes pasos para integrar el código con Kinesis Video Streams para que su aplicación pueda empezar a transmitir datos a su transmisión de vídeo de Kinesis:

1. Crear una instancia del objeto `KinesisVideoClient`.
2. Puede crear un objeto `MediaSource` proporcionando información sobre el origen de medios. Por ejemplo, al crear un origen de medios de cámara, puede proporcionar información como, por ejemplo, la identificación de la cámara y la especificación del método de cifrado que utilice.

   Cuando quiera iniciar el streaming, debe crear un origen de medios personalizado.

Procedimiento: utilice el SDK para productores de Android

Este procedimiento muestra cómo utilizar el cliente de producción de Kinesis Video Streams para Android en su aplicación de Android para enviar datos a su transmisión de vídeo de Kinesis.

El procedimiento incluye los pasos siguientes:

- the section called “Requisitos previos”
- the section called “Descarga y configura el código”
- the section called “Examine el código”

- the section called “Ejecute y verifique el código”

Requisitos previos

Recomendamos Android Studio para examinar, editar y ejecutar el código de la aplicación. Recomendamos utilizar la última versión estable.

En el código de ejemplo, proporciona las credenciales de Amazon Cognito.

Siga estos procedimientos para configurar un grupo de usuarios y un grupo de identidades de Amazon Cognito.

- Configure un grupo de usuarios
- Configure un grupo de identidades

Configure un grupo de usuarios

Para configurar un grupo de usuarios

1. Inicie sesión en la consola de Amazon Cognito y compruebe que la región es correcta.
2. En la barra de navegación de la izquierda, elija Grupos de usuarios.
3. En la sección Grupos de usuarios, elija Crear grupo de usuarios.
4. Complete las siguientes secciones:

   a. Paso 1: Configurar la experiencia de inicio de sesión: en la sección de opciones de inicio de sesión del grupo de usuarios de Cognito, seleccione las opciones adecuadas.

      Seleccione Siguiente.

   b. Paso 2: Configure los requisitos de seguridad: seleccione las opciones adecuadas.

      Seleccione Siguiente.

   c. Paso 3: Configure la experiencia de registro: seleccione las opciones adecuadas.

      Seleccione Siguiente.

   d. Paso 4: Configurar la entrega de mensajes: selecciona las opciones adecuadas.

      En el campo de selección de roles de IAM, seleccione un rol existente o cree uno nuevo.

      Seleccione Siguiente.

e. Paso 5: Integre su aplicación: seleccione las opciones adecuadas.

   En el campo Cliente inicial de la aplicación, selecciona Cliente confidencial.

   Seleccione Siguiente.

f. Paso 6: Revisar y crear: revise las selecciones de las secciones anteriores y, a continuación, elija Crear grupo de usuarios.

5. En la página Grupos de usuarios, seleccione el grupo que acaba de crear.

   Copie el ID del grupo de usuarios y anótelo para más adelante. En el `awsconfiguration.json` archivo, esto es`CognitoUserPool.Default.PoolId`.

6. Selecciona la pestaña Integración de aplicaciones y ve al final de la página.
7. En la sección de la lista de clientes de aplicaciones, elige el nombre del cliente de aplicaciones que acabas de crear.

   Copia el ID de cliente y anótalo para más adelante. En el `awsconfiguration.json` archivo, esto es`CognitoUserPool.Default.AppClientId`.

8. Muestre el secreto del cliente y anótelo para más adelante. En el `awsconfiguration.json` archivo, esto es`CognitoUserPool.Default.AppClientSecret`.

Configure un grupo de identidades

Para configurar un grupo de identidades

1. Inicie sesión en la consola de Amazon Cognito y compruebe que la región es correcta.
2. En la barra de navegación de la izquierda, elija Identity Pools.
3. Elija Crear grupo de identidades.
4. Configure el grupo de identidades.

   a. Paso 1: Configurar la confianza en los grupos de identidades: complete las siguientes secciones:

      - Acceso de usuario: seleccione Acceso autenticado
      - Fuentes de identidad autenticadas: seleccione un grupo de usuarios de Amazon Cognito

      Seleccione Siguiente.

b. Paso 2: Configurar los permisos: en la sección Función autenticada, complete los siguientes campos:

   - Función de IAM: seleccione Crear una nueva función de IAM
   - Nombre del rol de IAM: introduzca un nombre y anótelo para un paso posterior.

   Seleccione Siguiente.

c. Paso 3: Conectar proveedores de identidad: en la sección de detalles del grupo de usuarios, complete los siguientes campos:

   - ID del grupo de usuarios: seleccione el grupo de usuarios que creó anteriormente.
   - ID de cliente de aplicación: seleccione el ID de cliente de aplicación que creó anteriormente.

   Seleccione Siguiente.

d. Paso 4: Configurar las propiedades: escriba un nombre en el campo Nombre del grupo de identidades.

   Seleccione Siguiente.

e. Paso 5: Revisar y crear: revise sus selecciones en cada una de las secciones y, a continuación, seleccione Crear grupo de identidades.

5. En la página Grupos de identidades, selecciona tu nuevo grupo de identidades.

   Copia el ID del grupo de identidades y anótalo para más adelante. En el `awsconfiguration.json` archivo, esto `esCredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Actualice los permisos del rol de IAM.

   a. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/

   b. En el menú de navegación de la izquierda, selecciona Roles.

   c. Busca y selecciona el rol que creaste arriba.

> **Note**
>
> Usa la barra de búsqueda, si es necesario.

d. Seleccione la política de permisos adjunta.

   Seleccione Editar.

e. Seleccione la pestaña JSON y sustituya la política por la siguiente:

   JSON

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

   Seleccione Siguiente.

f. Selecciona la casilla situada junto a Establecer esta nueva versión como predeterminada si aún no está seleccionada.

   Seleccione Guardar cambios.

Descarga y configura el código de la biblioteca del productor de Android

En esta sección del procedimiento de la biblioteca para productores de Android, descargas el código de ejemplo de Android y abres el proyecto en Android Studio.

Para ver los requisitos previos y otros detalles sobre este ejemplo, consulta Cómo usar la biblioteca de productores de Android.

1. Crea un directorio y, a continuación, clónalo AWS Mobile SDK for Android desde el GitHub repositorio.

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

2. Abra Android Studio.
3. En la pantalla de apertura, elija Open an existing Android Studio project.
4. Vaya al directorio `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` y elija OK.
5. Abra el archivo `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json`.

   En el `CredentialsProvider` nodo, proporcione el ID del grupo de identidades del procedimiento Para configurar un grupo de identidades de la sección Requisitos previos e indique el suyo Región de AWS (por ejemplo,**`us-west-2`**).

   En el `CognitoUserPool` nodo, introduzca el secreto del cliente de la aplicación, el ID del cliente de la aplicación y el ID del grupo del procedimiento Para configurar un grupo de usuarios de la sección Requisitos previos e introduzca el suyo Región de AWS (por ejemplo,**`us-west-2`**).
6. El archivo `awsconfiguration.json` será similar al siguiente:

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "us-west-2:01234567-89ab-cdef-0123-456789abcdef",
        "Region": "us-west-2"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "CognitoUserPool": {
    "Default": {
      "AppClientSecret": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmno",
```

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

7. Actualízalo `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` con tu región (en el siguiente ejemplo, se establece en 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;
```

Para obtener información sobre las Región de AWS constantes, consulta Regiones.

Examine el código

En esta sección del procedimiento de la biblioteca de productores de Android, examinarás el código de ejemplo.

La aplicación de prueba de Android (`AmazonKinesisVideoDemoApp`) muestra el siguiente patrón de codificación:

- Cree una instancia de `KinesisVideoClient`.
- Cree una instancia de `MediaSource`.
- Inicie el streaming. Inicie el `MediaSource` y empezará a enviar datos al cliente.

En las siguientes secciones presentamos más detalles.

Crea una instancia de KinesisVideoClient

Puede crear el objeto `KinesisVideoClient` llamando a la operación `createKinesisVideoClient`.

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(
```

```
                    getActivity(),
                    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,
                    KinesisVideoDemoApp.getCredentialsProvider());
```

Para que `KinesisVideoClient` realice llamadas de red, necesita credenciales de autenticación. Pasa una instancia de`AWSCredentialsProvider`, que lee sus credenciales de Amazon Cognito del `awsconfiguration.json` archivo que modificó en la sección anterior.

Cree una instancia de MediaSource

Para enviar bytes a la transmisión de vídeo de Kinesis, debe generar los datos. Amazon Kinesis Video Streams proporciona `MediaSource` la interfaz, que representa la fuente de datos.

Por ejemplo, la biblioteca de Android de Kinesis Video Streams proporciona `AndroidCameraMediaSource` la implementación de `MediaSource` la interfaz. Esta clase lee los datos de una de las cámaras del dispositivo.

En el siguiente ejemplo de código (del archivo `fragment/StreamConfigurationFragment.java`), se crea la configuración del origen multimedia:

```
private AndroidCameraMediaSourceConfiguration getCurrentConfiguration() {
return new AndroidCameraMediaSourceConfiguration(
        AndroidCameraMediaSourceConfiguration.builder()
                .withCameraId(mCamerasDropdown.getSelectedItem().getCameraId())

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

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

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

 mCamerasDropdown.getSelectedItem().isEndcoderHardwareAccelerated())
                .withFrameRate(FRAMERATE_20)
                .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
                .withEncodingBitRate(BITRATE_384_KBPS)
                .withCameraOrientation(-
mCamerasDropdown.getSelectedItem().getCameraOrientation())

 .withNalAdaptationFlags(StreamInfo.NalAdaptationFlags.NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS
                .withIsAbsoluteTimecode(false));
```

```
}
```

En el siguiente ejemplo de código (del archivo `fragment/StreamingFragment.java`), se crea el origen multimedia:

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

Inicie la fuente multimedia

Inicie el origen de medios para que pueda comenzar a generar los datos y enviarlos al cliente. El siguiente ejemplo de código procede del archivo `fragment/StreamingFragment.java`:

```
mCameraMediaSource.start();
```

Ejecute y verifique el código

Para ejecutar la aplicación de ejemplo de Android para la biblioteca de productores de Android, haga lo siguiente.

1. Realice la conexión a un dispositivo Android.
2. Elija Run, Run... y Edit configurations....
3. Seleccione el icono con el signo más (+) en la aplicación Android. En el campo Nombre, escriba **`AmazonKinesisVideoDemoApp`**. En el menú desplegable del módulo, selecciona AmazonKinesisVideoDemoApp. Seleccione Aceptar.
4. Elija Run, Run.
5. En la pantalla Select a Deployment Target, elija el dispositivo conectado y elija OK.
6. En la AWSKinesisVideoDemoAppaplicación del dispositivo, selecciona Crear nueva cuenta.
7. Escriba los valores de USERNAME, Password, Given name, Email address y Phone number; a continuación, elija Sign up.

> **Note**
>
> Estos valores tienen las siguientes restricciones:

- Contraseña: debe contener letras mayúsculas y minúsculas, números y caracteres especiales. Puede cambiar estas restricciones en la página del grupo de usuarios de la consola de Amazon Cognito.
- Dirección de correo electrónico: debe ser una dirección válida para que pueda recibir un código de confirmación.
- Número de teléfono: debe tener el formato **+*<Country code><Number>***; por ejemplo, **+12065551212**.

8. Introduzca el código que recibirá por correo electrónico y seleccione Confirmar. Seleccione Ok.
9. En la página siguiente, mantén los valores predeterminados y selecciona Transmitir.
10. Inicie sesión en la AWS Management Console consola Kinesis Video Streams en la región EE.UU. Oeste (Oregón) y ábrala.

    En la página Manage Streams, elija demo-stream.
11. El vídeo de streaming se reproduce en el reproductor incrustado. Es posible que necesite esperar un poco (hasta diez segundos en condiciones típicas de ancho de banda y procesador) mientras se acumulan los fotogramas hasta que aparece el vídeo.

> **Note**
>
> Si se produce la rotación de la pantalla del dispositivo (por ejemplo, de vertical a horizontal), la aplicación detiene el streaming de vídeo.

El código de ejemplo crea una transmisión. Cuando se inicia el `MediaSource` en el código, comienza a enviar fotogramas desde la cámara a `KinesisVideoClient`. A continuación, el cliente envía los datos a una transmisión de vídeo de Kinesis denominada demo-stream.

Utilice la biblioteca de productores de C++

Puede utilizar la biblioteca de productores de C++ proporcionada por Amazon Kinesis Video Streams para escribir código de aplicación para enviar datos multimedia desde un dispositivo a una transmisión de vídeo de Kinesis.

Modelo de objetos

La biblioteca C++ proporciona los siguientes objetos para gestionar el envío de datos a una transmisión de vídeo de Kinesis:

- KinesisVideoProducer: contiene información sobre su fuente multimedia y sus AWS credenciales, y mantiene las llamadas para informar sobre los eventos de Kinesis Video Streams.
- KinesisVideoStream: Representa la transmisión de vídeo de Kinesis. Contiene información sobre los parámetros de la transmisión de vídeo, como el nombre, el período de retención de datos y el tipo de contenido multimedia.

Coloca contenido multimedia en la transmisión

Puedes usar los métodos proporcionados por la biblioteca C++ (por ejemplo,`PutFrame`) para colocar datos en el `KinesisVideoStream` objeto. A continuación, la biblioteca administra el estado interno de los datos, lo que puede incluir las siguientes tareas:

- Realización de la autenticación.
- Observación de la latencia de red. Si la latencia es demasiado elevada, la biblioteca podría optar por anular fotogramas.
- Seguimiento del estado del streaming en curso.

Interfaces de devolución de llamada

Esta capa expone un conjunto de interfaces de devolución de llamada, lo que le permite comunicarse con la capa de la aplicación. Entre estas interfaces de devolución de llamada se incluyen las siguientes:

- Interfaz de llamadas de servicio (**`CallbackProvider`**): la biblioteca invoca los eventos obtenidos a través de esta interfaz cuando crea una transmisión, obtiene una descripción de la secuencia y elimina una transmisión.
- Interfaz preparada para el cliente o con poco espacio de almacenamiento (**`ClientCallbackProvider`**): la biblioteca invoca los eventos de esta interfaz cuando el cliente está preparado o cuando detecta que podría quedarse sin espacio de almacenamiento o memoria disponible.

- Interfaz de retrollamada para eventos de transmisión (**`StreamCallbackProvider`**): la biblioteca invoca eventos de esta interfaz cuando se producen eventos de transmisión, como cuando la transmisión entra en estado preparado, se pierden fotogramas o se producen errores de transmisión.

Kinesis Video Streams proporciona implementaciones predeterminadas para estas interfaces. También puede proporcionar su propia implementación personalizada, por ejemplo, si necesita una lógica de red personalizada o si desea exponer una condición de bajo almacenamiento en la interfaz de usuario.

Para obtener más información sobre las devoluciones de llamadas en las bibliotecas de los productores, consulte. Retrollamadas del SDK del productor

Procedimiento: utilice el SDK para productores de C++

Este procedimiento muestra cómo utilizar el cliente de Kinesis Video Streams y las fuentes multimedia en una aplicación de C++ para enviar datos a la transmisión de vídeo de Kinesis.

El procedimiento incluye los pasos siguientes:

Requisitos previos

Antes de configurar el SDK para productores de C++, asegúrese de cumplir los siguientes requisitos previos:

- Credenciales: en el código de ejemplo, las credenciales se proporcionan especificando un perfil que se configura en el archivo de perfil de AWS credenciales. Si aún no lo ha hecho, configure en primer lugar su perfil de credenciales.

  Para obtener más información, consulte Configurar AWS credenciales y regiones para el desarrollo.
- Integración del almacén de certificados: la biblioteca de productores de Kinesis Video Streams debe establecer una relación de confianza con el servicio al que llama. Esto se realiza mediante la validación de las autoridades de certificación (CAs) en el almacén de certificados público. En los modelos basados en Linux, este almacén se encuentra en el directorio `/etc/ssl/`.

  Descargue el certificado de la siguiente ubicación en el almacén de certificados:

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

- Instale las siguientes dependencias de compilación para macOS:
  - Autoconf 2.69 (licencia GPLv3 +/Autoconf: GNU GPL versión 3 o posterior)
  - CMake 3.7 o 3.8
  - Pkg-Config
  - xCode (macOS) / clang / gcc (xcode-select version 2347)
  - Kit de desarrollo de Java (JDK) (para la compilación de JNI de Java)
  - Lib-Pkg
- Instale las siguientes dependencias de compilación para Ubuntu:
  - Git: `sudo apt install git`
  - CMake: `sudo apt install cmake`
  - G++: `sudo apt install g++`
  - pkg-config: `sudo apt install pkg-config`
  - OpenJDK: `sudo apt install openjdk-8-jdk`

    > **Note**
    >
    > Esto solo es necesario si está creando la interfaz nativa de Java (JNI).

  - Establezca la variable de entorno `JAVA_HOME`: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Descargue y configure el código de la biblioteca del productor de C++

Para obtener información sobre cómo descargar y configurar la biblioteca de productores de C++, consulte Amazon Kinesis Video Streams CPP Producer GStreamer , Plugin y JNI.

Para ver los requisitos previos y obtener más información sobre este ejemplo, consulte. the section called “C++”

CMake argumentos

A continuación se muestra una tabla de referencia para los argumentos específicos del SDK CMake de C++ Producer. También puedes pasar las CMake opciones estándar a CMake .

> **Important**
>
> Todas son opcionales.

Banderas para incluir o excluir determinadas funciones

CMake argumento	Tipo	Predeterminado/a	Explicación
`BUILD_DEPENDENCIES`	Booleano	ON	Cree dependencias desde la fuente. De lo contrario, utilice las dependencias que ya están instaladas en el sistema. Si no se encuentra una de las dependencias requeridas, se devolverá un error.
`BUILD_GSTREAMER_PLUGIN`	Booleano	OFF	Crea el `kvssink` GStreamer complemento.
`BUILD_JNI`	Booleano	OFF	Crea la interfaz nativa de Java (JNI) para poder llamar a este código desde un entorno de ejecución de Java.
`ALIGNED_MEMORY_MODEL`	Booleano	OFF	Si las asignaciones de memoria deben alinearse con los límites de 8 bytes. Algunas arquitecturas no permiten el

CMake argumento	Tipo	Predeterminado/a	Explicación
			acceso a la memoria de forma no alineada.
`CONSTRAINED_DEVICE`	Booleano	OFF	Solo para dispositivos que no son de Windows. Cuando está activada, establece el tamaño de la pila de subprocesos en `0.5 MiB`. Necesario para las compilaciones de Alpine Linux. De lo contrario, se utiliza el sistema operativo predeterminado.
`BUILD_STATIC`	Booleano	OFF	Cree bibliotecas y ejecutables como compartidos (OFF) o estáticos (ON).
`ADD_MUCLIBC`	Booleano	OFF	Conéctese a uClibc en lugar de a la biblioteca C estándar, que es una bibliotec a estándar C más pequeña diseñada para sistemas embebidos.

CMake argumento	Tipo	Predeterminado/a	Explicación
`OPEN_SRC_INSTALL_PREFIX`	Cadena	.. /de código abierto/local	Ubicación para instalar las dependencias de código abierto, si se está compilando desde el código fuente.

Banderas para la compilación cruzada

> **Important**
>
> Configúrelas si las arquitecturas de la CPU de la máquina de destino y la del equipo anfitrión son diferentes.

CMake argumento	Tipo	Predeterminado/a	Explicación
`BUILD_LOG4CPLUS_HOST`	Cadena	""	Cree la `log4cplus` dependencia para la arquitectura de CPU especificada. Si no se establece, `log4cplus` detectará automáticamente y utilizará la arquitectura de CPU de la máquina host.
`BUILD_OPENSSL_PLATFORM`	Cadena	""	Cree la `OpenSSL` dependencia para la arquitectura de CPU especificada. Si no se establece, `OpenSSL`

CMake argumento	Tipo	Predeterminado/a	Explicación
			detectará automáticamente y utilizará la arquitectura de CPU de la máquina host.

Indicadores relacionados con las pruebas

CMake argumento	Tipo	Predeterminado/a	Explicación
`BUILD_TEST`	Booleano	OFF	Construya las pruebas unitarias y de integración. Para ejecutar todas las pruebas, ejecútelas `./tst/producerTest` desde el directorio de compilación. AWS Se necesitan credenciales para ejecutar las pruebas.
`CODE_COVERAGE`	Booleano	OFF	Solo está disponible para los compiladores GNU/Clang. Habilite la recopilación de cobertura de código con gcov y la generación de informes.
`COMPILER_WARNINGS`	Booleano	OFF	Solo disponible para compiladores GNU/Clang. Activa todas

CMake argumento	Tipo	Predeterminado/a	Explicación
			las advertencias del compilador.
`ADDRESS_SANITIZER`	Booleano	OFF	Solo disponible para compiladores GNU/Clang. AddressSanitizerConstruye con.
`MEMORY_SANITIZER`	Booleano	OFF	Solo disponible para compiladores GNU/Clang. MemorySanitizerConstruye con.
`THREAD_SANITIZER`	Booleano	OFF	Solo disponible para compiladores GNU/Clang. ThreadSanitizerConstruye con.
`UNDEFINED_BEHAVIOR_SANITIZER`	Booleano	OFF	Solo disponible para compiladores GNU/Clang. UndefinedBehaviorSanitizerConstruye con.

Para usar estos CMake argumentos, páselos como una lista de `-D`*`key`*`=`*`value`* pares separados por espacios siguiendo el `cmake ..` comando. Por ejemplo:

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

CMake buscará la cadena de herramientas del compilador siguiendo la variable. `$PATH` Antes de ejecutarlo CMake, configure las variables `CC` y de `CXX` entorno para configurar de forma explícita la cadena de herramientas que se utilizará en la compilación cruzada.

Escribe y examina el código

En esta sección[the section called “C++”](), se examina el código del arnés de pruebas de C++ (`tst/ProducerTestFixture.h`y de otros archivos). Descargó este código en la sección anterior.

El ejemplo de C++ independiente de la plataforma: muestra el siguiente patrón de código:

- Cree una instancia de `KinesisVideoProducer` para acceder a Kinesis Video Streams.
- Cree una instancia de `KinesisVideoStream`. Esto crea una transmisión de vídeo de Kinesis en su Cuenta de AWS archivo si aún no existe una transmisión con el mismo nombre.
- Llama a `putFrame` en la `KinesisVideoStream` para cada fotograma de datos, a medida que están disponibles, para enviarlos a la transmisión.

En las siguientes secciones se proporciona más información sobre este patrón de codificación.

Cree una instancia de KinesisVideoProducer

Puede crear el objeto `KinesisVideoProducer` llamando al método `KinesisVideoProducer::createSync`. En el siguiente ejemplo se crea el `KinesisVideoProducer` en el archivo `ProducerTestFixture.h`:

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

El método `createSync` usa los siguientes parámetros:

- Un objeto `DeviceInfoProvider`, que devuelve un objeto `DeviceInfo` que contiene información sobre el dispositivo o configuración de almacenamiento.

> **Note**
>
> Configure el tamaño del almacén de contenido mediante el parámetro `deviceInfo.storageInfo.storageSize`. Las transmisiones de contenido comparten el almacén de contenido. Para determinar los requisitos en cuanto a tamaño de

almacenamiento, multiplique el tamaño medio del fotograma por el número de fotogramas almacenados para obtener la duración máxima de todas las transmisiones. A continuación, multiplique el resultado por 1,2 para tener en cuenta la desfragmentación. Por ejemplo, suponga que su aplicación tiene la siguiente configuración:

- Tres transmisiones
- 3 minutos de duración máxima
- Cada transmisión es de 30 fotogramas por segundo (FPS)
- Cada uno de los fotogramas tiene un tamaño de 10 000 KB

Los requisitos de almacenamiento de contenido para esta aplicación son 3 (transmisiones) * 3 (minutos) * 60 (segundos en un minuto) * 10000 (kb) * 1,2 (margen de desfragmentación) = 194,4 Mb ~ 200 Mb.

- Un objeto `ClientCallbackProvider`, que devuelve indicadores de la función que informan de eventos específicos del cliente.
- Un objeto `StreamCallbackProvider`, que devuelve indicadores de función que se vuelven a llamar cuando se producen eventos específicos a la transmisión.
- Un `CredentialProvider` objeto que proporciona acceso a las variables de entorno de AWS credenciales.
- El Región de AWS («us-west-2"). El punto de enlace de servicio se determina a partir de la región.

Crea una instancia de KinesisVideoStream

Puede crear el objeto `KinesisVideoStream` llamando al método `KinesisVideoProducer::CreateStream` con un parámetro `StreamDefinition`. El ejemplo crea `KinesisVideoStream` en el archivo `ProducerTestFixture.h` con el tipo de pista como vídeo y con el ID de pista como 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));
```

El objeto `StreamDefinition` tiene los siguientes campos:

- Nombre de transmisión.
- Periodo de retención de datos.
- Etiquetas para la transmisión. Las aplicaciones consumidoras pueden usar estas etiquetas para encontrar la transmisión correcta o para obtener más información sobre la transmisión. Las etiquetas también se pueden ver en la AWS Management Console.
- AWS KMS clave de cifrado para la transmisión. Para obtener más información, consulte the section called “Protección de los datos”.
- Tipo de streaming. El único valor válido actualmente es `STREAMING_TYPE_REALTIME`.
- Tipo de contenido de medios.
- Latencia de medios. Este valor no se utiliza actualmente y debe establecerse en 0.
- Duración de la reproducción de cada fragmento.
- Escala de código temporal de medios.
- Si los medios utilizan fragmentación de fotogramas claves.
- Si los medios utilizan códigos de tiempo.
- Si los medios utilizan tiempos de fragmentos absolutos.

Añadir una pista de audio a la transmisión de vídeo de Kinesis

Puede añadir detalles de una pista de audio a una definición de flujo de pista de vídeo mediante el siguiente `addTrack` método`StreamDefinition`:

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

El `addTrack` método requiere los siguientes parámetros:

- ID de pista (igual que para el audio). Debe ser un valor único y distinto de cero.
- Nombre de pista definido por el usuario (por ejemplo, «audio» para la pista de audio).

- ID de códec para esta pista (por ejemplo, para la pista de audio «A_AAC»).
- Tipo de pista (por ejemplo, utilice el valor de enumeración de MKV_TRACK_INFO_TYPE_AUDIO para el audio).

Si tiene datos privados del códec para la pista de audio, puede pasarlos cuando llame a la función addTrack. También puedes enviar los datos privados del códec después de crear el objeto mientras llamas al método start. KinesisVideoStream KinesisVideoStream

Coloque un fotograma en la transmisión de vídeo de Kinesis

El contenido multimedia se coloca en la transmisión de vídeo de Kinesis mediante `KinesisVideoStream::putFrame` un `Frame` objeto que contiene el encabezado y los datos multimedia. En el ejemplo se llama a `putFrame` en el archivo `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**
>
> El ejemplo anterior del productor de C++ envía un búfer de datos de prueba. En una aplicación real, debe obtener el búfer y el tamaño del fotograma a partir de los datos del fotograma de un origen de medios (como una cámara).

El objeto `Frame` tiene los siguientes campos:

- Índice de fotograma. Este valor debe incrementarse de forma monotónica.
- Indicadores asociados con el fotograma. Por ejemplo, si el codificador se ha configurado para producir un fotograma clave, a este fotograma se le asignará el indicador `FRAME_FLAG_KEY_FRAME`.
- Marca de tiempo de decodificación.
- Marca de tiempo de presentación.
- Duración del fotograma (en unidades de 100 ns).
- Tamaño del fotograma en bytes.
- Datos de fotograma.

Para obtener más información acerca del formato de los fotogramas, consulte the section called "Modelo de datos".

Pon una KinesisVideoFrame en una pista específica de KinesisVideoStream

Puede usar la `PutFrameHelper` clase para colocar datos de fotogramas en una pista específica. En primer lugar, llama `getFrameDataBuffer` a para que el puntero apunte a uno de los búferes preasignados para rellenar los `KinesisVideoFrame` datos. Luego, puede llamar al `putFrameMultiTrack` para enviar el valor `KinesisVideoFrame` junto con el booleano para indicar el tipo de datos del marco. Use true si se trata de datos de vídeo o false si el fotograma contiene datos de audio. El `putFrameMultiTrack` método utiliza un mecanismo de cola para garantizar que los fragmentos MKV mantengan marcas de tiempo de fotogramas que aumentan monótonamente y que los dos fragmentos no se superpongan. Por ejemplo, la marca de tiempo MKV del primer fotograma de un fragmento siempre debe ser mayor que la marca de tiempo MKV del último fotograma del fragmento anterior.

El valor `PutFrameHelper` tiene los siguientes campos:

- Número máximo de fotogramas de audio en la cola.
- Número máximo de fotogramas de vídeo en la cola.
- Tamaño que se debe asignar a un único fotograma de audio.
- Tamaño que se debe asignar a un único fotograma de vídeo.

Acceda a las métricas y al registro de métricas

El SDK para productores de C++ incluye funciones para las métricas y el registro de métricas.

Puede utilizar las operaciones `getKinesisVideoMetrics` y las de la `getKinesisVideoStreamMetrics` API para recuperar información sobre Kinesis Video Streams y sus transmisiones activas.

El siguiente código procede del archivo `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);
```

El objeto `PClientMetrics` rellenado por `getKinesisVideoMetrics` contiene la siguiente información:

- contentStoreSize: El tamaño total en bytes del almacén de contenido (la memoria utilizada para almacenar los datos de streaming).
- contentStoreAvailableTamaño: memoria disponible en el almacén de contenido, en bytes.
- contentStoreAllocatedTamaño: la memoria asignada en el almacén de contenido.
- totalContentViewsTamaño: memoria total utilizada para la visualización del contenido. La vista de contenido es una serie de índices de información en el almacén de contenido.

- totalFrameRate: el número total de fotogramas por segundo en todas las transmisiones activas.
- totalTransferRate: El total de bits por segundo (bps) que se envían en todas las transmisiones.

El objeto `PStreamMetrics` rellenado por `getKinesisVideoStreamMetrics` contiene la siguiente información:

- currentViewDuration: la diferencia en unidades de 100 ns entre la cabecera de la visualización del contenido (cuando se codifican los fotogramas) y la posición actual (cuando los datos de los fotogramas se envían a Kinesis Video Streams).
- overallViewDuration: La diferencia en unidades de 100 ns entre la parte superior de la visualización de contenido (cuando los fotogramas están codificados) y la cola (cuando los fotogramas se vacían de la memoria, ya sea porque se supera el espacio total asignado para la visualización del contenido o porque se recibe un `PersistedAck` mensaje de Kinesis Video Streams y los fotogramas que se sabe que persisten se vacían).
- currentViewSize: el tamaño en bytes del contenido visualizado desde el encabezado (cuando los fotogramas están codificados) hasta la posición actual (cuando los fotogramas se envían a Kinesis Video Streams).
- overallViewSize: el tamaño total en bytes de la visualización del contenido.
- currentFrameRate: la última velocidad medida de la transmisión, en fotogramas por segundo.
- currentTransferRate: la última velocidad medida de la transmisión, en bytes por segundo.

Desglose

Si desea enviar los bytes restantes de un búfer y esperar a recibir una respuesta `ACK`, puede utilizar `stopSync`:

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

También puede llamar a `stop` para finalizar la transmisión:

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

Después de detener la secuencia, puede liberarla invocando la siguiente API:

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

Ejecute y verifique el código

Para ejecutar y verificar el código delthe section called “C++”, consulte las siguientes instrucciones específicas del sistema operativo:

- Linux
- macOS
- Windows
- Sistema operativo Raspberry Pi

Puedes monitorear el tráfico de tu transmisión viendo las métricas asociadas a tu transmisión en la CloudWatch consola de Amazon, como`PutMedia.IncomingBytes`.

Usa el SDK para productores de C++ como GStreamer complemento

GStreameres un popular marco multimedia utilizado por varias cámaras y fuentes de vídeo para crear canales multimedia personalizados mediante la combinación de complementos modulares. El complemento Kinesis Video GStreamer Streams agiliza la integración de su canal multimedia GStreamer existente con Kinesis Video Streams.

Para obtener información sobre el uso del SDK para productores de C++ como GStreamer complemento, consulte. Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink

Utilice el SDK del productor de C++ como GStreamer complemento en un contenedor de Docker

GStreameres un popular marco multimedia que utilizan varias cámaras y fuentes de vídeo para crear canales multimedia personalizados mediante la combinación de complementos modulares. El complemento Kinesis Video GStreamer Streams agiliza la integración de su canal multimedia GStreamer existente con Kinesis Video Streams.

Además, el uso de Docker para crear la GStreamer canalización estandariza el entorno operativo de Kinesis Video Streams, lo que agiliza la creación y la ejecución de la aplicación.

Para obtener información sobre el uso del SDK para productores de C++ como GStreamer complemento en un contenedor de Docker, consulte. Ejecute el GStreamer elemento en un contenedor de Docker

Utilice el registro con el SDK para productores de C++

El registro de las aplicaciones del SDK del productor de C++ se configura en el `kvs_log_configuration` archivo de la `kinesis-video-native-build` carpeta.

En el siguiente ejemplo, se muestra la primera línea del archivo de configuración predeterminado, que configura la aplicación para escribir las entradas de registro de nivel `DEBUG` en la AWS Management Console:

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

Puede configurar el nivel de registro en `INFO` para obtener un registro menos detallado.

Para configurar la aplicación para que escriba entradas de registro en un archivo de registro, actualice la primera línea del archivo de la siguiente manera:

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

Así se configura la aplicación para escribir entradas de registro en el archivo `kvs.log` de la carpeta `kinesis-video-native-build/log`.

Para cambiar la ubicación del archivo de registro, actualice la línea siguiente con la ruta nueva:

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

> **Note**
>
> Si se escribe el registro de nivel `DEBUG` en un archivo, este puede consumir el espacio de almacenamiento disponible en el dispositivo rápidamente.

Utilice la biblioteca de productores de C

Puede utilizar la biblioteca de productores en C proporcionada por Amazon Kinesis Video Streams para escribir código de aplicación para enviar datos multimedia desde un dispositivo a una transmisión de vídeo de Kinesis.

Modelo de objetos

La biblioteca de productores de Kinesis Video Streams C se basa en un componente común denominado Platform Independent Codebase (PIC), que está disponible GitHub en -pic/. https://github.com/awslabs/ amazon-kinesis-video-streams El PIC contiene una lógica empresarial independiente de la plataforma para los componentes fundamentales. La biblioteca de productores de Kinesis Video Streams C incluye el PIC con una capa adicional de API que permite realizar llamadas y eventos específicos de cada escenario y plataforma. La biblioteca de productores de Kinesis Video Streams C incluye los siguientes componentes basados en PIC:

- Proveedores de información de dispositivos: expone la `DeviceInfo` estructura que se puede suministrar directamente a la API PIC. Puede configurar un conjunto de proveedores, incluido un proveedor optimizado para escenarios de aplicación que puede optimizar el almacén de contenido en función del número y los tipos de transmisiones que gestiona la aplicación y de la cantidad de almacenamiento en búfer necesario configurado en función de la cantidad de RAM disponible.
- Proveedor de información de transmisión: expone la `StreamInfo` estructura que se puede suministrar directamente a la API PIC. Hay un conjunto de proveedores que son específicos para los tipos de aplicaciones y los tipos comunes de escenarios de streaming. Estos incluyen proveedores como vídeo, audio y audio y vídeo multipista. Cada uno de estos escenarios tiene valores predeterminados que puede personalizar de acuerdo con los requisitos de su aplicación.
- Proveedor de devolución de llamadas: expone la `ClientCallbacks` estructura que se puede suministrar directamente a la API PIC. Esto incluye un conjunto de proveedores de devolución de llamadas para redes (devoluciones de llamadas de API basadas en CURL), autorización (API de AWS credenciales) y reintentos de transmisión en caso de errores. La API del proveedor de llamadas requiere varios argumentos para configurarla, como la información y la información de autorización. Región de AWS Esto se hace mediante certificados de IoT o mediante AWS AccessKeyId SecretKey, o SessionToken. Puede mejorar el proveedor de devolución de llamada mediante devoluciones de llamada personalizadas si la aplicación requiere un procesamiento mayor de una devolución de llamada en particular para llevar a cabo determinada lógica específica de la aplicación.
- FrameOrderCoordinator— Ayuda a gestionar la sincronización de audio y vídeo en escenarios multipista. Tiene un comportamiento predeterminado, que puede personalizar para gestionar la lógica específica de su aplicación. También simplifica el empaquetado de metadatos del marco en la estructura PIC Frame antes de enviarlos a la API PIC de capa inferior. Para escenarios que no son multipista, este componente se utiliza como acceso directo a la API de PIC putFrame.

La C Library proporciona los siguientes objetos para administrar el envío de datos a una transmisión de vídeo de Kinesis:

- KinesisVideoClient— Contiene información sobre su dispositivo y mantiene las llamadas para informar sobre los eventos de Kinesis Video Streams.
- KinesisVideoStream— Representa información sobre los parámetros de la transmisión de vídeo, como el nombre, el período de retención de datos y el tipo de contenido multimedia.

Coloca contenido multimedia en la transmisión

Puede usar los métodos proporcionados por la biblioteca C (por ejemplo,`PutKinesisVideoFrame`) para colocar datos en el `KinesisVideoStream` objeto. A continuación, la biblioteca administra el estado interno de los datos, lo que puede incluir las siguientes tareas:

- Realización de la autenticación.
- Observación de la latencia de red. Si la latencia es demasiado elevada, la biblioteca podría optar por anular fotogramas.
- Seguimiento del estado del streaming en curso.

Procedimiento: utilice el SDK del productor de C

Este procedimiento muestra cómo utilizar el cliente de Kinesis Video Streams y las fuentes multimedia de una aplicación C para enviar fotogramas de vídeo codificados en H.264 a su transmisión de vídeo de Kinesis.

El procedimiento incluye los pasos siguientes:

- Descargue el código de la biblioteca del productor en C
- Escribe y examina el código
- Ejecute y verifique el código

Requisitos previos

Antes de configurar el SDK para productores de C, asegúrese de cumplir los siguientes requisitos previos:

- Credenciales: en el código de ejemplo, las credenciales se proporcionan especificando un perfil que se configura en el archivo de perfil de AWS credenciales. Si aún no lo ha hecho, configure en primer lugar su perfil de credenciales.

  Para obtener más información, consulte Configurar AWS credenciales y regiones para el desarrollo.
- Integración del almacén de certificados: la biblioteca de productores de Kinesis Video Streams debe establecer una relación de confianza con el servicio al que recurre. Esto se realiza mediante la validación de las autoridades de certificación (CAs) en el almacén de certificados público. En los modelos basados en Linux, este almacén se encuentra en el directorio `/etc/ssl/`.

  Descargue el certificado de la siguiente ubicación en el almacén de certificados:

  https://www.amazontrust.com/repository/SFSRootCAG2.pem
- Instale las siguientes dependencias de compilación para macOS:
  - Autoconf 2.69 (licencia GPLv3 +/Autoconf: GNU GPL versión 3 o posterior)
  - CMake 3.7 o 3.8
  - Pkg-Config
  - xCode (macOS) / clang / gcc (xcode-select version 2347)
  - Kit de desarrollo de Java (JDK) (para la compilación de JNI de Java)
  - Lib-Pkg
- Instale las siguientes dependencias de compilación para Ubuntu:
  - Git: `sudo apt install git`
  - CMake: `sudo apt install cmake`
  - G++: `sudo apt install g++`
  - pkg-config: `sudo apt install pkg-config`
  - OpenJDK: `sudo apt install openjdk-8-jdk`
  - Establezca la variable de entorno `JAVA_HOME`: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Descargue el código de la biblioteca del productor en C

En esta sección, se descargan las bibliotecas de bajo nivel. Para comprobar los requisitos previos y otros detalles de este ejemplo, consulte the section called “C++”.

1. Cree un directorio y, a continuación, clone el código fuente de ejemplo del GitHub repositorio.

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

> **Note**
>
> Si no ejecuta git clone con `--recursive`, ejecute `git submodule update --init` en el directorio `amazon-kinesis-video-streams-producer-c/open-source`. También debe instalar pkg-config y un CMake entorno de compilación. Para obtener más información, consulta -producer-c.gitREADME.md. https://github.com/awslabs/ amazon-kinesis-video-streams

2. Abra el código en el entorno de desarrollo integrado (IDE) de su elección (por ejemplo, Eclipse).

Escribe y examina el código

En esta sección, examinará el código de la aplicación de ejemplo `KvsVideoOnlyStreamingSample.c` en la `samples` carpeta del repositorio https://github.com/awslabs/amazon-kinesis-video-streams-producer-c. GitHub Descargó este código en el paso anterior. En este ejemplo se muestra cómo utilizar la biblioteca de productores de C para enviar fotogramas de vídeo codificados en H.264 de la `samples/h264SampleFrames` carpeta a la transmisión de vídeo de Kinesis.

Esta aplicación de ejemplo tiene tres secciones:

- Inicialización y configuración:
  - Inicialización y configuración de la canalización multimedia específica de la plataforma.
  - Inicializar y configurar KinesisVideoClient y KinesisVideoStream para la canalización, establecer las devoluciones de llamadas, integrar la autenticación específica para cada escenario, extraer y enviar los datos privados del códec y poner la transmisión en estado READY.
- Bucle principal:
  - Obtención del fotograma de la canalización multimedia con las marcas de tiempo y otras marcas.
  - Enviar el marco al. KinesisVideoStream

- Desglose:
  - Detener (sincronizar) KinesisVideoStream, liberar KinesisVideoStream, liberar KinesisVideoClient.

Esta aplicación de ejemplo lleva a cabo las siguientes tareas:

- Llama a la API `createDefaultDeviceInfo` para crear el objeto `deviceInfo` que contiene información sobre el dispositivo o la configuración de almacenamiento.

```
// 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;
```

- Llama a la API `createRealtimeVideoStreamInfoProvider` para crear el objeto `StreamInfo`.

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

- Llama a la `createDefaultCallbacksProviderWithAwsCredentials` API para crear el proveedor de llamadas predeterminado en función de las credenciales estáticas. AWS

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

- Llame a la `createKinesisVideoClient` API para crear el `KinesisVideoClient` objeto que contiene información sobre el almacenamiento de su dispositivo y mantiene las llamadas para informar sobre los eventos de Kinesis Video Streams.

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

- Llama a la API `createKinesisVideoStreamSync` para crear el objeto `KinesisVideoStream`.

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

- Configura un fotograma de ejemplo y llama a la API `PutKinesisVideoFrame` para enviar ese fotograma al objeto `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;
   }
```

- Desglose:

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

```
CHK_STATUS(freeKinesisVideoClient(&clientHandle));
```

Ejecute y verifique el código

Para ejecutar y comprobar el código dethe section called “C++”, haga lo siguiente:

1. Ejecuta los siguientes comandos para crear un `build` directorio en el SDK de C descargado y `cmake` ejecútalo desde él:

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

   Puede pasar las siguientes opciones a `cmake ..`

   - `-DBUILD_DEPENDENCIES`- si se deben crear bibliotecas dependientes a partir de la fuente.
   - `-DBUILD_TEST=TRUE`- construir pruebas unitarias y de integración. Puede ser útil para confirmar la compatibilidad con tu dispositivo.

     `./tst/webrtc_client_test`
   - `-DCODE_COVERAGE`- habilita los informes de cobertura.
   - `-DCOMPILER_WARNINGS`- activar todas las advertencias del compilador.
   - `-DADDRESS_SANITIZER`- construir con AddressSanitizer.
   - `-DMEMORY_SANITIZER`- construir con MemorySanitizer.
   - `-DTHREAD_SANITIZER`- construir con ThreadSanitizer.
   - `-DUNDEFINED_BEHAVIOR_SANITIZER`- construir con UndefinedBehaviorSanitizer.
   - `-DALIGNED_MEMORY_MODEL`: crear solo para dispositivos de modelo de memoria alineada. El valor predeterminado es `OFF`.
2. Navegue hasta el `build` directorio que acaba de crear con el paso anterior y ejecute `make` para crear el SDK WebRTC C y los ejemplos proporcionados.

   ```
   make
   ```
3. La aplicación de ejemplo `kinesis_video_cproducer_video_only_sample` envía los fotogramas de vídeo codificados en H.264 de la carpeta `samples/h264SampleFrames` a

Kinesis Video Streams. El siguiente comando envía los fotogramas de vídeo en bucle durante diez segundos a Kinesis Video Streams:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Si quiere enviar fotogramas codificados en H.264 desde otra carpeta (por ejemplo,`MyH264FramesFolder`), ejecute el ejemplo con los siguientes argumentos:

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

4. Para habilitar los logs detallados, defina las instrucciones "define" de C `HEAP_DEBUG` y `LOG_STREAMING` quitando la marca de comentario de las líneas de `CMakeList.txt` correspondientes.

Puede monitorear el progreso del conjunto de pruebas en la salida de depuración de su IDE. También puedes monitorear el tráfico de tu transmisión viendo las métricas asociadas a tu transmisión en la CloudWatch consola de Amazon, como`PutMedia.IncomingBytes`.

> **Note**
>
> Dado que la herramienta de ejecución de pruebas solo envía fotogramas de bytes vacíos, la consola no muestra los datos en forma de transmisión de vídeo.

Utilice el SDK para productores de C++ en Raspberry Pi

El Raspberry Pi es un equipo pequeño y económico que se puede utilizar para enseñar y aprender habilidades básicas de programación informática. En este tutorial se describe cómo configurar y utilizar el SDK para productores de C++ de Amazon Kinesis Video Streams en un dispositivo Raspberry Pi. Los pasos también incluyen cómo verificar la instalación mediante la aplicación de GStreamer demostración.

Temas

- Requisitos previos
- Cree un usuario de IAM con permiso para escribir en Kinesis Video Streams
- Conecte su Raspberry Pi a su red Wi-Fi
- Conéctate remotamente a tu Raspberry Pi

- Configura la cámara Raspberry Pi
- Requisitos previos de instalación del software
- Descargue y cree el SDK para productores de Kinesis Video Streams C++
- Transmita vídeo a su transmisión de vídeo de Kinesis
- Reproduzca contenido multimedia de su transmisión de vídeo de Kinesis
- Solución de problemas de compilación en el SDK para productores de C++ para Raspberry Pi

Requisitos previos

Antes de configurar el SDK para productores de C++ en su Raspberry Pi, asegúrese de cumplir los siguientes requisitos previos:

- Un dispositivo Raspberry Pi con la siguiente configuración:
  - Versión de placa: 3 Modelo B o posterior.
  - Un módulo de cámara conectado o una cámara USB (cámara web) conectada.
  - Una tarjeta SD con una capacidad de al menos 8 GB.
  - El sistema operativo Raspbian (versión de kernel 4.9 o posterior) instalado. Puedes descargar la imagen más reciente del sistema operativo Raspberry Pi (anteriormente llamado Raspbian) desde el sitio web de Raspberry Pi. Siga las instrucciones de Raspberry Pi para instalar la imagen descargada en una tarjeta SD.
- Y Cuenta de AWS con una transmisión de vídeo de Kinesis. Para obtener más información, consulte Introducción a secuencias de vídeo de Kinesis.

Cree un usuario de IAM con permiso para escribir en Kinesis Video Streams

Si aún no lo ha hecho, configure un usuario AWS Identity and Access Management (IAM) con permisos para escribir en una transmisión de vídeo de Kinesis.

Estos procedimientos están diseñados para ayudarle a empezar rápidamente a utilizar un key pair de claves de AWS acceso. Los dispositivos pueden usar certificados X.509 para conectarse AWS IoT. Consulte the section called “Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT” para obtener más información sobre cómo configurar el dispositivo para que utilice la autenticación basada en certificados.

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
2. En el menú de navegación de la izquierda, elija Users (Usuarios).
3. Para crear un nuevo usuario, seleccione Add user (Añadir usuario).
4. Proporcione un User name (Nombre de usuario) descriptivo como, por ejemplo, **`kinesis-video-raspberry-pi-producer`**.
5. En Access type (Tipo de acceso), seleccione Programmatic access (Acceso mediante programación).
6. Elija Siguiente: permisos.
7. En Establecer permisos para kinesis-video-raspberry-pi -producer, selecciona Adjuntar directamente las políticas existentes.
8. Elija Crear política. Se abre la página Create policy (Crear política) en una nueva pestaña del navegador web.
9. Seleccione la pestaña JSON.
10. Copie la siguiente política JSON y péguela en el área de texto. Esta política otorga a su usuario permiso para crear y escribir datos en las transmisiones de vídeo de Kinesis.

    JSON

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

11. Elija Revisar política.

12. Especifique un Name (Nombre) para la política como, por ejemplo, **`kinesis-video-stream-write-policy`**.
13. Elija Crear política.
14. Vuelva a la pestaña Add user (Añadir usuario) en su navegador y elija Refresh (Actualizar).
15. En el cuadro de búsqueda, escriba el nombre de la política que ha creado.
16. Seleccione la casilla de verificación situada junto a la nueva política en la lista.
17. Elija Siguiente: Revisar.
18. Seleccione la opción Crear usuario.
19. La consola muestra el Access key ID (ID de clave de acceso) del nuevo usuario. Elija Show (Mostrar) para mostrar la Secret access key (Clave de acceso secreta). Anote estos valores, ya que son necesarios a la hora de configurar la aplicación.

Conecte su Raspberry Pi a su red Wi-Fi

Si utiliza un monitor y teclado conectados, vaya a Configura la cámara Raspberry Pi.

Estas instrucciones están escritas para ayudarte a configurar tu Raspberry Pi cuando se ejecuta en modo headless, es decir, sin un teclado, monitor o cable de red conectados. Sigue las instrucciones que aparecen a continuación para configurar tu Raspberry Pi para que intente conectarse automáticamente a la red especificada, lo que permitirá que la máquina host pueda acceder a ella mediante SSH.

1. En el equipo, cree un archivo denominado `wpa_supplicant.conf`.
2. Copia el siguiente texto y pégalo en el `wpa_supplicant.conf` archivo:

```
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"
}
```

   Reemplace los valores `ssid` y `psk` por la información de su red wifi.

3. Copie el archivo `wpa_supplicant.conf` en la tarjeta SD. Se debe copiar en la raíz del volumen `boot`.
4. Inserte la tarjeta SD en el Raspberry Pi y encienda el dispositivo. Se conecta a su red wifi y SSH está habilitado.

Conéctate remotamente a tu Raspberry Pi

Si utiliza un monitor y teclado conectados, vaya a Configura la cámara Raspberry Pi.

Estas instrucciones están escritas para ayudarte a configurar tu Raspberry Pi cuando se ejecuta en modo headless, es decir, sin un teclado, monitor o cable de red conectados. Sigue las instrucciones que aparecen a continuación para localizar tu Raspberry Pi en la red y conectarla mediante SSH desde tu máquina host.

1. Antes de conectarse a su dispositivo Raspberry Pi de forma remota, realice una de las siguientes operaciones para determinar su dirección IP:
   - Si tiene acceso al router wifi de su red, mire los dispositivos wifi conectados. Busque el dispositivo denominado `Raspberry Pi` para encontrar la dirección IP del dispositivo.
   - Si no tiene acceso al router wifi de su red, puede utilizar otro software para buscar dispositivos en su red. Fing es una aplicación popular que está disponible tanto para dispositivos Android como iOS. Puede utilizar la versión gratuita de esta aplicación para buscar las direcciones IP de dispositivos en su red.
2. Cuando sepa la dirección IP del dispositivo Raspberry Pi, puede utilizar cualquier aplicación de terminal para conectarse.
   - En macOS o Linux, utilice `ssh`:

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

   - En Windows, utilice PuTTY, un cliente SSH gratuito para Windows.

   En una nueva instalación de Raspbian, el nombre de usuario es **`pi`** y la contraseña es **`raspberry`**. Le recomendamos que cambie la contraseña predeterminada.

Configura la cámara Raspberry Pi

Siga estos pasos para configurar el módulo de cámara Raspberry Pi para enviar vídeo desde el dispositivo a una transmisión de vídeo de Kinesis.

> **Note**
>
> Si utiliza una cámara web USB, vaya athe section called “Requisitos previos de instalación del software”.

Camera module 1

Siga estas instrucciones para actualizar el archivo de módulos, habilitar la interfaz de la cámara y verificar el funcionamiento de la cámara. La actualización del archivo de módulos indica a la Raspberry Pi qué módulos del núcleo debe cargar en el momento del arranque. El controlador de la cámara no se carga de forma predeterminada para conservar los recursos del sistema en los dispositivos Raspberry Pi que no utilizan una cámara.

1. Abre un editor para realizar cambios en el archivo del módulo. Abre el terminal y usa el siguiente comando para editar el archivo con el `nano` editor:

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

2. Añada la siguiente línea al final del archivo, si aún no se ha añadido:

   ```
   bcm2835-v4l2
   ```

3. Guarde el archivo y salga del editor. Para guardar y salir con el `nano` editor, utilice Ctrl+X.
4. Reinicie el Raspberry Pi:

   ```
   sudo reboot
   ```

5. Cuando el dispositivo se reinicie, vuelva a conectarse a través de su aplicación de terminal si se está conectando de forma remota.
6. Abrir `raspi-config`:

   ```
   sudo raspi-config
   ```

7. Elija Opciones de interfaz, Legacy Camera. En versiones anteriores del sistema operativo Raspbian, esta opción de menú podría estar en Opciones de interfaz, Cámara.

   Habilite la cámara si aún no está habilitada y reiníciela si se le solicita.

8. Compruebe que la cámara funciona escribiendo el siguiente comando:

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

   Si la cámara está configurada correctamente, este comando captura una imagen de la cámara, la guarda en un archivo denominado `test.jpg` y muestra mensajes informativos.

Camera module 2 or 3

Si utiliza un módulo de cámara 2, puede utilizar uno `bcm2835-v4l2` (antiguo) o uno `libcamera` (moderno). Sin embargo, se recomienda `libcamera` usar el paquete para mejorar el soporte y las funciones. Siga los pasos que se indican a continuación para confirmar que `libcamera` está up-to-date en su sistema.

1. libcamera debería venir preinstalado en tu Raspberry Pi. Comprueba si hay actualizaciones y actualiza a la última versión para ver las correcciones de errores y las actualizaciones de seguridad. Abre el terminal y escribe los siguientes comandos:

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

2. Reinicie el sistema para que las actualizaciones surtan efecto.

   ```
   sudo reboot
   ```

3. Pon a prueba tu cámara. Esta aplicación inicia una secuencia de previsualización de la cámara y la muestra en la pantalla.

   ```
   libcamera-hello
   ```

   Si tiene problemas con el módulo de la cámara, consulte la documentación de Raspberry Pi para solucionar el problema.

Requisitos previos de instalación del software

El SDK para productores de C++ requiere que instales los siguientes requisitos previos de software en Raspberry Pi.

1. Actualice la lista de paquetes e instale las bibliotecas necesarias para compilar el SDK. Abre la terminal y escribe los siguientes comandos:

```
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. Si estás usando la `libcamera` pila, instala también el `libcamerasrc` GStreamer complemento. Este GStreamer complemento no viene instalado de forma predeterminada.

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

3. Copie el siguiente archivo PEM en `/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
```

Descargue y cree el SDK para productores de Kinesis Video Streams C++

Siga los procedimientos que se indican a continuación para descargar y crear el SDK para productores de Kinesis Video Streams C++. Asegúrese de haber instalado los requisitos previos del software; consulte estos the section called "Requisitos previos de instalación del software" pasos.

1. Navegue hasta el directorio de descargas. Abre un terminal y cambia al directorio de descargas que prefieras.

   Por ejemplo:

   ```
   cd ~/Downloads
   ```

2. Clona el repositorio del SDK. Usa el `git clone` comando para descargar el SDK del GitHub repositorio. Escriba:

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

   Este comando clona solo una rama (la `master` rama), lo que reduce el tamaño y el tiempo de descarga. También coloca el contenido descargado en una carpeta llamada `kvs-producer-sdk-cpp` dentro del directorio actual.

3. Verifica la descarga. Una vez finalizado el proceso de clonación, enumere el contenido de la `kvs-producer-sdk-cpp` carpeta para comprobar que se ha descargado el SDK.

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

4. Prepara un directorio de compilación. Escriba:

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

5. Configure la compilación. Ejecute el siguiente `cmake` comando para configurar el entorno de compilación con opciones específicas:

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

   CMakeutiliza las siguientes opciones para generar las opciones adecuadas`Makefiles`:

- Utilizando la carpeta del proyecto (`..`) como directorio de origen.
- Usando el directorio actual (`.`) (`build/`) para el resultado de la compilación.
- `-DBUILD_GSTREAMER_PLUGIN=ON`permite la creación del GStreamer complemento kvssink.
- `-DBUILD_DEPENDENCIES=OFF`desactiva la creación de dependencias externas desde la fuente. El proyecto buscará y utilizará las dependencias externas instaladas en un paso anterior.
- `-DALIGNED_MEMORY_MODEL=ON`desactiva el modelo de memoria no alineado. Algunos dispositivos Raspberry Pi no admiten el acceso no alineado a la memoria.

> **Note**
>
> Para obtener una lista completa de CMake argumentos, consultethe section called "Descarga y configura el código".

6. Compilar el proyecto. Tras configurar la compilación, utilice el `make` comando para compilar utilizando el `Makefile` generado por`cmake`.

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

El `-j` argumento to `make` le permite ejecutar varios trabajos de compilación en paralelo. Para reducir los tiempos de compilación, usa el `nproc` comando para calcular dinámicamente el número de núcleos de CPU de tu Raspberry Pi.

7. Confirma que `libgstkvssink.so` está presente.

   Enumere los archivos del directorio actual.

   Mensaje:

```
ls
```

   Respuesta:

```
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. Confirme que GStreamer se puede cargar`kvssink`.

   Establezca la variable de `GST_PLUGIN_PATH` entorno en el directorio que la contiene`libgstkvssink.so`.

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

   Tenga GStreamer carga`kvssink`:

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

   Deberías ver alguna documentación al respecto`kvssink`. Usa las teclas de flecha para navegar y presiona q para salir.

9. (Opcional) Actualice el script de inicio del shell para incluir la configuración de la variable de `GST_PLUGIN_PATH` entorno. Esto garantiza que `GST_PLUGIN_PATH` esté configurado correctamente durante una nueva sesión de terminal. En los dispositivos Raspberry Pi, el script de inicio de la consola es`~/.bashrc`.

   Ejecute el siguiente comando para añadir el comando al final del script de inicio del shell.

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

   Escriba lo siguiente para ejecutar el script de inicio del shell o cierre el shell actual y abra uno nuevo.

   ```
   source ~/.bashrc
   ```

   Confirme `GST_PLUGIN_PATH` que está configurado y podrá cargar`kvssink`.

   ```
   echo $GST_PLUGIN_PATH
   ```

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

Transmita vídeo a su transmisión de vídeo de Kinesis

Para ejecutar la aplicación de ejemplo, necesita la siguiente información:

- El nombre de la secuencia que creó en la sección Requisitos previos.
- Las credenciales de cuenta (ID de clave de acceso y clave de acceso secreta) que creó en Cree un usuario de IAM con permiso para escribir en Kinesis Video Streams.
- GStreamer es capaz de localizar el `kvssink` complemento. Para obtener más información, consulte the section called "Descargue y cree el SDK para productores de C++".

1. Establece las credenciales y la región.

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

Para ver otros métodos de autenticación, consultethe section called "Proporcione las credenciales a `kvssink`".

> **Note**
>
> El SDK para productores de C++ usa la región EE.UU. Oeste (Oregón) (`us-west-2`) de forma predeterminada. Para usar la opción predeterminada, Región de AWS cree su transmisión de vídeo de Kinesis en la región EE.UU. Oeste (Oregón).
> Para usar una región diferente para la transmisión de vídeo de Kinesis, defina la siguiente variable de entorno en su región (por ejemplo,*`us-east-1`*):
>
> ```
> export AWS_DEFAULT_REGION=us-east-1
> ```

2. En función del medio de entrada, elija una de las siguientes opciones:

Sample GStreamer video

Esta GStreamer canalización genera una transmisión de vídeo de prueba en directo con un patrón de prueba estándar que se ejecuta a 10 fotogramas por segundo con una resolución de 640 x 480 píxeles. Se añade una superposición que muestra la fecha y hora actuales del sistema. A continuación, el vídeo se codifica en formato H.264 y los fotogramas clave

se generan como máximo cada 10 fotogramas, lo que da como resultado una duración del fragmento (también conocida como tamaño de grupo de imágenes (GoP)) de 1 segundo. kvssink toma la transmisión de vídeo codificada en H.264, la empaqueta en el formato contenedor Matroska (MKV) y la carga en la transmisión de vídeo de Kinesis.

Ejecuta el siguiente comando:

```
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"
```

Para detener la canalización, seleccione la ventana del terminal y pulse CTRL+C. GStreamer

El ejemplo de la GStreamer canalización de vídeo tiene el siguiente aspecto:

USB web cam

Ejecute el siguiente comando para que su cámara USB GStreamer detecte automáticamente:

```
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"
```

Para detener la GStreamer canalización, selecciona la ventana del terminal y pulsa CTRL+C.

En lugar de dejar que se GStreamer detecte automáticamente, puedes usarlo `v4l2src` con un identificador de dispositivo específico. Ejecuta el siguiente comando:

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

En el resultado, verás algunos dispositivos y el inicio de GStreamer un proceso sobre cómo usarlos:

```
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 ! ...
```

Para detener la GStreamer canalización, selecciona la ventana del terminal y pulsa CTRL+C.

Raspberry Pi camera module 1

Si utiliza el módulo de cámara Pi 1 o el módulo de cámara Pi 2 con`bcm2835-v4l2`, utilice lo siguiente:

```
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"
```

Para detener la GStreamer canalización, selecciona la ventana del terminal y pulsa CTRL+C.

Raspberry Pi camera module 2 or 3

Si utilizas la `libcamera` pila moderna, utiliza la siguiente GStreamer canalización:

```
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"
```

Para detener la GStreamer canalización, selecciona la ventana del terminal y pulsa CTRL+C.

Utilice hardware

Algunos modelos de Raspberry Pi vienen con codificadores H.264 acelerados por hardware. Puede usarlos en lugar de`x264enc`, que es un codificador de software.

1. Asegúrese de que los GStreamer complementos estén instalados:

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

2. Escriba:

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

   Determine si los siguientes elementos están disponibles:

   - omxh264enc
   - v4l2h264enc

   Si están disponibles, puede utilizarlos. Estos son algunos ejemplos de canalizaciones en las que se utilizan esos elementos:

   `omxh264enc`:

   ```
   gst-launch-1.0 v4l2src device=/dev/video0 \
     ! videoconvert \
     ! video/x-raw,format=I420,width=640,height=480 \
     ! omxh264enc control-rate=2 target-bitrate=512000 periodicity-idr=45 inline-
   header=FALSE \
   ```

```
  ! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
  ! kvssink stream-name="raspberry"
```

`v4l2h264enc`y`v4l2convert`:

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

Problemas de tiempo de ejecución

A continuación, se muestran algunos de los problemas de tiempo de ejecución más frecuentes y cómo solucionarlos.

No existe el elemento «xxxxxxxxx»

Si recibes un error como el siguiente, significa que te falta un GStreamer complemento:

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

Solución

En función del elemento que falte, determina la acción adecuada:

- `kvssink`: Consulte the section called “Descargue y cree el SDK para productores de C++”.
- `libcamerasrc`: Consulte the section called “Error «Falló la activación del grupo de búferes»” para instalar el `libcamerasrc` GStreamer elemento.
- `omxh264encov4l2h264enc`:

  Siga the section called “Requisitos previos de instalación del software” para instalar todas las GStreamer bibliotecas. Si las has instalado todas y estos elementos no aparecen, significa que tu Raspberry Pi no tiene el hardware necesario. En su `x264enc` lugar, usa el codificador de software.

- Otros: Siga estos pasos the section called “Requisitos previos de instalación del software” para instalar todas las GStreamer bibliotecas. Se encuentran diferentes GStreamer elementos en los distintos grupos de GStreamer complementos (buenos, malos, feos), así que asegúrate de instalarlos todos.

Error «Falló la activación del grupo de búferes»

Si recibes un error como el siguiente, significa que la canalización que se está utilizando se está utilizando`v4l2src`, pero debería usarla `libcamerasrc` en su lugar.

```
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)
```

Por ejemplo, si utiliza la siguiente canalización con el módulo de cámara 2 sin `libcamerasrc` instalar, GStreamer es posible que se produzca este error al intentar detectar automáticamente qué elementos utilizar.

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

Solución

Asegúrese de que `libcamerasrc` esté instalado y utilícelo como elemento fuente, en lugar de`v4l2src`. Escriba lo siguiente para instalar el `libcamerasrc` GStreamer elemento:

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

Una vez `libcamerasrc` instalado, si está utilizando el `autovideosrc` elemento, GStreamer debería cambiar automáticamente para utilizar la fuente correcta `libcamerasrc` en su lugar`v4l2src`.

Error de bus

Si recibes un error de bus poco después de empezar `kvssink` (normalmente, cuando `PutMedia` finaliza la llamada HTTP), significa que tu Raspberry Pi no admite el acceso no alineado a la memoria. Los registros tendrán el siguiente aspecto:

```
INFO Camera camera.cpp:1197 configuring streams: (0) 640x480-YUV420
INFO RPI pisp.cpp:1450 Sensor: /base/axi/pcie@120000/rp1/i2c@88000/imx708@1a - Selected
 sensor format: 1536x864-SBGGR10_1X10 - Selected CFE format: 1536x864-PC1B
[INFO ] kinesisVideoStreamFormatChanged(): Stream format changed.
[DEBUG] setRequestHeader(): Appending header to request: user-agent -> AWS-SDK-KVS-CPP-
CLIENT/3.4.2/1.5.3 GCC/12.2.0 Linux/6.6.51+rpt-rpi-v8 aarch64 CPPSDK
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-stream-name -> demo-
stream
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-producer-start-
timestamp -> 1732012345.678
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-
acknowledgment-required -> 1
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-timecode-type
 -> ABSOLUTE
[DEBUG] setRequestHeader(): Appending header to request: transfer-encoding -> chunked
[DEBUG] setRequestHeader(): Appending header to request: connection -> keep-alive
[INFO ] putStreamResultEvent(): Put stream result event. New upload handle 0
[WARN ] notifyDataAvailable(): [demo-stream] Failed to un-pause curl with error: 43.
 Curl object 0xe2f6f418
Bus error
```

El PIC de Kinesis Video Streams utiliza un acceso a la memoria no alineado para optimizar el uso de la memoria, lo que no es compatible con todos los dispositivos.

Solución

Para usar el SDK en el modo de acceso a memoria alineado, debe establecer explícitamente el `ALIGNED_MEMORY_MODEL` CMake indicador en `ON` al compilar`kvssink`, ya que el valor predeterminado es. `OFF` Consulte the section called “Descargue y cree el SDK para productores de C ++” para obtener instrucciones más detalladas.

La marca de tiempo se congela y la tubería se detiene

Cuando se utiliza `x264enc` en una GStreamer canalización, es posible que se produzcan situaciones en las que el cronograma de la canalización se ralentice considerablemente o se detenga por completo en unos pocos segundos.

Esto se debe a que la configuración `x264enc` predeterminada puede introducir una latencia de codificación alta, que supera la capacidad del búfer de entrada predeterminado. Como resultado, el búfer de entrada se llena, lo que provoca que los elementos anteriores se bloqueen y que la tubería se detenga.

Para obtener más información, consulte la Documentación de GStreamer .

Solución

Configure `x264enc` con la opción de `zerolatency` ajuste. Esto reduce significativamente la latencia de codificación al optimizarla para escenarios en tiempo real, lo que garantiza que los fotogramas se procesen y emitan con mayor rapidez.

Ejemplo de configuración:

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

> **Note**
>
> Si bien esta solución evita eficazmente el estancamiento de la canalización, puede afectar a la eficiencia y la calidad de la codificación. Para los escenarios que requieren baja latencia y alta calidad, considere enfoques alternativos, como utilizar optimizaciones de hardware o encontrar una cámara web que emita directamente H.264, omitiendo este paso de codificación.
> Para obtener más información, consulte the section called “Utilice hardware”.

No se pueden ejecutar varias canalizaciones desde el mismo **`v4l2`** dispositivo al mismo tiempo

A estos dispositivos solo se `/dev/video0` puede acceder mediante un proceso a la vez. Si varios procesos intentan acceder a él al mismo tiempo, el segundo proceso espera a que se complete el primero.

Solución

Cree un dispositivo de bucle invertido que permita que varios procesos utilicen la interfaz de bucle invertido al mismo tiempo. Para obtener más información, consulte Stack Exchange.

Error interno en el flujo de datos

Al crear una GStreamer canalización, los elementos se conectan uniendo la almohadilla de origen de un elemento a la almohadilla de absorción de otro elemento. Este proceso de vinculación permite el flujo de datos desde el elemento fuente al elemento receptor, formando una canalización de datos.

El mensaje de error «Error en el enlace del teclado» en el registro GStreamer indica que se ha producido un problema al intentar establecer una conexión (enlace) entre los paneles de dos elementos de la canalización.

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

Solución

Determina qué elementos no se pueden vincular entre sí. Para reducir el alcance de la tubería, retire los elementos de la tubería. Sustituya el elemento situado más a la derecha por uno `fakesink` y extráigalos uno por uno.

Es posible que tengas que ajustar las tapas y los elementos filtrantes y and/or cambiar los elementos que utiliza tu canalización.

Los casos más comunes son pedir una cámara `framerate` o una cámara `resolution` que no sea compatible. `gst-device-monitor-1.0`Úselo en el terminal para obtener el soporte `frameratesresolutions`, `yformats`. Puede utilizar el GStreamer elemento videoscale para ajustar la resolución del vídeo y videorate para ajustar la velocidad de fotogramas del vídeo.

Para inspeccionar los formatos compatibles con un GStreamer elemento individual, escriba `gst-inspect-1.0 element-name` en el terminal.

Reproduzca contenido multimedia de su transmisión de vídeo de Kinesis

Abra la consola de Kinesis Video Streams y seleccione el nombre de la transmisión que ha creado.

En la consola aparece la transmisión de vídeo enviada desde el dispositivo Raspberry Pi.

> **Note**
>
> Es posible que el vídeo tarde unos segundos en aparecer en la consola.

Una vez que se esté reproduciendo la transmisión, puedes experimentar con las siguientes funciones de la consola:

- En la sección Video preview (Vista previa de vídeo), utilice los controles de navegación para rebobinar o avanzar la secuencia.
- En la sección de información de la transmisión, revisa el códec, la resolución y la velocidad de bits de la transmisión. En este tutorial, los valores de resolución y velocidad de bits de la Raspberry Pi están configurados a un nivel bajo a propósito para minimizar el uso del ancho de banda.

  Para ver las CloudWatch métricas de Amazon que se están creando para tu transmisión, selecciona Ver métricas de transmisión en CloudWatch.
- En Data retention period (Período de retención de datos), observe que la secuencia de vídeo se conserva durante un día. Puede editar este valor y establecerlo en No data retention (Sin retención de datos) o establecer un valor desde un día hasta varios años.
- En Cifrado del lado del servidor, observa que tus datos se cifran en reposo mediante una clave mantenida por AWS Key Management Service ()AWS KMS.

Problemas de reproducción

A continuación, se muestran algunos problemas de reproducción frecuentes y cómo solucionarlos.

No hay contenido multimedia, pero hay ACKs PERSISTENTES en los registros

Si ve PERSISTED ACK en los registros, significa que Kinesis Video Streams ha ingerido y almacenado correctamente el contenido multimedia cargado por. `kvssink` Los ACK recibidos de Kinesis Video Streams tienen este aspecto. En el JSON, observe el valor de la `"EventType"` clave.

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

Solución

Espere uno o dos minutos en la consola de Kinesis Video Streams y, a continuación, utilice la flecha doble hacia la derecha. Si no aparece ningún archivo multimedia, compruebe que la transmisión se está enviando a la región correcta y revise la ortografía del nombre de la transmisión. Puedes encontrar esta información en los registros.

Consulte the section called “Proporcione una región a `kvssink`” para obtener más información sobre cómo kvssink determina qué región usar.

El contenido multimedia tarda mucho en cargarse en el AWS Management Console

> **Important**
>
> La experiencia de reproducción de la consola es diferente a la experiencia de reproducción en HLS y DASH. Utilice también la página web de muestra alojada en el reproductor multimedia GitHub para probar la reproducción. El código fuente de la página web se puede encontrar aquí.

Es posible que el contenido multimedia se cargue lentamente en la consola debido a un ancho de banda de red deficiente o a un dispositivo limitado, pero también puede estar relacionado con la codificación y fragmentación del vídeo.

Conceptos básicos de codificación de vídeo:

- Los codificadores H.264 y H.265 utilizan fotogramas clave (fotogramas I) y fotogramas predecibles (fotogramas P) para una compresión eficiente.
- Los fotogramas clave contienen datos de imagen completos, mientras que los fotogramas P solo contienen cambios con respecto a los fotogramas anteriores.
- El «intervalo de fotogramas clave» determina la frecuencia con la que aparecen los fotogramas clave en la transmisión de vídeo.

Fragmentación en la transmisión:

- En Kinesis Video Streams, los fragmentos nuevos comienzan con cada fotograma clave. Para obtener más información, consulte the section called “Modelo de datos”.
- La longitud del fragmento (en segundos) se puede estimar de la siguiente manera: intervalo entre fotogramas clave ÷ velocidad de fotogramas

Ejemplo:

Para una transmisión con un intervalo entre fotogramas clave de 30 y una velocidad de fotogramas de 15 fps:

Longitud del fragmento = 30 ÷ 15 = 2 segundos

Debido a que los intervalos entre fotogramas clave y fotogramas son mayores, los fragmentos más largos aumentan la latencia en la transmisión multimedia.

Solución

Para mejorar los tiempos de carga, considere reducir el intervalo entre fotogramas clave. Esto creará fragmentos más cortos, lo que reducirá la latencia, pero también aumentará el tamaño del archivo de vídeo.

Para el `x264enc` GStreamer elemento, puedes establecer explícitamente el intervalo entre fotogramas y claves mediante la `key-int-max` propiedad:

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

Al revisar la salida del registro, observe la frecuencia con la que el cliente de carga recibe ACKs de Kinesis Video Streams. Cuantos más fotogramas clave se generen, más se devolverán ACKs .

El contenido multimedia está distorsionado o tiene artefactos

Para solucionar este problema, asegúrese de que todos los cables estén bien conectados. Revise la salida de los módulos de cámara `libcamera-hello` (o `raspistill` de las cámaras Pi antiguas).

En GStreamer proceso, `kvssink` sustitúyalo por `autovideosink` o `matroskamux` y`filesink`. Por ejemplo:

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

Revisa el archivo de salida `filesink` o el reproductor multimedia que se abre `autovideosink` al usarlo para comprobar si los artefactos también están ahí.

Revise también el resultado de la siguiente canalización:

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

Añadir elementos a la canalización, como Dewarp, puede corregir las salidas de las cámaras tipo ojo de pez.

Revisa los códecs de salida compatibles con tu cámara y ajusta los elementos según sea necesario.

Por ejemplo, si su cámara USB solo admite la salida JPEG, necesitará usar los `jpegdec` elementos `jpegparse` y para transformar el contenido multimedia antes de codificarlo en H.264. `x264enc` Busca ayuda en los GStreamer foros para otros usuarios con canales o configuraciones de cámara web similares.

Solución de problemas de compilación en el SDK para productores de C++ para Raspberry Pi

Si tienes un problema con la compilación y quieres probar diferentes CMake argumentos, asegúrate de realizar una compilación limpia. Elimine las `build` carpetas `open-sourcedependency`, y antes de volver a intentarlo.

Problemas de compilación con OpenSSL

Si recibe un resultado similar al siguiente, indica que OpenSSL ha detectado incorrectamente la arquitectura del 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]'
```

En este ejemplo, está intentando crear una versión de 64 bits (`linux-aarch64`) cuando esta Raspberry Pi es en realidad de 32 bits. Algunos dispositivos Raspberry Pi tienen un núcleo de 64 bits, pero un espacio de usuario de 32 bits.

Determine para qué arquitectura está intentando construir OpenSSL. Puedes encontrar la línea de registro durante el `configure` paso para 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 la arquitectura de tu sistema:

- Revise el nivel de bits del núcleo: ejecute `uname -m`
- Revise el espacio de usuario bit-ness: run `getconf LONG_BIT`

También puedes revisar la información de la CPU con nuestros comandos`cat /proc/cpuinfo`. `lscpu`

Solución

Para resolver este problema, añada el siguiente CMake argumento al compilar para garantizar que OpenSSL se compila correctamente para la arquitectura ARM de 32 bits:

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

Solucione problemas de carga **`kvssink`** en GStreamer

Confirme `GST_PLUGIN_PATH`

Asegúrese de que la variable de `GST_PLUGIN_PATH` entorno de su sesión de shell actual apunte al directorio que la contiene`kvssink`. Las variables de entorno son específicas de cada sesión, por lo que tendrás que configurarlas para cada nueva sesión. Para que este cambio sea permanente, consulta la sección «Actualiza el script de inicio del shell para incluir la configuración de la variable de entorno GST_PLUGIN_PATH».

Error: no se puede abrir el archivo de objetos compartidos: no existe ese archivo o directorio

Si se produce el error`Cannot open shared object file: No such file or directory`, ejecute el siguiente comando:

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

Si recibe el siguiente resultado, indica que el enlazador dinámico no puede encontrar las bibliotecas necesarias para`kvssink`. Esto suele ocurrir debido a:

- Mudarse `kvssink` a una ubicación diferente de donde se construyó.
- Compilación cruzada para una arquitectura de CPU incorrecta.
- Falta una dependencia necesaria.

Salida:

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

Solución

En el caso de las bibliotecas trasladadas, añada el directorio que contiene las bibliotecas que faltan`LD_LIBRARY_PATH`.

Desde el directorio raíz del repositorio original, puede localizar la biblioteca que falta mediante la `find` utilidad. En la terminal, escriba:

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

Salida:

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

El separador de rutas de archivos en los dispositivos Linux es`:`. El siguiente comando añade una nueva ruta de carpeta a la variable de `LD_LIBRARY_PATH` entorno existente, conservando los valores anteriores.

En tu terminal, escribe:

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

> **Important**
>
> Las variables de entorno son específicas de la sesión. Para conservar los cambios en todas las sesiones, modifique el script de inicio del shell.

Es posible que también tengas que añadir el `open-source/local/lib` a tu`$LD_LIBRARY_PATH`.

Error:. /path/to/libcproducer.so.1: encabezado ELF no válido

Si recibe este error al cargar bibliotecas compartidas, es posible que se deba a que los enlaces simbólicos están rotos ()`symlinks`. Los enlaces simbólicos pueden romperse si el sistema operativo de la máquina host no coincide con el de la máquina de destino. Por ejemplo, la compilación cruzada en una Raspberry MacBook Pi.

Otra posible causa es que los binarios creados eran para una arquitectura incorrecta. Por ejemplo, si los binarios se crearon para x86 (Raspberry Pi usa ARM). CPUs

Navegue hasta la ubicación de la biblioteca especificada en el error y escriba: `ls -la` para inspeccionar la biblioteca. `symlinks`

Respuesta:

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

En el ejemplo de salida anterior, no `symlinks` están rotos. Broken `symlinks` no tendrá flechas apuntando a sus objetivos.

Solución

Hay dos opciones para corregir los enlaces simbólicos:

- Recomendado: Vuelva a crearlo `symlink` con el `ln` comando. Escriba:

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

- Copie el archivo de biblioteca actual y cámbiele el nombre para que coincida con. `symlink`

> **Note**
>
> Esta opción aumenta el uso del almacenamiento.

Como práctica recomendada, compile en el mismo sistema operativo con herramientas como Docker para evitar problemas de compilación cruzada.

Dependencias faltantes:

Si el nombre de la biblioteca que falta empieza por`libkvs`, consulte la sección anterior sobre «bibliotecas trasladadas» para instalar las bibliotecas de Kinesis Video Streams del dispositivo anfitrión al dispositivo de destino.

De lo contrario, asegúrese de the section called “Requisitos previos de instalación del software” que todos los requisitos previos del software de código abierto estén instalados en el dispositivo de destino.

Referencia de códigos de error

Esta sección contiene información sobre códigos de error y estado para el Cargar a Kinesis Video Streams.

Para obtener información sobre las soluciones a problemas comunes, consulte Solución de problemas.

Temas

- Errores y códigos de estado devueltos por PutFrame Callbacks: código independiente de la plataforma (PIC)
- Errores y códigos de estado devueltos por PutFrame llamadas de retorno: biblioteca de productores de C

Errores y códigos de estado devueltos por PutFrame Callbacks: código independiente de la plataforma (PIC)

Las siguientes secciones contienen información sobre los errores y el estado que devuelven las llamadas para la `PutFrame` operación dentro del Código Independiente de la Plataforma (PIC).

Temas

- Códigos de error y estado devueltos por la biblioteca cliente
- Códigos de error y estado devueltos por la biblioteca de duración
- Códigos de error y estado devueltos por la biblioteca común
- Códigos de error y estado devueltos por la biblioteca de pilas
- Códigos de error y estado devueltos por la biblioteca MKVGen
- Códigos de error y estado devueltos por la biblioteca Trace
- Códigos de error y estado devueltos por la biblioteca Utils
- Códigos de error y estado devueltos por la biblioteca View

Códigos de error y estado devueltos por la biblioteca cliente

La siguiente tabla contiene información sobre errores y estados que devuelven los métodos de la biblioteca de Kinesis Video `Client` Streams.

Código	Mensaje	Descripción	Acción recomendada
`0x52000001`	`STATUS_MAX_STREAM_COUNT`	Se ha alcanzado el recuento máximo de secuencia.	Especifique un recuento de secuencia máximo de mayor tamaño en `DeviceInfo` tal y como se especifica en Cuotas de SDK para productores.

Código	Mensaje	Descripción	Acción recomendada
0x52000002	STATUS_MIN_STREAM_COUNT	Error mínimo de recuento de secuencia.	Especifique el número máximo de transmisiones de entrada superior a cero. DeviceInfo
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	Longitud de nombre de dispositivo no válida.	Consulte la longitud máxima del nombre del dispositivo en caracteres especificada enCuotas de SDK para productores.
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	Versión de estructura de DeviceInfo no válida.	Especifique la versión actual correcta de la estructura.
0x52000005	STATUS_MAX_TAG_COUNT	Se ha alcanzado el recuento máximo de etiquetas.	Consulte el número máximo de etiquetas actual que se especifica enCuotas de SDK para productores.
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	Versión de estructura de Callbacks no válida.	Especifique la versión actual correcta de la estructura.
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	Versión de estructura de StreamInfo no válida.	Especifique la versión actual correcta de la estructura.

Código	Mensaje	Descripción	Acción recomendada
`0x52000009`	`STATUS_INVALID_STREAM_NAME_LENGTH`	Longitud de nombre de secuencia no válida.	Consulte la longitud máxima del nombre de la transmisión en caracteres especificada enCuotas de SDK para productores.
`0x5200000a`	`STATUS_INVALID_STORAGE_SIZE`	Se ha especificado un tamaño de almacenamiento no válido.	El tamaño de almacenamiento en bytes deben estar dentro de los límites especificados en Cuotas de SDK para productores.
`0x5200000b`	`STATUS_INVALID_ROOT_DIRECTORY_LENGTH`	Longitud de cadena de directorio raíz no válida.	Consulte la longitud máxima de la ruta del directorio raíz especificada enCuotas de SDK para productores.
`0x5200000c`	`STATUS_INVALID_SPILL_RATIO`	Proporción de desbordamiento no válida.	Exprese la proporción de derrames como un porcentaje de 0 a 100.
`0x5200000d`	`STATUS_INVALID_STORAGE_INFO_VERSION`	Versión de estructura de `StorageInfo` no válida.	Especifique la versión actual correcta de la estructura.

Código	Mensaje	Descripción	Acción recomendada
`0x5200000e`	`STATUS_INVALID_STREAM_STATE`	La secuencia está en un estado que no permite la operación actual.	Por lo general, este error se produce cuando el SDK no alcanza el estado requerido para realizar la operación solicitada. Por ejemplo, ocurre si la llamada al API `GetStreamingEndpoint` devuelve un error y la aplicación cliente hace caso omiso del mismo y sigue colocando fotogramas en la secuencia.
`0x5200000f`	`STATUS_SERVICE_CALL_CALLBACKS_MISSING`	En la estructura `Callbacks` faltan puntos de entrada de función para algunas funciones obligatorias.	Compruebe que las llamadas obligatorias estén implementadas en la aplicación cliente. Este error solo está expuesto a los clientes de Platform Independent Code (PIC). C++ y otros contenedores de nivel superior cumplen estas llamadas.

Código	Mensaje	Descripción	Acción recomendada
`0x52000010`	`STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR`	No tiene autorización.	Compruebe el token de seguridad, el certificado, la integración y la caducidad del token de seguridad. Compruebe que el token tiene los derechos asociados correctos. Para las aplicaciones de ejemplo de Kinesis Video Streams, compruebe que la variable de entorno esté configurada correctamente.
`0x52000011`	`STATUS_DESCRIBE_STREAM_CALL_FAILED`	Error de API de `DescribeStream` .	Este error se devuelve después del error de reintento del API `DescribeStream` . El cliente PIC devuelve este error cuando deja de reintentarlo.
`0x52000012`	`STATUS_INVALID_DESCRIBE_STREAM_RESPONSE`	Estructura `DescribeStreamResponse` no válida.	La estructura que se transfirió al `DescribeStreamResultEvent` es nula o bien contiene elementos no válidos como un Nombre de recurso de Amazon (ARN) nulo.

Código	Mensaje	Descripción	Acción recomendada
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	La secuencia se está eliminando.	La eliminación de la secuencia provocó un error de API. Compruebe que ningún otro proceso esté intentando eliminar la transmisión mientras la transmisión está en uso.
0x52000014	STATUS_SERVICE_CALL_INVALID_ARG_ERROR	Se han especificado argumentos no válidos para la llamada de servicio.	El backend devuelve este error cuando un argumento de llamada de servicio no es válido o cuando el SDK encuentra un error que no puede interpretar.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	No se ha encontrado el dispositivo.	Comprueba que el dispositivo no se elimine mientras está en uso.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	No se ha aprovisionado el dispositivo.	Comprueba que se haya aprovisionado el dispositivo.

Código	Mensaje	Descripción	Acción recomendada
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	El servicio ha devuelto un recurso genérico no encontrado.	Este error se produce cuando el servicio no puede localizar el recurso (por ejemplo, una secuencia). Puede significar cosas diferentes en distintos contextos, pero la causa más probable es que se utilice APIs antes de crear la transmisión. El uso del SDK confirma que la transmisión se crea primero.
0x52000018	STATUS_INVALID_AUTH_LEN	Longitud no válida de información de autenticación.	Consulte los valores actuales que se especifican en Cuotas de SDK para productores.
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	La llamada al API `CreateStream` no se realizó correctamente.	Consulte la cadena de error para obtener información más detallada acerca de por qué la operación no se realizó correctamente.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	La llamada a `GetStreamingToken` no se realizó correctamente.	Consulte la cadena de error para obtener información más detallada acerca de por qué la operación no se realizó correctamente.

Código	Mensaje	Descripción	Acción recomendada
`0x5200002b`	`STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED`	La llamada al API `GetStreamingEndpoint` no se realizó correctamente.	Consulte la cadena de error para obtener información más detallada acerca de por qué la operación no se realizó correctamente.
`0x5200002c`	`STATUS_INVALID_URI_LEN`	El API `GetStreamingEndpoint` devolvió una longitud de cadena URI no válida.	Consulte los valores máximos actuales que se especifican en Cuotas de SDK para productores.
`0x5200002d`	`STATUS_PUT_STREAM_CALL_FAILED`	La llamada al API `PutMedia` no se realizó correctamente.	Consulte la cadena de error para obtener información más detallada acerca de por qué la operación no se realizó correctamente.

Código	Mensaje	Descripción	Acción recomendada
`0x5200002e`	`STATUS_STORE_OUT_OF_MEMORY`	El almacén de contenido se ha quedado sin memoria.	El almacén de contenido se comparte entre las secuencias y debe tener capacidad suficiente para almacenar las duraciones máxima para todas las secuencias + ~ 20% (teniendo en cuenta la desfragmentación). Es importante no desbordar el almacenamiento. Elija valores para la duración máxima por secuencia que se correspondan con el tamaño de almacenamiento acumulado y las tolerancias de latencia. Recomendamos dejar caer los fotogramas a medida que salen de la ventana de visualización de contenido en lugar de colocarlos simplemente (presión de memoria en el almacén de contenido). Esto se debe a que si se eliminan los fotogramas, se produce una devolución de llamada en las notificaciones sobre la presión del flujo. A continuación, la aplicación puede ajustar los componentes multimedi

Código	Mensaje	Descripción	Acción recomendada
			a de subida (como el codificador) para reducir la tasa de bits, anular fotogramas o actuar en consecuencia.
`0x5200002f`	`STATUS_NO_MORE_DATA_AVAILABLE`	No hay más datos disponibles actualmente para una secuencia.	Se trata de un posible resultado válido cuando la canalización multimedia genera fotogramas a una velocidad menor que los consume el subproceso de red para enviarlos al servicio. Los clientes de nivel superior (por ejemplo, C++, Java o Android) no ven esta advertencia porque se gestiona internamente.
`0x52000030`	`STATUS_INVALID_TAG_VERSION`	Versión de estructura de `Tag` no válida.	Especifique la versión actual correcta de la estructura.
`0x52000031`	`STATUS_SERVICE_CALL_UNKNOWN_ERROR`	La pila de redes ha devuelto un error genérico o desconocido.	Consulte los logs para ver información más detallada.
`0x52000032`	`STATUS_SERVICE_CALL_RESOURCE_IN_USE_ERROR`	Recurso en uso.	Se devuelve desde el servicio. Para obtener más información, consulte la Referencia de la API de Kinesis Video Streams.

Código	Mensaje	Descripción	Acción recomendada
0x52000033	STATUS_SERVICE_CALL_CLIENT_LIMIT_ERROR	Límite de cliente.	Se devuelve desde el servicio. Para obtener más información, consulte la Referencia de la API de Kinesis Video Streams.
0x52000034	STATUS_SERVICE_CALL_DEVICE_LIMIT_ERROR	Límite de dispositivos.	Se devuelve desde el servicio. Para obtener más información, consulte la Referencia de la API de Kinesis Video Streams.
0x52000035	STATUS_SERVICE_CALL_STREAM_LIMIT_ERROR	Límite de secuencia.	Se devuelve desde el servicio. Para obtener más información, consulte la Referencia de la API de Kinesis Video Streams.
0x52000036	STATUS_SERVICE_CALL_RESOURCE_DELETED_ERROR	El recurso se ha eliminado o se está eliminando.	Se devuelve desde el servicio. Para obtener más información, consulte la Referencia de la API de Kinesis Video Streams.
0x52000037	STATUS_SERVICE_CALL_TIMEOUT_ERROR	Se ha agotado el tiempo de espera de la llamada de servicio.	Llamar a un API de servicio concreto ha dado lugar a que se supere el tiempo de espera. Compruebe que tiene una conexión de red válida. El PIC volverá a intentar la operación de forma automática.

Código	Mensaje	Descripción	Acción recomendada
`0x52000038`	`STATUS_STREAM_READY_CALLBACK_FAILED`	Notificación de secuencia lista.	Esta notificación se envía desde el PIC al cliente para indicar que se ha creado la secuencia asíncrona.
`0x52000039`	`STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL`	Se han especificado etiquetas no válidas.	El número de etiquetas no es cero, pero las etiquetas están vacías. Compruebe que las etiquetas estén especificadas o que el recuento sea cero.
`0x5200003a`	`STATUS_INVALID_STREAM_DESCRIPTION_VERSION`	Versión de estructura de `StreamDescription` no válida.	Especifique la versión actual correcta de la estructura.
`0x5200003b`	`STATUS_INVALID_TAG_NAME_LEN`	Longitud de nombre de etiqueta no válida.	Consulte los límites para el nombre de la etiqueta que se especifican en Cuotas de SDK para productores.
`0x5200003c`	`STATUS_INVALID_TAG_VALUE_LEN`	Longitud de valor de etiqueta no válida.	Consulte los límites para el valor de la etiqueta que se especifican en Cuotas de SDK para productores.

Código	Mensaje	Descripción	Acción recomendada
`0x5200003d`	`STATUS_TAG_STREAM_CALL_FAILED`	El API `TagResource` ha devuelto un error.	La llamada al API `TagResource` no se realizó correctamente. Compruebe si hay una conexión de red válida. Consulte los logs para obtener más información sobre el error.
`0x5200003e`	`STATUS_INVALID_CUSTOM_DATA`	Datos personalizados no válidos al llamar al PIC APIs.	Se especificaron datos personalizados no válidos en una llamada al PIC APIs. Esto puede ocurrir únicamente en los clientes que usan PIC directamente.
`0x5200003f`	`STATUS_INVALID_CREATE_STREAM_RESPONSE`	Estructura `CreateStreamResponse` no válida.	La estructura o sus campos de miembro no son válidos (es decir, el ARN es nulo o mayor del especificado en Cuotas de SDK para productores).

Código	Mensaje	Descripción	Acción recomendada
`0x52000040`	`STATUS_CLIENT_AUTH_CALL_FAILED`	Error en autenticación de cliente.	El PIC no pudo obtener la información de autenticación adecuada (`AccessKeyId oSecretAccessKey`) tras varios intentos. Compruebe la integración de autenticación. Las aplicaciones de muestra utilizan variables de entorno para transferir información de credenciales a la biblioteca C++ Producer.
`0x52000041`	`STATUS_GET_CLIENT_TOKEN_CALL_FAILED`	Error al obtener la llamada del token de seguridad.	Esta situación puede darse para clientes que utilizan PIC directamente. Después de una serie de reintentos, la llamada devuelve este error.
`0x52000042`	`STATUS_CLIENT_PROVISION_CALL_FAILED`	Error de aprovisionamiento.	El aprovisionamiento no está implementado.
`0x52000043`	`STATUS_CREATE_CLIENT_CALL_FAILED`	No se pudo crear el Producer Client.	El PIC ha devuelto un error genérico tras una serie de reintentos cuando la creación de cliente devuelve un error.

Código	Mensaje	Descripción	Acción recomendada
`0x52000044`	`STATUS_CLIENT_READY_CALLBACK_FAILED`	Error al llevar el Producer Client a un estado LISTO.	Devuelto por la máquina de estado de PIC si el PIC genera un error al desplazarse al estado LISTO. Consulte los logs para obtener más información sobre la causa raíz.
`0x52000045`	`STATUS_TAG_CLIENT_CALL_FAILED`	El `TagResource` del Producer Client ha fallado.	La llamada al API `TagResource` ha devuelto un error para Producer Client. Consulte los logs para obtener más información sobre la causa raíz.
`0x52000046`	`STATUS_INVALID_CREATE_DEVICE_RESPONSE`	Error en la creación de Dispositivo/Producer.	Los niveles superiores SDKs (por ejemplo, C++ o Java) aún no implementan la API de creación de dispositivos o productores. Los clientes que utilizan PIC directamente pueden indicar un error mediante la notificación de resultado .

Código	Mensaje	Descripción	Acción recomendada
`0x52000047`	`STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW`	La marca temporal de la ACK recibida no está en la vista.	Este error se produce si el fotograma correspondiente a la ACK recibida cae fuera de la ventana de la vista de contenido. Por lo general, esto ocurre si la entrega de ACK es lenta. Se puede interpretar como una advertencia y una indicación de que el enlace de descarga es lento.
`0x52000048`	`STATUS_INVALID_FRAGMENT_ACK_VERSION`	Versión de estructura de `FragmentAck` no válida.	Especifique la versión actual correcta de la estructura `FragmentAck` .
`0x52000049`	`STATUS_INVALID_TOKEN_EXPIRATION`	Vencimiento de token de seguridad no válido.	La caducidad del token de seguridad debería tener una marca de tiempo absoluta en el futuro superior a la marca de tiempo actual, con un período de gracia. Para los límites para el periodo de gracia, consulte el Cuotas de SDK para productores.

Código	Mensaje	Descripción	Acción recomendada
`0x5200004a`	`STATUS_END_OF_STREAM`	Indicador de finalización de secuencia (EOS).	En la llamada al API `GetStreamData` , indica que la sesión de control de carga ha finalizado. Esto ocurre si la sesión finaliza o genera errores o si el token de la sesión ha vencido y la sesión se está girando.
`0x5200004b`	`STATUS_DUPLICATE_STREAM_NAME`	Duplicar nombre de secuencia.	Varias secuencias no puede tener el mismo nombre de secuencia. Elija un nombre exclusivo para la secuencia.
`0x5200004c`	`STATUS_INVALID_RETENTION_PERIOD`	Período de retención no válido.	Se especifica un período de retención no válido en la estructura `StreamInfo` . Para obtener información acerca del rango de valores válido para el periodo de retención, consulte Cuotas de SDK para productores.

Código	Mensaje	Descripción	Acción recomendada
`0x5200004d`	`STATUS_INVALID_ACK_KEY_START`	No válido `FragmentAck` .	No se ha podido analizar la cadena ACK del fragmento. Indicador de inicio de clave no válido. La cadena ACK del fragmento podría estar dañada. Se puede corregir automáticamente, y este error se puede tratar como una advertencia.
`0x5200004e`	`STATUS_INVALID_ACK_DUPLICATE_KEY_NAME`	No válido `FragmentAck` .	No se ha podido analizar la cadena ACK del fragmento. Varias claves tienen el mismo nombre. La cadena ACK del fragmento podría estar dañada. Se puede corregir automáticamente, y este error se puede tratar como una advertencia.
`0x5200004f`	`STATUS_INVALID_ACK_INVALID_VALUE_START`	No válido `FragmentAck` .	No se pudo analizar la cadena ACK del fragmento debido a un indicador de inicio de valor de clave no válido. La cadena ACK del fragmento podría estar dañada. Se puede corregir automáticamente, y este error se puede tratar como una advertencia.

Código	Mensaje	Descripción	Acción recomendada
0x52000050	STATUS_INVALID_ACK_INVALID_VALUE_END	No válido FragmentAck .	No se pudo analizar la cadena ACK del fragmento debido a un indicador de finalización de valor de clave no válido. La cadena ACK del fragmento podría estar dañada. Se puede corregir automáticamente, y este error se puede tratar como una advertencia.
0x52000051	STATUS_INVALID_PARSED_ACK_TYPE	No válido FragmentAck .	No se pudo analizar la cadena ACK del fragmento porque se especificó un tipo de fragmento ACK no válido.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	La secuencia se detuvo.	La secuencia se ha detenido, pero todavía se está colocando un fotograma en la secuencia .
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Versión de estructura de StreamMetrics no válida.	Especifique la versión actual correcta de la estructura StreamMetrics .
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Versión de estructura de ClientMetrics no válida.	Especifique la versión actual correcta de la estructura ClientMetrics .

Código	Mensaje	Descripción	Acción recomendada
`0x52000055`	`STATUS_INVALID_CLIENT_READY_STATE`	No se pudo inicializar el Producer para alcanzar un estado LISTO.	No se pudo alcanzar el estado LISTO durante la inicialización del Producer Client. Consulte los logs para obtener más información.
`0x52000056`	`STATUS_STATE_MACHINE_STATE_NOT_FOUND`	Error interno de máquina de estado.	No es un error visible públicamente.
`0x52000057`	`STATUS_INVALID_FRAGMENT_ACK_TYPE`	Tipo ACK no válido especificado en la estructura `FragmentAck` .	La estructura `FragmentAck` debe contener los tipos de ACK definidos en el encabezado público.
`0x52000058`	`STATUS_INVALID_STREAM_READY_STATE`	Error interno de transición de máquina de estado.	No es un error visible públicamente.
`0x52000059`	`STATUS_CLIENT_FREED_BEFORE_STREAM`	El objeto de la secuencia se liberó después de liberar el Producer.	Se ha producido un intento de liberar un objeto de secuencia después de liberar el objeto de Producer. Esto puede ocurrir únicamente en los clientes que usan PIC directamente.
`0x5200005a`	`STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED`	Error de almacenamiento interno.	Un error interno que indica que el tamaño real de la asignación del almacén de contenido es menor que el tamaño del marco y el fragmento empaquetados.

Código	Mensaje	Descripción	Acción recomendada
`0x5200005b`	`STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION`	Error de almacenamiento interno.	El tamaño almacenado o de la asignación en la vista de contenido es mayor que el tamaño de asignación en el almacén de contenido.
`0x5200005c`	`STATUS_ACK_ERR_STREAM_READ_ERROR`	ACK de error de lectura de secuencia.	Un error que el ACK devolvió desde el servidor y que indica un error de lectura o análisis de la transmisión. Esto ocurre por lo general cuando el backend devuelve un error al recuperar la secuencia. El restreaming automático normalmente puede corregir este error.

Código	Mensaje	Descripción	Acción recomendada
`0x5200005d`	`STATUS_ACK_ERR_FRAGMENT_SIZE_REACHED`	Se ha alcanzado el tamaño máximo de fragmento.	El tamaño máximo de fragmento en bytes se define en Cuotas de SDK para productores. Este error indica que hay fotogramas muy grandes o bien que no hay fotogramas clave para crear fragmentos de tamaño manejable. Compruebe la configuración del codificador y compruebe que los fotogramas clave se estén produciendo correctamente. Para secuencias que tengan una densidad muy alta, configure el codificador para producir fragmentos con duraciones más pequeñas para administrar el tamaño máximo.

Código	Mensaje	Descripción	Acción recomendada
`0x5200005e`	`STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED`	Se ha alcanzado la duración máxima de fragmento.	La duración máxima de fragmento se define en Cuotas de SDK para productores. Este error indica que hay una tasa de fotogramas por segundo muy baja o bien que no hay fotogramas clave para crear fragmentos de duración manejable. Compruebe la configuración del codificador y compruebe que los fotogramas clave se están produciendo correctamente a intervalos regulares.
`0x5200005f`	`STATUS_ACK_ERR_CONNECTION_DURATION_REACHED`	Se ha alcanzado la duración máxima de conexión.	Kinesis Video Streams impone la duración máxima de conexión especificada en. Cuotas de SDK para productores El SDK para productores rota automáticamente la transmisión o el token antes de que se alcance el máximo. Los clientes que utilizan el SDK no deberían recibir este error.

Código	Mensaje	Descripción	Acción recomendada
`0x52000060`	`STATUS_ACK_ERR_FRAGMENT_TIMECODE_NOT_MONOTONIC`	Los códigos temporales no aumentan de forma monótona.	El SDK para productores aplica marcas de tiempo, por lo que los clientes que utilizan el SDK no deberían recibir este error.
`0x52000061`	`STATUS_ACK_ERR_MULTI_TRACK_MKV`	Varias pistas en el MKV.	El SDK para productores exige transmisiones de una sola pista, por lo que los clientes que utilizan el SDK no deberían recibir este error.

Código	Mensaje	Descripción	Acción recomendada
`0x52000062`	`STATUS_ACK_ERR_INVALID_MKV_DATA`	Datos de MKV no válidos.	El analizador de MKV de backend ha encontrado un error al analizar la secuencia. Los clientes que utilizan el SDK pueden encontrar este error si la transmisión está dañada durante la transición. Esto también puede ocurrir si las presiones del búfer obligan al SDK a eliminar los fotogramas finales que se transmiten parcialmente. En este último caso, te recomendamos que reduzcas los FPS y la resolución, que aumentes la relación de compresión o (si hay una red con «ráfagas») que permitas almacenar más contenido y almacenar más tiempo en el búfer para adaptarte a las presiones temporales.

Código	Mensaje	Descripción	Acción recomendada
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	Marca temporal de productor no válida.	El servicio devuelve este ACK de error si el reloj del Producer está muy adelantado. Los niveles superiores SDKs (por ejemplo, Java o C++) utilizan alguna versión del reloj del sistema para ajustar la hora actual a la llamada del PIC. Compruebe que el reloj del sistema esté configurado correctamente. Los clientes que utilicen el PIC directamente deben comprobar que sus funciones de devolución de llamada devuelven la marca de tiempo correcta.
0x52000064	STATUS_ACK_ERR_STREAM_NOT_ACTIVE	Secuencia inactiva.	Se realizó una llamada al API de backend mientras la secuencia no estaba en estado "Activo". Esto ocurre cuando el cliente crea la secuencia e inmediatamente continua para insertar fotogramas en la misma. El SDK gestiona esta situación a través de la máquina de estado y el mecanismo de recuperación.

Código	Mensaje	Descripción	Acción recomendada
0x52000065	STATUS_ACK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS error de acceso denegado.	Se devuelve cuando la cuenta no tiene acceso a la clave especificada.
0x52000066	STATUS_ACK_ERR_KMS_KEY_DISABLED	AWS KMS la clave está deshabilitada.	La clave especificada se ha deshabilitado.
0x52000067	STATUS_ACK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS error de validación de clave.	Error de validación genérico. Para obtener más información, consulte la Referencia de la API de AWS Key Management Service.
0x52000068	STATUS_ACK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key no disponible.	La clave no está disponible. Para obtener más información, consulte la Referencia de la API de AWS Key Management Service.
0x52000069	STATUS_ACK_ERR_KMS_KEY_INVALID_USAGE	Uso de la clave KMS no válido.	No AWS KMS key está configurado para usarse en este contexto. Para obtener más información, consulte la Referencia de la API de AWS Key Management Service.
0x5200006a	STATUS_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS estado no válido.	Para obtener más información, consulte la Referencia de la API de AWS Key Management Service.

Código	Mensaje	Descripción	Acción recomendada
`0x5200006b`	`STATUS_ACK_ERR_KMS_KEY_NOT_FOUND`	No se encontró la clave KMS.	No se ha encontrado la clave. Para obtener más información, consulte la Referencia de la API de AWS Key Management Service.
`0x5200006c`	`STATUS_ACK_ERR_STREAM_DELETED`	La secuencia se ha eliminado o se está eliminando.	La transmisión se está eliminando mediante otra aplicación o a través de la AWS Management Console.
`0x5200006d`	`STATUS_ACK_ERR_ACK_INTERNAL_ERROR`	Error interno.	Error interno de servicio genérico.
`0x5200006e`	`STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR`	Error de archivado de fragmento.	Se devuelve cuando el servicio no persiste de forma duradera e indexa el fragmento. Aunque es raro, puede ocurrir por diversos motivos. De forma predeterminada, el SDK vuelve a intentar enviar el fragmento.
`0x5200006f`	`STATUS_ACK_ERR_UNKNOWN_ACK_ERROR`	Error desconocido.	El servicio ha devuelto un error desconocido.
`0x52000070`	`STATUS_MISSING_ERR_ACK_ID`	Falta información de ACK.	El analizador de ACK ha completado el análisis, pero falta la información de `FragmentAck` .

Código	Mensaje	Descripción	Acción recomendada
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	Longitud de segmento de ACK no válida.	Se ha especificado una cadena de segmento ACK con una longitud no válida al analizador ACK. Para obtener más información, consulte Cuotas de SDK para productores.
0x52000074	STATUS_MAX_FRAGMENT_METADATA_COUNT	El número máximo de elementos de metadatos se ha añadido a un fragmento.	Una transmisión de vídeo de Kinesis puede añadir hasta 10 elementos de metadatos a un fragmento, ya sea añadiendo un elemento no persistente a un fragmento o añadiendo un elemento persistente a la cola de metadatos. Para obtener más información, consulte Uso de metadatos de streaming con Kinesis Video Streams.
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	Se ha alcanzado un límite (número máximo de metadatos, longitud de nombre de metadatos o longitud de valor de metadatos).	El SDK de Producer limita el número y el tamaño de los elementos de metadatos. Este error no se produce a menos que se cambien los límites del código del SDK para productores. Para obtener más información, consulte Uso de metadatos de streaming con Kinesis Video Streams.

Código	Mensaje	Descripción	Acción recomendada
0x52000076	STATUS_BLOCKING_PUT_INTERRUPTED_STREAM_TERMINATED	No implementado.	
0x52000077	STATUS_INVALID_METADATA_NAME	El nombre de metadatos no es válido.	El nombre de los metadatos no puede empezar por la cadena "AWS». Si se produce este error, el elemento de metadatos no se agrega al fragmento o a la cola de metadatos. Para obtener más información, consulte Uso de metadatos de streaming con Kinesis Video Streams.
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	El final de un fotograma del fragmento se encuentra en un estado no válido.	El final del fragmento no debe enviarse en una secuencia non-key-frame fragmentada.
0x52000079	STATUS_TRACK_INFO_MISSING	Falta información de pista.	El número de pista debe ser mayor que cero y coincidir con el identificador de la pista.
0x5200007a	STATUS_MAX_TRACK_COUNT_EXCEEDED	Se ha superado el número máximo de pistas.	Puedes tener un máximo de tres pistas por transmisión.

Código	Mensaje	Descripción	Acción recomendada
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	El tiempo de retención del modo de streaming sin conexión se ha configurado en cero.	El tiempo de retención del modo de transmisión sin conexión no debe estar establecido en cero.
0x5200007c	STATUS_ACK_ERR_TRACK_NUMBER_MISMATCH	No coincide el número de pista del error ACK.	
0x5200007d	STATUS_ACK_ERR_FRAMES_MISSING_FOR_TRACK	Faltan los fotogramas de una pista.	
0x5200007e	STATUS_ACK_ERR_MORE_THAN_ALLOWED_TRACKS_FOUND	Se ha superado el número máximo permitido de pistas.	
0x5200007f	STATUS_UPLOAD_HANDLE_ABORTED	Se ha anulado el controlador de carga.	
0x52000080	STATUS_INVALID_CERT_PATH_LENGTH	Longitud de ruta de certificado no válida.	
0x52000081	STATUS_DUPLICATE_TRACK_ID_FOUND	Se ha encontrado un ID de pista duplicado.	

Código	Mensaje	Descripción	Acción recomendada
0x52000082	STATUS_INVALID_CLIENT_INFO_VERSION		
0x52000083	STATUS_INVALID_CLIENT_ID_STRING_LENGTH		
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUTMEDIA_LAST_PERSIST_ACK_NOT_RECEIVED		

Código	Mensaje	Descripción	Acción recomendada
0x52000089	STATUS_NON_ALIGNED_HEAP_WITH_IN_CONTENT_STORE_ALLOCATORS		
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_IMAGE_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_IMAGE_METADATA_VALUE_LENGTH		

Códigos de error y estado devueltos por la biblioteca de duración

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `Duration` biblioteca.

Código	Mensaje
`0xFFFFFFFFFFFFFFFF`	`INVALID_DURATION_VALUE`

Códigos de error y estado devueltos por la biblioteca común

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `Common` biblioteca.

> **Note**
>
> Estos códigos de información de error y estado son comunes a muchos APIs.

Código	Código sin 0 a la izquierda	Mensaje	Descripción
0x00000001	0x1	`STATUS_NULL_ARG`	Se ha transferido NULL para un argumento obligatorio.
0x00000002	0x2	`STATUS_INVALID_ARG`	Se ha especificado un valor no válido para un argumento.
0x00000003	0x3	`STATUS_INVALID_ARG_LEN`	Se ha especificado una longitud de argumento no válida.
0x00000004	0x4	`STATUS_NOT_ENOUGH_MEMORY`	No se pudo asignar suficiente memoria.
0x00000005	0x5	`STATUS_BUFFER_TOO_SMALL`	El tamaño del búfer especificado es demasiado pequeño.

Código	Código sin 0 a la izquierda	Mensaje	Descripción
0x00000006	0x6	`STATUS_UNEXPECTED_EOF`	Se ha alcanzado un final de archivo inesperado.
0x00000007	0x7	`STATUS_FORMAT_ERROR`	Se encontró un formato no válido.
0x00000008	0x8	`STATUS_INVALID_HANDLE_ERROR`	Valor de controlador no válido.
0x00000009	0x9	`STATUS_OPEN_FILE_FAILED`	Error al abrir un archivo.
0x0000000a	0xa	`STATUS_READ_FILE_FAILED`	Error al leer desde un archivo.
0x0000000b	0xb	`STATUS_WRITE_TO_FILE_FAILED`	Error al escribir en un archivo.
0x0000000c	0xc	`STATUS_INTERNAL_ERROR`	Un error interno que normalmente no ocurre y podría indicar un error de API de servicio o SDK.
0x0000000d	0xd	`STATUS_INVALID_OPERATION`	Se ha producido una operación no válida o la operación no está permitida.
0x0000000e	0xe	`STATUS_NOT_IMPLEMENTED`	La característica no está implementada.

Código	Código sin 0 a la izquierda	Mensaje	Descripción
0x0000000f	0xf	`STATUS_OPERATION_TIMED_OUT`	La operación ha agotado el tiempo de espera.
0x00000010	0x10	`STATUS_NOT_FOUND`	No se ha encontrado un recurso necesario.
0x00000011	0x11	`STATUS_CREATE_THREAD_FAILED`	No se pudo crear un hilo.
0x00000012	0x12	`STATUS_THREAD_NOT_ENOUGH_RESOURCES`	Los recursos son insuficientes para crear otro subproceso o se ha detectado un límite impuesto por el sistema en el número de subprocesos.
0x00000013	0x13	`STATUS_THREAD_INVALID_ARG`	Se especificaron atributos de hilo no válidos o ya hay otro hilo esperando unirse a este hilo.
0x00000014	0x14	`STATUS_THREAD_PERMISSIONS`	No hay permiso para establecer la política de programación ni los parámetros especificados en los atributos del hilo.

Código	Código sin 0 a la izquierda	Mensaje	Descripción
0x00000015	0x15	`STATUS_THREAD_DEADLOCKED`	Se ha detectado un punto muerto o el hilo de unión especifica el hilo de llamada.
0x00000016	0x16	`STATUS_THREAD_DOES_NOT_EXIST`	No se puede encontrar el hilo con el ID de hilo especificado.
0x00000017	0x17	`STATUS_JOIN_THREAD_FAILED`	La operación de unión de subprocesos devolvió un error desconocido o genérico.
0x00000018	0x18	`STATUS_WAIT_FAILED`	Se ha superado el tiempo máximo de espera para la variable condicional.
0x00000019	0x19	`STATUS_CANCEL_THREAD_FAILED`	La operación de cancelación del subproceso devolvió un error desconocido o genérico.
0x0000001a	0x1a	`STATUS_THREAD_IS_NOT_JOINABLE`	La operación de unión de hilos se solicita en un hilo que no se puede unir.

Código	Código sin 0 a la izquierda	Mensaje	Descripción
0x0000001b	0x1b	`STATUS_DETACH_THREAD_FAILED`	La operación de separación del hilo devolvió un error desconocido o genérico.
0x0000001C	0x1c	`STATUS_THREAD_ATTR_INIT_FAILED`	No se pudo inicializar el objeto de atributos del hilo.
0x0000001d	0x1d	`STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED`	No se pudo establecer el tamaño de la pila para el objeto de atributos del hilo.
0x0000001e	0x1e	`STATUS_MEMORY_NOT_FREED`	Solo se usa en las pruebas. Indica que no se ha liberado toda la memoria solicitada.
0x0000001f	0x1f	`STATUS_INVALID_THREAD_PARAMS_VERSION`	Versión de estructura «ThreadParams» no válida. Especifique la versión actual correcta de la estructura.

Códigos de error y estado devueltos por la biblioteca de pilas

La siguiente tabla contiene la información de error y estado que devuelven los métodos de la `Heap` biblioteca.

Código	Mensaje	Descripción
0x10000001	STATUS_HEAP_FLAGS_ERROR	Se ha especificado una combinación no válida de marcadores.
0x10000002	STATUS_HEAP_NOT_INITIALIZED	Se ha intentado realizar una operación antes de que se inicializará el montón.
0x10000003	STATUS_HEAP_CORRUPTED	El montón estaba dañado o se sobrescribió la banda de protección (en modo de depuración). Un desbordamiento del búfer en el código del cliente podría dar lugar a una corrupción del montón.
0x10000004	STATUS_HEAP_VRAM_LIB_MISSING	La biblioteca de modo kernel o de usuario de VRAM (RAM de vídeo) no se puede cargar o falta. Compruebe si la plataforma subyacente es compatible con asignaciones de VRAM.
0x10000005	STATUS_HEAP_VRAM_LIB_REOPEN	Error al abrir la biblioteca de VRAM.
0x10000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	Error al cargar la exportación de la función INIT.
0x10000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	Error al cargar la exportación de la función ALLOC.
0x10000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	Error al cargar la exportación de la función FREE.

Código	Mensaje	Descripción
0x10000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	Error al cargar la exportación de la función LOCK.
0x1000000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	Error al cargar la exportación de la función UNLOCK.
0x1000000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	Error al cargar la exportación de la función UNINIT.
0x1000000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	Error al cargar la exportación de la función GETMAX.
0x1000000d	STATUS_HEAP_DIRECT_MEM_INIT	Error al inicializar el grupo de montón principal en el montón híbrido.
0x1000000e	STATUS_HEAP_VRAM_INIT_FAILED	Error de inicialización dinámica de VRAM.
0x1000000f	STATUS_HEAP_LIBRARY_FREE_FAILED	Error al eliminar la asignación y liberar la biblioteca de VRAM.
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	Error de asignación de VRAM.
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	Error de liberación de VRAM.
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	Error de mapeo de VRAM.
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	Error de desmapeo de VRAM.
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	Error de desinicialización de VRAM.

Código	Mensaje	Descripción
0x10000015	`STATUS_INVALID_ALLOCATION_SIZE`	
0x10000016	`STATUS_HEAP_REALLOC_ERROR`	
0x10000017	`STATUS_HEAP_FILE_HEAP_FILE_CORRUPT`	

Códigos de error y estado devueltos por la biblioteca MKVGen

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `MKVGen` biblioteca.

Código	Mensaje	Descripción/Acción recomendada
0x32000001	`STATUS_MKV_INVALID_FRAME_DATA`	Miembros no válidos de la estructura de datos `Frame`. Asegúrese de que los datos de duración, tamaño y fotograma sean válidos y estén dentro de los límites especificados enCuotas de SDK para productores.
0x32000002	`STATUS_MKV_INVALID_FRAME_TIMESTAMP`	Marca temporal no válida. La PTS (marca temporal de presentación) calculada y DTS (marca temporal de decodificación) son mayores o iguales que la marca temporal del fotograma de inicio del fragmento. Esto es una indicación de un posible

Código	Mensaje	Descripción/Acción recomendada
		reinicio de canalización multimedia o de un problema de estabilidad de codificador. Para obtener información sobre la resolución de problemas, consulte Error: "Failed to submit frame to Kinesis Video client" (No se pudo enviar el fotograma al cliente de Kinesis Video)
0x32000003	`STATUS_MKV_INVALID_CLUSTER_DURATION`	Se ha especificado una duración de fragmento no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x32000004	`STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH`	Longitud de cadena de tipo de contenido no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x32000005	`STATUS_MKV_NUMBER_TOO_BIG`	Se ha producido un intento de codificar un número que es demasiado grande para representarlo en formato EBML (Metalenguaje binario extensible). No debería exponerse a los clientes de SDK.

Código	Mensaje	Descripción/Acción recomendada
0x32000006	`STATUS_MKV_INVALID_CODEC_ID_LENGTH`	Longitud de cadena de ID de códec no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x32000007	`STATUS_MKV_INVALID_TRACK_NAME_LENGTH`	Longitud de cadena de nombre de pista no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x32000008	`STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH`	Longitud de datos privados de códec no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x32000009	`STATUS_MKV_CODEC_PRIVATE_NULL`	Los datos privados (CPD) del códec son NULOS, mientras que el tamaño del CPD es superior a cero.
0x3200000a	`STATUS_MKV_INVALID_TIMECODE_SCALE`	Valor de escala de código temporal no válida. Para obtener más información, consulte Cuotas de SDK para productores.
0x3200000b	`STATUS_MKV_MAX_FRAME_TIMECODE`	El código temporal del fotograma es mayor que el máximo. Para obtener más información, consulte Cuotas de SDK para productores.

Código	Mensaje	Descripción/Acción recomendada
0x3200000c	`STATUS_MKV_LARGE_FRAME_TIMECODE`	Se ha alcanzado el código temporal máximo de fotograma. El formato MKV utiliza 16 bits firmados para representar el código temporal relativo del fotograma al principio del clúster. El error se genera si no se puede representar el código de tiempo del fotograma. Este error indica una selección de escala de código temporal deficiente o que la duración del clúster es demasiado larga, de modo que la representación del código temporal del fotograma desborda el espacio de 16 bits con signo.

Código	Mensaje	Descripción/Acción recomendada
0x3200000d	`STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA`	Se ha encontrado un código de inicio de Anexo B no válido. Por ejemplo, se especificó la marca de adaptación de Anexo B y el código encuentra una secuencia de inicio no válida de más de tres ceros. Un formato de Anexo B válido debe disponer de una secuencia de "prevención de emulación" para escapar de una secuencia de tres o más ceros en la secuencia de bytes. Para obtener más información, consulte la especificación MPEG. Para obtener información sobre este error en Android, consulte Error STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) en Android.
0x3200000e	`STATUS_MKV_INVALID_AVCC_NALU_IN_FRAME_DATA`	El empaquetado AVCC NALU no es válido cuando se especifica el indicador AVCC de adaptación. Compruebe que el bytestream esté en un formato AVCC válido. Para obtener más información, consulte la especificación MPEG.

Código	Mensaje	Descripción/Acción recomendada
0x3200000f	`STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED`	Se especificaron tanto la adaptación del AVCC como el anexo B. NALUs Especifique uno o especifique ninguno.
0x32000010	`STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD`	Formato de Anexo B de CPD no válido cuando se ha especificado la marca de Anexo B de adaptación. Compruebe que el CPD esté en un formato de anexo B válido. Si no es así, quite el indicador de adaptación al anexo B del CPD.
0x32000011	`STATUS_MKV_PTS_DTS_ARE_NOT_SAME`	Kinesis Video Streams exige que la PTS (marca de tiempo de presentación) y la DTS (marca de tiempo de decodificación) sean las mismas para los fotogramas de inicio del fragmento. Estos son los fotogramas principales que inician el fragmento.
0x32000012	`STATUS_MKV_INVALID_H264_H265_CPD`	Error al analizar los datos privados del códec H264/H265 .
0x32000013	`STATUS_MKV_INVALID_H264_H265_SPS_WIDTH`	Error al extraer la anchura de los datos privados de códec.

Código	Mensaje	Descripción/Acción recomendada
0x32000014	`STATUS_MKV_INVALID_H264_H265_SPS_HEIGHT`	Error al extraer la altura de los datos privados de códec.
0x32000015	`STATUS_MKV_INVALID_H264_H265_SPS_NALU`	La NALU H264/H265 SPS no es válida.
0x32000016	`STATUS_MKV_INVALID_BIH_CPD`	Formato de encabezado de información de mapa de bits no válido en los datos privados del códec.
0x32000017	`STATUS_MKV_INVALID_HEVC_NALU_COUNT`	Número no válido de unidades de capa de abstracción de red (NALU) de codificación de vídeo de alta eficiencia (HEVC).
0x32000018	`STATUS_MKV_INVALID_HEVC_FORMAT`	Formato HEVC no válido.
0x32000019	`STATUS_MKV_HEVC_SPS_NALU_MISSING`	Falta el HEVC NALUs en el conjunto de parámetros de secuencia (SPS).
0x3200001a	`STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE`	Tamaño no válido de NALU de SPS de HEVC.
0x3200001b	`STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC`	IDC de formato Chroma no válido.
0x3200001c	`STATUS_MKV_INVALID_HEVC_SPS_RESERVED`	SPS reservado de HEVC no válido.

Código	Mensaje	Descripción/Acción recomendada
0x3200001d	`STATUS_MKV_MIN_ANNEX_B_CPD_SIZE`	Tamaño mínimo AnnexBb del valor beta privado del códec. Para H264, este valor debe ser igual o mayor que 11. Para H265, este valor debe ser igual o mayor que 15.
0x3200001e	`STATUS_MKV_ANNEXB_CPD_MISSING_NALUS`	Faltan datos privados del códec en el anexo B. NALUs
0x3200001f	`STATUS_MKV_INVALID_ANNEXB_CPD_NALUS`	La beta privada del códec no es válida en el anexo B. NALUs
0x32000020	`STATUS_MKV_INVALID_TAG_NAME_LENGTH`	Longitud de nombre de etiqueta no válida. El valor válido es mayor que cero y menor que 128.
0x32000021	`STATUS_MKV_INVALID_TAG_VALUE_LENGTH`	Longitud de valor de etiqueta no válida. El valor válido es superior a cero e inferior a 256.
0x32000022	`STATUS_MKV_INVALID_GENERATOR_STATE_TAGS`	Etiquetas de estado de generador no válidas.
0x32000023	`STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX`	Índice de frecuencia de muestreo de datos privados de códec AAC no válido.
0x32000024	`STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG`	Configuración del canal de datos privado de códec AAC no válido.

Código	Mensaje	Descripción/Acción recomendada
0x32000025	`STATUS_MKV_INVALID_AAC_CPD`	Datos privados de códec AAC no válidos.
0x32000026	`STATUS_MKV_TRACK_INFO_NOT_FOUND`	Información de pista no encontrada.
0x32000027	`STATUS_MKV_INVALID_SEGMENT_UUID`	UUID de segmento no válido.
0x32000028	`STATUS_MKV_INVALID_TRACK_UID`	UID de pista no válido.
0x32000029	`STATUS_MKV_INVALID_CLIENT_ID_LENGTH`	
0x3200002a	`STATUS_MKV_INVALID_AMS_ACM_CPD`	
0x3200002b	`STATUS_MKV_MISSING_SPS_FROM_H264_CPD`	
0x3200002c	`STATUS_MKV_MISSING_PPS_FROM_H264_CPD`	
0x3200002d	`STATUS_MKV_INVALID_PARENT_TYPE`	

Códigos de error y estado devueltos por la biblioteca Trace

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `Trace` biblioteca.

Código	Mensaje
`0x10100001`	`STATUS_MIN_PROFILER_BUFFER`

Códigos de error y estado devueltos por la biblioteca Utils

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `Utils` biblioteca.

Código	Mensaje
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT _ERROR
0x4000000f	STATUS_STRFTIME_FALIED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT _STR_LEN_EXCEEDED

Código	Mensaje
0x40000011	STATUS_UTIL_MAX_TAG_COUNT
0x40000012	STATUS_UTIL_INVALID_TAG_VERSION
0x40000013	STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL
0x40000014	STATUS_UTIL_INVALID_TAG_NAME_LEN
0x40000015	STATUS_UTIL_INVALID_TAG_VALUE_LEN
0x4000002a	STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE
0x4000002b	STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED
0x4000002c	STATUS_THREADPOOL_MAX_COUNT
0x4000002d	STATUS_THREADPOOL_INTERNAL_ERROR
0x40100001	STATUS_HASH_KEY_NOT_PRESENT
0x40100002	STATUS_HASH_KEY_ALREADY_PRESENT
0x40100003	STATUS_HASH_ENTRY_ITERATION_ABORT
0x41000001	STATUS_BIT_READER_OUT_OF_RANGE
0x41000002	STATUS_BIT_READER_INVALID_SIZE
0x41100001	STATUS_TIMER_QUEUE_STOP_SCHEDULING
0x41100002	STATUS_INVALID_TIMER_COUNT_VALUE

Código	Mensaje
0x41100003	STATUS_INVALID_TIMER_PERIOD_VALUE
0x41100004	STATUS_MAX_TIMER_COUNT_REACHED
0x41100005	STATUS_TIMER_QUEUE_SHUTDOWN
0x41200001	STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN
0x41200002	STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

Códigos de error y estado devueltos por la biblioteca View

La siguiente tabla contiene información de error y estado que devuelven los métodos de la `View` biblioteca.

Código	Mensaje	Descripción
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	Se ha especificado un recuento de elemento de vista de contenido no válido. Para obtener más información, consulte Cuotas de SDK para productores.
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	Se ha especificado una duración vista de contenido no válida. Para obtener más información, consulte Cuotas de SDK para productores.

Código	Mensaje	Descripción
0x30000003	`STATUS_CONTENT_VIEW_NO_MORE_ITEMS`	Se ha realizado un intento de superar la posición de cabeza.
0x30000004	`STATUS_CONTENT_VIEW_INVALID_INDEX`	Se ha especificado un índice no válido.

Código	Mensaje	Descripción
0x30000005	`STATUS_CONTENT_VIEW_INVALID_TIMESTAMP`	Se ha producido un solapamiento de marca temporal o una marca temporal no válida. La marca de tiempo de decodificación del cuadro debe ser mayor o igual que la marca de tiempo del cuadro anterior, más la duración del cuadro anterior: . `` `DTS(n) >= DTS(n-1) + Duration(n-1)` `` Este error a menudo indica un codificador "inestable". El codificador produce una ráfaga de fotogramas codificados y sus marcas temporales son más pequeñas que las duraciones dentro del fotograma. O la secuencia está configurada para utilizar marcas temporales de SDK y los fotogramas se envían a una velocidad superior a las duraciones del fotograma. Para contribuir a la "fluctuación" en el codificador, especifique una duración de fotograma menor en la estructura `StreamInfo.StreamCaps`. Por ejemplo, si la transmisión es de 25 FPS, la duración de cada fotograma es de 40 ms. Sin embargo, para controlar la «fluctuación» del codificador,

Código	Mensaje	Descripción
		le recomendamos que utilice la mitad de esa duración de fotogramas (20 ms). Algunas secuencias requieren un control más preciso durante la sincronización para detección de errores.
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	Se ha especificado una longitud de datos de elemento de vista de contenido no válida.

Errores y códigos de estado devueltos por PutFrame llamadas de retorno: biblioteca de productores de C

La siguiente sección contiene información sobre los errores y el estado que devuelven las llamadas para la `PutFrame` operación en la biblioteca del productor de C.

Código	Mensaje	Descripción	Acción recomendada
`0x15000001`	`STATUS_STOP_CALLBACK_CHAIN`	La cadena de devolución de llamada se ha detenido.	
`0x15000002`	`STATUS_MAX_CALLBACK_CHAIN`	Se alcanzó máximo de la cadena de devolución de llamada.	
`0x15000003`	`STATUS_INVALID_PLATFORM_CALLBACKS_VERSION`	Versión de estructura de `PlatformCallbacks` no válida.	Especifique la versión actual correcta de la estructura.

Código	Mensaje	Descripción	Acción recomendada
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Versión de estructura de `ProducerCallbacks` no válida.	Especifique la versión actual correcta de la estructura.
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Versión de estructura de `StreamCallbacks` no válida.	Especifique la versión actual correcta de la estructura.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Versión de estructura de `AuthCallbacks` no válida.	Especifique la versión actual correcta de la estructura.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	Versión de estructura de `ApiCallbacks` no válida.	Especifique la versión actual correcta de la estructura.
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Versión de estructura de `AwsCredentials` no válida.	Especifique la versión actual correcta de la estructura.
0x15000009	STATUS_MAX_REQUEST_HEADER_COUNT	Se alcanzó el recuento máximo del encabezado de solicitud.	
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	Se alcanzó la longitud de nombre máxima del encabezado de solicitud.	

Código	Mensaje	Descripción	Acción recomendada
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	Se alcanzó la longitud máxima del valor de encabezado de solicitud.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	JSON de retorno no válido para una llamada a la API.	
0x1500000d	STATUS_CURL_INIT_FAILED	Error de inicialización de Curl.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Error de inicialización de Curl lib.	
0x1500000f	STATUS_INVALID_DESCRIBE_STREAM_RETURN_JSON	El JSON devuelto no es válido para. DescribeStream	
0x15000010	STATUS_HMAC_GENERATION_ERROR	Error de generación de HMAC.	
0x15000011	STATUS_IOT_FAILED	Falló la autorización de IoT.	
0x15000012	STATUS_MAX_ROLE_ALIAS_LEN_EXCEEDED	Se alcanzó la longitud máxima del alias de función.	Especifique un alias más corto.

Código	Mensaje	Descripción	Acción recomendada
0x15000013	STATUS_MAX_USER_AGENT_NAME_POSTFIX_LEN_EXCEEDED	Se alcanzó la longitud máxima del posfijo del nombre de agente.	
0x15000014	STATUS_MAX_CUSTOM_USER_AGENT_LEN_EXCEEDED	Se alcanzó la longitud máxima del agente de usuario del cliente.	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	Longitud de agente de usuario no válida.	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	Periodo de almacenamiento en caché de punto de enlace no válido.	Especifique un periodo de almacenamiento en caché inferior a 24 horas.
0x15000017	STATUS_IOT_EXPIRATION_OCCURS_IN_PAST	La marca de tiempo de caducidad de IoT se produce en el pasado.	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	El análisis de caducidad de IoT ha fallado.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FREE_FUNC		

Código	Mensaje	Descripción	Acción recomendada
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAX_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_LWS_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CA_CERT_PATH		

Código	Mensaje	Descripción	Acción recomendada
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RESET_FAILED		

Código	Mensaje	Descripción	Acción recomendada
0x16000001	STATUS_CURL_PERFORM_FAILED	CURL devolvió un código que no se realizó correctamente.	Revise los registros para obtener información adicional. Un error de CURL común es: «No se ha podido resolver el nombre del host». «, compruebe la conectividad a Internet del dispositivo. Otro error común es el código de error 403. Esto indica que los certificados de IoT no se crean ni especifican correctamente. Compruebe que las rutas de los archivos a los certificados y permisos de IoT estén configuradas correctamente. Para obtener más información, consulte the section called "Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT".
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH	Recibió una respuesta de longitud 0 al buscar las credenciales de IoT.	Revisa el panel de AWS salud e inténtalo de nuevo más tarde.

Código	Mensaje	Descripción	Acción recomendada
`0x16000003`	`STATUS_IOT_NULL_AWS_CREDS`	El JSON devuelto por el punto final de credenciales de IoT no contenía el objeto de credenciales.	Revisa el elemento «mensaje» del JSON para obtener más información.
`0x16000004`	`STATUS_IOT_INVALID_URI_LEN`	La URL que se pasa a la función de búsqueda de credenciales de IoT no tiene una longitud entre 1 y 10 000.	Revisa la URL transferida a esta función.
`0x16000005`	`STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT`	El elemento de «caducidad» del JSON al obtener las credenciales de IoT no tiene el formato:`YYYY-MM-DDTHH:mm:SSZ` .	Revisa el panel de AWS salud e inténtalo de nuevo más tarde.

Referencia de marcas de adaptación de capa de abstracción de red (NAL)

Esta sección contiene información sobre las marcas disponibles para la enumeración `StreamInfo.NalAdaptationFlags`.

La secuencia elemental de una aplicación puede tener formato Annex-B o AVCC:

- El formato del anexo B delimita NALUs (unidades de la capa de abstracción de red) con dos bytes de ceros, seguidos de uno o tres bytes de ceros, seguidos del número 1 (denominado código de inicio, por ejemplo, 00000001).

- El formato AVCC también es envolvente NALUs, pero cada NALU va precedido de un valor que indica el tamaño de la NALU (normalmente cuatro bytes).

Muchos codificadores producen el formato de secuencia de bits Annex-B. Algunos procesadores de flujo de bits de nivel superior (como un motor de reproducción o el reproductor Media Source Extensions (MSE) que contiene AWS Management Console) utilizan el formato AVCC para sus fotogramas.

Los datos privados del códec (CPD), que son SPS/PPS (conjunto de parámetros de secuencia/conjunto de parámetros de imagen) del códec H.264, también pueden estar en formato de anexo B o AVCC. Sin embargo, para los CPD, los formatos son distintos de los descritos anteriormente.

Los indicadores indican al SDK que debe adaptarlo al AVCC o al anexo B para los datos de fotogramas y el NALUs CPD de la siguiente manera:

Indicador	Adaptación
`NAL_ADAPTATION_FLAG_NONE`	Sin adaptación.
`NAL_ADAPTATION_ANNEXB_NALS`	Adapte el anexo B NALUs a la AVCC. NALUs
`NAL_ADAPTATION_AVCC_NALS`	Adapte el AVCC al anexo B. NALUs NALUs
`NAL_ADAPTATION_ANNEXB_CPD_NALS`	Adapte el anexo B NALUs para los datos privados del códec al formato AVCC. NALUs
`NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS`	Adapte el anexo B NALUs para el códec y encuadre los datos privados al formato AVCC. NALUs

Para obtener más información sobre los tipos de NALU, consulte Sección 1.3: Tipos de unidad de capa de abstracción de red en RFC 3984.

Estructuras del SDK para productores

En esta sección se incluye información sobre las estructuras que puede utilizar para proporcionar datos al objeto Producer de Kinesis Video Streams.

Temas

- DeviceInfo/DefaultDeviceInfoProvider
- StorageInfo

DeviceInfo/DefaultDeviceInfoProvider

Los DefaultDeviceInfoProviderobjetos DeviceInfoy controlan el comportamiento del objeto Producer de Kinesis Video Streams.

Campos miembros

- versión: valor entero que se utiliza para garantizar que se utilice la versión correcta de la estructura con la versión actual del código base. La versión actual se especifica utilizando la macro `DEVICE_INFO_CURRENT_VERSION`.
- nombre: el nombre legible para las personas del dispositivo.
- TagCount/tags: no se utiliza actualmente.
- StreamCount: el número máximo de transmisiones que puede gestionar el dispositivo. Este asigna previamente el almacenamiento de punteros a los objetos de secuencia inicialmente, pero los objetos de secuencia reales se crean posteriormente. El valor predeterminado es 16 secuencias, pero puede cambiar este número en el archivo `DefaultDeviceInfoProvider.cpp`.
- storageInfo: un objeto que describe la configuración de almacenamiento principal. Para obtener más información, consulte StorageInfo.

StorageInfo

Especifica la configuración del almacenamiento principal de Kinesis Video Streams.

La implementación predeterminada se basa en una implementación en montón rápida de baja fragmentación, optimizada para streaming. Utiliza el asignador `MEMALLOC`, que se puede sobrescribir en una plataforma determinada. Algunas plataformas disponen de asignación de memoria virtual sin contar con la asignación con páginas físicas. A medida que se utiliza la memoria, las páginas

virtuales están respaldadas por las páginas físicas. Esto da como resultado una presión de memoria baja en el sistema general cuando el almacenamiento está infrautilizado.

Calcule el tamaño de almacenamiento predeterminado con la siguiente fórmula. El parámetro `DefragmentationFactor` debe estar establecido en 1,2 (20 por ciento).

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

En el siguiente ejemplo, un dispositivo tiene secuencias de audio y vídeo. La secuencia de audio tiene 512 muestras por segundo, con una muestra media de 100 bytes. La transmisión de vídeo tiene 25 fotogramas por segundo, con un promedio de 10 000 bytes. Cada transmisión tiene 3 minutos de duración del búfer.

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

Si el dispositivo tiene más memoria disponible, le recomendamos que añada más memoria al almacenamiento para evitar una fragmentación grave.

Compruebe que el tamaño de almacenamiento es adecuado para alojar todos los búferes de todas las transmisiones con una alta complejidad de codificación (cuando el tamaño del fotograma es mayor debido a un movimiento elevado) o cuando el ancho de banda es bajo. Si el productor alcanza la presión de la memoria, emite llamadas de retorno por exceso de almacenamiento (). `StorageOverflowPressureFunc` Sin embargo, cuando no hay memoria disponible en el almacén de contenido, se descarta el fotograma que se está insertando en Kinesis Video Streams con un error `STATUS_STORE_OUT_OF_MEMORY = 0x5200002e` (). Para obtener más información, consulte Códigos de error y estado devueltos por la biblioteca cliente. Esto también puede ocurrir si los acuses de recibo de la aplicación (ACKs) no están disponibles o si los pendientes se retrasan. ACKs En este caso, los búferes se llenan hasta alcanzar la capacidad de «duración del búfer» antes de que comiencen a agotarse los fotogramas anteriores.

Campos de miembros

- versión: valor entero que se utiliza para garantizar que se utilice la versión correcta de la estructura con la versión actual del código base.
- StorageType: enumeración que especifica el `DEVICE_STORAGE_TYPE` respaldo y la implementación subyacentes del almacenamiento. Actualmente el único valor admitido es `DEVICE_STORAGE_TYPE_IN_MEM`. Una próxima implementación admitirá

`DEVICE_STORAGE_TYPE_HYBRID_FILE`, lo que indica que el almacenamiento utiliza el almacén de contenido con respaldo de archivos.

- StorageSize: el tamaño de almacenamiento en bytes que se va a preasignar. La asignación mínima permitida es de 10 MB; y la máxima es de 10 GB. (Esto cambiará con la implementación futura del almacén de contenido respaldado por archivos).
- SpillRatio: valor entero que representa el porcentaje de almacenamiento que se va a asignar desde el tipo de almacenamiento de memoria directo (RAM), a diferencia del almacenamiento de desbordamiento secundario (almacenamiento de archivos). En la actualidad no se utiliza.
- rootDirectory: ruta del directorio donde se encuentra el almacén de contenido respaldado por archivos. En la actualidad no se utiliza.

Estructuras de transmisión de vídeo de Kinesis

Puede utilizar las siguientes estructuras para proporcionar datos a una instancia de una transmisión de vídeo de Kinesis.

Temas

- StreamDefinition/StreamInfo
- ClientMetrics
- StreamMetrics

StreamDefinition/StreamInfo

El objeto `StreamDefinition` en la capa de C++ envuelve el objeto `StreamInfo` en el código independiente de plataforma y proporciona algunos valores predeterminados en el constructor.

Campos de miembros

Campo	Tipo de datos:	Descripción	Valor predeterminado
stream_name	`string`	Nombre de secuencia opcional. Para obtener más informaci ón acerca de la longitud del nombre	Si no se especifica ningún nombre, se genera uno de forma aleatoria.

Campo	Tipo de datos:	Descripción	Valor predeterminado
		de secuencia, consulte Cuotas de SDK para productores. Cada secuencia debe tener un nombre único.	
retention_period	`duration<uint64_t, ratio<3600>>`	El periodo de retención de la secuencia, en segundos. Especificar **`0`** indica que no hay ninguna retención.	3600 (una hora)
etiquetas	`const map<string, string>*`	Un mapa de pares clave/valor que contiene información de usuario. Si la secuencia ya tiene un conjunto de etiquetas, las nuevas etiquetas se añaden al conjunto de etiquetas existente.	No hay etiquetas
kms_key_id	`string`	El identificador de AWS KMS clave que se utilizará para cifrar la transmisión. Para obtener más información, consulte Protección de datos en Kinesis Video Streams.	La clave de KMS predeterminada (`aws/kinesisvideo`.)

Campo	Tipo de datos:	Descripción	Valor predeterminado
streaming_type	Enumeración de `STREAMING_TYPE`	El único valor admitido es `STREAMING_TYPE_REALTIME` .	
content_type	`string`	El formato del contenido de la secuencia. La consola Kinesis Video Streams puede reproducir contenido en este formato`video/h264` .	`video/h264`
max_latency	`duration<uint64_t, milli>`	La latencia máxima en milisegundos de la secuencia. La devolución de llamada de presión de latencia de secuencia (si se ha especificado) se invoca cuando se supera la duración del búfer este tiempo. Especificar **`0`** indica que no se invocará ninguna devolución de llamada de presión de latencia de transmisión.	`milliseconds::zero()`

Campo	Tipo de datos:	Descripción	Valor predeterminado
fragment_duration	`duration<uint64_t>`	La duración del fragmento que desea, en segundos. Este valor se utiliza en combinación con el valor `key_frame_fragmentation` . Si este valor es`false`, Kinesis Video Streams genera fragmentos en un fotograma clave una vez transcurrido este tiempo. Por ejemplo, una secuencia de audio Advanced Audio Coding (AAC) tiene cada fotograma como fotograma clave. Si se especifica `key_frame_fragmentation = false` , la fragmentación se realiza en un fotograma clave después de que expire este tiempo, lo que generará fragmentos de 2 segundos.	2

Campo	Tipo de datos:	Descripción	Valor predeterminado
timecode_scale	`duration<uint64_t, milli>`	La escala de código de tiempo de MKV en milisegundos, que especifica la granularidad de los códigos de tiempo para los fotogramas dentro del clúster de MKV. El código de tiempo del fotograma de MKV siempre tiene relación con el inicio del clúster. MKV utiliza un valor de 16 bits firmado (0-32767) para representar el código de tiempo dentro del clúster (fragmento). Compruebe que el código de tiempo del fotograma se pueda representar con la escala de código de tiempo indicada. El valor de escala de código de tiempo predeterminado de 1 ms garantiza que el marco mayor que se puede represent ar sea 32767 ms ~= 32 segundos. Esto supera la duración de	1

Campo	Tipo de datos:	Descripción	Valor predeterminado
		fragmento máxima que se especifica en Cuotas de servicio de Amazon Kinesis Video Streams, que es 10 segundos.	
key_frame_fragmentation	`bool`	Si se producen fragmentos en un fotograma clave. Si es `true`, el SDK produce un inicio del fragmento cada vez que haya un fotograma clave. Si`false`, Kinesis Video Streams espera `fragment_duration` al menos y produce un nuevo fragmento en el fotograma clave que le sigue.	`true`

Campo	Tipo de datos:	Descripción	Valor predeterminado
frame_timecodes	`bool`	Si se utilizan códigos de tiempo de fotograma o se generan marcas de tiempo utilizando la devolución de llamada de tiempo actual. Muchos codificad ores no producen marcas temporales con los fotogramas. Por lo tanto, `false` la especificación de este parámetro garantiza que los fotogramas tengan una marca de tiempo a medida que se colocan en Kinesis Video Streams.	`true`

Campo	Tipo de datos:	Descripción	Valor predeterminado
absolute_fragment_times	`bool`	Kinesis Video Streams utiliza MKV como mecanismo de empaquetado subyacente. La especificación MKV es estricta acerca de los códigos de tiempo de fotograma relativos al inicio de clúster (fragment o). Sin embargo, los códigos de tiempo del clúster pueden ser absolutos o relativos a la hora de inicio para la transmisión. Si las marcas de tiempo son relativas, la llamada a la API de servicio de `PutMedia` utiliza la marca de tiempo de inicio de la secuencia opcional y ajusta las marcas temporales del clúster. El servicio siempre almacena los fragmentos con sus marcas de tiempo absolutas.	`true`

Campo	Tipo de datos:	Descripción	Valor predeterminado
fragment_acks	`bool`	Si desea recibir un fragmento a nivel de aplicación ACKs (agradecimientos).	`true`, lo que significa que el SDK recibirá el SDK ACKs y actuará en consecuencia.
restart_on_error	`bool`	Si se reinicia ante errores específicos.	`true`, lo que significa que el SDK intenta reiniciar el streaming si se producen errores.
recalculate_metrics	`bool`	Si se vuelven a calcular las métricas. Cada llamada para recuperar las métricas puede recalcularlos para obtener el valor de "ejecución" más reciente, lo que podría tener un pequeño impacto en la CPU. Es posible que tenga que establecerse en `false` en dispositi vos de huella/potencia extremadamente baja para ahorrar ciclos de la CPU. De lo contrario, no recomendamos usarlo `false` para este valor.	`true`

Campo	Tipo de datos:	Descripción	Valor predeterminado
nal_adaptation_flags	`uint32_t`	Especifica las marcas de adaptación de unidad de capa de abstracción de red (NALU). Si el flujo de bits está codificado en H.264, puede procesars e sin procesar o empaquetarse. NALUs Estas se encuentran en el Anexo B o en formato AVCC. La mayoría de los productor es y consumidores de flujos elemental es (codificadores y decodificadores de lectura) utilizan el formato del anexo B porque presenta ventajas, como la recuperación de errores. Los sistemas de nivel superior utilizan el formato AVCC, que es el formato predeterm inado para MPEG, HLS, DASH, etc. La reproducción de consola utiliza las MSE (extensiones	El valor predeterm inado consiste en adaptar el formato de Anexo B al formato AVCC tanto para los datos de fotogramas como para los datos privados de códec.

Campo	Tipo de datos:	Descripción	Valor predeterminado
		de fuente multimedia) del navegador para descodificar y reproducir la secuencia que utiliza el formato AVCC. Para H.264 (y para M-JPEG y H.265), el SDK proporciona capacidades de adaptación. Muchas secuencias elementales se encuentran en el siguiente formato. En este ejemplo, Ab es el código de inicio de Anexo B (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)` En el caso del H.264, los datos privados del códec (CPD) se encuentran en los parámetro	

Campo	Tipo de datos:	Descripción	Valor predeterminado
		s SPS (conjunto de parámetros de secuencia) y PPS (conjunto de parámetros de imagen), y se pueden adaptar al formato AVCC. A menos que la canalizac ión multimedia proporcione el CPD por separado, la aplicación puede extraer el CPD desde el fotograma. Para ello, puede buscar el primer fotograma IDR (que debe contener el SPS y el PPS), extraer los dos NALUs (que sí lo están`Ab(Sps)Ab(Pps) )` y colocarlo en el CPD. `StreamDefinition` Para obtener más información, consulte the section called "Banderas de adaptación NAL".	

Campo	Tipo de datos:	Descripción	Valor predeterminado
frame_rate	`uint32_t`	La velocidad de fotogramas prevista. Este valor se utiliza para calcular mejor las necesidades de búfer.	25
avg_bandwidth_bps	`uint32_t`	El ancho de banda medio previsto para la secuencia. Este valor se utiliza para calcular mejor las necesidades de búfer.	4 * 1024 * 1024

Campo	Tipo de datos:	Descripción	Valor predeterminado
buffer_duration	`duration<uint64_t>`	La duración del búfer de la secuencia , en segundos. El SDK mantiene los fotogramas en el almacén de contenido hasta el`buffer_duration` , tras lo cual los fotogramas anteriores se eliminan a medida que la ventana avanza. Si el fotograma que se va a eliminar no se ha enviado al backend, se invoca la llamada de retroceso del fotograma eliminado . Si la duración del búfer actual es mayor que `max_latency` , se invoca la devolución de llamada de presión de latencia de la secuencia. El búfer se recorta al siguiente inicio de fragmento cuando se recibe el ACK almacenado de forma persistente del fragmento. Esto indica que el contenido se ha almacenado de forma persistente y	120

Campo	Tipo de datos:	Descripción	Valor predeterminado
		duradera en la nube, por lo que ya no es necesario almacenar el contenido en el dispositivo local.	

Campo	Tipo de datos:	Descripción	Valor predeterminado
replay_duration	`duration<uint64_t>`	El tiempo, en segundos, que se tarda en girar el lector actual hacia atrás para volver a reproducirlo en caso de error si está activado el reinicio. La restauración se detiene al inicio del búfer (en caso de que se acabe de iniciar el streaming o la ACK almacenada de forma persistente haya llegado). La restauración intenta ir a un fotograma clave que indique un inicio de fragmento. Si el error que provoca el reinicio no indica que el servidor esté inactivo (el servidor sigue activo y contiene los datos del fotograma en sus búferes internos), la reversión se detiene en el último fotograma ACK recibido. A continuación, avanza al siguiente fotograma clave, ya que todo	40

Campo	Tipo de datos:	Descripción	Valor predeterminado
		el fragmento está almacenado en la memoria del host.	
connection_staleness	`duration<uint64_t>`	El tiempo, expresado en segundos, tras el cual se invoca la devolución de señal de inactividad de la transmisión si el SDK no recibe el ACK almacenado en búfer. Indica que los fotogramas se envían desde el dispositivo, pero el backend no los reconoce. Esta condición indica un corte de conexión en el salto intermedio o en el balanceador de carga.	30
codec_id	`string`	El ID del códec para la pista MKV.	`V_MPEG4/ISO/AVC`
track_name	`string`	El nombre de la pista MKV.	`kinesis_video`

Campo	Tipo de datos:	Descripción	Valor predeterminado
codecPrivateData	`unsigned char*`	El búfer de datos privados de códec (CPD). Si la canalización multimedia tiene la información sobre los CPD antes de que se inicie la secuencia, se pueden configurar en `StreamDefinition.codecPrivateData`. Los bits copiados y el búfer se pueden volver a utilizar o liberar después de la llamada para crear la secuencia. Sin embargo, si los datos no están disponibles cuando se crea la transmisión, se pueden configurar en una de las sobrecargas de la `KinesisVideoStream.start(cpd)` función.	`null`
codecPrivateDataSize (Tamaño)	`uint32_t`	El tamaño del búfer de datos privados de códec.	0

ClientMetrics

El ClientMetricsobjeto se rellena al llamar`getKinesisVideoMetrics`.

Campos de miembros

Campo	Tipo de datos:	Descripción
versión	`UINT32`	La versión de la estructur a, definida en la macro `CLIENT_METRICS_CUR RENT_VERSION` .
contentStoreSize	`UINT64`	El tamaño del almacén de contenido general en bytes. Este es el valor especificado en `DeviceInfo.Storage Info.storageSize` .
contentStoreAvailableTamaño	`UINT64`	Tamaño de almacenamiento disponible actual en bytes.
contentStoreAllocatedTamaño	`UINT64`	Tamaño asignado actual. Los tamaños asignado más disponible deben ser ligeramente más pequeños que el tamaño de almacenam iento general, debido a los datos la contabilidad interna y la implementación del almacén de contenido.
totalContentViewsTamaño	`UINT64`	El tamaño de la memoria asignada para todas las vistas de contenido para todas las secuencias. Esto no se tiene en cuenta para el tamaño de almacenamiento. Esta

Campo	Tipo de datos:	Descripción
		memoria se asigna mediante la macro `MEMALLOC`, que se puede sobrescribir para proporcionar un asignador personalizado.
totalFrameRate	`UINT64`	La velocidad de fotograma total observada en todas las secuencias.
totalTransferRate	`UINT64`	La velocidad total de la secuencia observada en bytes por segundo en todas las secuencias.

StreamMetrics

El StreamMetricsobjeto se llena al llamar`getKinesisVideoMetrics`.

Campos de miembros

Campo	Tipo de datos:	Descripción
versión	`UINT32`	La versión de la estructur a, definida en la macro `STREAM_METRICS_CUR RENT_VERSION` .
currentViewDuration	`UINT64`	La duración de los fotograma s acumulados. En el caso de una red rápida, esta duración es cero o la duración de la trama (a medida que se transmite la trama). Si la duración es superior a la `max_latency` especificada

Campo	Tipo de datos:	Descripción
		`enStreamDefinition` , se invoca la llamada de retorno de latencia de la transmisión si así se especifica. La duración se especifica en unidades 100 ns, que es la unidad de tiempo predeterminada para la capa PIC.
overallViewDuration	`UINT64`	La duración de vista general. Si la transmisión se configura con ausencia ACKs o persistencia, este valor aumenta a medida que se colocan los fotogramas en la transmisión de vídeo de Kinesis y pasa a ser igual al `buffer_duration` de. `StreamDefinition` Cuando ACKs están activados y se recibe el ACK persisten te, el búfer se recorta hasta el siguiente fotograma clave. Esto se debe a que la marca de tiempo del ACK indica el principio de todo el fragmento . La duración se especifica en unidades 100 ns, que es la unidad de tiempo predeterm inada para la capa PIC.
currentViewSize	`UINT64`	El tamaño en bytes del búfer actual.

Campo	Tipo de datos:	Descripción
overallViewSize	`UINT64`	El tamaño general de vista en bytes.
currentFrameRate	`UINT64`	La velocidad de fotogramas observada para la secuencia actual.
currentTransferRate	`UINT64`	La velocidad de transfere ncia observada en bytes por segundo para la secuencia actual.

Retrollamadas del SDK del productor

Las clases y los métodos del SDK para productores de Amazon Kinesis Video Streams no mantienen sus propios procesos. En su lugar, utilizan las llamadas y eventos de función de entrada para programar la comunicación de las devoluciones de llamadas con la aplicación.

Existen dos patrones de devolución de llamada que puede utilizar la aplicación para interactuar con el SDK:

- `CallbackProvider`— Este objeto expone todas las llamadas del componente de código independiente de la plataforma (PIC) a la aplicación. Este patrón permite funcionalidad completa, pero también significa que la implementación debe gestionar todos los métodos de la API pública y las firmas en la capa C++.
- StreamCallbackProvidery ClientCallbackProvider — Estos objetos exponen las devoluciones de llamada específicas de la transmisión y del cliente, y la capa C++ del SDK expone el resto de las devoluciones de llamada. Este es el patrón de devolución de llamada preferido para interactuar con el SDK Producer.

El siguiente diagrama ilustra el modelo de objeto de los objetos de devolución de llamada:

En el diagrama anterior, `DefaultCallbackProvider` se deriva de `CallbackProvider` (que expone todas las devoluciones de llamada en el PIC) y contiene `StreamCallbackProvider` y `ClientCallbackProvider`.

Este tema contiene las siguientes secciones:

- ClientCallbackProvider
- StreamCallbackProvider
- ClientCallbacks estructura
- Implementaciones de callback para volver a intentar la transmisión

ClientCallbackProvider

El objeto `ClientCallbackProvider` expone funciones de devolución de llamada de nivel de cliente. Los detalles de las funciones se describen en la sección the section called “ClientCallbacks”.

Métodos de devolución de llamada:

- `getClientReadyCallback`— Indica que el cliente está preparado.
- `getStorageOverflowPressureCallback`— Reporta el desbordamiento o la presión del almacenamiento. Esta devolución de llamada se invoca cuando el uso de almacenamiento cae por debajo del valor `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, que es el 5 por ciento del tamaño de almacenamiento total. Para obtener más información, consulte StorageInfo.

StreamCallbackProvider

El objeto `StreamCallbackProvider` expone funciones de devolución de llamada de nivel de transmisión.

Métodos de devolución de llamada:

- `getDroppedFragmentReportCallback`: notifica un fragmento eliminado.
- `getDroppedFrameReportCallback`— Indica que se ha caído un fotograma.
- `getFragmentAckReceivedCallback`— Indica que se ha recibido un fragmento de ACK para la transmisión.
- `getStreamClosedCallback`— Indica el estado de cierre de una transmisión.

- `getStreamConnectionStaleCallback`— Indica una condición de conexión obsoleta. En esta condición, el productor envía datos al servicio pero no recibe confirmaciones.
- `getStreamDataAvailableCallback`— Indica que los datos están disponibles en la transmisión.
- `getStreamErrorReportCallback`— Indica una condición de error en la transmisión.
- `getStreamLatencyPressureCallback`— Indica una condición de latencia de transmisión, que ocurre cuando el tamaño del búfer acumulado es mayor que el `max_latency` valor. Para obtener más información, consulte StreamDefinition/StreamInfo.
- `getStreamReadyCallback`: —Indica una condición de preparación para la transmisión.
- `getStreamUnderflowReportCallback`— Indica el estado de desbordamiento de un arroyo. Esta función no se utiliza actualmente y está reservada para su uso futuro.

Para ver el código fuente de`StreamCallbackProvider`, consulte StreamCallbackProvider.h.

ClientCallbacks estructura

La estructura de `ClientCallbacks` contiene los puntos de entrada de la función de devolución de llamada que invoca el PIC cuando se producen eventos específicos. La estructura también contiene información de versión en el campo `CALLBACKS_CURRENT_VERSION` y un campo `customData` para datos definidos por el usuario que se devuelve con las funciones de devolución de llamada individuales.

La aplicación cliente puede utilizar un puntero `this` para que el campo `custom_data` asigne funciones de miembros a las funciones `ClientCallback` estáticas en tiempo de ejecución, tal como se muestra en el siguiente código de ejemplo:

```
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(...);
```

Eventos

Función	Descripción	Tipo
`CreateDeviceFunc`	Actualmente no implement ado en el backend. Esta llamada devuelve un error cuando llama desde Java o C ++. Otros clientes realizan la inicialización específica de la plataforma.	API de backend
`CreateStreamFunc`	Se invoca cuando se crea la secuencia.	API de backend
`DescribeStreamFunc`	Se invoca cuando se invoca a `DescribeStream` .	API de backend
`GetStreamingEndpoi ntFunc`	Se invoca cuando se invoca a `GetStreamingEndpoint` .	API de backend
`GetStreamingTokenF unc`	Se invoca cuando se invoca a `GetStreamingToken` .	API de backend
`PutStreamFunc`	Se invoca cuando se invoca a `PutStream` .	API de backend
`TagResourceFunc`	Se invoca cuando se invoca a `TagResource` .	API de backend
`CreateMutexFunc`	Crea una exclusión mutua de sincronización.	Sincronización
`FreeMutexFunc`	Libera la exclusión mutua.	Sincronización
`LockMutexFunc`	Bloquea la exclusión mutua de sincronización.	Sincronización

Función	Descripción	Tipo
`TryLockMutexFunc`	Intenta bloquear la exclusión mutua. No implementada actualmente.	Sincronización
`UnlockMutexFunc`	Desbloquea la exclusión mutua.	Sincronización
`ClientReadyFunc`	Se invoca cuando el cliente entra en estado listo.	Notificación
`DroppedFrameReportFunc`	Notifica cuando se elimina un fotograma.	Notificación
`DroppedFragmentReportFunc`	Notifica cuando se elimina un fragmento. Esta función no se utiliza actualmente y está reservada para su uso futuro.	Notificación
`FragmentAckReceivedFunc`	Se invoca cuando se recibe un ACK (almacenado en búfer, recibido, almacenado de forma persistente y error) de fragmento.	Notificación
`StorageOverflowPressureFunc`	Se invoca cuando el uso de almacenamiento cae por debajo del valor `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, que se define como el 5 por ciento del tamaño de almacenamiento total.	Notificación

Función	Descripción	Tipo
`StreamClosedFunc`	Se invoca cuando se transmite n los últimos bits de los fotogramas restantes.	Notificación
`StreamConnectionStaleFunc`	Se invoca cuando la transmisión entra en un estado de conexión obsoleta. En esta condición, el productor envía datos al servicio pero no recibe reconocimientos.	Notificación
`StreamDataAvailableFunc`	Se invoca cuando los datos de transmisión están disponibles.	Notificación
`StreamErrorReportFunc`	Se invoca cuando se produce un error de transmisión. El PIC cierra automáticamente la transmisión con esta condición .	Notificación
`StreamLatencyPressureFunc`	Se invoca cuando la transmisión entra en una condición de latencia, que es cuando el tamaño de búfer acumulado es mayor que el valor `max_latency` . Para obtener más información, consulte StreamDefinition/StreamInfo.	Notificación
`StreamReadyFunc`	Se invoca cuando la transmisión entra en estado listo.	Notificación
`StreamUnderflowReportFunc`	Esta función no se utiliza actualmente y está reservada para su uso futuro.	Notificación

Función	Descripción	Tipo
`DeviceCertToTokenFunc`	Devuelve el certificado de conexión como un token.	Integración en la plataforma
`GetCurrentTimeFunc`	Devuelve la hora actual.	Integración en la plataforma
`GetDeviceCertificateFunc`	Devuelve el certificado del dispositivo. Esta función no se utiliza actualmente y está reservada para su uso futuro.	Integración en la plataforma
`GetDeviceFingerprintFunc`	Devuelve la huella del dispositivo. Esta función no se utiliza actualmente y está reservada para su uso futuro.	Integración en la plataforma
`GetRandomNumberFunc`	Devuelve un número aleatorio entre 0 y `RAND_MAX`.	Integración en la plataforma
`GetSecurityTokenFunc`	Devuelve el token de seguridad que se pasa a las funciones que se comunican con la API de backend. La implementación puede especificar el `AccessKeyId` en serie, `SecretKeyId` y el token de la sesión.	Integración en la plataforma
`LogPrintFunc`	Registra un línea de texto con la etiqueta y el nivel de registro. Para obtener más información, consulte `PlatformUtils.h` .	Integración en la plataforma

Para las funciones de integración en la plataforma de la tabla anterior, el último parámetro es una estructura `ServiceCallContext`, que tiene los siguientes campos:

- `version`: la versión de la estructura.
- `callAfter`: tiempo absoluto tras el que se invoca la función.
- `timeout`: el tiempo de espera de la operación en unidades de 100 nanosegundos.
- `customData`: un valor definido por el usuario que transferir de vuelta al cliente.
- `pAuthInfo`: las credenciales para la llamada. Para obtener más información, consulte la siguiente estructura (`__AuthInfo`).

La información de autorización se proporciona mediante la estructura `__AuthInfo`, que pueden ser credenciales serializadas o un token de autenticación específico del proveedor. Esta estructura tiene los campos siguientes:

- `version`: la versión de la estructura `__AuthInfo`.
- `type`: un valor `AUTH_INFO_TYPE` que define el tipo de la credencial (certificado o token de seguridad).
- `data`: una matriz de bytes que contiene la información de autenticación.
- `size`: el tamaño del parámetro `data`.
- `expiration`: el vencimiento de las credenciales en unidades de 100 nanosegundos.

Implementaciones de callback para volver a intentar la transmisión

El SDK de Kinesis Video Producer proporciona el estado de streaming a través de funciones de devolución de llamada. Te recomendamos que implementes los siguientes mecanismos de devolución de llamadas para recuperarte de cualquier problema momentáneo de red que se produzca durante la transmisión.

- Retorno de llamada por presión de latencia de transmisión: este mecanismo de devolución de llamada se inicia cuando el SDK detecta una condición de latencia de transmisión. Esto ocurre cuando el tamaño del búfer acumulado es mayor que el valor de MAX_LATENCY. Cuando se crea la secuencia, la aplicación de streaming establece MAX_LATENCY en el valor predeterminado de 60 segundos. La implementación típica de esta devolución de llamada es restablecer la conexión. Puedes usar el ejemplo de implementación en https://github.com/awslabs/amazon-kinesis-video-streamsproducer-sdk-cpp-/ blob/master/kinesis-video-c-producer/src/source/

StreamLatencyStateMachine .c según sea necesario. Tenga en cuenta que no existe la opción de almacenar los marcos no entregados debido a una interrupción de la red en un almacenamiento secundario para rellenarlos.

- Retorno de llamada por inactividad de la transmisión: esta devolución de llamada se inicia cuando el productor puede enviar datos al servicio Amazon Kinesis Data Streams (enlace superior), pero no puede recuperar los acuses de recibo (ACK almacenado en búfer) a tiempo (el valor predeterminado es de 60 segundos). Según la configuración de la red, se puede iniciar la devolución de llamada por presión de latencia de la transmisión o la devolución de llamada por inactividad de la transmisión, o ambas. Al igual que en la devolución de llamada de presión de latencia de streaming, la implementación típica es restablecer la conexión e iniciar una nueva conexión para streaming. Puede utilizar el ejemplo de implementación en https://github.com/awslabs/amazon-kinesis-video-streams-producer- c/blob/master/src/source/ConnectionStaleStateMachine .c según sea necesario.
- Reclamación por error de transmisión: esta llamada se inicia cuando el SDK detecta un tiempo de espera en la conexión de red u otros errores durante la llamada a las llamadas al servicio de la API de KVS.
- Retrollamada de fotogramas descartados: esta llamada se inicia cuando el almacenamiento está lleno, ya sea debido a una velocidad de red lenta o a un error de transmisión. Si la velocidad de la red hace que se pierdan fotogramas, puede aumentar el tamaño del almacenamiento, reducir el tamaño del fotograma del vídeo o la velocidad de fotogramas para que coincida con la velocidad de la red.

Uso de metadatos de streaming con Kinesis Video Streams

Puede utilizar el SDK para productores de Amazon Kinesis Video Streams para incrustar metadatos a nivel de fragmentos individuales en una transmisión de vídeo de Kinesis. Los metadatos de Kinesis Video Streams son un par clave-valor mutable. Puede utilizarlos para describir el contenido del fragmento, incrustar las lecturas de los sensores asociadas que deben transferirse junto con el fragmento real o satisfacer otras necesidades personalizadas. Los metadatos están disponibles como parte de las operaciones de API de the section called “GetMedia” o de the section called “GetMediaForFragmentList”. Se almacena junto con los fragmentos durante todo el período de retención de la transmisión. Las aplicaciones que lo consumen pueden leer, procesar y reaccionar en función de los metadatos que utilizanVea la salida de las cámaras mediante la biblioteca de analizadores.

Hay dos formas de insertar los metadatos con los fragmentos de una secuencia:

- No persistente: puede añadir metadatos una sola vez o según las necesidades a los fragmentos de una transmisión, en función de los criterios empresariales específicos que se hayan aplicado. Un ejemplo es una cámara inteligente que detecta movimiento y añade metadatos a los fragmentos correspondientes que contienen el movimiento antes de enviarlos a su transmisión de vídeo de Kinesis. Puede aplicar los metadatos al fragmento en el siguiente formato: `Motion = true`.
- Persistente: puede adjuntar metadatos a fragmentos sucesivos y consecutivos de una transmisión en función de una necesidad continua. Un ejemplo es una cámara inteligente que envía las coordenadas de latitud y longitud actuales asociadas a todos los fragmentos que envía a su transmisión de vídeo de Kinesis. Puede aplicar los metadatos a todos los fragmentos en el siguiente formato: `Lat = 47.608013N , Long = -122.335167W`.

Puede asociar metadatos simultáneamente en los dos modos al mismo fragmento en función de las necesidades de la aplicación. Los metadatos insertados podrían incluir los objetos detectados, las actividades monitorizadas, las coordenadas GPS o cualquier otro dato personalizado que se desee asociar a los fragmentos de la secuencia. Los metadatos se codifican como pares de cadenas clave-valor.

Añadir metadatos a una transmisión de vídeo de Kinesis

Los metadatos que se añaden a una transmisión de vídeo de Kinesis se modelan como etiquetas MKV, que se implementan como pares clave-valor.

Los metadatos pueden ser transitorios, como los que marcan un evento dentro de la secuencia, o persistentes, como los que identifican los fragmentos en los que se está produciendo un evento concreto. Un elemento de metadatos persistente permanece y se aplica a cada fragmento consecutivo hasta que se cancele.

> **Note**
>
> Los elementos de metadatos que se añaden mediante el uso de Cargar a Kinesis Video Streams son distintos del etiquetado a nivel de flujo APIs implementado conthe section called “TagStream”, ythe section called “UntagStream”. the section called “ListTagsForStream”

API de metadatos de streaming

Puede utilizar las siguientes operaciones en el SDK del productor para implementar los metadatos de streaming.

Temas

- PIC
- SDK para productores de C++
- SDK para productores de Java
- Metadatos persistentes y no persistentes

PIC

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

SDK para productores de 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 para productores de Java

Puede utilizar el SDK para productores de Java para añadir metadatos a un `MediaSource` uso de`MediaSourceSink.onCodecPrivateData`:

```
void onFragmentMetadata(final @Nonnull String metadataName, final @Nonnull String
 metadataValue, final boolean persistent)
throws KinesisVideoException;
```

Metadatos persistentes y no persistentes

En el caso de los metadatos no persistentes, puede agregar varios elementos de metadatos con el mismo nombre. El SDK del productor recopila los elementos de metadatos de la cola de metadatos

hasta que se añaden al siguiente fragmento. La cola de metadatos se vacía a medida que los elementos de metadatos se aplican a la secuencia. Para repetir los metadatos, llame de nuevo a `putKinesisVideoFragmentMetadata` o a `putFragmentMetadata`.

En el caso de los metadatos persistentes, el SDK del productor recopila los elementos de metadatos de la cola de metadatos del mismo modo que en el caso de los metadatos no persistentes. Sin embargo, los elementos de metadatos no se eliminan de la cola cuando se añaden al siguiente fragmento.

Llamar a `putKinesisVideoFragmentMetadata` o `putFragmentMetadata` con `persistent` establecido como `true` provoca el siguiente comportamiento:

- Llamar a la API coloca el elemento de metadatos en la cola. Los metadatos se añaden como etiqueta de MKV a todos los fragmentos mientras el elemento está en la cola.
- Llamar a la API con el mismo nombre y con un valor diferente del de un elemento de metadatos añadido anteriormente sobrescribe el elemento.
- Llamar a la API con un valor vacío elimina (cancela) el elemento de metadatos de la cola de metadatos.

Reproducción de Kinesis Video Streams

Puede ver una transmisión de vídeo de Kinesis mediante los siguientes métodos:

- GetMedia— Puede utilizar la `GetMedia` API para crear sus propias aplicaciones para procesar Kinesis Video Streams. `GetMedia`es una API en tiempo real con baja latencia. Para crear un reproductor que lo use`GetMedia`, debes crearlo tú mismo. Para obtener información sobre cómo desarrollar una aplicación que muestre una transmisión de vídeo de Kinesis mediante`GetMedia`, consulte. Transmita usando la biblioteca de analizadores
- HLS: HTTP Live Streaming (HLS) es un protocolo de comunicaciones de transmisión multimedia basado en HTTP estándar del sector. Puede utilizar HLS para ver una transmisión de vídeo de Kinesis, ya sea para su reproducción en directo o para ver un vídeo archivado.

  Puede utilizar HLS para la reproducción en directo. La latencia suele oscilar entre 3 y 5 segundos, pero puede oscilar entre 1 y 10 segundos, según el caso de uso, el reproductor y las condiciones de la red. Puede utilizar un reproductor de terceros (como Video.js o Google Shaka Player) para mostrar la transmisión de vídeo proporcionando la URL de la sesión de streaming HLS, ya sea mediante programación o de forma manual. También puede reproducir vídeo introduciendo la URL de la sesión de streaming HLS en la barra de direcciones de los navegadores Apple Safari o Microsoft Edge.
- El MPEG-DASH (transmisión dinámica adaptativa a través de HTTP (DASH), también conocido como MPEG-DASH, es un protocolo de transmisión con velocidad de bits adaptativa que permite la transmisión de contenido multimedia de alta calidad a través de Internet desde servidores web HTTP convencionales.

  Puede utilizar MPEG-DASH para la reproducción en directo. La latencia suele oscilar entre 3 y 5 segundos, pero puede oscilar entre 1 y 10 segundos, según el caso de uso, el reproductor y las condiciones de la red. Puedes usar un reproductor de terceros (como dash.js o Google Shaka Player) para mostrar la transmisión de vídeo proporcionando la URL de la sesión de streaming de MPEG-DASH, ya sea mediante programación o manualmente.
- GetClip— Puedes usar la `GetClip` API para descargar un clip (en un MP4 archivo) que contenga el contenido multimedia archivado y bajo demanda de la transmisión de vídeo especificada durante el intervalo de tiempo especificado. Para obtener más información, consulte la referencia de las API de GetClip.

Temas

- Requisitos de la pista de reproducción de vídeo
- Reproducción de vídeo con HLS
- Reproducción de vídeo con MPEG-DASH

Requisitos de la pista de reproducción de vídeo

Amazon Kinesis Video Streams admite contenido multimedia codificado en varios formatos. Si la transmisión de vídeo de Kinesis utiliza un formato no compatible con uno de los cuatro que se APIs indican a continuación, utilice `GetMedia`o `GetMediaForFragmentList`, ya que no tienen limitaciones de tipo de pista.

Temas

- GetClip requisitos
- Obtenga los requisitos de DASHStreaming la URL de la sesión
- Obtenga los requisitos de HLSStreaming la URL de la sesión
- GetImages requisitos

GetClip requisitos

Para obtener más información acerca esta API, consulte `GetClip`.

Descripción de la pista 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.264	V_ MPEG/ISO/AVC	N/A	N/A
Vídeo H.264	V_ MPEG/ISO/AVC	Audio AAC	A_AAC
Vídeo H.264	V_ MPEG/ISO/AVC	Audio G.711 (solo A-Law)	A_MS/ACM
Vídeo H.265	V_ MPEGH/ISO/HEVC	N/A	N/A

Descripción de la pista 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.265	V_ MPEGH/ISO/HEVC	Audio AAC	A_AAC

> **Important**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. No se admiten cambios de CPD entre los fragmentos de destino del clip resultante. El CPD debe ser coherente en todos los medios consultados; de lo contrario, se devolverá un error.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. Se devuelve un error si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si se cambia una pista de audio AAC por una pista de audio A-Law.

Obtenga los requisitos de DASHStreaming la URL de la sesión

Para obtener más información acerca esta API, consulte `GetDASHStreamingSessionURL`.

Descripción de Track 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.264	V_ MPEG/ISO/AVC	N/A	N/A
Vídeo H.264	V_ MPEG/ISO/AVC	Audio AAC	A_AAC
Vídeo H.264	V_ MPEG/ISO/AVC	Audio G.711 (solo A-Law)	A_MS/ACM

Descripción de Track 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.264	V_ MPEG/ISO/AVC	Audio G.711 (solo U-Law)	A_MS/ACM
Audio AAC	A_AAC	N/A	N/A
Vídeo H.265	V_ MPEGH/ISO/HEVC	N/A	N/A
Vídeo H.265	V_ MPEGH/ISO/HEVC	Audio AAC	A_AAC

> **Important**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. Los cambios de CPD no se admiten durante una sesión de streaming. El CPD debe permanecer coherente en todos los medios consultados.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. La transmisión fallará si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si una pista de audio AAC pasa a ser una pista de audio A-Law.

Obtenga los requisitos de HLSStreaming la URL de la sesión

Para obtener más información acerca esta API, consulte `GetHLSStreamingSessionURL`.

HLS Mp4

Descripción de Track 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.264	V_ MPEG/ISO/AVC	N/A	N/A
Vídeo H.264	V_ MPEG/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	N/A	N/A
Vídeo H.265	V_ MPEGH/ISO/HEVC	N/A	N/A
Vídeo H.265	V_ MPEGH/ISO/HEVC	Audio AAC	A_AAC

HLS TS

Descripción de Track 1	ID de códec de la pista 1	Descripción de Track 2	ID del códec Track 2
Vídeo H.264	V_ MPEG/ISO/AVC	N/A	N/A
Vídeo H.264	V_ MPEG/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	N/A	N/A

> **Note**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. Tanto en el TS como en el CPD MP4, se admiten cambios durante una sesión de streaming. Por lo tanto, los fragmentos de una sesión pueden tener una información diferente en el CPD sin interrumpir la reproducción. Para cada sesión de streaming, solo se permiten 500 cambios de CPD.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. La transmisión fallará si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si una pista de audio AAC pasa a ser una pista de audio A-Law.

GetImages requisitos

Para obtener más información acerca esta API, consulte `GetImages`.

> **Note**
>
> El `GetImages` soporte debe contener una pista de vídeo en la pista 1.

Reproducción de vídeo con HLS

La transmisión en vivo HTTP (HLS) es un protocolo de comunicación de transmisión multimedia basado en HTTP estándar del sector. Puede utilizar HLS para ver una transmisión de vídeo de Kinesis, ya sea para su reproducción en directo o para ver un vídeo archivado.

Puede utilizar HLS para la reproducción en directo. La latencia suele oscilar entre 3 y 5 segundos, pero puede oscilar entre 1 y 10 segundos, según el caso de uso, el reproductor y las condiciones de la red. Puede utilizar un reproductor de terceros (como Video.js o Google Shaka Player) para mostrar la transmisión de vídeo proporcionando la URL de la sesión de streaming HLS, ya sea mediante programación o de forma manual. También puede reproducir vídeo introduciendo la URL de la sesión de streaming HLS en la barra de direcciones de los navegadores Apple Safari o Microsoft Edge.

Para ver una transmisión de vídeo de Kinesis mediante HLS, cree primero una sesión de streaming mediante Get SessionURL. HLSStreaming Esta acción devuelve una URL (que contiene un token de sesión) que permite tener acceso a la sesión HLS. A continuación, puede utilizar la URL en un reproductor multimedia o en una aplicación independiente para mostrar la transmisión.

> **Important**
>
> No todos los archivos multimedia enviados a Kinesis Video Streams se pueden reproducir mediante HLS. Consulte los the section called "GetHLSStreamingSessionURL" requisitos de carga específicos.

Temas

- Utilícela AWS CLI para recuperar la URL de una sesión de streaming de HLS
- Ejemplo: utilice HLS en HTML y JavaScript
- Solución de problemas de HLS

Utilícela AWS CLI para recuperar la URL de una sesión de streaming de HLS

El siguiente procedimiento muestra cómo utilizar la AWS CLI URL de una sesión de streaming HLS para una transmisión de vídeo de Kinesis.

Para obtener instrucciones de instalación, consulte la Guía del AWS Command Line Interface usuario. Tras la instalación, configúrelo AWS CLI con las credenciales y la región.

Como alternativa, abra el AWS CloudShell terminal, que lo tiene AWS CLI instalado y configurado. Para obtener más información, consulte la guía del usuario de AWS CloudShell.

Recupere el punto de enlace URL de HLS para su transmisión de vídeo de Kinesis.

1. Escriba lo siguiente en la terminal:

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

   Recibirás una respuesta como la siguiente:

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

2. Realice la solicitud de URL de la sesión de streaming de HLS al punto final devuelto.

   Live

   Para la reproducción en directo, la lista de reproducción multimedia del HLS se actualiza continuamente con los archivos multimedia más recientes a medida que están disponibles. Al reproducir este tipo de sesión en un reproductor multimedia, la interfaz de usuario suele mostrar una notificación «en directo», sin necesidad de controlar la posición en la ventana de reproducción que se va a mostrar.

   Asegúrese de subir contenido multimedia a esta transmisión cuando ejecute este 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

   En el caso de la reproducción en directo, la reproducción comienza a partir de una hora de inicio especificada. La lista de reproducción multimedia HLS también se actualiza continuamente con los archivos multimedia más recientes a medida que están disponibles. La sesión seguirá incluyendo el contenido multimedia recién ingerido hasta que caduque o hasta la hora de finalización especificada, lo que ocurra primero. Este modo resulta útil para poder iniciar la reproducción desde el momento en que se detecta un evento y continuar retransmitiendo en directo contenido multimedia que aún no se haya ingerido en el momento de la creación de la sesión.

   Determine una marca de tiempo de inicio.

   Para este ejemplo, utilizamos el formato de tiempo en segundos de Unix Epoch. Consulte la sección de marcas de tiempo de la Guía del AWS Command Line Interface usuario para obtener más información sobre el formato de marcas de tiempo.

   Consulte UnixTime.org para ver una herramienta de conversión.

   - 1708471800 es igual a las 15:30:00 GMT- 08:00 del 20 de febrero de 2024

En este ejemplo, no especificamos una marca de tiempo de finalización, lo que significa que la sesión seguirá incluyendo el contenido multimedia recién ingerido hasta que la sesión caduque.

Invoca la `GetHLSStreamingSessionURL` API con el modo de `LIVE_REPLAY` reproducción y un selector de fragmentos HLS especificado.

```
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

Para la reproducción bajo demanda, la lista de reproducción multimedia del HLS contiene el contenido multimedia especificado por el selector de fragmentos del HLS. Cuando este tipo de sesión se reproduce en un reproductor multimedia, la interfaz de usuario suele mostrar un control de barra para seleccionar la posición de visualización en la ventana de reproducción.

Para crear una URL para una sección determinada de la transmisión, primero determine las marcas horarias de inicio y finalización.

Para este ejemplo, utilizamos el formato de tiempo en segundos de Unix Epoch. Consulte la sección de marcas de tiempo de la Guía del AWS Command Line Interface usuario para obtener más información sobre el formato de marcas de tiempo.

Consulte UnixTime.org para ver una herramienta de conversión.

- 1708471800 es igual a las 15:30:00 GMT- 08:00 del 20 de febrero de 2024
- 1708471860 es igual a las 15:31:00 GMT- 08:00 del 20 de febrero de 2024

Invoca la `GetHLSStreamingSessionURL` API con el modo de `ON_DEMAND` reproducción y un selector de fragmentos HLS especificado.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \
```

```
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \
  --stream-name YourStreamName \
  --playback-mode ON_DEMAND \
  --hls-fragment-selector \

"FragmentSelectorType=SERVER_TIMESTAMP,TimestampRange={StartTimestamp=1708471800,EndTim
```

> **Note**
>
> Las marcas de tiempo deben tener una diferencia de 24 horas entre sí, tal y como se menciona en la documentación. the section called “HLSTimestampRange”

Recibirás una respuesta como la siguiente:

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

> **Important**
>
> No compartas ni guardes este token donde una entidad no autorizada pueda acceder a él. El token proporciona acceso al contenido de la transmisión. Proteja el token con las mismas medidas que utilizaría con sus AWS credenciales.

Puedes usar esta URL y cualquier reproductor HLS para ver la transmisión HLS.

Por ejemplo, utilice el reproductor multimedia VLC.

También puede reproducir la transmisión HLS introduciendo la URL de la sesión de transmisión HLS en la barra de ubicación de los navegadores Apple Safari o Microsoft Edge.

Ejemplo: utilice HLS en HTML y JavaScript

El siguiente ejemplo muestra cómo usar el AWS SDK para la versión JavaScript 2 para recuperar una sesión de streaming HLS para una transmisión de vídeo de Kinesis y reproducirla en una página web. El ejemplo muestra cómo reproducir vídeo en los siguientes reproductores:

- Video.js
- Google Shaka Player
- hls.js

Vea el código de ejemplo completo y la página web alojada en. GitHub Esta página web estática simplifica las pruebas y la experimentación con las salidas HLS y MPEG-DASH de la transmisión de vídeo de Amazon Kinesis. La página de ejemplo proporciona campos de entrada para los siguientes parámetros:

- Región de AWS: La región en la que se encuentra la transmisión de vídeo de Kinesis
- Nombre del flujo de vídeo de Kinesis Video Streams
- Modo de reproducción: el modo de reproducción HLS (LIVE, LIVE_REPLAY o ON_DEMAND)
- Tipo de selector de fragmentos: método utilizado para seleccionar fragmentos (SERVER_TIMESTAMP o PRODUCER_TIMESTAMP)
- Número de fragmento: el número de fragmento inicial (si corresponde)
- Formato de contenedor: el formato del contenedor multimedia (FRAGMENTED_ MP4 o MPEG_TS)

La aplicación recupera estos valores de los cuadros de entrada de la página HTML y los utiliza para crear la solicitud de una sesión de streaming HLS que se muestra en la página.

Guía de código sobre los temas:

- Importa el AWS SDK JavaScript para navegadores
- Configurar el cliente de Kinesis Video Streams
- Recupere el punto de conexión para la reproducción de HLS
- Configurar el cliente multimedia archivado de Kinesis Video Streams
- Recupere la URL de la sesión de streaming de HLS
- Muestre la transmisión HLS en la página web

Importa el AWS SDK JavaScript para navegadores

En la página web, incluye la siguiente etiqueta de script para importar el AWS SDK de la JavaScript versión 2 al proyecto.

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

Para obtener más información, consulta la JavaScript documentación del AWS SDK.

Configurar el cliente de Kinesis Video Streams

Para acceder a la transmisión de vídeo con HLS, primero cree y configure el cliente Kinesis Video Streams. Consulte Configuración de credenciales en un navegador web para ver otros métodos de autenticación.

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

La aplicación recupera los valores necesarios de los cuadros de entrada de la página HTML.

Recupere el punto de conexión para la reproducción de HLS

Utilice el cliente de Kinesis Video Streams para invocar the section called “GetDataEndpoint” la API y recuperar el punto final.

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

Este código almacena el punto final en la `hlsDataEndpoint` variable.

Configurar el cliente multimedia archivado de Kinesis Video Streams

En la configuración del cliente de contenido multimedia archivado de Kinesis Video Streams, especifique el punto de conexión que obtuvo en el paso anterior.

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

Recupere la URL de la sesión de streaming de HLS

Utilice el cliente multimedia archivado de Kinesis Video Streams para invocar the section called "GetHLSStreamingSessionURL" la API y recuperar la URL de reproducción del HLS.

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

Muestre la transmisión HLS en la página web

Cuando tenga la URL de la sesión de streaming HLS, proporciónesela al reproductor de vídeo. El método para proporcionar la URL al reproductor de vídeo es específico del reproductor utilizado.

Video.js

Haga lo siguiente para importar Video.js y sus clases CSS al script de nuestro navegador:

```
<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>
```

Cree un elemento `video` HTML para mostrar el vídeo:

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

Establezca la URL HLS como fuente del elemento de vídeo HTML:

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

Shaka

Haga lo siguiente para importar el reproductor Shaka de Google al script de nuestro navegador:

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

Crea un elemento `video` HTML para mostrar el vídeo:

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

Crea un reproductor Shaka especificando el elemento de vídeo y llama al método de carga:

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

hls.js

Haga lo siguiente para importar el archivo hls.js al script de nuestro navegador:

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

Cree un elemento `video` HTML para mostrar el vídeo:

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

Cree un reproductor hls.js, asígnele la URL HLS y dígale que se reproduzca:

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

Solución de problemas de HLS

En esta sección se describen los problemas que pueden surgir al utilizar HTTP Live Streaming (HLS) con Kinesis Video Streams.

Problemas

- La recuperación de la URL de la sesión de streaming HLS se ha realizado correctamente, pero la reproducción no se puede realizar en el reproductor de vídeo
- Latencia demasiado alta entre el productor y el reproductor

La recuperación de la URL de la sesión de streaming HLS se ha realizado correctamente, pero la reproducción no se puede realizar en el reproductor de vídeo

Esta situación se produce cuando puede recuperar correctamente una URL de una sesión de streaming HLS con `GetHLSStreamingSessionURL`, pero el vídeo no se reproduce cuando se proporciona la URL a un reproductor de vídeo.

Para solucionar este problema, pruebe lo siguiente:

- Determine si la transmisión de vídeo se reproduce en la consola de Kinesis Video Streams. Observe los errores que se muestren en la consola.
- Si la duración del fragmento es inferior a un segundo, auméntela a un segundo. Si la duración del fragmento es demasiado corta, es posible que el servicio limite el reproductor porque solicita fragmentos de vídeo con demasiada frecuencia.
- Compruebe que la URL de cada sesión de streaming HLS solo la esté utilizando un único reproductor. Si hay más de un reproductor utilizando la URL de una sesión de streaming HLS, el servicio puede recibir demasiadas solicitudes y limitarlas.

- Compruebe que el reproductor admita todas las opciones que haya especificado para la sesión de flujo de HLS Video Streams. Pruebe distintas combinaciones de valores para los parámetros siguientes:
  - `ContainerFormat`
  - `PlaybackMode`
  - `FragmentSelectorType`
  - `DiscontinuityMode`
  - `MaxMediaPlaylistFragmentResults`

  Algunos reproductores multimedia (como HTML5 los reproductores móviles) normalmente solo admiten HLS con el formato MP4 contenedor f. Es posible que otros reproductores multimedia (como Flash y reproductores personalizados) solo admitan HLS con el formato contenedor MPEG TS. Se recomienda experimentar con el `ContainerFormat` parámetro para iniciar la solución de problemas.
- Verifique que cada fragmento tenga un número coherente de pistas. Compruebe que los fragmentos de la transmisión no cambien entre tener una pista de audio y vídeo o solo una pista de vídeo. Compruebe también que los ajustes del codificador (resolución y velocidad de fotogramas) no cambien entre los fragmentos de cada pista.

Latencia demasiado alta entre el productor y el reproductor

Esta situación se produce cuando hay una latencia demasiado alta entre el momento en que se captura el vídeo y el momento en que se reproduce en el reproductor de vídeo.

El vídeo se reproduce a través de HLS en cada fragmento. Por lo tanto, la latencia no puede ser menor que la duración del fragmento. La latencia también incluye el tiempo necesario para almacenar los datos en el búfer y transferirlos. Si la solución requiere una latencia menor que un segundo, considere la posibilidad de utilizar la API `GetMedia` en su lugar.

Puede modificar los siguientes parámetros para reducir la latencia global, pero al ajustarlos, también puede reducirse la calidad de vídeo o puede aumentar la frecuencia de repetición de operaciones de almacenamiento en búfer.

- Duración del fragmento: la duración del fragmento es la cantidad de vídeo entre las divisiones de la transmisión, controlada por la frecuencia de los fotogramas clave generados por el codificador de vídeo. El valor recomendado es de un segundo. Una duración de los fragmentos más breve significa que hay que esperar menos tiempo a que finalice el fragmento antes de transmitir los

datos de vídeo al servicio. El servicio también procesa más rápido los fragmentos más cortos. Sin embargo, si la duración del fragmento es demasiado breve, aumenta la probabilidad de que el reproductor se quede sin contenido y tenga que detenerse y almacenar contenido en el búfer. Si la duración del fragmento es inferior a 500 milisegundos, el productor puede crear demasiadas solicitudes, lo que hace que el servicio las limite.

- Velocidad de bits: una transmisión de vídeo con una velocidad de bits más baja tarda menos en leerse, escribirse y transmitirse. Sin embargo, una transmisión de vídeo con una tasa de bits inferior normalmente tiene una menor calidad de vídeo.
- Recuento de fragmentos en las listas de reproducción multimedia: un reproductor sensible a la latencia solo debe cargar los fragmentos más recientes de una lista de reproducción multimedia. En su lugar, la mayoría de los reproductores comienzan con el fragmento más antiguo. Al reducir el número de fragmentos de la lista de reproducción, se reduce el tiempo de separación entre los fragmentos anteriores y los nuevos. Con un tamaño de lista de reproducción más pequeño, es posible omitir un fragmento durante la reproducción si se produce un retraso en la adición de nuevos fragmentos a la lista de reproducción o si el reproductor tarda en recibir una lista de reproducción actualizada. Recomendamos usar de 3 a 5 fragmentos y usar un reproductor que esté configurado para cargar solo los fragmentos más recientes de una lista de reproducción.
- Tamaño del búfer del reproductor: la mayoría de los reproductores de vídeo tienen una duración mínima del búfer configurable, normalmente con un valor predeterminado de 10 segundos. Para obtener la latencia más baja, puede establecer este valor en 0 segundos. Sin embargo, si lo hace, el reproductor volverá a almacenar el búfer si hay algún retraso en la producción de fragmentos, ya que el reproductor no tendrá ningún búfer para absorber el retraso.
- Reproductor «catch up»: los reproductores de vídeo no suelen captar automáticamente la reproducción hasta la parte delantera del búfer de vídeo si el búfer se llena, como cuando un fragmento retrasado provoca una acumulación de fragmentos pendientes de reproducción. Un reproductor personalizado puede evitarlo descartando fotogramas o aumentando la velocidad de reproducción (por ejemplo, a 1,1x) para alcanzar el principio del búfer. Esto hace que la reproducción se entrecorte o que aumente de velocidad mientras el reproductor recupera el tiempo perdido, y la repetición de operaciones de almacenamiento en búfer puede ser más frecuente debido a que el tamaño del búfer es pequeño.

Reproducción de vídeo con MPEG-DASH

Para ver una transmisión de vídeo de Kinesis mediante MPEG-DASH, primero debe crear una sesión de streaming mediante Get SessionURL. DASHStreaming Esta acción devuelve una URL (que

contiene un token de sesión) que permite tener acceso a la sesión MPEG-DASH. A continuación, puede utilizar la URL en un reproductor multimedia o en una aplicación independiente para mostrar la transmisión.

Una transmisión de vídeo de Amazon Kinesis tiene los siguientes requisitos para proporcionar vídeo a través de MPEG-DASH:

- Para conocer los requisitos de las pistas de reproducción de vídeo en streaming, consulte. the section called “Obtenga la URL de la sesión DASHStreaming”
- La retención de datos debe ser mayor o igual que 0.
- La pista de vídeo de cada fragmento debe contener los datos privados del códec en la codificación de vídeo avanzada (AVC) para el formato H.264 y en HEVC para el formato H.265. Para obtener más información, consulte la especificación especificación ISO/IEC 14496-15 para MPEG. Si desea obtener más información sobre cómo adaptar datos de transmisiones a un determinado formato, consulte Referencia de marcas de adaptación de capa de abstracción de red (NAL).
- La pista de audio (si la hay) de cada fragmento debe contener los datos privados del códec en formato AAC (especificación ISO/IEC 13818-7 para AAC) o en formato MS Wave.

Ejemplo: usar MPEG-DASH en HTML y JavaScript

El siguiente ejemplo muestra cómo recuperar una sesión de streaming MPEG-DASH para una transmisión de vídeo de Kinesis y reproducirla en una página web. El ejemplo muestra cómo reproducir vídeo en los siguientes reproductores:

- Google Shaka Player
- dash.js

Temas

- Configurar el cliente de Kinesis Video Streams para la reproducción de MPEG-DASH
- Recupere el terminal de contenido archivado de Kinesis Video Streams para reproducirlo en MPEG-DASH
- Recupere la URL de la sesión de streaming MPEG-DASH
- Muestre el vídeo en streaming con la reproducción MPEG-DASH
- Ejemplo completo

Configurar el cliente de Kinesis Video Streams para la reproducción de MPEG-DASH

Para acceder a la transmisión de vídeo con MPEG-DASH, primero cree y configure el cliente Kinesis Video Streams (para recuperar el punto de conexión del servicio) y el cliente multimedia archivado (para recuperar la sesión de streaming MPEG-DASH). La aplicación recupera los valores necesarios de los cuadros de entrada de la página 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);
```

Recupere el terminal de contenido archivado de Kinesis Video Streams para reproducirlo en MPEG-DASH

Una vez iniciados los clientes, recupere el punto final de contenido archivado de Kinesis Video Streams para poder recuperar la URL de la sesión de streaming MPEG-DASH de la siguiente manera:

```
 // 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);
```

Recupere la URL de la sesión de streaming MPEG-DASH

Cuando tengas el punto final del contenido archivado, llama a la API Get DASHStreaming SessionURL para recuperar la URL de la sesión de streaming de MPEG-DASH de la siguiente manera:

```
// 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);
```

Muestre el vídeo en streaming con la reproducción MPEG-DASH

Cuando tenga la URL de la sesión de streaming MPEG-DASH, proporciónesela al reproductor de vídeo. El método para proporcionar la URL al reproductor de vídeo es específico del reproductor utilizado.

En el siguiente ejemplo de código, se muestra cómo proporcionar la URL de la sesión de streaming a un reproductor Google Shaka:

```
// 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');
}
```

En el siguiente ejemplo de código, se muestra cómo proporcionar la URL de la sesión de streaming a un reproductor 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');
```

```
}
```

Ejemplo completo

Puede descargar o ver el código de ejemplo completo en GitHub.

Configurar notificaciones en Kinesis Video Streams

Cuando un fragmento multimedia está disponible para su consumo, Kinesis Video Streams notifica a los clientes Amazon Simple Notification Service mediante notificaciones (Amazon SNS).

> **Note**
>
> Amazon Kinesis Video Streams utiliza Amazon SNS Standard Topics para la comunicación. Los temas de FIFO no se admiten actualmente.

En los temas siguientes se explica cómo empezar con las notificaciones.

Temas

- Administra las configuraciones de notificaciones
- Acerca de las etiquetas MKV del productor
- Mensajes de Amazon SNS

Administra las configuraciones de notificaciones

Para gestionar las configuraciones de notificaciones, utilice `UpdateNotificationConfiguration` y`DescribeNotificationConfiguration`. Para obtener más información, consulte las secciones siguientes.

UpdateNotificationConfiguration

Utilice esta operación de API para actualizar la información de notificación de una transmisión. Para obtener más información sobre la `UpdateNotificationConfiguration` función, consulte UpdateNotificationConfiguration la Guía para desarrolladores de Amazon Kinesis Video Streams.

> **Note**
>
> Tras actualizar la configuración de la notificación, se tarda al menos un minuto en iniciar la notificación. Espere al menos un minuto antes de realizar la `PutMedia` llamada de actualización.

DescribeNotificationConfiguration

Usa esta API para describir una configuración de notificaciones adjunta a una transmisión. Para obtener más información sobre la `DescribeNotificationConfiguration` función, consulte DescribeNotificationConfigurationla Guía para desarrolladores de Amazon Kinesis Video Streams.

Acerca de las etiquetas MKV del productor

Puede usar el SDK para productores de Kinesis Video Streams para etiquetar fragmentos específicos de interés exponiendo una operación de API en el SDK. Consulte un ejemplo de cómo funciona esto en esta sección del código. Al llamar a esta API, el SDK añadirá un conjunto de etiquetas MKV predefinidas junto con los datos del fragmento. Kinesis Video Streams reconocerá estas etiquetas MKV especiales e iniciará notificaciones para los fragmentos etiquetados.

Todos los metadatos de fragmentos proporcionados junto con las etiquetas MKV de notificación se publicarán como parte de la carga útil del tema Amazon SNS.

Sintaxis de las etiquetas MKV del productor

```
|+ 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
```

Límites de etiquetas MKV

En la siguiente tabla se enumeran las limitaciones asociadas a las etiquetas de metadatos. Si el límite de etiquetas de metadatos es ajustable, puedes solicitar un aumento a través de tu administrador de cuentas.

Límite	Valor máximo	Ajustable
Longitud de clave de metadatos opcional	128	No
Longitud del valor de los metadatos opcional	256	No
Número máximo de metadatos opcionales	10	Sí

Mensajes de Amazon SNS

Este tema contiene más información sobre los mensajes de Amazon SNS y las cargas útiles de los temas.

Temas

- Carga útil del tema Amazon SNS
- Ver tus mensajes de Amazon SNS

Carga útil del tema Amazon SNS

Cualquier notificación iniciada a través del flujo de trabajo anterior entregará la carga útil del tema de Amazon SNS, como se muestra en el siguiente ejemplo. Este ejemplo es un mensaje de Amazon SNS que se produce después de consumir datos de notificación de una cola 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
    }
}
```

Ver tus mensajes de Amazon SNS

No puedes leer los mensajes directamente de un tema de Amazon SNS porque no hay una API para hacerlo. Para ver los mensajes, suscriba una cola de SQS al tema de SNS o elija cualquier otro destino compatible con Amazon SNS. Sin embargo, la opción más eficaz para ver los mensajes es utilizar Amazon SQS.

Para ver sus mensajes de Amazon SNS mediante Amazon SQS

1. Cree una cola de Amazon SQS.
2. Desde AWS Management Console, abre el tema Amazon SNS establecido como destino en. `NotificationConfiguration`
3. Elija Crear suscripción y, a continuación, elija la cola de Amazon SQS creada en el primer paso.
4. Ejecute una `PutMedia` sesión con la configuración de notificaciones habilitada y con las etiquetas MKV de notificación añadidas a los fragmentos.
5. Elija la cola de Amazon SQS en la consola de Amazon SQS y, a continuación, seleccione Enviar y recibir mensajes para la cola de Amazon SQS.

6. Busca mensajes. Este comando debería mostrar todas las notificaciones generadas por la `PutMedia` sesión. Para obtener información sobre las votaciones, consulte Sondeos cortos y largos de Amazon SQS.

Extrae imágenes de transmisiones de vídeo

Puede utilizar Amazon Kinesis Video APIs Streams para realizar extracciones de imágenes bajo demanda SDKs y extracciones automatizadas de imágenes en tiempo real a partir de sus transmisiones de vídeo de Kinesis. Puede utilizar estas imágenes para aplicaciones de reproducción mejoradas, como miniaturas o limpieza mejorada, o para utilizarlas en flujos de trabajo de aprendizaje automático.

Kinesis Video Streams permite extraer imágenes de transmisiones de vídeo de dos maneras:

- Generación de imágenes bajo demanda: utilice la `the section called “GetImages”` API para extraer una o varias imágenes del vídeo almacenado en Kinesis Video Streams.
- the section called “Generación de imágenes en tiempo real”- Configure Kinesis Video Streams para que extraiga automáticamente imágenes de los datos de vídeo en tiempo real en función de las etiquetas de fragmentos del vídeo a medida que se ingiera y entregue las imágenes a un bucket de S3.

Generación automatizada de imágenes en tiempo real

Amazon Kinesis Video Streams ofrece la capacidad de transcodificar y entregar imágenes. Kinesis Video Streams extrae automáticamente imágenes de los datos de vídeo en tiempo real y las envía al bucket de Amazon S3 que especifique. La implementación de la extracción de imágenes automatizada y en tiempo real implica los siguientes pasos:

- Crear un depósito S3 para recibir las imágenes generadas.
- Configurar la propiedad de ImageGenerationConfigurationtransmisión que indica a Kinesis Video Streams cómo crear las imágenes y dónde enviarlas.
- Añadir etiquetas de generación de imágenes: Kinesis Video Streams solo genera imágenes a partir de fragmentos que tengan la etiqueta de generación de imágenes. Estas etiquetas se añaden al cargar vídeos mediante el SDK de Kinesis Video Streams Producer junto con `putKinesisVideoEventMetadata` el método.

Los siguientes procedimientos proporcionan instrucciones para completar cada uno de estos pasos.

Si utiliza una clave gestionada por el cliente, asegúrese de que la función que realiza las `PutMedia` llamadas (persona que carga) tenga los siguientes permisos necesarios para cifrar y descifrar datos y para acceder al bucket de Amazon S3.

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

Para obtener más información, consulte the section called "¿Cómo puedo comenzar a usar el cifrado en el servidor?".

Para configurar el destino de las imágenes generadas

1. Cree un depósito de destino de S3 al que se enviarán las imágenes.

   Siga la Guía del usuario de Amazon S3 para crear un bucket de Amazon S3.

   Anote el URI del bucket, que necesitará en el siguiente paso al actualizar la configuración de generación de imágenes de la transmisión.
2. Comprueba que lo tienes AWS CLI instalado y configurado. Para obtener más información, consulte la Guía del AWS Command Line Interface usuario de la versión 2.
3. Cree un nuevo archivo llamado `update-image-generation-input.json` con el siguiente contenido como entrada. Actualice los valores de los marcadores de posición con los valores que desee usar. Para conocer los valores máximos y mínimos admitidos, consulta la 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. Actualice la configuración de generación de imágenes de la transmisión mediante la the section called “UpdateImageGenerationConfiguration” API y adjunte el archivo JSON como entrada, como se muestra en el siguiente comando. Tenga en cuenta que la ruta del archivo apunta al archivo del directorio actual.

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

5. En caso de éxito, se devuelve una respuesta vacía y no se imprime nada en el terminal.

> **Note**
>
> Tras actualizar la configuración de generación de imágenes, se tarda al menos 1 minuto en iniciar el flujo de trabajo de generación de imágenes. Espera al menos 1 minuto antes de subir el vídeo a tu transmisión.

6. Verifica los ajustes de configuración. AWS CLI Utilízala para llamar a la the section called “DescribeImageGenerationConfiguration” API de tu transmisión.

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

Kinesis Video Streams solo generará y entregará imágenes para los fragmentos que tengan la etiqueta de generación de imágenes. Todos los metadatos de fragmentos adicionales que se proporcionen junto con las etiquetas de generación de imágenes de Amazon S3 se guardarán como metadatos de Amazon S3.

> **Note**
>
> Las etiquetas de generación de imágenes se refieren a etiquetas de metadatos fragmentados y no a etiquetas a nivel de transmisión.

> **Important**
>
> Las etiquetas de generación de imágenes se tienen en cuenta para el límite de etiquetas de fragmentos y metadatos. Para obtener más información, consulte the section called "Cuotas del servicio de metadatos de streaming".

A continuación, se muestra un ejemplo del aspecto de la estructura de las etiquetas de metadatos de los fragmentos con la `mkvinfo` utilidad. La etiqueta de generación de imágenes es una etiqueta MKV simple con una clave igual `AWS_KINESISVIDEO_IMAGE_GENERATION` y ningún valor. Para obtener más información, consulte el ejemplo de etiquetas de vídeo en la documentación de 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
```

Añadir etiquetas de generación de imágenes a los fragmentos

Kinesis Video Streams genera y entrega imágenes solo para los fragmentos que tienen la etiqueta de generación de imágenes. Kinesis Video Streams reconoce estas etiquetas MKV especiales e inicia el flujo de trabajo de generación de imágenes en función de la configuración de procesamiento de imágenes de la transmisión.

Cuando utilice el SDK de Kinesis Video Streams Producer para cargar contenido multimedia, utilice `putKinesisVideoEventMetadata` el método para añadir la etiqueta de generación de imágenes a cada fragmento que desee etiquetar. Cuando `putFrame` se invoca, un fragmento nuevo comienza con un marco que contiene la `keyframe` bandera.

Si vas a subir un vídeo pregrabado, es posible que se cargue a una velocidad diferente a la que se grabó, en función de la velocidad de la red. Le recomendamos que utilice la marca de tiempo del productor para configurar la generación de imágenes si desea generar imágenes a intervalos regulares en función de las marcas de tiempo originales del vídeo y no utilice las marcas de tiempo del servidor generadas en función de la velocidad a la que Amazon Kinesis Video Streams recibió el vídeo.

Para ver un ejemplo completo de este código, consulte el ejemplo de código en. `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);
```

Los elementos del código de ejemplo para configurar los marcos de muestra se explican a continuación:

1. Cada fragmento debe terminar con un final de fragment (`eofr`). Esta afirmación dice que cada vez que se reciba un nuevo fotograma clave, que señale el comienzo del siguiente fotograma, hay que poner una `eofr` antes de añadir el siguiente fotograma a la secuencia.
2. Coloca el fotograma actual en la secuencia.
3. Añada la etiqueta de generación de imágenes. Se puede llamar al `putKinesisVideoEventMetadata` método en cualquier momento después de la `putFrame(keyFrame)` llamada y antes de`putFrame(eofr)`. Solo se debe llamar como máximo una vez por fragmento. Como cada fragmento tendrá solo un fotograma clave, lo llamamos en este momento por motivos de simplicidad. `putKinesisVideoEventMetadata`Se comprueba si el código devuelto es correcto (0).

4. Añada otros metadatos de fragmentos personalizados, que Kinesis Video Streams transformará en metadatos de objetos de Amazon S3.
5. Finalice el fragmento final en esta sesión de carga.

Uso de las muestras para añadir etiquetas de generación de imágenes

Puede utilizarla `kvs_gstreamer_audio_video_sample` en el SDK de C++ Producer si desea una opción de línea de comandos para añadir etiquetas de generación de imágenes. Para activar esta función, añada el `-e both` argumento `-e image` o, tal y como se muestra en el siguiente ejemplo.

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

Para obtener más información sobre este ejemplo de aplicación, consulte el productor de CPP de Amazon Kinesis Video Streams GStreamer , el complemento y el archivo README de JNI en. GitHub

Ruta de objetos de Amazon S3 (imagen)

La ruta del objeto de S3 describe la ubicación en el depósito de S3 configurado donde se entregará la imagen generada. Utiliza el siguiente formato:

```
ImagePrefix_AccountID_StreamName_ImageTimecode_RandomID.file-extension
```

Los elementos de la ruta del objeto se definen de la siguiente manera:

- `ImagePrefix`- Valor de `AWS_KINESISVIDEO_IMAGE_PREFIX` si está presente.
- `AccountID`- El Cuenta de AWS ID con el que se crea la transmisión.
- `StreamName`- Nombre de la secuencia a partir de la cual se genera la imagen.
- `ImageTimecode`- Código temporal de época (en milisegundos) del fragmento en el que se genera la imagen.
- `RandomID`- GUID aleatorio.
- `file-extension`- JPG o PNG según el formato de imagen solicitado.

En este ejemplo, la ruta del objeto de las imágenes generadas tendrá el siguiente aspecto:

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

Recuperando los metadatos de la imagen

Puede usar la consola S3 o la CLI para recuperar los metadatos de las imágenes generadas.

Kinesis Video Streams establece el número de fragmento, la marca horaria del productor y del servidor y los metadatos del tipo de contenido de la imagen generada, todos ellos formateados como metadatos de objetos de Amazon S3. Si hay etiquetas MKV adicionales, esas etiquetas también se añadirán como metadatos de objetos de Amazon S3. El siguiente ejemplo muestra cómo utilizar el comando de la API head-object de Amazon S3 para recuperar los metadatos del objeto. La respuesta incluye los metadatos creados por 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"
    }
}
```

Para obtener más información sobre los metadatos de los objetos de S3, consulte https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingMetadata.html.

Recomendaciones de URI de Amazon S3 para protegerse contra la limitación

Si escribe miles de imágenes en Amazon S3, existe el riesgo de que se produzcan estrangulamientos. Para obtener más información, consulte Límites de solicitudes de venta de prefijos en S3.

Un prefijo de Amazon S3 comienza con un límite PUT de 3500 solicitudes PUT por segundo y aumentará gradualmente con el tiempo para prefijos únicos. Evite usar fechas y horas como prefijos de Amazon S3. Los datos codificados con el tiempo afectarán a los prefijos de uno en uno y también cambiarán con regularidad, lo que invalidará las ampliaciones de prefijos anteriores.

Para permitir un escalado de Amazon S3 más rápido y uniforme, recomendamos añadir un prefijo aleatorio, como un código hexadecimal o un UUID, al URI de destino de Amazon S3. Por ejemplo, los prefijos de código hexadecimal dividirán las solicitudes de forma aleatoria de forma natural entre 16 prefijos diferentes (un prefijo para cada carácter hexadecimal único), lo que permitirá recibir 56 000 solicitudes PUT por segundo después de que Amazon S3 haya escalado automáticamente.

Solución de problemas

Las imágenes no se enviaron al bucket de Amazon S3

Para solucionar este problema, hay algunas cosas que debes tener en cuenta:

1. Permisos ausentes
2. La configuración de generación de imágenes es incorrecta
3. La etiqueta no se agregó al fragmento

Permisos ausentes

Si utiliza una clave de KMS gestionada por el cliente, asegúrese de que la función que realiza las `PutMedia` llamadas (persona que carga) tenga los permisos de cifrado y descifrado adecuados y que tenga acceso al bucket de Amazon S3, de la siguiente manera:

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

Para obtener más información, consulte the section called “¿Cómo puedo comenzar a usar el cifrado en el servidor?”.

Compruebe el destino

AWS CLI Utilízala para llamar a la DescribeImageGenerationConfiguration API de tu transmisión.

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

Revisa `DestinationConfig` la respuesta y confirma que es correcta.

Compruebe que la etiqueta de generación de imágenes se haya añadido al fragmento

1. Compruebe que la `putKinesisVideoEventMetadata` llamada se ha realizado correctamente.

   El `putKinesisVideoEventMetadata` método devuelve un código de estado 0 en caso de éxito. Se recomienda comprobar si los valores devueltos por las funciones son 0. Si se devuelve un código de estado distinto de cero, conviértalo en hexadecimal y compruebe si desea the section called "Referencia de códigos de error" obtener más información.

   Asegúrese de tener activados los registros de errores y revíselos para ver si hay otros errores en la aplicación. Revise y compare el patrón de llamadas de presentación de marcos de su solicitud con la implementación recomendada:Añadir etiquetas de generación de imágenes a los fragmentos.

2. Verifique el archivo MKV generado localmente

   Confirme que el SDK del productor o la aplicación de muestra adjuntaron las etiquetas correctamente.

   a. Establezca la variable de entorno `KVS_DEBUG_DUMP_DATA_FILE_DIR`. Si se establece este valor, el SDK del productor escribe los archivos multimedia que habría enviado a Kinesis Video Streams en la ubicación especificada.

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

      > **Note**
      >
      > El SDK no creará un directorio nuevo si la ruta no existe. Cree la carpeta si es necesario.

   b. Vuelva a ejecutar la aplicación. Debería ver cómo se escriben `.mkv` los archivos en el directorio de salida especificado.

c. Verifica el contenido para asegurarte de que la etiqueta esté presente con MKVTool Nix u otro software.

   i. Instala MKVTool Nix: `brew install mkvtoolnix`

   ii. Ejecute MKVTool Nix con uno de los `.mkv` archivos del directorio de salida.

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

   iii. Revise el resultado de MKVTool Nix. Si el método SDK del `KinesisVideoStream::PutFragmentMetadata` productor se invocó correctamente, deberías ver la siguiente etiqueta MKV.

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

> **Note**
>
> Las etiquetas pertenecen al clúster anterior.

d. Si la etiqueta MKV no está presente, asegúrese de que se haya llamado al método SDK del `KinesisVideoStream::PutEventMetadata` productor con el `STREAM_EVENT_TYPE_IMAGE_GENERATION` argumento y de que haya devuelto un código de éxito (0).

Acceda a la analítica de vídeo

Esta sección contiene información sobre cómo acceder a los análisis de vídeo mediante la biblioteca de analizadores y Amazon CloudWatch.

Temas

- Consumir metadatos incrustados en una transmisión de vídeo de Kinesis
- Vea la salida de las cámaras mediante la biblioteca de analizadores
- Supervisión de Amazon Kinesis Video Streams
- Límites de metadatos de transmisión

Consumir metadatos incrustados en una transmisión de vídeo de Kinesis

Para consumir los metadatos de una transmisión de vídeo de Kinesis, utilice una implementación de: `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");
     }
    }
}
```

La interfaz se encuentra en la clase FragmentMetadataVisitor de la Vea la salida de las cámaras mediante la biblioteca de analizadores.

La clase `FragmentMetadataVisitor` contiene una implementación de `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 clase `KinesisVideoRendererExample` contiene un ejemplo del uso de un `BasicMkvTagProcessor`. En el siguiente ejemplo, se añade un `BasicMkvTagProcessor` a los `MediaProcessingArguments` de una aplicación:

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

Se llama al método `BasicMkvTagProcessor.process` cuando llegan los metadatos del fragmento. Puede recuperar los metadatos acumulados con `GetTags`. Para recuperar un único elemento de metadatos, primero llame `clear` para borrar los metadatos recopilados y, a continuación, vuelva a recuperar los elementos de metadatos.

Vea la salida de las cámaras mediante la biblioteca de analizadores

La biblioteca de analizadores de transmisión de vídeo de Kinesis es un conjunto de herramientas que puede utilizar en aplicaciones Java para consumir los datos MKV de una transmisión de vídeo de Kinesis.

La biblioteca incluye las siguientes herramientas:

- StreamingMkvReader: esta clase lee elementos MKV especificados a partir de una transmisión de vídeo.

- FragmentMetadataVisitor: esta clase recupera los metadatos de fragmentos (elementos multimedia) y pistas (transmisiones de datos individuales que contienen información de medios, como audio o subtítulos).
- OutputSegmentMerger: esta clase combina fragmentos consecutivos de una transmisión de vídeo.
- KinesisVideoExample: Esta es una aplicación de ejemplo que muestra cómo utilizar la biblioteca de analizadores de transmisiones de vídeo de Kinesis.

La biblioteca incluye también pruebas que muestran cómo se utilizan las herramientas.

Requisitos previos

Debe disponer de lo siguiente para examinar y utilizar la biblioteca de analizadores de transmisiones de vídeo de Kinesis:

- Una cuenta de Amazon Web Services (AWS). Si aún no tiene una Cuenta de AWS, consultethe section called “Inscríbase en una Cuenta de AWS”.
- Un entorno de desarrollo integrado (IDE) de Java, como Eclipse Java Neon o JetBrains IntelliJ Idea.
- Java 11, como Amazon Corretto 11.

Descarga el código

En esta sección, descargará la biblioteca de Java y el código de prueba, e importará el proyecto en su IDE Java.

Para comprobar los requisitos previos y otros detalles de este procedimiento, consulte the section called “Transmita usando la biblioteca de analizadores”.

1. Cree un directorio y clone el código fuente de la biblioteca desde el GitHub repositorio (https://github.com/aws/amazon-kinesis-video-streams-parser-library).

   ```
   git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
   ```

2. Abra el IDE de Java que esté utilizando (por ejemplo, Eclipse o IntelliJ IDEA) e importe el proyecto Apache Maven que ha descargado:

   - En Eclipse: seleccione File, Import..., Maven, Existing Maven Projects y vaya a la carpeta `kinesis-video-streams-parser-lib`.

- En IntelliJ Idea: seleccione Import. Vaya al archivo pom.xml en la raíz del paquete descargado.

Para obtener más información, consulte la documentación del IDE relacionada.

Examine el código

En esta sección, puede examinar la biblioteca de Java y el código de prueba y aprender a utilizar las herramientas de la biblioteca en su propio código.

La biblioteca del analizador de transmisiones de vídeo de Kinesis contiene las siguientes herramientas:

- StreamingMkvReader
- FragmentMetadataVisitor
- OutputSegmentMerger
- KinesisVideoExample

StreamingMkvReader

Esta clase lee elementos MKV especificados de una transmisión sin generar ningún bloqueo.

El siguiente ejemplo de código (de `FragmentMetadataVisitorTest`) muestra cómo crear y utilizar un `Streaming MkvReader` para recuperar objetos `MkvElement` de una transmisión de entrada llamada `inputStream`:

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

FragmentMetadataVisitor

Esta clase recupera los metadatos de los fragmentos (elementos multimedia) y rastrea los flujos de datos individuales que contienen información multimedia, como los datos privados del códec o el ancho o la altura de los píxeles.

El siguiente ejemplo de código (del archivo `FragmentMetadataVisitorTest`) muestra cómo utilizar `FragmentMetadataVisitor` para recuperar datos de un objeto `MkvElement`:

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
        StreamingMkvReader mkvStreamReader =
                StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));
        int segmentCount = 0;
        while(mkvStreamReader.mightHaveNext()) {
            Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
            if (mkvElement.isPresent()) {
                mkvElement.get().accept(fragmentVisitor);
                if
 (MkvTypeInfos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
 {
                    MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
                    Frame frame =
 ((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
                    MkvTrackMetadata trackMetadata =
 fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
                    assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
                }
                if
 (MkvTypeInfos.SEGMENT.equals(mkvElement.get().getElementMetaData().getTypeInfo())) {
                    if (mkvElement.get() instanceof MkvEndMasterElement) {
                        if (segmentCount < continuationTokens.size()) {
                            Optional<String> continuationToken =
 fragmentVisitor.getContinuationToken();
                            Assert.assertTrue(continuationToken.isPresent());
                            Assert.assertEquals(continuationTokens.get(segmentCount),
 continuationToken.get());
                        }
                        segmentCount++;
                    }
                }
            }

        }
```

El ejemplo anterior muestra el siguiente patrón de codificación:

- Cree una clase `FragmentMetadataVisitor` para analizar los datos y una clase StreamingMkvReader para proporcionar los datos.
- Para cada `MkvElement` de la transmisión, pruebe si sus metadatos son del tipo `SIMPLEBLOCK`.
- En caso afirmativo, recupere el `MkvDataElement` del `MkvElement`.
- Recupere el `Frame` (datos multimedia) del `MkvDataElement`.
- Recupere los `MkvTrackMetadata` para el `Frame` del `FragmentMetadataVisitor`.
- Recupere y compruebe los siguientes datos de los objetos `Frame` y `MkvTrackMetadata`:
  - El número de pista.
  - La altura en píxeles del fotograma.
  - La anchura en píxeles del fotograma.
  - El ID del códec para el códec que utilice para codificar el fotograma.
  - Que el fotograma haya llegado en orden. Compruebe que el número de pista del fotograma anterior, si está presente, sea inferior al del fotograma actual.

Para utilizar `FragmentMetadataVisitor` en su proyecto, transfiera los objetos `MkvElement` al visitante con su método `accept`:

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

Esta clase reúne los metadatos de diferentes pistas en la transmisión en una transmisión con un único segmento.

El siguiente ejemplo de código (del archivo `FragmentMetadataVisitorTest`) muestra cómo utilizar `OutputSegmentMerger` para combinar metadatos de pistas a partir de una matriz de bytes denominada `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);
    }
}
```

El ejemplo anterior muestra el siguiente patrón de codificación:

- Cree una FragmentMetadataVisitorpara recuperar los metadatos de la secuencia.
- Cree una transmisión de salida para recibir los metadatos combinados.
- Cree un `OutputSegmentMerger`, transfiriendo el `ByteArrayOutputStream`.
- Cree un `CompositeMkvElementVisitor` que contenga los dos visitantes.
- Cree un `InputStream` que apunte al archivo especificado.
- Combine cada elemento de los datos de entrada en la transmisión de salida.

KinesisVideoExample

Esta es una aplicación de ejemplo que muestra cómo utilizar la biblioteca de analizadores de transmisión de vídeo de Kinesis.

Esta clase realiza las operaciones siguientes:

- Crea una transmisión de vídeo de Kinesis. Si ya existe una transmisión con ese nombre, se elimina y se vuelve a crear.
- Llamadas PutMediapara transmitir fragmentos de vídeo a la transmisión de vídeo de Kinesis.
- Llamadas GetMediapara transmitir fragmentos de vídeo de la transmisión de vídeo de Kinesis.
- Utiliza StreamingMkvReader para analizar los fragmentos devueltos de la transmisión y utiliza FragmentMetadataVisitor para registrar los fragmentos.

Eliminación y nueva creación de la transmisión

El siguiente ejemplo de código (del `StreamOps.java` archivo) elimina una transmisión de vídeo de Kinesis determinada:

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

El siguiente ejemplo de código (del `StreamOps.java` archivo) crea una transmisión de vídeo de Kinesis con el nombre especificado:

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

Llama PutMedia

El siguiente ejemplo de código (del `PutMediaWorker.java` archivo) llama PutMediaa la transmisión:

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

Llama GetMedia

El siguiente ejemplo de código (del `GetMediaWorker.java` archivo) llama GetMediaa la transmisión:

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

Analiza el GetMedia resultado

En esta sección se describe cómo utilizar StreamingMkvReader, FragmentMetadataVisitor y `CompositeMkvElementVisitor` para analizar, guardar en archivo y registrar los datos devueltos de `GetMedia`.

Lea el resultado de GetMedia con StreamingMkvReader

El siguiente ejemplo de código (del `GetMediaWorker.java` archivo) crea un StreamingMkvReader y lo usa para analizar el resultado de la GetMediaoperación:

```
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);
}
```

En el ejemplo de código anterior, StreamingMkvReader recupera objetos `MKVElement` de la carga del resultado de `GetMedia`. En la siguiente sección, los elementos se pasan a un FragmentMetadataVisitor.

Recupera fragmentos con FragmentMetadataVisitor

Los siguientes ejemplos de código (de los archivos `KinesisVideoExample.java` y `StreamingMkvReader.java`) crean un FragmentMetadataVisitor. A continuación, los objetos `MkvElement` por los que ha iterado StreamingMkvReader se pasan al visitante con el método `accept`.

de *`KinesisVideoExample.java`*:

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

de *StreamingMkvReader.java*:

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

Registro de los elementos y escritura en un archivo

El siguiente ejemplo de código (del archivo `KinesisVideoExample.java`) crea los siguientes objetos y los devuelve como parte del valor de retorno de la función `GetMediaProcessingArguments`:

- Un `LogVisitor` (una extensión de `MkvElementVisitor`) que escribe en el log del sistema.
- Un `OutputStream` que escribe los datos entrantes en un archivo MKV.
- Un `BufferedOutputStream` que almacena en búfer los datos vinculados para `OutputStream`.
- Un the section called “OutputSegmentMerger” que fusiona los elementos consecutivos en el resultado de `GetMedia` con los mismos datos de pista y EBML.
- Un `CompositeMkvElementVisitor` que compone el FragmentMetadataVisitorthe section called “OutputSegmentMerger”, y `LogVisitor` en un único elemento: el visitante.

```
//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);
```

Luego, los argumentos de procesamiento multimedia se pasan a la`GetMediaWorker`, que a su vez pasa a la`ExecutorService`, que lleva al trabajador a un hilo diferente:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
        getCredentialsProvider(),
        getStreamName(),
        new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
        amazonKinesisVideo,
        getMediaProcessingArgumentsLocal.getMkvElementVisitor());
executorService.submit(getMediaWorker);
```

Ejecuta el código

La biblioteca de analizadores de transmisión de vídeo de Kinesis contiene herramientas diseñadas para que las utilice en sus propios proyectos. El proyecto contiene pruebas de unidades de las herramientas que puede ejecutar para verificar su instalación.

Las siguientes pruebas de unidades están incluidas en la biblioteca:

- mkv
  - `ElementSizeAndOffsetVisitorTest`
  - `MkvValueTest`
  - `StreamingMkvReaderTest`
- utilidades
  - `FragmentMetadataVisitorTest`
  - `OutputSegmentMergerTest`

Supervisión de Amazon Kinesis Video Streams

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de Amazon Kinesis Video Streams y AWS sus soluciones. Le recomendamos recopilar

datos de monitoreo de todas las partes de la AWS solución para ayudarlo a depurar un error multipunto, en caso de que se produzca. Antes de empezar a monitorizar Amazon Kinesis Video Streams, le recomendamos que cree un plan de supervisión que incluya respuestas a las siguientes preguntas:

- ¿Cuáles son los objetivos de la supervisión?
- ¿Qué recursos va a supervisar?
- ¿Con qué frecuencia va a supervisar estos recursos?
- ¿Qué herramientas de supervisión va a utilizar?
- ¿Quién se encargará de realizar las tareas de supervisión?
- ¿Quién debería recibir una notificación cuando surjan problemas?

Una vez que haya definido sus objetivos de monitoreo y creado su plan de monitoreo, el siguiente paso es establecer un punto de referencia para el rendimiento normal de Amazon Kinesis Video Streams en su entorno. Debe medir el rendimiento de Amazon Kinesis Video Streams en distintos momentos y en diferentes condiciones de carga. Cuando supervise Amazon Kinesis Video Streams, almacene un historial de los datos de monitoreo que haya recopilado. Puede comparar el rendimiento actual de Amazon Kinesis Video Streams con estos datos históricos para ayudarle a identificar los patrones de rendimiento normales y las anomalías de rendimiento, y a diseñar métodos para abordar los problemas que puedan surgir.

Temas

- Supervise las métricas de Amazon Kinesis Video Streams con CloudWatch
- Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch
- Registre las llamadas a la API de Amazon Kinesis Video Streams con AWS CloudTrail

Supervise las métricas de Amazon Kinesis Video Streams con CloudWatch

Puede monitorizar una transmisión de vídeo de Kinesis mediante Amazon CloudWatch, que recopila y procesa datos sin procesar de Amazon Kinesis Video Streams para convertirlos en métricas legibles y prácticamente en tiempo real. Estas estadísticas se registran durante un período de 15 meses para que pueda acceder a la información histórica y obtener una mejor perspectiva del rendimiento de su aplicación o servicio web.

En la consola de Amazon Kinesis Video Streams, puede CloudWatch ver las métricas de una transmisión de vídeo de Amazon Kinesis de dos maneras:

- En la página Dashboard (Panel), elija la pestaña Video streams (Transmisiones de vídeo) en la sección Account-level metrics for Current Region (Métricas de nivel de cuenta de la región actual).
- Selecciona la pestaña Monitoring (Monitoreo) en la página de detalles de la transmisión de vídeo.

Amazon Kinesis Video Streams proporciona las siguientes métricas:

Métrica	Descripción
`ArchivedFragmentsConsumed.Media`	El número de puntos de cuota de contenido multimedia fragmentado que consumieron todos los APIs. Para obtener una explicación del concepto de puntos de cuota, consultethe section called “Cuotas de fragmentos de metadatos y fragmentos de medios”. Unidades: recuento
`ArchivedFragmentsConsumed.Metadata`	El número de fragmentos de puntos de cuota de metadatos que consumieron todos los APIs. Para obtener una explicación del concepto de puntos de cuota, consultethe section called “Cuotas de fragmentos de metadatos y fragmentos de medios”. Unidades: recuento
`PutMedia.Requests`	El número de solicitudes de `PutMedia` API para una transmisión determinada. Unidades: recuento
`PutMedia.IncomingBytes`	El número de bytes recibidos como parte `PutMedia` de la transmisión. Unidades: bytes

Métrica	Descripción
`PutMedia.IncomingFragments`	El número de fragmentos completos recibidos como parte `PutMedia` de la transmisión. Unidades: recuento
`PutMedia.IncomingFrames`	El número de fotogramas completos recibidos como parte `PutMedia` de la transmisión. Unidades: recuento
`PutMedia.ActiveConnections`	Número total de conexiones al host de servicio. Unidades: recuento
`PutMedia.ConnectionErrors`	Los errores al establecer la `PutMedia` conexión para la transmisión. Unidades: recuento
`PutMedia.FragmentIngestionLatency`	La diferencia de tiempo entre el momento en que Amazon Kinesis Video Streams recibe el primer y el último byte de un fragmento. Unidades: milisegundos
`PutMedia.FragmentPersistLatency`	El tiempo transcurrido desde que se reciben y archivan los datos completos del fragmento. Unidades: recuento
`PutMedia.Latency`	La diferencia de tiempo entre la solicitud y la respuesta HTTP desde el momento en InletService que se estableció la conexión. Unidades: recuento

Métrica	Descripción
`PutMedia.BufferingAckLatency`	La diferencia de tiempo entre el momento en que Amazon Kinesis Video Streams recibe el primer byte de un fragmento nuevo y el momento en que se envía el ACK de almacenamiento en búfer para el fragmento. Unidades: milisegundos
`PutMedia.ReceivedAckLatency`	La diferencia de tiempo entre el momento en que Amazon Kinesis Video Streams recibe el último byte de un fragmento nuevo y el momento en que se envía el ACK recibido para el fragmento. Unidades: milisegundos
`PutMedia.PersistedAckLatency`	La diferencia de tiempo entre el momento en que Amazon Kinesis Video Streams recibe el último byte de un fragmento nuevo y el momento en que se envía el ACK persistente para el fragmento. Unidades: milisegundos
`PutMedia.ErrorAckCount`	El número de errores ACKs enviados `PutMedia` durante la transmisión. Unidades: recuento
`PutMedia.Success`	1 para cada fragmento escrito correctamente; 0 por cada fragmento que no se ha escrito. El valor promedio de esta métrica indica cuántos fragmentos completos y válidos se han enviado. Unidades: recuento
`GetMedia.Requests`	El número de solicitudes de `GetMedia` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetMedia.OutgoingBytes`	El número total de bytes enviados desde el servicio como parte de la `GetMedia` API para una transmisión determinada. Unidades: bytes
`GetMedia.OutgoingFragments`	El número de fragmentos enviados `GetMedia` durante la transmisión. Unidades: recuento
`GetMedia.OutgoingFrames`	El número de fotogramas enviados durante `GetMedia` una transmisión determinada. Unidades: recuento
`GetMedia.MillisBehindNow`	La diferencia de tiempo entre la marca de tiempo actual del servidor y la marca de tiempo del servidor del último fragmento enviado. Unidades: milisegundos
`GetMedia.ConnectionErrors`	Número de conexiones que no se establecieron correctamente. Unidades: recuento

Métrica	Descripción
`GetMedia.Success`	1 por cada fragmento enviado correctamente; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulte el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetMediaForFragmentList.OutgoingBytes`	El número total de bytes enviados desde el servicio como parte de la `GetMediaForFragmentList` API para una transmisión determinada. Unidades: bytes
`GetMediaForFragmentList.OutgoingFragments`	El número total de fragmentos enviados desde el servicio como parte de la `GetMediaForFragmentList` API para una transmisión determinada. Unidades: recuento
`GetMediaForFragmentList.OutgoingFrames`	El número total de fotogramas enviados desde el servicio como parte de la `GetMediaForFragmentList` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetMediaForFragmentList.Requests`	El número de solicitudes de `GetMediaForFragmentList` API para una transmisión determinada. Unidades: recuento
`GetMediaForFragmentList.Success`	1 por cada fragmento enviado correctamente; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`ListFragments.Latency`	La latencia de la `ListFragments` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`ListFragments.Requests`	El número de solicitudes de `ListFragments` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`ListFragments.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetHLSStreamingSessionURL.Latency`	La latencia de la `GetHLSStreamingSessionURL` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetHLSStreamingSessionURL.Requests`	El número de solicitudes de `GetHLSStreamingSessionURL` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetHLSStreamingSessionURL.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetHLSMasterPlaylist.Latency`	La latencia de la `GetHLSMasterPlaylist` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetHLSMasterPlaylist.Requests`	El número de solicitudes de `GetHLSMasterPlaylist` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetHLSMasterPlaylist.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetHLSMediaPlaylist.Latency`	La latencia de la `GetHLSMediaPlaylist` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetHLSMediaPlaylist.Requests`	El número de solicitudes de `GetHLSMediaPlaylist` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetHLSMediaPlaylist.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetMP4InitFragment.Latency`	La latencia de la `GetMP4InitFragment` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetMP4InitFragment.Requests`	El número de solicitudes de `GetMP4InitFragment` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetMP4InitFragment.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetMP4MediaFragment.Latency`	La latencia de la `GetMP4MediaFragment` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetMP4MediaFragment.Requests`	El número de solicitudes de `GetMP4MediaFragment` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetMP4MediaFragment.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetMP4MediaFragment.OutgoingBytes`	El número total de bytes enviados desde el servicio como parte de la `GetMP4MediaFragment` API para una transmisión determinada. Unidades: bytes
`GetTSFragment.Latency`	La latencia de la `GetTSFragment` API requiere el nombre de la transmisión dada. Unidades: milisegundos
`GetTSFragment.Requests`	El número de solicitudes de `GetTSFragment` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetTSFragment.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetTSFragment.OutgoingBytes`	El número total de bytes enviados desde el servicio como parte de la `GetTSFragment` API para una transmisión determinada. Unidades: bytes
`GetDASHStreamingSessionURL.Latency`	La latencia de la `GetDASHStreamingSessionURL` API requiere el nombre de la transmisión dada. Unidades: milisegundos
`GetDASHStreamingSessionURL.Requests`	El número de solicitudes de `GetDASHStreamingSessionURL` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetDASHStreamingSessionURL.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetDASHManifest.Latency`	La latencia de la `GetDASHManifest` API requiere el nombre de la transmisión determinada. Unidades: milisegundos
`GetDASHManifest.Requests`	El número de solicitudes de `GetDASHManifest` API para una transmisión determinada. Unidades: recuento

Métrica	Descripción
`GetDASHManifest.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetClip.Latency`	La latencia de la GetClip API requiere el nombre de la transmisión de vídeo dada. Unidades: milisegundos
`GetClip.Requests`	El número de solicitudes de GetClip API para una transmisión de vídeo determinada. Unidades: recuento

Métrica	Descripción
`GetClip.Success`	1 por cada solicitud correcta; 0 por cada error. El valor promedio indica la tasa de éxito. **Note** Los errores incluyen errores 400 (usuario) y errores 500 (sistema). Para obtener más información sobre cómo habilitar un resumen de las solicitudes y respuestas, incluida la AWS solicitud IDs, consulta el registro resumido de solicitudes y respuestas. Unidades: recuento
`GetClip.OutgoingBytes`	El número total de bytes enviados desde el servicio como parte de la GetClip API para una transmisión de vídeo determinada. Unidades: bytes

CloudWatch guía de métricas

CloudWatch las métricas pueden ayudar a encontrar respuestas a las siguientes preguntas:

Temas

- ¿Llegan los datos al servicio Amazon Kinesis Video Streams?
- ¿Por qué el servicio Amazon Kinesis Video Streams no ingiere correctamente los datos?
- ¿Por qué no se pueden leer los datos del servicio Amazon Kinesis Video Streams a la misma velocidad que los envía el productor?
- ¿Por qué no se ve vídeo en la consola, o por qué se reproduce el vídeo con retraso?
- ¿Cuál es el retraso en la lectura de los datos en tiempo real, y por qué se retrasa el cliente con relación al encabezado de la transmisión?
- ¿El cliente lee datos de la transmisión de vídeo de Kinesis y a qué velocidad?

- ¿Por qué el cliente no puede leer los datos de la transmisión de vídeo de Kinesis?

¿Llegan los datos al servicio Amazon Kinesis Video Streams?

Métricas relevantes:

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

Elementos de acción:

- Si estas métricas disminuyen, compruebe si la aplicación sigue enviando datos al servicio.
- Compruebe el ancho de banda de la red. Si el ancho de banda de la red es insuficiente, podría estar reduciendo la velocidad a la que el servicio recibe los datos.

¿Por qué el servicio Amazon Kinesis Video Streams no ingiere correctamente los datos?

Métricas relevantes:

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

Elementos de acción:

- Si se produce un aumento`PutMedia.ConnectionErrors`, observe la respuesta HTTP y los códigos de error recibidos por el cliente productor para ver qué errores se están produciendo al establecer la conexión.
- Si se produce una disminución `PutMedia.Success` o un aumento`PutMedia.ErrorAckCount`, consulta el código de error ack que aparece en las respuestas de ack enviadas por el servicio para ver por qué falla la ingesta de datos. Para obtener más información, consulte AckErrorCode.Values.

¿Por qué no se pueden leer los datos del servicio Amazon Kinesis Video Streams a la misma velocidad que los envía el productor?

Métricas relevantes:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Elementos de acción:

- Si estas métricas disminuyen, compruebe el ancho de banda de red de sus conexiones. Las conexiones con un ancho de banda bajo podrían ser la causa de que los datos lleguen al servicio a una velocidad menor.

¿Por qué no se ve vídeo en la consola, o por qué se reproduce el vídeo con retraso?

Métricas relevantes:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Elementos de acción:

- Si se produce un aumento `PutMedia.FragmentIngestionLatency` o una disminución`PutMedia.IncomingFragments`, comprueba el ancho de banda de la red y comprueba si los datos se siguen enviando.
- Si se produce una caída`PutMedia.Success`, comprueba los códigos de error del paquete. Para obtener más información, consulte AckErrorCode.Values.
- Si hay un aumento en `PutMedia.FragmentPersistLatency` o`ListFragments.Latency`, lo más probable es que tengas un problema con el servicio. Si el problema persiste durante un período prolongado, consulta con tu contacto del servicio de atención al cliente para ver si hay algún problema con el servicio.

¿Cuál es el retraso en la lectura de los datos en tiempo real, y por qué se retrasa el cliente con relación al encabezado de la transmisión?

Métricas relevantes:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Elementos de acción:

- Si hay un aumento`GetMedia.ConnectionErrors`, es posible que el consumidor se esté retrasando en la lectura de la transmisión debido a los frecuentes intentos de volver a conectarse a la transmisión. Compruebe los códigos de error/respuesta HTTP que se devuelven para la solicitud `GetMedia`.
- Si se produce una caída`GetMedia.Success`, es probable que se deba a que el servicio no puede enviar los datos al consumidor, lo que provocaría una interrupción de la conexión y la reconexión de los consumidores, lo que provocaría que el consumidor se quedara a la zaga de la transmisión.
- Si se produce un aumento`GetMedia.MillisBehindNow`, comprueba los límites de ancho de banda para ver si recibes los datos a un ritmo más lento debido a que el ancho de banda es menor.

¿El cliente lee datos de la transmisión de vídeo de Kinesis y a qué velocidad?

Métricas relevantes:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

Elementos de acción:

- Estas métricas indican la velocidad a la que se leen los datos archivados y en tiempo real.

¿Por qué el cliente no puede leer los datos de la transmisión de vídeo de Kinesis?

Métricas relevantes:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Elementos de acción:

- Si hay un aumento`GetMedia.ConnectionErrors`, observe la respuesta HTTP y los códigos de error devueltos por la `GetMedia` solicitud. Para obtener más información, consulta AckErrorCode.Values.
- Si estás intentando leer los datos más recientes o en tiempo real, comprueba `PutMedia.IncomingBytes` si están ingresando datos en la transmisión para que el servicio los envíe a los consumidores.
- Si se produce una caída `GetMedia.Success` o`GetMediaForFragmentList.Success`, probablemente, se deba a que el servicio no ha podido enviar los datos al consumidor. Si el problema persiste durante un período prolongado, consulta con tu contacto del servicio de atención al cliente para ver si hay algún problema con el servicio.

Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch

Puede monitorizar el Amazon Kinesis Video Streams Edge Agent con CloudWatch Amazon, que recopila y procesa datos sin procesar para convertirlos en métricas legibles y prácticamente en tiempo real. Estas estadísticas se registran durante un período de 15 meses. Con esta información histórica, puede obtener una mejor perspectiva del rendimiento de su aplicación web o del servicio Amazon Kinesis Video Streams Edge Agent.

Para ver las métricas, haga lo siguiente:

1. Inicie sesión en AWS Management Console y abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el menú de navegación de la izquierda, en Métricas, selecciona Todas las métricas.

3. Selecciona la pestaña Explorar y, a continuación, selecciona el espacio de nombres EdgeRuntimeAgentpersonalizado.

El agente Edge de Amazon Kinesis Video Streams publica las siguientes métricas en el espacio de nombres: `EdgeRuntimeAgent`

Dimensiones	Estado	Descripción
Nombre de la transmisión, `RecordJob`	Running	Se publica de forma continua cuando `RecordJob` está en ejecución. Unidades: ninguna. «1" se publica mientras `RecordJob` esté en este estado.
	FatalError	Se publica si se produce `RecordJob` un error grave. Unidades: ninguna. «1" se publica una vez, cuando se produce este evento. **Note** Consulte los registros para obtener información adicional.
	Completado	Publica cuando `RecordJob` se completa un. Unidades: ninguna. «1" se publica una vez, cuando se produce este evento.
Nombre de la transmisión, `UploadJob`	Running	Se publica de forma continua cuando `UploadJob` está en ejecución. Unidades: ninguna. «1" se publica mientras `UploadJob` esté en este estado.
	FatalError	Se publica si se produce `UploadJob` un error grave.

Dimensiones	Estado	Descripción
		Unidades: ninguna. «1" se publica una vez, cuando se produce este evento. Note Consulte los registros para obtener información adicional.
	Completado	Se publica cuando `UploadJob` se completa. Unidades: ninguna. «1" se publica una vez, cuando se produce este evento.
Nombre de flujo	PercentageSpaceUsed	Es el porcentaje utilizado del espacio total asignado en las configuraciones de Amazon Kinesis Video Streams Edge Agent para los soportes de grabación. Para obtener más información, consulta the section called “LocalSizeConfig”. Unidades: porcentaje (escala 0—1).
Nombre de cosa	¡Vivo	Publica cada minuto desde el agente Edge de Amazon Kinesis Video Streams, independientemente de las configuraciones que se estén ejecutando en él. Esto se puede utilizar para saber si el agente Edge de Amazon Kinesis Video Streams está activo y preparado para aceptar configuraciones. Unidades: ninguna. Se publica «1» cada minuto.
	RecordJobs.HealthyJobCount	Recuento total de trabajos de grabación programados y en ejecución en Amazon Kinesis Video Streams Edge Agent. Unidad: recuento.

Dimensiones	Estado	Descripción
	UploadJobs.HealthyJobCount	Recuento total de trabajos de carga programados y en ejecución en Amazon Kinesis Video Streams Edge Agent. Unidad: recuento.
	RecordJobs.UnhealthyJobCount	Recuento total de trabajos de registro con errores actualmente. Unidad: recuento.
	UploadJobs.UnhealthyJobCount	Recuento total de trabajos de carga con errores actualmente. Unidad: recuento.
	RecordJobs.RunningJobCount	Recuento total de trabajos de registro en ejecución activa. Unidad: recuento.
	UploadJobs.RunningJobCount	Recuento total de trabajos de carga en ejecución activa. Unidad: recuento.
	RecordJobs.EdgeConfigCount	Recuento total de configuraciones de grabación en proceso en Amazon Kinesis Video Streams Edge Agent. Unidad: recuento.
	UploadJobs.EdgeConfigCount	Recuento total de configuraciones de carga en proceso en Amazon Kinesis Video Streams Edge Agent. Unidad: recuento.

CloudWatch guía de métricas para Amazon Kinesis Video Streams Edge Agent

CloudWatch las métricas pueden resultar útiles para encontrar respuestas a las siguientes preguntas:

Temas

- ¿El agente Edge de Amazon Kinesis Video Streams tiene suficiente espacio para grabar?
- ¿Está activo el agente Edge de Amazon Kinesis Video Streams?
- ¿Hay trabajos insalubres?
- ¿Hay algún trabajo que necesite intervención externa?

¿El agente Edge de Amazon Kinesis Video Streams tiene suficiente espacio para grabar?

Métricas relevantes: `PercentageSpaceUsed`

Acción: no es necesaria ninguna acción.

¿Está activo el agente Edge de Amazon Kinesis Video Streams?

Métricas relevantes: `Alive`

Acción: Si en algún momento deja de recibir esta métrica, significa que el agente Edge de Amazon Kinesis Video Streams ha detectado uno o varios de los siguientes factores:

- Un problema de tiempo de ejecución de la aplicación: limitación de memoria u otros recursos, error, etc.
- El AWS IoT dispositivo que el agente está ejecutando se apagó, se bloqueó o se cerró
- El AWS IoT dispositivo no tiene conectividad de red

¿Hay trabajos insalubres?

Métricas relevantes:

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

Acción: inspeccione los registros y busque la `FatalError` métrica.

- Si la `FatalError` métrica está presente, se ha producido un error grave y es necesario reiniciar el trabajo manualmente. Inspeccione los registros y solucione el problema antes `StartEdgeConfigurationUpdate` de reiniciarlos manualmente.
- Si la `FatalError` métrica no está presente, se ha producido un error transitorio (no grave) y Amazon Kinesis Video Streams Edge Agent está reintentando el trabajo.

> **Note**
>
> Para que el agente vuelva a intentar un trabajo en el que se cometió un error grave, utilice. the section called “StartEdgeConfigurationUpdate”

¿Hay algún trabajo que necesite intervención externa?

Métricas relevantes:

- `PercentageSpaceUsed`— Si supera un valor determinado, el trabajo de grabación se detiene y se reanuda solo cuando hay espacio disponible (cuando el material deja de retenerse). Puede enviar una configuración actualizada con un valor superior `MaxLocalMediaSizeInMB` para actualizar el trabajo inmediatamente.
- `RecordJob.FatalError`/`UploadJob.FatalError`— Investigue los registros del agente y vuelva a enviar la configuración para que se reanude el trabajo.

Acción: Realice una llamada a la API con la configuración para reiniciar los trabajos que tengan este problema.

Registre las llamadas a la API de Amazon Kinesis Video Streams con AWS CloudTrail

Amazon Kinesis Video Streams funciona AWS CloudTrail con un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o Servicio de AWS una persona en Amazon Kinesis Video Streams. CloudTrail captura todas las llamadas a la API de Amazon Kinesis Video Streams como eventos. Las llamadas capturadas incluyen llamadas desde la consola de Amazon Kinesis Video Streams y llamadas en código a las operaciones de la API de Amazon Kinesis Video Streams. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de Amazon Kinesis Video Streams. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon Kinesis Video Streams, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y habilitarlo, consulte la Guía del AWS CloudTrail usuario.

Amazon Kinesis Video Streams y CloudTrail

CloudTrail está activado en su AWS cuenta al crear la cuenta. Cuando se produce una actividad de eventos admitida en Amazon Kinesis Video Streams, esa actividad se registra en CloudTrail un evento junto con AWS otros eventos de servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para obtener un registro continuo de los eventos de su AWS cuenta, incluidos los eventos de Amazon Kinesis Video Streams, cree una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros Servicios de AWS para que analicen más a fondo los datos de eventos recopilados en los CloudTrail registros y actúen en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Amazon Kinesis Video Streams admite el registro de las siguientes acciones como eventos CloudTrail en archivos de registro:

- CreateStream
- DeleteStream
- DescribeStream
- GetDataEndpoint
- ListStreams
- ListTagsForStream
- TagStream
- UntagStream
- UpdateDataRetention

- UpdateStream

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- si la solicitud se realizó con las credenciales del nodo raíz o del usuario
- si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado
- Si la solicitud la realizó otro Servicio de AWS.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail.

Ejemplo: entradas del archivo de registro de Amazon Kinesis Video Streams

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no son un rastro de la pila ordenada de las llamadas a la API públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la CreateStreamacción.

```
{
    "Records": [
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2018-05-25T00:16:31Z",
            "eventSource": " kinesisvideo.amazonaws.com",
            "eventName": "CreateStream",
            "awsRegion": "us-east-1",
```

```
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
            "requestParameters": {
                "streamName": "VideoStream",
                "dataRetentionInHours": 2,
                "mediaType": "mediaType",
                "kmsKeyId": "arn:aws:kms::us-east-1:123456789012:alias",
  "deviceName": "my-device"
        },
            "responseElements": {
  "streamARN":arn:aws:kinesisvideo:us-east-1:123456789012:stream/VideoStream/12345"
             },
            "requestID": "db6c59f8-c757-11e3-bc3b-57923b443c1c",
            "eventID": "b7acfcd0-6ca9-4ee1-a3d7-c4e8d420d99b"
        },
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2018-05-25:17:06Z",
            "eventSource": " kinesisvideo.amazonaws.com",
            "eventName": "DeleteStream",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
            "requestParameters": {
                "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/
VideoStream/12345",
                "currentVersion": "keqrjeqkj9"
             },
            "responseElements": null,
            "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
            "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
        },
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
```

```
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2014-04-19T00:15:02Z",
            "eventSource": " kinesisvideo.amazonaws.com",
            "eventName": "DescribeStream",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
            "requestParameters": {
                "streamName": "VideoStream"
             },
            "responseElements": null,
            "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
            "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
        },
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2014-04-19T00:15:03Z",
            "eventSource": "kinesisvideo.amazonaws.com",
            "eventName": "GetDataEndpoint",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
            "requestParameters": {
                "streamName": "VideoStream",
                "aPIName": "LIST_FRAGMENTS"
"
            },
            "responseElements": null,
            "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
            "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
        },
```

```
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "accountId": "123456789012",
                "accessKeyId": "EXAMPLE_KEY_ID",
                "userName": "Alice"
            },
            "eventTime": "2018-05-25T00:16:56Z",
            "eventSource": "kinesisvideo.amazonaws.com",
            "eventName": "ListStreams",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
            "requestParameters": {
                "maxResults": 100,
                "streamNameCondition": {"comparisonValue":"MyVideoStream"
 comparisonOperator":"BEGINS_WITH"}}
            },
            "responseElements": null,
            "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
            "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
        }
    ]
}
```

Límites de metadatos de transmisión

Consulte the section called “Cuotas del servicio de metadatos de streaming” para obtener más información sobre los límites que se aplican a la adición de metadatos de streaming a una transmisión de vídeo de Kinesis.

Programe la grabación y el almacenamiento de vídeo con Amazon Kinesis Video Streams Edge Agent

Amazon Kinesis Video Streams ofrece una forma eficiente y rentable de conectarse a cámaras IP en las instalaciones del cliente. Con el Amazon Kinesis Video Streams Edge Agent, puede grabar y almacenar localmente el vídeo de las cámaras y transmitir los vídeos a la nube según un cronograma definido por el cliente para su almacenamiento, reproducción y procesamiento analítico a largo plazo.

> **Note**
>
> Para acceder al agente Edge de Amazon Kinesis Video Streams, complete este breve formulario.

Puede descargar el agente Edge de Amazon Kinesis Video Streams e implementarlo en sus dispositivos de computación perimetral locales. También puede desplegarlos fácilmente en contenedores Docker que se ejecutan en EC2 instancias de Amazon. Tras la implementación, puede utilizar la API de Amazon Kinesis Video Streams para actualizar las configuraciones de grabación de vídeo y carga en la nube. La función funciona con cualquier cámara IP que pueda transmitir a través del protocolo RTSP. No requiere la implementación de ningún firmware adicional en las cámaras.

Ofrecemos las siguientes instalaciones para el agente Edge de Amazon Kinesis Video Streams:

- Como AWS IoT Greengrass V2 componente: puede instalar el agente Edge de Amazon Kinesis Video Streams como AWS IoT Greengrass un componente en AWS IoT Greengrass cualquier dispositivo certificado. Para obtener más información AWS IoT Greengrass, consulte la Guía para AWS IoT Greengrass Version 2 desarrolladores.
- Activado AWS Snowball Edge: puede ejecutar el agente de Amazon Kinesis Video Streams Edge en dispositivos Snowball Edge. Para obtener más información, consulte la Guía para desarrolladores de AWS Snowball Edge Edge.
- En una AWS IoT implementación nativa: puede instalar el agente Edge de Amazon Kinesis Video Streams de forma nativa en cualquier instancia de procesamiento. El SDK de Edge se utiliza AWS IoT Corepara administrar Edge a través de. the section called “Amazon Kinesis Video Streams”

Para empezar a utilizar Amazon Kinesis Video Streams Edge Agent, continúe con los procedimientos correspondientes que se indican a continuación.

Temas

- Operaciones de la API de Amazon Kinesis Video Streams Edge Agent
- Supervisión de Amazon Kinesis Video Streams Edge Agent
- Implemente en modo no operativo AWS IoT Greengrass
- Implemente el agente Edge de Amazon Kinesis Video Streams en AWS IoT Greengrass
- Preguntas frecuentes sobre Amazon Kinesis Video Streams Edge Agent

Operaciones de la API de Amazon Kinesis Video Streams Edge Agent

Utilice las siguientes operaciones de API para configurar el agente Edge de Amazon Kinesis Video Streams:

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

Supervisión de Amazon Kinesis Video Streams Edge Agent

Para supervisar su agente Edge de Amazon Kinesis Video Streams, the section called “Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch” consulte.

Implemente en modo no operativo AWS IoT Greengrass

En esta sección se proporciona una guía completa para utilizar Amazon Kinesis Video Streams fuera AWS IoT Greengrass del entorno. Tanto si trabaja con dispositivos periféricos como con otras plataformas, esta información le ayudará a configurar y utilizar Kinesis Video Streams de forma eficaz.

Encontrará información detallada sobre:

- Configuración del entorno de desarrollo de
- Creación de una transmisión de vídeo de Kinesis

- Descarga y compilación del SDK de Kinesis Video Streams Producer
- Escribir y examinar una aplicación de muestra
- Ejecutar la aplicación de muestra

Siga los pasos que se indican a continuación para ejecutar el agente Edge de Amazon Kinesis Video Streams AWS IoT con MQTT como una implementación independiente.

Temas

- Cómo instalar las dependencias
- Cree recursos para su cámara IP (RTSP) URLs
- Creación de una política de permisos de IAM
- Creación de un rol de IAM
- Cree el alias del rol AWS IoT
- Cree la AWS IoT política
- Crea cualquier AWS IoT cosa y obtén AWS IoT Core credenciales
- Cree el agente Edge de Amazon Kinesis Video Streams
- Instale el CloudWatch agente en el dispositivo
- Ejecute el agente Edge de Amazon Kinesis Video Streams como un proceso nativo

Cómo instalar las dependencias

Antes de empezar a utilizar el SDK para productores de Amazon Kinesis Video Streams, debe configurar el entorno de desarrollo con las dependencias necesarias. Esta página le guía a través del proceso de instalación de los componentes de software y las bibliotecas necesarios en su sistema.

> **Note**
>
> Para obtener una lista de los sistemas operativos compatibles, consultethe section called "¿Qué sistemas operativos admite Amazon Kinesis Video Streams Edge Agent?".

Instale las dependencias en el dispositivo

1. Para ejecutar el agente Amazon Kinesis Video Streams Edge, instale las siguientes bibliotecas adecuadas en su dispositivo:

Ubuntu

Escriba:

```
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

Escriba:

```
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
```

Instálelo `log4cplus-2.1.0` desde la fuente.

```
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
```

Instale `apache-maven-3.9.2` desde la fuente.

```
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**
>
> Si aparece una pantalla en la que se indica que es necesario reiniciar algunos servicios, pulse Entrar para seleccionar Aceptar.

Para obtener información adicional, consulte la Guía del usuario de Amazon Corretto 11.

2. Instale el AWS Command Line Interface. Consulte la sección Instalación o actualización de la última versión de los AWS CLI procedimientos en la Guía del AWS Command Line Interface usuario.

Cree recursos para su cámara IP (RTSP) URLs

Siga estos procedimientos para crear las transmisiones y los secretos necesarios en AWS Secrets Manager. Realice este paso primero, ya que necesitará ARNs los recursos creados en las políticas.

Cree Amazon Kinesis Video Streams

Cree Amazon Kinesis Video Streams mediante AWS Management Console la AWS CLI, o la API.

En AWS Management Console, abra la consola de Amazon Kinesis Video Streams. Seleccione Transmisiones de vídeo en el menú de navegación de la izquierda.

Para obtener más información, consulte the section called “Creación de una transmisión de vídeo de Amazon Kinesis”.

Crea secretos en AWS Secrets Manager

En el AWS Management Console, abre la AWS Secrets Manager consola. Selecciona Secretos en el menú de navegación de la izquierda.

Compruebe que se ha seleccionado la región adecuada.

1. Elija Almacenar un secreto nuevo.
   a. Paso 1: Elige el tipo de secreto
      - Seleccione Otro tipo de secreto.

- En la sección Pares clave/valor, añade un par clave-valor.

  Clave: `MediaURI`

  > **Note**
  >
  > La clave debe ser. `MediaURI` Esto distingue entre mayúsculas y minúsculas. Si lo introduce de forma incorrecta, la aplicación no funciona.

  Valor:*`Your MediaURI`*.

  Example

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

b. Paso 2: Configurar el secreto. Dale un nombre a este secreto. Ponle el nombre que quieras.

c. Paso 3: Configurar la rotación (opcional). Elija Siguiente.

d. Paso 4: Revisar. Elija Almacenar.

2. Si su secreto no se muestra inmediatamente, seleccione el botón de actualización.

   Elige el nombre de tu secreto. Toma nota del ARN secreto.

3. Repite este proceso para cada MediaURI desde el que quieras transmitir.

> **Note**
>
> La AWS red bloquea algunas fuentes RTSP públicas. No puede acceder a ellos desde la EC2 instancia de Amazon o si se ejecuta de forma no gestionada mientras está conectado a la VPN.
>
> > **Important**
> >
> > La URL RTSP de su cámara debe transmitir vídeo en formato h.264. La duración del fragmento no debe superar el límite mencionado en. the section called “Cuotas de SDK para productores”
> > Amazon Kinesis Video Streams Edge Agent solo admite vídeo.

Ejecute `gst-discoverer-1.0 Your RtspUrl` para asegurarse de que se puede acceder a la cámara desde el dispositivo.

ARNs Guárdalo para todas las transmisiones y secretos que hayas creado. Los necesitas para el siguiente paso.

Creación de una política de permisos de IAM

Siga estos procedimientos para crear una política de IAM. Esta política de permisos permite el control de acceso selectivo (un subconjunto de operaciones compatibles) a un AWS recurso. En este caso, los AWS recursos son las transmisiones de vídeo a las que desea que transmita el agente Edge de Amazon Kinesis Video Streams. Los recursos también incluyen los AWS Secrets Manager secretos que el agente Edge de Amazon Kinesis Video Streams puede recuperar. Para obtener más información, consulte Políticas de IAM.

Cree una política mediante el editor de políticas JSON

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo, elija Políticas.

   Si es la primera vez que elige Políticas, aparecerá la página Welcome to Managed Policies (Bienvenido a políticas administradas). Elija Comenzar.
3. En la parte superior de la página, seleccione Crear política.
4. En la sección Editor de políticas, seleccione la opción JSON.
5. Ingrese el siguiente documento de política JSON:

   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**
>
> Sustituya `arn:aws:kinesisvideo:*:*:stream/streamName1/*` y `arn:aws:kinesisvideo:*:*:stream/streamName2/*` ARNs por la para las transmisiones de vídeo y `arn:aws:secretsmanager:*:*:secret:*` sustitúyala por la ARNs que contenga los secretos de MediaURI en los que creó. the section called

"Cree recursos para su cámara IP RTSP URLs" Úselo ARNs para los secretos a los que desee que acceda el agente Edge de Amazon Kinesis Video Streams.

6. Elija Siguiente.

> **Note**
>
> Puede alternar entre las opciones Visual y JSON del editor en todo momento. No obstante, si realiza cambios o selecciona Siguiente en la opción Visual del editor, es posible que IAM reestructure la política, con el fin de optimizarla para el editor visual. Para obtener más información, consulte la reestructuración de políticas en la Guía del usuario de IAM.

7. En la página Revisar y crear, introduzca un nombre de política y una descripción opcional para la política que va a crear. Revise los Permisos definidos en esta política para ver los permisos que concede la política.
8. Elija Crear política para guardar la nueva política.

Creación de un rol de IAM

Puede asumir el rol que cree en este paso para obtener credenciales temporales de AWS Security Token Service (AWS STS). AWS IoT Esto se hace al realizar solicitudes de autorización de credenciales desde el agente Edge de Amazon Kinesis Video Streams.

Cree el rol de servicio para Amazon Kinesis Video Streams (consola de IAM)

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
2. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, elija Crear rol.
3. Elija el tipo de rol de política de confianza personalizada y pegue la siguiente política:

   JSON

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

```
        "Effect": "Allow",
        "Principal": {
            "Service": "credentials.iot.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Seleccione la casilla situada junto a la política de IAM en the section called “Creación de una política de permisos de IAM” la que creó.
5. Elija Siguiente.
6. Introduzca un nombre de rol o un sufijo de nombre de rol para ayudarle a identificar el propósito de este rol.

   Example

   Ejemplo: `KvsEdgeAgentRole`
7. (Opcional) En Descripción, ingrese una descripción para el nuevo rol.
8. (Opcional) Añada metadatos al rol adjuntando etiquetas como pares de clave/valor.

   Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetar los recursos de IAM en la Guía del usuario de IAM.
9. Revise el rol y, a continuación, seleccione Crear rol.

Cree el alias del rol AWS IoT

Siga estos procedimientos para crear un alias de AWS IoT rol para el rol de IAM en el que lo creó. the section called “Creación de un rol de IAM” Un alias de rol es un modelo de datos alternativo que apunta al rol de IAM. La solicitud del proveedor de AWS IoT credenciales debe incluir un alias de rol para indicar qué rol de IAM debe asumir para poder obtener credenciales temporales de AWS Security Token Service ()AWS STS. Para obtener más información, consulta Cómo usar un certificado para obtener un token de seguridad.

Cree el alias del AWS IoT rol

1. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/.
2. Compruebe que se ha seleccionado la región adecuada.

3. En el panel de navegación de la izquierda, seleccione Seguridad y, a continuación, elija Alias de rol.
4. Seleccione Crear alias de rol.
5. Introduzca un nombre para el alias de su rol.

   Example

   Ejemplo: `KvsEdgeAgentRoleAlias`
6. En el menú desplegable Rol, selecciona el rol de IAM en el que lo creaste. the section called "Creación de un rol de IAM"
7. Seleccione Crear. En la página siguiente, verá una nota en la que se indica que el alias de su rol se creó correctamente.
8. Busque y seleccione el alias de rol recién creado. Anote el alias ARN del rol. Lo necesitará para la AWS IoT política en el siguiente paso.

Cree la AWS IoT política

Siga estos procedimientos para crear una AWS IoT política que se adjuntará al certificado del dispositivo. Esto otorga permisos a AWS IoT las capacidades y permite asumir el alias de rol mediante el certificado.

Con AWS IoT Core las políticas, puede controlar el acceso al plano AWS IoT Core de datos. El plano de AWS IoT Core datos consta de operaciones que puede utilizar para hacer lo siguiente:

- Conéctese al agente de AWS IoT Core mensajes
- Envía y recibe mensajes MQTT
- Obtén o actualiza la sombra del dispositivo de una cosa

Para obtener más información, consulte Políticas de AWS IoT Core.

Utilice el editor de AWS IoT políticas para crear una AWS IoT política

1. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/.
2. En el panel de navegación de la izquierda, selecciona Seguridad y, a continuación, Políticas.
3. Elija Crear política.

4. Escriba un nombre para la política.

   Example

   Un ejemplo de nombre de política es KvsEdgeAccessIoTPolicy.

5. (Opcional) Agregar metadatos a la política al adjuntar las etiquetas como pares de clave-valor.

   Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetar AWS IoT los recursos en la Guía para AWS IoT Core desarrolladores.

6. Seleccione la pestaña JSON.

7. Pegue el siguiente documento de políticas JSON:

   JSON

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

> **Note**
>
> `your-role-alias-arn`Sustitúyalo por el ARN del alias de rol en el que creaste. the section called “Cree el alias del AWS IoT rol”

8. Seleccione Crear para guardar el trabajo.

Crea cualquier AWS IoT cosa y obtén AWS IoT Core credenciales

En este punto, has creado:

- Una política de permisos de IAM. Consulte the section called “Creación de una política de permisos de IAM”.
- Un rol de IAM, con la política de permisos adjunta. Consulte the section called “Creación de un rol de IAM”.
- Un alias de AWS IoT rol para el rol de IAM. Consulte the section called “Cree el alias del AWS IoT rol”.
- Una AWS IoT política que actualmente no está asociada a ningún AWS recurso. Consulte the section called “Cree la AWS IoT política”.

Para crear y registrar cualquier AWS IoT cosa y obtener credenciales de AWS IoT Core acceso

1. Registre el dispositivo como una AWS IoT cosa y genere el certificado X.509 para el dispositivo.
   a. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/.
   b. Seleccione la región correspondiente.
   c. En el panel de navegación de la izquierda, selecciona Todos los dispositivos y, a continuación, selecciona Cosas.
   d. Elige Crear cosas.
   e. Selecciona Crear una sola cosa y, a continuación, selecciona Siguiente.
      1. Paso 1. Especifique las propiedades de la cosa

         Escriba un nombre para el objeto y, a continuación, seleccione Siguiente.

2. Paso 2. Configura el certificado del dispositivo

   Seleccione Generar automáticamente un nuevo certificado (recomendado) y, a continuación, elija Siguiente.

3. Paso 3. Adjunta políticas al certificado

   Busque la política de permisos en la que creóthe section called “Cree la AWS IoT política”.

   Selecciona la casilla de verificación situada junto a tu política y selecciona Crear cosa.

f. En la ventana que aparece, descarga los siguientes archivos:

   - Certificado de dispositivo. Este es el certificado X.509.
   - Archivo de clave pública
   - Archivo de clave privada
   - Punto final de Amazon Trust Services (clave RSA de 2048 bits: Amazon Root CA 1)

   Anote la ubicación de cada uno de estos archivos para un paso posterior.

g. Seleccione Listo. En la página siguiente, verás una nota que indica que tu objeto se ha creado correctamente.

h. Transfiere los archivos descargados arriba a AWS IoT tu dispositivo, si aún no están allí.

2. Obtén el punto de conexión del proveedor de credenciales para tu AWS cuenta.

   AWS CLI

   Ejecuta el siguiente comando:

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

   AWS Management Console

   En AWS CloudShell, ejecute el siguiente comando:

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

   Anote esta información para un paso posterior.

3. Obtén el punto final de datos del dispositivo para tu AWS cuenta.

   AWS CLI

   Ejecuta el siguiente comando:

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

   AWS Management Console

   Haga lo siguiente:

   1. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/.
   2. En el panel de navegación de la izquierda, selecciona Configuración.
   3. Localice el punto final de datos del dispositivo.

   Tome nota de esta información para un paso posterior.

4. (Opcional) Compruebe que los certificados se generaron correctamente.

   Ejecute el siguiente comando para validar que los elementos se generaron correctamente.

   ```
   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
   ```

   Para obtener más información, consulta Cómo usar un certificado para obtener un token de seguridad.

Cree el agente Edge de Amazon Kinesis Video Streams

Cree el agente Edge de Amazon Kinesis Video Streams

1. Descargue el `tar` archivo mediante el enlace que se le proporcionó.

Si ha rellenado el formulario de interés para Amazon Kinesis Video Streams Edge Agent, consulte el enlace de descarga en su correo electrónico. Si no ha rellenado el formulario, rellénelo aquí.

2. Verifica la suma de verificación.
3. Extrae los archivos binarios y el tarro de tu dispositivo.

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

   Tras la extracción, la estructura de carpetas tendrá el siguiente aspecto:

```
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**
>
> El nombre de la carpeta de publicación debe configurarse de forma que refleje el número de la última versión binaria. Por ejemplo, una versión 1.0.0 tendrá el nombre de la carpeta establecido como 1.0.0.

4. Cree el tarro de dependencias.

> **Note**
>
> El contenedor incluido con el `kvs-edge-agent.tar.gz` no tiene las dependencias. Siga los pasos siguientes para crear esas bibliotecas.

Navegue hasta la `kvs-edge-agent` carpeta que contiene`pom.xml`.

Tipo `mvn clean package`.

Esto genera un archivo jar que contiene las dependencias en las que requiere el agente Edge de Amazon Kinesis Video Streams. `kvs-edge-agent/target/libs.jar`

5. `libs.jar`Colóquelo en la carpeta que contiene los artefactos del componente.

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

6. Configure las variables de entorno utilizando los valores de los pasos anteriores. La siguiente tabla proporciona descripciones de las variables.

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_REGION`	Sí	La región que se utiliza. Ejemplo: us-west-2
`AWS_IOT_CA_CERT`	Sí	Ruta del archivo al certifica do de CA utilizado para establecer la confianza con

Nombre de la variable de entorno	Obligatorio	Descripción
		el servicio de backend a través de TLS. Ejemplo: `/file/path/to/AmazonRootCA1.pem`
`AWS_IOT_CORE_CERT`	Sí	Ruta del archivo al certificado X.509. Ejemplo: `/file/path/to/certificateID-certificate.pem.crt`
`AWS_IOT_CORE_CREDENTIAL_ENDPOINT`	Sí	El punto de conexión del proveedor de AWS IoT Core credenciales de su cuenta. AWS Ejemplo: `credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com`
`AWS_IOT_CORE_DATA_ATS_ENDPOINT`	Sí	El punto final del plano de AWS IoT Core datos de su AWS cuenta. Ejemplo: `data-account-specific-prefix.iot.aws-region.amazonaws.com`

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_IOT_CORE_PRIVATE_KEY`	Sí	Ruta del archivo a la clave privada utilizada en el par de claves pública/privada. Para obtener más información, consulte Administración de claves en. AWS IoT Ejemplo: `/file/path/to/certificateID-private.pem.key`
`AWS_IOT_CORE_ROLE_ALIAS`	Sí	El nombre del alias del rol que apunta al rol de AWS IAM que se va a usar al conectarse AWS IoT Core. Ejemplo: `kvs-edge-role-alias`
`AWS_IOT_CORE_THING_NAME`	Sí	El nombre del elemento AWS IoT en el que se ejecuta la aplicación. Ejemplo: `my-edge-device-thing`

Nombre de la variable de entorno	Obligatorio	Descripción
`GST_PLUGIN_PATH`	Sí	Ruta del archivo que apunta a la carpeta que contiene las bibliotecas dependientes de la `IngestorPipelineJNI` plataforma a `gstkvssink` y las bibliotecas dependientes de la plataforma. GStreamer Carguemos estos complementos. Para obtener más información, consulte the section called "Descargue, cree y configure el GStreamer elemento". Ejemplo: `/`*`download-location`*` /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*`EdgeAgentVersion`*` /`

Nombre de la variable de entorno	Obligatorio	Descripción
`LD_LIBRARY_PATH`	Sí	Ruta del archivo que apunta al directorio que contiene las bibliotecas dependientes de la `KinesisVideoProducer` plataforma `cproducer` y las bibliotecas dependientes de la plataforma. Ejemplo: `/`*`download-location`*` /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*`EdgeAgentVersion`*` /lib/`
`AWS_KVS_EDGE_CLOUDWATCH_ENABLED`	No	Determina si el agente Edge de Amazon Kinesis Video Streams publicará las estadísticas Amazon CloudWatch del estado del trabajo. Valores aceptados :`TRUE/FALSE`(no distingue entre mayúsculas y minúsculas). El valor predeterminado es `FALSE` si no se proporciona. Ejemplo: FALSO

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_KVS_EDGE_LOG_LEVEL`	No	El nivel de registro de las salidas del agente Edge de Amazon Kinesis Video Streams. Valores aceptados: • OFF • ALL • FATAL • ERROR • WARN • INFO, predeterminado, si no se proporciona • DEBUG • TRACE Ejemplo: `INFO`
`AWS_KVS_EDGE_LOG_MAX_FILE_SIZE`	No	Cuando el archivo de registro alcance este tamaño, se transferirá. • Mínimo: 0 • Máximo: 10000 • Predeterminado: 20, si no se proporciona • Unidades: megabytes (MB) Ejemplo: 5

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY`	No	La ruta del archivo que apunta al directorio en el que se generan los registros de Amazon Kinesis Video Streams Edge Agent. El valor predeterminado es `./log` si no se proporciona. Ejemplo: `/file/path/`
`AWS_KVS_EDGE_LOG_ROLLOVER_COUNT`	No	El número de registros acumulados que se deben conservar antes de eliminarlos. • Mínimo: 1 • Máximo: 100 • Predeterminado: 10, si no se proporciona Ejemplo: 20
`AWS_KVS_EDGE_RECORDING_DIRECTORY`	No	La ruta del archivo que apunta al directorio en el que se escribirá el soporte grabado. El valor predeterminado es el directorio actual si no se proporciona. Ejemplo: `/file/path/`

Nombre de la variable de entorno	Obligatorio	Descripción
`GST_DEBUG`	No	Especifica el nivel de GStreamer registros que se van a generar. Para obtener más información, consulte la Documentación de GStreamer . Ejemplo: 0
`GST_DEBUG_FILE`	No	Especifica el archivo de salida de los registros de GStreamer depuración. Si no se establece, los registros de depuración muestran un error estándar. Para obtener más información, consulte la Documentación de GStreamer . Ejemplo: `/tmp/gstreamer-logging .log`

7. Borra la GStreamer caché. Escriba:

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

Para obtener más información, consulte la documentación GStreamer del registro.

8. Prepare y ejecute el comando java. El agente Edge de Amazon Kinesis Video Streams acepta los siguientes argumentos:

Nombre de propiedad de Java	Obligatorio	Descripción
`java.library.path`	No	Ruta del archivo que apunta a la carpeta que contiene las

Nombre de propiedad de Java	Obligatorio	Descripción
		bibliotecas `gstkvssink` y las bibliotecas `IngestorPipelineJNI` dependientes. Si no se proporcionan, el agente Edge de Amazon Kinesis Video Streams las buscará en el directorio actual. **Important** El agente Edge de Amazon Kinesis Video Streams no funcionará correctamente si no encuentra estos archivos. Ejemplo: `/file/path/`

Para configurarlos, agréguelos `-Djava-property-name=value` al comando java utilizado para ejecutar el jar.

Por ejemplo:

```
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**
>
> Ejecute el comando java anterior desde el mismo directorio que/*download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*.

9. Envíe las configuraciones a la aplicación mediante AWS CLI.

   a. Cree un archivo nuevo,*example-edge-configuration*.json.

      Pegue el código siguiente en el archivo. Este es un ejemplo de configuración que graba todos los días desde las 9:00:00 a.m. hasta las 4:59:59 p.m. (según la hora del sistema del dispositivo). AWS IoT También carga el contenido multimedia grabado todos los días de 19:00 a 21:59:59 horas.

      Para obtener más información, consulte the section called “StartEdgeConfigurationUpdate”.

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

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

b. Para enviar el archivo al agente Edge de Amazon Kinesis Video Streams, escriba lo siguiente en AWS CLI:

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

10. Repita el paso anterior para cada transmisión para el agente Edge de Amazon Kinesis Video Streams.

Instale el CloudWatch agente en el dispositivo

> **Note**
>
> Tenga en cuenta las CloudWatchcuotas.

Siga estos procedimientos para instalar y configurar el CloudWatch agente para que cargue automáticamente los registros generados por el agente CloudWatch Edge de Amazon Kinesis Video Streams a. Se trata de un paso opcional.

Para conocer los procedimientos para instalar el CloudWatch agente en su dispositivo, consulte la Guía del CloudWatch usuario de Amazon.

Cuando se le solicite la configuración, seleccione una de las siguientes configuraciones.

> **Important**
>
> `file_path`En las siguientes configuraciones, se supone que se utiliza la ubicación de salida de registro predeterminada.

La ruta de archivo utilizada supone que está ejecutando el agente Edge de Amazon Kinesis Video Streams desde la ubicación:*download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*version*.

- Para configurar el CloudWatch agente para que cargue registros y publique las métricas de RAM y CPU del dispositivo, pegue lo siguiente en el archivo de configuración.

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

- Para cargar solo los registros y no recopilar la RAM y la CPU del dispositivo, utilice la siguiente configuración:

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

Ejecute el agente Edge de Amazon Kinesis Video Streams como un proceso nativo

Configure el agente Edge de Amazon Kinesis Video Streams como un servicio systemd. Se trata de un paso opcional.

`systemd`es un administrador de sistemas y servicios en dispositivos Linux. `systemd`es la forma recomendada de gestionar el proceso, ya que reiniciará el agente Edge de Amazon Kinesis Video Streams en caso de que la aplicación detecte un error o el dispositivo que la ejecuta se quede sin alimentación.

Haga lo siguiente:

Ejecute el agente Edge de Amazon Kinesis Video Streams como un proceso nativo

1. Cree un archivo nuevo `/etc/systemd/system` y asígnele un nombre. *`aws.kinesisvideo.edge-runtime-agent`*`.service`

Pegue lo siguiente:

```
[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
```

Para obtener más información sobre los parámetros aceptados por el archivo de configuración del `systemd` servicio, consulte la documentación.

> **Note**
>
> Añada las variables de entorno necesarias en la `...` ubicación, tal y como se especifica enthe section called “Cree el agente Edge”.

2. Vuelva a cargar los archivos de servicio para incluir el nuevo servicio.

   Tipo `sudo systemctl daemon-reload`.

3. Inicie el servicio .

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

4. Compruebe el estado del servicio Amazon Kinesis Video Streams Edge Agent para comprobar que se está ejecutando.

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

   El siguiente es un ejemplo del resultado que verá.

```
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. Inspeccione los registros para ver si hay errores.

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

6. Escriba `systemctl --help` aquí la lista completa de opciones con las que gestionar el procesos`systemctl`.

   Los siguientes son algunos comandos comunes para administrar el agente Edge de Amazon Kinesis Video Streams:

   - Para reiniciar, escriba`sudo systemctl restart` *`aws.kinesisvideo.edge-runtime-agent`*`.service`.
   - Para detener, escriba`sudo systemctl stop` *`aws.kinesisvideo.edge-runtime-agent`*`.service`.
   - Para iniciar automáticamente cada vez que se reinicie el dispositivo, escriba`sudo systemctl enable` *`aws.kinesisvideo.edge-runtime-agent`*`.service`.

Implemente el agente Edge de Amazon Kinesis Video Streams en AWS IoT Greengrass

En esta sección se proporciona una guía completa para utilizar Amazon Kinesis Video Streams AWS IoT Greengrass con. Al combinar estos servicios, puede transmitir vídeo de manera eficiente desde dispositivos periféricos a la nube, lo que permite una amplia gama de aplicaciones en IoT, vigilancia y más.

Encontrará información detallada sobre:

- Configuración del entorno de desarrollo de
- Creación de una transmisión de vídeo de Kinesis
- Creación y empaquetado de una función Lambda
- Configuración del dispositivo principal de Kinesis Video Streams
- Implementación en el dispositivo principal
- Verificación de la transmisión

Siga estos pasos para implementar el Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass para grabar y cargar contenido multimedia desde cámaras IP.

Temas

- Crear una EC2 instancia de Amazon en Ubuntu
- Configure el dispositivo AWS IoT Greengrass V2 principal en el dispositivo
- Cree Amazon Kinesis Video Streams AWS Secrets Manager y los recursos para su cámara IP (RTSP) URLs
- Agregue permisos a la función del servicio de intercambio de fichas (TES)
- Instale el componente AWS IoT Greengrass Secret Manager en el dispositivo
- Implemente el componente Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent en el dispositivo
- Instale el componente de gestión de AWS IoT Greengrass registros en el dispositivo

Crear una EC2 instancia de Amazon en Ubuntu

Haga lo siguiente para crear una EC2 instancia de Amazon en Ubuntu.

Crear una EC2 instancia de Amazon en Ubuntu

1. Inicia sesión en la EC2 consola de Amazon AWS Management Console y ábrela en https://console.aws.amazon.com/ec2/.

   Comprueba que se ha seleccionado la región adecuada.
2. Elija Iniciar instancia.

   Complete los siguientes campos:

   - Nombre: escriba un nombre para la instancia.
   - Imágenes de aplicaciones y sistemas operativos (Amazon Machine Image): seleccione Ubuntu.
   - Tipo de instancia: seleccione t2.large.
   - Inicio de sesión con pares de claves: cree su propio par de claves.
   - Configuración de red: mantenga la configuración predeterminada.
   - Configurar el almacenamiento: aumente el volumen a 256 GiB.
   - Configuración avanzada: mantenga la configuración predeterminada.
3. Lanza la instancia y usa SSH en ella.

   Haga lo siguiente:

   1. Selecciona Instancias en el menú de navegación de la izquierda y, a continuación, selecciona el ID de la instancia.
   2. Selecciona Connect en la esquina superior derecha.
   3. Elige el cliente SSH y sigue las instrucciones que aparecen en pantalla.
   4. Abre una terminal y navega hasta el `.pem` archivo descargado (probablemente dentro`~/Downloads`).
   5. La primera vez que siga estos procedimientos, recibirá el mensaje «No se puede establecer la autenticidad del anfitrión (...)». Escriba sí.
4. Instale las bibliotecas del sistema para crear el agente Edge de Amazon Kinesis Video Streams en la instancia.

```
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**
>
> Si aparece una pantalla que indica que es necesario reiniciar algunos servicios, pulse Entrar para seleccionar Aceptar.

Para obtener más información, consulte la Guía del usuario de Amazon Corretto 11.

Configure el dispositivo AWS IoT Greengrass V2 principal en el dispositivo

Siga estos procedimientos para instalar el software AWS IoT Greengrass core nucleus en la EC2 instancia de Amazon.

Configura el dispositivo AWS IoT Greengrass principal

1. Inicie sesión en AWS Management Console, https://console.aws.amazon.com/iot/.

   Compruebe que se ha seleccionado la región adecuada.
2. En el menú de navegación de la izquierda, selecciona Dispositivos Greengrass, Dispositivos principales.
3. Selecciona Configurar un dispositivo principal.
4. Completa los pasos que aparecen en la pantalla.
   - Paso 1: Registrar un dispositivo principal de Greengrass. Escriba un nombre para el dispositivo.
   - Paso 2: agréguelo a un grupo de cosas para aplicar una implementación continua. Seleccione Sin grupo.
   - Paso 3: Instale el software Greengrass Core. Seleccione Linux.
     - Paso 3.1: Instale Java en el dispositivo

Java se instala como parte dethe section called “Crea una instancia de Ubuntu”. Vuelva a ese paso si aún no tiene Java instalado.

- Paso 3.2: Copie AWS las credenciales en el dispositivo

  Abre la `bash/zsh` opción y pega los comandos de exportación en la EC2 instancia de Amazon.

- Paso 3.3: Ejecuta el instalador

  1. Copia y ejecuta los comandos Descargar el instalador y Ejecutar el instalador en la EC2 instancia de Amazon de Ubuntu.

     > **Note**
     >
     > El comando Ejecutar el instalador se actualizará automáticamente en función del nombre que haya elegido en el paso anterior.

  2. Anote la función de servicio de intercambio de fichas (TES) que se ha creado. Lo necesitará más adelante.

     > **Note**
     >
     > De forma predeterminada, el rol creado se denomina GreengrassV2 TokenExchangeRole.

Cree Amazon Kinesis Video Streams AWS Secrets Manager y los recursos para su cámara IP (RTSP) URLs

Siga estos procedimientos para crear las transmisiones y los secretos necesarios en AWS Secrets Manager. Realice este paso primero, ya que necesitará ARNs los recursos creados en las políticas.

Cree Amazon Kinesis Video Streams

Cree Amazon Kinesis Video Streams mediante AWS Management Console la AWS CLI, o la API.

En AWS Management Console, abra la consola de Amazon Kinesis Video Streams. Seleccione Transmisiones de vídeo en el menú de navegación de la izquierda.

Para obtener más información, consulte the section called “Creación de una transmisión de vídeo de Amazon Kinesis”.

Crea secretos en AWS Secrets Manager

En el AWS Management Console, abre la AWS Secrets Manager consola. Selecciona Secretos en el menú de navegación de la izquierda.

Compruebe que se ha seleccionado la región adecuada.

1. Elija Almacenar un secreto nuevo.
   a. Paso 1: Elige el tipo de secreto
      - Seleccione Otro tipo de secreto.
      - En la sección Pares clave/valor, añade un par clave-valor.

        Clave: `MediaURI`

        > **Note**
        > La clave debe ser. `MediaURI` Esto distingue entre mayúsculas y minúsculas. Si lo introduce de forma incorrecta, la aplicación no funciona.

        Valor:*`Your MediaURI`*.

        Example

        Ejemplo:`rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/YourCameraMediaURI`.
   b. Paso 2: Configurar el secreto. Dale un nombre a este secreto. Ponle el nombre que quieras.
   c. Paso 3: Configurar la rotación (opcional). Elija Siguiente.
   d. Paso 4: Revisar. Elija Almacenar.
2. Si su secreto no se muestra inmediatamente, seleccione el botón de actualización.

   Elige el nombre de tu secreto. Toma nota del ARN secreto.
3. Repite este proceso para cada MediaURI desde el que quieras transmitir.

> **Note**
>
> La AWS red bloquea algunas fuentes RTSP públicas. No puede acceder a ellos desde la EC2 instancia de Amazon o si se ejecuta de forma no gestionada mientras está conectado a la VPN.
>
> > **Important**
> >
> > La URL RTSP de su cámara debe transmitir vídeo en formato h.264. La duración del fragmento no debe superar el límite mencionado en. the section called "Cuotas de SDK para productores"
> > Amazon Kinesis Video Streams Edge Agent solo admite vídeo.
>
> Ejecute `gst-discoverer-1.0 Your RtspUrl` para asegurarse de que se puede acceder a la cámara desde el dispositivo.

ARNs Guárdalo para todas las transmisiones y secretos que hayas creado. Los necesitas para el siguiente paso.

Agregue permisos a la función del servicio de intercambio de fichas (TES)

Otorgue la función de servicio de intercambio de fichas (TES) al dispositivo que asume los permisos para ver los secretos. Esto es necesario para que el AWS Secrets Manager AWS IoT Greengrass componente funcione correctamente.

Agregue permisos a la función TES

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. Elija Roles en el menú de navegación de la izquierda y busque el rol de TES que creó anteriormente en el proceso.
3. En el menú desplegable Añadir permisos, selecciona Adjuntar políticas.
4. Elija Crear política.
5. Desplázate hacia abajo y selecciona Editar.
6. En el editor de políticas, elige JSON y edita la política.

Sustituya la política por lo siguiente:

> **Note**
>
> Sustituya `arn:aws:kinesisvideo:*:*:stream/streamName1/*` y `arn:aws:kinesisvideo:*:*:stream/streamName2/*` ARNs por la para las transmisiones que creó en un paso anterior.

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. En la página Add tags (Añadir etiquetas), elija Next: Review (Siguiente: Revisión).
8. Asigne un nombre a su política y, a continuación, seleccione Crear política.

Un ejemplo de nombre de política es KvsEdgeAccessPolicy.

9. Cierre la pestaña y vuelva a la pestaña en la que estaba adjuntando una política a la función de TES.

   Pulse el botón de actualización y, a continuación, busque la política recién creada.

   Seleccione la casilla de verificación y elija Adjuntar políticas.

   En la siguiente pantalla, verá una nota que dice que la política se asignó correctamente al rol.

10. Crea y adjunta otra política, esta vez para tus secretos.

    Sustituya la política por la siguiente:

> **Note**
>
> `arn:aws:secretsmanager:*:*:secret:*`Sustitúyala por la ARNs que contiene los secretos de MediaURI que creaste en ella. the section called “Cree recursos para el RTSP de su cámara IP URLs”

JSON

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

11. Crea y adjunta otra política, esta vez para Amazon CloudWatch las métricas. Sustituya la política por lo siguiente:

JSON

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

Instale el componente AWS IoT Greengrass Secret Manager en el dispositivo

El agente Edge de Amazon Kinesis Video Streams requiere que AWS IoT Greengrass primero se instale el componente Secret Manager en el dispositivo.

Instale el componente Secret Manager

1. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/. Compruebe que se ha seleccionado la región adecuada.
2. En el menú de navegación de la izquierda, selecciona Greengrass devices, Deployments.

   Elija el despliegue con el mismo objetivo que el que hemos creado. the section called “Configure el dispositivo AWS IoT Greengrass principal”
3. En el menú desplegable Acciones de la esquina superior derecha, selecciona Revisar.

   En la ventana emergente que aparece, selecciona Revisar la implementación.
4. Complete las siguientes secciones:

   - Paso 1: especifique el objetivo. Elija Siguiente.

- Paso 2: Seleccione los componentes.
  - Compruebe que el componente AWS.Greengrass.cli esté seleccionado. No desinstale este componente.
  - Active la opción Mostrar solo los componentes seleccionados y busque aws.greengrass. SecretManager.
  - Marca la casilla junto a aws.greengrass. SecretManagery, a continuación, selecciona Siguiente.
- Paso 3: Configurar los componentes. Configure el componente AWS IoT Greengrass Secret Manager para descargar los secretos desde el AWS IoT Greengrass entorno.

  Seleccione aws.greengrass. SecretManagercomponente y, a continuación, elija Configurar componente.

  En la pantalla que aparece, actualice AWS Secrets Manager ARNs el cuadro Configuración para fusionar.

  > **Note**
  >
  > `arn:aws:secretsmanager:*:*:secret:*` ARNs Sustitúyalos por los secretos en los que los creóthe section called “Cree recursos para el RTSP de su cámara IP URLs”.

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

> **Note**
>
> `cloudSecretses` una lista de objetos con la clave`arn`. Para obtener más información, consulte la sección de configuración del administrador secreto de la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Cuando hayas terminado, selecciona Confirmar y, a continuación, Siguiente.

- Paso 4: Configura los ajustes avanzados. Seleccione Siguiente.
- Paso 5: Revisar. Seleccione Implementar.

5. Confirme que el AWS Secrets Manager componente y los permisos se instalaron correctamente.

   En la EC2 instancia Amazon de Ubuntu, escriba `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` para comprobar que el componente recibió la configuración actualizada.

6. Inspeccione los registros AWS IoT Greengrass principales.

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

   Revise si hay errores de implementación.

   Si se ha producido un error, revise la implementación para eliminar el `aws.greengrass.SecretManager` componente.

   Escriba `sudo service greengrass restart` esta opción para reiniciar el servicio AWS IoT Greengrass principal.

   Si el error de implementación estuvo relacionado con la falta de permisos, revise la the section called “Agregue permisos a la función TES” sección para asegurarse de que el rol de TES tiene los permisos adecuados. A continuación, repita esta sección.

Actualice los secretos del componente AWS IoT Greengrass Secret Manager

> **Important**
>
> El componente AWS IoT Greengrass Secret Manager recupera y almacena en caché los secretos solo cuando se actualiza la implementación.

Para actualizar los secretos del componente AWS IoT Greengrass Secret Manager, siga los pasos anteriores del 1 al 6 e introduzca el siguiente cambio.

Paso 3: Configurar los componentes. Configure el componente AWS IoT Greengrass Secret Manager para descargar los secretos desde el AWS IoT Greengrass entorno.

Seleccione aws.greengrass. SecretManagercomponente y, a continuación, elija Configurar componente.

En la pantalla que aparece, péguelo `[""]` en el cuadro Restablecer rutas y actualícelo AWS Secrets Manager ARNs en el cuadro Configuración para fusionar.

Para obtener más información, consulte Restablecer actualizaciones.

Implemente el componente Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent en el dispositivo

Haga lo siguiente para implementar el componente Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent en el dispositivo:

Implemente el componente

1. Descargue el `tar` archivo mediante el enlace proporcionado.

   Si ha rellenado el formulario de interés para Amazon Kinesis Video Streams Edge Agent, consulte el enlace de descarga en su correo electrónico. Si no ha rellenado el formulario, rellénelo aquí.
2. Verifica la suma de verificación.
3. Extrae los archivos binarios y el tarro de tu dispositivo.

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

Tras la extracción, la estructura de carpetas tendrá el siguiente aspecto:

```
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**
>
> El nombre de la carpeta de publicación debe configurarse de forma que refleje el número de la última versión binaria. Por ejemplo, una versión 1.0.0 tendrá el nombre de la carpeta establecido como 1.0.0.

4. Cree el tarro de dependencias.

> **Note**
>
> El contenedor incluido con el kvs-edge-agent archivo.tar.gz no tiene las dependencias. Siga los pasos siguientes para crear esas bibliotecas.

Navegue hasta la `kvs-edge-agent` carpeta que contiene`pom.xml`.

Tipo `mvn clean package`.

Esto generará un archivo jar que contiene las dependencias en las que requiere el agente Edge de Amazon Kinesis Video Streams. `kvs-edge-agent/target/libs.jar`

5. Coloque el archivo libs.jar en la carpeta que contiene los artefactos del componente.

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

6. Opcional. Configure las propiedades. El agente Edge de Amazon Kinesis Video Streams acepta las siguientes variables de entorno AWS IoT Greengrass en el modo:

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_REGION`	Sí	La región que se utiliza. Ejemplo: us-west-2 AWS IoT Greengrass El software principal establece este valor automáticamente. Para obtener más informaci ón, consulte el tema de referencia sobre las variables de entorno de los component es en la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Nombre de la variable de entorno	Obligatorio	Descripción
`GST_PLUGIN_PATH`	Sí	Ruta del archivo que apunta a la carpeta que contiene las bibliotecas dependientes de la `IngestorPipelineJNI` plataforma a `gstkvssink` y las bibliotecas dependientes de la plataforma. Esto permite GStreamer cargar estos complementos. Para obtener más información, consulte the section called "Descargue, cree y configure el GStreamer elemento". Ejemplo: `/`*`download-location`*` /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*`EdgeAgentVersion`*` /`

Nombre de la variable de entorno	Obligatorio	Descripción
`LD_LIBRARY_PATH`	Sí	Ruta del archivo que apunta al directorio que contiene las bibliotecas dependientes de la `KinesisVideoProducer` plataforma `cproducer` y las bibliotecas dependientes de la plataforma. Ejemplo: `/`*`download-location`*` /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*`EdgeAgentVersion`*` /lib/`
`AWS_KVS_EDGE_CLOUDWATCH_ENABLED`	No	Determina si el agente Edge de Amazon Kinesis Video Streams publicará las estadísticas Amazon CloudWatch del estado del trabajo. Valores aceptados :`TRUE/FALSE`(no distingue entre mayúsculas y minúsculas). El valor predeterminado es `FALSE` si no se proporciona. Ejemplo: FALSO

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_KVS_EDGE_LOG_LEVEL`	No	El nivel de registro de las salidas del agente Edge de Amazon Kinesis Video Streams. Valores aceptados: • OFF • ALL • FATAL • ERROR • WARN • INFO, predeterminado, si no se proporciona • DEBUG • TRACE Ejemplo: `INFO`
`AWS_KVS_EDGE_LOG_MAX_FILE_SIZE`	No	Cuando el archivo de registro alcance este tamaño, se transferirá. • Mínimo: 1 • Máximo: 100 • Predeterminado: 20, si no se proporciona • Unidades: megabytes (MB) Ejemplo: 5

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY`	No	La ruta del archivo que apunta al directorio en el que se generan los registros de Amazon Kinesis Video Streams Edge Agent. El valor predeterminado es `./log` si no se proporciona. Ejemplo: `/file/path/`
`AWS_KVS_EDGE_LOG_ROLLOVER_COUNT`	No	El número de registros acumulados que se deben conservar antes de eliminarlos. • Mínimo: 1 • Máximo: 100 • Predeterminado: 10, si no se proporciona Ejemplo: 20
`AWS_KVS_EDGE_RECORDING_DIRECTORY`	No	La ruta del archivo que apunta al directorio en el que se escribirá el soporte grabado. El valor predeterminado es el directorio actual si no se proporciona. Ejemplo: `/file/path/`

Nombre de la variable de entorno	Obligatorio	Descripción
`GREENGRASS_ROOT_DIRECTORY`	No	La ruta del archivo al directorio AWS IoT Greengrass raíz. El valor predeterminado es `/greengrass/v2/` si no se proporciona. Ejemplo: `/file/path/`
`GST_DEBUG`	No	Especifica el nivel de GStreamer registros que se van a generar. Para obtener más información, consulte la Documentación de GStreamer . Ejemplo: 0
`GST_DEBUG_FILE`	No	Especifica el archivo de salida de los registros de GStreamer depuración. Si no se establece, los registros de depuración muestran un error estándar. Para obtener más información, consulte la Documentación de GStreamer . Ejemplo: `/tmp/gstreamer-logging .log`

Abra `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` y modifique el script de ejecución para añadir cualquiera de las variables de entorno anteriores.

> **Important**
>
> Asegúrese de que el script de ejecución modificado no contenga ningún carácter de tabulación. El software AWS IoT Greengrass principal no podrá leer la receta.

7. Implemente el componente Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent.

   Escriba:

```
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"
```

   Para obtener información adicional, consulte las siguientes secciones de la Guía para AWS IoT Greengrass Version 2 desarrolladores:

   - AWS IoT Greengrass Comandos CLI
   - Implemente AWS IoT Greengrass componentes en los dispositivos

8. Envíe las configuraciones a la aplicación mediante AWS CLI.

   a. Cree un archivo nuevo,*example-edge-configuration*.json.

      Pegue el código siguiente en el archivo. Este es un ejemplo de configuración que graba todos los días desde las 9:00:00 a.m. hasta las 4:59:59 p.m. (según la hora del sistema del dispositivo). AWS IoT También carga el contenido multimedia grabado todos los días de 19:00 a 21:59:59 horas.

      Para obtener más información, consulte the section called “StartEdgeConfigurationUpdate”.

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

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

b. Escriba lo siguiente AWS CLI para enviar el archivo al agente Edge de Amazon Kinesis Video Streams:

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

9. Repita el paso anterior para cada transmisión para el agente Edge de Amazon Kinesis Video Streams.

Instale el componente de gestión de AWS IoT Greengrass registros en el dispositivo

> **Note**
>
> Tenga en cuenta las CloudWatchcuotas.

Siga estos procedimientos para configurar los registros del agente Edge de Amazon Kinesis Video Streams para que se CloudWatch carguen automáticamente mediante AWS IoT Greengrass el componente de administrador de registros. Se trata de un paso opcional.

Instale el componente de administrador de AWS IoT Greengrass registros

1. Confirme que la función del AWS IoT Greengrass dispositivo tiene los permisos adecuados.
   a. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
   b. Haga clic en Funciones en el menú de navegación de la izquierda.
   c. Elija el nombre del rol de TES creado enthe section called “Configure el dispositivo AWS IoT Greengrass principal”. Utilice la barra de búsqueda si es necesario.
   d. Seleccione la política de `GreengrassV2TokenExchangeRoleAccess`.
   e. Seleccione la pestaña JSON y compruebe que la política tiene el siguiente aspecto:

      JSON

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

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

f. Si la `GreengrassV2TokenExchangeRoleAccess` política no existe o si faltan algunos permisos necesarios, cree una nueva política de IAM con estos permisos y adjúntela al rol de TES creado enthe section called “Configure el dispositivo AWS IoT Greengrass principal”.

2. Inicie sesión en AWS Management Console y abra la AWS IoT Core consola en https://console.aws.amazon.com/iot/. Compruebe que se ha seleccionado la región adecuada.
3. En el menú de navegación de la izquierda, selecciona Greengrass devices, Deployments.

   Elija el despliegue con el mismo objetivo que el elemento en el que creó. the section called “Configure el dispositivo AWS IoT Greengrass principal”
4. En la esquina superior derecha, selecciona Acciones y, a continuación, selecciona Revisar.

   En la ventana emergente que aparece, selecciona Revisar la implementación.
5. Complete las siguientes secciones:

   a. Paso 1: especifique el objetivo. Elija Siguiente.

   b. Paso 2: Seleccione los componentes.

      i. Compruebe que el componente AWS.Greengrass.cli y aws.greengrass. SecretManagerlos componentes siguen seleccionados.

      > **Important**
      >
      > No desinstale estos componentes.

      ii. Active la opción Mostrar solo los componentes seleccionados y busque aws.greengrass. LogManager.

      iii. Selecciona la casilla situada junto a aws.greengrass. LogManagery, a continuación, selecciona Siguiente.

   c. Paso 3: Configurar los componentes. Configure el componente de administrador de AWS IoT Greengrass registros para cargar los registros generados por el agente Edge de Amazon Kinesis Video Streams.

Seleccione aws.greengrass. LogManagercomponente y, a continuación, elija Configurar componente.

En la pantalla que aparece, pegue la siguiente configuración del administrador de registros en el cuadro Configuración para fusionar.

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

> **Important**
>
> `logFileDirectoryPath`En la configuración anterior, se supone que se utiliza la ubicación de salida de registro predeterminada.

> **Note**
>
> Para obtener más información sobre cada uno de los parámetros de la configuración del administrador de registros, consulte la sección del administrador de registros de la Guía para AWS IoT Greengrass Version 2 desarrolladores.

Cuando termine, seleccione Confirmar y, a continuación, elija Siguiente.

d. Paso 4: Configura los ajustes avanzados. Seleccione Siguiente.

e. Paso 5: Revisar. Seleccione Implementar.

6. Confirme que el componente y los permisos del administrador de AWS registros se instalaron correctamente.

7. En la EC2 instancia Amazon de Ubuntu, escriba `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` para comprobar que el componente recibió la configuración actualizada.

8. Inspeccione los registros AWS IoT Greengrass principales.

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

   Revise si hay errores de implementación.

   Si se ha producido un error, revise la implementación para eliminar el `aws.greengrass.LogManager` componente.

   Escriba `sudo service greengrass restart` esta opción para reiniciar el servicio AWS IoT Greengrass principal.

   Si el error de implementación estuvo relacionado con la falta de permisos, compruebe the section called “Agregue permisos a la función TES” que el rol de TES tiene los permisos adecuados. A continuación, repita esta sección.

Preguntas frecuentes sobre Amazon Kinesis Video Streams Edge Agent

Las siguientes son algunas preguntas frecuentes sobre el servicio Amazon Kinesis Video Streams Edge Agent.

¿Qué sistemas operativos admite Amazon Kinesis Video Streams Edge Agent?

Amazon Kinesis Video Streams Edge Agent actualmente es compatible con los siguientes sistemas operativos:

Ubuntu

- 2.2.x
  - AMD64
- 18.x
  - ARM

AL2

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

¿El agente Edge de Amazon Kinesis Video Streams es compatible con contenido multimedia H.265?

Amazon Kinesis Video Streams Edge Agent solo admite transmisiones H.264 elementales.

¿Funciona el agente Edge de Amazon Kinesis Video Streams en? AL2

Sí.

¿Cómo puedo ejecutar varias transmisiones dentro de un objeto AWS IoT o dispositivo?

Envíe otro the section called “StartEdgeConfigurationUpdate” al mismo Amazon Kinesis Video AWS Secrets Manager ARNs Streams/`HubDeviceArn`, pero diferente.

¿Cómo puedo editar un mensaje **`StartEdgeConfigurationUpdate`** después de haberlo enviado?

Envíe una actualización the section called “StartEdgeConfigurationUpdate” a la misma `HubDeviceArn` con el mismo ARN de Amazon Kinesis Video Streams. Cuando la aplicación recibe el mensaje de Amazon Kinesis Video Streams, anula la configuración anterior de esa transmisión. En ese momento, se efectuarán los cambios.

¿Tiene algún ejemplo común**`ScheduleConfigs`**?

El agente Amazon Kinesis Video Streams Edge utiliza la hora del sistema del dispositivo en el que se ejecuta.

Descripción	ScheduleExpression	DurationInSeconds
Grabación ininterrumpida, carga cada hora	`(null ScheduleConfig)`	
De 9:00 a.m. a 4:59:59 p.m., todos los días	`0 0 9-16 * * ? *`	3599
De 9:00 a.m. a 4:59:59 p.m., de lunes a viernes	`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
Los fines de semana de 9:00:00 a.m. a 4:59:59 p.m.	`0 0 9-16 ? * SAT,SUN *`	3599

Descripción	ScheduleExpression	DurationInSeconds
De 10:00 p. m. a 23:59:59 p. m., de lunes a viernes	`0 0 22,23 ? * MON-FRI *`	3599
De 9:00:00 a.m. a 10:00:00 a.m., todos los días	`0 0 9 * * ? *`	3600
De 16:00 a 17:59:59 todos los días	`0 0 16-17 * * ? *`	3599

Para ver más ejemplos, consulte la documentación de Quartz.

¿Hay un límite máximo de transmisión?

El agente Edge de Amazon Kinesis Video Streams tiene actualmente un límite estricto de 16 transmisiones por dispositivo. Utilice la the section called “DeleteEdgeConfiguration” API para eliminar las transmisiones de un dispositivo. La actualización de una configuración para la misma transmisión mediante el the section called “StartEdgeConfigurationUpdate” no aumenta el recuento de transmisiones del dispositivo.

¿Cómo puedo reiniciar un trabajo que se ha producido un error?

Si se produce un error, el agente Edge de Amazon Kinesis Video Streams intentará reiniciar el trabajo. Sin embargo, si se producen algunos errores (como los errores de configuración), debe reiniciar el trabajo manualmente.

Para determinar qué trabajos deben reiniciarse manualmente, consulte la FatalErrormétrica en. the section called “Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch”

Vuelva a enviar el the section called “StartEdgeConfigurationUpdate” para reiniciar el trabajo de la transmisión.

¿Cómo superviso el estado de mi agente Edge de Amazon Kinesis Video Streams?

Para obtener más información, consulte the section called “Supervise el agente Edge de Amazon Kinesis Video Streams con CloudWatch”.

Transmita vídeo a través de una VPC

Esta versión beta está disponible en versión preliminar en la región de Europa (París), eu-west-3. Para acceder a estos componentes y a nuestra guía de introducción, envíenos un correo electrónico.

El servicio de punto final de VPC Amazon Kinesis Video Streams le permite transmitir y consumir vídeo a través de la red de Amazon sin que ningún dato pase por la Internet pública.

Para solicitar el acceso, envíenos un correo electrónico con la siguiente información:

- ID de cuenta
- Transmitir ARNs
- ID de VPC

> **Note**
>
> Es posible que tardemos hasta una semana en añadirte al servicio.

Si no ha trabajado con puntos finales de VPC en el pasado, consulte la siguiente información para familiarizarse con el concepto:

- AWS PrivateLink antecedentes
- Guía de introducción a la VPC

Información adicional

Cuando te agreguen a la versión beta, te enviaremos por correo electrónico un enlace con información adicional sobre esta función.

Procedimientos de punto final de VPC

Cuotas

Las principales diferencias de cuota son:

- Cuota más baja para todo el ancho de banda APIs (2 mbps):
  - PutMedia
  - GetMedia
  - GetMediaForFragmentList
- Se permiten 10 transmisiones por cliente

Creación de un punto de conexión de

Cuando aparezca en la lista de permitidos, recibirá el nombre del servicio de punto final de VPC para Amazon Kinesis Video Streams. Se verá así. `com.amazonaws.region.kinesisvideo`

Cree un punto de enlace de VPC de interfaz para Amazon Kinesis Video Streams mediante la consola Amazon VPC o el (). AWS Command Line Interface AWS CLI

En el AWS CLI, escriba lo siguiente:

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

> **Important**
>
> El tráfico dentro de su VPC utilizará un DNS privado para enrutarse a través del punto final. Si no lo habilitas, tendrás que implementar tu propia lógica de DNS. Para obtener más información sobre el DNS privado, consulta AWS PrivateLink la documentación.

Para obtener más información sobre la AWS CLI opción, consulte create-vpc-endpoint.

Controle el acceso a los puntos finales

Puede adjuntar una política de puntos de conexión a su punto de enlace de VPC que controle el acceso a Amazon Kinesis Video Streams. La política especifica la siguiente información:

- el principal que puede realizar acciones,
- las acciones que se pueden realizar, y
- los recursos con los que se pueden realizar las acciones.

Para obtener más información, consulte Controlar el acceso a los servicios con puntos de enlace de VPC mediante políticas de puntos de conexión en la Guía. AWS PrivateLink

El siguiente es un ejemplo de una política de puntos finales para Amazon Kinesis Video Streams. Cuando se adjunta a un punto final, esta política deniega el acceso a las `PutMedia` acciones enumeradas a todos los principales de todos los recursos.

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

Ejemplos de Amazon Kinesis Video Streams

Los siguientes ejemplos de código muestran cómo trabajar con la API de Kinesis Video Streams:

Ejemplos: Envío de datos a Kinesis Video Streams

- Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink: muestra cómo crear el SDK para productores de Kinesis Video Streams para usarlo como destino GStreamer .
- Ejecute el GStreamer elemento en un contenedor de Docker: muestra cómo utilizar una imagen de Docker prediseñada para enviar vídeo del Protocolo de transmisión en tiempo real (RTSP) desde una cámara IP a Kinesis Video Streams.
- Ejemplo: transmisión desde una fuente RTSP: muestra cómo crear su propia imagen de Docker y cómo enviar vídeo RTSP desde una cámara IP a Kinesis Video Streams.
- Ejemplo: envío de datos a Kinesis Video Streams mediante PutMedia la API: muestra cómo utilizarla Utilice la biblioteca de productores de Java para enviar datos a Kinesis Video Streams que ya están en formato contenedor (MKV) mediante PutMedialа API.

Ejemplos: Recuperación de datos de Kinesis Video Streams

- KinesisVideoExample: Muestra cómo analizar y registrar fragmentos de vídeo mediante la biblioteca de analizadores de Kinesis Video Streams.
- Ejemplo: análisis y renderización de fragmentos de Kinesis Video Streams: Muestra cómo analizar y renderizar fragmentos de transmisión de vídeo de Kinesis mediante JCodecy. JFrame

Ejemplos: reproducción de datos de vídeo

- Ejemplo: utilice HLS en HTML y JavaScript: muestra cómo recuperar una sesión de streaming HLS para una transmisión de vídeo de Kinesis y reproducirla en una página web.

Requisitos previos

- En el código de ejemplo, las credenciales se proporcionan especificando un perfil que se establece en el archivo de perfil de AWS credenciales o se proporcionan las credenciales en las propiedades

del sistema Java de su entorno de desarrollo integrado (IDE). Si aún no lo ha hecho, configure primero sus credenciales. Para obtener más información, consulte Configurar AWS credenciales y regiones para el desarrollo.

- Le recomendamos que utilice un IDE Java para ver y ejecutar el código, como, por ejemplo, uno de los siguientes:
  - Eclipse Java Neon
  - JetBrains IntelliJ IDEA

Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink

En este tema se describe cómo crear el SDK para productores de Amazon Kinesis Video Streams para usarlo como GStreamer complemento.

Temas

- Descargue, cree y configure el GStreamer elemento
- Ejecuta el elemento GStreamer
- Ejemplos de comandos de lanzamiento GStreamer
- Ejecute el GStreamer elemento en un contenedor de Docker
- GStreamer referencia de parámetros de elementos

GStreameres un marco multimedia popular que utilizan varias cámaras y fuentes de vídeo para crear canales multimedia personalizados mediante la combinación de complementos modulares. El complemento Kinesis Video GStreamer Streams agiliza la integración de su canal multimedia GStreamer existente con Kinesis Video Streams. Tras la integración GStreamer, puede transmitir vídeo desde una cámara web o una cámara del Protocolo de Transmisión en Tiempo Real (RTSP) a Kinesis Video Streams para reproducirlo, almacenarlo y analizarlo en tiempo real o posteriormente.

El GStreamer complemento gestiona automáticamente la transferencia de su transmisión de vídeo a Kinesis Video Streams al encapsular la funcionalidad proporcionada por el SDK para productores de Kinesis Video Streams GStreamer en un elemento receptor,. `kvssink` El GStreamer marco proporciona un entorno gestionado estándar para crear el flujo de contenido multimedia desde un dispositivo, como una cámara u otra fuente de vídeo, para su posterior procesamiento, renderización o almacenamiento.

La GStreamer canalización suele consistir en el enlace entre una fuente (cámara de vídeo) y el elemento receptor (ya sea un reproductor para renderizar el vídeo o un almacenamiento para recuperarlo sin conexión a Internet). En este ejemplo, se utiliza el elemento SDK Producer como receptor o destino multimedia, para su fuente de vídeo (cámara web o cámara IP). El elemento del complemento que encapsula el SDK envía entonces la transmisión de vídeo a Kinesis Video Streams.

En este tema se describe cómo crear una canalización GStreamer multimedia capaz de transmitir vídeo desde una fuente de vídeo, como una cámara web o una transmisión RTSP, normalmente conectada a Kinesis Video Streams a través de etapas de codificación intermedias (mediante codificación H.264). Cuando la transmisión de vídeo esté disponible como transmisión de vídeo de Kinesis, podrá utilizarla the section called “Transmita usando la biblioteca de analizadores” para seguir procesando, reproduciendo, almacenando o analizando la transmisión de vídeo.

Descargue, cree y configure el GStreamer elemento

El ejemplo del GStreamer complemento se incluye en el SDK para productores de Kinesis Video Streams C++. Para obtener información acerca de los requisitos de SDK y de la descarga, consulte Descargue y configure el código de la biblioteca del productor de C++.

Puedes crear el receptor del SDK del GStreamer productor como una biblioteca dinámica en macOS, Ubuntu, Raspberry Pi o Windows. El GStreamer complemento se encuentra en tu `build` directorio. Para cargar este complemento, debe estar en su`GST_PLUGIN_PATH`. Ejecuta el siguiente comando:

```
export GST_PLUGIN_PATH=`pwd`/build
```

> **Note**
>
> En macOS, solo puede transmitir vídeo desde una cámara de red cuando se ejecuta GStreamer en un contenedor Docker. No es compatible el streaming de vídeo desde una cámara USB en macOS en un contenedor de Docker.

Ejecuta el elemento GStreamer

Para ejecutar GStreamer con el elemento SDK del productor de Kinesis Video Streams como receptor, utilice `gst-launch-1.0` el comando. Utilice los elementos originales que sean adecuados

para el GStreamer complemento que vaya a utilizar. Por ejemplo, v4l2src para dispositivos v4l2 en sistemas Linux o rtspsrc para dispositivos RTSP. Especifique `kvssink` como receptor (destino final de la canalización) para enviar vídeo al SDK Producer.

Además de proporcionar credenciales y proporcionar una región, el `kvssink` elemento tiene el siguiente parámetro obligatorio:

- `stream-name`— El nombre del Kinesis Video Streams de destino.

Para obtener información acerca de los parámetros opcionales `kvssink`, consulte GStreamer referencia de parámetros de elementos.

Para obtener la información más reciente sobre los GStreamer complementos y los parámetros, consulte GStreamer Complementos. También puedes usar el `gst-inspect-1.0` comando seguido del nombre de un GStreamer elemento o complemento para imprimir su información y comprobar que está disponible en tu dispositivo:

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

Si la compilación `kvssink` ha fallado o GST_PLUGIN_PATH no está configurado correctamente, el resultado tendrá un aspecto similar al siguiente:

```
No such element or plugin 'kvssink'
```

Ejemplos de comandos de lanzamiento GStreamer

Los siguientes ejemplos muestran cómo usar el `kvssink` GStreamer complemento para transmitir vídeo desde diferentes tipos de dispositivos.

Ejemplo 1: Transmitir vídeo desde una cámara RTSP en Ubuntu

El siguiente comando crea una GStreamer canalización en Ubuntu que transmite desde una cámara RTSP de red mediante el complemento GStreamer rtspsrc:

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
 rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

Ejemplo 2: Codifique y transmita vídeo desde una cámara USB en Ubuntu

El siguiente comando crea una GStreamer canalización en Ubuntu que codifica la transmisión desde una cámara USB en formato H.264 y la transmite a Kinesis Video Streams. En este ejemplo se utiliza el complemento v4l2src. GStreamer

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc  bframes=0 key-int-max=45 bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Ejemplo 3: Transmite vídeo precodificado desde una cámara USB en Ubuntu

El siguiente comando crea una GStreamer canalización en Ubuntu que transmite vídeo que la cámara ya ha codificado en formato H.264 a Kinesis Video Streams. En este ejemplo se utiliza el complemento GStreamer v4l2src.

```
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"
```

Ejemplo 4: Transmitir vídeo desde una cámara de red en macOS

El siguiente comando crea una GStreamer canalización en macOS que transmite vídeo a Kinesis Video Streams desde una cámara de red. En este ejemplo se utiliza el complemento 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"
```

Ejemplo 5: Transmita vídeo desde una cámara de red en Windows

El siguiente comando crea una GStreamer canalización en Windows que transmite vídeo a Kinesis Video Streams desde una cámara de red. En este ejemplo se utiliza el complemento 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"
```

Ejemplo 6: Transmite vídeo desde una cámara de Raspberry Pi

El siguiente comando crea una GStreamer canalización en Raspberry Pi que transmite vídeo a Kinesis Video Streams. En este ejemplo se utiliza el complemento GStreamer v4l2src.

```
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"
```

Ejemplo 7: Transmite audio y vídeo en Raspberry Pi y Ubuntu

Descubra cómo ejecutar el comando gst-launch-1.0 para comenzar el streaming de audio y vídeo en Raspberry-PI y Ubuntu.

Ejemplo 8: Transmite audio y vídeo desde las fuentes del dispositivo en macOS

Descubra cómo ejecutar el comando gst-launch-1.0 para comenzar el streaming de audio y vídeo en MacOS.

Ejemplo 9: Cargue un archivo MKV que contenga audio y vídeo

Descubra cómo ejecutar el comando gst-launch-1.0 para cargar un archivo MKV que contenga audio y vídeo. Necesitará un archivo de prueba MKV con contenido multimedia codificado en H.264 y AAC.

Ejecute el GStreamer elemento en un contenedor de Docker

Docker es una plataforma para desarrollar, implementar y ejecutar aplicaciones utilizando contenedores. El uso de Docker para crear la GStreamer canalización estandariza el entorno operativo de Kinesis Video Streams, lo que agiliza la creación y el uso de la aplicación.

Para instalar y configurar Docker, consulte lo siguiente:

- Instrucciones de descarga de Docker
- Introducción a Docker

Tras instalar Docker, puede descargar el SDK (GStreamer y el complemento) para C++ Producer de Kinesis Video Streams desde Amazon Elastic Container Registry mediante uno de los comandos que se proporcionan a continuación`docker pull`.

Para ejecutar GStreamer el elemento del SDK del productor de Kinesis Video Streams como sumidero en un contenedor de Docker, haga lo siguiente:

Temas

- Autentique su cliente de Docker
- Descargue la imagen de Docker para Ubuntu, macOS, Windows o Raspberry Pi
- Ejecute la imagen de Docker

Autentique su cliente de Docker

Autentique su cliente de Docker en el registro de Amazon ECR del que va a extraer la imagen. Debe obtener los tokens de autenticación para cada registro utilizado. Los tokens son válidos durante 12 horas. Para obtener más información, consulte Autenticación de registros en la Guía del usuario de Amazon Elastic Container Registry.

Example : autenticación con Amazon ECR

Para autenticarse con Amazon ECR, copie y pegue el siguiente comando como se muestra.

```
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
```

Si la acción se realiza correctamente, se muestra el texto `Login Succeeded` (Inicio de sesión correcto).

Descargue la imagen de Docker para Ubuntu, macOS, Windows o Raspberry Pi

Descargue la imagen de Docker en su entorno de Docker con uno de los siguientes comandos, en función de su sistema operativo:

Descargue la imagen de Docker para Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Descargue la imagen de Docker para macOS

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Descargue la imagen de Docker para Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

Descargue la imagen de Docker para Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

Para comprobar que la imagen se ha agregado correctamente, utilice el comando siguiente:

```
docker images
```

Ejecute la imagen de Docker

Utilice uno de los comandos siguientes para ejecutar la imagen de Docker, en función de su sistema operativo:

Ejecute la imagen de Docker en 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
```

Ejecuta la imagen de Docker en macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Ejecute la imagen de Docker en 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
```

Ejecute la imagen de Docker en 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 lanza el contenedor y le presenta una línea de comandos para usar los comandos dentro del contenedor.

En el contenedor, establezca las variables de entorno utilizando el siguiente comando:

```
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
```

Comience a transmitir y `kvssink` utilice el `gst-launch-1.0` para ejecutar una canalización adecuada para su dispositivo y fuente de vídeo. Para ver, por ejemplo, canalizaciones, consulteEjemplos de comandos de lanzamiento GStreamer .

GStreamer referencia de parámetros de elementos

Para enviar un vídeo al SDK de C++ para productores de Amazon Kinesis Video Streams, debe *`kvssink`* especificarlo como destino final o destino final de la canalización. Esta referencia proporciona información sobre parámetros `kvssink` obligatorios y opcionales. Para obtener más información, consulte the section called “GStreamer Complemento: kvssink”.

Temas

- the section called “Proporcione las credenciales a `kvssink`”
- the section called “Proporcione una región a `kvssink`”
- the section called “`kvssink`parámetros opcionales”

Proporcione las credenciales a **`kvssink`**

Para permitir que el `kvssink` GStreamer elemento realice solicitudes AWS, proporcione AWS las credenciales que utilizará cuando llame al servicio Amazon Kinesis Video Streams. La cadena de proveedores de credenciales busca las credenciales en el siguiente orden:

1. AWS IoT credenciales

Para configurar AWS IoT las credenciales, consultethe section called “Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT”.

El valor del `iot-credentials` parámetro debe empezar por `iot-certificate,` e ir seguido de una lista separada por comas de los siguientes pares *`key`* =*`value`*.

Clave	Obligatorio	Descripción
`ca-path`	Sí	Ruta del archivo al certificado de CA utilizado para establecer la confianza con el servicio de backend a través de TLS. Example Ejemplo: `/file/path/to/certificate.pem`
`cert-path`	Sí	Ruta del archivo al certificado X.509. Example Ejemplo: `/file/path/to/certificateID -certificate.pem.crt`
`endpoint`	Sí	El punto de conexión del proveedor de AWS IoT Core credenciales de su cuenta.

Clave	Obligatorio	Descripción
		AWS Consulte la guía AWS IoT para desarrolladores. Example Ejemplo: *credential-account-specific-prefix* `.credentials.iot.` *aws-region* `.amazonaws.com`
`key-path`	Sí	Ruta del archivo a la clave privada utilizada en el public/private key pair. Example Ejemplo: */file/path/to/certificateID* `-private.pem.key`
`role-aliases`	Sí	El nombre del alias del rol que apunta al rol de AWS IAM que se va a usar al conectarse AWS IoT Core. Example Ejemplo: *KvsCameraIoTRoleAlias*

Clave	Obligatorio	Descripción
`iot-thing-name`	No	La `iot-thing-name` es opcional. Si no `iot-thing-name` se proporciona, se utiliza el valor del `stream-name` parámetro. Example Ejemplo: *`kvs_example_camera`*

Example

Ejemplo:

```
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. Variables de entorno

Para poder `kvssink` utilizar las credenciales del entorno, defina las siguientes variables de entorno:

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_ACCESS_KEY_ID`	Sí	La clave de AWS acceso que se utiliza para acceder a Amazon Kinesis Video Streams.
`AWS_SECRET_ACCESS_KEY`	Sí	La clave AWS secreta asociada a la clave de acceso.

Nombre de la variable de entorno	Obligatorio	Descripción
`AWS_SESSION_TOKEN`	No	Especifica el valor del token de sesión necesario si utiliza credenciales de seguridad temporales directamente de AWS STS las operaciones.

La configuración de la variable de entorno cambia el valor usado hasta el final de su sesión del intérprete de comandos o hasta que otorgue a la variable un valor diferente. Para que las variables sean persistentes en futuras sesiones, configúrelas en el script de inicio de su shell.

3. **`access-key`**, **`secret-key`** parámetros

Para especificar las credenciales directamente como `kvssink` parámetro, defina los siguientes parámetros:

`kvssink`Nombre del parámetro	Obligatorio	Descripción
`access-key`	Sí	La clave de AWS acceso que se utiliza para acceder a Amazon Kinesis Video Streams.
`secret-key`	Sí	La clave AWS secreta asociada a la clave de acceso.
`session-token`	No	Especifica el valor del token de sesión necesario si utiliza credenciales de seguridad temporales directamente de AWS STS las operaciones.

Example

Uso de credenciales estáticas:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE"
```

Example

Uso de credenciales temporales:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 access-key="AKIDEXAMPLE" secret-key="SKEXAMPLE" session-token="STEXAMPLE"
```

4. Archivo de credenciales

> **Important**
>
> Si ha seleccionado uno de los métodos anteriores, no podrá usar el `credential-path` `kvssink` parámetro.

kvssinkNombre del parámetro	Obligatorio	Descripción
`credential-path`	Sí	Ruta al archivo de texto que contiene las credenciales en un formato específico.

El archivo de texto debe contener credenciales en uno de los siguientes formatos:

- CREDENCIALES *`YourAccessKey YourSecretKey`*
- CREDENCIALES *`YourAccessKey Expiration YourSecretKey SessionToken`*

Example

Ejemplo: el *`credentials`*`.txt` archivo se encuentra en `/home/ubuntu` y contiene lo siguiente:

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

Para usarlo en`kvssink`, escriba:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
 credential-path="/home/ubuntu/credentials.txt"
```

> **Note**
>
> El tiempo de caducidad debería ser de al menos 5 + 30 + 3 = 38 segundos en el futuro. El período de gracia se define como la `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` variable en `IotCredentialProvider.h`. Si las credenciales están muy cerca de caducar al empezar`kvssink`, recibirá el código de error`0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`.

> **Important**
>
> `kvssink`no modifica el archivo de credenciales. Si utiliza credenciales temporales, una fuente externa debe actualizar el archivo de credenciales antes de la fecha de caducidad menos el período de gracia.

Proporcione una región a **`kvssink`**

El orden de búsqueda de regiones es el siguiente:

1. `AWS_DEFAULT_REGION`la variable de entorno se revisa primero. Si está establecida, esa región se usa para configurar el cliente.
2. `aws-region`El parámetro se revisa a continuación. Si está establecido, esa región se usa para configurar el cliente.
3. Si no se ha utilizado ninguno de los métodos anteriores, el `kvssink` valor predeterminado es. `us-west-2`

`kvssink`parámetros opcionales

El elemento `kvssink` tiene los siguientes parámetros opcionales. Para obtener más información sobre estos parámetros, consulte Estructuras de transmisión de vídeo de Kinesis.

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`stream-name`	El nombre de la transmisión de vídeo de Amazon Kinesis de destino. **Important** Si no se especific a ningún nombre de transmisión, se utilizará el nombre de transmisión predeterm inado: «DEFAULT_ STREAM». Si aún no existe una transmisi ón con ese nombre predeterm inado, se creará.		
`absolute-fragment-times`	Si se utilizan tiempos de fragmentos absolutos.	Booleano	true
`access-key`	La clave de AWS acceso que se utiliza para acceder		

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
	a Kinesis Video Streams. Debe tener AWS configuradas las credenciales o proporcionar este parámetro. Para proporcionar esta información, escriba lo siguiente: `export AWS_ACCES S_KEY_ID=`		
`avg-bandwidth-bps`	El ancho de banda medio previsto para la secuencia.	Bits por segundo	4194304

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`aws-region`	El que Región de AWS se va a usar. **Note** También puede proporcionar la variable de `AWS_DEFAULT_REGION` entorno a la región. Las variables de entorno tienen prioridad si se establecen los parámetros de la variable de entorno y kvssink. **Important** La región se establecerá de forma predeterminada `us-west-2` si no se especifica lo contrario.	Cadena	`"us-west-2"`

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`buffer-duration`	La duración del búfer de la secuencia.	Segundos	120
`codec-id`	El ID del códec de la secuencia.	Cadena	`"V_MPEG4/ISO/AVC"`
`connection-staleness`	El tiempo transcurrido, al que se denomina devolución de señal por inactividad de la transmisión.	Segundos	60
`content-type`	El tipo de contenido de la secuencia.	Cadena	`"video/h264"`
`fragment-acks`	Si se debe usar el fragmento. ACKs	Booleano	true
`fragment-duration`	La duración del fragmento que desea.	Milisegundos	2000
`framerate`	La velocidad de fotogramas prevista.	Fotogramas por segundo	25
`frame-timecodes`	Si se utilizan códigos de tiempo de fotograma o se generan marcas de tiempo utilizando la devolución de llamada de tiempo actual.	Booleano	true
`key-frame-fragmentation`	Si se producen fragmentos en un fotograma clave.	Booleano	true

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`log-config`	La ruta de configuración de registro.	Cadena	`"../kvs_log_configuration"`
`max-latency`	La latencia máxima de la secuencia.	Segundos	60
`recalculate-metrics`	Si se vuelven a calcular las métricas.	Booleano	true
`replay-duration`	La duración para restaurar el lector actual para reproducir durante un error si está habilitada la reactivación.	Segundos	40
`restart-on-error`	Si se reinicia cuando se produce un error.	Booleano	true
`retention-period`	El periodo de tiempo que se mantiene la secuencia.	Horas	2
`rotation-period`	El período de rotación de claves. Para obtener más información, consulte AWS KMS Teclas giratorias.	Segundos	3600

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`secret-key`	La clave AWS secreta que se usa para acceder a Kinesis Video Streams. Debe tener AWS configuradas las credenciales o proporcionar este parámetro. `export AWS_SECRET_ACCESS_KEY=`		
`session-token`	Especifica el valor del token de sesión necesario si utiliza credenciales de seguridad temporales directamente de AWS STS las operaciones.		
`storage-size`	El tamaño de almacenamiento del dispositivo en mebibyte (MiB). Para obtener información acerca de la configuración del almacenamiento de dispositivo, consulte StorageInfo.	Mebibyte (MiB)	128

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`streaming-type`	El tipo de streaming . Los valores válidos son: • 0: tiempo real • 1: casi tiempo real (no se admite actualmente) • 2: desconectado	Enum `GstKvsSinkStreamingType`	0: tiempo real
`timecode-scale`	La escala de código de tiempo de MKV.	Milisegundos	1
`track-name`	El nombre de la pista MKV.	Cadena	`"kinesis_video"`

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
`iot-certificate`	AWS IoT credenciales que se utilizarán en el `kvssink` elemento. `iot-certificate` acepta las siguientes claves y valores: **Note** El `iot-thing-name` es opcional. Si no `iot-thing-name` se proporciona, se utiliza el valor del `stream-name` parámetro. • `endpoint=iotcredentialsproviderendpoint` • `cert-path =/localdirectorypath /to/certificate`	Cadena	Ninguno

Parámetro	Descripción	Unidad/tipo	Predeterminado/a
	• `key-path=/localdirectorypath /to/private/key` • `ca-path=/localdirectorypath/to/ca-cert` • `role-aliases =role-aliases` • `iot-thing-name=YourIotThingName`		

Ejemplo: envío de datos a Kinesis Video Streams mediante PutMedia la API

En este ejemplo se muestra cómo utilizar la PutMediaAPI. Muestra cómo enviar datos que ya están en formato contenedor (MKV). Si los datos deben agruparse en un formato contenedor antes de enviarlos (por ejemplo, si va a agrupar los datos de vídeo de la cámara en fotogramas), consulteCargar a Kinesis Video Streams.

> **Note**
>
> La `PutMedia` operación solo está disponible en C++ y Java SDKs. Esto se debe a la gestión dúplex completa de las conexiones, el flujo de datos y las confirmaciones. No se admite en otros idiomas.

Este ejemplo incluye los siguientes pasos:

- Descarga y configura el código

- Escribe y examina el código
- Ejecute y verifique el código

Descarga y configura el código

Siga los pasos para descargar el código de ejemplo de Java, importar el proyecto a su IDE de Java, configurar las ubicaciones de las bibliotecas y configurar el código para usar sus AWS credenciales.

1. Cree un directorio y clone el código fuente del ejemplo desde el GitHub repositorio. El ejemplo de `PutMedia` forma parte de la Java.

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

2. Abra el IDE de Java que esté utilizando (por ejemplo, Eclipse o IntelliJ IDEA) e importe el proyecto Apache Maven que ha descargado:

   - En Eclipse: seleccione File (Archivo), Import... (Importar...), Maven (Maven), Existing Maven Projects (Proyectos Maven existentes) y vaya a la raíz del paquete descargado. Seleccione el archivo `pom.xml`.
   - En IntelliJ Idea: seleccione Import. Vaya al archivo `pom.xml` en la raíz del paquete descargado.

   Para obtener más información, consulte la documentación del IDE relacionada.

3. Actualice el proyecto, de modo que el IDE pueda encontrar las bibliotecas que ha importado.
   - Para IntelliJ IDEA, realice lo siguiente:
     a. Abra el menú contextual (clic secundario) en el directorio lib del proyecto y seleccione Add as library.
     b. Elija Archivo y, a continuación, elija Estructura del proyecto.
     c. En Project Settings, seleccione Modules.
     d. En la pestaña Sources (Orígenes), configure Language Level (Nivel de lenguaje) en **7** o más.
   - Para Eclipse, haga lo siguiente:
     a. Abra el menú contextual (clic secundario) del proyecto y seleccione Properties, Java Build Path, Source. A continuación, proceda del modo siguiente:

1. En la pestaña Source, haga doble clic en Native library location.
2. En el asistente Native Library Folder Configuration, seleccione Workspace.
3. En la opción Native Library Folder, seleccione el directorio lib del proyecto.

b. Abra el menú contextual (clic secundario) del proyecto y seleccione Properties. A continuación, proceda del modo siguiente:

1. En la pestaña Libraries, elija Add Jars.
2. En el asistente JAR selection (Selección de JAR), seleccione todos los archivos .jar del directorio `lib` del proyecto.

Escribe y examina el código

El ejemplo de API `PutMedia` () muestra el siguiente patrón de codificación:`PutMediaDemo`

Temas

- Crea el PutMediaClient
- Transmisión de medios y pausa del subproceso

Los ejemplos de código de esta sección pertenecen a la clase `PutMediaDemo`.

Crea el PutMediaClient

La creación del `PutMediaClient` objeto requiere los siguientes parámetros:

- La URI del punto de conexión `PutMedia`.
- Un `InputStream` que señale al archivo MKV que se ha de transmitir.
- El nombre de la transmisión. En este ejemplo se utiliza la misma transmisión creada en la Utilice la biblioteca de productores de Java (`my-stream`). Para utilizar otra transmisión, cambie el siguiente parámetro:

```
private static final String STREAM_NAME="my-stream";
```

> **Note**
>
> El ejemplo de la `PutMedia` API no crea una transmisión. Debe crear una transmisión mediante la aplicación de prueba de la Utilice la biblioteca de productores de Java consola Kinesis Video Streams o AWS CLI la.

- Marca temporal actual.
- Tipo de código temporal. En el ejemplo se utiliza `RELATIVE`, lo que indica que la marca temporal es relativa al inicio del contenedor.
- Un objeto `AWSKinesisVideoV4Signer` que comprueba que los paquetes recibidos han sido enviados por el remitente autorizado.
- El ancho de banda de subida máximo en Kbps,
- Un objeto `AckConsumer` para recibir los reconocimientos de los paquetes recibidos.

El siguiente código crea el objeto `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();
```

Transmisión de medios y pausa del subproceso

Tras crear el cliente, la muestra inicia el streaming asincrónico con `putMediaInBackground`. Posteriormente, el subproceso principal se pausa con `latch.await` hasta que devuelve `AckConsumer` y en ese punto se cierra el cliente.

```
/* start streaming video in a background thread */
           client.putMediaInBackground();

           /* wait for request/response to complete */
           latch.await();

           /* close the client */
           client.close();
```

Ejecute y verifique el código

Para ejecutar el ejemplo de API `PutMedia`, siga estos pasos:

1. Cree una transmisión con el nombre `my-stream` correspondiente en la consola de Kinesis Video Streams o mediante AWS CLI.
2. Cambie el directorio de trabajo actual al directorio del SDK Producer de Java:

   ```
   cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
   producer-sdk-java/
   ```

3. Compile el SDK de Java y la aplicación de demostración:

   ```
   mvn package
   ```

4. Cree un nombre de archivo temporal en el directorio `/tmp`:

   ```
   jar_files=$(mktemp)
   ```

5. Cree una cadena classpath de dependencias desde el repositorio local a un archivo:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. Establezca el valor de la variable de entorno `LD_LIBRARY_PATH` del modo siguiente:

```
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. Ejecute la demostración desde la línea de comandos de la siguiente manera, proporcionando sus AWS credenciales:

```
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. Abra la consola de Kinesis Video Streams y elija su transmisión en la página Administrar transmisiones. El vídeo se reproduce en el panel Video Preview.

Ejemplo: transmisión desde una fuente RTSP

C++Contiene una definición para un contenedor Docker que se conecta a una cámara de red del Protocolo de transmisión en tiempo real (RTSP). El uso de Docker estandariza el entorno operativo de Kinesis Video Streams, lo que agiliza la creación y el uso de la aplicación.

El siguiente procedimiento demuestra cómo configurar y utilizar la aplicación de demostración RTSP.

Temas

- Tutoriales de vídeo
- Requisitos previos
- Cree la imagen de Docker
- Ejecute la aplicación de ejemplo de RTSP

Tutoriales de vídeo

Este vídeo muestra cómo configurar una Raspberry Pi para enviar feeds RTSP a la AWS nube y a Amazon Kinesis Video Streams. Se trata de una demostración end-to-end.

En este vídeo se muestra cómo capturar imágenes de un feed para utilizar la visión artificial y Amazon Rekognition para procesar las imágenes y enviar alertas.

Requisitos previos

Para ejecutar la aplicación de ejemplo RTSP de Kinesis Video Streams, debe disponer de lo siguiente:

- Docker: para obtener información sobre cómo instalar y utilizar Docker, consulte los siguientes enlaces:
  - Instrucciones de descarga de Docker
  - Introducción a Docker
- Origen cámara de red RTSP: para obtener más información acerca de las cámaras recomendadas, consulte Requisitos del sistema.

Cree la imagen de Docker

Primero, cree la imagen de Docker en la que se ejecutará la aplicación de demostración.

1. Clone el repositorio de demostraciones de Amazon Kinesis Video Streams.

   ```
   git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
   ```

2. Cambie al directorio que contiene el Dockerfile. En este caso, es el directorio docker-rtsp.

   ```
   cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
   ```

3. Usa el siguiente comando para crear la imagen de Docker. Este comando crea la imagen y la etiqueta como rtspdockertest.

   ```
   docker build -t rtspdockertest .
   ```

4. Ejecute `docker images` y busque el ID de imagen con el que está etiquetado. `rtspdockertest`

Por ejemplo, en el resultado de ejemplo que aparece a continuación, el `IMAGE ID` es`54f0d65f69b2`.

```
REPOSITORY        TAG       IMAGE ID        CREATED           PLATFORM       SIZE
       BLOB SIZE
rtspdockertest    latest    54f0d65f69b2    10 minutes ago    linux/arm64    653.1
 MiB    292.4 MiB
```

Lo necesitará en un paso posterior.

Ejecute la aplicación de ejemplo de RTSP

Puede ejecutar la aplicación de ejemplo de RTSP desde dentro o fuera del contenedor de Docker. Siga las instrucciones correspondientes que aparecen a continuación.

Temas

- Dentro del contenedor Docker
- Fuera del contenedor de Docker

Dentro del contenedor Docker

Ejecute la aplicación de ejemplo de RTSP

1. Inicie el contenedor Docker de Amazon Kinesis Video Streams mediante el siguiente comando:

```
docker run -it YourImageId /bin/bash
```

2. Para iniciar la aplicación de muestra, proporcione sus AWS credenciales, el nombre de la transmisión de vídeo de Amazon Kinesis y la URL de la cámara de red RTSP.

> **Important**
>
> Si utiliza credenciales temporales, también tendrá que proporcionar las suyas. `AWS_SESSION_TOKEN` Consulta el segundo ejemplo que aparece a continuación.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
```

```
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Credenciales temporales:

```
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. Inicie sesión en la consola de Kinesis Video Streams AWS Management Console y ábrala.

   Vea la transmisión.

4. Para salir del contenedor de Docker, cierra la ventana de la terminal o escribe`exit`.

Fuera del contenedor de Docker

Desde fuera del contenedor de Docker, usa el siguiente 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"
```

Ejemplo: análisis y renderización de fragmentos de Kinesis Video Streams

Transmita usando la biblioteca de analizadoresContiene una aplicación de demostración denominada `KinesisVideoRendererExample` que muestra cómo analizar y renderizar fragmentos de transmisiones de vídeo de Amazon Kinesis. El ejemplo se utiliza JCodecpara decodificar los fotogramas codificados en H.264 que se ingieren con la aplicación. Ejemplo: complemento del GStreamer SDK para productores de Kinesis Video Streams: kvssink Después de decodificar el marco utilizando JCodec, la imagen visible se renderiza utilizando. JFrame

Este ejemplo le muestra cómo hacer lo siguiente:

- Recupere fotogramas de una transmisión de vídeo de Kinesis mediante la `GetMedia` API y procese la transmisión para su visualización.
- Vea el contenido de vídeo de las transmisiones en una aplicación personalizada en lugar de utilizar la consola de Kinesis Video Streams.

También puede usar las clases de este ejemplo para ver contenido de transmisión de vídeo de Kinesis que no esté codificado como H.264, como una transmisión de archivos JPEG que no requieren decodificación antes de mostrarse.

El siguiente procedimiento demuestra cómo configurar y utilizar la aplicación de demostración Renderer.

Requisitos previos

Para examinar y utilizar la biblioteca de ejemplos de Renderer, debe tener lo siguiente:

- Una cuenta de Amazon Web Services (AWS). Si aún no tiene una AWS cuenta, consulte Introducción a Kinesis Video Streams.
- Un entorno de desarrollo integrado (IDE) de Java, como Eclipse Java Neon o JetBrains IntelliJ Idea.

Ejemplo de ejecución del renderizador

1. Cree un directorio y, a continuación, clone el código fuente del ejemplo desde el GitHub repositorio.

   ```
   git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
   ```

2. Abra el IDE de Java que utilice (por ejemplo, Eclipse o IntelliJ IDEA) e importe el proyecto de Apache Maven que ha descargado:

   - En Eclipse: seleccione File, Import, Maven, Existing Maven Projects. Vaya al directorio `kinesis-video-streams-parser-lib`.
   - En IntelliJ Idea: seleccione Import. Vaya al archivo `pom.xml` en la raíz del paquete descargado.

> **Note**
>
> Si IntelliJ no encuentra sus dependencias, es posible que tenga que hacer lo siguiente:
>
> - Compilación limpia: elija File (Archivo), Settings (Configuración), Build, Execution, Deployment (Compilación, ejecución, implementación), Compiler (Compilador). Compruebe que esté seleccionada la opción Borrar el directorio de salida al reconstruir y, a continuación, elija Construir, compilar proyecto.
> - Vuelva a importar el proyecto: abra el menú contextual (clic con el botón derecho) del proyecto y seleccione Maven, Reimport (Volver a importar).

Para obtener más información, consulte la documentación del IDE relacionada.

3. Desde su IDE de Java, abra `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest`.
4. Elimine la directiva `@Ignore` del archivo.
5. Actualice el `.stream` parámetro con el nombre de la transmisión de vídeo de Kinesis.
6. Ejecute la prueba `KinesisVideoRendererExample`.

Cómo funciona

La aplicación de ejemplo muestra lo siguiente:

- Envío de datos MKV
- Análisis de fragmentos de MKV para convertirlos en fotogramas
- Decodificar y mostrar el marco

Envío de datos MKV

En el ejemplo, se envían muestras de datos MKV desde el `rendering_example_video.mkv` archivo y se utilizan `PutMedia` para enviar datos de vídeo a una transmisión denominada. render-example-stream

La aplicación crea un `PutMediaWorker`:

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

Para obtener más información acerca de la clase `PutMediaWorker`, consulte Llama PutMedia en la documentación de Transmita usando la biblioteca de analizadores.

Análisis de fragmentos de MKV para convertirlos en fotogramas

A continuación, el ejemplo recupera y analiza los fragmentos MKV de la secuencia utilizando un `GetMediaWorker`:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getFrameVisitor());
executorService.submit(getMediaWorker);
```

Para obtener más información acerca de la clase `GetMediaWorker`, consulte Llama GetMedia en la documentación de Transmita usando la biblioteca de analizadores.

Decodificar y mostrar el marco

A continuación, el ejemplo decodifica y muestra el marco utilizando JFrame.

El siguiente ejemplo de código procede de la clase `KinesisVideoFrameViewer`, que amplía `JFrame`:

```
 public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

La imagen se muestra como una instancia de java.awt.image. BufferedImage. Para ver ejemplos que ilustran cómo trabajar con `BufferedImage`, consulte Reading/Loading an Image.

referencia de la API

Las secciones de este nodo contienen la documentación de referencia de la API. Usa la tabla de contenido del panel izquierdo para ir a las distintas secciones de referencia de la API.

Acciones

Amazon Kinesis Video Streams admite las siguientes acciones:

- CreateSignalingChannel
- CreateStream
- DeleteEdgeConfiguration
- DeleteSignalingChannel
- DeleteStream
- DescribeEdgeConfiguration
- DescribeImageGenerationConfiguration
- DescribeMappedResourceConfiguration
- DescribeMediaStorageConfiguration
- DescribeNotificationConfiguration
- DescribeSignalingChannel
- DescribeStream
- GetDataEndpoint
- GetSignalingChannelEndpoint
- ListEdgeAgentConfigurations
- ListSignalingChannels
- ListStreams
- ListTagsForResource
- ListTagsForStream
- StartEdgeConfigurationUpdate
- TagResource

- TagStream
- UntagResource
- UntagStream
- UpdateDataRetention
- UpdateImageGenerationConfiguration
- UpdateMediaStorageConfiguration
- UpdateNotificationConfiguration
- UpdateSignalingChannel
- UpdateStream

Amazon Kinesis Video Streams Media admite las siguientes acciones:

- GetMedia
- PutMedia

Los medios archivados de Amazon Kinesis Video Streams admiten las siguientes acciones:

- GetClip
- GetDASHStreamingSessionURL
- GetHLSStreamingSessionURL
- GetImages
- GetMediaForFragmentList
- ListFragments

Los canales de señalización de vídeo de Amazon Kinesis admiten las siguientes acciones:

- GetIceServerConfig
- SendAlexaOfferToMaster

El almacenamiento WebRTC de vídeo de Amazon Kinesis admite las siguientes acciones:

- JoinStorageSession
- JoinStorageSessionAsViewer

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams admite las siguientes acciones:

- CreateSignalingChannel
- CreateStream
- DeleteEdgeConfiguration
- DeleteSignalingChannel
- DeleteStream
- DescribeEdgeConfiguration
- DescribeImageGenerationConfiguration
- DescribeMappedResourceConfiguration
- DescribeMediaStorageConfiguration
- DescribeNotificationConfiguration
- DescribeSignalingChannel
- DescribeStream
- GetDataEndpoint
- GetSignalingChannelEndpoint
- ListEdgeAgentConfigurations
- ListSignalingChannels
- ListStreams
- ListTagsForResource
- ListTagsForStream
- StartEdgeConfigurationUpdate
- TagResource
- TagStream
- UntagResource
- UntagStream
- UpdateDataRetention
- UpdateImageGenerationConfiguration
- UpdateMediaStorageConfiguration

- UpdateNotificationConfiguration
- UpdateSignalingChannel
- UpdateStream

CreateSignalingChannel

Servicio: Amazon Kinesis Video Streams

Crea un canal de señalización.

`CreateSignalingChannel` es una operación asíncrona.

Sintaxis de la solicitud

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelName": "string",
   "ChannelType": "string",
   "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelName

Un nombre para el canal de señalización que está creando. Debe ser único para cada uno Cuenta de AWS y Región de AWS.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: sí

ChannelType

Un tipo de canal de señalización que está creando. Actualmente, `SINGLE_MASTER` es el único tipo de canal admitido.

Tipo: cadena

Valores válidos: `SINGLE_MASTER | FULL_MESH`

Obligatorio: no

SingleMasterConfiguration

Estructura que contiene la configuración del tipo `SINGLE_MASTER` de canal. La configuración predeterminada para el tiempo de vida del mensaje del canal es de 60 segundos (1 minuto).

Tipo: objeto SingleMasterConfiguration

Obligatorio: no

Tags

Un conjunto de etiquetas (pares clave-valor) que quieres asociar a este canal.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 50 artículos.

Obligatorio: no

Sintaxis de la respuesta

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

{
   "ChannelARN": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal creado.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

AccountChannelLimitExceededException

Ha alcanzado el límite máximo de canales de señalización activos para ello Cuenta de AWS en esta región.

Código de estado HTTP: 400

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

TagsPerResourceExceededLimitException

Ha superado el límite de etiquetas que puede asociar al recurso. Una transmisión de vídeo de Kinesis puede admitir hasta 50 etiquetas.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

CreateStream

Servicio: Amazon Kinesis Video Streams

Crea una nueva transmisión de vídeo de Kinesis.

Cuando crea una nueva transmisión, Kinesis Video Streams le asigna un número de versión. Cuando cambia los metadatos de la transmisión, Kinesis Video Streams actualiza la versión.

`CreateStream` es una operación asíncrona.

Para obtener más información acerca de cómo funciona el servicio, consulte How it Works (Cómo funciona).

Debe tener permisos para la acción `KinesisVideo:CreateStream`.

Sintaxis de la solicitud

```
POST /createStream HTTP/1.1
Content-type: application/json

{
   "DataRetentionInHours": number,
   "DeviceName": "string",
   "KmsKeyId": "string",
   "MediaType": "string",
   "StreamName": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

DataRetentionInHours

El número de horas que desea conservar los datos de la transmisión. Kinesis Video Streams conserva los datos en un almacén de datos que está asociado a la transmisión.

El valor predeterminado es 0, lo que indica que el flujo no conserva los datos. El mínimo es de 1 hora.

Cuando el `DataRetentionInHours` valor es 0, los consumidores pueden seguir consumiendo los fragmentos que permanecen en el búfer del host del servicio, que tiene un límite de tiempo de retención de 5 minutos y un límite de retención de memoria de 200 MB. Los fragmentos se eliminan del búfer cuando se alcanza cualquiera de los dos límites.

Tipo: entero

Rango válido: valor mínimo de 0.

Obligatorio: no

DeviceName

El nombre del dispositivo que está grabando en la transmisión.

> **Note**
>
> En la implementación actual, Kinesis Video Streams no usa este nombre.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

KmsKeyId

El ID de la clave AWS Key Management Service (AWS KMS) que desea que Kinesis Video Streams utilice para cifrar los datos de la transmisión.

Si no se especifica ningún identificador de clave, se utiliza la clave predeterminada gestionada por Kinesis Video ()`aws/kinesisvideo`.

Para obtener más información, consulte DescribeKey.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 2048 caracteres.

Patrón: `.+`

Obligatorio: no

MediaType

El tipo de medio de la transmisión. Los consumidores de la transmisión pueden usar esta información al procesar la transmisión. Para obtener más información sobre los tipos de medios, consulte Tipos de medios. Si decide especificarlos`MediaType`, consulte los requisitos de nomenclatura para ver las directrices.

Los ejemplos de valores válidos incluyen "video/h264" and "video/h264,audio/aac».

Este parámetro es opcional; el valor predeterminado es `null` (o está vacío en JSON).

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Obligatorio: no

StreamName

Un nombre para la transmisión que está creando.

El nombre de la transmisión es un identificador de la transmisión y debe ser único para cada cuenta y región.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: sí

Tags

Una lista de etiquetas para asociar a la transmisión especificada. Cada etiqueta es un par clave-valor (el valor es opcional).

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 50 elementos.

Limitaciones de longitud de la clave: longitud mínima de 1. Longitud máxima de 128.

Patrón de clave: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "StreamARN": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccountStreamLimitExceededException

El número de transmisiones creadas para la cuenta es demasiado alto.

Código de estado HTTP: 400

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

DeviceStreamLimitExceededException

No implementado.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidDeviceException

No implementado.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

TagsPerResourceExceededLimitException

Ha superado el límite de etiquetas que puede asociar al recurso. Una transmisión de vídeo de Kinesis puede admitir hasta 50 etiquetas.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DeleteEdgeConfiguration

Servicio: Amazon Kinesis Video Streams

Una API asíncrona que elimina la configuración perimetral existente de una transmisión, así como los medios correspondientes, del agente perimetral.

Al invocar esta API, el estado de sincronización se establece en. `DELETING` Se inicia un proceso de eliminación, en el que se detienen los trabajos perimetrales activos y se eliminan todos los archivos multimedia del dispositivo perimetral. El tiempo de eliminación varía en función de la cantidad total de contenido multimedia almacenado. Si se produce un error en el proceso de eliminación, el estado de la sincronización cambia a`DELETE_FAILED`. Tendrás que volver a intentar la eliminación.

Cuando el proceso de eliminación se haya completado correctamente, ya no se podrá acceder a la configuración perimetral.

> **Note**
>
> Esta API no está disponible en la región de AWS África (Ciudad del Cabo), af-south-1.

Sintaxis de la solicitud

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de . Especifique el o el`StreamName`. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo del que se va a eliminar la configuración de borde. Especifique el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

StreamEdgeConfigurationNotFoundException

La excepción se representa cuando Amazon Kinesis Video Stream no encuentra la configuración perimetral de una transmisión que haya especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python

- AWS SDK para Ruby V3

DeleteSignalingChannel

Servicio: Amazon Kinesis Video Streams

Elimina un canal de señalización especificado. `DeleteSignalingChannel`es una operación asíncrona. Si no especificas la versión actual del canal, se eliminará la versión más reciente.

Sintaxis de la solicitud

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "CurrentVersion": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal de señalización que desea eliminar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

CurrentVersion

La versión actual del canal de señalización que desea eliminar. Puede obtener la versión actual invocando las operaciones de la `ListSignalingChannels` API `DescribeSignalingChannel` o.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

VersionMismatchException

La versión de transmisión que especificó no es la versión más reciente. Para obtener la versión más reciente, usa la DescribeStreamAPI.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DeleteStream

Servicio: Amazon Kinesis Video Streams

Elimina una transmisión de vídeo de Kinesis y los datos contenidos en la transmisión.

Este método marca la transmisión para su eliminación y hace que los datos de la transmisión sean inaccesibles inmediatamente.

Para asegurarse de que dispone de la última versión de la transmisión antes de eliminarla, puede especificar la versión de la transmisión. Kinesis Video Streams asigna una versión a cada transmisión. Al actualizar una transmisión, Kinesis Video Streams asigna un nuevo número de versión. Para obtener la versión más reciente de la transmisión, utilice la `DescribeStream` API.

Esta operación necesita permiso para la acción `KinesisVideo:DeleteStream`.

Sintaxis de la solicitud

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "StreamARN": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

CurrentVersion

Opcional: la versión de la transmisión que quieres eliminar.

Especifique la versión como medida de seguridad para asegurarse de que está eliminando la transmisión correcta. Para obtener la versión de la transmisión, usa la `DescribeStream` API.

Si no se especifica, solo `CreationTime` se marca la casilla antes de eliminar la transmisión.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión que deseas eliminar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

VersionMismatchException

La versión de transmisión que especificó no es la versión más reciente. Para obtener la versión más reciente, usa la DescribeStreamAPI.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeEdgeConfiguration

Servicio: Amazon Kinesis Video Streams

Describe la configuración perimetral de una transmisión que se estableció mediante la `StartEdgeConfigurationUpdate` API y el estado más reciente de las tareas de grabación y carga del agente perimetral. Utilice esta API para obtener el estado de la configuración y determinar si está sincronizada con el agente perimetral. Utilice esta API para evaluar el estado del agente perimetral.

> **Note**
>
> Esta API no está disponible en la región de AWS África (Ciudad del Cabo), af-south-1.

Sintaxis de la solicitud

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de . Especifique el o el`StreamName`. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo cuya configuración de borde desea actualizar. Especifique el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "CreationTime": number,
   "EdgeAgentStatus": {
      "LastRecorderStatus": {
         "JobStatusDetails": "string",
         "LastCollectedTime": number,
         "LastUpdatedTime": number,
         "RecorderStatus": "string"
      },
      "LastUploaderStatus": {
         "JobStatusDetails": "string",
         "LastCollectedTime": number,
         "LastUpdatedTime": number,
         "UploaderStatus": "string"
      }
   },
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
```

```
            "LocalSizeConfig": {
               "MaxLocalMediaSizeInMB": number,
               "StrategyOnFullSize": "string"
            }
         },
         "HubDeviceArn": "string",
         "RecorderConfig": {
            "MediaSourceConfig": {
               "MediaUriSecretArn": "string",
               "MediaUriType": "string"
            },
            "ScheduleConfig": {
               "DurationInSeconds": number,
               "ScheduleExpression": "string"
            }
         },
         "UploaderConfig": {
            "ScheduleConfig": {
               "DurationInSeconds": number,
               "ScheduleExpression": "string"
            }
         }
      },
      "FailedStatusDetails": "string",
      "LastUpdatedTime": number,
      "StreamARN": "string",
      "StreamName": "string",
      "SyncStatus": "string"
   }
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

CreationTime

La marca de tiempo en la que se creó por primera vez la configuración de borde de un flujo.

Tipo: marca temporal

EdgeAgentStatus

Objeto que contiene los detalles de estado más recientes de las tareas de grabación y carga de un agente perimetral. Utilice esta información para determinar el estado actual de un agente perimetral.

Tipo: objeto EdgeAgentStatus

EdgeConfig

Una descripción de la configuración perimetral de la transmisión que se utilizará para sincronizarse con el componente Edge Agent IoT Greengrass. El componente Edge Agent se ejecutará en un dispositivo IoT Hub configurado en sus instalaciones.

Tipo: objeto EdgeConfig

FailedStatusDetails

Una descripción del estado de fallo generado.

Tipo: cadena

LastUpdatedTime

La marca temporal en la que se actualizó por última vez la configuración perimetral de una transmisión.

Tipo: marca temporal

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

El nombre del flujo desde el que se actualizó la configuración de borde.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

SyncStatus

El estado más reciente de la actualización de la configuración de borde.

Tipo: cadena

Valores válidos: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

StreamEdgeConfigurationNotFoundException

La excepción se representa cuando Amazon Kinesis Video Stream no encuentra la configuración perimetral de una transmisión que haya especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeImageGenerationConfiguration

Servicio: Amazon Kinesis Video Streams

Obtiene la `ImageGenerationConfiguration` transmisión de vídeo de Kinesis determinada.

Sintaxis de la solicitud

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de vídeo de Kinesis desde la que se va a recuperar la configuración de generación de imágenes. Debe especificar el `StreamName` o el. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la secuencia desde la que se va a recuperar la configuración de generación de imágenes. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "ImageGenerationConfiguration": {
      "DestinationConfig": {
         "DestinationRegion": "string",
         "Uri": "string"
      },
      "Format": "string",
      "FormatConfig": {
         "string" : "string"
      },
      "HeightPixels": number,
      "ImageSelectorType": "string",
      "SamplingInterval": number,
      "Status": "string",
      "WidthPixels": number
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

ImageGenerationConfiguration

Estructura que contiene la información necesaria para la entrega de imágenes de Kinesis vídeo Stream (KVS). Si esta estructura es nula, la configuración se eliminará de la transmisión.

Tipo: objeto ImageGenerationConfiguration

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin

- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeMappedResourceConfiguration

Servicio: Amazon Kinesis Video Streams

Devuelve la información más reciente sobre la transmisión. La `streamName` o `streamARN` debe proporcionarse en la entrada.

Sintaxis de la solicitud

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json

{
   "MaxResults": number,
   "NextToken": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

MaxResults

El número máximo de resultados que se devolverán en la respuesta.

Tipo: entero

Rango válido: valor fijo de 1.

Obligatorio: no

NextToken

El token que debes proporcionar en tu próxima solicitud para obtener otro lote de resultados.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "MappedResourceConfigurationList": [
      {
         "ARN": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

MappedResourceConfigurationList

Estructura que encapsula o contiene las propiedades de configuración del almacenamiento multimedia.

Tipo: matriz de objetos MappedResourceConfigurationListItem

Miembros de la matriz: número mínimo de 0 artículos. Número máximo de 1 elemento.

NextToken

El token que se usó en la `NextToken` solicitud para obtener el siguiente conjunto de resultados.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeMediaStorageConfiguration

Servicio: Amazon Kinesis Video Streams

Devuelve la información más reciente sobre el canal. Especifique la `ChannelName` o `ChannelARN` en la entrada.

Sintaxis de la solicitud

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ChannelName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

ChannelName

El nombre del canal.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "MediaStorageConfiguration": {
      "Status": "string",
      "StreamARN": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

MediaStorageConfiguration

Estructura que encapsula o contiene las propiedades de configuración del almacenamiento multimedia.

Tipo: objeto MediaStorageConfiguration

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeNotificationConfiguration

Servicio: Amazon Kinesis Video Streams

Obtiene la `NotificationConfiguration` transmisión de vídeo de Kinesis determinada.

Sintaxis de la solicitud

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de vídeo de Kinesis desde la que desea recuperar la configuración de notificaciones. Debe especificar el StreamARN `StreamName` o el StreamARN.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión desde la que se va a recuperar la configuración de notificaciones. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "NotificationConfiguration": {
      "DestinationConfig": {
         "Uri": "string"
      },
      "Status": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NotificationConfiguration

La estructura que contiene la información necesaria para las notificaciones. Si la estructura es nula, la configuración se eliminará del flujo.

Tipo: objeto NotificationConfiguration

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeSignalingChannel

Servicio: Amazon Kinesis Video Streams

Devuelve la información más actualizada sobre el canal de señalización. Debe especificar el nombre o el nombre de recurso de Amazon (ARN) del canal que desea describir.

Sintaxis de la solicitud

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ChannelName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El ARN del canal de señalización que desea describir.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

ChannelName

El nombre del canal de señalización que desea describir.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "ChannelInfo": {
      "ChannelARN": "string",
      "ChannelName": "string",
      "ChannelStatus": "string",
      "ChannelType": "string",
      "CreationTime": number,
      "SingleMasterConfiguration": {
         "MessageTtlSeconds": number
      },
      "Version": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

ChannelInfo

Estructura que encapsula los metadatos y las propiedades del canal de señalización especificado.

Tipo: objeto ChannelInfo

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeStream

Servicio: Amazon Kinesis Video Streams

Devuelve la información más reciente sobre la transmisión especificada. Debe especificar el `StreamName` o el`StreamARN`.

Sintaxis de la solicitud

```
POST /describeStream HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "StreamInfo": {
      "CreationTime": number,
      "DataRetentionInHours": number,
      "DeviceName": "string",
      "KmsKeyId": "string",
      "MediaType": "string",
      "Status": "string",
      "StreamARN": "string",
      "StreamName": "string",
      "Version": "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

StreamInfo

Un objeto que describe la transmisión.

Tipo: objeto StreamInfo

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetDataEndpoint

Servicio: Amazon Kinesis Video Streams

Obtiene un punto final para una transmisión específica para lectura o escritura. Utilice este punto final de la aplicación para leer el flujo especificado (mediante las `GetMediaForFragmentList` operaciones `GetMedia` o) o escribir en él (mediante la `PutMedia` operación).

> **Note**
>
> El punto final devuelto no tiene el nombre de la API adjunto. El cliente debe añadir el nombre de la API al punto final devuelto.

En la solicitud, especifique la transmisión mediante `StreamName` o`StreamARN`.

Sintaxis de la solicitud

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
   "APIName": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

APIName

El nombre de la acción de API para la que se va a obtener un punto final.

Tipo: cadena

Valores válidos: `PUT_MEDIA | GET_MEDIA | LIST_FRAGMENTS | GET_MEDIA_FOR_FRAGMENT_LIST | GET_HLS_STREAMING_SESSION_URL | GET_DASH_STREAMING_SESSION_URL | GET_CLIP | GET_IMAGES`

Obligatorio: sí

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión para la que desea obtener el punto final. Debe especificar este parámetro o un `StreamName` en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión para la que desea obtener el punto final. Debe especificar este parámetro o un `StreamARN` en la solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "DataEndpoint": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

DataEndpoint

El valor del punto final. Para leer los datos de la transmisión o escribirlos en ella, especifique este punto final en la aplicación.

Tipo: cadena

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetSignalingChannelEndpoint

Servicio: Amazon Kinesis Video Streams

Proporciona un punto final para que el canal de señalización especificado envíe y reciba mensajes. Esta API usa el parámetro `SingleMasterChannelEndpointConfiguration` de entrada, que consta de las `Role` propiedades `Protocols` y.

`Protocols`se utiliza para determinar el mecanismo de comunicación. Por ejemplo, si lo especificas `WSS` como protocolo, esta API genera un punto final websocket seguro. Si lo especificas `HTTPS` como protocolo, esta API genera un punto final HTTPS. Si lo especificas `WEBRTC` como protocolo, pero el canal de señalización no está configurado para la ingesta, recibirás el error`InvalidArgumentException`.

`Role`determina los permisos de mensajería. Una `MASTER` función hace que esta API genere un punto final que un cliente puede usar para comunicarse con cualquiera de los espectadores del canal. Un `VIEWER` rol hace que esta API genere un punto final que un cliente puede usar para comunicarse solo con un`MASTER`.

Sintaxis de la solicitud

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "SingleMasterChannelEndpointConfiguration": {
      "Protocols": [ "string" ],
      "Role": "string"
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal de señalización para el que desea obtener un punto final.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

SingleMasterChannelEndpointConfiguration

Estructura que contiene la configuración del punto final para el tipo de `SINGLE_MASTER` canal.

Tipo: objeto SingleMasterChannelEndpointConfiguration

Obligatorio: no

Sintaxis de la respuesta

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

{
   "ResourceEndpointList": [
      {
         "Protocol": "string",
         "ResourceEndpoint": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

ResourceEndpointList

Una lista de puntos finales del canal de señalización especificado.

Tipo: matriz de objetos ResourceEndpointListItem

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListEdgeAgentConfigurations

Servicio: Amazon Kinesis Video Streams

Devuelve una matriz de configuraciones perimetrales asociadas al agente perimetral especificado.

En la solicitud, debe especificar el agente perimetral`HubDeviceArn`.

> **Note**
>
> Esta API no está disponible en la región de AWS África (Ciudad del Cabo), af-south-1.

Sintaxis de la solicitud

```
POST /listEdgeAgentConfigurations HTTP/1.1
Content-type: application/json

{
   "HubDeviceArn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

HubDeviceArn

El arn del «Internet de las cosas (IoT)» del agente perimetral.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Obligatorio: sí

MaxResults

El número máximo de configuraciones perimetrales que se devolverán en la respuesta. El valor predeterminado es 5.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 10.

Obligatorio: no

NextToken

Si especifica este parámetro, cuando el resultado de una `ListEdgeAgentConfigurations` operación se trunca, la llamada devuelve el valor de `NextToken` la respuesta. Para obtener otro lote de configuraciones perimetrales, proporciona este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "EdgeConfigs": [
      {
         "CreationTime": number,
         "EdgeConfig": {
            "DeletionConfig": {
               "DeleteAfterUpload": boolean,
               "EdgeRetentionInHours": number,
               "LocalSizeConfig": {
                  "MaxLocalMediaSizeInMB": number,
                  "StrategyOnFullSize": "string"
               }
            },
```

```
            "HubDeviceArn": "string",
            "RecorderConfig": {
               "MediaSourceConfig": {
                  "MediaUriSecretArn": "string",
                  "MediaUriType": "string"
               },
               "ScheduleConfig": {
                  "DurationInSeconds": number,
                  "ScheduleExpression": "string"
               }
            },
            "UploaderConfig": {
               "ScheduleConfig": {
                  "DurationInSeconds": number,
                  "ScheduleExpression": "string"
               }
            }
         },
         "FailedStatusDetails": "string",
         "LastUpdatedTime": number,
         "StreamARN": "string",
         "StreamName": "string",
         "SyncStatus": "string"
      }
   ],
   "NextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

EdgeConfigs

Una descripción de la configuración perimetral de un solo flujo.

Tipo: matriz de objetos ListEdgeAgentConfigurationsEdgeConfig

NextToken

Si la respuesta está truncada, la llamada devuelve este elemento con un token determinado. Para obtener el siguiente lote de configuraciones perimetrales, usa este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListSignalingChannels

Servicio: Amazon Kinesis Video Streams

Devuelve una matriz de `ChannelInfo` objetos. Cada objeto describe un canal de señalización. Para recuperar solo los canales que cumplen una condición específica, puede especificar un`ChannelNameCondition`.

Sintaxis de la solicitud

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
   "ChannelNameCondition": {
      "ComparisonOperator": "string",
      "ComparisonValue": "string"
   },
   "MaxResults": number,
   "NextToken": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelNameCondition

Opcional: devuelve solo los canales que cumplen una condición específica.

Tipo: objeto ChannelNameCondition

Obligatorio: no

MaxResults

El número máximo de canales que se devolverán en la respuesta. El valor predeterminado es 500.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 10000.

Obligatorio: no

NextToken

Si especifica este parámetro, cuando el resultado de una `ListSignalingChannels` operación se trunca, la llamada devuelve el valor de `NextToken` la respuesta. Para obtener otro lote de canales, proporciona este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "ChannelInfoList": [
      {
         "ChannelARN": "string",
         "ChannelName": "string",
         "ChannelStatus": "string",
         "ChannelType": "string",
         "CreationTime": number,
         "SingleMasterConfiguration": {
            "MessageTtlSeconds": number
         },
         "Version": "string"
      }
   ],
   "NextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

ChannelInfoList

Una matriz de objetos `ChannelInfo`.

Tipo: matriz de objetos ChannelInfo

NextToken

Si la respuesta está truncada, la llamada devuelve este elemento con un token. Para obtener el siguiente lote de transmisiones, usa este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tienes los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListStreams

Servicio: Amazon Kinesis Video Streams

Devuelve una matriz de `StreamInfo` objetos. Cada objeto describe un flujo. Para recuperar solo los flujos que cumplan una condición específica, puede especificar un`StreamNameCondition`.

Sintaxis de la solicitud

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
   "MaxResults": number,
   "NextToken": "string",
   "StreamNameCondition": {
      "ComparisonOperator": "string",
      "ComparisonValue": "string"
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

MaxResults

El número máximo de secuencias que se devolverán en la respuesta. El valor predeterminado es 10 000.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 10000.

Obligatorio: no

NextToken

Si especifica este parámetro, cuando el resultado de una `ListStreams` operación se trunca, la llamada devuelve el valor de `NextToken` la respuesta. Para obtener otro lote de transmisiones, proporciona este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

StreamNameCondition

Opcional: devuelve solo las transmisiones que cumplen una condición específica. Actualmente, solo puede especificar el prefijo del nombre de una secuencia como condición.

Tipo: objeto StreamNameCondition

Obligatorio: no

Sintaxis de la respuesta

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

{
   "NextToken": "string",
   "StreamInfoList": [
      {
         "CreationTime": number,
         "DataRetentionInHours": number,
         "DeviceName": "string",
         "KmsKeyId": "string",
         "MediaType": "string",
         "Status": "string",
         "StreamARN": "string",
         "StreamName": "string",
         "Version": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

Si la respuesta está truncada, la llamada devuelve este elemento con un token. Para obtener el siguiente lote de transmisiones, usa este token en tu próxima solicitud.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

StreamInfoList

Una matriz de objetos `StreamInfo`.

Tipo: matriz de objetos StreamInfo

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS

- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListTagsForResource

Servicio: Amazon Kinesis Video Streams

Devuelve una lista de etiquetas asociadas al canal de señalización especificado.

Sintaxis de la solicitud

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
   "NextToken": "string",
   "ResourceARN": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

NextToken

Si especifica este parámetro y el resultado de una `ListTagsForResource` llamada se trunca, la respuesta incluye un token que puede usar en la siguiente solicitud para buscar el siguiente lote de etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

ResourceARN

El nombre de recurso de Amazon (ARN) del canal de señalización para el que desea enumerar las etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

Sintaxis de la respuesta

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

{
   "NextToken": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

Si especifica este parámetro y el resultado de una `ListTagsForResource` llamada se trunca, la respuesta incluye un token que puede usar en la siguiente solicitud para buscar el siguiente conjunto de etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Tags

Un mapa de claves y valores de etiquetas asociados al canal de señalización especificado.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 50 elementos.

Limitaciones de longitud de la clave: longitud mínima de 1. Longitud máxima de 128.

Patrón de clave: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListTagsForStream

Servicio: Amazon Kinesis Video Streams

Devuelve una lista de etiquetas asociadas a la transmisión especificada.

En la solicitud, debe especificar el `StreamName` o el`StreamARN`.

Sintaxis de la solicitud

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json

{
   "NextToken": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

NextToken

Si especificas este parámetro y el resultado de una `ListTagsForStream` llamada se trunca, la respuesta incluye un token que puedes usar en la siguiente solicitud para buscar el siguiente lote de etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión cuyas etiquetas quieres incluir en la lista.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión cuyas etiquetas quieres incluir en la lista.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "NextToken": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

Si especificas este parámetro y el resultado de una `ListTags` llamada se trunca, la respuesta incluye un token que puedes usar en la siguiente solicitud para buscar el siguiente conjunto de etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 1024 caracteres.

Patrón: `[a-zA-Z0-9+/=]*`

Tags

Un mapa de las claves y los valores de las etiquetas asociados al flujo especificado.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 50 elementos.

Limitaciones de longitud de la clave: longitud mínima de 1. Longitud máxima de 128.

Patrón de clave: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidResourceFormatException

El formato de no `StreamARN` es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

StartEdgeConfigurationUpdate

Servicio: Amazon Kinesis Video Streams

Una API asíncrona que actualiza la configuración perimetral existente de una transmisión. El Kinesis Video Stream sincronizará la configuración perimetral de la transmisión con el componente Edge Agent IoT Greengrass que se ejecuta en un dispositivo IoT Hub, configurado en sus instalaciones. El tiempo de sincronización puede variar y depende de la conectividad del dispositivo hub. Se `SyncStatus` actualizará a medida que se reconozca la configuración perimetral y se sincronizará con el agente perimetral.

Si se invoca esta API por primera vez, se creará una nueva configuración perimetral para la transmisión y el estado de la sincronización se establecerá en. `SYNCING` Tendrás que esperar a que el estado de sincronización alcance un estado terminal como:`IN_SYNC`, o`SYNC_FAILED`, antes de volver a usar esta API. Si invocas esta API durante el proceso de sincronización, `ResourceInUseException` aparecerá una. La conectividad de la configuración perimetral de la transmisión y el agente perimetral se volverá a intentar durante 15 minutos. Transcurridos 15 minutos, el estado pasará al `SYNC_FAILED` estado.

Para mover una configuración perimetral de un dispositivo a otro, utilice DeleteEdgeConfiguration para eliminar la configuración perimetral actual. A continuación, puede invocar StartEdgeConfigurationUpdate con un ARN de dispositivo hub actualizado.

> **Note**
>
> Esta API no está disponible en la región de AWS África (Ciudad del Cabo), af-south-1.

Sintaxis de la solicitud

```
POST /startEdgeConfigurationUpdate HTTP/1.1
Content-type: application/json

{
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
         "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
```

```
            }
        },
        "HubDeviceArn": "string",
        "RecorderConfig": {
            "MediaSourceConfig": {
                "MediaUriSecretArn": "string",
                "MediaUriType": "string"
            },
            "ScheduleConfig": {
                "DurationInSeconds": number,
                "ScheduleExpression": "string"
            }
        },
        "UploaderConfig": {
            "ScheduleConfig": {
                "DurationInSeconds": number,
                "ScheduleExpression": "string"
            }
        }
    },
    "StreamARN": "string",
    "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

EdgeConfig

Los detalles de configuración perimetral necesarios para invocar el proceso de actualización.

Tipo: objeto EdgeConfig

Obligatorio: sí

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de . Especifique el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo cuya configuración de borde desea actualizar. Especifique el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "CreationTime": number,
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
         "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
         }
      },
      "HubDeviceArn": "string",
      "RecorderConfig": {
         "MediaSourceConfig": {
            "MediaUriSecretArn": "string",
            "MediaUriType": "string"
```

```
            },
            "ScheduleConfig": {
               "DurationInSeconds": number,
               "ScheduleExpression": "string"
            }
         },
         "UploaderConfig": {
            "ScheduleConfig": {
               "DurationInSeconds": number,
               "ScheduleExpression": "string"
            }
         }
      },
      "FailedStatusDetails": "string",
      "LastUpdatedTime": number,
      "StreamARN": "string",
      "StreamName": "string",
      "SyncStatus": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

CreationTime

La marca de tiempo en la que se creó por primera vez la configuración de borde de un flujo.

Tipo: marca temporal

EdgeConfig

Una descripción de la configuración perimetral de la transmisión que se utilizará para sincronizarse con el componente Edge Agent IoT Greengrass. El componente Edge Agent se ejecutará en un dispositivo IoT Hub configurado en sus instalaciones.

Tipo: objeto EdgeConfig

FailedStatusDetails

Una descripción del estado de fallo generado.

Tipo: cadena

LastUpdatedTime

La marca temporal en la que se actualizó por última vez la configuración perimetral de una transmisión.

Tipo: marca temporal

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

El nombre del flujo desde el que se actualizó la configuración de borde.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

SyncStatus

El estado de sincronización actual de la configuración perimetral de la transmisión. Al invocar esta API, el estado de sincronización se establecerá en ese `SYNCING` estado. Usa la `DescribeEdgeConfiguration` API para obtener el estado más reciente de la configuración perimetral.

Tipo: cadena

Valores válidos: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NoDataRetentionException

La retención de datos de Stream en horas es igual a cero.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

TagResource

Servicio: Amazon Kinesis Video Streams

Añade una o más etiquetas a un canal de señalización. Una etiqueta es un par clave-valor (el valor es opcional) que se puede definir y asignar a los AWS recursos. Si especifica una etiqueta que ya existe, el valor de la etiqueta se reemplaza por el valor que especifique en la solicitud. Para obtener más información, consulte Uso de etiquetas de asignación de costos en la Guía del usuario Administración de facturación y costos de AWS y en la Guía del usuario de administración de costos.

Sintaxis de la solicitud

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
   "ResourceARN": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ResourceARN

El nombre de recurso de Amazon (ARN) del canal de señalización al que desea añadir etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

Tags

Una lista de etiquetas para asociarlas al canal de señalización especificado. Cada etiqueta es un par clave-valor.

Tipo: matriz de objetos Tag

Miembros de la matriz: número mínimo de 1 artículo. Número máximo de 50 artículos.

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

TagsPerResourceExceededLimitException

Ha superado el límite de etiquetas que puede asociar al recurso. Una transmisión de vídeo de Kinesis puede admitir hasta 50 etiquetas.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

TagStream

Servicio: Amazon Kinesis Video Streams

Añade una o más etiquetas a una transmisión. Una etiqueta es un par clave-valor (el valor es opcional) que se puede definir y asignar a los AWS recursos. Si especifica una etiqueta que ya existe, el valor de la etiqueta se reemplaza por el valor que especifique en la solicitud. Para obtener más información, consulte Uso de etiquetas de asignación de costos en la Guía del usuario Administración de facturación y costos de AWS y en la Guía del usuario de administración de costos.

Debe proporcionar el `StreamName` o el`StreamARN`.

Esta operación necesita permiso para la acción `KinesisVideo:TagStream`.

Una transmisión de vídeo de Kinesis puede admitir hasta 50 etiquetas.

Sintaxis de la solicitud

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre del recurso de Amazon (ARN) del recurso al que desea añadir la etiqueta o etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión a la que desea añadir la etiqueta o etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Tags

Una lista de etiquetas para asociar a la transmisión especificada. Cada etiqueta es un par clave-valor (el valor es opcional).

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 50 elementos.

Limitaciones de longitud de la clave: longitud mínima de 1. Longitud máxima de 128.

Patrón de clave: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidResourceFormatException

El formato de no `StreamARN` es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

TagsPerResourceExceededLimitException

Ha superado el límite de etiquetas que puede asociar al recurso. Una transmisión de vídeo de Kinesis puede admitir hasta 50 etiquetas.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UntagResource

Servicio: Amazon Kinesis Video Streams

Elimina una o más etiquetas de un canal de señalización. En la solicitud, especifique solo una o varias claves de etiqueta; no especifique el valor. Si especificas una clave de etiqueta que no existe, se ignorará.

Sintaxis de la solicitud

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
   "ResourceARN": "string",
   "TagKeyList": [ "string" ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ResourceARN

El nombre de recurso de Amazon (ARN) del canal de señalización del que desea eliminar las etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

TagKeyList

Una lista de las claves de las etiquetas que desea eliminar.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 artículo. Número máximo de 50 artículos.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UntagStream

Servicio: Amazon Kinesis Video Streams

Elimina una o más etiquetas de una transmisión. En la solicitud, especifique solo una o varias claves de etiqueta; no especifique el valor. Si especificas una clave de etiqueta que no existe, se ignorará.

En la solicitud, debes proporcionar la tecla `StreamName` o`StreamARN`.

Sintaxis de la solicitud

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string",
   "TagKeyList": [ "string" ]
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de la que quieres eliminar las etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión de la que quieres eliminar las etiquetas.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

TagKeyList

Una lista de las claves de las etiquetas que desea eliminar.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 artículo. Número máximo de 50 artículos.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidResourceFormatException

El formato de no `StreamARN` es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateDataRetention

Servicio: Amazon Kinesis Video Streams

Aumenta o reduce el período de retención de datos de la transmisión en función del valor que especifiques. Para indicar si desea aumentar o disminuir el período de retención de datos, especifique el `Operation` parámetro en el cuerpo de la solicitud. En la solicitud, debe especificar el `StreamName` o el`StreamARN`.

Esta operación necesita permiso para la acción `KinesisVideo:UpdateDataRetention`.

El cambio del período de retención de datos afecta a los datos de la transmisión de la siguiente manera:

- Si se aumenta el período de retención de datos, los datos existentes se conservan durante el nuevo período de retención. Por ejemplo, si el período de retención de datos se incrementa de una hora a siete horas, todos los datos existentes se conservan durante siete horas.
- Si se reduce el período de retención de datos, los datos existentes se conservan durante el nuevo período de retención. Por ejemplo, si el período de retención de datos se reduce de siete horas a una hora, todos los datos existentes se conservan durante una hora y los datos que tengan más de una hora se eliminan inmediatamente.

Sintaxis de la solicitud

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "DataRetentionChangeInHours": number,
   "Operation": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

CurrentVersion

La versión de la transmisión cuyo período de retención quieres cambiar. Para obtener la versión, llama a la API `DescribeStream` o a la `ListStreams` API.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: sí

DataRetentionChangeInHours

El número de horas en las que se va a ajustar la retención actual. El valor que especifique se suma o resta del valor actual, según el. `operation`

El valor mínimo de retención de datos es 0 y el valor máximo es 87600 (diez años).

Tipo: entero

Rango válido: valor mínimo de 1.

Obligatorio: sí

Operation

Indica si desea aumentar o disminuir el período de retención.

Tipo: cadena

Valores válidos: `INCREASE_DATA_RETENTION | DECREASE_DATA_RETENTION`

Obligatorio: sí

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión cuyo período de retención quieres cambiar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión cuyo período de retención quieres cambiar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

VersionMismatchException

La versión de transmisión que especificó no es la versión más reciente. Para obtener la versión más reciente, usa la DescribeStreamAPI.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++

- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateImageGenerationConfiguration

Servicio: Amazon Kinesis Video Streams

Actualiza los `ImageProcessingConfiguration` campos `StreamInfo` y.

Sintaxis de la solicitud

```
POST /updateImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
   "ImageGenerationConfiguration": {
      "DestinationConfig": {
         "DestinationRegion": "string",
         "Uri": "string"
      },
      "Format": "string",
      "FormatConfig": {
         "string" : "string"
      },
      "HeightPixels": number,
      "ImageSelectorType": "string",
      "SamplingInterval": number,
      "Status": "string",
      "WidthPixels": number
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ImageGenerationConfiguration

Estructura que contiene la información necesaria para la entrega de las imágenes KVS. Si la estructura es nula, la configuración se eliminará del flujo.

Tipo: objeto ImageGenerationConfiguration

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de vídeo de Kinesis desde la que desea actualizar la configuración de generación de imágenes. Debe especificar el `StreamName` o el. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo desde el que se va a actualizar la configuración de generación de imágenes. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NoDataRetentionException

La retención de datos de Stream en horas es igual a cero.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateMediaStorageConfiguration

Servicio: Amazon Kinesis Video Streams

`SignalingChannel`Asocia a una transmisión para almacenar el contenido multimedia. Hay dos modos de señalización que puede especificar:

- Si `StorageStatus` está activado, los datos se almacenarán en el lugar `StreamARN` indicado. Para que la ingestión de WebRTC funcione, la transmisión debe tener habilitada la retención de datos.
- Si `StorageStatus` está deshabilitado, no se almacenará ningún dato y el `StreamARN` parámetro no será necesario.

> **Important**
>
> Si `StorageStatus` está activado, ya no se producen conexiones directas peer-to-peer (maestro-espectador). Los pares se conectan directamente a la sesión de almacenamiento. Debe llamar a la `JoinStorageSession` API para activar el envío de una oferta de SDP y establecer una conexión entre un par y la sesión de almacenamiento.

Sintaxis de la solicitud

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "MediaStorageConfiguration": {
      "Status": "string",
      "StreamARN": "string"
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

MediaStorageConfiguration

Estructura que encapsula o contiene las propiedades de configuración del almacenamiento multimedia.

Tipo: objeto MediaStorageConfiguration

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NoDataRetentionException

La retención de datos de Stream en horas es igual a cero.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateNotificationConfiguration

Servicio: Amazon Kinesis Video Streams

Actualiza la información de notificación de una transmisión.

Sintaxis de la solicitud

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json

{
   "NotificationConfiguration": {
      "DestinationConfig": {
         "Uri": "string"
      },
      "Status": "string"
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

NotificationConfiguration

La estructura que contiene la información necesaria para las notificaciones. Si la estructura es nula, la configuración se eliminará del flujo.

Tipo: objeto NotificationConfiguration

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de vídeo de Kinesis desde la que desea actualizar la configuración de notificaciones. Debe especificar el `StreamName` o el. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión desde la que se va a actualizar la configuración de notificaciones. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NoDataRetentionException

La retención de datos de Stream en horas es igual a cero.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué se ha asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o la `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateSignalingChannel

Servicio: Amazon Kinesis Video Streams

Actualiza el canal de señalización existente. Se trata de una operación asíncrona y tarda un tiempo en completarse.

Si el `MessageTtlSeconds` valor se actualiza (aumenta o reduce), solo se aplica a los mensajes nuevos que se envíen a través de este canal una vez que se haya actualizado. Los mensajes existentes siguen caducados según el `MessageTtlSeconds` valor anterior.

Sintaxis de la solicitud

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "CurrentVersion": "string",
   "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
   }
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal de señalización que desea actualizar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

CurrentVersion

La versión actual del canal de señalización que desea actualizar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: sí

SingleMasterConfiguration

La estructura que contiene la configuración del `SINGLE_MASTER` tipo de canal de señalización que desea actualizar. Este parámetro y los mensajes del canal time-to-live son necesarios para los canales con ese tipo de `SINGLE_MASTER` canal.

Tipo: objeto SingleMasterConfiguration

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 401

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué se ha asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o la `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

VersionMismatchException

La versión de transmisión que especificó no es la versión más reciente. Para obtener la versión más reciente, usa la DescribeStreamAPI.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

UpdateStream

Servicio: Amazon Kinesis Video Streams

Actualiza los metadatos de la transmisión, como el nombre del dispositivo y el tipo de medio.

Debe proporcionar el nombre de la transmisión o el nombre de recurso de Amazon (ARN) de la transmisión.

Para asegurarse de que dispone de la versión más reciente de la transmisión antes de actualizarla, puede especificar la versión de la transmisión. Kinesis Video Streams asigna una versión a cada transmisión. Al actualizar una transmisión, Kinesis Video Streams asigna un nuevo número de versión. Para obtener la versión más reciente de la transmisión, utilice la `DescribeStream` API.

`UpdateStream`es una operación asíncrona y tarda en completarse.

Sintaxis de la solicitud

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "DeviceName": "string",
   "MediaType": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

CurrentVersion

La versión de la transmisión cuyos metadatos desea actualizar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: sí

DeviceName

El nombre del dispositivo que está grabando en la transmisión.

> **Note**
>
> En la implementación actual, Kinesis Video Streams no usa este nombre.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

MediaType

El tipo de contenido multimedia de la transmisión. Se utiliza `MediaType` para especificar el tipo de contenido que contiene la transmisión para los consumidores de la transmisión. Para obtener más información sobre los tipos de medios, consulte Tipos de medios. Si decide especificarlos`MediaType`, consulte Requisitos de nomenclatura.

Para reproducir vídeo en la consola, debe especificar el tipo de vídeo correcto. Por ejemplo, si el vídeo de la transmisión es H.264, especifíquelo `video/h264` como. `MediaType`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Obligatorio: no

StreamARN

El ARN de la transmisión cuyos metadatos desea actualizar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión cuyos metadatos desea actualizar.

El nombre de la transmisión es un identificador de la transmisión y debe ser único para cada cuenta y región.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceInUseException

Si la entrada `StreamARN` o `ChannelARN` entrada ya `CLOUD_STORAGE_MODE` está mapeada a otro recurso de Kinesis Video Stream, o si la `StreamARN` entrada `ChannelARN` proporcionada no está en estado Activo, pruebe una de las siguientes opciones:

1. La `DescribeMediaStorageConfiguration` API para determinar a qué está asignado el canal de transmisión en cuestión.
2. La `DescribeMappedResourceConfiguration` API para determinar el canal al que está mapeada la transmisión determinada.
3. La `DescribeStream` o `DescribeSignalingChannel` API para determinar el estado del recurso.

Código de estado HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams no encuentra la transmisión que ha especificado.

Código de estado HTTP: 404

VersionMismatchException

La versión de transmisión que especificó no es la versión más reciente. Para obtener la versión más reciente, usa la DescribeStreamAPI.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Contenido multimedia de Amazon Kinesis Video Streams

Amazon Kinesis Video Streams Media admite las siguientes acciones:

- GetMedia
- PutMedia

GetMedia

Servicio: Amazon Kinesis Video Streams Media

Utilice esta API para recuperar contenido multimedia de una transmisión de vídeo de Kinesis. En la solicitud, debe identificar el nombre de la transmisión o el nombre de Amazon Resource Name (ARN) de la transmisión y el fragmento inicial. A continuación, Kinesis Video Streams devuelve una secuencia de fragmentos ordenados por número de fragmento.

> **Note**
>
> Primero debe llamar a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `GetMedia` solicitudes a este punto final mediante el parámetro `--endpoint-url`.

Cuando coloca datos multimedia (fragmentos) en una transmisión, Kinesis Video Streams almacena cada fragmento entrante y los metadatos relacionados en lo que se denomina un «fragmento». Para obtener más información, consulte PutMedia. La `GetMedia` API devuelve una secuencia de estos fragmentos empezando por el fragmento que especifique en la solicitud.

La `GetMedia` API está diseñada para funcionar como una API de streaming en una conexión de larga duración. No está diseñada para su uso de la RESTful manera tradicional, en la que se establece y cierra una nueva conexión HTTP para cada fragmento. Al invocar la `GetMedia` API, Kinesis Video Streams entrega fragmentos de forma continua a través de una conexión persistente de larga duración mediante la codificación de transferencia fragmentada HTTP.

Al utilizar la API, se aplican los siguientes límites: `GetMedia`

- Un cliente puede llamar `GetMedia` hasta cinco veces por segundo por transmisión.
- Kinesis Video Streams envía datos multimedia a una velocidad de hasta 25 megabytes por segundo (o 200 megabits por segundo) durante una sesión. `GetMedia`

> **Note**
>
> `GetMedia`Utilícela como una conexión de streaming de larga duración para recuperar varios fragmentos en una única conexión persistente. No utilices la `GetMedia` API de RESTful forma tradicional, en la que estableces y cierras una nueva conexión HTTP para cada fragmento. Si intenta realizar más de tres `GetMedia` conexiones

simultáneas, Kinesis Video Streams limita las últimas conexiones con un error. `ConnectionLimitExceededException`

> **Note**
>
> El código de estado de la respuesta `GetMedia` HTTP se devolverá inmediatamente, pero la lectura de la carga útil de la respuesta HTTP se agotará transcurridos 3 segundos si no hay fragmentos ingeridos disponibles para su reproducción.

> **Note**
>
> Si se produce un error después de invocar una API multimedia de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:
>
> - `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
> - `x-amz-RequestId`Encabezado HTTP: si quieres informar de un problema AWS, el equipo de soporte puede diagnosticar mejor el problema si se le proporciona el identificador de la solicitud.
>
> Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente. Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
```

```
    "StartSelector": {
       "AfterFragmentNumber": "string",
       "ContinuationToken": "string",
       "StartSelectorType": "string",
       "StartTimestamp": number
    },
    "StreamARN": "string",
    "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

StartSelector

Identifica el fragmento inicial que se debe obtener de la transmisión especificada.

Tipo: objeto StartSelector

Obligatorio: sí

StreamARN

El ARN de la transmisión desde la que desea obtener el contenido multimedia. Si no especifica el`streamARN`, debe especificar el`streamName`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión de vídeo de Kinesis desde donde desea obtener el contenido multimedia. Si no especifica el`streamName`, debe especificar el`streamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

La respuesta devuelve los siguientes encabezados HTTP.

ContentType

El tipo de contenido del medio solicitado.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[a-zA-Z0-9_\.\-]+$`

La respuesta devuelve lo siguiente como el cuerpo HTTP.

Payload

La carga útil que devuelve Kinesis Video Streams es una secuencia de fragmentos de la transmisión especificada. Para obtener más información sobre los fragmentos, consulte. PutMedia Los fragmentos que Kinesis Video Streams devuelve en `GetMedia` la convocatoria también incluyen las siguientes etiquetas Matroska (MKV) adicionales:

- AWS_KINESISVIDEO_CONTINUATION_TOKEN (cadena UTF-8): en caso de que su `GetMedia` llamada finalice, puede usar este token de continuación en su próxima solicitud para obtener el siguiente fragmento en el que terminó la última solicitud.

- AWS_KINESISVIDEO_MILLIS_BEHIND_NOW (cadena UTF-8): las aplicaciones cliente pueden usar este valor de etiqueta para determinar qué distancia hay entre el fragmento devuelto en la respuesta y el último fragmento de la transmisión.
- AWS_KINESISVIDEO_FRAGMENT_NUMBER - Número de fragmento devuelto en el fragmento.
- AWS_KINESISVIDEO_SERVER_TIMESTAMP - Marca horaria del fragmento en el servidor.
- AWS_KINESISVIDEO_PRODUCER_TIMESTAMP - Marca de tiempo del productor del fragmento.

Si se produce un error, aparecerán las siguientes etiquetas:

- AWS_KINESISVIDEO_ERROR_CODE - Descripción de cadena de un error que provocó que GetMedia se detuviera.
- AWS_KINESISVIDEO_ERROR_ID: código entero del error.

Los códigos de error son los siguientes:

- 3002: error al escribir en la transmisión
- 4000: No se encuentra el fragmento solicitado
- 4500: Acceso denegado a la clave KMS de la transmisión
- 4501: La clave KMS de Stream está deshabilitada
- 4502: Error de validación en la clave KMS de la transmisión
- 4503: la clave KMS especificada en la transmisión no está disponible
- 4504: uso no válido de la clave KMS especificada en la transmisión
- 4505: estado no válido de la clave KMS especificada en la transmisión
- 4506: no se encuentra la clave KMS especificada en la transmisión
- 5000: error interno

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

ConnectionLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de conexiones de cliente permitidas.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidEndpointException

La persona que llamó utilizó un punto final incorrecto para escribir datos en una transmisión. Al recibir una excepción de este tipo, el usuario debe llamar `GetDataEndpoint` con el `APIName` valor establecido en `PUT_MEDIA` y utilizar el punto final de respuesta para invocar la siguiente `PutMedia` llamada.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar ninguna operación en la transmisión determinada o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

Código de estado: 404. La transmisión con el nombre indicado no existe.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET

- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

PutMedia

Servicio: Amazon Kinesis Video Streams Media

Utilice esta API para enviar datos multimedia a una transmisión de vídeo de Kinesis.

> **Note**
>
> Primero debe llamar a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `PutMedia` solicitudes a este punto final mediante el parámetro --endpoint-url.

En la solicitud, usa los encabezados HTTP para proporcionar información sobre los parámetros, por ejemplo, el nombre de la transmisión, la marca de tiempo y si el valor de la marca de tiempo es absoluto o relativo al momento en que el productor comenzó a grabar. Utiliza el cuerpo de la solicitud para enviar los datos multimedia. Kinesis Video Streams solo admite el formato contenedor Matroska (MKV) para enviar datos multimedia mediante esta API.

Dispone de las siguientes opciones para enviar datos mediante esta API:

- Envíe datos multimedia en tiempo real: por ejemplo, una cámara de seguridad puede enviar fotogramas en tiempo real a medida que los genera. Este enfoque minimiza la latencia entre la grabación de vídeo y los datos enviados por cable. Esto se conoce como productor continuo. En este caso, una aplicación de consumo puede leer la transmisión en tiempo real o cuando sea necesario.
- Envía datos multimedia sin conexión (en lotes): por ejemplo, una cámara corporal puede grabar vídeo durante horas y almacenarlo en el dispositivo. Más adelante, cuando conecte la cámara al puerto de acoplamiento, la cámara podrá iniciar una `PutMedia` sesión para enviar datos a una transmisión de vídeo de Kinesis. En este escenario, la latencia no es un problema.

Cuando utilices esta API, ten en cuenta las siguientes consideraciones:

- Debe especificar `streamName` o `streamARN`, pero no ambos.
- Para poder reproducir el contenido multimedia en la consola o mediante HLS, la pista 1 de cada fragmento debe contener vídeo codificado en H.264, el codecID de los metadatos del fragmento debe ser «V_MPEG/ISO/AVC» y los metadatos del fragmento deben incluir datos privados del códec h.264 con formato AVCC. Opcionalmente, la pista 2 de cada fragmento debe contener audio

codificado en AAC, el codeCid de los metadatos del fragmento debe ser «A_AAC» y los metadatos del fragmento deben incluir datos privados del códec AAC.

- La `PutMedia` API está diseñada para funcionar como una API de streaming a través de una conexión de larga duración. No está diseñada para su uso de la RESTful manera tradicional, en la que se establece y cierra una nueva conexión HTTP para cada fragmento. Cuando utilices la `PutMedia` API, utiliza la codificación de transferencia fragmentada HTTP para enviar fragmentos de forma continua a través de una conexión persistente.
- Por cada fragmento recibido en una `PutMedia` sesión, Kinesis Video Streams envía uno o más acuses de recibo. Las posibles consideraciones de red del lado del cliente pueden provocar que no reciba todos estos acuses de recibo a medida que se generan.

> **Note**
>
> `PutMedia`Utilízala como una conexión de streaming de larga duración para enviar varios fragmentos en una única conexión persistente. Si intenta realizar más de una `PutMedia` conexión simultánea, Kinesis Video Streams limita las últimas conexiones con un error. `ConnectionLimitExceededException`

Al utilizar la API, se aplican los siguientes límites: `PutMedia`

- Un cliente puede llamar `PutMedia` hasta cinco veces por segundo por transmisión.
- Un cliente puede enviar hasta cinco fragmentos por segundo por transmisión.
- Kinesis Video Streams lee datos multimedia a una velocidad de hasta 12,5 MB/segundo o 100 Mbps durante una sesión. `PutMedia`

Tenga en cuenta las siguientes restricciones. En estos casos, Kinesis Video Streams envía la confirmación de error en la respuesta.

- No se permiten los fragmentos con códigos de tiempo que superen el límite máximo permitido y que contengan más de 50 MB de datos.
- No se permiten los fragmentos que contengan más de tres pistas. Cada fotograma de cada fragmento debe tener el mismo número de pista que una de las pistas definidas en el encabezado del fragmento. Además, cada fragmento debe contener al menos un fotograma para cada pista definida en el encabezado del fragmento.

- Cada fragmento debe contener al menos un fotograma para cada pista definida en los metadatos del fragmento.
- La marca de tiempo del primer fotograma de un fragmento debe ser posterior a la última del fragmento anterior.
- Si una secuencia MKV contiene más de un segmento MKV o contiene elementos MKV no permitidos (por ejemplo`track*`), también se reconoce el error.

Kinesis Video Streams almacena cada fragmento entrante y los metadatos relacionados en lo que se denomina un «fragmento». Los metadatos del fragmento incluyen lo siguiente:

- Los encabezados MKV proporcionados al inicio de la solicitud `PutMedia`
- Los siguientes metadatos específicos de Kinesis Video Streams para el fragmento:
  - `server_timestamp`- Marca de tiempo en que Kinesis Video Streams empezó a recibir el fragmento.
  - `producer_timestamp`- Marca de tiempo, cuando el productor empezó a grabar el fragmento. Kinesis Video Streams utiliza tres datos recibidos en la solicitud para calcular este valor.
    - El valor del código de tiempo del fragmento recibido en el cuerpo de la solicitud junto con el fragmento.
    - Dos encabezados de solicitud: `producerStartTimestamp` (cuando el productor comenzó a grabar) y `fragmentTimeCodeType` (si el código de tiempo del fragmento de la carga útil es absoluto o relativo).

    A continuación, Kinesis Video Streams calcula `producer_timestamp` el fragmento de la siguiente manera:

    Si `fragmentTimeCodeType` es relativo, entonces

    `producer_timestamp`= `producerStartTimeStamp` + código de tiempo del fragmento

    Si `fragmentTimeCodeType` es absoluto, entonces

    `producer_timestamp`= código de tiempo del fragmento (convertido a milisegundos)
  - Número de fragmento único asignado por Kinesis Video Streams.

> **Note**
>
> Al realizar la `GetMedia` solicitud, Kinesis Video Streams devuelve una transmisión de estos fragmentos. El cliente puede procesar los metadatos según sea necesario.

> **Note**
>
> Esta operación solo está disponible para el AWS SDK para Java. No se admite en AWS SDKs otros idiomas.

> **Note**
>
> Kinesis Video Streams no analiza ni valida los datos privados del códec durante la ingesta y el archivado mediante la API. PutMedia KVS extrae y valida la información necesaria de los datos privados del códec para MPEG-TS y MP4 el empaquetado de fragmentos al consumir la transmisión a través del HLS. APIs

> **Note**
>
> Si se produce un error después de invocar una API multimedia de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:
>
> - `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
> - `x-amz-RequestId`Encabezado HTTP: si quieres informar de un problema AWS, el equipo de soporte puede diagnosticar mejor el problema si se le proporciona el identificador de la solicitud.
>
> Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente.

Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /putMedia HTTP/1.1
x-amzn-stream-name: StreamName
x-amzn-stream-arn: StreamARN
x-amzn-fragment-timecode-type: FragmentTimecodeType
x-amzn-producer-start-timestamp: ProducerStartTimestamp

Payload
```

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

FragmentTimecodeType

Este valor se transfiere como el encabezado HTTP `x-amzn-fragment-timecode-type`.

Indica si los códigos de tiempo de los fragmentos (carga útil, cuerpo de la solicitud HTTP) son absolutos o relativos a. `producerStartTimestamp` Kinesis Video Streams utiliza esta información para calcular `producer_timestamp` el fragmento recibido en la solicitud, tal como se describe en la descripción general de la API.

Valores válidos: `ABSOLUTE | RELATIVE`

Obligatorio: sí

ProducerStartTimestamp

Este valor se transfiere como el encabezado HTTP `x-amzn-producer-start-timestamp`.

Esta es la marca de tiempo del productor en la que el productor comenzó a grabar el contenido multimedia (no la marca de tiempo de los fragmentos específicos de la solicitud).

StreamARN

Este valor se transfiere como el encabezado HTTP `x-amzn-stream-arn`.

Nombre del recurso de Amazon (ARN) de la transmisión de vídeo de Kinesis en la que desea escribir el contenido multimedia. Si no especifica el`streamARN`, debe especificar el. `streamName`

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

Este valor se transfiere como el encabezado HTTP `x-amzn-stream-name`.

Nombre de la transmisión de vídeo de Kinesis en la que desea escribir el contenido multimedia. Si no especifica el`streamName`, debe especificar el`streamARN`.

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Cuerpo de la solicitud

La solicitud acepta los siguientes datos binarios.

Payload

El contenido multimedia que se grabará en la transmisión de vídeo de Kinesis. En la implementación actual, Kinesis Video Streams solo admite el formato contenedor Matroska (MKV) con un único segmento MKV. Un segmento puede contener uno o más clústeres.

> **Note**
>
> Cada clúster MKV se asigna a un fragmento de transmisión de vídeo de Kinesis. La duración del clúster que elija se convierte en la duración del fragmento.

Sintaxis de la respuesta

```
HTTP/1.1 200

Payload
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

La respuesta devuelve lo siguiente como el cuerpo HTTP.

Payload

Cuando Kinesis Video Streams recibe correctamente `PutMedia` una solicitud, el servicio valida los encabezados de la solicitud. A continuación, el servicio comienza a leer la carga útil y envía primero una respuesta HTTP 200.

A continuación, el servicio devuelve una secuencia que contiene una serie de objetos JSON (`Acknowledgement`objetos) separados por líneas nuevas. Los acuses de recibo se reciben en la misma conexión a través de la cual se envían los datos multimedia. Puede haber muchos acuses de recibo para una `PutMedia` solicitud. Cada uno `Acknowledgement` consta de los siguientes pares clave-valor:

- `AckEventType`- Tipo de evento que representa el acuse de recibo.
  - Almacenamiento en búfer: Kinesis Video Streams ha empezado a recibir el fragmento. Kinesis Video Streams envía el primer acuse de recibo de almacenamiento en búfer cuando se recibe el primer byte de datos fragmentados.
  - Recibido: Kinesis Video Streams recibió el fragmento completo. Si no configuró la transmisión para conservar los datos, el productor puede dejar de almacenar el fragmento en búfer al recibir este acuse de recibo.
  - Persistido: Kinesis Video Streams ha conservado el fragmento (por ejemplo, en Amazon S3). Obtendrá este reconocimiento si configuró la transmisión para conservar los datos. Tras recibir este acuse de recibo, el productor puede dejar de almacenar el fragmento en búfer.
  - Error: Kinesis Video Streams detectó un error al procesar el fragmento. Puede revisar el código de error y determinar el siguiente curso de acción.
  - Inactiva: la `PutMedia` sesión está en curso. Sin embargo, Kinesis Video Streams no recibe datos actualmente. Kinesis Video Streams envía este acuse de recibo periódicamente durante un máximo de 30 segundos después de la última recepción de los datos. Si no se recibe ningún dato en los 30 segundos, Kinesis Video Streams cierra la solicitud.

    > **Note**
    >
    > Este reconocimiento puede ayudar al productor a determinar si la `PutMedia` conexión está activa, incluso si no envía ningún dato.

- `FragmentTimecode`- Fragmenta el código de tiempo para el que se envía el acuse de recibo.

Es posible que falte el elemento si **`AckEventType`** está inactivo.

- `FragmentNumber`- Número de fragmento generado por Kinesis Video Streams para el que se envía el acuse de recibo.
- `ErrorIdy ErrorCode` - Si `AckEventType` es así`Error`, este campo proporciona el código de error correspondiente. A continuación se muestra la lista de errores IDs y sus códigos de error y mensajes de error correspondientes:
  - 4000 - STREAM_READ_ERROR: error al leer el flujo de datos.
  - 4001 - MAX_FRAGMENT_SIZE_REACHED: el tamaño del fragmento supera el límite máximo permitido (50 MB).
  - 4002 - MAX_FRAGMENT_DURATION_REACHED - La duración del fragmento supera el límite máximo permitido.
  - 4003 - MAX_CONNECTION_DURATION_REACHED: la duración de la conexión supera el umbral máximo permitido.
  - 4004 - FRAGMENT_TIMECODE_LESSER_THAN_PREVIOUS - El código de tiempo del fragmento es inferior al código de tiempo anterior (durante una llamada, no se pueden enviar fragmentos desordenados). `PutMedia`
  - 4005 - MORE_THAN_ALLOWED_TRACKS_FOUND - Se encuentra más de una pista en MKV. (obsoleto)
  - 4006 - INVALID_MKV_DATA - No se pudo analizar el flujo de entrada como un formato MKV válido.
  - 4007 - INVALID_PRODUCER_TIMESTAMP: marca de tiempo del productor no válida.
  - 4008 - STREAM_NOT_ACTIVE: la transmisión ya no existe (eliminada).
  - 4009 - FRAGMENT_METADATA_LIMIT_REACHED - Se ha alcanzado el límite de metadatos de fragmentos. Consulta la sección Límites de la guía para desarrolladores.
  - 4010 - TRACK_NUMBER_MISMATCH - El número de pista de un fotograma MKV no coincide con el de las pistas del encabezado MKV.
  - 4011 - FRAMES_MISSING_FOR_TRACK - El fragmento no contenía fotogramas para al menos una de las pistas del encabezado MKV.
  - 4012 - INVALID_FRAGMENT_METADATA - El nombre de los metadatos del fragmento no puede empezar por la cadena. `AWS_`
  - 4500 - KMS_KEY_ACCESS_DENIED: se deniega el acceso a la clave KMS especificada en la transmisión.

- 4501 - KMS_KEY_DISABLED: la clave KMS especificada en la transmisión está deshabilitada.
- 4502 - KMS_KEY_VALIDATION_ERROR: la clave KMS especificada en la transmisión no se pudo validar.
- 4503 - KMS_KEY_UNAVAILABLE: la clave KMS especificada en la transmisión no está disponible.
- 4504 - KMS_KEY_INVALID_USAGE: uso no válido de la clave KMS especificada en la transmisión.
- 4505 - KMS_KEY_INVALID_STATE - La clave KMS especificada en la transmisión está en un estado no válido.
- 4506 - KMS_KEY_NOT_FOUND - No se encuentra la clave KMS especificada en la transmisión.
- 5000 - INTERNAL_ERROR: error de servicio interno.
- 5001 - ARCHIVAL_ERROR - Kinesis Video Streams no pudo conservar los fragmentos en el almacén de datos.

> **Note**
>
> El productor, al enviar la carga útil para una `PutMedia` solicitud de larga duración, debería leer la respuesta para ver si hay acuse de recibo. Es posible que un productor reciba varios reconocimientos al mismo tiempo, debido al almacenamiento en búfer de un servidor proxy intermedio. Un productor que desee recibir los acuses de recibo a tiempo puede enviar menos fragmentos en cada solicitud. `PutMedia`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

ConnectionLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de conexiones de cliente permitidas.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidEndpointException

La persona que llamó utilizó un punto final incorrecto para escribir datos en una transmisión. Al recibir una excepción de este tipo, el usuario debe llamar `GetDataEndpoint` con el `APIName` valor establecido en `PUT_MEDIA` y utilizar el punto final de respuesta para invocar la siguiente `PutMedia` llamada.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar ninguna operación en la transmisión determinada o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

Código de estado: 404. La transmisión con el nombre indicado no existe.

Código de estado HTTP: 404

Ejemplos

Formato de reconocimiento

El formato del reconocimiento es el siguiente:

```
{
       Acknowledgement : {
          "EventType": enum
```

```
            "FragmentTimecode": Long,
            "FragmentNumber": Long,
            "ErrorId" : String
        }
}
```

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Contenido multimedia archivado de Amazon Kinesis Video Streams

Los medios archivados de Amazon Kinesis Video Streams admiten las siguientes acciones:

- GetClip
- GetDASHStreamingSessionURL
- GetHLSStreamingSessionURL
- GetImages
- GetMediaForFragmentList
- ListFragments

GetClip

Servicio: Amazon Kinesis Video Streams Archived Media

Descarga un MP4 archivo (clip) que contiene el contenido multimedia archivado y bajo demanda de la transmisión de vídeo especificada durante el intervalo de tiempo especificado.

StreamName Tanto el parámetro StreamArn como el de StreamArn son opcionales, pero debes especificar el StreamArn StreamName o el StreamArn al invocar esta operación de API.

> **Note**
>
> Primero debes llamar a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `GetClip` solicitudes a este punto final mediante el parámetro --endpoint-url.

Una transmisión de vídeo de Amazon Kinesis tiene los siguientes requisitos para proporcionar datos a través de: MP4

- Requisitos de pista de reproducción de vídeo.
- La retención de datos debe ser mayor o igual que 0.
- La pista de vídeo de cada fragmento debe contener los datos privados del códec en la codificación de vídeo avanzada (AVC) para el formato H.264 y en HEVC para el formato H.265. Para obtener más información, consulte la especificación especificación ISO/IEC 14496-15 para MPEG. Si desea obtener más información sobre cómo adaptar datos de transmisiones a un determinado formato, consulte Referencia de marcas de adaptación de capa de abstracción de red (NAL).
- La pista de audio (si la hay) de cada fragmento debe contener los datos privados del códec en formato AAC (especificación ISO/IEC 13818-7 para AAC) o en formato MS Wave.

Puedes monitorizar la cantidad de datos salientes supervisando la CloudWatch métrica de `GetClip.OutgoingBytes` Amazon. Para obtener información sobre cómo CloudWatch monitorizar Kinesis Video Streams, consulte Supervisión de Kinesis Video Streams. Para obtener información sobre precios, consulte Precios AWS y precios de Amazon Kinesis Video Streams. Se aplican cargos por AWS los datos salientes.

> **Important**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. No se admiten cambios de CPD entre los fragmentos de destino del clip resultante. El CPD debe ser coherente en todos los medios consultados; de lo contrario, se devolverá un error.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. Se devuelve un error si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si se cambia una pista de audio AAC por una pista de audio A-Law.

Sintaxis de la solicitud

```
POST /getClip HTTP/1.1
Content-type: application/json

{
   "ClipFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ClipFragmentSelector

El intervalo de tiempo del clip solicitado y el origen de las marcas de tiempo.

Tipo: objeto ClipFragmentSelector

Obligatorio: sí

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de la que se va a recuperar el clip multimedia.

Debe especificar el StreamARN StreamName o el StreamARN.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión de la que se va a recuperar el clip multimedia.

Debe especificar el StreamArn StreamName o el StreamArn.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-Type: ContentType
```

```
Payload
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

La respuesta devuelve los siguientes encabezados HTTP.

ContentType

El tipo de contenido del contenido multimedia del clip solicitado.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[a-zA-Z0-9_\.\-]+$`

La respuesta devuelve lo siguiente como el cuerpo HTTP.

Payload

MP4 Archivo tradicional que contiene el clip multimedia de la secuencia de vídeo especificada. La salida contendrá los primeros 100 MB o los primeros 200 fragmentos de la marca de tiempo de inicio especificada. Para obtener más información, consulte Cuotas de Kinesis Video Streams.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

InvalidCodecPrivateDataException

Los datos privados del códec de al menos una de las pistas de la transmisión de vídeo no son válidos para esta operación.

Código de estado HTTP: 400

InvalidMediaFrameException

No se pudieron analizar uno o más fotogramas del clip solicitado en función del códec especificado.

Código de estado HTTP: 400

MissingCodecPrivateDataException

No se encontró ningún dato privado del códec en al menos una de las pistas de la transmisión de vídeo.

Código de estado HTTP: 400

NoDataRetentionException

`GetImages`se solicitó para una transmisión que no retiene datos (es decir, tiene un valor `DataRetentionInHours` de 0).

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

UnsupportedStreamMediaTypeException

El tipo de contenido multimedia (por ejemplo, vídeo h.264 o h.265 o audio AAC o G.711) no se pudo determinar a partir del códec de las pistas del primer fragmento IDs de una sesión de reproducción. El ID de códec de la pista 1 debería ser `V_MPEG/ISO/AVC` y, opcionalmente, el ID de códec de la pista 2 debería ser. `A_AAC`

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetDASHStreamingSessionURL

Servicio: Amazon Kinesis Video Streams Archived Media

Recupera una URL de transmisión adaptativa dinámica MPEG a través de HTTP (DASH) para la transmisión. A continuación, puede abrir la URL en un reproductor multimedia para ver el contenido de la transmisión.

`StreamName`Tanto el parámetro como el `StreamARN` parámetro son opcionales, pero debes especificar uno `StreamName` o el `StreamARN` al invocar esta operación de API.

Una transmisión de vídeo de Amazon Kinesis tiene los siguientes requisitos para proporcionar datos a través de MPEG-DASH:

- Requisitos de pista de reproducción de vídeo.
- La retención de datos debe ser mayor o igual que 0.
- La pista de vídeo de cada fragmento debe contener los datos privados del códec en la codificación de vídeo avanzada (AVC) para el formato H.264 y en HEVC para el formato H.265. Para obtener más información, consulte la especificación especificación ISO/IEC 14496-15 para MPEG. Si desea obtener más información sobre cómo adaptar datos de transmisiones a un determinado formato, consulte Referencia de marcas de adaptación de capa de abstracción de red (NAL).
- La pista de audio (si la hay) de cada fragmento debe contener los datos privados del códec en formato AAC (especificación ISO/IEC 13818-7 para AAC) o en formato MS Wave.

El siguiente procedimiento muestra cómo utilizar MPEG-DASH con Kinesis Video Streams:

1. Llame a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `GetDASHStreamingSessionURL` solicitudes a este punto final mediante el parámetro --endpoint-url.
2. Recupera la URL de MPEG-DASH usando. `GetDASHStreamingSessionURL` Kinesis Video Streams crea una sesión de streaming MPEG-DASH que se utiliza para acceder al contenido de una transmisión mediante el protocolo MPEG-DASH. `GetDASHStreamingSessionURL`devuelve una URL autenticada (que incluye un token de sesión cifrado) para el manifiesto MPEG-DASH de la sesión (el recurso raíz necesario para la transmisión con MPEG-DASH).

> **Note**
>
> No compartas ni guardes este token donde una entidad no autorizada pueda acceder a él. El token proporciona acceso al contenido de la transmisión. Proteja el token con las mismas medidas que utiliza con sus AWS credenciales.

El contenido multimedia que está disponible a través del manifiesto consiste únicamente en la transmisión, el intervalo de tiempo y el formato solicitados. No hay otros datos multimedia disponibles (como fotogramas fuera de la ventana solicitada o velocidades de bits alternativas).

3. Proporcione la URL (que contiene el token de sesión cifrado) del manifiesto MPEG-DASH a un reproductor multimedia compatible con el protocolo MPEG-DASH. Kinesis Video Streams hace que el fragmento de inicialización y los fragmentos multimedia estén disponibles a través de la URL del manifiesto. El fragmento de inicialización contiene los datos privados del códec de la transmisión y otros datos necesarios para configurar el decodificador y el renderizador de vídeo o audio. Los fragmentos multimedia contienen fotogramas de vídeo codificados o muestras de audio codificadas.
4. El reproductor multimedia recibe la URL autenticada y solicita los metadatos y los datos multimedia de la transmisión con normalidad. Cuando el reproductor multimedia solicita datos, ejecuta las siguientes acciones:
   - ObtenerDASHManifest: recupera un manifiesto MPEG DASH, que contiene los metadatos del contenido multimedia que desea reproducir.
   - Obtener MP4InitFragment: recupera el fragmento de inicialización. MP4 El reproductor multimedia suele cargar el fragmento de inicialización antes de cargar cualquier fragmento multimedia. Este fragmento contiene los átomos `fytp` «» y `moov` «» y los MP4 átomos secundarios necesarios para inicializar el decodificador del reproductor multimedia.

     El fragmento de inicialización no corresponde a un fragmento de una transmisión de vídeo de Kinesis. Contiene solo los datos privados del códec para la transmisión y la pista correspondiente, que el reproductor multimedia necesita para decodificar los fotogramas multimedia.
   - Obtener MP4MediaFragment: recupera fragmentos multimedia MP4. Estos fragmentos contienen los átomos «`moof`» y «`mdat`» y sus MP4 átomos secundarios, que contienen los marcos multimedia del fragmento codificado y sus marcas temporales.

> **Important**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. Los cambios de CPD no se admiten durante una sesión de streaming. El CPD debe permanecer coherente en todos los medios consultados.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. La transmisión fallará si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si una pista de audio AAC pasa a ser una pista de audio A-Law.

Los datos recuperados con esta acción son facturables. Para obtener más información, consulte Pricing.

> **Note**
>
> Para conocer las restricciones que se aplican a las sesiones MPEG-DASH, consulte Cuotas de Kinesis Video Streams.

Puedes monitorizar la cantidad de datos que consume el reproductor multimedia supervisando la CloudWatch métrica de `GetMP4MediaFragment.OutgoingBytes` Amazon. Para obtener información sobre cómo CloudWatch monitorizar Kinesis Video Streams, consulte Supervisión de Kinesis Video Streams. Para obtener información sobre precios, consulte Precios AWS y precios de Amazon Kinesis Video Streams. Se aplican cargos tanto por las sesiones de HLS como por los AWS datos salientes.

Para obtener más información sobre HLS, consulta HTTP Live Streaming en el sitio para desarrolladores de Apple.

> **Important**
>
> Si se produce un error después de invocar una API de contenido multimedia archivado de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:
>
> - `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
> - `x-amz-RequestId`Encabezado HTTP: si desea informar de un problema AWS al equipo de soporte, puede diagnosticar mejor el problema si se le proporciona el identificador de solicitud.
>
> Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente.
> Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /getDASHStreamingSessionURL HTTP/1.1
Content-type: application/json

{
   "DASHFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "DisplayFragmentNumber": "string",
   "DisplayFragmentTimestamp": "string",
   "Expires": number,
   "MaxManifestFragmentResults": number,
   "PlaybackMode": "string",
   "StreamARN": "string",
```

```
    "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

DASHFragmentSelector

El intervalo de tiempo del fragmento solicitado y el origen de las marcas de tiempo.

Este parámetro es obligatorio si `PlaybackMode` es `ON_DEMAND` o. `LIVE_REPLAY` Si lo es, este parámetro PlaybackMode es opcional `LIVE`. Si `PlaybackMode` es `LIVE` así, se `FragmentSelectorType` puede establecer, pero no se `TimestampRange` debe establecer. Si `PlaybackMode` es `ON_DEMAND` o`LIVE_REPLAY`, ambos `FragmentSelectorType` y `TimestampRange` deben estar configurados.

Tipo: objeto DASHFragmentSelector

Obligatorio: no

DisplayFragmentNumber

Los fragmentos se identifican en el archivo de manifiesto en función de su número de secuencia en la sesión. Si DisplayFragmentNumber se establece en`ALWAYS`, el número de fragmento de Kinesis Video Streams se añade a cada elemento S del archivo de manifiesto con el nombre de atributo «kvs:fn». Estos números de fragmento se pueden usar para registrar o para usarlos con otros APIs (por ejemplo, y). `GetMedia` `GetMediaForFragmentList` Se necesita un reproductor multimedia MPEG-DASH personalizado para aprovechar este atributo personalizado.

El valor predeterminado es `NEVER`.

Tipo: cadena

Valores válidos: `ALWAYS | NEVER`

Obligatorio: no

DisplayFragmentTimestamp

Según la especificación MPEG-DASH, el tiempo de reloj de pared de los fragmentos del archivo de manifiesto se puede obtener mediante los atributos del propio manifiesto. Sin embargo, por lo general, los reproductores multimedia compatibles con MPEG-DASH no gestionan adecuadamente las brechas en la cronología multimedia. Kinesis Video Streams ajusta la cronología multimedia del archivo de manifiesto para permitir la reproducción de contenido multimedia con discontinuidades. Por lo tanto, es posible que la hora del reloj de pared derivada del archivo de manifiesto no sea precisa. Si DisplayFragmentTimestamp se establece en`ALWAYS`, se añade la marca de tiempo exacta del fragmento a cada elemento S del archivo de manifiesto con el nombre de atributo «kvs:ts». Se necesita un reproductor multimedia MPEG-DASH personalizado para aprovechar este atributo personalizado.

El valor predeterminado es `NEVER`. Si DASHFragmentSelector es así`SERVER_TIMESTAMP`, las marcas de tiempo serán las marcas de tiempo de inicio del servidor. Del mismo modo, cuando DASHFragmentSelector es`PRODUCER_TIMESTAMP`, las marcas de tiempo serán las marcas de tiempo de inicio del productor.

Tipo: cadena

Valores válidos: `ALWAYS | NEVER`

Obligatorio: no

Expires

El tiempo en segundos que falta para que caduque la sesión solicitada. Este valor puede estar entre 300 (5 minutos) y 43200 (12 horas).

Cuando una sesión caduca, no se pueden realizar nuevas llamadas a `GetDashManifest` esa sesión ni se `GetMP4MediaFragment` pueden realizar para ella. `GetMP4InitFragment`

El valor predeterminado es 300 (5 minutos).

Tipo: entero

Rango válido: valor mínimo de 300. Valor máximo de 43 200.

Obligatorio: no

MaxManifestFragmentResults

El número máximo de fragmentos que se devuelven en el manifiesto MPEG-DASH.

Si el valor `PlaybackMode` es`LIVE`, se devuelven los fragmentos más recientes hasta este valor. Si el `PlaybackMode` es`ON_DEMAND`, se devuelven los fragmentos más antiguos, hasta este número máximo.

Cuando hay un mayor número de fragmentos disponibles en un manifiesto MPEG-DASH en directo, los reproductores de vídeo suelen almacenar el contenido en búfer antes de iniciar la reproducción. Al aumentar el tamaño del búfer, se aumenta la latencia de reproducción, pero se reduce la probabilidad de que se vuelva a almacenar en búfer durante la reproducción. Recomendamos que un manifiesto MPEG-DASH en directo tenga un mínimo de 3 fragmentos y un máximo de 10 fragmentos.

El valor predeterminado es de 5 fragmentos si `PlaybackMode` es `LIVE` o `LIVE_REPLAY` y 1000 si es. `PlaybackMode` `ON_DEMAND`

El valor máximo de 1000 fragmentos corresponde a más de 16 minutos de vídeo en las transmisiones con fragmentos de 1 segundo y a más de 2 horas y media de vídeo en las transmisiones con fragmentos de 10 segundos.

Tipo: largo

Rango válido: valor mínimo de 1. Valor máximo de 5000.

Obligatorio: no

PlaybackMode

Ya sea para recuperar datos en directo, reproducirlos en directo o archivados bajo demanda.

Entre las características de los tres tipos de sesiones se incluyen las siguientes:

- **`LIVE`**: Para las sesiones de este tipo, el manifiesto MPEG-DASH se actualiza continuamente con los fragmentos más recientes a medida que están disponibles. Recomendamos que el reproductor multimedia recupere un nuevo manifiesto en un intervalo de un segundo. Cuando se reproduce este tipo de sesión en un reproductor multimedia, la interfaz de usuario suele mostrar una notificación «en directo», sin necesidad de controlar la posición de visualización en la ventana de reproducción.

> **Note**
>
> En el `LIVE` modo, los fragmentos más recientes disponibles se incluyen en un manifiesto MPEG-DASH, incluso si hay un espacio entre los fragmentos (es decir,

si falta un fragmento). Un espacio como este puede provocar que el reproductor multimedia se detenga o que se interrumpa la reproducción. En este modo, los fragmentos no se añaden al manifiesto MPEG-DASH si son más antiguos que el fragmento más reciente de la lista de reproducción. Si el fragmento que falta pasa a estar disponible después de añadir otro fragmento al manifiesto, el fragmento anterior no se añadirá ni se llenará el espacio vacío.

- **`LIVE_REPLAY`**: Para las sesiones de este tipo, el manifiesto MPEG-DASH se actualiza de forma similar a como se actualiza para el `LIVE` modo, con la salvedad de que comienza por incluir fragmentos de una hora de inicio determinada. En lugar de añadir fragmentos a medida que se ingieren, los fragmentos se añaden a medida que transcurre la duración del siguiente fragmento. Por ejemplo, si los fragmentos de la sesión duran dos segundos, se añade un fragmento nuevo al manifiesto cada dos segundos. Este modo resulta útil para poder iniciar la reproducción desde el momento en que se detecta un evento y seguir retransmitiendo en directo contenido multimedia que aún no se haya ingerido en el momento de la creación de la sesión. Este modo también es útil para transmitir contenido multimedia previamente archivado sin estar limitado por el límite de 1000 fragmentos establecido en el `ON_DEMAND` modo.
- **`ON_DEMAND`**: Para sesiones de este tipo, el manifiesto MPEG-DASH contiene todos los fragmentos de la sesión, hasta el número especificado en. `MaxManifestFragmentResults` El manifiesto se debe recuperar solo una vez por sesión. Cuando este tipo de sesión se reproduce en un reproductor multimedia, la interfaz de usuario suele mostrar un control de barra para seleccionar la posición de la ventana de reproducción que se va a mostrar.

En todos los modos de reproducción, si `FragmentSelectorType` es `PRODUCER_TIMESTAMP` así, y si hay varios fragmentos con la misma marca de tiempo de inicio, el fragmento que tenga el número de fragmento mayor (es decir, el fragmento más reciente) se incluye en el manifiesto MPEG-DASH. Los demás fragmentos no están incluidos. Los fragmentos que tienen marcas de tiempo diferentes pero que tienen duraciones superpuestas se siguen incluyendo en el manifiesto MPEG-DASH. Esto puede provocar un comportamiento inesperado en el reproductor multimedia.

El valor predeterminado es `LIVE`.

Tipo: cadena

Valores válidos: `LIVE | LIVE_REPLAY | ON_DEMAND`

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de la que se va a recuperar la URL del manifiesto MPEG-DASH.

Debe especificar el o el. `StreamName` `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión de la que se va a recuperar la URL del manifiesto MPEG-DASH.

Debe especificar el o el`StreamName`. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "DASHStreamingSessionURL": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

DASHStreamingSessionURL

La URL (que contiene el token de sesión) que un reproductor multimedia puede utilizar para recuperar el manifiesto MPEG-DASH.

Tipo: cadena

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

InvalidCodecPrivateDataException

Los datos privados del códec de al menos una de las pistas de la transmisión de vídeo no son válidos para esta operación.

Código de estado HTTP: 400

MissingCodecPrivateDataException

No se encontró ningún dato privado del códec en al menos una de las pistas de la transmisión de vídeo.

Código de estado HTTP: 400

NoDataRetentionException

`GetImagesse` solicitó para una transmisión que no retiene datos (es decir, tiene un valor `DataRetentionInHours` de 0).

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

UnsupportedStreamMediaTypeException

El tipo de contenido multimedia (por ejemplo, vídeo h.264 o h.265 o audio AAC o G.711) no se pudo determinar a partir del códec de las pistas del primer fragmento IDs de una sesión de reproducción. El ID de códec de la pista 1 debería ser `V_MPEG/ISO/AVC` y, opcionalmente, el ID de códec de la pista 2 debería ser. `A_AAC`

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3

- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetHLSStreamingSessionURL

Servicio: Amazon Kinesis Video Streams Archived Media

Recupera una URL de transmisión en directo HTTP (HLS) para la transmisión. A continuación, puede abrir la URL en un navegador o reproductor multimedia para ver el contenido de la transmisión.

`StreamName`Tanto el parámetro como el `StreamARN` parámetro son opcionales, pero debes especificar uno `StreamName` o el `StreamARN` al invocar esta operación de API.

Una transmisión de vídeo de Amazon Kinesis tiene los siguientes requisitos para proporcionar datos a través de HLS:

- Requisitos de pista de reproducción de vídeo.
- La retención de datos debe ser mayor o igual que 0.
- La pista de vídeo de cada fragmento debe contener datos privados del códec en codificación de vídeo avanzada (AVC) para el formato H.264 o HEVC para el formato H.265 (especificación MPEG-4 ISO/IEC 14496-15). Si desea obtener más información sobre cómo adaptar datos de transmisiones a un determinado formato, consulte Referencia de marcas de adaptación de capa de abstracción de red (NAL).
- La pista de audio (si está presente) de cada fragmento debe contener datos privados del códec en formato AAC (especificación AAC ISO/IEC 13818-7).

Las sesiones HLS de Kinesis Video Streams contienen fragmentos en formato MPEG-4 fragmentado (también denominado MP4 f o CMAF) o MPEG-2 (también denominado fragmentos TS, que también admite la especificación HLS). Para obtener más información sobre los tipos de fragmentos HLS, consulte la especificación HLS.

El siguiente procedimiento muestra cómo utilizar HLS con Kinesis Video Streams:

1. Llame a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `GetHLSStreamingSessionURL` solicitudes a este punto final mediante el parámetro --endpoint-url.
2. Recupera la URL de HLS usando. `GetHLSStreamingSessionURL` Kinesis Video Streams crea una sesión de streaming HLS que se utiliza para acceder al contenido de una transmisión mediante el protocolo HLS. `GetHLSStreamingSessionURL`devuelve una URL autenticada (que incluye un token de sesión cifrado) de la lista de reproducción maestra de HLS de la sesión (el recurso raíz necesario para la transmisión con HLS).

> **Note**
>
> No compartas ni guardes este token en un lugar donde una entidad no autorizada pueda acceder a él. El token proporciona acceso al contenido de la transmisión. Proteja el token con las mismas medidas que utilizaría con sus AWS credenciales.

El contenido multimedia que está disponible a través de la lista de reproducción consiste únicamente en la transmisión, el intervalo de tiempo y el formato solicitados. No hay otros datos multimedia disponibles (como fotogramas fuera de la ventana solicitada o velocidades de bits alternativas).

3. Proporcione la URL (que contiene el token de sesión cifrado) de la lista de reproducción maestra del HLS a un reproductor multimedia compatible con el protocolo HLS. Kinesis Video Streams hace que la lista de reproducción multimedia, el fragmento de inicialización y los fragmentos multimedia del HLS estén disponibles a través de la URL de la lista de reproducción maestra. El fragmento de inicialización contiene los datos privados del códec de la transmisión y otros datos necesarios para configurar el decodificador y el renderizador de vídeo o audio. Los fragmentos multimedia contienen fotogramas de vídeo codificados en H.264 o muestras de audio codificadas en AAC.
4. El reproductor multimedia recibe la URL autenticada y solicita los metadatos y los datos multimedia de la transmisión con normalidad. Cuando el reproductor multimedia solicita datos, ejecuta las siguientes acciones:
   - Obtener HLSMaster lista de reproducción: recupera una lista de reproducción maestra de HLS, que contiene una URL para la `GetHLSMediaPlaylist` acción de cada pista y metadatos adicionales para el reproductor multimedia, incluida la velocidad de bits y la resolución estimadas.
   - Obtener HLSMedia lista de reproducción: recupera una lista de reproducción multimedia de HLS, que contiene una URL para acceder al fragmento de MP4 inicialización con la `GetMP4InitFragment` acción y URLs a los fragmentos multimedia con las MP4 acciones. `GetMP4MediaFragment` La lista de reproducción multimedia HLS también contiene metadatos sobre la transmisión que el reproductor necesita para reproducirla, por ejemplo, si es o. `PlaybackMode` `LIVE` `ON_DEMAND` La lista de reproducción multimedia HLS suele ser estática para las sesiones con un `PlaybackType` valor de. `ON_DEMAND` La lista de reproducción multimedia HLS se actualiza continuamente con nuevos fragmentos para las sesiones con un `PlaybackType` de. `LIVE` Hay una lista de reproducción multimedia HLS distinta para la pista

de vídeo y la pista de audio (si procede) que contiene MP4 contenido multimedia URLs para la pista específica.

- Obtener MP4InitFragment: recupera el fragmento de MP4 inicialización. El reproductor multimedia suele cargar el fragmento de inicialización antes de cargar cualquier fragmento multimedia. Este fragmento contiene los átomos `fytp` «» y `moov` «» y los MP4 átomos secundarios necesarios para inicializar el decodificador del reproductor multimedia.

  El fragmento de inicialización no corresponde a un fragmento de una transmisión de vídeo de Kinesis. Contiene solo los datos privados del códec para la transmisión y la pista correspondiente, que el reproductor multimedia necesita para decodificar los fotogramas multimedia.
- Obtener MP4MediaFragment: recupera fragmentos multimedia MP4. Estos fragmentos contienen los átomos «`moof`» y «`mdat`» y sus MP4 átomos secundarios, que contienen los marcos multimedia del fragmento codificado y sus marcas de tiempo.

> **Note**
>
> Los datos privados del códec (CPD) contenidos en cada fragmento contienen información de inicialización específica del códec, como la velocidad de fotogramas, la resolución y el perfil de codificación, que son necesarios para decodificar correctamente el fragmento. Tanto en el TS como en el CPD MP4, se admiten cambios durante una sesión de streaming. Por lo tanto, los fragmentos de una sesión pueden tener una información diferente en el CPD sin interrumpir la reproducción. Para cada sesión de streaming, solo se permiten 500 cambios de CPD.

> **Important**
>
> No se admiten los cambios de seguimiento. Las pistas deben permanecer consistentes en todo el contenido multimedia consultado. La transmisión fallará si los fragmentos de la transmisión pasan de tener solo vídeo a tener audio y vídeo, o si una pista de audio AAC pasa a ser una pista de audio A-Law.

Los datos recuperados con esta acción son facturables. Para obtener información, consulte Precios de .

- ObtenerTSFragment: recupera fragmentos de MPEG TS que contienen datos de inicialización y multimedia de todas las pistas de la transmisión.

  > **Note**
  >
  > Si `ContainerFormat` es así`MPEG_TS`, esta API se usa en lugar de `GetMP4InitFragment` y para recuperar contenido multimedia de `GetMP4MediaFragment` la transmisión.

  Los datos recuperados con esta acción son facturables. Para obtener más información, consulte los precios de Kinesis Video Streams.

La URL de una sesión de streaming no debe compartirse entre los jugadores. El servicio puede limitar una sesión si varios reproductores multimedia la comparten. Para conocer los límites de conexión, consulte Cuotas de Kinesis Video Streams.

Puedes monitorizar la cantidad de datos que consume el reproductor multimedia supervisando la CloudWatch métrica de `GetMP4MediaFragment.OutgoingBytes` Amazon. Para obtener información sobre cómo CloudWatch monitorizar Kinesis Video Streams, consulte Supervisión de Kinesis Video Streams. Para obtener información sobre precios, consulte Precios AWS y precios de Amazon Kinesis Video Streams. Se aplican cargos tanto por las sesiones de HLS como por los AWS datos salientes.

Consulte los ejemplos de reproducción de vídeo en la guía de documentación: Utilícela AWS CLI para recuperar la URL de una sesión de streaming de HLS yEjemplo: utilice HLS en HTML y JavaScript.

Para obtener más información sobre HLS, consulte HTTP Live Streaming en el sitio para desarrolladores de Apple.

> **Important**
>
> Si se produce un error después de invocar una API de contenido multimedia archivado de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:

- `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
- `x-amz-RequestId`Encabezado HTTP: si quieres informar de un problema AWS, el equipo de soporte puede diagnosticar mejor el problema si se le proporciona el identificador de la solicitud.

Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente. Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /getHLSStreamingSessionURL HTTP/1.1
Content-type: application/json

{
   "ContainerFormat": "string",
   "DiscontinuityMode": "string",
   "DisplayFragmentTimestamp": "string",
   "Expires": number,
   "HLSFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "MaxMediaPlaylistFragmentResults": number,
   "PlaybackMode": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ContainerFormat

Especifica el formato que se debe utilizar para empaquetar el soporte. Al especificar el formato del `FRAGMENTED_MP4` contenedor, se empaquetan los medios en MP4 fragmentos (f MP4 o CMAF). Este es el embalaje recomendado porque los gastos generales de embalaje son mínimos. La otra opción de formato de contenedor es`MPEG_TS`. HLS ha admitido fragmentos de MPEG TS desde su lanzamiento y, a veces, es el único paquete compatible con los reproductores HLS más antiguos. Por lo general, MPEG TS tiene un coste de empaquetado del 5 al 25 por ciento. Esto significa que MPEG TS normalmente requiere entre un 5 y un 25 por ciento más de ancho de banda y cuesta que f. MP4

El valor predeterminado es `FRAGMENTED_MP4`.

Tipo: cadena

Valores válidos: `FRAGMENTED_MP4 | MPEG_TS`

Obligatorio: no

DiscontinuityMode

Especifica cuándo se añaden a las listas de reproducción multimedia los indicadores que marcan las discontinuidades entre los fragmentos.

Los reproductores multimedia suelen crear una cronología del contenido multimedia que se va a reproducir, en función de las marcas temporales de cada fragmento. Esto significa que, si hay algún solapamiento o espacio entre los fragmentos (como es habitual si HLSFragmentSelector está configurado de esta forma`SERVER_TIMESTAMP`), la línea temporal del reproductor multimedia también tendrá pequeños espacios entre los fragmentos en algunos lugares y sobrescribirá los fotogramas en otros lugares. Los huecos en la línea temporal del reproductor multimedia pueden provocar que la reproducción se detenga y las superposiciones pueden provocar que la reproducción sea inestable. Cuando hay indicadores de discontinuidad entre los

fragmentos, se espera que el reproductor multimedia restablezca la línea temporal y, por lo tanto, el siguiente fragmento se reproduzca inmediatamente después del fragmento anterior.

Se admiten los siguientes modos:

- `ALWAYS`: se coloca un marcador de discontinuidad entre cada fragmento de la lista de reproducción multimedia del HLS. Se recomienda utilizar un valor de `ALWAYS` si las marcas de tiempo de los fragmentos no son precisas.
- `NEVER`: no hay marcadores de discontinuidad colocados en ninguna parte. Se recomienda utilizar un valor de `NEVER` para garantizar que la línea temporal del reproductor multimedia se corresponda con la mayor precisión posible con las marcas horarias del productor.
- `ON_DISCONTINUITY`: se coloca un marcador de discontinuidad entre los fragmentos que tienen un espacio o una superposición de más de 50 milisegundos. En la mayoría de los escenarios de reproducción, se recomienda utilizar un valor de `ON_DISCONTINUITY` tal forma que la línea temporal del reproductor multimedia solo se restablezca cuando haya un problema importante con la línea temporal multimedia (por ejemplo, falta un fragmento).

El valor predeterminado HLSFragmentSelector es `ALWAYS` cuando está configurado `SERVER_TIMESTAMP` en y `NEVER` cuando está configurado en`PRODUCER_TIMESTAMP`.

Tipo: cadena

Valores válidos: `ALWAYS | NEVER | ON_DISCONTINUITY`

Obligatorio: no

DisplayFragmentTimestamp

Especifica cuándo deben incluirse las marcas de tiempo de inicio del fragmento en la lista de reproducción multimedia del HLS. Normalmente, los reproductores multimedia indican la posición del cabezal de reproducción como un tiempo relativo al inicio del primer fragmento de la sesión de reproducción. Sin embargo, cuando las marcas de tiempo de inicio se incluyen en la lista de reproducción multimedia del HLS, algunos reproductores multimedia pueden indicar el cabezal de reproducción actual como una hora absoluta en función de las marcas de tiempo del fragmento. Esto puede resultar útil para crear una experiencia de reproducción que muestre a los espectadores la hora exacta del contenido multimedia.

El valor predeterminado es `NEVER`. Si HLSFragmentSelector es así`SERVER_TIMESTAMP`, las marcas horarias serán las marcas horarias de inicio del servidor. Del mismo modo, cuando

HLSFragmentSelector esPRODUCER_TIMESTAMP, las marcas de tiempo serán las marcas de tiempo de inicio del productor.

Tipo: cadena

Valores válidos: ALWAYS | NEVER

Obligatorio: no

Expires

El tiempo en segundos que falta para que caduque la sesión solicitada. Este valor puede estar entre 300 (5 minutos) y 43200 (12 horas).

Cuando una sesión caduca, no se pueden realizar nuevas llamadas a GetHLSMasterPlaylistGetHLSMediaPlaylist,GetMP4InitFragment,GetMP4MediaFragment, ni se GetTSFragment pueden realizar nuevas llamadas para esa sesión.

El valor predeterminado es 300 (5 minutos).

Tipo: entero

Rango válido: valor mínimo de 300. Valor máximo de 43 200.

Obligatorio: no

HLSFragmentSelector

El intervalo de tiempo del fragmento solicitado y el origen de las marcas de tiempo.

Este parámetro es obligatorio si PlaybackMode es ON_DEMAND o. LIVE_REPLAY Si lo es, este parámetro PlaybackMode es opcional LIVE. Si PlaybackMode es LIVE así, se FragmentSelectorType puede establecer, pero no se TimestampRange debe establecer. Si PlaybackMode es ON_DEMAND oLIVE_REPLAY, ambos FragmentSelectorType y TimestampRange deben estar configurados.

Tipo: objeto HLSFragmentSelector

Obligatorio: no

MaxMediaPlaylistFragmentResults

El número máximo de fragmentos que se devuelven en las listas de reproducción multimedia del HLS.

Si el valor `PlaybackMode` es`LIVE`, se devuelven los fragmentos más recientes hasta este valor. Si el `PlaybackMode` es`ON_DEMAND`, se devuelven los fragmentos más antiguos, hasta este número máximo.

Cuando hay un número mayor de fragmentos disponibles en una lista de reproducción multimedia HLS en directo, los reproductores de vídeo suelen almacenar el contenido en búfer antes de iniciar la reproducción. Al aumentar el tamaño del búfer, se aumenta la latencia de reproducción, pero se reduce la probabilidad de que se vuelva a almacenar en búfer durante la reproducción. Recomendamos que una lista de reproducción multimedia HLS en directo tenga un mínimo de 3 fragmentos y un máximo de 10 fragmentos.

El valor predeterminado es de 5 fragmentos si `PlaybackMode` es `LIVE` o `LIVE_REPLAY` y 1000 si `PlaybackMode` es`ON_DEMAND`.

El valor máximo de 5000 fragmentos corresponde a más de 80 minutos de vídeo en las transmisiones con fragmentos de 1 segundo y a más de 13 horas de vídeo en las transmisiones con fragmentos de 10 segundos.

Tipo: largo

Rango válido: valor mínimo de 1. Valor máximo de 5000.

Obligatorio: no

PlaybackMode

Ya sea para recuperar datos en directo, reproducirlos en directo o archivados bajo demanda.

Entre las características de los tres tipos de sesiones se incluyen las siguientes:

- **`LIVE`**: Para las sesiones de este tipo, la lista de reproducción multimedia del HLS se actualiza continuamente con los fragmentos más recientes a medida que están disponibles. Se recomienda que el reproductor multimedia recupere una nueva lista de reproducción en un intervalo de un segundo. Cuando se reproduce este tipo de sesión en un reproductor multimedia, la interfaz de usuario suele mostrar una notificación «en directo», sin necesidad de controlar el deslizador para seleccionar la posición en la ventana de reproducción en la que desea visualizarla.

> **Note**
>
> En el `LIVE` modo, los fragmentos más recientes disponibles se incluyen en una lista de reproducción multimedia del HLS, incluso si hay un espacio entre los fragmentos (es

decir, si falta un fragmento). Un espacio como este puede provocar que el reproductor multimedia se detenga o que se interrumpa la reproducción. En este modo, los fragmentos no se añaden a la lista de reproducción multimedia del HLS si son más antiguos que el fragmento más reciente de la lista de reproducción. Si el fragmento que falta pasa a estar disponible después de añadir otro fragmento a la lista de reproducción, el fragmento anterior no se añadirá y el espacio no se rellenará.

- **`LIVE_REPLAY`**: En las sesiones de este tipo, la lista de reproducción multimedia HLS se actualiza de forma similar a como se actualiza para el `LIVE` modo, con la salvedad de que comienza por incluir fragmentos de una hora de inicio determinada. En lugar de añadir fragmentos a medida que se ingieren, los fragmentos se añaden a medida que transcurre la duración del siguiente fragmento. Por ejemplo, si los fragmentos de la sesión duran dos segundos, se añade un fragmento nuevo a la lista de reproducción multimedia cada dos segundos. Este modo resulta útil para poder iniciar la reproducción desde el momento en que se detecta un evento y continuar retransmitiendo en directo contenido multimedia que aún no se haya ingerido en el momento de la creación de la sesión. Este modo también es útil para transmitir contenido multimedia previamente archivado sin estar limitado por el límite de 1000 fragmentos establecido en el `ON_DEMAND` modo.
- **`ON_DEMAND`**: Para sesiones de este tipo, la lista de reproducción multimedia HLS contiene todos los fragmentos de la sesión, hasta el número especificado en. `MaxMediaPlaylistFragmentResults` La lista de reproducción solo se debe recuperar una vez por sesión. Cuando este tipo de sesión se reproduce en un reproductor multimedia, la interfaz de usuario suele mostrar un control de barra para seleccionar la posición de la ventana de reproducción que se va a mostrar.

En todos los modos de reproducción, si `FragmentSelectorType` es `PRODUCER_TIMESTAMP` así, y si hay varios fragmentos con la misma marca de tiempo de inicio, el fragmento que tenga el mayor número de fragmentos (es decir, el fragmento más reciente) se incluye en la lista de reproducción multimedia del HLS. Los demás fragmentos no están incluidos. Los fragmentos que tienen marcas de tiempo diferentes pero que tienen duraciones superpuestas se siguen incluyendo en la lista de reproducción multimedia del HLS. Esto puede provocar un comportamiento inesperado en el reproductor multimedia.

El valor predeterminado es `LIVE`.

Tipo: cadena

Valores válidos: `LIVE | LIVE_REPLAY | ON_DEMAND`

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión de la que se va a recuperar la URL de la lista de reproducción maestra de HLS.

Debe especificar el `StreamName` o el. `StreamARN`

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la transmisión de la que se va a recuperar la URL de la lista de reproducción maestra del HLS.

Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "HLSStreamingSessionURL": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

HLSStreamingSessionURL

La URL (que contiene el token de sesión) que un reproductor multimedia puede utilizar para recuperar la lista de reproducción maestra del HLS.

Tipo: cadena

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

InvalidCodecPrivateDataException

Los datos privados del códec de al menos una de las pistas de la transmisión de vídeo no son válidos para esta operación.

Código de estado HTTP: 400

MissingCodecPrivateDataException

No se encontró ningún dato privado del códec en al menos una de las pistas de la transmisión de vídeo.

Código de estado HTTP: 400

NoDataRetentionException

`GetImages`se solicitó para una transmisión que no retiene datos (es decir, tiene un valor `DataRetentionInHours` de 0).

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

UnsupportedStreamMediaTypeException

El tipo de contenido multimedia (por ejemplo, vídeo h.264 o h.265 o audio AAC o G.711) no se pudo determinar a partir del códec de las pistas del primer fragmento IDs de una sesión de reproducción. El ID de códec de la pista 1 debería ser `V_MPEG/ISO/AVC` y, opcionalmente, el ID de códec de la pista 2 debería ser. `A_AAC`

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS

- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetImages

Servicio: Amazon Kinesis Video Streams Archived Media

El soporte gestionado para imágenes proporciona una forma totalmente gestionada de obtener imágenes de los datos de vídeo transmitidos y almacenados en Kinesis Video Streams. Puede usar imágenes para ejecutar cargas de trabajo de aprendizaje automático (ML), como la detección de personas, mascotas o vehículos. Las imágenes también se pueden utilizar para añadir elementos interactivos a la reproducción, como previsualizaciones de imágenes para eventos de movimiento o para barrer un videoclip.

GetImages también recupera una lista de imágenes correspondientes a cada marca de tiempo para un intervalo de tiempo, intervalo de muestreo y configuración de formato de imagen determinados.

> **Note**
>
> Primero debe llamar a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `GetImages` solicitudes a este punto final mediante el parámetro --endpoint-url.

Requisitos de la pista de reproducción de vídeo.

Sintaxis de la solicitud

```
POST /getImages HTTP/1.1
Content-type: application/json

{
   "EndTimestamp": number,
   "Format": "string",
   "FormatConfig": {
      "string" : "string"
   },
   "HeightPixels": number,
   "ImageSelectorType": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SamplingInterval": number,
   "StartTimestamp": number,
   "StreamARN": "string",
   "StreamName": "string",
```

```
    "WidthPixels": number
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

EndTimestamp

La marca de tiempo final del rango de imágenes que se va a generar. Si el intervalo de tiempo `StartTimestamp` comprendido entre y `EndTimestamp` supera los 300 segundos`StartTimestamp`, recibirá un. `IllegalArgumentException`

Tipo: marca temporal

Obligatorio: sí

Format

El formato que se utilizará para codificar la imagen.

Tipo: cadena

Valores válidos: `JPEG | PNG`

Obligatorio: sí

FormatConfig

La lista de una estructura de pares clave-valor que contiene parámetros adicionales que se pueden aplicar al generar la imagen. La `FormatConfig` clave es la`JPEGQuality`, que indica la clave de calidad JPEG que se utilizará para generar la imagen. El `FormatConfig` valor acepta enteros de 1 a 100. Si el valor es 1, la imagen se generará con menos calidad y con la mejor compresión. Si el valor es 100, la imagen se generará con la mejor calidad y menos compresión. Si no se proporciona ningún valor, el valor predeterminado de la `JPEGQuality` clave se establecerá en 80.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 1 elemento.

Claves válidas: `JPEGQuality`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `^[a-zA-Z_0-9]+`

Obligatorio: no

HeightPixels

La altura de la imagen de salida que se utiliza junto con el `WidthPixels` parámetro. Cuando se proporcionan ambos `HeightPixels` `WidthPixels` parámetros, la imagen se estirará para ajustarse a la relación de aspecto especificada. Si solo se proporciona el `HeightPixels` parámetro, se utilizará su relación de aspecto original para `WidthPixels` calcularla. Si no se proporciona ninguno de los parámetros, se devolverá el tamaño original de la imagen.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 2160.

Obligatorio: no

ImageSelectorType

El origen de las marcas de tiempo del servidor o del productor que se van a utilizar para generar las imágenes.

Tipo: cadena

Valores válidos: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Obligatorio: sí

MaxResults

El número máximo de imágenes que devolverá la API.

> **Note**
>
> El límite predeterminado es de 25 imágenes por respuesta de la API. Si se proporciona un valor `MaxResults` superior a este valor, se obtendrá un tamaño de página de 25. Los resultados adicionales se paginarán.

Tipo: largo

Rango válido: valor mínimo de 1. Valor máximo de 100.

Obligatorio: no

NextToken

Un token que especifica dónde empezar a paginar el siguiente conjunto de imágenes. Es el `GetImages:NextToken` de una respuesta truncada anteriormente.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 4096 caracteres.

Patrón: `[a-zA-Z0-9+/]+={0,2}`

Obligatorio: no

SamplingInterval

El intervalo de tiempo en milisegundos (ms) en el que se deben generar las imágenes a partir de la transmisión. El valor mínimo que se puede proporcionar es de 200 ms (5 imágenes por segundo). Si el intervalo de marca temporal es inferior al intervalo de muestreo, se `startTimestamp` devolverá la imagen del, si está disponible.

Tipo: entero

Obligatorio: no

StartTimestamp

El punto de partida desde el que se deben generar las imágenes. `StartTimestamp`Debe estar dentro de un rango inclusivo de marcas de tiempo para que se devuelva una imagen.

Tipo: marca temporal

Obligatorio: sí

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión desde la que se van a recuperar las imágenes. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la secuencia de la que se van a recuperar las imágenes. Debe especificar el `StreamName` o el`StreamARN`.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

WidthPixels

El ancho de la imagen de salida que se utiliza junto con el `HeightPixels` parámetro. Cuando se proporcionan ambos `WidthPixels` `HeightPixels` parámetros, la imagen se estirará para ajustarse a la relación de aspecto especificada. Si solo se proporciona el `WidthPixels` parámetro o si solo `HeightPixels` se proporciona el parámetro, `ValidationException` aparecerá un. Si no se proporciona ninguno de los parámetros, se devolverá el tamaño de imagen original de la secuencia.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 3840.

Obligatorio: no

Sintaxis de la respuesta

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

{
   "Images": [
      {
```

```
            "Error": "string",
            "ImageContent": "string",
            "TimeStamp": number
        }
    ],
    "NextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Images

La lista de imágenes generadas a partir de la transmisión de vídeo. Si no hay contenido multimedia disponible para la marca de tiempo indicada, el `NO_MEDIA` error aparecerá en la salida. Si se produce un error durante la generación de la imagen, `MEDIA_ERROR` aparecerá en el resultado como la causa de la falta de imagen.

Tipo: matriz de objetos Image

NextToken

El token cifrado que se utilizó en la solicitud para obtener más imágenes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 4096 caracteres.

Patrón: `[a-zA-Z0-9+/]+={0,2}`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

NoDataRetentionException

`GetImages`se solicitó para una transmisión que no retiene datos (es decir, tiene un valor `DataRetentionInHours` de 0).

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2

- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetMediaForFragmentList

Servicio: Amazon Kinesis Video Streams Archived Media

Obtiene el contenido multimedia de una lista de fragmentos (especificada por el número de fragmento) de los datos archivados en una transmisión de vídeo de Amazon Kinesis.

> **Note**
>
> Primero debe llamar a la `GetDataEndpoint` API para obtener un punto de enlace. A continuación, envíe las `GetMediaForFragmentList` solicitudes a este punto final mediante el parámetro --endpoint-url.

Para conocer los límites, consulte las cuotas de Kinesis Video Streams.

> **Important**
>
> Si se produce un error después de invocar una API de contenido multimedia archivado de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:
>
> - `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
> - `x-amz-RequestId`Encabezado HTTP: si quieres informar de un problema AWS, el equipo de soporte puede diagnosticar mejor el problema si se le proporciona el identificador de la solicitud.
>
> Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente. Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /getMediaForFragmentList HTTP/1.1
```

```
Content-type: application/json

{
   "Fragments": [ "string" ],
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

Fragments

Una lista del número de fragmentos para recuperar el contenido multimedia. Puede recuperar estos valores conListFragments.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 artículo. La cantidad máxima es de 1000 elementos.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[0-9]+$`

Obligatorio: sí

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión desde la que se van a recuperar los fragmentos multimedia. Especifique este parámetro o el `StreamName` parámetro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la secuencia desde la que se van a recuperar los fragmentos multimedia. Especifique este parámetro o el `StreamARN` parámetro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

La respuesta devuelve los siguientes encabezados HTTP.

ContentType

El tipo de contenido del medio solicitado.

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[a-zA-Z0-9_\.\-]+$`

La respuesta devuelve lo siguiente como el cuerpo HTTP.

Payload

La carga útil que devuelve Kinesis Video Streams es una secuencia de fragmentos de la transmisión especificada. Para obtener información sobre los fragmentos, consulte. PutMedia

Los fragmentos que Kinesis Video Streams devuelve en `GetMediaForFragmentList` la convocatoria también incluyen las siguientes etiquetas Matroska (MKV) adicionales:

- AWS_KINESISVIDEO_FRAGMENT_NUMBER - Número de fragmento devuelto en el fragmento.
- AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP - Marca de tiempo del fragmento en el servidor.
- AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP - Marca de tiempo del fragmento por parte del productor.

Si se produce una excepción, se incluirán las siguientes etiquetas:

- AWS_KINESISVIDEO_FRAGMENT_NUMBER - El número del fragmento que generó la excepción.
- AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE - El código entero del error.
- AWS_KINESISVIDEO_EXCEPTION_MESSAGE - Una descripción textual de la excepción.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListFragments

Servicio: Amazon Kinesis Video Streams Archived Media

Devuelve una lista de Fragment objetos del flujo y el rango de marcas de tiempo especificados dentro de los datos archivados.

Al final, enumerar los fragmentos es coherente. Esto significa que, aunque el productor reciba un acuse de recibo de que un fragmento se conserva, es posible que el resultado no se devuelva inmediatamente al solicitarlo. `ListFragments` Sin embargo, los resultados suelen estar disponibles en menos de un segundo.

> **Note**
>
> Primero debe llamar a la `GetDataEndpoint` API para obtener un punto final. A continuación, envíe las `ListFragments` solicitudes a este punto final mediante el parámetro --endpoint-url.

> **Important**
>
> Si se produce un error después de invocar una API de contenido multimedia archivado de Kinesis Video Streams, además del código de estado HTTP y el cuerpo de la respuesta, se incluye la siguiente información:
>
> - `x-amz-ErrorType`Encabezado HTTP: contiene un tipo de error más específico además del que proporciona el código de estado HTTP.
> - `x-amz-RequestId`Encabezado HTTP: si quieres informar de un problema AWS, el equipo de soporte puede diagnosticar mejor el problema si se le proporciona el identificador de la solicitud.
>
> Tanto el código de estado HTTP como el ErrorType encabezado se pueden utilizar para tomar decisiones programáticas sobre si los errores se pueden volver a cometer y en qué condiciones, así como para proporcionar información sobre las medidas que el programador del cliente podría tener que tomar para volver a intentarlo correctamente. Para obtener más información, consulte la sección de errores al final de este tema, así como la sección Errores comunes.

Sintaxis de la solicitud

```
POST /listFragments HTTP/1.1
Content-type: application/json

{
   "FragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "MaxResults": number,
   "NextToken": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

FragmentSelector

Describe el rango de marcas de tiempo y el origen de las marcas de tiempo del rango de fragmentos que se van a devolver.

> **Note**
>
> Esto solo es necesario cuando `NextToken` no se incluye en la API.

Tipo: objeto FragmentSelector

Obligatorio: no

MaxResults

El número total de fragmentos que se van a devolver. Si el número total de fragmentos disponibles es superior al valor especificado en`max-results`, NextToken se proporciona un ListFragments: en el resultado, que puede utilizar para reanudar la paginación.

El valor predeterminado es 100.

Tipo: largo

Rango válido: valor mínimo de 1. Valor máximo de 1000.

Obligatorio: no

NextToken

Un token destinado a especificar dónde iniciar la paginación. Este es el ListFragments: NextToken de una respuesta previamente truncada.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 4096 caracteres.

Patrón: `[a-zA-Z0-9+/]+={0,2}`

Obligatorio: no

StreamARN

El nombre del recurso de Amazon (ARN) de la transmisión desde la que se va a recuperar una lista de fragmentos. Especifique este parámetro o el `StreamName` parámetro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre de la secuencia desde la que se va a recuperar una lista de fragmentos. Especifique este parámetro o el `StreamARN` parámetro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "Fragments": [
      {
         "FragmentLengthInMilliseconds": number,
         "FragmentNumber": "string",
         "FragmentSizeInBytes": number,
         "ProducerTimestamp": number,
         "ServerTimestamp": number
      }
   ],
   "NextToken": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Fragments

Una lista de Fragment objetos archivados de la secuencia que cumplen los criterios del selector. Los resultados no están en un orden específico, ni siquiera en todas las páginas.

Si no hay fragmentos en la secuencia que cumplan los criterios de selección, se devuelve una lista vacía.

Tipo: matriz de objetos Fragment

NextToken

Si la lista devuelta está truncada, la operación devuelve este token para usarlo en la siguiente página de resultados. Este valor se produce `null` cuando no hay más resultados que devolver.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 4096 caracteres.

Patrón: `[a-zA-Z0-9+/]+={0,2}`

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado un límite. Intente realizar la llamada más tarde. Para obtener información sobre los límites, consulte Cuotas de Kinesis Video Streams.

Código de estado HTTP: 400

InvalidArgumentException

Un parámetro específico supera sus restricciones, no se admite o no se puede usar.

Código de estado HTTP: 400

NotAuthorizedException

Código de estado: 403. La persona que llama no está autorizada a realizar ninguna operación en la transmisión en cuestión o el token ha caducado.

Código de estado HTTP: 401

ResourceNotFoundException

`GetImages`mostrará este error cuando Kinesis Video Streams no encuentre la transmisión que ha especificado.

`GetHLSStreamingSessionURL`y `GetDASHStreamingSessionURL` arroja este error si una sesión con un `PlaybackMode` de `ON_DEMAND` o `LIVE_REPLAY` se solicita una transmisión que

no tiene fragmentos en el intervalo de tiempo solicitado, o si `LIVE` se solicita una sesión con un `PlaybackMode` de o para una transmisión que no tiene fragmentos en los últimos 30 segundos.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Canales de señalización de vídeo de Amazon Kinesis

Los canales de señalización de vídeo de Amazon Kinesis admiten las siguientes acciones:

- GetIceServerConfig
- SendAlexaOfferToMaster

GetIceServerConfig

Servicio: Amazon Kinesis Video Signaling Channels

Nota: Antes de usar esta API, debes llamar a la `GetSignalingChannelEndpoint` API para solicitar el punto final HTTPS. A continuación, especificas el punto final y la región en tu solicitud de `GetIceServerConfig` API.

Obtiene la información de configuración del servidor del Interactive Connectivity Establishment (ICE) URIs, incluidos el nombre de usuario y la contraseña que se pueden utilizar para configurar la conexión WebRTC. El componente ICE utiliza esta información de configuración para configurar la conexión WebRTC, incluida la autenticación con el servidor de retransmisión Traversal Using Relays around NAT (TURN).

TURN es un protocolo que se utiliza para mejorar la conectividad de las aplicaciones. peer-to-peer Al proporcionar un servicio de retransmisión basado en la nube, TURN garantiza que se pueda establecer una conexión incluso cuando uno o más pares no puedan establecer una peer-to-peer conexión directa. Para obtener más información, consulte una API REST para acceder a los servicios de TURN.

Puede invocar esta API para establecer un mecanismo alternativo en caso de que alguno de los pares no pueda establecer una peer-to-peer conexión directa a través de un canal de señalización. Debe especificar el nombre de recurso de Amazon (ARN) de su canal de señalización para poder invocar esta API.

Sintaxis de la solicitud

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ClientId": "string",
   "Service": "string",
   "Username": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El ARN del canal de señalización que se utilizará para la peer-to-peer conexión entre pares configurados.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

ClientId

Identificador único para el espectador. Debe ser único dentro del canal de señalización.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Service

Especifica el servicio deseado. Actualmente, `TURN` es el único valor válido.

Tipo: cadena

Valores válidos: `TURN`

Obligatorio: no

Username

Un seudónimo opcional que se asociará a las credenciales.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Sintaxis de la respuesta

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

{
   "IceServerList": [
      {
         "Password": "string",
         "Ttl": number,
         "Uris": [ "string" ],
         "Username": "string"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

IceServerList

La lista de objetos de información del servidor ICE.

Tipo: matriz de objetos IceServer

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Su solicitud se ha limitado porque ha superado el límite de llamadas de clientes permitidas. Intenta hacer la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

InvalidClientException

El cliente especificado no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

No se encuentra el recurso especificado.

Código de estado HTTP: 404

SessionExpiredException

Si la sesión del cliente ha caducado. Una vez que el cliente está conectado, la sesión es válida durante 45 minutos. El cliente debe volver a conectarse al canal para seguir enviando/recibiendo mensajes.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas AWS SDKs específicos, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2

- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

SendAlexaOfferToMaster

Servicio: Amazon Kinesis Video Signaling Channels

> **Note**
>
> Antes de usar esta API, debe llamar a la `GetSignalingChannelEndpoint` API para obtener un punto final. A continuación, especificas el punto final y la región en tu solicitud de `SendAlexaOfferToMaster` API.

Esta API le permite conectar dispositivos habilitados para WebRTC con dispositivos de pantalla Alexa. Cuando se invoca, envía la oferta del Protocolo de descripción de sesiones (SDP) de Alexa al par maestro. La oferta se entrega tan pronto como el maestro se conecta al canal de señalización especificado. Esta API devuelve la respuesta SDP del maestro conectado. Si el maestro no está conectado al canal de señalización, las solicitudes de reenvío se realizan hasta que caduque el mensaje.

Sintaxis de la solicitud

```
POST /v1/send-alexa-offer-to-master HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "MessagePayload": "string",
   "SenderClientId": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

ChannelARN

El nombre de recurso de Amazon (ARN) del canal de señalización mediante el cual se comunican Alexa y el par maestro.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

MessagePayload

El SDP codificado en base64 ofrece contenido.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 10000.

Patrón: `[a-zA-Z0-9+/=]+`

Obligatorio: sí

SenderClientId

El identificador único del cliente remitente.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: sí

Sintaxis de la respuesta

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

{
   "Answer": "string"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Answer

El contenido de la respuesta SDP codificado en base64.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 10000.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

ClientLimitExceededException

Tu solicitud se ha retrasado porque has superado el límite de llamadas de clientes permitidas. Intenta hacer la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

NotAuthorizedException

La persona que llama no está autorizada a realizar esta operación.

Código de estado HTTP: 401

ResourceNotFoundException

No se encuentra el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Almacenamiento WebRTC de vídeo en Amazon Kinesis

El almacenamiento WebRTC de vídeo de Amazon Kinesis admite las siguientes acciones:

- JoinStorageSession
- JoinStorageSessionAsViewer

JoinStorageSession

Servicio: Amazon Kinesis Video WebRTC Storage

> **Note**
>
> Antes de usar esta API, debe llamar a la `GetSignalingChannelEndpoint` API para solicitar el punto final WEBRTC. A continuación, especifique el punto final y la región en su solicitud de `JoinStorageSession` API.

Únase a la sesión actual de WebRTC de and/or audio multidireccional de vídeo unidireccional como dispositivo de producción de vídeo para un canal de entrada. Si no hay ninguna sesión existente para el canal, cree una nueva sesión de streaming y proporcione el nombre de recurso de Amazon (ARN) del canal de señalización.

Actualmente, para `SINGLE_MASTER` este tipo, un dispositivo de producción de vídeo es capaz de incorporar contenido multimedia de audio y vídeo en una transmisión. Solo los dispositivos de producción de vídeo pueden unirse a la sesión y grabar contenido multimedia.

> **Important**
>
> Actualmente, se requieren pistas de audio y vídeo para la ingesta de WebRTC.
> Requisitos actuales:
>
> - Pista de vídeo: H.264
> - Pista de audio: Opus

El vídeo ingerido resultante en la transmisión de vídeo de Kinesis tendrá los siguientes parámetros: vídeo H.264 y audio AAC.

Una vez que el participante principal haya negociado una conexión a través de WebRTC, la sesión multimedia ingerida se almacenará en la transmisión de vídeo de Kinesis. De este modo, varios espectadores podrán reproducir contenido multimedia en tiempo real a través de nuestra función de reproducción. APIs

También puede utilizar las funciones existentes de Kinesis Video Streams, `HLS` como `DASH` la reproducción, la generación de imágenes GetImagesmediante, etc., con medios WebRTC incorporados.

> **Note**
>
> Actualmente, no se admiten la entrega de imágenes ni las notificaciones de S3.

> **Note**
>
> Suponga que solo se puede asociar un cliente de un dispositivo de producción de vídeo a una sesión del canal. Si más de un cliente se une a la sesión de un canal específico como dispositivo de producción de vídeo, prevalece la solicitud más reciente del cliente.

Información adicional

- Idempotente: esta API no es idempotente.
- Comportamiento de reintento: se considera una nueva llamada a la API.
- Llamadas simultáneas: se permiten las llamadas simultáneas. Se envía una oferta por cada llamada.

Sintaxis de la solicitud

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json

{
   "channelArn": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

channelArn

El nombre de recurso de Amazon (ARN) del canal de señalización.

> **Important**
>
> Tenga en cuenta el uso de mayúsculas en este parámetro de entrada.

Tipo: cadena

Patrón: `arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 403

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

No se encuentra el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

JoinStorageSessionAsViewer

Servicio: Amazon Kinesis Video WebRTC Storage

> **Note**
>
> La ingesta de WebRTC con soporte para varios visores (versión preliminar) se proporciona en la versión preliminar, tal como se define en los términos AWS del servicio, y está sujeta a cambios. Actualmente solo está disponible en us-east-1 (IAD).
> Para participar en la vista previa, envíanos un correo electrónico a -preview@amazon.com. kvs-webrtc-multi-view

> **Note**
>
> Antes de usar esta API, debe llamar a la `GetSignalingChannelEndpoint` API para solicitar el punto final WEBRTC. A continuación, especifique el punto final y la región en su solicitud de `JoinStorageSessionAsViewer` API.

`JoinStorageSessionAsViewer`permite al espectador unirse a una sesión de streaming WebRTC grabada en la nube en curso. Esta API inicia una conexión WebRTC enviando una oferta de SDP y candidatos a ICE entre el espectador y el agente de grabación, lo que permite al espectador recibir vídeo en tiempo real del maestro a través del agente de grabación y participar en una comunicación de audio bidireccional a través del agente de grabación. Una vez conectado, el audio del espectador, si se proporciona, se reenvía a todos los demás pares conectados, incluido el participante principal, y se incorpora a la transmisión de WebRTC que se guarda en la transmisión de Kinesis Video.

> **Important**
>
> Por el momento, los espectadores no pueden enviar ninguna pista de vídeo. Los espectadores pueden enviar una pista de audio opcional o no enviar ninguna pista. Requisitos actuales de envío de los espectadores y participantes:
>
> - Pista de vídeo: no compatible
> - Pista de audio (opcional): Opus

Si el participante principal está conectado actualmente al dispositivo de producción de vídeo, el vídeo ingerido resultante en la transmisión de vídeo de Kinesis tendrá los siguientes parámetros: vídeo H.264 y audio AAC.

> **Note**
>
> Los espectadores participantes se conectan directamente a la sesión de almacenamiento y no directamente al participante principal. La sesión de almacenamiento se encarga de mezclar, duplicar y enrutar el contenido multimedia al destino adecuado.

> **Note**
>
> Si el participante principal no está presente, los espectadores no podrán escucharse entre sí.

Una vez que el espectador participante haya negociado una conexión a través de WebRTC, la sesión de audio ingerida se almacenará en la transmisión de vídeo de Kinesis siempre que el participante principal también esté conectado a la sesión de almacenamiento.

También puede utilizar las funciones existentes de Kinesis Video Streams, `HLS` como `DASH` la reproducción, la generación de imágenes GetImagesmediante, etc., con medios WebRTC incorporados.

> **Note**
>
> Actualmente, no se admiten la entrega de imágenes ni las notificaciones de S3.

> **Note**
>
> Suponga que solo se puede asociar un cliente de un dispositivo de producción de vídeo a una sesión del canal. Si más de un cliente se une a la sesión de un canal específico como dispositivo de producción de vídeo, prevalece la solicitud más reciente del cliente.

Límites

Los límites actuales están vigentes:

- Número máximo de espectadores: 3
- Tiempo máximo que los espectadores permanecen conectados a la sesión de almacenamiento sin la presencia de un participante principal: 3 minutos

> **Important**
>
> Si un espectador se desconecta de la sesión de almacenamiento (cierra la conexión entre pares), su cuota (límite de espectadores) permanece consumida durante 1 minuto. Durante este período de 1 minuto, el espectador puede invocar esta API con el mismo ID de cliente para volver a unirse a la sesión sin consumir una cuota de espectadores adicional. Transcurrido un minuto, se liberará la cuota de espectadores y estará disponible para que otros espectadores se unan.

Información adicional

- Idempotente: esta API no es idempotente.
- Comportamiento de reintento: se considera una nueva llamada a la API.
- Llamadas simultáneas: se permiten las llamadas simultáneas. Se envía una oferta por cada llamada.

Sintaxis de la solicitud

```
POST /joinStorageSessionAsViewer HTTP/1.1
Content-type: application/json

{
   "channelArn": "string",
   "clientId": "string"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

channelArn

El nombre de recurso de Amazon (ARN) del canal de señalización.

> **Important**
>
> Tenga en cuenta el uso de mayúsculas en este parámetro de entrada.

Tipo: cadena

Patrón: `arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: sí

clientId

El identificador único del cliente remitente.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

Para obtener información acerca de los errores comunes a todas las acciones, consulte Errores comunes.

AccessDeniedException

No tiene los permisos necesarios para realizar esta operación.

Código de estado HTTP: 403

ClientLimitExceededException

Kinesis Video Streams ha limitado la solicitud porque ha superado el límite de llamadas de clientes permitidas. Intente realizar la llamada más tarde.

Código de estado HTTP: 400

InvalidArgumentException

El valor de este parámetro de entrada no es válido.

Código de estado HTTP: 400

ResourceNotFoundException

No se encuentra el recurso especificado.

Código de estado HTTP: 404

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Data Types

Amazon Kinesis Video Streams admite los siguientes tipos de datos:

- ChannelInfo
- ChannelNameCondition
- DeletionConfig
- EdgeAgentStatus
- EdgeConfig
- ImageGenerationConfiguration
- ImageGenerationDestinationConfig
- LastRecorderStatus
- LastUploaderStatus
- ListEdgeAgentConfigurationsEdgeConfig
- LocalSizeConfig
- MappedResourceConfigurationListItem
- MediaSourceConfig
- MediaStorageConfiguration
- NotificationConfiguration
- NotificationDestinationConfig
- RecorderConfig
- ResourceEndpointListItem
- ScheduleConfig
- SingleMasterChannelEndpointConfiguration
- SingleMasterConfiguration
- StreamInfo
- StreamNameCondition
- Tag
- UploaderConfig

Amazon Kinesis Video Streams Media admite los siguientes tipos de datos:

- StartSelector

Los medios archivados de Amazon Kinesis Video Streams admiten los siguientes tipos de datos:

- ClipFragmentSelector
- ClipTimestampRange
- DASHFragmentSelector
- DASHTimestampRange
- Fragment
- FragmentSelector
- HLSFragmentSelector
- HLSTimestampRange
- Image
- TimestampRange

Los canales de señalización de vídeo de Amazon Kinesis admiten los siguientes tipos de datos:

- IceServer

El almacenamiento WebRTC de vídeo de Amazon Kinesis admite los siguientes tipos de datos:

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams admite los siguientes tipos de datos:

- ChannelInfo
- ChannelNameCondition
- DeletionConfig
- EdgeAgentStatus
- EdgeConfig
- ImageGenerationConfiguration
- ImageGenerationDestinationConfig
- LastRecorderStatus
- LastUploaderStatus

- ListEdgeAgentConfigurationsEdgeConfig
- LocalSizeConfig
- MappedResourceConfigurationListItem
- MediaSourceConfig
- MediaStorageConfiguration
- NotificationConfiguration
- NotificationDestinationConfig
- RecorderConfig
- ResourceEndpointListItem
- ScheduleConfig
- SingleMasterChannelEndpointConfiguration
- SingleMasterConfiguration
- StreamInfo
- StreamNameCondition
- Tag
- UploaderConfig

ChannelInfo

Servicio: Amazon Kinesis Video Streams

Estructura que encapsula los metadatos y las propiedades de un canal de señalización.

Contenido

ChannelARN

El nombre de recurso de Amazon (ARN) del canal de señalización.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

ChannelName

El nombre del canal de señalización.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

ChannelStatus

Estado actual del canal de señalización.

Tipo: cadena

Valores válidos: `CREATING | ACTIVE | UPDATING | DELETING`

Obligatorio: no

ChannelType

El tipo de canal de señalización.

Tipo: cadena

Valores válidos: `SINGLE_MASTER | FULL_MESH`

Obligatorio: no

CreationTime

El momento en el que se creó el canal de señalización.

Tipo: marca temporal

Obligatorio: no

SingleMasterConfiguration

Estructura que contiene la configuración del tipo `SINGLE_MASTER` de canal.

Tipo: objeto SingleMasterConfiguration

Obligatorio: no

Version

La versión actual del canal de señalización.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ChannelNameCondition

Servicio: Amazon Kinesis Video Streams

Un parámetro de entrada opcional para la `ListSignalingChannels` API. Si se especifica este parámetro durante la invocación`ListSignalingChannels`, la API devuelve solo los canales que cumplen una condición especificada en`ChannelNameCondition`.

Contenido

ComparisonOperator

Un operador de comparación. Actualmente, solo puede especificar el `BEGINS_WITH` operador, que busca los canales de señalización cuyos nombres comiencen con un prefijo determinado.

Tipo: cadena

Valores válidos: `BEGINS_WITH`

Obligatorio: no

ComparisonValue

Un valor para comparar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DeletionConfig

Servicio: Amazon Kinesis Video Streams

Los detalles de configuración necesarios para eliminar la conexión de la transmisión del agente perimetral.

Contenido

DeleteAfterUpload

El `boolean` valor que se utiliza para indicar si desea o no marcar el contenido multimedia para su eliminación, una vez que se ha cargado en la nube de Kinesis Video Stream. Los archivos multimedia se pueden eliminar si alguno de los valores de la configuración de eliminación está establecido`true`, por ejemplo`EdgeRetentionInHours`, cuando se ha alcanzado el `MaxLocalMediaSizeInMB` límite de o de.

Como el valor predeterminado está establecido en`true`, configure el programa de carga de forma que los archivos multimedia no se eliminen antes de cargarlos inicialmente AWS en la nube.

Tipo: Booleano

Obligatorio: no

EdgeRetentionInHours

El número de horas que desea conservar los datos de la transmisión en el Edge Agent. El valor predeterminado del tiempo de retención es de 720 horas, lo que se traduce en 30 días.

Tipo: entero

Rango válido: valor mínimo de 1. El valor máximo es 720.

Obligatorio: no

LocalSizeConfig

El valor del tamaño local necesario para eliminar la configuración de ejes.

Tipo: objeto LocalSizeConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

EdgeAgentStatus

Servicio: Amazon Kinesis Video Streams

Un objeto que contiene los detalles de estado más recientes de las tareas de grabación y carga de un agente perimetral. Utilice esta información para determinar el estado actual de un agente perimetral.

Contenido

LastRecorderStatus

El estado más reciente de un trabajo de grabación perimetral de una transmisión.

Tipo: objeto LastRecorderStatus

Obligatorio: no

LastUploaderStatus

El estado más reciente del trabajo de carga de una transmisión desde la periferia a la nube.

Tipo: objeto LastUploaderStatus

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

EdgeConfig

Servicio: Amazon Kinesis Video Streams

Una descripción de la configuración perimetral de la transmisión que se utilizará para sincronizarse con el componente Edge Agent IoT Greengrass. El componente Edge Agent se ejecutará en un dispositivo IoT Hub configurado en sus instalaciones.

Contenido

HubDeviceArn

El «tema del Internet de las cosas (IoT)» de la corriente.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Obligatorio: sí

RecorderConfig

La configuración de la grabadora consta de los `MediaSourceConfig` detalles locales, que se utilizan como credenciales para acceder a los archivos multimedia locales transmitidos por la cámara.

Tipo: objeto RecorderConfig

Obligatorio: sí

DeletionConfig

La configuración de eliminación se compone de los detalles de configuración del tiempo de retención (`EdgeRetentionInHours`) y del tamaño local (`LocalSizeConfig`) que se utilizan para realizar la eliminación.

Tipo: objeto DeletionConfig

Obligatorio: no

UploaderConfig

La configuración del cargador contiene los `ScheduleExpression` detalles que se utilizan para programar los trabajos de carga de los archivos multimedia grabados desde el agente Edge a una transmisión de vídeo de Kinesis.

Tipo: objeto UploaderConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ImageGenerationConfiguration

Servicio: Amazon Kinesis Video Streams

La estructura que contiene la información necesaria para la entrega de las imágenes KVS. Si es nulo, la configuración se eliminará de la transmisión.

Contenido

DestinationConfig

Estructura que contiene la información necesaria para entregar las imágenes a un cliente.

Tipo: objeto ImageGenerationDestinationConfig

Obligatorio: sí

Format

El formato de imagen aceptado.

Tipo: cadena

Valores válidos: `JPEG | PNG`

Obligatorio: sí

ImageSelectorType

El origen de las marcas de tiempo del servidor o del productor que se van a utilizar para generar las imágenes.

Tipo: cadena

Valores válidos: `SERVER_TIMESTAMP | PRODUCER_TIMESTAMP`

Obligatorio: sí

SamplingInterval

El intervalo de tiempo en milisegundos (ms) en el que se deben generar las imágenes a partir de la transmisión. El valor mínimo que se puede proporcionar es de 200 ms. Si el intervalo de marca temporal es inferior al intervalo de muestreo, se devolverá la imagen del `StartTimestamp` intervalo, si está disponible.

Tipo: entero

Obligatorio: sí

Status

Indica si la `ContinuousImageGenerationConfigurations` API está habilitada o deshabilitada.

Tipo: cadena

Valores válidos: `ENABLED | DISABLED`

Obligatorio: sí

FormatConfig

La lista de una estructura de pares clave-valor que contiene parámetros adicionales que se pueden aplicar cuando se genera la imagen. La `FormatConfig` clave es la`JPEGQuality`, que indica la clave de calidad JPEG que se utilizará para generar la imagen. El `FormatConfig` valor acepta enteros de 1 a 100. Si el valor es 1, la imagen se generará con menos calidad y con la mejor compresión. Si el valor es 100, la imagen se generará con la mejor calidad y menos compresión. Si no se proporciona ningún valor, el valor predeterminado de la `JPEGQuality` clave se establecerá en 80.

Tipo: mapa de cadena a cadena

Entradas de mapa: número máximo de 1 elemento.

Claves válidas: `JPEGQuality`

Limitaciones de longitud de los valores: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón de valores: `^[a-zA-Z_0-9]+`

Obligatorio: no

HeightPixels

La altura de la imagen de salida que se utiliza junto con el `WidthPixels` parámetro. Cuando se proporcionan ambos `HeightPixels` `WidthPixels` parámetros, la imagen se estirará para ajustarse a la relación de aspecto especificada. Si solo se proporciona el `HeightPixels` parámetro, se utilizará su relación de aspecto original para `WidthPixels` calcularla. Si no se proporciona ninguno de los parámetros, se devolverá el tamaño original de la imagen.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 2160.

Obligatorio: no

WidthPixels

El ancho de la imagen de salida que se utiliza junto con el `HeightPixels` parámetro. Cuando se proporcionan ambos `WidthPixels` `HeightPixels` parámetros, la imagen se estirará para ajustarse a la relación de aspecto especificada. Si solo se proporciona el `WidthPixels` parámetro, se utilizará su relación de aspecto original para `HeightPixels` calcularla. Si no se proporciona ninguno de los parámetros, se devolverá el tamaño original de la imagen.

Tipo: entero

Rango válido: valor mínimo de 1. Valor máximo de 3840.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ImageGenerationDestinationConfig

Servicio: Amazon Kinesis Video Streams

La estructura que contiene la información necesaria para entregar las imágenes a un cliente.

Contenido

DestinationRegion

La AWS región del depósito S3 en la que se entregarán las imágenes. `DestinationRegion`Debe coincidir con la región en la que se encuentra la transmisión.

Tipo: cadena

Restricciones de longitud: longitud mínima de 9. Longitud máxima de 14.

Patrón: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Obligatorio: sí

Uri

El identificador uniforme de recursos (URI) que identifica dónde se entregarán las imágenes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 255 caracteres.

Patrón: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LastRecorderStatus

Servicio: Amazon Kinesis Video Streams

El estado más reciente de un trabajo de grabación perimetral de una transmisión.

Contenido

JobStatusDetails

Descripción del estado más reciente de un trabajo de grabación.

Tipo: cadena

Requerido: no

LastCollectedTime

La marca de tiempo en la que se ejecutó por última vez el trabajo de grabación y el medio se almacenó en el disco local.

Tipo: marca temporal

Obligatorio: no

LastUpdatedTime

La marca de tiempo en la que se actualizó por última vez el estado de la grabadora.

Tipo: marca temporal

Obligatorio: no

RecorderStatus

El estado del último trabajo de grabación.

Tipo: cadena

Valores válidos: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LastUploaderStatus

Servicio: Amazon Kinesis Video Streams

El estado más reciente del trabajo de carga de una transmisión desde la periferia a la nube.

Contenido

JobStatusDetails

Una descripción del estado más reciente de un trabajo de subida.

Tipo: cadena

Requerido: no

LastCollectedTime

La marca temporal en la que se ejecutó por última vez el trabajo de carga y el contenido multimedia se recopiló en la nube.

Tipo: marca temporal

Obligatorio: no

LastUpdatedTime

La marca de tiempo en la que se actualizó por última vez el estado de la persona que subió el archivo.

Tipo: marca temporal

Obligatorio: no

UploaderStatus

El estado del último trabajo de carga.

Tipo: cadena

Valores válidos: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ListEdgeAgentConfigurationsEdgeConfig

Servicio: Amazon Kinesis Video Streams

Descripción de la configuración perimetral de un solo flujo.

Contenido

CreationTime

La marca temporal en la que la transmisión creó por primera vez la configuración de borde.

Tipo: marca temporal

Obligatorio: no

EdgeConfig

Una descripción de la configuración perimetral de la transmisión que se utilizará para sincronizarse con el componente Edge Agent IoT Greengrass. El componente Edge Agent se ejecutará en un dispositivo IoT Hub configurado en sus instalaciones.

Tipo: objeto EdgeConfig

Obligatorio: no

FailedStatusDetails

Una descripción del estado de fallo generado.

Tipo: cadena

Requerido: no

LastUpdatedTime

La marca de tiempo de la última vez que la transmisión actualizó la configuración de Edge.

Tipo: marca temporal

Obligatorio: no

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

SyncStatus

El estado de sincronización actual de la configuración perimetral de la transmisión.

Tipo: cadena

Valores válidos: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LocalSizeConfig

Servicio: Amazon Kinesis Video Streams

Los detalles de configuración, que incluyen el tamaño máximo del medio (`MaxLocalMediaSizeInMB`) que desea almacenar para una transmisión en el Edge Agent, así como la estrategia que se debe utilizar (`StrategyOnFullSize`) cuando se alcance el tamaño máximo de una transmisión.

Contenido

MaxLocalMediaSizeInMB

El tamaño máximo total del contenido multimedia que desea almacenar para una transmisión en el Edge Agent.

Tipo: entero

Rango válido: valor mínimo de 64. Valor máximo de 2000000.

Obligatorio: no

StrategyOnFullSize

La estrategia que se debe seguir cuando se alcanza el `MaxLocalMediaSizeInMB` límite de una transmisión.

Tipo: cadena

Valores válidos: `DELETE_OLDEST_MEDIA | DENY_NEW_MEDIA`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MappedResourceConfigurationListItem

Servicio: Amazon Kinesis Video Streams

Estructura que encapsula o contiene las propiedades de configuración del almacenamiento multimedia.

Contenido

ARN

El nombre del recurso de Amazon (ARN) del recurso Kinesis Video Stream, asociado a la transmisión.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

Type

El tipo de recurso asociado a la transmisión de vídeo de Kinesis.

Tipo: cadena

Requerido: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MediaSourceConfig

Servicio: Amazon Kinesis Video Streams

Los detalles de configuración, que consisten en las credenciales necesarias (`MediaUriSecretArnyMediaUriType`) para acceder a los archivos multimedia que se transmiten a la cámara.

Contenido

MediaUriSecretArn

El AWS ARN de Secrets Manager para el nombre de usuario y la contraseña de la cámara o la ubicación de un archivo multimedia local.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 20. La longitud máxima es de 2048 caracteres.

Patrón: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

Obligatorio: sí

MediaUriType

El tipo de identificador uniforme de recursos (URI). El `FILE_URI` valor se puede usar para transmitir archivos multimedia locales.

> **Note**
>
> La vista previa solo admite el formato URI de la fuente `RTSP_URI` multimedia.

Tipo: cadena

Valores válidos: `RTSP_URI | FILE_URI`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

MediaStorageConfiguration

Servicio: Amazon Kinesis Video Streams

Estructura que encapsula o contiene las propiedades de configuración del almacenamiento multimedia.

- Si `StorageStatus` está habilitada, los datos se almacenarán en la ubicación `StreamARN` proporcionada. Para que la ingestión de WebRTC funcione, la transmisión debe tener habilitada la retención de datos.
- Si `StorageStatus` está deshabilitado, no se almacenará ningún dato y el `StreamARN` parámetro no será necesario.

Contenido

Status

El estado de la configuración de almacenamiento multimedia.

Tipo: cadena

Valores válidos: `ENABLED | DISABLED`

Obligatorio: sí

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

NotificationConfiguration

Servicio: Amazon Kinesis Video Streams

Utilice esta API para configurar las notificaciones del Amazon Simple Notification Service (Amazon SNS) para cuando los fragmentos estén disponibles en una transmisión. Si este parámetro es nulo, la configuración se eliminará de la transmisión.

Consulte Notificaciones en Kinesis Video Streams para obtener más información.

Contenido

DestinationConfig

La información de destino necesaria para entregar una notificación a un cliente.

Tipo: objeto NotificationDestinationConfig

Obligatorio: sí

Status

Indica si una configuración de notificaciones está habilitada o deshabilitada.

Tipo: cadena

Valores válidos: `ENABLED | DISABLED`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

NotificationDestinationConfig

Servicio: Amazon Kinesis Video Streams

Estructura que contiene la información necesaria para entregar una notificación a un cliente.

Contenido

Uri

El identificador uniforme de recursos (URI) que identifica dónde se entregarán las imágenes.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 255 caracteres.

Patrón: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

RecorderConfig

Servicio: Amazon Kinesis Video Streams

La configuración de la grabadora consta de los `MediaSourceConfig` detalles locales que se utilizan como credenciales para acceder a los archivos multimedia locales transmitidos por la cámara.

Contenido

MediaSourceConfig

Los detalles de configuración, que consisten en las credenciales necesarias (`MediaUriSecretArnyMediaUriType`) para acceder a los archivos multimedia transmitidos a la cámara.

Tipo: objeto MediaSourceConfig

Obligatorio: sí

ScheduleConfig

La configuración que consiste en la `ScheduleExpression` programación de la grabación desde una cámara, o un archivo multimedia local, en el Edge Agent y los `DurationInMinutes` detalles que la especifican. Si no se proporciona el `ScheduleExpression` atributo, el agente Edge siempre estará configurado en modo de grabación.

Tipo: objeto ScheduleConfig

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ResourceEndpointListItem

Servicio: Amazon Kinesis Video Streams

Objeto que describe el punto final del canal de señalización devuelto por la `GetSignalingChannelEndpoint` API.

El punto final del servidor multimedia se corresponderá con el `WEBRTC` Protocolo.

Contenido

Protocol

El protocolo del canal de señalización devuelto por la `GetSignalingChannelEndpoint` API.

Tipo: cadena

Valores válidos: `WSS | HTTPS | WEBRTC`

Obligatorio: no

ResourceEndpoint

El punto final del canal de señalización devuelto por la `GetSignalingChannelEndpoint` API.

Tipo: cadena

Requerido: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ScheduleConfig

Servicio: Amazon Kinesis Video Streams

Esta API le permite especificar el tiempo que la cámara, o el archivo multimedia local, debe grabar en el Edge Agent. Se `ScheduleConfig` compone de los atributos `ScheduleExpression` y los `DurationInMinutes` atributos.

Si no `ScheduleConfig` se proporciona en el`RecorderConfig`, el Edge Agent siempre estará configurado en modo de grabación.

Si no `ScheduleConfig` aparece en el`UploaderConfig`, el Edge Agent lo cargará a intervalos regulares (cada 1 hora).

Contenido

DurationInSeconds

La duración total de la grabación del contenido multimedia. Si se proporciona el `ScheduleExpression` atributo, también se debe especificar el `DurationInSeconds` atributo.

Tipo: entero

Rango válido: valor mínimo de 60. Valor máximo de 3600.

Obligatorio: sí

ScheduleExpression

La expresión cron de Quartz que se encarga de programar los trabajos para grabarlos desde la cámara, o un archivo multimedia local, en el Edge Agent. Si no `ScheduleExpression` se proporciona para el`RecorderConfig`, el Edge Agent siempre estará configurado en modo de grabación.

Para obtener más información sobre Quartz, consulte la página de tutoriales de Cron Trigger para comprender las expresiones válidas y su uso.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 11. La longitud máxima es de 100 caracteres.

Patrón: `[^\n]{11,100}`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SingleMasterChannelEndpointConfiguration

Servicio: Amazon Kinesis Video Streams

Un objeto que contiene la configuración del punto final para el tipo de `SINGLE_MASTER` canal.

Contenido

Protocols

Esta propiedad se utiliza para determinar la naturaleza de la comunicación a través de este canal `SINGLE_MASTER` de señalización. Si `WSS` se especifica, esta API devuelve un punto final de websocket. Si `HTTPS` se especifica, esta API devuelve un `HTTPS` punto final.

Tipo: matriz de cadenas

Miembros de la matriz: número mínimo de 1 elemento. La cantidad máxima es de 5 artículos.

Valores válidos: `WSS | HTTPS | WEBRTC`

Obligatorio: no

Role

Esta propiedad se usa para determinar los permisos de mensajería en este canal `SINGLE_MASTER` de señalización. Si `MASTER` se especifica, esta API devuelve un punto final que un cliente puede usar para recibir ofertas y enviar respuestas a cualquiera de los espectadores de este canal de señalización. Si `VIEWER` se especifica, esta API devuelve un punto final que un cliente solo puede usar para enviar ofertas a otro `MASTER` cliente en este canal de señalización.

Tipo: cadena

Valores válidos: `MASTER | VIEWER`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2

- AWS SDK para Ruby V3

SingleMasterConfiguration

Servicio: Amazon Kinesis Video Streams

Estructura que contiene la configuración del tipo `SINGLE_MASTER` de canal.

Contenido

MessageTtlSeconds

El período de tiempo (en segundos) que un canal de señalización retiene los mensajes no entregados antes de descartarlos. Se utiliza UpdateSignalingChannel para actualizar este valor.

Tipo: entero

Rango válido: valor mínimo de 5. Valor máximo de 120.

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

StreamInfo

Servicio: Amazon Kinesis Video Streams

Objeto que describe una transmisión de vídeo de Kinesis.

Contenido

CreationTime

Una marca de tiempo que indica cuándo se creó la transmisión.

Tipo: marca temporal

Obligatorio: no

DataRetentionInHours

Cuánto tiempo retiene los datos la transmisión, en horas.

Tipo: entero

Rango válido: valor mínimo de 0.

Obligatorio: no

DeviceName

El nombre del dispositivo asociado a la transmisión.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

KmsKeyId

El ID de la clave AWS Key Management Service (AWS KMS) que Kinesis Video Streams utiliza para cifrar los datos de la transmisión.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 2048 caracteres.

Patrón: `.+`

Obligatorio: no

MediaType

El `MediaType` de la transmisión.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Obligatorio: no

Status

El estado de la transmisión.

Tipo: cadena

Valores válidos: `CREATING | ACTIVE | UPDATING | DELETING`

Obligatorio: no

StreamARN

El nombre de recurso de Amazon (ARN) del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 1024 caracteres.

Patrón: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obligatorio: no

StreamName

El nombre del flujo de .

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Version

La versión del flujo.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 64.

Patrón: `[a-zA-Z0-9]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

StreamNameCondition

Servicio: Amazon Kinesis Video Streams

Especifica la condición que deben cumplir las transmisiones para que se devuelvan cuando publiques transmisiones (consulta la `ListStreams` API). Una condición tiene una operación de comparación y un valor. Actualmente, solo puede especificar el `BEGINS_WITH` operador, que busca secuencias cuyos nombres comiencen con un prefijo determinado.

Contenido

ComparisonOperator

Un operador de comparación. Actualmente, solo puede especificar el `BEGINS_WITH` operador, que busca secuencias cuyos nombres comiencen con un prefijo determinado.

Tipo: cadena

Valores válidos: `BEGINS_WITH`

Obligatorio: no

ComparisonValue

Un valor para comparar.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Tag

Servicio: Amazon Kinesis Video Streams

Un par de clave y valor que está asociado al canal de señalización especificado.

Contenido

Key

La clave de la etiqueta que está asociada al canal de señalización especificado.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Obligatorio: sí

Value

El valor de la etiqueta que está asociada al canal de señalización especificado.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 256 caracteres.

Patrón: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

UploaderConfig

Servicio: Amazon Kinesis Video Streams

La configuración que consta de la programación `ScheduleExpression` y los `DurationInMinutes` detalles que especifican la programación para grabar desde una cámara, o un archivo multimedia local, en el Edge Agent. Si no `ScheduleConfig` se proporciona en el`UploaderConfig`, el agente perimetral se cargará a intervalos regulares (cada 1 hora).

Contenido

ScheduleConfig

La configuración que consta de la programación `ScheduleExpression` y los `DurationInMinutes` detalles que especifican la programación para grabar desde una cámara, o un archivo multimedia local, en el Edge Agent. Si esto no `ScheduleConfig` se proporciona`UploaderConfig`, el agente perimetral se cargará a intervalos regulares (cada 1 hora).

Tipo: objeto ScheduleConfig

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Contenido multimedia de Amazon Kinesis Video Streams

Amazon Kinesis Video Streams Media admite los siguientes tipos de datos:

- StartSelector

StartSelector

Servicio: Amazon Kinesis Video Streams Media

Identifica el fragmento de la transmisión de vídeo de Kinesis en el que desea que `GetMedia` la API comience a devolver datos multimedia. Dispone de las siguientes opciones para identificar el fragmento inicial:

- Elige el fragmento más reciente (o el más antiguo).
- Identifica un fragmento específico. Puede identificar un fragmento específico proporcionando un número de fragmento o una marca de tiempo (servidor o productor).
- Los metadatos de cada fragmento incluyen un token de continuación en forma de etiqueta Matroska (MKV) (). `AWS_KINESISVIDEO_CONTINUATION_TOKEN` Si tu `GetMedia` solicitud anterior ha finalizado, puedes usar este valor de etiqueta en la siguiente solicitud. `GetMedia` A continuación, la API comienza a devolver fragmentos empezando por donde terminó la última API.

Contenido

StartSelectorType

Identifica el fragmento de la transmisión de vídeo de Kinesis desde el que desea empezar a obtener los datos.

- AHORA: comience con la última parte de la transmisión.
- MÁS TEMPRANO: comienza con el fragmento más antiguo disponible de la transmisión.
- FRAGMENT_NUMBER: comienza con el fragmento que sigue a un fragmento específico. También debe especificar el parámetro `AfterFragmentNumber`.
- PRODUCER_TIMESTAMP o SERVER_TIMESTAMP: comience con el fragmento que contiene un fragmento con la marca de tiempo del productor o servidor especificada. Para especificar la marca `StartTimestamp` de tiempo, añada.
- CONTINUATION_TOKEN: se lee con el token de continuación especificado.

> **Note**
>
> Si eliges NOW, EARLY o CONTINUATION_TOKEN como valores`startSelectorType`, no proporcionarás ninguna información adicional en el. `startSelector`

Tipo: cadena

Valores válidos: `FRAGMENT_NUMBER | SERVER_TIMESTAMP | PRODUCER_TIMESTAMP | NOW | EARLIEST | CONTINUATION_TOKEN`

Obligatorio: sí

AfterFragmentNumber

Especifica el número de fragmento desde el que quieres que la `GetMedia` API comience a devolver los fragmentos.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[0-9]+$`

Obligatorio: no

ContinuationToken

Símbolo de continuación que Kinesis Video Streams devolvió en la respuesta `GetMedia` anterior. A continuación, la `GetMedia` API comienza con el fragmento identificado por el token de continuación.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[a-zA-Z0-9_\.\-]+$`

Obligatorio: no

StartTimestamp

Un valor de marca temporal. Este valor es obligatorio si elige PRODUCER_TIMESTAMP o SERVER_TIMESTAMP como. `startSelectorType` A continuación, la `GetMedia` API comienza con el fragmento que contiene el fragmento que tiene la marca de tiempo especificada.

Tipo: marca temporal

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Contenido multimedia archivado de Amazon Kinesis Video Streams

Los medios archivados de Amazon Kinesis Video Streams admiten los siguientes tipos de datos:

- ClipFragmentSelector
- ClipTimestampRange
- DASHFragmentSelector
- DASHTimestampRange
- Fragment
- FragmentSelector
- HLSFragmentSelector
- HLSTimestampRange
- Image
- TimestampRange

ClipFragmentSelector

Servicio: Amazon Kinesis Video Streams Archived Media

Describe el rango de marcas de tiempo y el origen de las marcas de tiempo de un rango de fragmentos.

Los fragmentos que tienen marcas de tiempo del productor duplicadas se deduplican. Esto significa que si los productores producen un flujo de fragmentos con marcas de tiempo del productor que son aproximadamente iguales a la hora real del reloj, el clip contendrá todos los fragmentos dentro del rango de marcas de tiempo solicitado. Si algunos fragmentos se ingieren dentro del mismo intervalo de tiempo y en momentos muy diferentes, solo se devuelve la colección de fragmentos ingerida más antigua.

Contenido

FragmentSelectorType

El origen de las marcas de tiempo que se van a utilizar (servidor o productor).

Tipo: cadena

Valores válidos: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Obligatorio: sí

TimestampRange

El rango de marcas de tiempo que se van a devolver.

Tipo: objeto ClipTimestampRange

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ClipTimestampRange

Servicio: Amazon Kinesis Video Streams Archived Media

El rango de marcas de tiempo para las que se devuelven los fragmentos.

Contenido

EndTimestamp

El final del intervalo de marcas de tiempo del medio solicitado.

Este valor debe estar dentro de las 24 horas siguientes al especificado `StartTimestamp` y debe ser posterior a ese valor. `StartTimestamp` Si la `FragmentSelectorType` solicitud es`SERVER_TIMESTAMP`, este valor debe estar en el pasado.

Este valor es inclusivo. `EndTimestamp`Se compara con la marca de tiempo (inicial) del fragmento. Los fragmentos que comienzan antes del `EndTimestamp` valor y continúan después de él se incluyen en la sesión.

Tipo: marca temporal

Obligatorio: sí

StartTimestamp

La marca de tiempo inicial en el rango de marcas de tiempo para las que se van a devolver los fragmentos.

Solo se incluyen en la sesión los fragmentos que comiencen exactamente en o después`StartTimestamp`. Los fragmentos que comiencen antes `StartTimestamp` y continúen después no se incluyen en la sesión. Si `FragmentSelectorType` es `SERVER_TIMESTAMP` así, `StartTimestamp` debe ser posterior al encabezado de la transmisión.

Tipo: marca temporal

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DASHFragmentSelector

Servicio: Amazon Kinesis Video Streams Archived Media

Contiene el rango de marcas de tiempo del medio solicitado y el origen de las marcas de tiempo.

Contenido

FragmentSelectorType

El origen de las marcas de tiempo del medio solicitado.

Cuando `FragmentSelectorType` se establece en Get DASHStreaming sessionUrl: PlaybackMode is `ON_DEMAND` o`LIVE_REPLAY`, el primer fragmento ingerido con una marca de tiempo del productor dentro del valor especificado FragmentSelector: TimestampRange se incluye en la lista de reproducción multimedia. `PRODUCER_TIMESTAMP` Además, se incluyen los fragmentos con las marcas de tiempo del productor dentro del fragmento `TimestampRange` ingerido inmediatamente después del primer fragmento (hasta el valor Get SessionURL:). DASHStreaming MaxManifestFragmentResults

Los fragmentos que tienen marcas de tiempo del productor duplicadas se deduplican. Esto significa que si los productores producen un flujo de fragmentos con marcas de tiempo del productor que son aproximadamente iguales a la hora real del reloj, el manifiesto MPEG-DASH contendrá todos los fragmentos dentro del rango de marcas de tiempo solicitado. Si algunos fragmentos se ingieren dentro del mismo intervalo de tiempo y en momentos muy diferentes, solo se devuelve la colección de fragmentos ingerida más antigua.

Cuando `FragmentSelectorType` se establece en Get DASHStreaming SessionUrl: PlaybackMode es decir`LIVE`, las marcas de tiempo del productor se utilizan en los fragmentos y para la deduplicación. `PRODUCER_TIMESTAMP` MP4 Sin embargo, los fragmentos ingeridos más recientemente en función de las marcas de tiempo del servidor se incluyen en el manifiesto MPEG-DASH. Esto significa que, aunque los fragmentos ingeridos en el pasado tengan marcas de tiempo del productor con valores actuales, no se incluyen en la lista de reproducción multimedia del HLS.

El valor predeterminado es `SERVER_TIMESTAMP`.

Tipo: cadena

Valores válidos: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Obligatorio: no

TimestampRange

El inicio y el final del intervalo de marcas de tiempo del contenido multimedia solicitado.

Este valor no debería estar presente si `PlaybackType` lo está. `LIVE`

Tipo: objeto DASHTimestampRange

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

DASHTimestampRange

Servicio: Amazon Kinesis Video Streams Archived Media

El inicio y el final del intervalo de marcas de tiempo del medio solicitado.

Este valor no debería estar presente si `PlaybackType` lo está. `LIVE`

Los valores incluidos `DASHimestampRange` son inclusivos. Los fragmentos que comiencen exactamente a la hora de inicio o después se incluyen en la sesión. Los fragmentos que comiencen antes de la hora de inicio y continúen después de esa hora no se incluyen en la sesión.

Contenido

EndTimestamp

El final del intervalo de fecha y hora del contenido multimedia solicitado. Este valor debe estar dentro de las 24 horas siguientes al especificado `StartTimestamp` y debe ser posterior a ese valor. `StartTimestamp`

Si la `FragmentSelectorType` solicitud es`SERVER_TIMESTAMP`, este valor debe estar en el pasado.

El `EndTimestamp` valor es obligatorio para el `ON_DEMAND` modo, pero opcional para `LIVE_REPLAY` el modo. Si no `EndTimestamp` está configurado para el `LIVE_REPLAY` modo, la sesión seguirá incluyendo los fragmentos recién ingeridos hasta que la sesión caduque.

> **Note**
>
> Este valor es inclusivo. `EndTimestamp`Se compara con la marca de tiempo (inicial) del fragmento. Los fragmentos que comienzan antes del `EndTimestamp` valor y continúan después de él se incluyen en la sesión.

Tipo: marca temporal

Obligatorio: no

StartTimestamp

El inicio del intervalo de marcas de tiempo del contenido multimedia solicitado.

Si se especifica el `DASHTimestampRange` valor, es `StartTimestamp` obligatorio.

Solo se incluyen en la sesión los fragmentos que comiencen exactamente en o después`StartTimestamp`. Los fragmentos que comiencen antes `StartTimestamp` y continúen después no se incluyen en la sesión. Si `FragmentSelectorType` es `SERVER_TIMESTAMP` así, `StartTimestamp` debe ser posterior al encabezado de la transmisión.

Tipo: marca temporal

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Fragment

Servicio: Amazon Kinesis Video Streams Archived Media

Representa un segmento de vídeo u otros datos delimitados por el tiempo.

Contenido

FragmentLengthInMilliseconds

La duración de la reproducción u otro valor de tiempo asociado al fragmento.

Tipo: largo

Obligatorio: no

FragmentNumber

El identificador único del fragmento. Este valor aumenta monótonamente en función del orden de ingesta.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 128.

Patrón: `^[0-9]+$`

Obligatorio: no

FragmentSizeInBytes

El tamaño total del fragmento, incluida la información sobre el fragmento y los datos multimedia contenidos.

Tipo: largo

Obligatorio: no

ProducerTimestamp

La marca de tiempo del productor correspondiente al fragmento, en milisegundos.

Tipo: marca temporal

Obligatorio: no

ServerTimestamp

La marca de tiempo del AWS servidor correspondiente al fragmento, en milisegundos.

Tipo: marca temporal

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

FragmentSelector

Servicio: Amazon Kinesis Video Streams Archived Media

Describe el rango de marcas de tiempo y el origen de las marcas de tiempo de un rango de fragmentos.

Solo se devuelven los fragmentos con una marca de tiempo de inicio superior o igual a la hora de inicio determinada e inferior o igual a la hora de finalización. Por ejemplo, si una transmisión contiene fragmentos con las siguientes marcas de tiempo de inicio:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

Un rango de selección de fragmentos con una hora de inicio 00:00:01 y una hora de finalización de 00:00:04 devolvería los fragmentos con una hora de inicio de 00:00:02 y 00:00:04.

Contenido

FragmentSelectorType

El origen de las marcas de tiempo que se van a utilizar (servidor o productor).

Tipo: cadena

Valores válidos: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Obligatorio: sí

TimestampRange

El rango de marcas de tiempo que se van a devolver.

Tipo: objeto TimestampRange

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

HLSFragmentSelector

Servicio: Amazon Kinesis Video Streams Archived Media

Contiene el rango de marcas de tiempo del medio solicitado y el origen de las marcas de tiempo.

Contenido

FragmentSelectorType

El origen de las marcas de tiempo del medio solicitado.

Cuando `FragmentSelectorType` se establece en Get HLSStreaming sessionUrl: PlaybackMode is `ON_DEMAND` o`LIVE_REPLAY`, el primer fragmento ingerido con una marca de tiempo del productor dentro del valor especificado FragmentSelector: TimestampRange se incluye en la lista de reproducción multimedia. `PRODUCER_TIMESTAMP` Además, se incluyen los fragmentos con las marcas de tiempo del productor dentro del fragmento `TimestampRange` ingerido inmediatamente después del primer fragmento (hasta el valor Get SessionURL:). HLSStreaming MaxMediaPlaylistFragmentResults

Los fragmentos que tienen marcas de tiempo del productor duplicadas se deduplican. Esto significa que si los productores producen un flujo de fragmentos con marcas de tiempo del productor que son aproximadamente iguales a la hora real del reloj, las listas de reproducción multimedia del HLS contendrán todos los fragmentos dentro del intervalo de marcas de tiempo solicitado. Si algunos fragmentos se ingieren dentro del mismo intervalo de tiempo y en momentos muy diferentes, solo se devuelve la colección de fragmentos ingerida más antigua.

Cuando `FragmentSelectorType` se establece en Get HLSStreaming SessionUrl: PlaybackMode es decir`LIVE`, las marcas de tiempo del productor se utilizan en los fragmentos y para la deduplicación. `PRODUCER_TIMESTAMP` MP4 Sin embargo, los fragmentos ingeridos más recientemente en función de las marcas de tiempo del servidor se incluyen en la lista de reproducción multimedia del HLS. Esto significa que, aunque los fragmentos ingeridos en el pasado tengan marcas de tiempo del productor con valores actuales, no se incluyen en la lista de reproducción multimedia del HLS.

El valor predeterminado es `SERVER_TIMESTAMP`.

Tipo: cadena

Valores válidos: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Obligatorio: no

TimestampRange

El inicio y el final del intervalo de marcas de tiempo del contenido multimedia solicitado.

Este valor no debería estar presente si `PlaybackType` lo está. `LIVE`

Tipo: objeto HLSTimestampRange

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

HLSTimestampRange

Servicio: Amazon Kinesis Video Streams Archived Media

El inicio y el final del intervalo de marcas de tiempo del medio solicitado.

Este valor no debería estar presente si `PlaybackType` lo está. `LIVE`

Contenido

EndTimestamp

El final del intervalo de marcas de tiempo del medio solicitado. Este valor debe estar dentro de las 24 horas siguientes al especificado `StartTimestamp` y debe ser posterior a ese valor. `StartTimestamp`

Si la `FragmentSelectorType` solicitud es`SERVER_TIMESTAMP`, este valor debe estar en el pasado.

El `EndTimestamp` valor es obligatorio para el `ON_DEMAND` modo, pero opcional para `LIVE_REPLAY` el modo. Si no `EndTimestamp` está configurado para el `LIVE_REPLAY` modo, la sesión seguirá incluyendo los fragmentos recién ingeridos hasta que la sesión caduque.

> **Note**
>
> Este valor es inclusivo. `EndTimestamp`Se compara con la marca de tiempo (inicial) del fragmento. Los fragmentos que comienzan antes del `EndTimestamp` valor y continúan después de él se incluyen en la sesión.

Tipo: marca temporal

Obligatorio: no

StartTimestamp

El inicio del intervalo de marcas de tiempo del contenido multimedia solicitado.

Si se especifica el `HLSTimestampRange` valor, es `StartTimestamp` obligatorio.

Solo se incluyen en la sesión los fragmentos que comiencen exactamente en o después`StartTimestamp`. Los fragmentos que comiencen antes `StartTimestamp` y continúen

después no se incluyen en la sesión. Si `FragmentSelectorType` es `SERVER_TIMESTAMP` así, `StartTimestamp` debe ser posterior al encabezado de la transmisión.

Tipo: marca temporal

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Image

Servicio: Amazon Kinesis Video Streams Archived Media

Una estructura que contiene las `TimestampError`, y`ImageContent`.

Contenido

Error

El mensaje de error que se muestra cuando no se ha extraído la imagen de la marca de tiempo proporcionada debido a un error que no se puede probar. Se devolverá un error si:

- No existe ningún medio para lo especificado`Timestamp`.
- El soporte del tiempo especificado no permite extraer una imagen. En este caso, el contenido multimedia es solo de audio o se ha ingerido el contenido incorrecto.

Tipo: cadena

Valores válidos: `NO_MEDIA | MEDIA_ERROR`

Obligatorio: no

ImageContent

Atributo del `Image` objeto codificado en Base64.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. Longitud máxima de 6291456.

Obligatorio: no

TimeStamp

Atributo del `Image` objeto que se utiliza para extraer una imagen de la transmisión de vídeo. Este campo se utiliza para gestionar los huecos en las imágenes o para comprender mejor la ventana de paginación.

Tipo: marca temporal

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

TimestampRange

Servicio: Amazon Kinesis Video Streams Archived Media

El rango de marcas de tiempo para las que se devuelven los fragmentos.

Contenido

EndTimestamp

La marca de tiempo final del rango de marcas de tiempo para las que se van a devolver los fragmentos.

Tipo: marca temporal

Obligatorio: sí

StartTimestamp

La marca de tiempo inicial del rango de marcas de tiempo para las que se van a devolver los fragmentos.

Tipo: marca temporal

Obligatorio: sí

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas AWS SDKs específicos, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Canales de señalización de vídeo de Amazon Kinesis

Los canales de señalización de vídeo de Amazon Kinesis admiten los siguientes tipos de datos:

- IceServer

IceServer

Servicio: Amazon Kinesis Video Signaling Channels

Una estructura para los datos de conexión del servidor ICE.

Contenido

Password

Una contraseña para iniciar sesión en el servidor ICE.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Ttl

El período de tiempo, en segundos, durante la cual son válidos el nombre de usuario y la contraseña.

Tipo: entero

Rango válido: valor fijo de 300.

Obligatorio: no

Uris

Matriz de URIs, en la forma especificada en el I-D. petithuguenin-behave-turn-urisespecificación. Estos URIs proporcionan las diferentes direcciones y/o protocolos que se pueden utilizar para llegar al servidor TURN.

Tipo: matriz de cadenas

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Obligatorio: no

Username

Un nombre de usuario para iniciar sesión en el servidor ICE.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 1. La longitud máxima es de 256 caracteres.

Patrón: `[a-zA-Z0-9_.-]+`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en un lenguaje específico AWS SDKs, consulte lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Almacenamiento WebRTC de vídeo en Amazon Kinesis

El almacenamiento WebRTC de vídeo de Amazon Kinesis admite los siguientes tipos de datos:

Errores comunes

En esta sección se enumeran los errores comunes a las acciones de la API de todos los AWS servicios. En el caso de los errores específicos de una acción de la API de este servicio, consulte el tema de dicha acción de la API.

AccessDeniedException

No tiene acceso suficiente para realizar esta acción.

Código de estado HTTP: 400

IncompleteSignature

La firma de la solicitud no se ajusta a AWS los estándares.

Código de estado HTTP: 400

InternalFailure

El procesamiento de la solicitud ha devuelto un error debido a un error o una excepción desconocidos.

Código de estado HTTP: 500

InvalidAction

La acción u operación solicitada no es válida. Compruebe que la acción se ha escrito correctamente.

Código de estado HTTP: 400

InvalidClientTokenId

El identificador de clave de AWS acceso o certificado X.509 proporcionado no existe en nuestros registros.

Código de estado HTTP: 403

NotAuthorized

No tiene permiso para realizar esta acción.

Código de estado HTTP: 400

OptInRequired

El identificador de clave de AWS acceso necesita una suscripción al servicio.

Código de estado HTTP: 403

RequestExpired

La solicitud llegó al servicio más de 15 minutos después del sello de fecha de la solicitud o más de 15 minutos después de la fecha de caducidad de la solicitud (por ejemplo, en el caso de los prefirmados URLs), o el sello de fecha de la solicitud es más de 15 minutos en el futuro.

Código de estado HTTP: 400

ServiceUnavailable

La solicitud no se ha ejecutado correctamente debido a un error temporal del servidor.

Código de estado HTTP: 503

ThrottlingException

La solicitud fue denegada debido a una limitación de la solicitud.

Código de estado HTTP: 400

ValidationError

La entrada no cumple las restricciones especificadas por un AWS servicio.

Código de estado HTTP: 400

Parámetros comunes

La siguiente lista contiene los parámetros que utilizan todas las acciones para firmar solicitudes de Signature Version 4 con una cadena de consulta. Los parámetros específicos de acción se enumeran en el tema correspondiente a la acción. Para obtener más información sobre la versión 4 de Signature, consulte Firmar solicitudes de AWS API en la Guía del usuario de IAM.

Action

Las acciones que se van a realizar.

Tipo: cadena

Obligatorio: sí

Version

La versión de API para la que está escrita la solicitud, expresada en el formato YYYY-MM-DD.

Tipo: cadena

Obligatorio: sí

X-Amz-Algorithm

El algoritmo de hash que utilizó para crear la solicitud de firma.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Valores válidos: `AWS4-HMAC-SHA256`

Obligatorio: condicional

X-Amz-Credential

El valor del ámbito de la credencial, que es una cadena que incluye la clave de acceso, la fecha, la región a la que se dirige, el servicio que solicita y una cadena de terminación (“aws4_request”). El valor se expresa en el siguiente formato: access_key/AAAAMMDD/region/service/aws4_request.

Para obtener más información, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-Date

La fecha utilizada para crear la firma. El formato debe ser ISO 8601 formato básico (AAAAMMDD‘T’HHMMSS‘Z’). Por ejemplo, la siguiente fecha y hora es un X-Amz-Date valor válido:`20120325T120000Z`.

Condición: X-Amz-Date es opcional en todas las solicitudes; puede utilizarse para anular la fecha empleada para firmar solicitudes. Si el encabezado de fecha se especifica en el formato básico ISO 8601, no X-Amz-Date es obligatorio. Cuando X-Amz-Date se usa, siempre anula el valor del encabezado de fecha. Para obtener más información, consulte Elementos de la firma de una solicitud de AWS API en la Guía del usuario de IAM.

Tipo: cadena

Obligatorio: condicional

X-Amz-Security-Token

El token de seguridad temporal que se obtuvo mediante una llamada a AWS Security Token Service (AWS STS). Para obtener una lista de servicios compatibles con las credenciales de seguridad temporales de AWS STS, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Condición: si utilizas credenciales de seguridad temporales de AWS STS, debes incluir el token de seguridad.

Tipo: cadena

Obligatorio: condicional

X-Amz-Signature

Especifica la firma codificada hexadecimal que se calculó a partir de la cadena que se va a firmar y la clave de firma derivada.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-SignedHeaders

Especifica todos los encabezados HTTP que se incluyeron como parte de la solicitud canónica. Para obtener más información sobre cómo especificar encabezados firmados, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

Seguridad en Amazon Kinesis Video Streams

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, se beneficiará de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad que se aplican a Kinesis Video Streams, consulte Servicios de AWS en el ámbito del programa de conformidad..
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar Kinesis Video Streams. En los temas siguientes se muestra cómo configurar Kinesis Video Streams para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que pueden ayudarle a supervisar y proteger sus recursos de Kinesis Video Streams.

Temas

- Protección de datos en Kinesis Video Streams
- Control del acceso a los recursos de Kinesis Video Streams mediante IAM
- Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT
- Validación de conformidad para Amazon Kinesis Video Streams
- Resiliencia en Amazon Kinesis Video Streams
- Seguridad de infraestructura en Kinesis Video Streams
- Mejores prácticas de seguridad para Kinesis Video Streams

Protección de datos en Kinesis Video Streams

Puede utilizar el cifrado del lado del servidor (SSE) mediante claves AWS Key Management Service (AWS KMS) para cumplir con los estrictos requisitos de administración de datos cifrando los datos en reposo en Amazon Kinesis Video Streams.

Temas

- ¿Qué es el cifrado del lado del servidor para Kinesis Video Streams?
- Cuestiones sobre costos, regiones y rendimiento
- ¿Cómo puedo comenzar a usar el cifrado en el servidor?
- Creación y uso de una clave gestionada por el cliente
- Permisos para usar una clave administrada por el cliente

¿Qué es el cifrado del lado del servidor para Kinesis Video Streams?

El cifrado del lado del servidor es una función de Kinesis Video Streams que cifra automáticamente los datos antes de almacenarlos en reposo mediante una AWS KMS clave que usted especifique. Los datos se cifran antes de escribirse en la capa de almacenamiento de transmisiones de Kinesis Video Streams y se descifran una vez recuperados del almacenamiento. Como resultado, sus datos siempre se cifran en reposo en el servicio Kinesis Video Streams.

Con el cifrado del lado del servidor, los productores y consumidores de transmisiones de vídeo de Kinesis no necesitan gestionar las claves de KMS ni las operaciones criptográficas. Si la retención de datos está habilitada, los datos se cifran automáticamente al entrar y salir de Kinesis Video Streams, por lo que los datos en reposo se cifran. AWS KMS proporciona todas las claves que utiliza la función de cifrado del lado del servidor. AWS KMS optimiza el uso de una clave KMS para Kinesis Video Streams que se administra AWS mediante una clave AWS KMS especificada por el usuario importada al servicio. AWS KMS

Cuestiones sobre costos, regiones y rendimiento

Al aplicar el cifrado del lado del servidor, está sujeto al uso de la AWS KMS API y a los costes de las claves. A diferencia de AWS KMS las claves personalizadas, la clave `aws/kinesisvideo` KMS predeterminada se ofrece de forma gratuita. Sin embargo, debe seguir pagando en su nombre los costes de uso de la API en los que incurra Kinesis Video Streams.

Los costes de uso de la API se aplican a todas las claves de KMS, incluidas las personalizadas. Los AWS KMS costes aumentan en función del número de credenciales de usuario que utilice para generar y consumir datos, ya que cada credencial de usuario requiere una llamada a AWS KMS la API única.

A continuación se describen los costos por recurso:

Claves

- La clave KMS para Kinesis Video Streams gestionada AWS por (alias `aws/kinesisvideo` =) es gratuita.
- Las claves KMS generadas por los usuarios están sujetas a costes AWS KMS key . Para obtener más información, consulte AWS Key Management Service Precios.

AWS KMS Uso de API

Las solicitudes de la API para generar nuevas claves de cifrado de datos o recuperar las claves de cifrado existentes aumentan a medida que aumenta el tráfico y están sujetas a los costes de AWS KMS uso. Para obtener más información, consulte AWS Key Management Service Precios: uso.

Kinesis Video Streams genera solicitudes clave incluso cuando la retención está establecida en 0 (sin retención).

Disponibilidad del cifrado en el servidor por región

El cifrado del lado del servidor de las transmisiones de vídeo de Kinesis está disponible en todos los sitios donde esté disponible Kinesis Regiones de AWS Video Streams.

¿Cómo puedo comenzar a usar el cifrado en el servidor?

El cifrado del lado del servidor siempre está activado en Kinesis Video Streams. Si no se especifica una clave proporcionada por el usuario al crear la transmisión, se utiliza la Clave administrada de AWS (proporcionada por Kinesis Video Streams).

Se debe asignar una clave KMS proporcionada por el usuario a una transmisión de vídeo de Kinesis al crearla. No podrá asignar una clave diferente a una transmisión mediante la UpdateStreamAPI más adelante.

Puede asignar una clave KMS proporcionada por el usuario a una transmisión de vídeo de Kinesis de dos maneras:

- Al crear una transmisión de vídeo de Kinesis en AWS Management Console, especifique la clave KMS en la pestaña Cifrado de la página Crear una nueva transmisión de vídeo.
- Al crear una transmisión de vídeo de Kinesis mediante la CreateStreamAPI, especifique el ID de clave en el `KmsKeyId` parámetro.

Creación y uso de una clave gestionada por el cliente

En esta sección se describe cómo crear y utilizar sus propias claves de KMS en lugar de utilizar la clave administrada por Amazon Kinesis Video Streams.

Creación de una clave administrada por el cliente

Para obtener información sobre cómo crear sus propias claves, consulte Creación de claves en la Guía para AWS Key Management Service desarrolladores. Tras crear las claves para su cuenta, el servicio Kinesis Video Streams las devuelve a la lista de claves gestionadas por el cliente.

Uso de claves administradas por el cliente

Una vez que se hayan aplicado los permisos correctos a sus consumidores, productores y administradores, podrá utilizar las claves KMS personalizadas propias Cuenta de AWS o Cuenta de AWS ajenas. Todas las claves KMS de su cuenta aparecen en la lista de claves administradas por el cliente de la consola.

Para usar claves KMS personalizadas que estén ubicadas en otra cuenta, debe tener permisos para usarlas. Asimismo, debe crear la transmisión utilizando la API `CreateStream`. No puedes usar claves de KMS de cuentas diferentes en las transmisiones creadas en la consola.

> **Note**
>
> No se accede a la clave KMS hasta que se realiza la `GetMedia` operación `PutMedia` o. Así, se obtienen los siguientes resultados:
>
> - Si la clave que especificas no existe, la `CreateStream` operación se realiza correctamente, pero `PutMedia` las `GetMedia` operaciones de la transmisión fallan.
> - Si utilizas la clave proporcionada (`aws/kinesisvideo`), la clave no estará en tu cuenta hasta que no realices la primera `PutMedia` `GetMedia` operación.

Permisos para usar una clave administrada por el cliente

Para poder usar el cifrado del lado del servidor con una clave administrada por el cliente, debe configurar las políticas de claves de KMS para permitir el cifrado de las transmisiones y el cifrado y descifrado de los registros de las transmisiones. Para ver ejemplos y más información sobre AWS KMS los permisos, consulte Permisos de AWS KMS API: referencia sobre acciones y recursos.

> **Note**
>
> El uso de la clave de servicio predeterminada para el cifrado no requiere la aplicación de permisos de IAM personalizados.

Antes de usar una clave administrada por el cliente, compruebe que los productores y consumidores de transmisiones de vídeo de Kinesis (principales de IAM) sean usuarios de la AWS KMS política de claves predeterminada. De lo contrario, las labores de escritura y lectura de una secuencia producirán un error, lo que, en definitiva, podría resultar en pérdida de datos, retrasos en el procesamiento, aplicaciones colgadas. Puede administrar los permisos para las claves de KMS con las políticas de IAM. Para obtener más información, consulte Uso de políticas de IAM con. AWS KMS

Ejemplo de permisos para productores

Los productores de transmisiones de vídeo de Kinesis deben tener el `kms:GenerateDataKey` permiso para:

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

```
        "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
    }
  ]
}
```

Ejemplo de permisos de consumo

Los consumidores de streaming de vídeo de Kinesis deben tener el `kms:Decrypt` permiso para:

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

Control del acceso a los recursos de Kinesis Video Streams mediante IAM

Puede usar AWS Identity and Access Management (IAM) con Amazon Kinesis Video Streams para controlar si los usuarios de su organización pueden realizar una tarea mediante operaciones específicas de la API de Kinesis Video Streams y si pueden usar recursos específicos. AWS

Para obtener más información sobre IAM, consulte lo siguiente:

- AWS Identity and Access Management (IAM)
- Introducción

- Guía del usuario de IAM

Contenido

- Sintaxis de la política
- Acciones para Kinesis Video Streams
- Nombres de recursos de Amazon (ARNs) para Kinesis Video Streams
- Otorgar acceso a otras cuentas de IAM a una transmisión de vídeo de Kinesis
- Ejemplos de políticas para Kinesis Video Streams

Sintaxis de la política

Una política de IAM es un documento JSON que contiene una o varias instrucciones. Cada instrucción tiene la estructura siguiente:

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

Una instrucción está compuesta por varios elementos:

- Efecto: el efecto puede ser `Allow` o`Deny`. De forma predeterminada, los usuarios no tienen permiso para utilizar los recursos y las acciones de la API, por lo que se deniegan todas las solicitudes. Si se concede un permiso explícito se anula el valor predeterminado. Una denegación explícita invalida cualquier permiso concedido.
- Acción: la acción es la acción específica de la API para la que se concede o deniega el permiso.
- Recurso: el recurso al que afecta la acción. Para especificar un recurso en la instrucción debe utilizar el nombre de recurso de Amazon (ARN).

- Condición: las condiciones son opcionales. Se pueden usar para controlar cuándo está en vigor la política.

Al crear y gestionar las políticas de IAM, le recomendamos que utilice el generador de políticas de IAM y el simulador de políticas de IAM.

Acciones para Kinesis Video Streams

En una instrucción de política de IAM, puede especificar cualquier acción de API de cualquier servicio que sea compatible con IAM. Para Kinesis Video Streams, utilice el siguiente prefijo con el nombre de la acción de la API:. `kinesisvideo:` Por ejemplo, `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams` y `kinesisvideo:DescribeStream`.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

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

También puede utilizar caracteres comodín para especificar varias acciones. Por ejemplo, puede especificar todas las acciones cuyo nombre comience por la palabra "Get" del siguiente modo:

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

Para especificar todas las operaciones de Kinesis Video Streams use el carácter comodín de asterisco (*) del siguiente modo:

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

Para obtener la lista completa de las acciones de la API de Kinesis Video Streams, consulte Referencia de API de Kinesis Video Streams.

Nombres de recursos de Amazon (ARNs) para Kinesis Video Streams

Cada declaración de política de IAM se aplica a los recursos que especifique mediante su uso. ARNs

Utilice el siguiente formato de recurso de ARN para la Kinesis Video Streams:

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

Por ejemplo:

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

Puede obtener el ARN de una transmisión usando. DescribeStream

Otorgar acceso a otras cuentas de IAM a una transmisión de vídeo de Kinesis

Puede que necesite conceder permiso a otras cuentas de IAM para realizar operaciones en las transmisiones de Kinesis Video Streams. En la siguiente información general se describen los pasos generales para conceder acceso a las transmisiones de vídeo a otras cuentas:

1. Obtenga el ID de cuenta de 12 dígitos de la cuenta a la que quiere conceder permisos para realizar operaciones en el recurso de transmisión creado en su cuenta.

   Ejemplo: En los siguientes pasos, utilizaremos 1111 como ID de cuenta de la cuenta a la que quiere conceder permiso y 9999 como ID de Kinesis Video Streams

2. Cree una política gestionada por IAM en la cuenta propietaria de la transmisión (9999) que permita el nivel de acceso que desee conceder.

   Ejemplo de política:

   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"
        }
    ]
}
```

Para ver otros ejemplos de políticas para los recursos de Kinesis Video Streams, Ejemplos de políticas consulte la siguiente sección.

3. Cree un rol en la cuenta propietaria de la transmisión (9999) y especifique la cuenta a la que quiere conceder permisos (1111). Esto añadirá una entidad de confianza a la función.

   Ejemplo de política de confianza:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111111111111:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

   Adjunta la política que creaste en el paso anterior a este rol.

   Ahora ha creado un rol en la cuenta 7139999 que tiene permiso para realizar operaciones como `DescribeStreamGetDataEndpoint`, y `PutMedia` en un ARN de recurso de transmisión en la política administrada. Este nuevo rol también confía en que la otra cuenta, 1111, asuma este rol.

   > **Important**
   >
   > Toma nota del ARN del rol, lo necesitarás en el siguiente paso.

4. Crea una política gestionada en la otra cuenta (1111) que permita realizar `AssumeRole` acciones en el rol que creaste en la cuenta 9999 en el paso anterior. Deberás mencionar la función ARN del paso anterior.

   Ejemplo de política:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
    }
}
```

5. Adjunta la política creada en el paso anterior a una entidad de IAM, como un rol o un usuario en la cuenta 1111. Este usuario ahora tiene permiso para asumir un rol `CustomRoleName` en la cuenta 7139999.

   Las credenciales de este usuario llaman a la AWS STS `AssumeRole` API para obtener las credenciales de sesión, que posteriormente se utilizan para llamar a Kinesis Video APIs Streams en la transmisión creada en la cuenta 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. Establezca la clave de acceso, la clave secreta y las credenciales de sesión en función del conjunto anterior en el entorno.

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=
```

7. Ejecute Kinesis Video APIs Streams para describir y obtener el punto final de datos de la transmisión en la cuenta # 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"
}
```

Para obtener step-by-step instrucciones genéricas sobre cómo conceder el acceso a varias cuentas, consulte Delegar el acceso a Cuentas de AWS través de funciones de IAM.

Ejemplos de políticas para Kinesis Video Streams

Los siguientes ejemplos de políticas demuestran cómo puede controlar el acceso de los usuarios a sus Kinesis Video Streams.

Example 1: Permita a los usuarios obtener datos de cualquier transmisión de vídeo de Kinesis

Esta política permite a un usuario o grupo realizar las `ListTagsForStream` operaciones`DescribeStream`, `GetDataEndpoint` `GetMediaListStreams`, y en cualquier transmisión de vídeo de Kinesis. Esta política es adecuada para los usuarios que pueden recibir datos de cualquier transmisión de vídeo.

JSON

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

Example 2: Permitir a un usuario crear una transmisión de vídeo de Kinesis y escribir datos en ella

Esta política permite a un usuario o grupo realizar las operaciones `CreateStream` y `PutMedia`. Esta política es adecuada para una cámara de seguridad que pueda crear una transmisión de vídeo y enviar datos a la misma.

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

Example 3: Permitir a un usuario el acceso total a todos los recursos de Kinesis Video Streams

Esta política permite a un usuario o grupo realizar cualquier operación de Kinesis Video Streams en cualquier recurso. Esta política es adecuada para los administradores.

JSON

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

Example 4: Permitir a un usuario escribir datos en una transmisión de vídeo de Kinesis específica

Esta política permite a un usuario o a un grupo escribir datos en una transmisión de vídeo específica. Esta política es adecuada para un dispositivo que pueda enviar datos a una única transmisión.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesisvideo:PutMedia",
            "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
your_stream/0123456789012"
        }
    ]
}
```

Control del acceso a los recursos de Kinesis Video Streams mediante AWS IoT

En esta sección se describe cómo permitir que un dispositivo (por ejemplo, una cámara) envíe datos de audio y vídeo únicamente a una transmisión de vídeo de Kinesis concreta. Para ello, utilice el proveedor de AWS IoT credenciales y un rol AWS Identity and Access Management (IAM).

Los dispositivos pueden usar certificados X.509 para conectarse AWS IoT mediante protocolos de autenticación mutua TLS. Otros Servicios de AWS (por ejemplo, Kinesis Video Streams) no admiten la autenticación basada en certificados, pero se les puede llamar AWS con credenciales AWS en formato Signature Version 4. El algoritmo Signature, versión 4, normalmente requiere que la persona que llama tenga un identificador de clave de acceso y una clave de acceso secreta. AWS IoT tiene un proveedor de credenciales que le permite utilizar el certificado X.509 integrado como identidad única del dispositivo para autenticar AWS las solicitudes (por ejemplo, las solicitudes a Kinesis Video Streams). Esto elimina la necesidad de almacenar un identificador de clave de acceso y una clave de acceso secreta en el dispositivo.

El proveedor de credenciales autentica un cliente (en este caso, un SDK de Kinesis Video Streams que se ejecuta en la cámara y al que desea enviar datos a una transmisión de vídeo) mediante un certificado X.509 y emite un token de seguridad temporal con privilegios limitados. Puede usar el token para firmar y autenticar cualquier AWS solicitud (en este caso, una llamada a Kinesis Video Streams). Para obtener más información, consulte Autorizar llamadas directas a los servicios. AWS

Esta forma de autenticar las solicitudes de la cámara a Kinesis Video Streams requiere que cree y configure una función de IAM y adjunte las políticas de IAM adecuadas a la función para que el proveedor de credenciales pueda asumir AWS IoT la función en su nombre.

Para obtener más información al respecto AWS IoT, consulte la documentación.AWS IoT Core Para obtener más información acerca del uso de (IAM), consulte AWS Identity and Access Management (IAM).

Temas

- AWS IoT ThingName como nombre de flujo
- AWS IoT CertificateId como nombre de transmisión
- Utilice AWS IoT las credenciales para transmitir a un nombre de transmisión codificado

AWS IoT ThingName como nombre de flujo

Temas

- Paso 1: Crea cualquier tipo de AWS IoT cosa y cualquier AWS IoT cosa
- Paso 2: Cree un rol de IAM para que lo asuma AWS IoT
- Paso 3: Crear y configurar el certificado X.509
- Paso 4: Pruebe las AWS IoT credenciales con la transmisión de vídeo de Kinesis
- Paso 5: implementar AWS IoT certificados y credenciales en el sistema de archivos de la cámara y transmitir los datos a la transmisión de vídeo

Paso 1: Crea cualquier tipo de AWS IoT cosa y cualquier AWS IoT cosa

En AWS IoT, una cosa es una representación de un dispositivo o entidad lógica específicos. En este caso, AWS IoT algo representa la transmisión de vídeo de Kinesis y quiere configurar el control de acceso a nivel de recursos. Para crear una cosa, primero debe crear un AWS IoT tipo de cosa. Puede usar los tipos de AWS IoT cosas para almacenar la información de descripción y configuración que es común a todas las cosas asociadas al mismo tipo de cosa.

1. El siguiente comando de ejemplo crea un tipo de objeto `kvs_example_camera`:

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

2. Este comando de ejemplo crea la `kvs_example_camera_stream` cosa del tipo de `kvs_example_camera` cosa:

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

Paso 2: Cree un rol de IAM para que lo asuma AWS IoT

Las funciones de IAM son similares a las de los usuarios, ya que una función es una AWS identidad con políticas de permisos que determinan lo que la identidad puede y no puede hacer en ella. AWS Cualquier persona que lo necesite puede asumir un rol. Cuando se asume un rol, este proporciona credenciales de seguridad temporales para la sesión del rol.

La función que cree en este paso se puede utilizar AWS IoT para obtener credenciales temporales del servicio de token de seguridad (STS) al realizar solicitudes de autorización de credenciales desde el cliente. En este caso, el cliente es el SDK de Kinesis Video Streams que se ejecuta en la cámara.

Realice los siguientes pasos para crear y configurar este rol de IAM:

1. Crear un rol de IAM.

   El siguiente comando de ejemplo crea un rol de IAM denominado `KVSCameraCertificateBasedIAMRole`:

   ```
   aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
    --assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
   ```

   Puede utilizar el siguiente JSON de política de confianza para `iam-policy-document.json`:

   JSON

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

2. A continuación, adjunte una política de permisos al rol de IAM que creó anteriormente. Esta política de permisos permite el control de acceso selectivo (un subconjunto de operaciones compatibles) a un AWS recurso. En este caso, el AWS recurso es la transmisión de vídeo a la que desea que su cámara envíe datos. En otras palabras, una vez completados todos los pasos de configuración, esta cámara podrá enviar datos únicamente a esta secuencia de vídeo.

   ```
   aws --profile default iam put-role-policy --role-name
    KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
   document 'file://iam-permission-document.json'
   ```

Puede utilizar la siguiente política de IAM (JSON) para: `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}/*"
        }
    ]
}
```

Tenga en cuenta que esta política autoriza las acciones especificadas solo en una transmisión de vídeo (AWS recurso) especificada por el marcador de posición. (`${credentials-iot:ThingName}`) Este marcador de posición adopta el valor del atributo AWS IoT Thing `ThingName` cuando el proveedor de AWS IoT credenciales envía el nombre de la transmisión de vídeo en la solicitud.

3. A continuación, cree un alias de rol para su rol de IAM. El alias de rol es un modelo de datos alternativo que apunta al rol de IAM. La solicitud de un proveedor de AWS IoT credenciales debe incluir un alias de rol que indique qué rol de IAM debe asumir para obtener las credenciales temporales del STS.

   El siguiente comando de ejemplo crea un alias de rol denominado `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. Ahora puede crear la política que le permitirá AWS IoT asumir el rol con el certificado (una vez que esté adjunto) utilizando el alias del rol.

   El siguiente comando de ejemplo crea una política para AWS IoT call`KvsCameraIoTPolicy`.

   ```
   aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-
   document 'file://iot-policy-document.json'
   ```

   Puede usar el siguiente comando para crear el `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
   ```

Paso 3: Crear y configurar el certificado X.509

La comunicación entre un dispositivo (su transmisión de vídeo) y AWS IoT está protegida mediante el uso de certificados X.509.

1. Cree el certificado al que debe adjuntar la política AWS IoT que creó anteriormente.

   ```
   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. Adjunte la política para AWS IoT (`KvsCameraIoTPolicy`creada anteriormente) a este certificado.

```
aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
 $(jq --raw-output '.certificateArn' certificate)
```

3. Adjunta tu AWS IoT cosa (`kvs_example_camera_stream`) al certificado que acabas de crear:

```
aws --profile default iot attach-thing-principal --thing-name
 kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
 certificate)
```

4. Para autorizar las solicitudes a través del proveedor de AWS IoT credenciales, necesitas el punto final de AWS IoT credenciales, que es exclusivo de tu Cuenta de AWS ID. Puedes usar el siguiente comando para obtener el punto final de las AWS IoT credenciales.

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
 --output text > iot-credential-provider.txt
```

5. Además del certificado X.509 creado anteriormente, también debe tener un certificado de CA para establecer la confianza con el servicio de backend a través de TLS. Puede obtener el certificado de CA utilizando el siguiente comando:

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output
 cacert.pem
```

Paso 4: Pruebe las AWS IoT credenciales con la transmisión de vídeo de Kinesis

Ahora puede probar las AWS IoT credenciales que ha configurado hasta ahora.

1. En primer lugar, cree una secuencia de vídeo de Kinesis con la que desee probar esta configuración.

> **Important**
>
> Cree una transmisión de vídeo con un nombre idéntico al nombre de la AWS IoT cosa que creó en el paso anterior (`kvs_example_camera_stream`).

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

2. A continuación, llama al proveedor de AWS IoT credenciales para obtener las credenciales temporales:

```
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**
>
> Puede usar el siguiente comando para obtener`IOT_GET_CREDENTIAL_ENDPOINT`:
>
> ```
> IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
> ```

El JSON de salida contiene el `accessKeysecretKey`, y el`sessionToken`, que puede utilizar para acceder a Kinesis Video Streams.

3. Para la prueba, puede utilizar estas credenciales para invocar la API de Kinesis Video `DescribeStream` Streams para la transmisión de vídeo de `kvs_example_camera_stream` muestra.

```
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
```

Paso 5: implementar AWS IoT certificados y credenciales en el sistema de archivos de la cámara y transmitir los datos a la transmisión de vídeo

> **Note**
>
> En los pasos de esta sección se describe el envío de contenido multimedia a una transmisión de vídeo de Kinesis desde una cámara que utiliza el. the section called “C++”

1. Copie el certificado X.509, la clave privada y el certificado de CA generados en los pasos anteriores en el sistema de archivos de la cámara. Especifique las rutas en las que se

almacenan estos archivos, el nombre del alias de la función y el punto final de AWS IoT las credenciales para ejecutar el `gst-launch-1.0` comando o la aplicación de muestra.

2. El siguiente comando de ejemplo utiliza la autorización de AWS IoT un certificado para enviar vídeo 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 como nombre de transmisión

Para representar su dispositivo (por ejemplo, su cámara) mediante una AWS IoT cosa, pero autorizar un nombre de transmisión diferente, puede utilizar el AWS IoT `certificateId` atributo como nombre de la transmisión y proporcionar permisos a Kinesis Video Streams en la transmisión AWS IoT mediante. Los pasos para lograrlo son similares a los descritos anteriormente, con algunos cambios.

- Modifique la política de permisos para su función de IAM (`iam-permission-document.json`) de la siguiente manera:

  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**
>
> El ARN del recurso utiliza un ID de certificado como marcador de posición del nombre de secuencia. El permiso de IAM funcionará cuando utilices el ID del certificado como nombre de la transmisión. Obtenga el ID del certificado del certificado para poder usarlo como nombre de flujo en la siguiente descripción de la llamada a la API de transmisión.
>
> ```
> export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
> ```

- Verifique este cambio utilizando el comando describe-stream de Kinesis Video Streams en la CLI:

```
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}
```

- Pase el CertificateID al proveedor de credenciales de AWS IoT la aplicación de ejemplo del SDK de C++ de Kinesis Video Streams:

```
credential_provider =
 make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
        cert_path,
        private_key_path,
        role_alias,
        ca_cert_path,
        certificateId);
```

> **Note**
>
> Tenga en cuenta que lo está pasando `thingname` al proveedor de credenciales. AWS IoT Se puede utilizar `getenv` para pasar el nombre de la cosa a la aplicación de demostración de forma similar a como se pasa el resto de los AWS IoT atributos. Utilice el ID de certificado como nombre de secuencia en los parámetros de la línea de comandos cuando ejecute la aplicación de ejemplo.

Utilice AWS IoT las credenciales para transmitir a un nombre de transmisión codificado

Para representar su dispositivo (por ejemplo, su cámara) a través de una AWS IoT cosa, pero autorizar la transmisión a una transmisión de vídeo específica de Amazon Kinesis, proporcione los permisos de Amazon Kinesis Video Streams en la transmisión mediante. AWS IoT El proceso es similar al de las secciones anteriores, con algunos cambios.

Modifique la política de permisos para su función de IAM (`iam-permission-document.json`) de la siguiente manera:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesisvideo:DescribeStream",
                "kinesisvideo:PutMedia",
                "kinesisvideo:TagStream",
                "kinesisvideo:GetDataEndpoint"
            ],
            "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
        }
    ]
}
```

Copie el certificado X.509, la clave privada y el certificado de CA generados en los pasos anteriores en el sistema de archivos de la cámara.

Especifique las rutas en las que se almacenan estos archivos, el nombre del alias de la función, el nombre AWS IoT del objeto y el punto final de AWS IoT las credenciales para ejecutar el `gst-launch-1.0` comando o la aplicación de muestra.

El siguiente comando de ejemplo utiliza la autorización de un AWS IoT certificado para enviar vídeo a 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"
```

Validación de conformidad para Amazon Kinesis Video Streams

Para saber si un programa de cumplimiento Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Referencia de servicios válidos de HIPAA: muestra una lista con los servicios válidos de HIPAA. No todos Servicios de AWS cumplen con los requisitos de la HIPAA.
- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).
- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.

- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, como el PCI DSS, al cumplir con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS uso para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en Amazon Kinesis Video Streams

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS

Seguridad de infraestructura en Kinesis Video Streams

Como servicio gestionado, Amazon Kinesis Video Streams está protegido por AWS los procedimientos de seguridad de red global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utilice las llamadas a la API AWS publicadas para acceder a Kinesis Video Streams a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-

Hellman (ECDHE). La mayoría de los sistemas modernos, como Java 7 y posteriores, son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta que estén asociadas a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Mejores prácticas de seguridad para Kinesis Video Streams

Amazon Kinesis Video Streams ofrece una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no constituyen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

Para consultar las prácticas recomendadas de seguridad de dispositivos remotos, consulte Prácticas recomendadas de seguridad para agentes de dispositivos.

Implementación del acceso a los privilegios mínimos

Cuando concede permisos, debe decidir a quién concede cada permiso y para qué recursos de Kinesis Video Streams se lo concede. Habilite las acciones específicas que desea permitir en dichos recursos. Por lo tanto, debe conceder únicamente los permisos obligatorios para realizar una tarea. La implementación del acceso con privilegios mínimos es esencial a la hora de reducir los riesgos de seguridad y el impacto que podrían causar los errores o los intentos malintencionados.

Por ejemplo, un productor que envía datos a Kinesis Video Streams únicamente requiere `PutMedia`, `GetStreamingEndpoint`, y `DescribeStream`. No conceda permisos a las aplicaciones productoras para todas las acciones (`*`) o para otras acciones, como `GetMedia`.

Para obtener más información, consulte ¿Qué son los privilegios mínimos y por qué los necesita?

Uso de roles de IAM

Las aplicaciones de productor y cliente deben tener credenciales válidas para acceder a Kinesis Video Streams. No debe almacenar las credenciales de AWS directamente en una aplicación de cliente o en un bucket de Amazon S3. Se trata de credenciales de larga duración que no se rotan automáticamente y que podrían tener un impacto empresarial significativo si se ven comprometidas.

En su lugar, debe utilizar un rol de IAM para administrar las credenciales temporales de sus aplicaciones de productor y cliente para acceder a Kinesis Video Streams. Cuando utiliza un rol, no tiene que usar credenciales de larga duración (como un nombre de usuario y una contraseña o claves de acceso) para acceder a otros recursos.

Para obtener más información, consulte los siguientes temas de la guía del usuario de IAM:

- Roles de IAM
- Situaciones habituales con los roles: usuarios, aplicaciones y servicios

Se usa CloudTrail para monitorear las llamadas a la API

Kinesis Video Streams funciona AWS CloudTrail con un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o Servicio de AWS una persona en Kinesis Video Streams.

Puede usar la información recopilada por CloudTrail para determinar la solicitud que se realizó a Kinesis Video Streams, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información, consulte the section called “Registre las llamadas a la API con CloudTrail”.

Solución de problemas de Kinesis Video Streams

Utilice la siguiente información para solucionar los problemas habituales que se producen con Amazon Kinesis Video Streams.

Temas

- Problemas generales
- Problemas con la API
- Problemas de HLS
- Problemas con Java
- Problemas con la biblioteca del productor
- Problemas con la biblioteca de analizadores de flujo
- Problemas de red

Problemas generales

En esta sección se describen los problemas generales que pueden surgir al trabajar con Kinesis Video Streams.

Problemas

- Latencia demasiado alta

Latencia demasiado alta

La latencia puede deberse a la duración de los fragmentos que se envían al servicio Kinesis Video Streams. Una forma de reducir la latencia entre el productor y el servicio consiste en configurar la canalización de medios para que genere fragmentos de menor duración.

Para reducir el número de fotogramas enviados en cada fragmento, reduzca el siguiente valor en`kinesis_video_gstreamer_sample_app.cpp`:

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512,
 NULL);
```

> **Note**
>
> Las latencias son más altas en el navegador Mozilla Firefox debido a la implementación interna de la representación de vídeo.

Problemas con la API

En esta sección se describen los problemas de API que puede encontrar al trabajar con Kinesis Video Streams.

Problemas

- Error: "Opciones desconocidas"
- Error: "Unable to determine service/operation name to be authorized" (No se puede determinar el nombre del servicio o la operación que se debe autorizar)
- Error: "Failed to put a frame in the stream" (Error al colocar un fotograma en la transmisión)
- Error: «El servicio cerró la conexión antes de recibir AckEvent la conexión definitiva»
- Error: "STATUS_STORE_OUT_OF_MEMORY"
- Error: «La credencial debe estar limitada a una región válida».

Error: "Opciones desconocidas"

`GetMedia` y `GetMediaForFragmentList` pueden generar el siguiente error:

```
Unknown options: <filename>.mkv
```

Este error se produce si lo configuró AWS CLI con un `output` tipo `dejson`. Vuelva a AWS CLI configurarlo con el tipo de salida predeterminado (`none`). Para obtener información sobre cómo configurar el AWS CLI, consulte configure en la Referencia de AWS CLI comandos.

Error: "Unable to determine service/operation name to be authorized" (No se puede determinar el nombre del servicio o la operación que se debe autorizar)

`GetMedia` puede generar el siguiente error:

```
Unable to determine service/operation name to be authorized
```

Este error podría producirse si no se especifica correctamente el punto de enlace. Cuando accedas al punto final, asegúrate de incluir el siguiente parámetro en la `GetDataEndpoint` llamada, en función de la API a la que se vaya a llamar:

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

Error: "Failed to put a frame in the stream" (Error al colocar un fotograma en la transmisión)

`PutMedia` puede generar el siguiente error:

```
Failed to put a frame in the stream
```

Este error podría producirse si la conectividad o los permisos no están disponibles para el servicio. Ejecuta lo siguiente en y comprueba que se pueda recuperar la información de la transmisión: AWS CLI

```
aws kinesisvideo describe-stream --stream-name StreamName --endpoint https://
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Si la llamada falla, consulta Solución de AWS CLI errores para obtener más información.

Error: «El servicio cerró la conexión antes de recibir AckEvent la conexión definitiva»

`PutMedia` puede generar el siguiente error:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was
 received
```

Este error podría producirse si `PushbackInputStream` no está implementado correctamente. Compruebe que los `unread()` métodos estén implementados correctamente.

Error: "STATUS_STORE_OUT_OF_MEMORY"

`PutMedia` puede generar el siguiente error:

```
The content store is out of memory.
```

Este error se produce cuando el almacén de contenido no se asigna con tamaño suficiente. Para aumentar el tamaño del almacén de contenido, aumente el valor de `StorageInfo.storageSize`. Para obtener más información, consulte StorageInfo.

Error: «La credencial debe estar limitada a una región válida».

Este error se produce si la región de firma no coincide con la región del punto final.

Por ejemplo, si especificas `us-west-2` como región de firma, pero intentas conectarte al punto final `kinesisvideo.us-east-1.amazonaws.com` (`us-east-1`), recibirás este error.

En algunas aplicaciones, como kvssink, la cadena alternativa de regiones es la predeterminada. `us-west-2` Comprueba que has configurado tu región correctamente según la aplicación que estés utilizando.

Problemas de HLS

Si la transmisión de vídeo no se reproduce correctamente, consultethe section called “Solución de problemas de HLS”.

Problemas con Java

En esta sección se describe cómo solucionar los problemas comunes de Java que se producen al trabajar con Kinesis Video Streams.

Problemas

- Habilitar los logs de Java

Habilitar los logs de Java

Para solucionar problemas con las bibliotecas y muestras de Java, resulta útil habilitar y examinar los registros de depuración. Para habilitar los logs de depuración, haga lo siguiente:

1. Agregue `log4j` al archivo `pom.xml` en el nodo `dependencies`:

```
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
```

2. En el directorio `target/classes`, cree un archivo denominado `log4j.properties` con el contenido siguiente:

```
# 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
```

Los logs de depuración imprimirán en la consola del IDE.

Problemas con la biblioteca del productor

En esta sección se describen los problemas que podrían producirse cuando se utiliza Cargar a Kinesis Video Streams.

Problemas

- No se puede compilar el SDK del productor
- La transmisión de vídeo no aparece en la consola
- Error: «El token de seguridad incluido en la solicitud no es válido» al transmitir datos mediante la aplicación de demostración GStreamer
- Error: "Failed to submit frame to Kinesis Video client" (No se pudo enviar el fotograma al cliente de Kinesis Video)

- GStreamer la aplicación se detiene con el mensaje «la transmisión se ha detenido, no se ha negociado el motivo» en OS X
- Error: «No se pudo asignar el montón» al crear el Kinesis Video Client GStreamer en una demostración en Raspberry Pi
- Error: «Instrucción ilegal» al ejecutar la GStreamer demostración en Raspberry Pi
- La cámara no se carga en un dispositivo Raspberry Pi
- No se puede encontrar la cámara en macOS High Sierra
- No se ha encontrado el archivo jni.h al compilar en macOS High Sierra
- Errores de Curl al ejecutar la aplicación de demostración GStreamer
- Aserción de rango/marca temporal en el tiempo de ejecución en un dispositivo Raspberry Pi
- Aserción en gst_value_set_fraction_range_full en un dispositivo Raspberry Pi
- Error STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) en Android
- Error al alcanzar la duración máxima del fragmento
- Error «Nombre de objeto no válido pasado» al utilizar la autorización de IoT

No se puede compilar el SDK del productor

Compruebe que las bibliotecas necesarias se encuentran en la ruta. Para ello, utilice el comando siguiente:

```
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
```

La transmisión de vídeo no aparece en la consola

Para mostrar la secuencia de vídeo en la consola, debe codificarse mediante H.264 en formato AvCC. Si no aparece la secuencia , verifique lo siguiente:

- Su Banderas de adaptación NAL se establece en `NAL_ADAPTATION_ANNEXB_NALS | NAL_ADAPTATION_ANNEXB_CPD_NALS` si la secuencia original está en formato Anexo B. Este es el valor predeterminado en el constructor `StreamDefinition`.
- Está facilitando los datos privados del códec correctamente. Para H.264, estos son el conjunto de parámetros de secuencia (SPS) y el conjunto de parámetros de imagen (PPS). Dependiendo

del origen de medios, estos datos se pueden recuperar del origen multimedia por separado o codificados en el fotograma.

Muchas secuencias elementales tienen el siguiente formato, donde Ab es el código de inicio de Anexo B (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)
```

El CPD (códec de datos privados), si el H.264 está en la transmisión como SPS y PPS, se puede adaptar al formato AVCC. A menos que la canalización multimedia proporcione el CPD por separado, la aplicación puede extraer el CPD del fotograma buscando el primer fotograma Idr (que debe contener el SPS y el PPS), extraer los dos NALUs (que será Ab (Sps) Ab (Pps)) y colocarlo en el CPD`StreamDefinition`.

Error: «El token de seguridad incluido en la solicitud no es válido» al transmitir datos mediante la aplicación de demostración GStreamer

Si se produce este error, existe un problema con las credenciales. Compruebe lo siguiente:

- Si está utilizando credenciales temporales, debe especificar el token de sesión.
- Compruebe que las credenciales temporales no hayan caducado.
- Compruebe que tiene configurados los derechos adecuados.
- En macOS, verifique que no tiene credenciales almacenadas en caché en Keychain.

Error: "Failed to submit frame to Kinesis Video client" (No se pudo enviar el fotograma al cliente de Kinesis Video)

Si se produce este error, las marcas temporales no están configuradas correctamente en la secuencia de origen. Pruebe lo siguiente:

- Utilice la muestra de SDK más reciente, ya que es posible que incluya una actualización que solucione el problema.
- Ajuste la transmisión de alta calidad a una velocidad de bits más alta y corrija cualquier fluctuación en la transmisión de origen si la cámara lo permite.

GStreamer la aplicación se detiene con el mensaje «la transmisión se ha detenido, no se ha negociado el motivo» en OS X

El streaming puede detenerse en OS X con el siguiente mensaje:

```
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 posible solución para esto es eliminar los parámetros de velocidad de fotogramas de la `gst_caps_new_simple` llamada en: `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);
```

Error: «No se pudo asignar el montón» al crear el Kinesis Video Client GStreamer en una demostración en Raspberry Pi

La aplicación GStreamer de ejemplo intenta asignar 512 MB de RAM, que podrían no estar disponibles en su sistema. Puede reducir esta asignación si reduce el valor siguiente en `KinesisVideoProducer.cpp`:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Error: «Instrucción ilegal» al ejecutar la GStreamer demostración en Raspberry Pi

Si se produce el siguiente error al ejecutar la GStreamer demostración, compruebe que ha compilado la aplicación para la versión correcta de su dispositivo. (Por ejemplo, comprueba que no estás compilando para Raspberry Pi 3 cuando utilizas Raspberry Pi 2).

```
INFO - Initializing curl.
Illegal instruction
```

La cámara no se carga en un dispositivo Raspberry Pi

Para comprobar si se ha cargado la cámara, ejecute lo siguiente:

```
ls /dev/video*
```

Si no encuentra nada, ejecute lo siguiente:

```
vcgencmd get_camera
```

El resultado debería tener un aspecto similar al siguiente:

```
supported=1 detected=1
```

Si el controlador no detecta la cámara, haga lo siguiente:

1. Compruebe la configuración física de la cámara y verifique que está conectada correctamente.
2. Ejecute lo siguiente para actualizar el firmware:

   ```
   sudo rpi-update
   ```

3. Reinicie el dispositivo.
4. Ejecute lo siguiente para cargar el controlador:

   ```
   sudo modprobe bcm2835-v4l2
   ```

5. Compruebe que la cámara se ha detectado:

   ```
   ls /dev/video*
   ```

No se puede encontrar la cámara en macOS High Sierra

En macOS High Sierra, la aplicación de demostración no encuentra la cámara si hay más de una cámara disponible.

No se ha encontrado el archivo jni.h al compilar en macOS High Sierra

Para solucionar este error, actualice su instalación de Xcode a la versión más reciente.

Errores de Curl al ejecutar la aplicación de demostración GStreamer

Para resolver los errores de curl al ejecutar la aplicación de GStreamer demostración, copie este archivo de certificado en. `/etc/ssl/cert.pem`

Aserción de rango/marca temporal en el tiempo de ejecución en un dispositivo Raspberry Pi

Si se produce una aserción de rango o de marca temporal en el tiempo de ejecución, actualice el firmware y reinicie el dispositivo:

```
sudo rpi-update
$ sudo reboot
```

Aserción en gst_value_set_fraction_range_full en un dispositivo Raspberry Pi

La siguiente aserción aparece si se está ejecutando el servicio `uv4l`:

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
 denominator_end) < 0' failed
```

Si ocurre esto, detenga el servicio `uv4l` y reinicie la aplicación.

Error STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) en Android

El siguiente error aparece si Banderas de adaptación NAL son incorrectos para la secuencia multimedia:

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Si se produce este error, proporcione la marca `.withNalAdaptationFlags` correcta para su medio (por ejemplo, `NAL_ADAPTATION_ANNEXB_CPD_NALS`). Proporcione esta marca en la siguiente línea de la Android:

https://github.com/awslabs/aws-sdk-android-samples/blob/master/AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/StreamConfigurationFragment.java#L169

Error al alcanzar la duración máxima del fragmento

Este error se produce cuando un fragmento multimedia de una secuencia supera el límite máximo de duración de fragmento. Consulte el límite máximo de duración de los fragmentos en la the section called “Cuotas del servicio de API multimedia y multimedia archivada” sección.

Pruebe lo siguiente para resolver este problema:

- Si utiliza una cámara web/USB, realice alguna de las siguientes operaciones:
  - Si utilizas la fragmentación basada en fotogramas clave, configura el codificador para que proporcione fotogramas clave en 10 segundos.
  - Si no utilizas la fragmentación basada en fotogramas clave, al definir la transmisiónEscribe y examina el código, establece el límite máximo de duración del fragmento en un valor inferior a 10 segundos.
  - Si utilizas codificadores de software (como x264) en la GStreamer canalización, puedes establecer el key-int-max atributo en un valor en un plazo de 10 segundos. Por ejemplo, key-int-max configúrelo en 60, con fps en 30, para habilitar los fotogramas clave cada 2 segundos.
- Si utilizas una cámara RPI, establece el atributo keyframe-interval en menos de 10 segundos.
- Si utilizas una cámara IP (RTSP), establece el tamaño del GOP en 60.

Error «Nombre de objeto no válido pasado» al utilizar la autorización de IoT

Para evitar este error (`HTTP Error 403: Response: {"message":"Invalid thing name passed"}`) cuando utilice credenciales de IoT para la autorización, asegúrese de que el valor de `stream-name` (un parámetro obligatorio del `kvssink` elemento) sea idéntico al valor de`iot-thingname`. Para obtener más información, consulte GStreamer referencia de parámetros de elementos.

Problemas con la biblioteca de analizadores de flujo

En esta sección se describen los problemas que podrían producirse cuando se utiliza Transmita usando la biblioteca de analizadores.

Problemas

- No se puede tener acceso a un solo fotograma de la transmisión
- Error de descodificación de fragmentos

No se puede tener acceso a un solo fotograma de la transmisión

Para acceder a un único fotograma desde una fuente de streaming en su aplicación de consumo, compruebe que la transmisión contenga los datos privados del códec correctos. Para obtener información sobre el formato de los datos de una transmisión, consulte Modelo de datos.

Para aprender a usar los datos privados del códec para acceder a un marco, consulta el siguiente archivo de prueba en el GitHub sitio web: .java KinesisVideoRendererExampleTest

Error de descodificación de fragmentos

Si los fragmentos no están correctamente codificados en un formato H.264 y en un nivel admitido por el navegador, es posible que aparezca el siguiente error al reproducir la transmisión en la consola:

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264
 content
```

Si esto ocurre, verifique lo siguiente:

- La resolución de los fotogramas coincide con la resolución especificada en los datos privados del códec.
- El perfil H.264 y el nivel de los fotogramas codificados coincide con el perfil y el nivel especificados en los datos privados del códec.
- El navegador admite la combinación de perfil y nivel. La mayoría de los navegadores actuales admiten todas las combinaciones de perfil y nivel.
- Las marcas temporales son precisas y están en el orden correcto, y no se crean marcas temporales duplicadas.
- La aplicación está codificando los datos de fotogramas con el formato H.264.

Problemas de red

Si ve errores de conexión, como «Se ha agotado el tiempo de espera de la conexión» o «Ha fallado la conexión», al intentar conectarse a Kinesis Video Streams, es posible que se deba a restricciones de rango de direcciones IP en su configuración de red.

Si su configuración tiene restricciones de rango de direcciones IP para Kinesis Video Streams, actualice la configuración de red para incluir en una lista los rangos de direcciones IP de Kinesis Video Streams.

> **Important**
>
> La lista de rangos de IP no es una lista exhaustiva de direcciones IP de Kinesis Video Streams. Incluya los rangos de direcciones IP que ve y tenga en cuenta que las direcciones IP pueden cambiar con el tiempo.

Para obtener más información, consulte Intervalos de AWS IP. Para recibir una notificación cuando cambien los rangos de IP, sigue el procedimiento de suscripción.

Historial de documentos de Amazon Kinesis Video Streams

En la siguiente tabla se describen los cambios importantes en la documentación desde la última versión de Amazon Kinesis Video Streams.

- Última versión de la API: 29-11-2017
- Última actualización de la documentación: 6 de enero de 2025

Cambio	Descripción	Fecha
C++ en Raspberry Pi	Se ha actualizado la documentación para usar el SDK para productores de C++ en Raspberry Pi.	6 de enero de 2025
Conexión con Amazon Kinesis Video Streams Edge Agent Edge-to-Cloud	Lanzamiento de una nueva función. Para obtener más información, consulte Programe la grabación y el almacenamiento de vídeo.	27 de junio de 2023
Introducción: enviar datos a una transmisión de vídeo de Kinesis	Tutorial básico para enviar datos multimedia desde una cámara a una transmisión de vídeo de Kinesis. Para obtener más información, consulte Enviar datos a una transmisión de vídeo de Amazon Kinesis.	21 de enero de 2019
Metadatos de streaming	Puede usar el SDK de Producer para incrustar metadatos en una transmisión de vídeo de Kinesis. Para obtener más información, consulte Uso de metadatos de	28 de septiembre de 2018

Cambio	Descripción	Fecha
	streaming con Kinesis Video Streams.	
Registro del SDK de C++ Producer	Puede configurar el registro para las aplicaciones del SDK de C++ Producer. Para obtener más información, consulte Utilice el registro con el SDK para productores de C ++.	18 de julio de 2018
Streaming de vídeo HLS	Ahora puede ver una transmisión de vídeo de Kinesis mediante HTTP Live Streaming. Para obtener más información, consulte Reproducción de Kinesis Video Streams.	13 de julio de 2018
Streaming desde una fuente RTSP	Aplicación de ejemplo para Kinesis Video Streams que se ejecuta en un contenedo r Docker y transmite vídeo desde una fuente RTSP. Para obtener más información, consulte RTSP y Docker.	20 de junio de 2018
Complemento SDK para C++ Producer GStreamer	Muestra cómo crearlo C++ para usarlo como GStreamer destino. Para obtener más información, consulte GStreamer Complemento: kvssink.	15 de junio de 2018

Cambio	Descripción	Fecha
Documentación de referencia de devolución de llamadas de SDK Producer	Documentación de referenci a para las devoluciones de llamada utilizadas por Cargar a Kinesis Video Streams. Para obtener más información, consulte Retrollamadas del SDK del productor.	12 de junio de 2018
Requisitos del sistema	Documentación de requisito s de memoria y almacenam iento para dispositivos de productores y SDK. Para obtener más información, consulte Requisitos del sistema para Amazon Kinesis Video Streams.	30 de mayo de 2018
CloudTrail apoyo	Documentación para CloudTrail monitorizar el uso de la API. Para obtener más información, consulte Registre las llamadas a la API de Amazon Kinesis Video Streams con AWS CloudTrail.	24 de mayo de 2018
Documentación de referenci a de estructuras de SDK Producer	Documentación de referencia para las estructuras utilizada s por Cargar a Kinesis Video Streams. Para obtener más información, consulte Estructuras del SDK para productores y Estructuras de transmisión de vídeo de Kinesis.	7 de mayo de 2018

Cambio	Descripción	Fecha
Documentación de ejemplo de Renderer	Documentación de la aplicación de ejemplo Renderer, que muestra cómo decodificar y mostrar fotogramas de una transmisión de vídeo de Kinesis. Para obtener más información, consulte Ejemplo: análisis y renderización de fragmentos de Kinesis Video Streams.	15 de marzo de 2018
Documentación de referencia de límites de SDK Producer	Información sobre límites para operaciones en la C++. Para obtener más información, consulte Cuotas de SDK para productores.	13 de marzo de 2018
Monitorización	Información sobre la supervisión de las métricas y las llamadas a la API de Kinesis Video Streams mediante CloudWatch Amazon AWS CloudTrail y. Para obtener más información, consulte Supervisión de Amazon Kinesis Video Streams.	5 de febrero de 2018
Referencia de marcas de adaptación de capa de abstracción de red (NAL)	Información sobre la configuración de las marcas de adaptación de NAL al consumir vídeo en streaming. Para obtener más información, consulte Banderas de adaptación NAL.	15 de enero de 2018

Cambio	Descripción	Fecha
Compatibilidad de Android para vídeo en streaming	Kinesis Video Streams ahora admite la transmisión de vídeo desde dispositivos Android. Para obtener más información, consulte Android.	12 de enero de 2018
Documentación de ejemplo de Kinesis Video	Documentación de la aplicación de ejemplo de Kinesis Video, que muestra cómo utilizarla Vea la salida de las cámaras mediante la biblioteca de analizadores en una aplicación. Para obtener más información, consulte KinesisVideoExample.	9 de enero de 2018
Se publicó la documentación de Kinesis Video Streams	Esta es la versión inicial de la Guía para desarrolladores de Amazon Kinesis Video Streams.	29 de noviembre de 2017

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.