



Guia do Desenvolvedor

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: Guia do Desenvolvedor

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Kinesis Video Streams?	1
Disponibilidade de regiões	2
Você é um usuário iniciante do Kinesis Video Streams?	3
Requisitos do sistema	5
Requisitos de câmera	5
Sistemas operacionais testados	6
Requisitos de armazenamento do SDK	6
Como funciona	7
API e bibliotecas de produtor	8
API Kinesis Video Streams	9
Padrão de descoberta de endpoints	11
Bibliotecas de produtores	12
Reprodução de vídeo	12
Requisitos de reprodução	13
Reprodução de vídeo com HLS	17
Reprodução de vídeo com MPEG-DASH	27
Uso de metadados de streaming	32
Adicionar metadados a um stream de vídeo do Kinesis	33
Consumindo metadados incorporados em um stream de vídeo do Kinesis	34
Limites de metadados de streaming	36
Modelo de dados	36
Elementos do cabeçalho de streaming	36
Transmita dados da trilha	43
Elementos do cabeçalho de quadros	44
Dados do quadro MKV	44
Conceitos básicos	45
Configurar uma conta	45
Inscreva-se para um Conta da AWS	46
Criar um usuário com acesso administrativo	46
Crie uma Conta da AWS chave	47
Crie um stream de vídeo do Kinesis	48
Crie um stream de vídeo usando o console	48
Crie um stream de vídeo usando o AWS CLI	48
Enviar dados para um stream de vídeo do Amazon Kinesis	49

Crie o SDK e as amostras	49
Execute as amostras para fazer upload de mídia para o Kinesis Video Streams	53
Revise os objetos de reconhecimento	55
Consuma dados de mídia	55
Exibir mídia no console	55
Consuma dados de mídia usando o HLS	56
Agente Edge	57
Operações de API do Amazon Kinesis Video Streams Edge Agent	58
Monitoramento do Amazon Kinesis Video Streams Edge Agent	58
Implemente em AWS IoT Greengrass modo não-	58
1. Instale as dependências	59
2. Crie recursos para os URLs RTSP da sua câmera IP	61
3. Criar uma política do IAM	63
4. Criar um perfil do IAM	65
5. Crie o alias da AWS IoT função	66
6. Crie a AWS IoT política	67
7. Crie AWS IoT algo e obtenha AWS IoT Core credenciais	68
8. Crie e execute o Amazon Kinesis Video Streams Edge Agent	71
9. (Opcional) Instale o CloudWatch agente	82
10. (Opcional) Execute o agente Amazon Kinesis Video Streams Edge	85
Implemente em AWS IoT Greengrass	88
1. Criar instância do Ubuntu	88
2. Configurar o dispositivo AWS IoT Greengrass principal	90
3. Crie recursos para os URLs RTSP da sua câmera IP	91
4. Adicionar permissões à função TES	93
5. Instale o componente Secret Manager	95
6. Implante o Amazon Kinesis Video Streams Edge Agent no dispositivo	98
7. (Opcional) Instale o componente do gerenciador de AWS IoT Greengrass registros	107
Perguntas frequentes	111
Quais sistemas operacionais são compatíveis com o Amazon Kinesis Video Streams Edge Agent?	111
O Amazon Kinesis Video Streams Edge Agent oferece suporte à mídia H.265?	112
O Amazon Kinesis Video Streams Edge Agent funciona no AL2?	112
Como posso executar vários streams dentro da AWS IoT coisa ou dispositivo?	112
Como posso editar um StartEdgeConfigurationUpdate depois de ter sido enviado? ..	112
Você tem algum exemplo em comumScheduleConfigs?	112

Existe um limite máximo de transmissão?	113
Como faço para reiniciar um trabalho com erro?	113
Como faço para monitorar a integridade do meu Amazon Kinesis Video Streams Edge Agent?	114
Transmita vídeo por meio de uma VPC	115
Mais informações	115
Procedimentos de endpoint de VPC	115
Imagens	118
Conceitos básicos do GetImages	118
Primeiros passos com a entrega do Amazon S3	119
UpdateImageGenerationConfiguration	119
DescribeImageGenerationConfiguration	121
Tags MKV do produtor	122
Adicionar tags de metadados no Producer SDK usandoPutEventMetaData	123
Limites	123
Metadados do objeto S3	124
Caminho do objeto S3 (imagem)	124
Recomendações de URI do Amazon S3 para proteção contra limitação	125
Notificações	126
UpdateNotificationConfiguration	126
DescribeNotificationConfiguration	126
Etiquetas MKV do produtor	122
Sintaxe para tags MKV do Producer	122
Limites de tags MKV	127
.....	127
.....	127
Carga útil de tópicos do Amazon SNS	127
Visualizando suas mensagens do Amazon SNS	128
Segurança	130
Proteção de dados	131
O que é criptografia do lado do servidor para o Kinesis Video Streams?	131
Custos, regiões e considerações de desempenho	131
Como faço para começar a usar a criptografia do lado do servidor?	132
Criação e uso de uma chave gerenciada pelo cliente	133
Permissões para usar uma chave gerenciada pelo cliente	134
Controle do acesso aos recursos do Kinesis Video Streams usando o IAM	135

Sintaxe da política	136
Ações para o Kinesis Video Streams	137
Nomes de recursos da Amazon (ARNs) para o Kinesis Video Streams	137
Conceder acesso a outras contas do IAM a um stream de vídeo do Kinesis	138
Exemplo de políticas	141
Controlando o acesso aos recursos do Kinesis Video Streams usando AWS IoT	143
AWS IoT ThingName como nome do stream	144
AWS IoT CertificateId como nome do stream	150
Use AWS IoT credenciais para transmitir para um nome de fluxo codificado	152
Monitorar	153
Compliance Validation	153
Resiliência	155
Segurança da infraestrutura	155
Práticas recomendadas de segurança	155
Implemente o acesso de privilégio mínimo	156
Usar funções do IAM	156
Use CloudTrail para monitorar chamadas de API	156
Bibliotecas de produtores	158
Cliente do Kinesis Video Streams Producer	158
Biblioteca do Kinesis Video Streams Producer	159
Tópicos relacionados	160
Biblioteca de produtores Java	160
Procedimento: usar o Producer SDK em Java	161
Etapa 1: Fazer download e configurar o código	162
Etapa 2: escrever e examinar o código	163
Etapa 3: executar e verificar o código	165
Biblioteca de produtores do Android	165
Procedimento: Usar o SDK de produtores do Android	166
Pré-requisitos	166
Etapa 1: Fazer download e configurar o código	170
Etapa 2: Examine o código	171
Etapa 3: executar e verificar o código	173
Biblioteca de produtores C++	175
Modelo de objeto	175
Colocando mídia no stream	175
Interfaces de retorno de chamada	175

Procedimento: usar o Producer SDK em C++	176
Etapa 1: Fazer download e configurar o código	178
Etapa 2: escrever e examinar o código	178
Etapa 3: executar e verificar o código	185
Usando o C++ Producer SDK como um plug-in do GStreamer	185
Usando o C++ Producer SDK como um plug-in do GStreamer em um contêiner Docker	186
Uso do registro em log	186
Biblioteca de produtores C	187
Modelo de objeto	187
Colocando mídia no stream	188
Procedimento: usar o C Producer SDK	188
Etapa 1: faça download do código	190
Etapa 2: escrever e examinar o código	190
Etapa 3: executar e verificar o código	193
SDK para produtores de C++ no Raspberry Pi	195
Pré-requisitos	196
Crie um usuário do IAM com permissão para gravar no Kinesis Video Streams	196
Conecte seu Raspberry Pi à sua rede Wi-Fi	198
Conecte-se remotamente ao seu Raspberry Pi	199
Configurar a câmera Raspberry Pi	199
Pré-requisitos de instalação do software	200
Baixe e crie o SDK do Kinesis Video Streams C++ Producer	201
Transmita vídeo para sua transmissão de vídeo do Kinesis e assista à transmissão ao vivo	202
Referência	203
Limites do SDK do produtor	203
Referência de código de erro	207
Sinalizadores de adaptação do NAL	264
Estruturas de produtor	265
Estruturas de streamings	268
Retornos de chamada	289
Biblioteca do analisador de streaming	297
Procedimento: Usar a biblioteca de análise de streaming de vídeo Kinesis	297
Pré-requisitos	297
Etapa 1: Baixe e configure o código	298
Próxima etapa	298

Etapa 2: escrever e examinar o código	298
StreamingMkvReader	299
FragmentMetadataVisitor	299
OutputSegmentMerger	301
KinesisVideoExample	302
Próxima etapa	306
Etapa 3: executar e verificar o código	306
Exemplos	307
Exemplos: envio de dados para o Kinesis Video Streams	307
Exemplos: recuperação de dados do Kinesis Video Streams	307
Exemplos: reprodução de dados de vídeo	307
Pré-requisitos	307
Plugin do GStreamer - kvssink	308
Baixe, crie e configure o elemento GStreamer	309
Execute o elemento GStreamer	309
Comandos de execução	310
Execute o elemento GStreamer em um contêiner Docker	312
Referência de parâmetros	315
PutMedia API	329
Etapa 1: Baixe e configure o código	330
Etapa 2: escrever e examinar o código	331
Etapa 3: executar e verificar o código	333
RTSP e Docker	334
Vídeos de tutorial	334
Pré-requisitos	334
Crie a imagem do Docker	335
Execute o aplicativo de exemplo RTSP	335
Renderizador	337
Pré-requisitos	337
Executando o exemplo do renderizador	338
Como funciona	338
Monitorar	341
Monitorando métricas com CloudWatch	342
CloudWatch orientação de métricas	357
Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch	362
CloudWatch orientação de métricas para o Amazon Kinesis Video Streams Edge Agent	365

Registro de chamadas de API do CloudTrail com	367
Amazon Kinesis Video Streams e CloudTrail	367
Exemplo: entradas do arquivo de log do Amazon Kinesis Video Streams	368
Cotas	373
Controle as cotas de serviço de API do plano	373
Cotas de serviços de API de mídia e mídia arquivada	378
Cotas de fragmentação de metadados e de mídia fragmentada	383
Fragmentar cotas de metadados	386
Tags de transmissão	387
Solução de problemas	388
Problemas gerais	388
Latência muito alta	388
Problemas de API	389
Erro: "opções desconhecidas"	389
Erro: "Unable to determine service/operation name to be authorized" (Não foi possível determinar o nome do serviço/operação a ser autorizado)	389
Erro: "Failed to put a frame in the stream" (Falha ao inserir um quadro no streaming)	390
Erro: "O serviço fechou a conexão antes do recebimento final AckEvent "	390
Erro: "STATUS_STORE_OUT_OF_MEMORY"	390
Problemas de HLS	391
Problemas de Java	391
Habilitar logs do Java	391
Problemas com a Biblioteca do Produtor	392
Não é possível compilar o Producer SDK	393
O streaming de vídeo não aparece no console	393
Erro: "O token de segurança incluído na solicitação é inválido" ao fazer streaming de dados usando o aplicativo de demonstração GStreamer	394
Erro: "falha ao enviar quadro ao cliente de vídeo do Kinesis"	394
O aplicativo GStreamer é interrompido com a mensagem "streaming interrompido, motivo não negociado" no OS X.	394
Erro: "Falha ao alocar heap" ao criar o cliente de vídeo Kinesis na demonstração do GStreamer no Raspberry Pi	395
Erro: "Instrução ilegal" ao executar demonstração do GStreamer no Raspberry Pi	395
A câmera não carrega no Raspberry Pi	395
Não é possível encontrar a câmera no macOS High Sierra	396
Arquivo jni.h não encontrado durante a compilação no macOS High Sierra	396

Erros de Curl durante a execução do aplicativo de demonstração GStreamer	396
Declaração de time stamp/intervalo no tempo de execução no Raspberry Pi	396
Declaração em gst_value_set_fraction_range_full no Raspberry Pi	397
Erro STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) no Android	397
A duração máxima do fragmento foi atingida (erro)	397
Erro "Invalid thing name passed" ao usar a autorização da IoT	398
Problemas na biblioteca do Stream Parser	398
Não é possível acessar um quadro único do streaming	398
Erro na decodificação de fragmentos	398
Problemas de rede	399
Histórico do documento	400
Referência da API	405
Ações	405
Amazon Kinesis Video Streams	406
Mídia do Amazon Kinesis Video Streams	529
Mídia arquivada do Amazon Kinesis Video Streams	546
Amazon Kinesis Video Streams	594
Amazon Kinesis WebRTC Streams	603
Tipos de dados	607
Amazon Kinesis Video Streams	609
Mídia do Amazon Kinesis Video Streams	650
Mídia arquivada do Amazon Kinesis Video Streams	653
Amazon Kinesis Video Streams	671
Amazon Kinesis WebRTC Streams	673
Erros comuns	673
Parâmetros gerais	675
.....	dclxxviii

O que é o Kinesis Video Streams?

Você pode usar o Amazon Kinesis Video Streams, um serviço AWS service (Serviço da AWS) totalmente gerenciado, para transmitir vídeo ao vivo de dispositivos para o, ou criar aplicativos Nuvem AWS para processamento de vídeo em tempo real ou análise de vídeo orientada por lotes.

O Kinesis Video Streams não é apenas armazenamento para dados de vídeo. Ele pode ser usado para assistir fluxos de vídeo em tempo real à medida que são recebidos na nuvem. Você pode monitorar suas transmissões ao vivo no AWS Management Console, ou desenvolver seu próprio aplicativo de monitoramento que usa a biblioteca da API Kinesis Video Streams para exibir vídeo ao vivo.

Você pode usar o Kinesis Video Streams para capturar grandes quantidades de dados de vídeo ao vivo de milhões de fontes, incluindo smartphones, câmeras de segurança, webcams, câmeras embutidas em carros, drones e outras fontes. Você também pode enviar dados sem vídeo, serializados no tempo, como dados de áudio, imagens térmicas, dados de profundidade e dados de RADAR. À medida que o vídeo ao vivo é transmitido dessas fontes para um stream de vídeo do Kinesis, você pode criar aplicativos para acessar os dados frame-by-frame em tempo real para processamento de baixa latência. O Kinesis Video Streams é independente da fonte. Você pode transmitir vídeo da webcam de um computador usando a [Plugin do GStreamer - kvssink](#) biblioteca ou de uma câmera na sua rede usando o protocolo de streaming em tempo real (RTSP).

Você também pode configurar seu stream de vídeo do Kinesis para armazenar dados de mídia de forma durável durante o período de retenção especificado. O Kinesis Video Streams armazena automaticamente esses dados e os criptografa em repouso. Além disso, o Kinesis Video Streams indexa os dados armazenados com base nos registros de data e hora do produtor e de ingestão. Você pode criar aplicativos que processam periodicamente os dados de vídeo em lote ou criar aplicativos que exijam acesso único aos dados históricos para diferentes casos de uso.

Seus aplicativos personalizados, em tempo real ou orientados por lotes, podem ser executados em instâncias do Amazon EC2. Esses aplicativos podem processar dados usando algoritmos de aprendizado profundo de código aberto ou usar aplicativos de terceiros que se integram ao Kinesis Video Streams.

Os benefícios de usar o Kinesis Video Streams incluem o seguinte:

- Conecte-se e transmita de milhões de dispositivos — Você pode usar o Kinesis Video Streams para conectar e transmitir vídeo, áudio e outros dados de milhões de dispositivos, desde

smartphones de consumo, drones e câmeras de painel. Você pode usar as bibliotecas de produtores do Kinesis Video Streams para configurar seus dispositivos e transmitir de forma confiável em tempo real ou como upload de mídia. after-the-fact

- Armazene, criptografe e indexe dados de forma durável — Você pode configurar seu stream de vídeo do Kinesis para armazenar dados de mídia de forma durável por períodos de retenção personalizados. O Kinesis Video Streams também gera um índice sobre os dados armazenados com base em registros de data e hora gerados pelo produtor ou do lado do serviço. Seus aplicativos podem recuperar dados especificados em um fluxo usando o índice de tempo.
- Concentre-se no gerenciamento de aplicativos em vez da infraestrutura — o Kinesis Video Streams não tem servidor, portanto, não há infraestrutura para configurar ou gerenciar. Você não precisa se preocupar com a implantação, configuração ou escalabilidade elástica da infraestrutura subjacente, pois seus fluxos de dados e o número de aplicativos consumidores aumentam e diminuem. O Kinesis Video Streams faz automaticamente toda a administração e manutenção necessárias para gerenciar streams, para que você possa se concentrar nos aplicativos, não na infraestrutura.
- Crie aplicativos em tempo real e em lote em fluxos de dados — Você pode usar o Kinesis Video Streams para criar aplicativos personalizados em tempo real que operam em fluxos de dados ao vivo e criar aplicativos em lote ou únicos que operam em dados persistentes de forma durável sem requisitos rígidos de latência. Você pode criar, implantar e gerenciar aplicativos personalizados: soluções de código aberto (Apache MXNet, OpenCV), desenvolvidas internamente ou de terceiros usando AWS Marketplace para processar e analisar seus streams. Você pode usar as APIs do Kinesis Get Video Streams para criar vários aplicativos simultâneos que processam dados em tempo real ou orientados por lotes.
- Transmita dados com mais segurança — o Kinesis Video Streams criptografa todos os dados à medida que eles fluem pelo serviço e quando persistem os dados. O Kinesis Video Streams aplica criptografia baseada em Transport Layer Security (TLS) no streaming de dados de dispositivos e criptografa todos os dados em repouso usando (). AWS Key Management Service AWS KMS Além disso, você pode gerenciar o acesso aos seus dados usando AWS Identity and Access Management (IAM).
- Pague conforme o uso — Para obter mais informações, consulte [AWS Pricing Calculator](#).

Disponibilidade de regiões

O Amazon Kinesis Video Streams está disponível nas seguintes regiões:

Nome da região	AWSCódigo da região
Leste dos EUA (Ohio)	us-east-2
Leste dos EUA (N. da Virgínia)	us-east-1
Oeste dos EUA (Oregon)	us-west-2
África (Cidade do Cabo)	af-south-1
Ásia-Pacífico (Hong Kong)	ap-east-1
Ásia-Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Ásia-Pacífico (Tóquio)	ap-northeast-1
Canadá (Central)	ca-central-1
China (Pequim)	cn-north-1
Europa (Frankfurt)	eu-central-1
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (Paris)	eu-west-3
América do Sul (São Paulo)	sa-east-1

Você é um usuário iniciante do Kinesis Video Streams?

Se você é um usuário iniciante do Kinesis Video Streams, recomendamos que você leia as seções a seguir na ordem:

1. [Kinesis Video Streams: como funciona](#)— Para aprender sobre os conceitos do Kinesis Video Streams.
2. [Comece a usar o Amazon Kinesis Video Streams](#)— Para configurar sua conta e testar o Kinesis Video Streams.
3. [Bibliotecas do Kinesis Video Streams Producer](#)— Para saber mais sobre a criação de um aplicativo produtor do Kinesis Video Streams.
4. [Biblioteca de análise de streaming de vídeo do Kinesis](#)— Aprender sobre o processamento de quadros de dados recebidos em um aplicativo consumidor do Kinesis Video Streams.
5. [Exemplos do Amazon Kinesis Video Streams](#)— Para ver mais exemplos do que você pode fazer com o Kinesis Video Streams.

Requisitos de sistema do Kinesis Video Streams

As seções a seguir contêm requisitos de hardware, software e armazenamento para o Amazon Kinesis Video Streams.

Tópicos

- [Requisitos de câmera](#)
- [Sistemas operacionais testados](#)
- [Requisitos de armazenamento do SDK](#)

Requisitos de câmera

As câmeras usadas para executar o SDK e amostras do Kinesis Video Streams Producer têm os seguintes requisitos de memória:

- A visualização do conteúdo do SDK requer 16 MB de memória.
- O exemplo de configuração padrão do aplicativo é 128 MiB de memória. Esse valor é apropriado para os produtores que têm boa conectividade de rede e nenhum requisito de armazenamento em buffer adicional. Se a conectividade de rede for ruim e mais armazenamento em buffer for necessário, você poderá calcular a necessidade de memória por segundo de armazenamento em buffer multiplicando a taxa de quadros por segundo pelo tamanho da memória do quadro. Para obter mais informações sobre alocação de memória, consulte [StorageInfo](#).

Recomendamos usar câmeras USB ou RTSP (Real Time Streaming Protocol) que codifiquem dados usando H.264, pois isso remove a carga de trabalho da codificação da CPU.

Atualmente, o aplicativo de demonstração não é compatível com o User Datagram Protocol (UDP) para streaming RTSP. Esse recurso será adicionado no futuro.

O Producer SDK é compatível com os tipos de câmeras a seguir:

- Câmeras da Web.
- Câmeras USB.
- Câmeras com codificação H.264 (preferencial).
- Câmeras sem codificação H.264.

- Módulo de câmera Raspberry Pi. Essa é a opção preferencial para dispositivos Raspberry Pi, porque ele se conecta à GPU para transferência de dados de vídeo e, portanto, não ocorre sobrecarga no processamento da CPU.
- Câmeras RTSP (rede). Essas câmeras são preferenciais, porque os fluxos de vídeo já estão codificados com H.264.

Sistemas operacionais testados

Testamos câmeras da Web e câmeras RTSP com os dispositivos e sistemas operacionais a seguir:

- Mac mini
 - High Sierra
- MacBook Notebooks profissionais
 - Sierra (10.12)
 - El Capitan (10.11)
- Laptops HP com Ubuntu 16.04 em execução
- Ubuntu 17.10 (contêiner do Docker)
- Raspberry Pi 3

Requisitos de armazenamento do SDK

A instalação do [Bibliotecas do Kinesis Video Streams Producer](#) tem um requisito de armazenamento mínimo de 170 MB e um requisito de armazenamento recomendado de 512 MB.

Kinesis Video Streams: como funciona

Tópicos

- [Suporte à API Kinesis Video Streams e às bibliotecas de produtores](#)
- [Reprodução do Kinesis Video Streams](#)
- [Usando metadados de streaming com o Kinesis Video Streams](#)
- [Modelo de dados do Kinesis Video Streams](#)

Você pode usar o Amazon Kinesis Video Streams, um serviço AWS service (Serviço da AWS) totalmente gerenciado, para transmitir vídeo ao vivo de dispositivos para o e Nuvem AWS armazená-lo de forma durável. Em seguida, você poderá criar seus próprios aplicativos para processamento de vídeo em tempo real ou realizar análises de vídeo orientadas por lotes.

O diagrama a seguir fornece uma visão geral de como o Kinesis Video Streams funciona.

O diagrama demonstra a interação entre os seguintes componentes:

- Produtor — qualquer fonte que coloque dados em um stream de vídeo do Kinesis. Um produtor pode ser qualquer dispositivo gerador de vídeo, como uma câmera de segurança, uma câmera usada no corpo, uma câmera de smartphone ou uma câmera de painel. Um produtor também pode enviar dados diferentes de vídeo, como feeds de áudio, imagens ou dados RADAR.

Um único produtor pode gerar um ou mais streamings de vídeo. Por exemplo, uma câmera de vídeo pode enviar dados de vídeo para um stream de vídeo do Kinesis e dados de áudio para outro.

- Bibliotecas do Kinesis Video Streams Producer — Um conjunto de software e bibliotecas que você pode instalar e configurar em seus dispositivos. Você pode usar essas bibliotecas para se conectar com segurança e transmitir vídeos de forma confiável de diferentes maneiras, inclusive em tempo real, depois de armazená-los em buffer por alguns segundos ou como upload de mídia. after-the-fact
- Stream de vídeo do Kinesis — um recurso que você pode usar para transportar dados de vídeo ao vivo, armazená-los opcionalmente e disponibilizá-los para consumo em tempo real e em lote ou uma única vez. Em uma configuração típica, um stream de vídeo do Kinesis tem apenas um produtor publicando dados nele.

O streaming pode transportar áudio, vídeo e streams semelhantes de dados codificados por tempo, como feeds de detecção de profundidade, feeds RADAR e muito mais. Você cria um stream de vídeo do Kinesis usando AWS Management Console ou programaticamente usando os SDKs. AWS

Vários aplicativos independentes podem consumir um stream de vídeo do Kinesis em paralelo.

- Consumidor — obtém dados, como fragmentos e quadros, de um stream de vídeo do Kinesis para visualizá-lo, processá-lo ou analisá-lo. Geralmente, esses consumidores são chamados de aplicativos Kinesis Video Streams. Você pode escrever aplicativos que consomem e processam dados no Kinesis Video Streams em tempo real ou depois que os dados são armazenados e indexados por tempo, quando o processamento de baixa latência não é necessário. Você pode criar esses aplicativos de consumo para serem executados em instâncias do Amazon EC2.
- [Biblioteca de análise de streaming de vídeo do Kinesis](#) — Permite que os aplicativos do Kinesis Video Streams obtenham mídia do stream de vídeo do Kinesis de forma confiável e com baixa latência. Além disso, ele analisa os limites do quadro na mídia para que os aplicativos possam se concentrar no processamento e na análise dos quadros em si.

Suporte à API Kinesis Video Streams e às bibliotecas de produtores

O Kinesis Video Streams fornece APIs para você criar e gerenciar streams e ler ou gravar dados de mídia de e para um stream. O console do Kinesis Video Streams, além da funcionalidade de administração, também oferece suporte video-on-demand ao vivo e à reprodução. O Kinesis Video Streams também fornece um conjunto de bibliotecas de produtores que você pode usar no código do seu aplicativo para extrair dados de suas fontes de mídia e fazer o upload para o stream de vídeo do Kinesis.

Tópicos

- [API Kinesis Video Streams](#)
- [Padrão de descoberta de endpoints](#)
- [Bibliotecas de produtores](#)

API Kinesis Video Streams

O Kinesis Video Streams fornece APIs para criar e gerenciar o Kinesis Video Streams. Ele também fornece APIs para ler e gravar dados de mídia em um streaming, da seguinte maneira:

- API do produtor — O Kinesis Video Streams PutMedia fornece uma API para gravar dados de mídia em um stream de vídeo do Kinesis. Em uma solicitação PutMedia, o produtor envia um streaming de fragmentos de mídia. Um fragmento é uma sequência independente de quadros. Os quadros que pertencem a um fragmento não devem depender de nenhum quadro de outros fragmentos. Para ter mais informações, consulte [PutMedia](#).

À medida que os fragmentos chegam, o Kinesis Video Streams atribui um número de fragmento exclusivo, em ordem crescente. Ele também armazena registros de data e hora do lado do produtor e do lado do servidor para cada fragmento, como metadados específicos do Kinesis Video Streams.

- APIs de consumidor — Os consumidores podem usar as seguintes APIs para obter dados de um stream:
 - GetMedia - Ao usar esta API, os consumidores devem identificar o fragmento inicial. A API retorna fragmentos na ordem em que foram adicionados ao streaming (em ordem crescente por número de fragmento). Os dados de mídia nos fragmentos são empacotados em um formato estruturado, como [Matroska \(MKV\)](#). Para ter mais informações, consulte [GetMedia](#).

Note

GetMedia sabe onde os fragmentos estão (arquivados no armazenamento de dados ou disponíveis em tempo real). Por exemplo, se GetMedia determinar que o fragmento inicial está arquivado, ele começará a retornar fragmentos do armazenamento de dados. Quando ele precisa retornar fragmentos mais novos que ainda não foram arquivados, GetMedia muda para a leitura de fragmentos de um buffer de fluxo na memória.

Este é um exemplo de um consumidor contínuo, que processa fragmentos na ordem em que eles são ingeridos pelo streaming.

GetMedia permite que os aplicativos de processamento de vídeo falhem ou atrasem e, em seguida, recuperem o atraso sem esforço adicional. Ao usar GetMedia, os aplicativos podem processar dados que estão arquivados no armazenamento de dados e, à medida que o

aplicativo recupera o atraso, o `GetMedia` continua alimentando os dados de mídia em tempo real assim que eles chegam.

- `GetMediaFromFragmentList` (e `ListFragments`) – aplicativos de processamento em lote são considerados consumidores off-line. Os consumidores off-line podem optar por obter explicitamente fragmentos de mídia específicos ou intervalos de vídeo ao combinar as APIs `ListFragments` e `GetMediaFromFragmentList`. `ListFragments` e `GetMediaFromFragmentList` permitem que um aplicativo identifique segmentos de vídeo de um período ou um intervalo de fragmentos específico e, em seguida, obtenha esses fragmentos de forma sequencial ou em paralelo para fins de processamento. Esta abordagem é adequada para pacotes de aplicativos MapReduce, que precisam processar rapidamente grandes quantidades de dados em paralelo.

Por exemplo, suponha que um consumidor queira processar fragmentos de vídeo de um dia. O consumidor deve fazer o seguinte:

1. Obter uma lista de fragmentos ao chamar a API `ListFragments` e especificar um período para selecionar a coleção de fragmentos desejada.

A API retorna metadados de todos os fragmentos no período especificado. Os metadados fornecem informações como número do fragmento, registros de data e hora do lado do produtor e do lado do servidor, etc.

2. Obter a lista de metadados do fragmento e recuperar os fragmentos, em qualquer ordem. Por exemplo, para processar todos os fragmentos do dia, o consumidor pode optar por dividir a lista em sublistas e fazer com que os trabalhadores (por exemplo, várias instâncias do Amazon EC2) busquem os fragmentos em paralelo usando o e `GetMediaFromFragmentList` os processem em paralelo.

O diagrama a seguir mostra o streaming de dados para fragmentos e blocos durante essas chamadas de API.

Quando um produtor envia uma solicitação `PutMedia`, ele envia metadados de mídia para a carga útil e, em seguida, uma sequência de fragmentos de dados de mídia. Ao receber os dados, o Kinesis Video Streams armazena os dados de mídia recebidos como partes do Kinesis Video Streams. Cada bloco consiste no seguinte:

- Uma cópia dos metadados da mídia
- Um fragmento

- Metadados específicos do Kinesis Video Streams; por exemplo, o número do fragmento e os carimbos de data/hora do lado do servidor e do produtor

Quando um consumidor solicita metadados de mídia, o Kinesis Video Streams retorna um stream de partes, começando com o número do fragmento que você especifica na solicitação.

Se você ativar a persistência de dados para o stream, depois de receber um fragmento no stream, o Kinesis Video Streams também salvará uma cópia do fragmento no armazenamento de dados.

Padrão de descoberta de endpoints

APIs REST do plano de controle

Para acessar as APIs REST do [plano de controle do Kinesis Video Streams](#), use os endpoints do serviço [Kinesis Video Streams](#).

APIs REST do plano de dados

O Kinesis Video Streams foi criado [usando uma](#) arquitetura celular para garantir melhores propriedades de escalabilidade e isolamento de tráfego. Como cada stream é mapeado para uma célula específica em uma região, seu aplicativo deve usar os endpoints corretos específicos da célula para os quais seu stream foi mapeado. Ao acessar as APIs REST do Data Plane, você mesmo precisará gerenciar e mapear os endpoints corretos. Esse processo, o padrão de descoberta de endpoints, é descrito abaixo:

1. O padrão de descoberta do endpoint começa com uma chamada para uma das `GetEndpoints` ações. Essas ações pertencem ao Plano de Controle.
 1. Se você estiver recuperando os endpoints dos [the section called “Mídia arquivada do Amazon Kinesis Video Streams”](#) serviços [the section called “Mídia do Amazon Kinesis Video Streams”](#) ou, use. [the section called “GetDataEndpoint”](#)
 2. Se você estiver recuperando os endpoints para [the section called “Amazon Kinesis Video Streams”](#), [the section called “Amazon Kinesis WebRTC Streams”](#), ou [Kinesis Video Signaling](#), use. [the section called “GetSignalingChannelEndpoint”](#)
2. Armazene em cache e reutilize o endpoint.
3. Se o endpoint em cache não funcionar mais, faça uma nova chamada `GetEndpoints` para atualizar o endpoint.

Bibliotecas de produtores

Depois de criar um stream de vídeo do Kinesis, você pode começar a enviar dados para o stream. No código do seu aplicativo, você pode usar essas bibliotecas para extrair dados de suas fontes de mídia e fazer o upload para o stream de vídeo do Kinesis. Para obter mais informações sobre bibliotecas de produtores, consulte [Bibliotecas do Kinesis Video Streams Producer](#).

Reprodução do Kinesis Video Streams

Você pode ver um stream de vídeo do Kinesis usando os seguintes métodos:

- **GetMedia**— Você pode usar a GetMedia API para criar seus próprios aplicativos para processar o Kinesis Video Streams. GetMedia é uma API em tempo real com baixa latência. Para criar um player que use GetMedia, você mesmo deve construí-lo. Para obter informações sobre como desenvolver um aplicativo que exiba um stream de vídeo do Kinesis usando GetMedia, consulte [Biblioteca do analisador de streaming](#)
- **HLS** — [HTTP Live Streaming \(HLS\)](#) é um protocolo de comunicação de streaming de mídia baseado em HTTP padrão do setor. Você pode usar o HLS para ver um stream de vídeo do Kinesis, seja para reprodução ao vivo ou para ver vídeos arquivados.

Você pode usar HLS para reprodução ao vivo. A latência geralmente é de 3 a 5 segundos, mas pode ser de 1 a 10 segundos, dependendo do caso de uso, do player e das condições da rede. Você pode usar um player de terceiros (como o [Video.js](#) ou o [Google Shaka Player](#)) para exibir o streaming de vídeo fornecendo a URL da sessão de streaming HLS de forma programática ou manual. Você também pode reproduzir vídeo inserindo o URL da sessão de streaming HLS na barra de localização dos navegadores [Apple Safari](#) ou [Microsoft Edge](#).

- **MPEG-DASH** — Dynamic [Adaptive Streaming over HTTP \(DASH\)](#), também conhecido como MPEG-DASH, é um protocolo de streaming de taxa de bits adaptável que permite streaming de alta qualidade de conteúdo de mídia pela Internet fornecido por servidores web HTTP convencionais.

Você pode usar MPEG-DASH para reprodução ao vivo. A latência geralmente é de 3 a 5 segundos, mas pode ser de 1 a 10 segundos, dependendo do caso de uso, do player e das condições da rede. Você pode usar um player de terceiros (como [dash.js](#) ou [Google Shaka Player](#)) para exibir o stream de vídeo fornecendo o URL da sessão de streaming MPEG-DASH, de forma programática ou manual.

- **GetClip**— Você pode usar a `GetClip` API para baixar um clipe (em um arquivo MP4) contendo a mídia arquivada sob demanda do stream de vídeo especificado durante o intervalo de tempo especificado. Para obter mais informações, consulte a Referência [GetClip](#) da API.

Tópicos

- [Requisitos de faixa de reprodução de vídeo](#)
- [Reprodução de vídeo com HLS](#)
- [Reprodução de vídeo com MPEG-DASH](#)

Requisitos de faixa de reprodução de vídeo

O Amazon Kinesis Video Streams suporta mídia codificada em vários formatos. Se seu stream de vídeo do Kinesis usa um formato não compatível com uma das quatro APIs listadas abaixo, use [GetMedia](#) ou [GetMediaForFragmentList](#), pois elas não têm limitações de tipo de faixa.

Tópicos

- [GetClip requisitos](#)
- [Requisitos de URL do GetDash StreamingSession](#)
- [Requisitos de URL do GetHLS StreamingSession](#)
- [GetImages requisitos](#)

GetClip requisitos

Para obter mais informações sobre essa API, consulte [GetClip](#).

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.264	V_MPEG/ISO/AVC	N/D	N/D
Vídeo H.264	V_MPEG/ISO/AVC	Áudio AAC	A_AAC
Vídeo H.264	V_MPEG/ISO/AVC	Áudio G.711 (somente A-Law)	A_MS/ACM
Vídeo H.265	V_MPEGH/ISO/HEVC	N/D	N/D

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.265	V_MPEGH/ISO/HEVC	Áudio AAC	A_AAC

Important

Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de codificação, que são necessárias para decodificar adequadamente o fragmento. Não há suporte para alterações de CPD entre os fragmentos de destino do clipe resultante. O CPD deve permanecer consistente na mídia consultada, caso contrário, um erro será retornado.

Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. Um erro será retornado se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.

Requisitos de URL do GetDash StreamingSession

Para obter mais informações sobre essa API, consulte [GetDASHStreamingSessionURL](#).

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.264	V_MPEG/ISO/AVC	N/D	N/D
Vídeo H.264	V_MPEG/ISO/AVC	Áudio AAC	A_AAC
Vídeo H.264	V_MPEG/ISO/AVC	Áudio G.711 (somente A-Law)	A_MS/ACM
Vídeo H.264	V_MPEG/ISO/AVC	Áudio G.711 (somente U-Law)	A_MS/ACM

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Áudio AAC	A_AAC	N/D	N/D
Vídeo H.265	V_MPEGH/ISO/HEVC	N/D	N/D
Vídeo H.265	V_MPEGH/ISO/HEVC	Áudio AAC	A_AAC

Important

Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de codificação, que são necessárias para decodificar adequadamente o fragmento. As alterações de CPD não são suportadas durante uma sessão de streaming. O CPD deve permanecer consistente por meio da mídia consultada.

Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. O streaming falhará se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.

Requisitos de URL do GetHLS StreamingSession

Para obter mais informações sobre essa API, consulte [GetHLSStreamingSessionURL](#).

HLS Mp4

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.264	V_MPEG/ISO/AVC	N/D	N/D
Vídeo H.264	V_MPEG/ISO/AVC	Áudio AAC	A_AAC
Áudio AAC	A_AAC	N/D	N/D

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.265	V_MPEGH/ISO/HEVC	N/D	N/D
Vídeo H.265	V_MPEGH/ISO/HEVC	Áudio AAC	A_AAC

HLS TS

Descrição da faixa 1	ID do codec Track 1	Descrição da faixa 2	ID do codec Track 2
Vídeo H.264	V_MPEG/ISO/AVC	N/D	N/D
Vídeo H.264	V_MPEG/ISO/AVC	Áudio AAC	A_AAC
Áudio AAC	A_AAC	N/D	N/D

Note

Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de codificação, que são necessárias para decodificar adequadamente o fragmento. Tanto para TS quanto para MP4, as alterações de CPD são suportadas durante uma sessão de streaming. Portanto, os fragmentos em uma sessão podem ter informações diferentes no CPD sem interromper a reprodução. Para cada sessão de streaming, apenas 500 alterações de CPD são permitidas.

Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. O streaming falhará se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.

GetImages requisitos

Para obter mais informações sobre essa API, consulte [GetImages](#).

Note

A GetImages mídia deve conter uma faixa de vídeo na faixa 1.

Reprodução de vídeo com HLS

O [HTTP Live Streaming \(HLS\)](#) é um protocolo de comunicação de streaming de mídia baseado em HTTP padrão do setor. Você pode usar o HLS para ver um stream de vídeo do Kinesis, seja para reprodução ao vivo ou para ver vídeos arquivados.

Você pode usar HLS para reprodução ao vivo. A latência geralmente é de 3 a 5 segundos, mas pode ser de 1 a 10 segundos, dependendo do caso de uso, do player e das condições da rede. Você pode usar um player de terceiros (como o [Video.js](#) ou o [Google Shaka Player](#)) para exibir o streaming de vídeo fornecendo a URL da sessão de streaming HLS de forma programática ou manual. Você também pode reproduzir vídeo inserindo o URL da sessão de streaming HLS na barra de localização dos navegadores [Apple Safari](#) ou [Microsoft Edge](#).

[Para ver um stream de vídeo do Kinesis usando o HLS, primeiro crie uma sessão de streaming usando o URL do GetHLS. StreamingSession](#) Essa ação retorna uma URL (contendo um token de sessão) para acessar a sessão HLS. Em seguida, você pode usar a URL em um player de mídia ou em um aplicativo autônomo para exibir o streaming.

Important

Nem todas as mídias enviadas para o Kinesis Video Streams podem ser reproduzidas por meio do HLS. Consulte [the section called “GetHLSStreamingSessionURL”](#) os requisitos específicos de carregamento.

Tópicos

- [Use o AWS CLI para recuperar um URL de sessão de streaming HLS](#)
- [Exemplo: Use HLS em HTML e JavaScript](#)
- [Solução de problemas de HLS](#)

Use o AWS CLI para recuperar um URL de sessão de streaming HLS

Siga o procedimento abaixo para usar o AWS CLI para gerar um URL de sessão de streaming HLS para um stream de vídeo do Kinesis.

Para obter instruções de instalação, consulte o [Guia AWS Command Line Interface do usuário](#). Após a instalação, [configure o AWS CLI](#) com credenciais e região.

Como alternativa, abra o AWS CloudShell terminal, que tem o AWS CLI instalado e configurado. Consulte o [Guia do usuário do AWS CloudShell](#) para obter mais informações.

Recupere o endpoint de URL do HLS para seu stream de vídeo do Kinesis.

1. Digite o seguinte no terminal:

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

Você receberá uma resposta parecida com esta:

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

2. Faça a solicitação de URL da sessão de streaming HLS para esse endpoint retornado.

Live

Para reprodução ao vivo, a playlist de mídia HLS é continuamente atualizada com a mídia mais recente à medida que ela é disponibilizada. Quando você reproduz esse tipo de sessão em um reprodutor de mídia, a interface do usuário normalmente exibe uma notificação “ao vivo”, sem controle manual para escolher a posição na janela de reprodução a ser exibida.

Certifique-se de fazer upload de mídia para esse stream ao executar esse comando.

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

Live replay

Para reprodução ao vivo, a reprodução começa a partir de um horário de início especificado. A lista de reprodução de mídia do HLS também é atualizada continuamente com as mídias mais recentes à medida que são disponibilizadas. A sessão continuará a incluir mídia recém-ingrida até que a sessão expire ou até o horário de término especificado, o que ocorrer primeiro. Esse modo é útil para poder iniciar a reprodução a partir da detecção de um evento e continuar transmitindo ao vivo a mídia que ainda não foi ingerida no momento da criação da sessão.

Determine um carimbo de data e hora de início.

Neste exemplo, usamos o formato de hora em segundos do Unix Epoch. Consulte a seção Carimbos de [data e hora no Guia do AWS Command Line Interface usuário para obter mais informações sobre a formatação de carimbos](#) de data e hora.

Consulte [UnixTime.org](https://unixtime.org) para ver uma ferramenta de conversão.

- 1708471800 é igual a 20 de fevereiro de 2024 3:30:00 PM GMT- 08:00

Neste exemplo, não especificamos uma data e hora de término, o que significa que a sessão continuará a incluir mídia recém-ingrida até que a sessão expire.

Invoque a `GetHLSStreamingSessionURL` API com o modo de `LIVE_REPLAY` reprodução e um seletor 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 reprodução sob demanda, a playlist de mídia HLS contém a mídia especificada pelo seletor de fragmentos HLS. Quando esse tipo de sessão é reproduzido em um reprodutor de

mídia, a interface do usuário normalmente exibe um controle de depuração para escolher a posição na janela de reprodução a ser exibida.

Para criar um URL para uma determinada seção do stream, primeiro determine os carimbos de data e hora de início e término.

Neste exemplo, usamos o formato de hora em segundos do Unix Epoch. Consulte a seção Carimbos de [data e hora no Guia do AWS Command Line Interface usuário para obter mais informações sobre a formatação de carimbos](#) de data e hora.

Consulte [UnixTime.org](https://unixtime.org) para ver uma ferramenta de conversão.

- 1708471800 é igual a 20 de fevereiro de 2024 3:30:00 PM GMT- 08:00
- 1708471860 é igual a 20 de fevereiro de 2024 3:31:00 PM GMT- 08:00

Invoque a `GetHLSStreamingSessionURL` API com o modo de `ON_DEMAND` reprodução e um seletor 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, EndTime
```

Note

Os carimbos de data/hora devem estar dentro de 24 horas um do outro, conforme mencionado na documentação. [the section called "HLSTimestampRange"](#)

Você receberá uma resposta parecida com esta:

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

⚠ Important

Não compartilhe nem armazene esse token onde uma entidade não autorizada possa acessá-lo. O token fornece acesso ao conteúdo do stream. Proteja o token com as mesmas medidas que você usaria com suas AWS credenciais.

Você pode usar esse URL e qualquer player do HLS para ver o stream do HLS.

Por exemplo, use o reprodutor de mídia VLC.

Você também pode reproduzir o HLS Stream inserindo o URL da sessão de streaming HLS na barra de localização dos navegadores Apple Safari ou Microsoft Edge.

Exemplo: Use HLS em HTML e JavaScript

O exemplo a seguir mostra como usar o AWS SDK para JavaScript v2 para recuperar uma sessão de streaming HLS para um stream de vídeo do Kinesis e reproduzi-la em uma página da web. O exemplo mostra como reproduzir vídeo nos seguintes players:

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

Veja o [código de exemplo completo](#) e a [página da web hospedada](#) em GitHub.

Tópicos explicativos sobre códigos:

- [Importe o AWS SDK JavaScript para navegadores](#)
- [Configurar o cliente Kinesis Video Streams](#)
- [Recupere o endpoint para reprodução HLS](#)
- [Configurar o cliente de mídia arquivada Kinesis Video Streams](#)
- [Recupere o URL da sessão de streaming do HLS](#)
- [Exibir o fluxo HLS na página da web](#)

Importe o AWS SDK JavaScript para navegadores

Na página da web, inclua a seguinte tag de script para importar o AWS SDK para JavaScript v2 no projeto.

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

Para obter mais informações, consulte o [AWS SDK para obter](#) a JavaScript documentação.

Configurar o cliente Kinesis Video Streams

Para acessar o streaming de vídeo com o HLS, primeiro crie e configure o cliente Kinesis Video Streams. Consulte [Definindo credenciais em um navegador da Web](#) para ver outros métodos de autenticação.

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

O aplicativo recupera os valores necessários das caixas de entrada na página HTML.

Recupere o endpoint para reprodução HLS

Use o cliente Kinesis Video Streams para [the section called "GetDataEndpoint"](#) invocar a API para recuperar o endpoint.

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

Esse código armazena o endpoint na `hlsDataEndpoint` variável.

Configurar o cliente de mídia arquivada Kinesis Video Streams

Na configuração do cliente de mídia arquivada do Kinesis Video Streams, especifique o endpoint que você obteve na etapa anterior.

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

Recupere o URL da sessão de streaming do HLS

Use o cliente de mídia arquivada Kinesis Video Streams para [the section called “GetHLSStreamingSessionURL”](#) invocar a API e recuperar o URL de reprodução do HLS.

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

Exibir o fluxo HLS na página da web

Quando tiver a URL da sessão de streaming HLS, forneça-a para o player de vídeo. O método para fornecer a URL ao player de vídeo é específico ao player usado.

Video.js

Faça o seguinte para importar o [Video.js](#) e suas classes CSS para o script do nosso 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>
```

Crie um elemento vídeo HTML para exibir o vídeo:

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

Defina o URL do HLS como a fonte do elemento de vídeo HTML:

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

Shaka

Faça o seguinte para importar o [player Google Shaka](#) para o script do nosso navegador:

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

Crie um elemento vídeo HTML para exibir o vídeo:

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

Crie um player Shaka especificando o elemento de vídeo e chame o método load:

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

hls.js

Faça o seguinte para importar o [hls.js para o](#) script do nosso navegador:

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

Crie um elemento vídeo HTML para exibir o vídeo:

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

Crie um player `hls.js`, forneça a URL do HLS e peça para ele jogar:

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

Solução de problemas de HLS

Esta seção descreve problemas que você pode encontrar ao usar HTTP Live Streaming (HLS) com o Kinesis Video Streams.

Problemas

- [A recuperação do URL da sessão de streaming HLS foi bem-sucedida, mas a reprodução no player de vídeo apresentou falha](#)
- [Latência muito alta entre o produtor e o player](#)

A recuperação do URL da sessão de streaming HLS foi bem-sucedida, mas a reprodução no player de vídeo apresentou falha

Essa situação ocorre quando o URL de uma sessão de streaming HLS pode ser recuperada com êxito usando `GetHLSStreamingSessionURL`, mas a reprodução do vídeo falha quando o URL é fornecido a um player de vídeo.

Para solucionar essa situação, tente o seguinte:

- Determine se o stream de vídeo é reproduzido no console do Kinesis Video Streams. Considere qualquer erro que o console mostra.
- Se a duração de fragmento for menor que um segundo, aumente-a para um segundo. Se a duração do fragmento for muito curta, o serviço poderá restringir o player porque ele está fazendo solicitações de fragmentos de vídeo com muita frequência.
- Verifique se a URL de cada sessão de streaming HLS está sendo usada por apenas um player. Se mais de um player estiver usando um único URL de sessão de streaming HLS, o serviço poderá receber muitas solicitações e limitá-las.

- Verifique se o player é compatível com todas as opções que você está especificando para a sessão de streaming do HLS. Experimente diferentes combinações de valores para os seguintes parâmetros:
 - `ContainerFormat`
 - `PlaybackMode`
 - `FragmentSelectorType`
 - `DiscontinuityMode`
 - `MaxMediaPlaylistFragmentResults`

Alguns players de mídia (como HTML5 e players móveis) normalmente oferecem suporte apenas a HLS com o formato de contêiner fMP4. Outros reprodutores de mídia (como Flash e reprodutores personalizados) podem suportar apenas HLS com o formato de contêiner MPEG TS. Recomendamos experimentar o `ContainerFormat` parâmetro para iniciar a solução de problemas.

- Verifique se cada fragmento tem um número consistente de trilhas. Verifique se os fragmentos no stream não estão mudando entre ter uma faixa de áudio e vídeo e somente uma faixa de vídeo. Verifique também se as configurações do codificador (resolução e taxa de quadros) não estão mudando entre os fragmentos em cada faixa.

Latência muito alta entre o produtor e o player

Essa situação ocorre quando a latência é muito alta em relação a quando o vídeo é capturado e quando é reproduzido no player de vídeo.

O vídeo é reproduzido por meio de HLS por fragmento. Portanto, a latência não pode ser menor que a duração do fragmento. A latência também inclui o tempo necessário para o armazenamento em buffer e a transferência de dados. Se a sua solução exigir latência menor que um segundo, considere usar a API `GetMedia`.

É possível ajustar os parâmetros a seguir para reduzir a latência geral, mas o ajuste desses parâmetros também pode reduzir a qualidade do vídeo ou aumentar a taxa de rearmazenamento em buffer.

- Duração do fragmento — A duração do fragmento é a quantidade de vídeo entre as divisões no stream, controlada pela frequência dos quadros-chave gerados pelo codificador de vídeo. O valor recomendado é um segundo. Ter uma duração de fragmento mais curta significa que menos tempo é gasto aguardando que o fragmento seja concluído para transmitir os dados do vídeo para

o serviço. Fragmentos mais curtos também são mais rápidos para o serviço processar. No entanto, se a duração de fragmento for muito curta, aumentará a probabilidade do player não ter conteúdo suficiente e precisar interromper e armazenar o conteúdo em buffer. Se a duração de fragmento for inferior a 500 milissegundos, o produtor poderá criar muitas solicitações, fazendo com que o serviço as limite.

- Taxa de bits — Um stream de vídeo com uma taxa de bits mais baixa leva menos tempo para ler, gravar e transmitir. No entanto, um streaming de vídeo com uma taxa de bits menor tem menor qualidade de vídeo.
- Contagem de fragmentos em playlists de mídia — Um player sensível à latência só deve carregar os fragmentos mais recentes em uma playlist de mídia. Em vez disso, a maioria dos jogadores começa com o fragmento mais antigo. Ao reduzir o número de fragmentos na playlist, você reduz o tempo de separação entre o fragmento anterior e o novo. Com um tamanho de playlist menor, é possível que um fragmento seja ignorado durante a reprodução se houver um atraso na adição de novos fragmentos à playlist ou se houver um atraso no player em obter uma playlist atualizada. Recomendamos usar de 3 a 5 fragmentos e usar um player configurado para carregar somente os fragmentos mais recentes de uma playlist.
- Tamanho do buffer do player — A maioria dos reprodutores de vídeo tem uma duração mínima configurável do buffer, geralmente com um padrão de 10 segundos. Para a latência mais baixa, o valor pode ser definido como 0 segundos. No entanto, isso significa que o player rebuffera se houver algum atraso na produção de fragmentos, pois o player não terá buffer para absorver o atraso.
- Player “catch up” — Os reprodutores de vídeo normalmente não capturam automaticamente a reprodução até a frente do buffer de vídeo se o buffer estiver cheio, como quando um fragmento atrasado causa a reprodução de um acúmulo de fragmentos. Um player personalizado pode evitar isso descartando quadros ou aumentando a velocidade de reprodução (por exemplo para 1.1x) para alcançar a frente do buffer. Isso faz com que a reprodução fique instável ou que a velocidade aumente conforme o player faz a correção, e o rearmazenamento em buffer pode ser mais frequente como o tamanho do buffer permanece curto.

Reprodução de vídeo com MPEG-DASH

[Para ver um stream de vídeo do Kinesis usando MPEG-DASH, primeiro você cria uma sessão de streaming usando GetDash URL. `StreamingSession`](#) Essa ação retorna uma URL (contendo um token de sessão) para acessar a sessão MPEG-DASH. Em seguida, você pode usar a URL em um player de mídia ou em um aplicativo autônomo para exibir o streaming.

Um stream de vídeo do Amazon Kinesis tem os seguintes requisitos para fornecer vídeo por meio do MPEG-DASH:

- Para obter os requisitos da faixa de reprodução de vídeo em streaming, consulte [the section called “Obter URL do Dash StreamingSession”](#).
- A retenção de dados deve ser maior que 0.
- A trilha de vídeo de cada fragmento deve conter dados privados de codec na Advanced Video Coding (AVC – Codificação de vídeo avançada) para H.264, e HEVC para formato H.265. Para obter mais informações, consulte [Especificação de MPEG-4 ISO/IEC 14496-15](#). Para obter informações sobre a adaptação de dados de streaming para um formato específico, consulte [Sinalizadores de adaptação do NAL](#).
- A trilha de áudio (se houver) de cada fragmento deve conter os dados privados de codec no formato AAC ([especificação de AAC ISO/IEC 13818-7](#)) ou o [formato MS Wave](#).

Exemplo: Usando MPEG-DASH em HTML e JavaScript

O exemplo a seguir mostra como recuperar uma sessão de streaming MPEG-DASH para um stream de vídeo do Kinesis e reproduzi-la em uma página da Web. O exemplo mostra como reproduzir vídeo nos seguintes players:

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

Tópicos

- [Configurar o cliente Kinesis Video Streams para reprodução de MPEG-DASH](#)
- [Recupere o endpoint de conteúdo arquivado do Kinesis Video Streams para reprodução de MPEG-DASH](#)
- [Recuperar a URL de sessão de streaming MPEG-DASH](#)
- [Exibir o streaming de vídeo com a reprodução MPEG-DASH](#)
- [Exemplo concluído](#)

Configurar o cliente Kinesis Video Streams para reprodução de MPEG-DASH

Para acessar o streaming de vídeo com o MPEG-DASH, primeiro crie e configure o cliente Kinesis Video Streams (para recuperar o endpoint do serviço) e o cliente de mídia arquivada (para recuperar

a sessão de streaming MPEG-DASH). O aplicativo recupera os valores necessários das caixas de entrada na 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 o endpoint de conteúdo arquivado do Kinesis Video Streams para reprodução de MPEG-DASH

Depois que os clientes forem iniciados, recupere o endpoint de conteúdo arquivado do Kinesis Video Streams para que você possa recuperar o URL da sessão de streaming MPEG-DASH da seguinte forma:

```
// 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);
```

Recuperar a URL de sessão de streaming MPEG-DASH

Quando você tiver o endpoint de conteúdo arquivado, chame a API [GetDash StreamingSession URL para recuperar o URL da sessão](#) de streaming MPEG-DASH da seguinte forma:

```
// 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);
    });
}
```

Exibir o streaming de vídeo com a reprodução MPEG-DASH

Quando tiver a URL da sessão de streaming MPEG-DASH, forneça-a para o player de vídeo. O método para fornecer a URL ao player de vídeo é específico ao player usado.

O código de exemplo a seguir mostra como fornecer a URL da sessão de streaming a um player [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');
}
```

O exemplo de código a seguir mostra como fornecer a URL da sessão de streaming para um player [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');
}
```

Exemplo concluído

Você pode [baixar ou visualizar o código de exemplo completo](#) em GitHub.

Usando metadados de streaming com o Kinesis Video Streams

Você pode usar o SDK do Amazon Kinesis Video Streams Producer para incorporar metadados no nível de fragmento individual em um stream de vídeo do Kinesis. Os metadados no Kinesis Video Streams são um par de valores-chave mutáveis. Você pode usá-lo para descrever o conteúdo do fragmento, incorporar leituras de sensores associadas que devem ser transferidas junto com o fragmento real ou atender a outras necessidades personalizadas. Os metadados ficam disponíveis como parte das operações de API [the section called “GetMedia”](#) ou [the section called “GetMediaForFragmentList”](#). Ele é armazenado junto com os fragmentos durante todo o período de retenção do stream. Seus aplicativos consumidores podem ler, processar e reagir com base nos metadados usando o [Biblioteca de análise de streaming de vídeo do Kinesis](#)

Há dois modos pelos quais os metadados podem ser incorporados aos fragmentos em um streaming:

- Não persistente — você pode afixar metadados de forma única ou ad hoc em fragmentos em um stream, com base nos critérios específicos da empresa que ocorreram. Um exemplo é uma câmera inteligente que detecta movimento e adiciona metadados aos fragmentos correspondentes que contêm o movimento antes de enviar os fragmentos para o stream de vídeo do Kinesis. Você pode aplicar metadados ao fragmento no seguinte formato: `Motion = true`
- Persistente — você pode afixar metadados em fragmentos sucessivos e consecutivos em um stream com base em uma necessidade contínua. Um exemplo é uma câmera inteligente que envia as coordenadas atuais de latitude e longitude associadas a todos os fragmentos enviados para o stream de vídeo do Kinesis. Você pode aplicar metadados a todos os fragmentos no seguinte formato: `Lat = 47.608013N , Long = -122.335167W`

É possível afixar metadados em ambos os modos ao mesmo fragmento simultaneamente com base nas necessidades do aplicativo. Os metadados incorporados podem incluir objetos detectados, atividade monitorada, coordenadas de GPS ou qualquer outro dado personalizado que você deseje associar aos fragmentos no streaming. Os metadados são codificados como pares de string chave-valor.

Tópicos

- [Adicionar metadados a um stream de vídeo do Kinesis](#)
- [Consumindo metadados incorporados em um stream de vídeo do Kinesis](#)
- [Limites de metadados de streaming](#)

Adicionar metadados a um stream de vídeo do Kinesis

Os metadados que você adiciona a um stream de vídeo do Kinesis são modelados como tags MKV, que são implementadas como pares de valores-chave.

Os metadados podem ser temporários, como para marcar um evento no streaming, ou persistentes, como para identificar fragmentos onde determinado evento está ocorrendo. Um item de metadados persistente permanece e é aplicado a cada fragmento consecutivo até ser cancelado.

Note

Os itens de metadados adicionados usando o [Bibliotecas de produtores](#) são distintos das APIs de marcação no nível do streaming implementadas com [the section called "TagStream"](#), [the section called "UntagStream"](#) e [the section called "ListTagsForStream"](#).

API de metadados de streaming

É possível usar as operações a seguir no Producer SDK para implementar metadados de streaming.

PIC

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

C++ Producer SDK

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

Java Producer SDK

Você pode usar o Java Producer SDK para adicionar metadados a um MediaSource usando: `MediaSourceSink.onCodecPrivateData`

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

Metadados persistentes e não persistentes

Para dados não persistentes, você pode adicionar vários itens de metadados com o mesmo nome. O Producer SDK coleta os itens de metadados na fila de metadados até que eles sejam colocados no início do próximo fragmento. A fila de metadados é apagada conforme os itens de metadados são aplicados ao streaming. Para repetir os metadados, chame `putKinesisVideoFragmentMetadata` ou `putFragmentMetadata` novamente.

Para metadados persistentes, o Producer SDK coleta os itens de metadados na fila de metadados da mesma maneira que faz com os metadados não persistentes. No entanto, os itens de metadados não são removidos da fila quando são anexados ao próximo fragmento.

Chamar `putKinesisVideoFragmentMetadata` ou `putFragmentMetadata` com `persistent` definido como `true` tem o seguinte comportamento:

- Chamar a API coloca o item de metadados na fila. Os metadados são adicionados como uma tag do MKV para cada fragmento enquanto o item está na fila.
- Chamar a API com o mesmo nome e um valor diferente como um item de metadados adicionado anteriormente substitui o item.
- Chamar a API com um valor vazio remove (cancela) o item de metadados da fila de metadados.

Consumindo metadados incorporados em um stream de vídeo do Kinesis

Para consumir os metadados em um stream de vídeo do Kinesis, use uma implementação 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");
    }
}
}
```

Esta interface é encontrada na classe [FragmentMetadataVisitor](#) no [Biblioteca de análise de streaming de vídeo do Kinesis](#).

A classe `FragmentMetadataVisitor` contém uma implementação 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();
    }
}
```

A classe `KinesisVideoRendererExample` contém um exemplo de como usar um `BasicMkvTagProcessor`. No seguinte exemplo, um `BasicMkvTagProcessor` é adicionado ao `MediaProcessingArguments` de um aplicativo:

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

O método `BasicMkvTagProcessor.process` é chamado quando metadados do fragmento chegam. É possível recuperar os metadados acumulados com `GetTags`. Para recuperar um único item de metadados, primeiro chame `clear` para limpar os metadados coletados e, em seguida, recupere os itens de metadados novamente.

Limites de metadados de streaming

Consulte [the section called “Fragmentar cotas de metadados”](#) para obter mais informações sobre os limites que se aplicam à adição de metadados de streaming a um stream de vídeo do Kinesis.

Modelo de dados do Kinesis Video Streams

As [Bibliotecas de produtores](#) e a [Biblioteca do analisador de streaming](#) enviam e recebem dados de vídeo em um formato que oferece suporte à incorporação de informações junto com os dados do vídeo. Este formato é baseado na especificação Matroska (MKV).

O [formato MKV](#) é uma especificação aberta para dados de mídia. Todas as bibliotecas e exemplos de código no Guia do desenvolvedor do Amazon Kinesis Video Streams enviam ou recebem dados no formato MKV.

[Bibliotecas do Kinesis Video Streams Producer](#) Ele usa os Frame tipos `StreamDefinition` e para produzir cabeçalhos de fluxo MKV, cabeçalhos de quadro e dados de quadro.

Para obter informações sobre a especificação completa do MKV, consulte [Especificações do Matroska](#).

As seções a seguir descrevem os componentes de dados formatados pelo MKV produzidos pela [Biblioteca de produtores C++](#).

Tópicos

- [Elementos do cabeçalho de streaming](#)
- [Transmita dados da trilha](#)
- [Elementos do cabeçalho de quadros](#)
- [Dados do quadro MKV](#)

Elementos do cabeçalho de streaming

Os seguintes elementos de cabeçalho do MKV são usados por `StreamDefinition` (definido no `StreamDefinition.h`).

Elemento	Descrição	Valores típicos
stream_name	Corresponde ao nome do stream de vídeo do Kinesis.	my-stream
retention_period	A duração, em horas, dos dados do stream é mantida pelo Kinesis Video Streams. Especifique 0 para um stream que não retém dados.	24
tags	uma coleção de valores-chave de dados do usuário. Esses dados são exibidos no AWS Management Console e podem ser lidos por aplicativos do cliente para filtrar ou obter informações sobre um streaming.	
kms_key_id	Se presente, a AWS KMS chave definida pelo usuário é usada para criptografar dados no stream. Se ausentes, os dados são criptografados pela chave () fornecida pelo Kinesis. aws/kinesis-video	01234567-89ab-cdef-0123-456789ab
tipo de streaming	Atualmente, o único tipo de streaming válido é STREAMING_TYPE_REALTIME .	STREAMING_TYPE_REALTIME
tipo de conteúdo	O tipo de conteúdo definido pelo usuário. Para que os dados de vídeo de streaming sejam reproduzidos no	vídeo/h264

Elemento	Descrição	Valores típicos
	console, o tipo de conteúdo precisa ser video/h264 .	
latência máxima	Esse valor não é usado atualmente e deve ser definido como 0.	0
fragment_duration	A estimativa de quanto tempo seus fragmentos devem ter, o que é usado para otimização. A duração real do fragmento é determinada por dados de streaming.	2
escala de código de hora	Indica a escala usada pelos carimbos de data/hora do quadro. O padrão é 1 milissegundo. Especificar 0 também atribui o valor padrão de 1 milissegundo. Esse valor pode estar entre 100 nanossegundos e 1 segundo. Para obter mais informações, consulte a TimecodeScale documentação do Matroska.	
key_frame_fragmentation	Se true, o streaming iniciará um novo cluster quando um keyframe for recebido.	true

Elemento	Descrição	Valores típicos
frame_timecodes	Se <code>true</code> , o Kinesis Video Streams usa os valores de data e hora de apresentação (pts) e de decodificação (dts) dos quadros recebidos. Se <code>false</code> , o Kinesis Video Streams carimba os quadros quando eles são recebidos com valores de tempo gerados pelo sistema.	true
absolute_fragment_time	Se <code>true</code> , os timecodes do cluster serão interpretados como se estivessem usando a hora absoluta (por exemplo, do relógio do sistema do produtor). Se for <code>false</code> , os códigos de hora do cluster serão interpretados como relativos à hora de início do streaming.	true
fragment_acks	Se <code>true</code> , confirmações (ACKs) são enviadas quando o Kinesis Video Streams recebe os dados. As ACKs podem ser recebidas usando os retornos de chamada <code>KinesisVideoStreamFragmentAck</code> ou <code>KinesisVideoStreamParseFragmentAck</code> .	true

Elemento	Descrição	Valores típicos
<code>restart_on_error</code>	Indica se o streaming deve retomar a transmissão depois que ocorrer um erro de streaming.	true
<code>nal_adaptation_flags</code>	Indica se a adaptação da NAL (camada de abstração de rede) ou os dados privados do codec estão presentes no conteúdo. Sinalizadores válidos incluem <code>NAL_ADAPTATION_ANNEXB_NALS</code> e <code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code> .	<code>NAL_ADAPTATION_ANNEXB_NALS</code>
<code>frame_rate</code>	Uma estimativa da taxa de quadros do conteúdo. Esse valor é usado para otimização. A taxa de quadros real é determinada pela taxa de dados recebidos. Especificar 0 atribui o padrão de 24.	24
<code>avg_bandwidth_bps</code>	Uma estimativa da largura de banda do conteúdo, em Mbps. Esse valor é usado para otimização. A taxa real é determinada pela largura de banda dos dados recebidos. Por exemplo, para um streaming de vídeo de resolução 720p em execução a 25 FPS, você pode esperar que a média de largura de banda seja 5 Mbps.	5

Elemento	Descrição	Valores típicos
buffer_duration	a duração de armazenam ento do conteúdo no produtor. Se houver baixa latência de rede, esse valor poderá ser reduzido. Se a latência da rede for alta, o aumento desse valor evita que os quadros sejam descartados antes que possam ser enviados, devido à falha na alocação ao colocar os quadros no buffer menor.	
replay_duration	A quantidade de tempo em que o fluxo de dados de vídeo é “rebobinado” se a conexão for perdida. Esse valor pode ser zero se os quadros perdidos devido à perda de conexão não forem uma preocupação. O valor pode ser aumentado se o aplicativo consumidor puder remover quadros redundantes. Esse valor deve ser menor que a duração do buffer, caso contrário, a duração do buffer será usada.	
connection_staleness	a duração de retenção de uma conexão quando nenhum dado é recebido.	

Elemento	Descrição	Valores típicos
codec_id	O codec usado pelo conteúdo. Para obter mais informações, consulte CodecID na Especificação do Matroska.	V_MPEG2
track_name	O nome de trilha definido pelo usuário.	my_track
codecPrivateData	os dados fornecidos pelo codificador utilizado para decodificar os dados do quadro, como a largura do quadro e a altura em pixels, o que é necessário para muitos consumidores de downstream. Na Biblioteca de produtores C++ , a matriz <code>gMkvTrackVideoBits</code> em <code>MkvStatics.cpp</code> inclui a largura e a altura em pixels do quadro.	
codecPrivateDataTamanho	O tamanho dos dados no parâmetro <code>codecPrivateData</code> .	
track_type	O tipo da trilha para o streaming.	MKV_TRACK_INFO_TYPE_AUDIO ou MKV_TRACK_INFO_TYPE_VIDEO
segment_uuid	UUID do segmento definido pelo usuário (16 bytes)	
default_track_id	Número exclusivo diferente de zero para a trilha.	1

Transmita dados da trilha

Os seguintes elementos da trilha do MKV são usados por `StreamDefinition` (definidos no `StreamDefinition.h`).

Elemento	Descrição	Valores típicos
<code>track_name</code>	Nome da trilha definido pelo usuário. Por exemplo, "audio" para a trilha de áudio.	audio
<code>codec_id</code>	ID de codec da trilha. Por exemplo, "A_AAC" para uma trilha de áudio.	A_AAC
<code>cpd</code>	Dados fornecidos pelo codificador usado para decodificar os dados do quadro. Esses dados podem incluir a largura e a altura do quadro em pixels, que são informações necessárias para muitos consumidores de downstream. Na C++ Producer Library , a <code>gMkvTrackVideoBits</code> matriz em <code>MkvStatics.cpp</code> inclui largura e altura em pixels para o quadro.	
<code>cpd_size</code>	O tamanho dos dados no <code>codecPrivateData</code> parâmetro.	
<code>track_type</code>	O tipo da trilha. Por exemplo, você pode usar um valor enum de <code>MKV_TRACK_INFO_TYPE_AUDIO</code> para áudio.	<code>MKV_TRACK_INFO_TYPE_AUDIO</code>

Elementos do cabeçalho de quadros

Os elementos de cabeçalho do MKV a seguir são usados por `FrameKinesisVideoPic` (definido no pacote `, em mkvgen/Include.h`):

- **Frame Index:** um valor monotonômico crescente.
- **Flags:** tipo de quadro. Entre os valores válidos estão os seguintes:
 - `FRAME_FLAGS_NONE`
 - `FRAME_FLAG_KEY_FRAME`: se `key_frame_fragmentation` estiver definido no streaming, os keyframes iniciarão um novo fragmento.
 - `FRAME_FLAG_DISCARDABLE_FRAME`: indica ao decodificador que ele pode descartar o quadro se a decodificação for lenta.
 - `FRAME_FLAG_INVISIBLE_FRAME`: a duração deste bloco é 0.
- **Carimbo de data e hora de decodificação:** o registro de data e hora de quando esse quadro foi decodificado. Se os quadros anteriores dependerem desse quadro para decodificação, esse carimbo de data/hora poderá ser anterior ao dos quadros anteriores. Esse valor é relativo ao início do fragmento.
- **Timestamp da apresentação:** O timestamp de quando esse quadro é exibido. Esse valor é relativo ao início do fragmento.
- **Duration:** a duração de reprodução do quadro.
- **Size:** o tamanho dos dados de quadro em bytes

Dados do quadro MKV

Os dados em `frame.frameData` podem conter apenas os dados do quadro ou conter mais informações de cabeçalho aninhadas, dependendo do esquema de codificação usado. Para serem exibidos no AWS Management Console, os dados devem ser codificados no codec [H.264](#), mas o Kinesis Video Streams pode receber fluxos de dados com serialização temporal em qualquer formato.

Comece a usar o Amazon Kinesis Video Streams

Esta seção descreve como realizar as seguintes tarefas no Amazon Kinesis Video Streams:

- Configure sua Conta da AWS e crie um administrador, caso ainda não tenha feito isso.
- Criar um streaming de vídeo do Kinesis.
- Envie dados da sua câmera para o stream de vídeo do Kinesis e visualize a mídia no console.

Se você é novato no Amazon Kinesis Video Streams, recomendamos que [Kinesis Video Streams: como funciona](#) você leia primeiro.

Note

Seguir a amostra de introdução não incorrerá em nenhuma cobrança para você Conta da AWS. Para saber os custos de dados em sua região, consulte os preços [do Amazon Kinesis Video Streams](#).

Tópicos

- [Configurar uma conta](#)
- [Crie um stream de vídeo do Kinesis](#)
- [Enviar dados para um stream de vídeo do Amazon Kinesis](#)
- [Consuma dados de mídia](#)

Configurar uma conta

Antes de usar o Amazon Kinesis Video Streams pela primeira vez, conclua as tarefas a seguir.

Tópicos

- [Inscreva-se para um Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)
- [Crie uma Conta da AWS chave](#)

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Signing in as the root user](#) (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

Crie uma Conta da AWS chave

Você precisará de uma Conta da AWS chave para acessar o Amazon Kinesis Video Streams de forma programática.

Para criar uma Conta da AWS chave, faça o seguinte:

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No barra de navegação, selecione Usuários e depois selecione o usuário Administrador.
3. Selecione a guia Credenciais de segurança e Criar chave de acesso.
4. Anote o ID da chave de acesso. Selecione Show (Mostrar) em Secret access key (Chave de acesso secreta). Anote a Secret access key (Chave de acesso secreta).

Crie um stream de vídeo do Kinesis

Esta seção descreve como criar um stream de vídeo do Kinesis.

Esta seção contém os seguintes procedimentos:

- [the section called “Crie um stream de vídeo usando o console”](#)
- [the section called “Crie um stream de vídeo usando o AWS CLI”](#)

Crie um stream de vídeo usando o console

1. Abra o console em <https://console.aws.amazon.com/kinesisvideo/home>.
2. Na página Video streams (Streamings de vídeo), selecione Create video stream (Criar streaming de vídeo).
3. Na página Criar uma nova transmissão de vídeo, insira *YourStreamName* o nome da transmissão. Deixe o botão Configuração padrão selecionado.
4. Escolha Criar streaming de vídeo.
5. Depois que o Amazon Kinesis Video Streams criar o stream, analise os detalhes YourStreamName na página.

Crie um stream de vídeo usando o AWS CLI

1. Verifique se você tem o AWS CLI instalado e configurado. Para obter mais informações, consulte a documentação do [AWS Command Line Interface](#).
2. Execute o comando Create-Stream na AWS CLI:

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

A resposta será semelhante à seguinte:

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

Enviar dados para um stream de vídeo do Amazon Kinesis

Esta seção descreve como enviar dados de mídia de uma câmera para o stream de vídeo do Kinesis que você criou na seção anterior. Esta seção usa o [Biblioteca de produtores C++](#) como um plug-in [Plugin do GStreamer - kvssink](#).

Para enviar mídia de vários dispositivos em vários sistemas operacionais, este tutorial usa a [biblioteca](#) de produtores de C++ do Kinesis Video Streams e o GStreamer, uma estrutura de mídia de código aberto que padroniza o acesso a câmeras e outras fontes de mídia.

Tópicos

- [Crie o SDK e as amostras](#)
- [Execute as amostras para fazer upload de mídia para o Kinesis Video Streams](#)
- [Revise os objetos de reconhecimento](#)

Crie o SDK e as amostras

Você pode criar o SDK e as amostras no seu computador ou no AWS Cloud9. Siga os procedimentos apropriados abaixo.

Build on your computer

Use as instruções no [arquivo readme](#) para criar a biblioteca do produtor e o aplicativo de amostra.

Isso inclui:

- Instalar as dependências

- Clonando o repositório
- Usando o CMake para gerar makefiles
- Construindo os arquivos binários usando make

Build in AWS Cloud9

Siga estes procedimentos para fazer o upload para o Kinesis Video AWS Cloud9 Streams em. Você não precisará baixar nada para o seu computador.

1. No AWS Management Console, abra [AWS Cloud9](#).

Selecione Criar ambiente.

2. Na tela Criar ambiente, faça o seguinte:
 - Nome - digite um nome para seu novo ambiente.
 - Plataforma - Selecione Ubuntu Server 22.04 LTS.

Você pode deixar os outros campos com as seleções padrão.

3. Quando o ambiente tiver sido criado, selecione Abrir na coluna Cloud9 IDE.

Na área central inferior da tela, você vê. Admin:~/environment \$ Este é o AWS Cloud9 terminal (Amazon EC2).

Note

Se você fechar acidentalmente o terminal, selecione Janela, Novo terminal.

Execute os seguintes comandos no terminal para alterar o volume para 20 GiB.

- a. Baixe o script .

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

- b. Conceda permissões de execução ao script.

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

c. Executar o script.

```
./resize_volume.sh
```

4. Obtenha as informações mais recentes sobre todos os softwares que você pode instalar ou atualizar por meio da Advanced Packaging Tool (APT).

Esse comando não atualiza o software em si, mas garante que seu sistema saiba quais são as versões mais recentes disponíveis.

```
sudo apt-get update
```

5. Instale as dependências do SDK do produtor 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. Use o git para clonar o SDK do produtor de C++.

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

7. Prepare um diretório de compilação.

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

8. Use o CMake para gerar makefiles.

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

O final da saída esperada tem a seguinte aparência:

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

- Use `make` para compilar o SDK e os aplicativos de amostra, bem como criar os executáveis finais.

```
make
```

O final da saída esperada tem a seguinte aparência:

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

- Confirme se os arquivos de amostra foram criados. Liste os arquivos no diretório atual:

```
ls
```

Confirme se os seguintes arquivos estão presentes:

- `kvs_gstreamer_sample`
 - `libgstkvssink.so`
- (Opcional) Você pode adicionar a configuração da variável de ambiente `GST_PLUGIN_PATH` ao script de inicialização do seu shell. Isso garante que `GST_PLUGIN_PATH` seja configurado corretamente durante uma nova sessão de terminal. Dentro AWS Cloud9, o script de inicialização do shell é: `~/.bashrc`

Execute o comando a seguir para anexar o comando ao final do script de inicialização do shell.

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

Digite o seguinte para executar o script de inicialização do shell:

```
source ~/.bashrc
```

Confirme se `GST_PLUGIN_PATH` está definido.

```
echo $GST_PLUGIN_PATH
```

Se você definir a saída corretamente, verá a saída a seguir. Se a saída estiver em branco, a variável de ambiente não será definida corretamente.

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

Execute as amostras para fazer upload de mídia para o Kinesis Video Streams

O aplicativo de amostra não oferece suporte às credenciais do IMDS. No seu terminal, exporte AWS as credenciais do seu usuário ou função do IAM e da região em que seu stream está localizado.

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

Se você estiver usando AWS credenciais temporárias, exporte também seu token de sessão:

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Baixe uma amostra de vídeo.mp4 para fazer o upload para o Kinesis Video Streams.

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

Especificações de vídeo:

- Resolução - 1280 x 720 pixels
- Taxa de quadros - 30 quadros por segundo
- Duração - 14,0 segundos
- Codificação de vídeo - H.264, na faixa 1
- Quadros-chave - A cada 3 segundos, resultando em uma duração de fragmento (também conhecida como tamanho de grupo de imagens (GoP)) de 3 segundos, com o fragmento final tendo 2 segundos de duração.

Execute o comando a seguir com o nome do stream que você criou anteriormente. Se você ainda não criou um stream, consulte [the section called “Crie um stream de vídeo do Kinesis”](#).

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

Sample video from GStreamer

Use o comando a seguir para gerar um vídeo usando o GStreamer.

Diga ao GStreamer onde localizar o plug-in do kvssink GStreamer. No seu diretório de compilação, especifique o caminho para a pasta que contém o `libgstkvssink.so` arquivo.

No seu diretório de compilação, execute o seguinte comando:

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

Esse pipeline do GStreamer gera um stream de vídeo de teste ao vivo com um padrão de teste padrão executado a 10 quadros por segundo com uma resolução de 640x480 pixels. Uma sobreposição é adicionada exibindo a hora e a data atuais do sistema. O vídeo é então codificado no formato H.264 e os quadros-chave são gerados no máximo a cada 10 quadros, resultando em uma duração do fragmento (também conhecida como tamanho de grupo de imagens (GoP)) de 1 segundo. `kvssink` pega o stream de vídeo codificado em H.264, o empacota no formato de contêiner Matroska (MKV) e o carrega no seu stream de vídeo do Kinesis.

Execute o seguinte 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 interromper o pipeline do GStreamer, selecione a janela do terminal e pressione CTRL+C.

Note

Para obter mais informações sobre como usar o plug-in do GStreamer para transmitir vídeo de um fluxo RTSP de uma câmera ou de uma câmera USB, consulte. [Exemplo: Plugin GStreamer do SDK do Kinesis Video Streams Producer - kvssink](#)

Revise os objetos de reconhecimento

Durante o upload, o Kinesis Video Streams enviará objetos de confirmação de volta ao cliente que está realizando o upload. Você deve vê-los impressos na saída do comando. Um exemplo se parece com o seguinte:

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

Se a confirmação `EventType` for `PERSISTED`, significa que o Kinesis Video Streams armazenou e criptografou de forma durável esse pedaço de mídia para recuperação, análise e armazenamento a longo prazo.

Para obter mais informações sobre reconhecimentos, consulte. [the section called "PutMedia"](#)

Consuma dados de mídia

Você pode consumir dados de mídia visualizando-os no console ou criando um aplicativo que lê dados de mídia de um stream usando Hypertext Live Streaming (HLS).

Exibir mídia no console

Em outra guia do navegador, abra AWS Management Console o. [No painel do Kinesis Video Streams, selecione Streams de vídeo.](#)

Selecione o nome do seu stream na lista de streams. Use a barra de pesquisa, se necessário.

Expanda a seção Reprodução de mídia. Se o vídeo ainda estiver sendo carregado, ele será exibido. Se o upload tiver terminado, selecione a seta dupla para a esquerda.

Consuma dados de mídia usando o HLS

Você pode criar um aplicativo cliente que consuma dados de um stream de vídeo do Kinesis usando o HLS. Para obter informações sobre como criar um aplicativo que consome os dados de mídia usando HLS, consulte [the section called “Reprodução de vídeo”](#).

Agente Edge do Amazon Kinesis Video Streams

O Amazon Kinesis Video Streams oferece uma maneira eficiente e econômica de se conectar a câmeras IP nas instalações do cliente. Com o Amazon Kinesis Video Streams Edge Agent, você pode gravar e armazenar localmente vídeos das câmeras e transmitir vídeos para a nuvem em um cronograma definido pelo cliente para armazenamento, reprodução e processamento analítico a longo prazo.

Note

Para acessar o Amazon Kinesis Video Streams Edge Agent, [preencha](#) este breve formulário.

Você pode baixar o Amazon Kinesis Video Streams Edge Agent e implantá-lo em seus dispositivos de computação de borda locais. Você também pode implantá-los facilmente em contêineres Docker executados em instâncias do Amazon EC2. Após a implantação, você pode usar a API Amazon Kinesis Video Streams para atualizar as configurações de gravação de vídeo e upload na nuvem. O recurso funciona com qualquer câmera IP que possa transmitir pelo protocolo RTSP. Não requer nenhuma implantação adicional de firmware nas câmeras.

Oferecemos as seguintes instalações para o Amazon Kinesis Video Streams Edge Agent:

- Como AWS IoT Greengrass V2 componente: você pode instalar o Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass como um componente AWS IoT Greengrass em qualquer dispositivo certificado. Para saber mais AWS IoT Greengrass, consulte o [Guia do AWS IoT Greengrass Version 2 desenvolvedor](#).
- Ativado AWS Snowball Edge: Você pode executar o agente do Amazon Kinesis Video Streams Edge em dispositivos Snowball Edge. Para saber mais, consulte o [Guia do Desenvolvedor do AWS Snowball Edge](#).
- Em uma AWS IoT implantação nativa: você pode instalar o Amazon Kinesis Video Streams Edge Agent nativamente em qualquer instância computacional. O Edge SDK usa [AWS IoT Core](#) para gerenciar o Edge por meio do [the section called “Amazon Kinesis Video Streams”](#).

Para começar a usar o Amazon Kinesis Video Streams Edge Agent, continue com os procedimentos apropriados abaixo.

Tópicos

- [Operações de API do Amazon Kinesis Video Streams Edge Agent](#)
- [Monitoramento do Amazon Kinesis Video Streams Edge Agent](#)
- [Execute o Amazon Kinesis Video Streams Edge Agent em modo não-AWS IoT Greengrass](#)
- [Implante o Amazon Kinesis Video Streams Edge Agent para AWS IoT Greengrass](#)
- [Perguntas frequentes sobre o Amazon Kinesis Video Streams Edge Agent](#)

Operações de API do Amazon Kinesis Video Streams Edge Agent

Use as seguintes operações de API para configurar o Amazon Kinesis Video Streams Edge Agent:

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

Monitoramento do Amazon Kinesis Video Streams Edge Agent

Para monitorar seu Amazon Kinesis Video Streams Edge Agent, consulte. [the section called “Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch”](#)

Execute o Amazon Kinesis Video Streams Edge Agent em modo não-AWS IoT Greengrass

Siga estas etapas para executar o Amazon Kinesis Video Streams Edge AWS IoT Agent com o MQTT como uma implantação independente.

Tópicos

- [Etapa 1: instalar as dependências necessárias no dispositivo](#)
- [Etapa 2: Crie os Amazon Kinesis Video AWS Secrets Manager Streams e os recursos para os URLs RTSP da sua câmera IP](#)
- [Etapa 3: criar uma política de permissões do IAM](#)
- [Etapa 4: criar uma função do IAM](#)
- [Etapa 5: criar o alias da AWS IoT função](#)

- [Etapa 6: criar a AWS IoT política](#)
- [Etapa 7: Crie qualquer AWS IoT coisa e obtenha credenciais para AWS IoT Core](#)
- [Etapa 8: Crie e execute o Amazon Kinesis Video Streams Edge Agent](#)
- [Etapa 9: \(opcional\) instalar o CloudWatch agente no dispositivo](#)
- [Etapa 10: \(Opcional\) Execute o Amazon Kinesis Video Streams Edge Agent como um processo nativo](#)

Etapa 1: instalar as dependências necessárias no dispositivo

Note

Para obter uma lista dos sistemas operacionais compatíveis, consulte [the section called “Quais sistemas operacionais são compatíveis com o Amazon Kinesis Video Streams Edge Agent?”](#).

Instale dependências no dispositivo

1. Para executar o Amazon Kinesis Video Streams Edge Agent, instale as seguintes bibliotecas apropriadas em seu dispositivo:

Ubuntu

Digite:

```
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

Digite:

```
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
```

Instale `log4cplus-2.1.0` a partir da fonte.

```
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` a partir da fonte.

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

Important

Se você ver uma tela informando que alguns serviços precisam ser reiniciados, pressione Enter para selecionar Ok.

Para obter informações adicionais, consulte o Guia do [usuário do Amazon Corretto 11](#).

2. Instale AWS Command Line Interface o. Consulte [Instalando ou atualizando a versão mais recente dos AWS CLI](#) procedimentos no Guia do AWS Command Line Interface usuário.

Etapa 2: Crie os Amazon Kinesis Video AWS Secrets Manager Streams e os recursos para os URLs RTSP da sua câmera IP

Siga esses procedimentos para criar os fluxos e os segredos necessários em AWS Secrets Manager. Execute essa etapa primeiro, porque você precisa dos ARNs dos recursos criados nas políticas.

Crie Amazon Kinesis Video Streams

Crie Amazon Kinesis Video Streams AWS Management Console usando AWS CLI a API, ou.

No AWS Management Console, abra o console do [Amazon Kinesis Video Streams](#). Escolha Streams de vídeo no painel de navegação à esquerda.

Para ter mais informações, consulte [the section called “Crie um stream de vídeo do Kinesis”](#).

Crie segredos em AWS Secrets Manager

No AWS Management Console, abra o [AWS Secrets Manager console](#). Escolha Segredos na navegação à esquerda.

Verifique se a região apropriada está selecionada.

1. Selecione Armazenar um novo segredo.

a. Etapa 1: escolha o tipo secreto

- Selecione Outro tipo de segredo.
- Na seção Pares de chave/valor, adicione um par de valores-chave.

Chave: MediaURI

Note

A chave deve serMediaURI. Isso diferencia maiúsculas de minúsculas. Se você inseri-lo incorretamente, o aplicativo não funcionará.

Valor:*Your MediaURI*.

Example

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

- b. Etapa 2: Configurar o segredo. Dê um nome a esse segredo. Dê o nome que quiser.
 - c. Etapa 3: Configurar a rotação - opcional. Escolha Próximo.
 - d. Etapa 4: Revise. Escolha Armazenar.
2. Se seu segredo não for exibido imediatamente, selecione o botão Atualizar.

Escolha o nome do seu segredo. Anote o ARN secreto.

3. Repita esse processo para cada MediaURI do qual você deseja transmitir.

Note

A AWS rede bloqueia algumas fontes públicas de RTSP. Você não pode acessá-los de dentro da instância do Amazon EC2 ou se estiver executando sem gerenciamento enquanto estiver conectado à VPN.

Important

O URL RTSP da sua câmera deve transmitir vídeo no formato h.264. A duração do fragmento não deve exceder o limite mencionado em [the section called “Limites do SDK do produtor”](#).

O Amazon Kinesis Video Streams Edge Agent só oferece suporte a vídeo.

Execute `gst-discoverer-1.0 Your RtspUrl` para garantir que sua câmera esteja acessível a partir do seu dispositivo.

Salve os ARNs de todos os streams e segredos que você criou. Você precisa deles para a próxima etapa.

Etapa 3: criar uma política de permissões do IAM

Siga esses procedimentos para criar uma política do IAM. Essa política de permissões permite o controle de acesso seletivo (um subconjunto de operações suportadas) para um AWS recurso. Nesse caso, os AWS recursos são os streams de vídeo para os quais você deseja que o Amazon Kinesis Video Streams Edge Agent transmita. Os recursos também incluem os AWS Secrets Manager segredos que o Amazon Kinesis Video Streams Edge Agent pode recuperar. Para obter mais informações, consulte as [políticas do IAM](#).

Crie uma política usando o editor de políticas JSON

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Políticas.

Se essa for a primeira vez que você escolhe Políticas, a página Bem-vindo às políticas gerenciadas será exibida. Escolha Começar.

3. Na parte superior da página, escolha Criar política.
4. Na seção Editor de políticas, escolha a opção JSON.
5. Insira o seguinte documento de política 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

arn:aws:kinesisvideo:*:*:stream/streamName2/* substitua arn:aws:kinesisvideo:*:*:stream/streamName1/* e pelos ARNs dos fluxos de vídeo e arn:aws:secretsmanager:*:*:secret:* substitua pelos ARNs que contêm os segredos do MediaURI que você criou em [the section called “2. Crie recursos para os URLs RTSP da sua câmera IP”](#) Use os ARNs para os segredos que você deseja que o Amazon Kinesis Video Streams Edge Agent acesse.

6. Escolha Próximo.

Note

É possível alternar entre as opções de editor Visual e JSON a qualquer momento. Porém, se você fizer alterações ou escolher Próximo no editor Visual, o IAM poderá reestruturar a política a fim de otimizá-la para o editor visual. Para obter mais informações, consulte [Reestruturação de políticas](#) no Guia do usuário do IAM.

7. Na página Revisar e criar, insira um nome de política e uma descrição opcional para a política que você está criando. Revise Permissões definidas nessa política para ver as permissões que são concedidas pela política.
8. Escolha Criar política para salvar sua nova política.

Etapa 4: criar uma função do IAM

A função que você cria nesta etapa pode ser AWS IoT assumida para obter credenciais temporárias do AWS Security Token Service (AWS STS). Isso é feito ao realizar solicitações de autorização de credenciais do Amazon Kinesis Video Streams Edge Agent.

Crie a função de serviço para o Amazon Kinesis Video Streams (console do IAM)

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Funções e, em seguida, Criar função.
3. Escolha o tipo de função da política de confiança personalizada e cole a seguinte política:

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

4. Selecione a caixa ao lado da política do IAM que você criou em [the section called “3. Criar uma política do IAM”](#).
5. Escolha Próximo.
6. Insira um nome da função ou um sufixo do nome da função para ajudá-lo a identificar a finalidade dessa função.

Example

Exemplo: KvsEdgeAgentRole

7. (Opcional) Em Descrição da função, insira uma descrição para a nova função.
8. (Opcional) Adicione metadados à função anexando tags como pares de chave/valor.

Para obter mais informações sobre o uso de tags no IAM, consulte Como [marcar recursos do IAM](#) no Guia do usuário do IAM.

9. Revise a função e escolha Criar perfil.

Etapa 5: criar o alias da AWS IoT função

Siga esses procedimentos para criar um alias de AWS IoT função para a função do IAM que você criou em [the section called “4. Criar um perfil do IAM”](#). Um alias de função é um modelo de dados alternativo que aponta para a função do IAM. Uma solicitação de provedor de AWS IoT credenciais deve incluir um alias de função para indicar qual função do IAM assumir para obter credenciais temporárias do AWS Security Token Service (STS). Para obter mais informações, consulte [Como usar um certificado para obter um token de segurança](#).

Crie o alias da AWS IoT função

1. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. Verifique se a região apropriada está selecionada.
3. No painel de navegação à esquerda, selecione Segurança e, em seguida, escolha Aliases de função.
4. Escolha Criar alias de função.
5. Insira um nome para seu alias de função.

Example

Exemplo: KvsEdgeAgentRoleAlias

6. No menu suspenso Role, selecione a função do IAM que você criou. [the section called “4. Criar um perfil do IAM”](#)
7. Escolha Criar. Na próxima página, você verá uma observação de que seu alias de função foi criado com sucesso.
8. Pesquise e selecione o alias de função recém-criado. Anote o alias da função ARN. Você precisa disso para a AWS IoT política na próxima etapa.

Etapa 6: criar a AWS IoT política

Siga esses procedimentos para criar uma AWS IoT política que será anexada ao certificado do dispositivo. Isso dá permissões aos AWS IoT recursos e permite a suposição do alias da função usando o certificado.

Com AWS IoT Core as políticas, você pode controlar o acesso ao plano AWS IoT Core de dados. O plano de AWS IoT Core dados consiste em operações que você pode usar para fazer o seguinte:

- Conecte-se ao agente de AWS IoT Core mensagens
- Enviar e receber mensagens MQTT
- Obtenha ou atualize o Device Shadow de uma coisa

Para obter mais informações, consulte [Políticas AWS IoT Core](#).

Use o editor de AWS IoT políticas para criar uma AWS IoT política

1. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. No painel de navegação à esquerda, selecione Segurança e, em seguida, escolha Políticas.
3. Escolha Criar política.
4. Insira um nome para sua política.

Example

Um exemplo de nome de política é KvsEdgeAccessIoTPolicy.


5. (Opcional) Adicione metadados à política associando tags como pares de chave-valor.

Para obter mais informações sobre o uso de tags no IAM, consulte Como [marcar seus AWS IoT recursos](#) no Guia do AWS IoT Core desenvolvedor.

6. Selecione a guia JSON.
7. Cole o seguinte documento da política 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 Substitua pelo ARN do alias de função que você criou em. [the section called “5. Crie o alias da AWS IoT função”](#)

8. Escolha Criar para salvar seu trabalho.

Etapa 7: Crie qualquer AWS IoT coisa e obtenha credenciais para AWS IoT Core

Neste ponto, você criou:

- Uma política de permissões do IAM. Consulte [the section called “3. Criar uma política do IAM”](#).
- Uma função do IAM, com a política de permissões anexada. Consulte [the section called “4. Criar um perfil do IAM”](#).
- Um alias de AWS IoT função para a função do IAM. Consulte [the section called “5. Crie o alias da AWS IoT função”](#).
- Uma AWS IoT política, atualmente não vinculada a nenhum AWS recurso. Consulte [the section called “6. Crie a AWS IoT política”](#).

Para criar e registrar qualquer AWS IoT coisa e obter credenciais de AWS IoT Core acesso

1. Registre o dispositivo como uma AWS IoT coisa e gere o certificado X.509 para o dispositivo.

- a. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
- b. Selecione a região apropriada.
- c. No painel de navegação à esquerda, selecione Todos os dispositivos e, em seguida, escolha Coisas.
- d. Escolha Criar coisas.
- e. Selecione Criar uma única coisa e, em seguida, escolha Avançar.

1. Etapa 1. Especifique as propriedades da coisa

Digite um nome para sua coisa e escolha Avançar.

2. Etapa 2. Configurar o certificado do dispositivo

Selecione Gerar automaticamente um novo certificado (recomendado) e escolha Avançar.

3. Etapa 3. Anexar políticas ao certificado

Pesquise a política de permissões que você criou em [the section called “6. Crie a AWS IoT política”](#).

Marque a caixa de seleção ao lado da sua política e escolha Criar coisa.

f. Na janela exibida, baixe os seguintes arquivos:

- Certificado do dispositivo. Este é o certificado X.509.
- Arquivo de chave pública
- Arquivo de chave privada
- Endpoint de serviços confiáveis da Amazon (chave RSA de 2048 bits: Amazon Root CA 1)

Anote a localização de cada um desses arquivos para uma etapa posterior.

g. Selecione Done (Concluído). Na próxima página, você verá uma nota de que sua coisa foi criada com sucesso.

- h. Transfira os arquivos baixados acima para sua AWS IoT coisa, se ainda não estiver lá.
2. Obtenha o endpoint do provedor de credenciais para sua AWS conta.

AWS CLI

Execute o seguinte comando:

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

AWS Management Console

Em [AWS CloudShell](#), execute o seguinte comando:

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

Anote essas informações para uma etapa posterior.

3. Obtenha o endpoint de dados do dispositivo para sua AWS conta.

AWS CLI

Execute o seguinte comando:

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

AWS Management Console

Faça o seguinte:

1. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>.
2. No painel de navegação à esquerda, selecione Configurações.
3. Localize o endpoint de dados do dispositivo.

Anote essas informações para uma etapa posterior.

4. (Opcional) Verifique se seus certificados foram gerados corretamente.

Execute o comando a seguir para validar se seus itens foram gerados corretamente.


```
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 obter mais informações, consulte [Como usar um certificado para obter um token de segurança](#).

Etapa 8: Crie e execute o Amazon Kinesis Video Streams Edge Agent

Crie e execute o Amazon Kinesis Video Streams Edge Agent

1. Faça o download tar do arquivo usando o link que foi fornecido a você.

Se você preencheu o formulário de interesse do Amazon Kinesis Video Streams Edge Agent, verifique seu e-mail para obter o link para download. Se você não preencheu o formulário, preencha-o [aqui](#).

2. Verifique a soma de verificação.
3. Extraia os binários e o jar em seu dispositivo.

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

Após a extração, sua estrutura de pastas terá a seguinte aparência:

```
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

O nome da pasta de lançamento deve ser configurado de forma a refletir o número da versão binária mais recente. Por exemplo, uma versão 1.0.0 terá o nome da pasta definido como 1.0.0.

4. Crie o jar de dependências.

Note

O jar incluído no `kvs-edge-agent.tar.gz` não tem as dependências. Use as etapas a seguir para criar essas bibliotecas.

Navegue até a `kvs-edge-agent` pasta que contém `pom.xml`.

Digite `mvn clean package`.

Isso gera um arquivo jar contendo as dependências que o Amazon Kinesis Video Streams Edge Agent exige em `kvs-edge-agent/target/libs.jar`

5. Coloque o `libs.jar` na pasta que contém os artefatos do componente.

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

6. Defina variáveis de ambiente usando os valores das etapas anteriores. A tabela a seguir fornece descrições das variáveis.

Nome da variável de ambiente	Obrigatório	Descrição
AWS_REGION	Sim	A região que é usada. Exemplo: us-west-2
AWS_IOT_CA_CERT	Sim	Caminho do arquivo para o certificado CA usado para estabelecer confiança com o serviço de back-end por meio do TLS. Exemplo: <i>/file/path/to/AmazonRootCA1.pem</i>
AWS_IOT_CORE_CERT	Sim	Caminho do arquivo para o certificado X.509. Exemplo: <i>/file/path/to/certificateID-certificate.pem.crt</i>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	Sim	O endpoint do provedor de endpoint de AWS IoT Core credenciais para sua conta. AWS Exemplo: <i>credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com</i>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_IOT_CORE_DATA_ATS_ENDPOINT	Sim	<p>O endpoint do plano de AWS IoT Core dados da sua AWS conta.</p> <p>Exemplo: <i>data-account-specific-prefix.iot.aws-region.amazonaws.com</i></p>
AWS_IOT_CORE_PRIVATE_KEY	Sim	<p>Caminho do arquivo para a chave privada usada no par de chaves pública/privada. Para obter mais informações, consulte Gerenciamento de chaves em AWS IoT.</p> <p>Exemplo: <i>/file/path/to/certificateID-private.pem.key</i></p>
AWS_IOT_CORE_ROLE_ALIAS	Sim	<p>O nome do alias da função que aponta para a função do AWS IAM a ser usada ao se conectar AWS IoT Core.</p> <p>Exemplo: <i>kvs-edge-role-alias</i></p>
AWS_IOT_CORE_THING_NAME	Sim	<p>O nome da AWS IoT coisa em que o aplicativo está sendo executado.</p> <p>Exemplo: <i>my-edge-device-thing</i></p>

Nome da variável de ambiente	Obrigatório	Descrição
GST_PLUGIN_PATH	Sim	<p>Caminho do arquivo apontando para a pasta que contém as gstreamer bibliotecas IngestorPipelineJNI dependentes da plataforma. Permite que o GStreamer carregue esses plugins. Para obter mais informações, consulte Baixar, criar e configurar o elemento GStreamer.</p> <p>Exemplo: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>

Nome da variável de ambiente	Obrigatório	Descrição
LD_LIBRARY_PATH	Sim	<p>Caminho do arquivo apontando para o diretório que contém <code>cproducer</code> as bibliotecas <code>KinesisVideoProducer</code> dependentes da plataforma.</p> <p>Exemplo: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Não	<p>Determina se o Amazon Kinesis Video Streams Edge Agent publicará métricas de saúde do trabalho em Amazon CloudWatch</p> <p>Valores aceitos: TRUE/FALSE (sem distinção entre maiúsculas e minúsculas). O padrão é FALSE se não for fornecido.</p> <p>Exemplo: FALSO</p>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_KVS_EDGE_LOG_LEVEL	Não	<p>O nível de registro das saídas do Amazon Kinesis Video Streams Edge Agent.</p> <p>Valores aceitos:</p> <ul style="list-style-type: none">• DESL.• ALL• FATAL• ERRO• WARN• INFO, padrão, se não for fornecido• DEBUG• TRACE <p>Exemplo: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	Não	<p>Quando o arquivo de log atingir esse tamanho, ocorrerá uma substituição.</p> <ul style="list-style-type: none">• Mínimo: 0• Máximo: 10000• Padrão: 20, se não for fornecido• Unidades: Megabytes (MB) <p>Exemplo: 5</p>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	Não	<p>O caminho do arquivo apontando para o diretório em que os registros do Amazon Kinesis Video Streams Edge Agent são gerados. O padrão é <code>./log</code> se não for fornecido.</p> <p>Exemplo: <code>./file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Não	<p>O número de registros acumulados a serem mantidos antes de serem excluídos.</p> <ul style="list-style-type: none">• Mínimo: 1• Máximo: 100• Padrão: 10, se não for fornecido <p>Exemplo: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Não	<p>O caminho do arquivo apontando para o diretório em que a mídia gravada será gravada. O padrão é o diretório atual, se não for fornecido.</p> <p>Exemplo: <code>./file/path/</code></p>

Nome da variável de ambiente	Obrigatório	Descrição
GST_DEBUG	Não	Especifica o nível de registros do GStreamer a serem gerados. Para obter mais informações, consulte a documentação do GStreamer . Exemplo: 0
GST_DEBUG_FILE	Não	Especifica o arquivo de saída dos registros de depuração do GStreamer. Se não for definido, os registros de depuração geram a saída para o erro padrão. Para obter mais informações, consulte a documentação do GStreamer . Exemplo: <i>/tmp/gstreamer-logging .log</i>

7. Limpe o cache do GStreamer. Digite:

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

Para obter mais informações, consulte a documentação do [registro do GStreamer](#).

8. Prepare e execute o comando java. O Amazon Kinesis Video Streams Edge Agent aceita os seguintes argumentos:

Nome da propriedade Java	Obrigatório	Descrição
<code>java.library.path</code>	Não	Caminho do arquivo apontando para a pasta que contém as bibliotecas

Nome da propriedade Java	Obrigatório	Descrição
		IngestorPipelineJNI dependentes gstkvsink e as bibliotecas dependentes. Se não for fornecido, o Amazon Kinesis Video Streams Edge Agent os procurará no diretório atual.

 Important

O Amazon Kinesis Video Streams Edge Agent não funcionará corretamente se não conseguir localizar esses arquivos.

Exemplo: */file/path/*

Para defini-los, adicione `-Djava-property-name=value` ao comando java usado para executar o jar.

Por exemplo: .

```
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

Execute o comando java acima no mesmo diretório de `/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion`.

9. Envie configurações para o aplicativo usando o AWS CLI

- a. Crie um novo arquivo, `example-edge-configuration.json`.

Cole o seguinte código no arquivo. Esse é um exemplo de configuração que grava diariamente das 9:00 às 16:59:59 (de acordo com a hora do sistema em seu dispositivo). AWS IoT Ele também carrega a mídia gravada diariamente das 19:00:00 às 21:59:59:59.

Para ter mais informações, 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 o arquivo para o Amazon Kinesis Video Streams Edge Agent, digite o seguinte no: AWS CLI

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

10. Repita a etapa anterior para cada stream do Amazon Kinesis Video Streams Edge Agent.

Etapa 9: (opcional) instalar o CloudWatch agente no dispositivo

Note

Esteja ciente das [CloudWatchcotas](#).

Siga esses procedimentos para instalar e configurar o CloudWatch agente para carregar automaticamente os registros gerados pelo Amazon Kinesis Video Streams Edge Agent para. CloudWatch

Para obter [os procedimentos](#) para instalar o CloudWatch agente em seu dispositivo, consulte o Guia CloudWatch do usuário da Amazon.

Quando a configuração for solicitada, selecione uma das configurações a seguir.

Important

As `file_path` configurações a seguir pressupõem que o local de saída de registro padrão seja usado.

O caminho do arquivo usado pressupõe que você esteja executando o Amazon Kinesis Video Streams Edge Agent a partir do local: *download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*version*

- Para configurar o CloudWatch agente para fazer upload de registros e publicar métricas de RAM e CPU do dispositivo, cole o seguinte no arquivo de configuração.

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



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

Etapa 10: (Opcional) Execute o Amazon Kinesis Video Streams Edge Agent como um processo nativo

Configure o Amazon Kinesis Video Streams Edge Agent como um serviço systemd.

systemd é um gerenciador de sistemas e serviços em dispositivos Linux. systemd é a forma recomendada de gerenciar o processo, pois ele reiniciará o Amazon Kinesis Video Streams Edge Agent caso o aplicativo encontre um erro ou o dispositivo que executa o aplicativo perca energia.

Faça o seguinte:

Execute o Amazon Kinesis Video Streams Edge Agent como um processo nativo

1. Crie um novo arquivo `/etc/systemd/system` e nomeie-o `aws.kinesisvideo.edge-runtime-agent.service`.

Cole o seguinte:

```

[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 obter mais informações sobre os parâmetros aceitos pelo arquivo `systemd` de configuração do serviço, consulte a [documentação](#).

Note

Adicione as variáveis de ambiente necessárias no `...` local, conforme especificado em [the section called “8. Crie e execute o Amazon Kinesis Video Streams Edge Agent”](#).

2. Recarregue os arquivos do serviço para incluir o novo serviço.

Digite `sudo systemctl daemon-reload`.

3. Inicie o serviço .

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

4. Verifique o status do serviço Amazon Kinesis Video Streams Edge Agent para verificar se ele está em execução.

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

Veja a seguir um exemplo da saída que você 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. Inspecione os registros em busca de erros.

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

6. Digite `systemctl --help` para obter a lista completa de opções para gerenciar o processo usando `systemctl`.

A seguir estão alguns comandos comuns para gerenciar o Amazon Kinesis Video Streams Edge Agent:

- Para reiniciar, digite `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`.
- Para parar, digite `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Para iniciar automaticamente a cada reinicialização do dispositivo, digite `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Implante o Amazon Kinesis Video Streams Edge Agent para AWS IoT Greengrass

Siga estas etapas para implantar o Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent para gravar e fazer upload de mídia de câmeras IP.

Tópicos

- [Etapa 1: criar uma instância do Ubuntu Amazon EC2](#)
- [Etapa 2: configurar o dispositivo AWS IoT Greengrass V2 principal no dispositivo](#)
- [Etapa 3: Crie os Amazon Kinesis Video AWS Secrets Manager Streams e os recursos para os URLs RTSP da sua câmera IP](#)
- [Etapa 4: Adicionar permissões à função do serviço de troca de tokens \(TES\)](#)
- [Etapa 5: instalar o componente AWS IoT Greengrass Secret Manager no dispositivo](#)
- [Etapa 6: Implantar o componente Amazon Kinesis Video Streams AWS IoT Greengrass Edge Agent no dispositivo](#)
- [Etapa 7: \(opcional\) instalar o componente do gerenciador de AWS IoT Greengrass registros no dispositivo](#)

Etapa 1: criar uma instância do Ubuntu Amazon EC2

Faça o seguinte para criar uma instância do Ubuntu Amazon EC2.

Crie uma instância do Ubuntu Amazon EC2

1. [Faça login no AWS Management Console e abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)

Verifique se a região apropriada está selecionada.

2. Escolha Executar instância.

Preencha os seguintes campos:

- Nome — digite um nome para a instância.
- Imagens do aplicativo e do sistema operacional (Amazon Machine Image) — Selecione Ubuntu.
- Tipo de instância — Selecione t2.large.

- Login do par de chaves — Crie seu próprio par de chaves.
 - Configurações de rede — Mantenha o padrão.
 - Configurar armazenamento — Aumente o volume para 256 GiB.
 - Configurações avançadas — Mantenha o padrão.
3. Execute a instância e faça o SSH nela.

Faça o seguinte:

1. Selecione Instâncias no painel de navegação à esquerda e, em seguida, selecione o ID da instância.
 2. Escolha Connect no canto superior direito.
 3. Escolha o cliente SSH e siga as instruções na tela.
 4. Abra um terminal e navegue até o .pem arquivo baixado (provavelmente em~/Downloads).
 5. Na primeira vez que você seguir esses procedimentos, você receberá a mensagem “A autenticidade do host (...) não pode ser estabelecida”. Digite sim.
4. Instale bibliotecas do sistema para criar o Amazon Kinesis Video Streams Edge Agent na instância.

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'

sudo apt-get update

sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-good gstreamer1.0-tools \
unzip java-11-amazon-corretto-jdk maven
```

Important

Se você ver uma tela informando que alguns serviços precisam ser reiniciados, pressione Enter para selecionar Ok.

Para obter mais informações, consulte o Guia do [usuário do Amazon Corretto 11](#).

Etapa 2: configurar o dispositivo AWS IoT Greengrass V2 principal no dispositivo

Siga esses procedimentos para instalar o software AWS IoT Greengrass núcleo principal na instância do Amazon EC2.

Configurar o dispositivo AWS IoT Greengrass principal

1. Faça login no AWS Management Console <https://console.aws.amazon.com/iot/>.

Verifique se a região apropriada está selecionada.

2. No painel de navegação à esquerda, selecione dispositivos Greengrass, dispositivos principais.
3. Escolha Configurar um dispositivo principal.
4. Conclua as etapas na tela.

- Etapa 1: Registre um dispositivo principal do Greengrass. Digite um nome para o dispositivo.
- Etapa 2: adicione a um grupo de coisas para aplicar uma implantação contínua. Selecione Nenhum grupo.
- Etapa 3: Instale o software Greengrass Core. Selecione Linux.
 - Etapa 3.1: instalar o Java no dispositivo

O Java é instalado como parte do [the section called "1. Criar instância do Ubuntu"](#). Retorne a essa etapa se você ainda não tiver o Java instalado.

- Etapa 3.2: Copiar AWS as credenciais para o dispositivo

Abra a bash/zsh opção e cole os comandos de exportação na instância do Amazon EC2.


- Etapa 3.3: Executar o instalador

1. Copie e execute os comandos Baixar o instalador e Executar o instalador na instância do Ubuntu Amazon EC2.

Note

O comando Executar o instalador será atualizado automaticamente com base no nome que você escolheu na etapa anterior.

2. Anote a função do serviço de troca de tokens (TES) criada. Você precisará disso mais tarde.

 Note

Por padrão, a função criada é chamada de `TokenExchangeRoleGreengrassV2`.

Etapa 3: Crie os Amazon Kinesis Video AWS Secrets Manager Streams e os recursos para os URLs RTSP da sua câmera IP

Siga esses procedimentos para criar os fluxos e os segredos necessários em AWS Secrets Manager. Execute essa etapa primeiro, porque você precisa dos ARNs dos recursos criados nas políticas.

Crie Amazon Kinesis Video Streams

Crie Amazon Kinesis Video Streams AWS Management Console usando AWS CLI a API, ou.

No AWS Management Console, abra o console do [Amazon Kinesis Video Streams](#). Escolha Streams de vídeo no painel de navegação à esquerda.

Para ter mais informações, consulte [the section called “Crie um stream de vídeo do Kinesis”](#).


Crie segredos em AWS Secrets Manager

No AWS Management Console, abra o [AWS Secrets Manager console](#). Escolha Segredos na navegação à esquerda.

Verifique se a região apropriada está selecionada.

1. Selecione Armazenar um novo segredo.
 - a. Etapa 1: escolha o tipo secreto
 - Selecione Outro tipo de segredo.
 - Na seção Pares de chave/valor, adicione um par de valores-chave.

Chave: `MediaURI`

 Note

A chave deve ser `MediaURI`. Isso diferencia maiúsculas de minúsculas. Se você inseri-lo incorretamente, o aplicativo não funcionará.

Valor:*Your MediaURI*.

Example

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

- b. Etapa 2: Configurar o segredo. Dê um nome a esse segredo. Dê o nome que quiser.
 - c. Etapa 3: Configurar a rotação - opcional. Escolha Próximo.
 - d. Etapa 4: Revise. Escolha Armazenar.
2. Se seu segredo não for exibido imediatamente, selecione o botão Atualizar.

Escolha o nome do seu segredo. Anote o ARN secreto.

3. Repita esse processo para cada MediaURI do qual você deseja transmitir.

Note

A AWS rede bloqueia algumas fontes públicas de RTSP. Você não pode acessá-los de dentro da instância do Amazon EC2 ou se estiver executando sem gerenciamento enquanto estiver conectado à VPN.

Important

O URL RTSP da sua câmera deve transmitir vídeo no formato h.264. A duração do fragmento não deve exceder o limite mencionado em [the section called “Limites do SDK do produtor”](#).

O Amazon Kinesis Video Streams Edge Agent só oferece suporte a vídeo.

Execute `gst-discoverer-1.0 Your RtspUrl` para garantir que sua câmera esteja acessível a partir do seu dispositivo.

Salve os ARNs de todos os streams e segredos que você criou. Você precisa deles para a próxima etapa.

Etapa 4: Adicionar permissões à função do serviço de troca de tokens (TES)

Conceda a função de serviço de troca de tokens (TES) ao dispositivo que assume permissões para examinar os segredos. Isso é necessário para que o AWS Secrets Manager AWS IoT Greengrass componente funcione corretamente.

Adicionar permissões à função TES

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Escolha Funções no painel de navegação à esquerda e pesquise a função TES que você criou anteriormente no processo.
3. No menu suspenso Adicionar permissões, selecione Anexar políticas.
4. Escolha Criar política.
5. Role para baixo e selecione Editar.
6. No editor de políticas, escolha JSON e edite a política.

Substitua a política pela seguinte:

Note

arn:aws:kinesisvideo:*:*:stream/streamName2/*Substitua
arn:aws:kinesisvideo:*:*:stream/streamName1/* e pelos ARNs dos fluxos
que você criou em uma etapa anterior.

```
{
  "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. Na página Add tags (Adicionar tags), escolha Next: Review(Próximo: Revisar).
8. Dê um nome à sua política e escolha Criar política.

Um exemplo de nome de política é KvsEdgeAccessPolicy.

9. Feche a guia e retorne à guia em que você estava anexando uma política à função TES.

Escolha o botão Atualizar e, em seguida, pesquise a política recém-criada.

Marque a caixa de seleção e escolha Anexar políticas.

Na próxima tela, você verá uma nota que diz que a política foi anexada com sucesso à função.

10. Crie e anexe outra política, desta vez para seus segredos.

Substitua a política pela seguinte:

Note

arn:aws:secretsmanager:*:*:secret:* Substitua pelos ARNs que contêm os segredos do MediaURI que você criou em [the section called “3. Crie recursos para os URLs RTSP da sua câmera IP”](#)

```

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

```



```

        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [
            "arn:aws:secretsmanager:*:*:secret:*",
            "arn:aws:secretsmanager:*:*:secret:*"
        ]
    }
]
}

```

11. Crie e anexe outra política, desta vez para Amazon CloudWatch métricas. Substitua a política pela seguinte:

```

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

```

Etapa 5: instalar o componente AWS IoT Greengrass Secret Manager no dispositivo

O Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass exige que o componente Secret Manager seja instalado primeiro no dispositivo.

Instale o componente Secret Manager

1. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>. Verifique se a região apropriada está selecionada.
2. No painel de navegação à esquerda, escolha Greengrass devices, Deployments.

Escolha a implantação com o mesmo destino do que criamos em [the section called “2. Configurar o dispositivo AWS IoT Greengrass principal”](#).

3. No menu suspenso Ações no canto superior direito, escolha Revisar.

No pop-up exibido, escolha Revisar implantação.

4. Complete as seguintes seções:

- Etapa 1: especifique o alvo. Escolha Próximo.
- Etapa 2: Selecione os componentes.
 - Verifique se o componente AWS.Greengrass.cli está selecionado. Não desinstale esse componente.
 - Ative a opção Mostrar somente componentes selecionados e pesquise por aws.greengrass.SecretManager.
 - Marque a caixa ao lado de aws.greengrass.SecretManager, em seguida, escolha Avançar.
- Etapa 3: Configurar componentes. Configure o componente AWS IoT Greengrass Secret Manager para baixar os segredos de dentro do AWS IoT Greengrass ambiente.

Selecione o aws.greengrass.SecretManager componente e, em seguida, escolha Configurar componente.

Na tela exibida, atualize os AWS Secrets Manager ARNs na caixa Configuração para mesclar.

Note

arn:aws:secretsmanager:*:*:secret:* Substitua pelos ARNs dos segredos que você criou em [the section called “3. Crie recursos para os URLs RTSP da sua câmera IP”](#).

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

```
]
}
```

Note

`cloudSecrets` é uma lista de objetos com a chave `arn`. Para obter mais informações, consulte a seção [Configuração do gerenciador secreto](#) no Guia do AWS IoT Greengrass Version 2 desenvolvedor.

Ao terminar, selecione Confirmar e, em seguida, escolha Avançar.

- Etapa 4: Defina as configurações avançadas. Escolha Próximo.
 - Etapa 5: Revise. Selecione Implantar.
5. Confirme se o AWS Secrets Manager componente e as permissões foram instalados corretamente.

Na instância do Ubuntu Amazon EC2, digite `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` para verificar se o componente recebeu a configuração atualizada.

6. Inspecione os registros AWS IoT Greengrass principais.

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

Verifique se há erros de implantação.

Se houver um erro, revise a implantação para remover o `aws.greengrass.SecretManager` componente.

Digite `sudo service greengrass restart` para reiniciar o serviço AWS IoT Greengrass principal.

Se o erro de implantação estiver relacionado à falta de permissões, revise a [the section called “4. Adicionar permissões à função TES”](#) seção para garantir que a função TES tenha as permissões adequadas. Em seguida, repita esta seção.

Atualize os segredos no componente AWS IoT Greengrass Secret Manager

Important

O componente AWS IoT Greengrass Secret Manager busca e armazena segredos em cache somente quando a implantação é atualizada.

Para atualizar os segredos no componente AWS IoT Greengrass Secret Manager, siga as etapas 1 a 6 anteriores, com a seguinte alteração.

Etapa 3: Configurar componentes. Configure o componente AWS IoT Greengrass Secret Manager para baixar os segredos de dentro do AWS IoT Greengrass ambiente.

Selecione o `aws.greengrass.SecretManagercomponente` e, em seguida, escolha Configurar componente.

Na tela exibida, cole `[""]` na caixa Redefinir caminhos e atualize os AWS Secrets Manager ARNs na caixa Configuração para mesclar.

Para obter mais informações, consulte [Redefinir atualizações](#).

Etapa 6: Implantar o componente Amazon Kinesis Video Streams AWS IoT Greengrass Edge Agent no dispositivo

Implante o componente Amazon Kinesis Video Streams AWS IoT Greengrass Edge Agent no dispositivo

1. Faça o download `tar` do arquivo usando o link fornecido.

Se você preencheu o formulário de interesse do Amazon Kinesis Video Streams Edge Agent, verifique seu e-mail para obter o link para download. Se você não preencheu o formulário, preencha-o [aqui](#).

2. Verifique a soma de verificação.
3. Extraia os binários e o jar em seu dispositivo.

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

Após a extração, sua estrutura de pastas terá a seguinte aparência:

```
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

O nome da pasta de lançamento deve ser configurado de forma a refletir o número da versão binária mais recente. Por exemplo, uma versão 1.0.0 terá o nome da pasta definido como 1.0.0.

4. Crie o jar de dependências.

Note

O jar incluído no kvs-edge-agent .tar.gz não tem as dependências. Use as etapas a seguir para criar essas bibliotecas.

Navegue até a `kvs-edge-agent` pasta que contém `pom.xml`.

Digite `mvn clean package`.

Isso gerará um arquivo `jar` contendo as dependências que o Amazon Kinesis Video Streams Edge Agent exige em `kvs-edge-agent/target/libs.jar`

5. Coloque o `libs.jar` na pasta que contém os artefatos do componente.

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

6. Opcional. Configure as propriedades. O Amazon Kinesis Video Streams Edge Agent aceita as seguintes variáveis de ambiente AWS IoT Greengrass no modo:

Nome da variável de ambiente	Obrigatório	Descrição
<code>AWS_REGION</code>	Sim	A região que é usada. Exemplo: <code>us-west-2</code> AWS IoT Greengrass O software principal define automaticamente esse valor para você. Para obter mais informações, consulte o tópico de referência da variável de ambiente do componente no Guia do AWS IoT Greengrass Version 2 desenvolvedor.
<code>GST_PLUGIN_PATH</code>	Sim	Caminho do arquivo apontando para a pasta que contém <code>gstkvssink</code> as bibliotecas <code>IngestorPipelineJNI</code> dependentes da plataforma. Isso

Nome da variável de ambiente	Obrigatório	Descrição
		<p>permite que o GStreamer carregue esses plug-ins. Para obter mais informações, consulte Baixar, criar e configurar o elemento GStreamer.</p> <p>Exemplo: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>
LD_LIBRARY_PATH	Sim	<p>Caminho do arquivo apontando para o diretório que contém cproducer as bibliotecas KinesisVideoProducer dependentes da plataforma.</p> <p>Exemplo: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Não	<p>Determina se o Amazon Kinesis Video Streams Edge Agent publicará métricas de saúde do trabalho em Amazon CloudWatch</p> <p>Valores aceitos: TRUE/FALSE (sem distinção entre maiúsculas e minúsculas). O padrão é FALSE se não for fornecido.</p> <p>Exemplo: FALSO</p>
AWS_KVS_EDGE_LOG_LEVEL	Não	<p>O nível de registro das saídas do Amazon Kinesis Video Streams Edge Agent.</p> <p>Valores aceitos:</p> <ul style="list-style-type: none">• DESL.• ALL• FATAL• ERRO• WARN• INFO, padrão, se não for fornecido• DEBUG• TRACE <p>Exemplo: INFO</p>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_KVS_EDGE_LOG_M AX_FILE_SIZE	Não	<p>Quando o arquivo de log atingir esse tamanho, ocorrerá uma substituição.</p> <ul style="list-style-type: none">• Mínimo: 1• Máximo: 100• Padrão: 20, se não for fornecido• Unidades: Megabytes (MB) <p>Exemplo: 5</p>
AWS_KVS_EDGE_LOG_0 OUTPUT_DIRECTORY	Não	<p>O caminho do arquivo apontando para o diretório em que os registros do Amazon Kinesis Video Streams Edge Agent são gerados. O padrão é <code>./log</code> se não for fornecido.</p> <p>Exemplo: <code>/file/path/</code></p>

Nome da variável de ambiente	Obrigatório	Descrição
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Não	<p>O número de registros acumulados a serem mantidos antes de serem excluídos.</p> <ul style="list-style-type: none">• Mínimo: 1• Máximo: 100• Padrão: 10, se não for fornecido <p>Exemplo: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Não	<p>O caminho do arquivo apontando para o diretório em que a mídia gravada será gravada. O padrão é o diretório atual, se não for fornecido.</p> <p>Exemplo: <i>/file/path/</i></p>
GREENGRASS_ROOT_DIRECTORY	Não	<p>O caminho do arquivo para o diretório AWS IoT Greengrass raiz.</p> <p>O padrão é <i>/greengrass/v2/</i> se não for fornecido.</p> <p>Exemplo: <i>/file/path/</i></p>

Nome da variável de ambiente	Obrigatório	Descrição
GST_DEBUG	Não	Especifica o nível de registros do GStreamer a serem gerados. Para obter mais informações, consulte a documentação do GStreamer . Exemplo: 0
GST_DEBUG_FILE	Não	Especifica o arquivo de saída dos registros de depuração do GStreamer. Se não for definido, os registros de depuração geram a saída para o erro padrão. Para obter mais informações, consulte a documentação do GStreamer . Exemplo: <i>/tmp/gstreamer-logging</i> .log

Abra `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` e modifique o script de execução para adicionar qualquer uma das variáveis de ambiente anteriores.

 Important

Certifique-se de que o script de execução modificado não contenha nenhum caractere de tabulação. O software AWS IoT Greengrass principal não conseguirá ler a receita.

7. Implante o componente Amazon Kinesis Video Streams AWS IoT Greengrass Edge Agent.

Digite:

```
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 obter informações adicionais, consulte as seções a seguir no Guia do AWS IoT Greengrass Version 2 desenvolvedor:

- [AWS IoT Greengrass Comandos CLI](#)
- [Implemente AWS IoT Greengrass componentes em dispositivos](#)

8. Envie configurações para o aplicativo usando o AWS CLI

- Crie um novo arquivo, *example-edge-configuration.json*.

Cole o seguinte código no arquivo. Esse é um exemplo de configuração que grava diariamente das 9:00 às 16:59:59 (de acordo com a hora do sistema em seu dispositivo). AWS IoT Ele também carrega a mídia gravada diariamente das 19:00:00 às 21:59:59.

Para ter mais informações, 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. Digite o seguinte AWS CLI para enviar o arquivo para o agente do Amazon Kinesis Video Streams Edge:

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

9. Repita a etapa anterior para cada stream do Amazon Kinesis Video Streams Edge Agent.

Etapa 7: (opcional) instalar o componente do gerenciador de AWS IoT Greengrass registros no dispositivo

Note

Esteja ciente das [CloudWatchcotas](#).

Siga esses procedimentos para configurar os logs do Amazon Kinesis Video Streams Edge Agent para serem CloudWatch carregados automaticamente AWS IoT Greengrass usando o componente gerenciador de registros.

Instale o componente do gerenciador de AWS IoT Greengrass registros

1. Confirme se a função do AWS IoT Greengrass dispositivo tem as [permissões apropriadas](#).
 - a. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
 - b. Clique em Funções no painel de navegação à esquerda.

- c. Escolha o nome da função TES criada em [the section called “2. Configurar o dispositivo AWS IoT Greengrass principal”](#). Use a barra de pesquisa, se necessário.
- d. Selecione a política GreengrassV2TokenExchangeRoleAccess.
- e. Selecione a guia JSON e verifique se a política tem a seguinte aparência:

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

- f. Se a GreengrassV2TokenExchangeRoleAccess política não existir ou se algumas permissões necessárias estiverem ausentes, crie uma nova política do IAM com essas permissões e anexe-a à função TES criada em [the section called “2. Configurar o dispositivo AWS IoT Greengrass principal”](#).
2. Faça login no AWS Management Console e abra o AWS IoT Core console em <https://console.aws.amazon.com/iot/>. Verifique se a região apropriada está selecionada.
 3. No painel de navegação à esquerda, escolha Greengrass devices, Deployments.

Escolha a implantação com o mesmo destino da coisa em que você criou [the section called “2. Configurar o dispositivo AWS IoT Greengrass principal”](#).
 4. No canto superior direito, selecione Ações e, em seguida, escolha Revisar.

No pop-up exibido, escolha Revisar implantação.
 5. Complete as seguintes seções:
 - a. Etapa 1: especifique o alvo. Escolha Próximo.
 - b. Etapa 2: Selecione os componentes.

- i. Verifique se o componente `aws.greengrass.cli` e `aws.greengrass.SecretManager` componentes ainda estão selecionados.

⚠ Important

Não desinstale esses componentes.

- ii. Ative a opção `Mostrar somente componentes selecionados` e pesquise por `aws.greengrass.LogManager`.
 - iii. Selecione a caixa ao lado de `aws.greengrass.LogManager`, em seguida, escolha `Avançar`.
- c. Etapa 3: Configurar componentes. Configure o componente do gerenciador de AWS IoT Greengrass registros para fazer o upload dos registros gerados pelo Amazon Kinesis Video Streams Edge Agent.

Selecione o `aws.greengrass.LogManager` componente e, em seguida, escolha `Configurar componente`.

Na tela exibida, cole a seguinte configuração do gerenciador de registros na caixa `Configuração para mesclar`.

```
{
  "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` Na configuração anterior, pressupõe que o local de saída de registro padrão seja usado.

Note

Para obter mais informações sobre cada um dos parâmetros da configuração do gerenciador de registros, consulte a seção [Gerenciador de registros](#) do Guia do AWS IoT Greengrass Version 2 desenvolvedor.

Ao terminar, selecione Confirmar e, em seguida, escolha Avançar.

- d. Etapa 4: Defina as configurações avançadas. Escolha Próximo.
 - e. Etapa 5: Revise. Selecione Implantar.
6. Confirme se o componente do gerenciador de AWS registros e as permissões foram instalados corretamente.
 7. Na instância do Ubuntu Amazon EC2, digite `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` para verificar se o componente recebeu a configuração atualizada.

8. Inspecione os registros AWS IoT Greengrass principais.

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

Verifique se há erros de implantação.

Se houver um erro, revise a implantação para remover o `aws.greengrass.LogManager` componente.

Digite `sudo service greengrass restart` para reiniciar o serviço AWS IoT Greengrass principal.

Se o erro de implantação estiver relacionado à falta de permissões, verifique [the section called “4. Adicionar permissões à função TES”](#) se a função TES tem as permissões adequadas. Em seguida, repita esta seção.

Perguntas frequentes sobre o Amazon Kinesis Video Streams Edge Agent

A seguir estão algumas perguntas comuns sobre o serviço Amazon Kinesis Video Streams Edge Agent.

Quais sistemas operacionais são compatíveis com o Amazon Kinesis Video Streams Edge Agent?

Atualmente, o Amazon Kinesis Video Streams Edge Agent é compatível com os seguintes sistemas operacionais:

Ubuntu

- 22.x
 - AMD64
- 18.x
 - Arm

AL2

- amzn2

- AMD64 amazonlinux:2.0.20210219.0-amd64 (Snowball)

O Amazon Kinesis Video Streams Edge Agent oferece suporte à mídia H.265?

O Amazon Kinesis Video Streams Edge Agent suporta somente streams elementares H.264.

O Amazon Kinesis Video Streams Edge Agent funciona no AL2?

Sim.

Como posso executar vários streams dentro da AWS IoT coisa ou dispositivo?

Envie outro [the section called "StartEdgeConfigurationUpdate"](#) para o mesmo HubDeviceArn Amazon Kinesis Video AWS Secrets Manager Streams/ARNs, mas diferente.

Como posso editar um **StartEdgeConfigurationUpdate** depois de ter sido enviado?

Envie uma atualização [the section called "StartEdgeConfigurationUpdate"](#) para o mesmo HubDeviceArn com o mesmo ARN do Amazon Kinesis Video Streams. Quando o aplicativo recebe a mensagem do Amazon Kinesis Video Streams, ele substitui a configuração anterior desse stream. As mudanças ocorrerão então.

Você tem algum exemplo em comum **ScheduleConfigs**?

O Amazon Kinesis Video Streams Edge Agent usa a hora do sistema do dispositivo em que está sendo executado.

Descrição	ScheduleExpression	DurationInSeconds
Gravação 24 horas por dia, 7 dias por semana, carregamento por hora	(null ScheduleConfig)	
9:00 da manhã - 16:59:59 da tarde todos os dias	0 0 9-16 * * ? *	3599

Descrição	ScheduleExpression	DurationInSeconds
9:00 da manhã - 16:59:59 da tarde nos dias úteis	0 0 9-16 ? * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599
9:00 da manhã - 16:59:59 da tarde nos fins de semana	0 0 9-16 ? * SAT,SUN *	3599
22:00 - 23:59:59 nos dias úteis	0 0 22,23 ? * MON-FRI *	3599
9:00 da manhã - 10:00:00 da manhã todos os dias	0 0 9 * * ? *	3600
16:00 - 17:59:59 todos os dias	0 0 16-17 * * ? *	3599

Para obter mais exemplos, consulte a [documentação do Quartz](#).

Existe um limite máximo de transmissão?

Atualmente, o Amazon Kinesis Video Streams Edge Agent tem um limite rígido de 16 streams por dispositivo. Use a [the section called “DeleteEdgeConfiguration”](#) API para excluir streams de um dispositivo. Atualizar uma configuração para o mesmo stream usando o [the section called “StartEdgeConfigurationUpdate”](#) não aumenta a contagem de streams do dispositivo.

Como faço para reiniciar um trabalho com erro?

Se um erro for encontrado, o Amazon Kinesis Video Streams Edge Agent tentará reiniciar o trabalho. No entanto, com alguns erros (como erros de configuração), você deve reiniciar manualmente o trabalho.

Para determinar quais trabalhos precisam ser reiniciados manualmente, consulte a `FatalError` métrica em [the section called “Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch”](#).

Reenvie o [the section called “StartEdgeConfigurationUpdate”](#) para reiniciar o trabalho do stream.

Como faço para monitorar a integridade do meu Amazon Kinesis Video Streams Edge Agent?

Para ter mais informações, consulte [the section called “Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch”](#).

Transmita vídeo por meio de uma VPC

Esta versão beta está disponível em versão prévia na região da Europa (Paris), eu-west-3. Para acessar esses componentes e nosso guia de introdução, [envie-nos um e-mail](#).

O serviço de endpoint VPC Amazon Kinesis Video Streams permite que você transmita e consuma vídeo pela rede Amazon sem que nenhum dado passe pela Internet pública.

Para solicitar acesso, [envie-nos um e-mail](#) com as seguintes informações:

- ID da conta
- Transmita RNs
- ID da VPC

Note

Pode levar até uma semana para adicionarmos você ao serviço.

Se você não trabalhou com VPC endpoints no passado, analise as informações a seguir para se familiarizar com o conceito:

- [AWS PrivateLink plano de fundo](#)
- [Guia de introdução ao VPC](#)

Mais informações

Depois que você for adicionado à versão beta, enviaremos um link por e-mail com informações adicionais sobre esse recurso.

Procedimentos de endpoint de VPC

Cotas

As principais diferenças de cota são:

- Cota menor para todas as APIs de largura de banda (2 mbps):
 - PutMedia
 - GetMedia
 - GetMediaForFragmentList
- 10 transmissões permitidas por cliente

Criar um endpoint

Depois de ser listado como autorizado, você receberá o nome do serviço de VPC endpoint para o Amazon Kinesis Video Streams. Vai parecer que `simcom.amazonaws.region.kinesisvideo`.

Crie uma [interface VPC endpoint](#) para o Amazon Kinesis Video Streams usando o console Amazon VPC ou o (). AWS Command Line Interface AWS CLI

No AWS CLI, digite o seguinte:

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

Important

O tráfego em sua VPC usará DNS privado para rotear pelo endpoint. Se você não habilitar isso, precisará implementar sua própria lógica de DNS. Para obter mais informações sobre DNS privado, consulte a [AWS PrivateLink documentação](#).

Para obter mais informações sobre a AWS CLI opção, consulte [create-vpc-endpoint](#).

Controle o acesso aos endpoints

Você pode anexar uma política de endpoint ao seu VPC endpoint que controla o acesso ao Amazon Kinesis Video Streams. Essa política especifica as seguintes informações:

- o principal que pode realizar ações,
- as ações que podem ser executadas, e
- os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte [Controle do acesso a serviços com VPC endpoints usando políticas de endpoint](#) no Guia. AWS PrivateLink

Veja a seguir um exemplo de uma política de endpoint para o Amazon Kinesis Video Streams. Quando anexada a um endpoint, essa política nega o acesso às PutMedia ações listadas para todos os diretores em todos os recursos.

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

Imagens nos streams de vídeo do Kinesis

Você pode usar as APIs e SDKs do Amazon Kinesis Video Streams para ajudá-lo a extrair imagens de seus streams de vídeo. Você pode usar essas imagens para aplicativos de reprodução aprimorados, como miniaturas ou depuração aprimorada, ou para uso em canais de aprendizado de máquina. O Kinesis Video Streams oferece extração de imagens sob demanda por meio de uma API ou extração automatizada de imagens de tags de metadados em um vídeo ingerido.

Para obter informações sobre como usar o suporte gerenciado do Kinesis Video Streams para imagens, consulte:

- [Geração de imagens sob demanda \(GetImages\)](#)- Essa API permite que os clientes extraiam uma única imagem ou várias imagens do vídeo armazenado no Kinesis Video Streams.
- [Geração automatizada de imagens \(entrega S3\)](#) - Configure o Kinesis Video Streams para extrair automaticamente imagens dos dados de vídeo em tempo real com base nas tags em um vídeo enviado e entregar as imagens no bucket do S3 especificado pelo cliente.

Tópicos

- [Conceitos básicos do GetImages](#)
- [Primeiros passos com a entrega do Amazon S3](#)

Conceitos básicos do GetImages

O suporte gerenciado para imagens fornece uma maneira totalmente gerenciada de obter imagens dos dados de vídeo transmitidos e armazenados no Kinesis Video Streams. Você pode usar imagens para executar cargas de trabalho de aprendizado de máquina (ML), como detecção de pessoas, animais de estimação ou veículos. As imagens também podem ser usadas para adicionar elementos interativos à reprodução, como visualizações de imagens para eventos de movimento e depuração para um videoclipe.

Para obter mais informações sobre o [GetImages](#) recurso, veja [GetImages](#) na O Amazon Kinesis Video transmite mídia arquivada Guia de referência da API.

Primeiros passos com a entrega do Amazon S3

Atualmente, os clientes executam e gerenciam seu próprio pipeline de transcodificação de imagens para criar imagens para várias finalidades, como depuração, visualização de imagens, execução de modelos de ML em imagens e muito mais. O Kinesis Video Streams oferece a capacidade de transcodificar e entregar as imagens. O Kinesis Video Streams extrairá automaticamente imagens dos dados de vídeo em tempo real com base em uma tag e entregará as imagens a um bucket S3 especificado pelo cliente.

UpdateImageGenerationConfiguration

Para configurar um stream de vídeo do Kinesis para permitir a geração de imagens para o Amazon S3:

1. Crie um Balde S3 para geração de imagens com base nas tags adicionadas no SDK usando a nova API. Observe o TIPO S3, que é exigido na próxima etapa, ao atualizar as configurações de geração de imagens para os fluxos.
2. Crie um arquivo JSON chamado `update-image-generation-input.json` com o seguinte conteúdo como entrada.

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

```
}
```

Você pode usar o AWS CLI para invocar o [UpdateImageGenerationConfiguration](#) Operação de API para adicionar o ARN do Amazon S3 criado anteriormente e alterar o status para ENABLED.

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

```

Solicitação:

UpdateImageGenerationConfiguration HTTP/1.1

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

Resposta:

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

```
Body: {  
}
```

Note

Leva pelo menos 1 minuto para iniciar o fluxo de trabalho de geração de imagem depois de atualizar a configuração de geração de imagem. Espere pelo menos 1 minuto antes de invocar `PutMedia` após a chamada de atualização.

DescribeImageGenerationConfiguration

Para visualizar as configurações de geração de imagens que já estão definidas para um stream, os clientes podem criar uma `DescribeImageGenerationConfiguration` solicitação, da seguinte forma.

Solicitação:

```
DescribeImageGenerationConfiguration HTTP/1.1
```

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

Resposta:

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

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

Para saber mais sobre o `DescribeImageGenerationConfiguration` recurso, veja [DescribeImageGenerationConfiguration](#) na Guia do desenvolvedor do Amazon Kinesis Video Streams.

Tags MKV do produtor

Você pode usar o SDK do Kinesis Video Streams Producer para marcar fragmentos específicos de interesse expondo uma operação de API no SDK. Para ver um exemplo de tag, consulte [este código](#). Ao chamar essa API, o SDK adicionará um conjunto de tags MKV predefinidas junto com os dados do fragmento. O Kinesis Video Streams reconhecerá essas tags MKV especiais e iniciará o fluxo de trabalho de geração de imagens com base na configuração de processamento de imagem desse stream.

Todos os metadados de fragmento fornecidos junto com as tags de geração de imagem do Amazon S3 serão salvos como metadados do Amazon S3.

Sintaxe para tags MKV do Producer

```
|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger image generation for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION
|
| // OPTIONAL: S3 prefix which will be set as prefix for generated image.
```

```

| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_PREFIX
| + String: image_prefix_in_s3 // 256 bytes max m

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

```

Adicionar tags de metadados no Producer SDK usando PutEventMetaData

O `PutEventMetaData` função acrescenta um arquivo MKV associado a um evento. `PutEventMetaData` usa dois parâmetros. O primeiro parâmetro é um evento cujo valor vem do `STREAM_EVENT_TYPE` enumeração. O segundo parâmetro, [pStreamEventMetadata](#), é opcional e pode ser usado para incluir metadados adicionais como pares de valores-chave. Há um limite de cinco pares de metadados de valores-chave que podem ser adicionados.

Limites

A tabela a seguir lista as limitações associadas às tags de metadados. Se o limite de tags de metadados for ajustável, você poderá solicitar um aumento por meio de seu gerente de conta.

Limite	Valor máximo	Ajustável
Comprimento do prefixo da imagem	256	não
Tamanho opcional da chave de metadados	128	não
Tamanho opcional do valor dos metadados	256	não
Número máximo de metadados opcionais	10	sim

Metadados do objeto S3

Por padrão, o Kinesis Video Streams definirá o número do fragmento, produtor, e carimbo de data/hora do servidor da imagem gerada como metadados de objeto do Amazon S3. Se algum dado de fragmento adicional for especificado nas tags MKV, essas tags também serão adicionadas aos metadados do objeto do Amazon S3. O exemplo a seguir mostra a sintaxe correta para os metadados de objetos do Amazon S3.

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

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

Caminho do objeto S3 (imagem)

A lista a seguir mostra o formato correto do caminho do objeto e descreve cada elemento dentro do caminho.

Formato:

ImagePrefix_ID da conta_StreamName_ImageTimecode_ID aleatória.extensão de arquivo

1. ImagePrefix - Valor de `AWS_KINESISVIDEO_IMAGE_PREFIX`.
2. AccountID - ID da conta sob a qual o stream é criado.
3. StreamName - Nome do stream para o qual a imagem é gerada.
4. ImageTimecode - Timecode de época no fragmento em que a imagem é gerada.
5. RandomID - GUID aleatório.
6. file-extension - JPG ou PNG com base no formato de imagem solicitado.

Recomendações de URI do Amazon S3 para proteção contra limitação

Se você gravar milhares de imagens no Amazon S3, existe o risco de limitação. Para obter mais informações, consulte [Limites da solicitação S3 Prefix Put](#).

Um prefixo do Amazon S3 começa com um limite PUT de 3.500 solicitações PUT por segundo e aumentará gradualmente com o tempo para prefixos exclusivos. Evite usar datas e horas como prefixos do Amazon S3. Os dados codificados por tempo afetarão um prefixo por vez e também serão alterados regularmente, invalidando os aumentos de prefixo anteriores. Para permitir uma escalabilidade mais rápida e consistente do Amazon S3, recomendamos adicionar um prefixo aleatório, como um código hexadecimal ou UUID ao URI de destino do Amazon S3. Por exemplo, prefixos de código hexadecimal naturalmente dividirão suas solicitações aleatoriamente entre 16 prefixos diferentes (um prefixo para cada caractere hexadecimal exclusivo), o que permitirá 56.000 solicitações PUT por segundo após a escalabilidade automática do Amazon S3.

Notificações no Kinesis Video Streams

Quando um fragmento de mídia está disponível para consumo, o Kinesis Video Streams notifica os clientes usando notificações (Amazon Simple Notification Service Amazon SNS). O tópico a seguir explica como começar a usar as notificações.

UpdateNotificationConfiguration

Use essa operação de API para atualizar as informações de notificação de um stream. Para obter mais informações sobre o `UpdateNotificationConfiguration` recurso, consulte o [UpdateNotificationConfiguration](#) Guia do desenvolvedor do Amazon Kinesis Video Streams.

Note

Leva pelo menos um minuto para iniciar a notificação depois de atualizar a configuração da notificação. Aguarde pelo menos um minuto antes de invocar `PutMedia` após a chamada de atualização.

DescribeNotificationConfiguration

Use essa API para descrever uma configuração de notificação anexada a um stream. Para obter mais informações sobre o `DescribeNotificationConfiguration` recurso, consulte o [DescribeNotificationConfiguration](#) Guia do desenvolvedor do Amazon Kinesis Video Streams.

Etiquetas MKV do produtor

Você pode usar o SDK do Kinesis Video Streams Producer para marcar fragmentos específicos de interesse expondo uma operação de API no SDK. Veja um exemplo de como isso funciona [nesta seção de código](#). Ao chamar essa API, o SDK adicionará um conjunto de tags MKV predefinidas junto com os dados do fragmento. O Kinesis Video Streams reconhecerá essas tags MKV especiais e iniciará notificações para os fragmentos marcados.

Todos os metadados de fragmento fornecidos junto com as tags MKV de notificação serão publicados como parte da carga útil do tópico do Amazon SNS.

Sintaxe para tags MKV do Producer

```

|+ 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

```

Limites de tags MKV

A tabela a seguir lista as limitações associadas às tags de metadados. Se o limite da tag de metadados for ajustável, você poderá solicitar um aumento por meio do seu gerente de conta.

Limite	Valor máximo	Ajustável
Tamanho opcional da chave de metadados	128	Não
Tamanho opcional do valor dos metadados	256	Não
Número máximo de metadados opcionais	10	Sim

Carga útil de tópicos do Amazon SNS

Qualquer notificação iniciada por meio do fluxo de trabalho anterior entregará a carga útil do tópico do Amazon SNS, conforme mostrado no exemplo a seguir. Este exemplo é uma mensagem do

Amazon SNS que ocorre após o consumo de dados de notificação de uma fila 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
  }
}
```

Visualizando suas mensagens do Amazon SNS

Você não pode ler mensagens diretamente de um tópico do Amazon SNS porque não há API para fazer isso. Para visualizar as mensagens, inscreva uma fila SQS no tópico SNS ou escolha qualquer outro destino compatível com o [Amazon SNS](#). No entanto, a opção mais eficiente para visualizar mensagens é usar o Amazon SQS.

Para visualizar suas mensagens do Amazon SNS usando o Amazon SQS

1. Crie uma fila [do Amazon SQS](#).
2. AWS Management Console Em, abra o tópico do Amazon SNS definido como destino em. NotificationConfiguration
3. Escolha Create Subscription e, em seguida, escolha a fila do Amazon SQS criada na primeira etapa.
4. Execute uma PutMedia sessão com a configuração de notificação ativada e com as tags MKV de notificação adicionadas aos fragmentos.
5. Escolha a fila do Amazon SQS no console do Amazon SQS e, em seguida, selecione Enviar e receber mensagens para a fila do Amazon SQS.
6. Pesquisa de mensagens. Esse comando deve mostrar todas as notificações geradas pela PutMedia sessão. Para obter informações sobre pesquisas, consulte sondagem [curta e longa do Amazon SQS](#).

Segurança no Amazon Kinesis Video Streams

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficiará de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isto como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. A eficácia da nossa segurança é regularmente testada e verificada por auditores de terceiros como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao Kinesis Video Streams, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, como a confidencialidade de seus dados, os requisitos da sua organização, leis e regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Kinesis Video Streams. Os tópicos a seguir mostram como configurar o Kinesis Video Streams para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que podem ajudá-lo a monitorar e proteger seus recursos do Kinesis Video Streams.

Tópicos

- [Proteção de dados no Kinesis Video Streams](#)
- [Controle do acesso aos recursos do Kinesis Video Streams usando o IAM](#)
- [Controlando o acesso aos recursos do Kinesis Video Streams usando AWS IoT](#)
- [Monitoramento dos Amazon Kinesis Video Streams](#)
- [Validação de conformidade para Amazon Kinesis Video Streams](#)
- [Resiliência no Amazon Kinesis Video Streams](#)
- [Segurança da infraestrutura no Kinesis Video Streams](#)
- [Melhores práticas de segurança para o Kinesis Video Streams](#)

Proteção de dados no Kinesis Video Streams

Você pode usar a criptografia do lado do servidor (SSE) usando chaves AWS Key Management Service (AWS KMS) para atender aos rígidos requisitos de gerenciamento de dados criptografando seus dados em repouso no Amazon Kinesis Video Streams.

Tópicos

- [O que é criptografia do lado do servidor para o Kinesis Video Streams?](#)
- [Custos, regiões e considerações de desempenho](#)
- [Como faço para começar a usar a criptografia do lado do servidor?](#)
- [Criação e uso de uma chave gerenciada pelo cliente](#)
- [Permissões para usar uma chave gerenciada pelo cliente](#)

O que é criptografia do lado do servidor para o Kinesis Video Streams?

A criptografia do lado do servidor é um recurso do Kinesis Video Streams que criptografa automaticamente os dados antes de serem armazenados em repouso usando uma chave especificada por você. AWS KMS Os dados são criptografados antes de serem gravados na camada de armazenamento de streams do Kinesis Video Streams e são descriptografados após serem recuperados do armazenamento. Como resultado, seus dados são sempre criptografados em repouso no serviço Kinesis Video Streams.

Com a criptografia do lado do servidor, seus produtores e consumidores de streaming de vídeo do Kinesis não precisam gerenciar chaves KMS ou operações criptográficas. Se a retenção de dados estiver ativada, seus dados serão criptografados automaticamente ao entrarem e saírem do Kinesis Video Streams, então seus dados em repouso serão criptografados. AWS KMS fornece todas as chaves usadas pelo recurso de criptografia do lado do servidor. AWS KMS simplifica o uso de uma chave KMS para o Kinesis Video Streams que é gerenciada AWS por, uma chave especificada pelo AWS KMS usuário importada para o serviço. AWS KMS

Custos, regiões e considerações de desempenho

Ao aplicar a criptografia do lado do servidor, você está sujeito ao uso AWS KMS da API e aos custos principais. Ao contrário das AWS KMS chaves personalizadas, a chave (Default) `aws/kinesis-video KMS` é oferecida gratuitamente. No entanto, você ainda precisa pagar pelos custos de uso da API que o Kinesis Video Streams incorre em seu nome.

Os custos de uso da API se aplicam a todas as chaves KMS, inclusive as personalizadas. Os AWS KMS custos aumentam de acordo com o número de credenciais de usuário que você usa em seus produtores e consumidores de dados, pois cada credencial de usuário exige uma chamada de API exclusiva para. AWS KMS

Veja a seguir a descrição dos custos por recurso:

Chaves

- A chave KMS para Kinesis Video Streams que é gerenciada AWS por (alias `aws/kinesis-video` =) é gratuita.
- As chaves KMS geradas pelo usuário estão sujeitas a AWS KMS key custos. Para obter mais informações, consulte [Preços do AWS Key Management Service](#).

AWS KMS Uso da API

As solicitações de API para gerar novas chaves de criptografia de dados ou recuperar chaves de criptografia existentes aumentam à medida que o tráfego aumenta e estão sujeitas a custos de AWS KMS uso. Para obter mais informações, consulte [AWS Key Management Service Preços: Uso](#).

O Kinesis Video Streams gera solicitações importantes mesmo quando a retenção está definida como 0 (sem retenção).

Disponibilidade de criptografia do lado do servidor por região

A criptografia do lado do servidor dos streams de vídeo do Kinesis está disponível em todos os locais em que o Kinesis Video Regiões da AWS Streams está disponível.

Como faço para começar a usar a criptografia do lado do servidor?

A criptografia do lado do servidor está sempre ativada no Kinesis Video Streams. Se uma chave fornecida pelo usuário não for especificada quando o stream for criado, a Chave gerenciada pela AWS (fornecida pelo Kinesis Video Streams) será usada.

Uma chave KMS fornecida pelo usuário deve ser atribuída a um stream de vídeo do Kinesis quando ele é criado. Você não pode atribuir uma chave diferente a um stream usando a [UpdateStreamAPI](#) posteriormente.

Você pode atribuir uma chave KMS fornecida pelo usuário a um stream de vídeo do Kinesis de duas maneiras:

- Ao criar um stream de vídeo do Kinesis no AWS Management Console, especifique a chave KMS na guia Criptografia na página Criar um novo stream de vídeo.
- Ao criar um stream de vídeo do Kinesis usando a [CreateStream](#) API, especifique o ID da chave no `KmsKeyId` parâmetro.

Criação e uso de uma chave gerenciada pelo cliente

Esta seção descreve como criar e usar suas próprias chaves KMS em vez de usar a chave administrada pelo Amazon Kinesis Video Streams.

Criar uma chave gerenciada pelo cliente

Para obter informações sobre como criar suas próprias chaves, consulte [Criação de chaves](#) no Guia do AWS Key Management Service desenvolvedor. Depois de criar as chaves para sua conta, o serviço Kinesis Video Streams retorna essas chaves na lista de chaves gerenciadas pelo cliente.

Com usar uma chave gerenciada pelo cliente.

Depois que as permissões corretas forem aplicadas aos seus consumidores, produtores e administradores, você poderá usar chaves KMS personalizadas na sua própria Conta da AWS ou em outra. Conta da AWS Todas as chaves KMS em sua conta aparecem na lista de chaves gerenciadas pelo cliente no console.

Para usar chaves KMS personalizadas localizadas em outra conta, você deve ter permissões para usar essas chaves. Você também deve criar o streaming usando a API `CreateStream`. Você não pode usar chaves KMS de contas diferentes em fluxos criados no console.

Note

A chave KMS não é acessada até que a `GetMedia` operação `PutMedia` ou seja executada. Estes são os resultados disso:

- Se a chave especificada não existir, a `CreateStream` operação será bem-sucedida, mas `PutMedia` `GetMedia` as operações no stream falharão.
- Se você usar a chave fornecida (`aws/kinesis-video`), a chave não estará presente na sua conta até que a primeira `GetMedia` operação `PutMedia` ou operação seja executada.

Permissões para usar uma chave gerenciada pelo cliente

Antes de usar a criptografia do lado do servidor com uma chave gerenciada pelo cliente, você deve configurar as políticas de chaves do KMS para permitir a criptografia de fluxos e a criptografia e descryptografia de registros de fluxo. Para exemplos e mais informações sobre AWS KMS permissões, consulte [Permissões de AWS KMS API: referência de ações e recursos](#).

Note

O uso da chave de serviço padrão para criptografia não exige a aplicação de permissões personalizadas do IAM.

Antes de usar uma chave gerenciada pelo cliente, verifique se os produtores e consumidores de stream de vídeo do Kinesis (principais do IAM) são usuários na política de chaves AWS KMS padrão. Caso contrário, as gravações e as leituras de um streaming falharão, o que pode resultar, em última análise, em perda de dados, processamento atrasado, ou travamento de aplicativos. Você pode gerenciar permissões para chaves do KMS usando políticas do IAM. Para obter mais informações, consulte Como [usar políticas do IAM com AWS KMS](#).

Exemplo de permissões de produtor

Seus produtores de stream de vídeo do Kinesis devem ter a `kms:GenerateDataKey` permissão:

```
{
  "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"
```



```
}  
]  
}
```

Exemplo de permissões do consumidor

Seus consumidores de streaming de vídeo do Kinesis devem ter a `kms:Decrypt` permissão:

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

Controle do acesso aos recursos do Kinesis Video Streams usando o IAM

Você pode usar o AWS Identity and Access Management (IAM) com o Amazon Kinesis Video Streams para controlar se os usuários da sua organização podem realizar uma tarefa usando operações específicas da API do Kinesis Video Streams e se podem usar recursos específicos. AWS

Para obter mais informações sobre IAM, consulte o seguinte:

- [AWS Identity and Access Management \(IAM\)](#)
- [Conceitos básicos](#)

- [Guia do usuário do IAM](#)

Conteúdo

- [Sintaxe da política](#)
- [Ações para o Kinesis Video Streams](#)
- [Nomes de recursos da Amazon \(ARNs\) para o Kinesis Video Streams](#)
- [Conceder acesso a outras contas do IAM a um stream de vídeo do Kinesis](#)
- [Exemplos de políticas para o Kinesis Video Streams](#)

Sintaxe da política

A política do IAM é um documento JSON que consiste em uma ou mais declarações. Cada instrução é estruturada da seguinte maneira:

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

Existem vários elementos que compõem uma instrução:

- **Efeito** — O efeito pode ser Allow ou Deny. Por padrão, os usuários não têm permissão para usar recursos e ações da API. Por isso, todas as solicitações são negadas. Um permitir explícito substitui o padrão. Uma negar explícito substitui todas as permissões.
- **Ação** — A ação é a ação específica da API para a qual você está concedendo ou negando permissão.
- **Recurso** — O recurso afetado pela ação. Para especificar um recurso na declaração, você deve usar o respectivo nome de recurso da Amazon (ARN).

- **Condição:** as condições são opcionais. Elas podem ser usadas para controlar quando a política está em vigor.

Ao criar e gerenciar políticas do IAM, recomendamos que você use o [IAM Policy Generator](#) e o [IAM Policy Simulator](#).

Ações para o Kinesis Video Streams

Em uma declaração de política do IAM, é possível especificar qualquer ação de API de qualquer serviço que dê suporte ao IAM. Para Kinesis Video Streams, use o seguinte prefixo com o nome da ação da API: `kinesisvideo:`. Por exemplo: `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams` e `kinesisvideo:DescribeStream`.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

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

Também é possível especificar várias ações usando asteriscos. Por exemplo, você pode especificar todas as ações cujo nome começa com a palavra "Obter", conforme o seguinte:

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

Para especificar todas as operações do Kinesis Video Streams, use o curinga asterisco (*), como a seguir:

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

Para obter a lista completa das ações da API do Kinesis Video Streams, consulte a [referência de API do Kinesis Video Streams](#).

Nomes de recursos da Amazon (ARNs) para o Kinesis Video Streams

Cada declaração de política do IAM se aplica aos recursos que você especifica usando os ARNs.

Use o seguinte formato de recursos do ARN para os fluxos de dados do Kinesis Video Streams:

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

Por exemplo: .

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

Você pode obter o ARN de um stream usando o [DescribeStream](#)

Conceder acesso a outras contas do IAM a um stream de vídeo do Kinesis

Talvez seja necessário conceder permissão a outras contas do IAM para realizar operações em streams no Kinesis Video Streams. A seguinte visão geral descreve as etapas para concessão de acesso aos streamings de vídeo em várias contas:

1. Obtenha o ID da conta de 12 dígitos da conta à qual você deseja conceder permissões para realizar operações no recurso de stream criado em sua conta.

Exemplo: nas etapas a seguir, usaremos 111111111111 como o ID da conta para a qual você deseja conceder permissão e 999999999999 como o ID do seu Kinesis Video Streams

2. Crie uma política gerenciada do IAM na conta proprietária do stream (999999999999) que permita o nível de acesso que você deseja conceder.

Política de amostra:

```
{
  "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 outros exemplos de políticas para os recursos do Kinesis Video Streams [Exemplo de políticas](#), consulte na próxima seção.

3. Crie uma função na conta proprietária do stream (999999999999) e especifique a conta para a qual você deseja conceder permissões (111111111111). Isso adicionará uma entidade confiável à função.

Exemplo de política confiável:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Anexe a política que você criou na etapa anterior a essa função.

Agora você criou uma função na conta 999999999999 que tem permissão para operações como `DescribeStreamGetDataEndpoint`, e em `PutMedia` um ARN de recurso de stream na política gerenciada. Essa nova função também confia na outra conta, 111111111111, para assumir essa função.

Important

Anote o ARN da função, você precisará dele na próxima etapa.

4. Crie uma política gerenciada na outra conta, 111111111111, que permita a `AssumeRole` ação na função que você criou na conta 999999999999 na etapa anterior. Você precisará mencionar o ARN da função na etapa anterior.

Política de amostra:

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

```

    "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
  }
}

```

5. Anexe a política criada na etapa anterior a uma entidade do IAM, como uma função ou um usuário na conta 111111111111. Esse usuário agora tem permissão para assumir a função CustomRoleName na conta 999999999999.

As credenciais desse usuário chamam a AWS STS AssumeRole API para obter as credenciais da sessão, que são usadas posteriormente para chamar as APIs do Kinesis Video Streams no stream criado na conta 999999999999.

```

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. Defina a chave de acesso, a chave secreta e as credenciais da sessão com base no conjunto anterior no ambiente.

```

set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=

```

7. Execute as APIs do Kinesis Video Streams para descrever e obter o endpoint de dados do stream na conta 999999999999.

```

aws kinesismedia 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 step-by-step obter instruções genéricas sobre como conceder acesso entre contas, consulte [Delegar acesso ao Contas da AWS uso de funções do IAM](#).

Exemplos de políticas para o Kinesis Video Streams

Os exemplos de políticas a seguir demonstram como você pode controlar o acesso do usuário ao seu Kinesis Video Streams.

Example 1: Permita que os usuários obtenham dados de qualquer stream de vídeo do Kinesis

Essa política permite que um usuário ou grupo realize as `ListTagsForStream` operações `DescribeStream`, `GetDataEndpoint`, `GetMediaListStreams`, e em qualquer stream de vídeo da Kinesis. É apropriada para usuários que podem obter dados de qualquer streaming de vídeo.

```

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

```

```

        "Resource": "*"
    }
]
}

```

Example 2: Permitir que um usuário crie um stream de vídeo do Kinesis e grave dados nele

Esta política permite que um usuário ou grupo execute as operações `CreateStream` e `PutMedia`. Esta política é apropriada para uma câmera de segurança que pode criar um streaming de vídeo e enviar dados para ele.

```

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

```

Example 3: Permitir que um usuário tenha acesso total a todos os recursos do Kinesis Video Streams

Esta política permite que um usuário ou grupo execute qualquer operação do Kinesis Video Streams em qualquer um dos recursos. Esta política é apropriada para administradores.

```

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

```


Example 4: Permitir que um usuário grave dados em um stream de vídeo específico do Kinesis

Esta política permite que um usuário ou grupo grave dados em um determinado streaming. Esta política é apropriada para um dispositivo que pode enviar dados para um streaming único.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
your_stream/0123456789012"
    }
  ]
}
```

Controlando o acesso aos recursos do Kinesis Video Streams usando AWS IoT

Esta seção descreve como permitir que um dispositivo (por exemplo, uma câmera) envie dados de áudio e vídeo somente para um determinado stream de vídeo do Kinesis. Você pode fazer isso usando o provedor de AWS IoT credenciais e uma função AWS Identity and Access Management (IAM).

Os dispositivos podem usar certificados X.509 para se conectar AWS IoT usando protocolos de autenticação mútua TLS. Outros Serviços da AWS (por exemplo, o Kinesis Video Streams) não oferecem suporte à autenticação baseada em certificado, mas podem ser AWS chamados usando AWS credenciais no formato Signature versão 4. O algoritmo Signature Version 4 normalmente exige que o chamador tenha um ID de chave de acesso e uma chave de acesso secreta. AWS IoT tem um provedor de credenciais que permite usar o certificado X.509 integrado como a identidade exclusiva do dispositivo para autenticar AWS solicitações (por exemplo, solicitações para o Kinesis Video Streams). Isso elimina a necessidade de armazenar um ID de chave de acesso e uma chave de acesso secreta em seu dispositivo.

O provedor de credenciais autentica um cliente (nesse caso, um SDK do Kinesis Video Streams executado na câmera para a qual você deseja enviar dados para um stream de vídeo) usando um certificado X.509 e emite um token de segurança temporário com privilégios limitados. Você pode

usar o token para assinar e autenticar qualquer AWS solicitação (nesse caso, uma chamada para o Kinesis Video Streams). Para obter mais informações, consulte [Autorização de chamadas diretas para AWS serviços](#).

Essa forma de autenticar as solicitações da sua câmera no Kinesis Video Streams exige que você crie e configure uma função do IAM e anexe políticas apropriadas do IAM à função para que o provedor de credenciais possa assumir AWS IoT a função em seu nome.

Para obter mais informações sobre AWS IoT, consulte a [AWS IoT Core documentação](#). Para obter mais informações sobre o IAM, consulte [AWS Identity and Access Management \(IAM\)](#).

Tópicos

- [AWS IoT ThingName como nome do stream](#)
- [AWS IoT CertificateId como nome do stream](#)
- [Use AWS IoT credenciais para transmitir para um nome de fluxo codificado](#)

AWS IoT ThingName como nome do stream

Tópicos

- [Etapa 1: criar qualquer tipo de AWS IoT coisa e qualquer AWS IoT coisa](#)
- [Etapa 2: criar uma função do IAM a ser assumida por AWS IoT](#)
- [Etapa 3: Criar e configurar o certificado X.509](#)
- [Etapa 4: Teste as AWS IoT credenciais com seu stream de vídeo do Kinesis](#)
- [Etapa 5: implantar AWS IoT certificados e credenciais no sistema de arquivos da sua câmera e transmitir dados para o seu stream de vídeo](#)

Etapa 1: criar qualquer tipo de AWS IoT coisa e qualquer AWS IoT coisa

Em AWS IoT, uma coisa é uma representação de um dispositivo específico ou entidade lógica. Nesse caso, AWS IoT algo representa seu stream de vídeo do Kinesis e você deseja configurar o controle de acesso em nível de recurso. Para criar uma coisa, primeiro você deve criar um tipo de AWS IoT coisa. Você pode usar tipos de AWS IoT coisas para armazenar informações de descrição e configuração comuns a todas as coisas associadas ao mesmo tipo de coisa.

1. O comando de exemplo a seguir cria um tipo de coisa `kvs_example_camera`:

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

2. Esse exemplo de comando cria `kvs_example_camera_stream` uma coisa do tipo `kvs_example_camera` coisa:

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

Etapa 2: criar uma função do IAM a ser assumida por AWS IoT

As funções do IAM são semelhantes às dos usuários, pois uma função é uma AWS identidade com políticas de permissões que determinam o que a identidade pode ou não fazer AWS. Uma função pode ser assumida por qualquer pessoa que precise dela. Quando você assume uma função, ela fornece credenciais de segurança temporárias para sua sessão de função.

A função que você cria nesta etapa pode ser assumida AWS IoT para obter credenciais temporárias do serviço de token de segurança (STS) ao realizar solicitações de autorização de credenciais do cliente. Nesse caso, o cliente é o SDK do Kinesis Video Streams que está sendo executado na sua câmera.

Execute as seguintes etapas para criar e configurar essa função do IAM:

1. Criar um perfil do IAM.

O comando de exemplo a seguir cria uma função do IAM chamada `KVSCameraCertificateBasedIAMRole`:

```
aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
--assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
```

Você pode usar o seguinte JSON de política de confiança para o `iam-policy-document.json`:

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

```

        "Effect": "Allow",
        "Principal": {
            "Service": "credentials.iot.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
}

```

2. Em seguida, anexe uma política de permissões à função do IAM que você criou anteriormente. Essa política de permissões permite o controle de acesso seletivo (um subconjunto de operações suportadas) para um AWS recurso. Nesse caso, o AWS recurso é o stream de vídeo para o qual você deseja que sua câmera envie dados. Em outras palavras, quando todas as etapas de configuração forem concluídas, essa câmera poderá enviar dados somente para esse streaming de vídeo.

```

aws --profile default iam put-role-policy --role-name
KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
document 'file://iam-permission-document.json'

```

Você pode usar o seguinte JSON de política do IAM para o iam-permission-document .json:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:ThingName}/*"
    }
  ]
}

```

Observe que essa política autoriza as ações especificadas somente em um stream de vídeo (AWS recurso) especificado pelo espaço reservado (\$ {credentials-iot:}). ThingName Esse

espaço reservado assume o valor do atributo AWS IoT thing ThingName quando o provedor de AWS IoT credenciais envia o nome do stream de vídeo na solicitação.

3. Em seguida, crie um alias de função para sua função do IAM. O alias da função é um modelo de dados alternativo que aponta para a função do IAM. Uma solicitação de provedor de AWS IoT credenciais deve incluir um alias de função para indicar qual função do IAM assumir para obter as credenciais temporárias do STS.

O exemplo de comando a seguir cria um alias de função chamado 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. Agora você pode criar a política que permitirá assumir AWS IoT a função com o certificado (depois de anexado) usando o alias da função.

O exemplo de comando a seguir cria uma política para AWS IoT calledKvsCameraIoTPolicy.

```
aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-
document 'file://iot-policy-document.json'
```

Você pode usar o comando a seguir para criar o documento JSON do iot-policy-document 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
```

Etapa 3: Criar e configurar o certificado X.509

A comunicação entre um dispositivo (seu stream de vídeo) AWS IoT é protegida pelo uso de certificados X.509.

1. Crie o certificado ao qual você deve anexar a política AWS IoT que você criou 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. Anexe a política para AWS IoT (KvsCameraIoTPolicy criada anteriormente) a esse certificado.

```
aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
$(jq --raw-output '.certificateArn' certificate)
```

3. Anexe sua AWS IoT coisa (kvs_example_camera_stream) ao certificado que você acabou de criar:

```
aws --profile default iot attach-thing-principal --thing-name
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
certificate)
```

4. Para autorizar solicitações por meio do provedor de AWS IoT credenciais, você precisa do endpoint de AWS IoT credenciais, que é exclusivo do seu ID. Conta da AWS Você pode usar o comando a seguir para obter o endpoint de AWS IoT credenciais.

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
--output text > iot-credential-provider.txt
```

5. Além do certificado X.509 criado anteriormente, você também deve ter um certificado CA para estabelecer confiança com o serviço de back-end por meio do TLS. Você pode obter o certificado CA usando o seguinte comando:

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output
cacert.pem
```

Etapa 4: Teste as AWS IoT credenciais com seu stream de vídeo do Kinesis

Agora você pode testar as AWS IoT credenciais que você configurou até agora.

1. Primeiro, crie um streaming de vídeo do Kinesis com o qual você deseja testar essa configuração.


 Important

Crie um stream de vídeo com um nome idêntico ao nome da AWS IoT coisa que você criou na etapa anterior (`kvs_example_camera_stream`).

```
aws kinesismvideo create-stream --data-retention-in-hours 24 --stream-name
kvs_example_camera_stream
```

2. Em seguida, ligue para o provedor de AWS IoT credenciais para obter as credenciais temporárias:

```
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

Você pode usar o comando a seguir para obter o `IOT_GET_CREDENTIAL_ENDPOINT`:

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

O JSON de saída contém o `AccessKey`, o `SecretKey` e o `SessionToken`, que você pode usar para acessar o Kinesis Video Streams.

3. Para seu teste, você pode usar essas credenciais para invocar a API Kinesis Video `DescribeStream` para o stream de vídeo de amostra. `kvs_example_camera_stream`

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesismvideo describe-stream --stream-name kvs_example_camera_stream
```

Etapa 5: implantar AWS IoT certificados e credenciais no sistema de arquivos da sua câmera e transmitir dados para o seu stream de vídeo

Note

As etapas desta seção descrevem o envio de mídia para um stream de vídeo do Kinesis a partir de uma câmera que está usando o [the section called “Biblioteca de produtores C++”](#)

1. Copie o certificado X.509, a chave privada e o certificado CA gerados nas etapas anteriores para o sistema de arquivos da sua câmera. Especifique os caminhos para onde esses arquivos são armazenados, o nome do alias da função e o endpoint de AWS IoT credenciais para executar o `gst-launch-1.0` comando ou seu aplicativo de amostra.
2. O exemplo de comando a seguir usa autorização de AWS IoT certificado para enviar vídeo para o 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 nome do stream

Para representar seu dispositivo (por exemplo, sua câmera) por meio de AWS IoT algo, mas autorizar um nome de stream diferente, você pode usar o AWS IoT `certificateId` atributo como nome do stream e fornecer permissões ao Kinesis Video Streams no stream usando. AWS IoT As etapas para fazer isso são semelhantes às descritas anteriormente, com algumas alterações.

- Modifique a política de permissões para sua função do IAM (`iam-permission-document.json`) da seguinte forma:

```
{
  "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

O ARN do recurso usa o ID do certificado como espaço reservado para o nome de streaming. A permissão do IAM funcionará quando você usar o ID do certificado como nome do stream. Obtenha o ID do certificado para que você possa usá-lo como nome do fluxo na descrição a seguir da chamada da API de fluxo.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Verifique essa alteração usando o comando `describe-stream` da CLI do Kinesis Video Streams:

```

AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name ${CERTIFICATE_ID}

```

- Passe o `CertificateId` para o provedor de credenciais AWS IoT no aplicativo de [amostra no SDK para C++ do Kinesis Video Streams](#):

```

credential_provider =
make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
    cert_path,
    private_key_path,
    role_alias,
    ca_cert_path,
    certificateId);

```

Note

Observe que você está passando o nome da coisa para o provedor de AWS IoT credenciais. Você pode usar `getenv` para passar o nome da coisa para o aplicativo de demonstração da mesma forma que passa os outros atributos. AWS IoT Use o ID do certificado como o nome de streaming nos parâmetros da linha de comando quando estiver executando o aplicativo de amostra.

Use AWS IoT credenciais para transmitir para um nome de fluxo codificado

Para representar seu dispositivo (por exemplo, sua câmera) por meio de AWS IoT algo, mas autorizar o streaming para um stream de vídeo específico do Amazon Kinesis, forneça permissões do Amazon Kinesis Video Streams no stream usando. AWS IoT O processo é semelhante às seções anteriores, com algumas alterações.

Modifique a política de permissões para sua função do IAM (`iam-permission-document.json`) da seguinte forma:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
    }
  ]
}
```

Copie o certificado X.509, a chave privada e o certificado CA gerados nas etapas anteriores para o sistema de arquivos da sua câmera.

Especifique os caminhos para onde esses arquivos são armazenados, o nome do alias da função, o nome da AWS IoT coisa e o endpoint de AWS IoT credenciais para executar o `gst-launch-1.0` comando ou seu aplicativo de amostra.

O exemplo de comando a seguir usa autorização de AWS IoT certificado para enviar vídeo para o 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"
```

Monitoramento dos Amazon Kinesis Video Streams

O Kinesis Video Streams fornece funcionalidade de monitoramento para seus streams de entrega. Para ter mais informações, consulte [Monitorar](#).

Validação de conformidade para Amazon Kinesis Video Streams


Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.

- [Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

 Note

Nem todos Serviços da AWS são elegíveis para a HIPAA. Para mais informações, consulte a [Referência dos serviços qualificados pela HIPAA](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#) — Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência no Amazon Kinesis Video Streams

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as Zonas de Disponibilidade, é possível projetar e operar aplicações e bancos de dados que executem o failover automaticamente entre as Zonas de Disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Segurança da infraestrutura no Kinesis Video Streams

Como um serviço gerenciado, o Amazon Kinesis Video Streams é protegido AWS pelos procedimentos globais de segurança de rede descritos [no whitepaper Amazon Web Services: Visão geral dos processos de](#) segurança.

Você usa chamadas de API AWS publicadas para acessar o Kinesis Video Streams pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos, como o Java 7 e versões posteriores, oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID de chave de acesso e uma chave de acesso secreta associada a um principal do IAM. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Melhores práticas de segurança para o Kinesis Video Streams

O Amazon Kinesis Video Streams fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Para as melhores práticas de segurança para os dispositivos remotos, consulte [Melhores práticas de segurança para Device Agents](#).

Implemente o acesso de privilégio mínimo

Ao conceder permissões, você decide quem receberá quais permissões para quais recursos do Kinesis Video Streams. Você habilita ações específicas que quer permitir nesses recursos. Portanto, você deve conceder somente as permissões necessárias para executar uma tarefa. A implementação do acesso de privilégio mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

Por exemplo, um produtor que envia dados para o Kinesis Video Streams requer apenas `PutMedia`, `GetStreamingEndpoint` e `DescribeStream`. Não conceda aos aplicativos produtores permissões para todas as ações (*) ou para outras ações, como `GetMedia`.

Para obter mais informações, consulte [O que é privilégio mínimo e por que você precisa disso?](#)

Usar funções do IAM

Os aplicativos de produtores e clientes devem ter credenciais válidas para acessar o Kinesis Video Streams. Você não deve armazenar credenciais da AWS diretamente em uma aplicação de cliente ou em um bucket do Amazon S3. Essas são credenciais de longo prazo que não são trocadas automaticamente e podem ter um impacto significativo nos negócios se forem comprometidas.

Em vez disso, você deve usar uma função do IAM para gerenciar credenciais temporárias para que seus aplicativos de produtor e cliente acessem o Kinesis Video Streams. Ao usar uma função, você não precisa usar credenciais de longo prazo (como nome de usuário e senha ou chaves de acesso) para acessar outros recursos.

Para obter mais informações, consulte os seguintes tópicos no Manual do usuário do IAM:

- [Funções do IAM](#)
- [Cenários comuns para funções: usuários, aplicativos e serviços](#)

Use CloudTrail para monitorar chamadas de API

O Kinesis Video Streams AWS CloudTrail trabalha com, um serviço que fornece um registro das ações realizadas por um usuário, uma função AWS service (Serviço da AWS) ou um no Kinesis Video Streams.

Você pode usar as informações coletadas por CloudTrail para determinar a solicitação que foi feita ao Kinesis Video Streams, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para ter mais informações, consulte [the section called “Registro de chamadas de API do CloudTrail com”](#).

Bibliotecas do Kinesis Video Streams Producer

As bibliotecas do Amazon Kinesis Video Streams Producer são um conjunto de bibliotecas no SDK do Kinesis Video Streams Producer. O cliente usa as bibliotecas e o SDK para criar o aplicativo no dispositivo para conectar-se com segurança ao Kinesis Video Streams e transmitir dados de mídia para visualização no console ou nos aplicativos cliente em tempo real.

O streaming dos dados de mídia pode ser feito das seguintes maneiras:

- Em tempo real
- Depois de guardá-lo por alguns segundos
- Após o upload da mídia

Depois de criar um stream do Kinesis Video Streams, você pode começar a enviar dados para ele. Você pode usar o SDK para criar um código de aplicativo que extrai os dados de vídeo, conhecidos como quadros, da fonte de mídia e os carrega no Kinesis Video Streams. Esses aplicativos também são conhecidos como aplicativos de produtores.

As Bibliotecas de produtores contêm os seguintes componentes:

- [Cliente do Kinesis Video Streams Producer](#)
- [Biblioteca do Kinesis Video Streams Producer](#)

Cliente do Kinesis Video Streams Producer

O Kinesis Video Streams Producer Client inclui um único `KinesisVideoClient` classe. Essa classe gerencia fontes de mídia, recebe dados das fontes e gerencia o ciclo de vida do stream à medida que os dados fluem de uma fonte de mídia para o Kinesis Video Streams. Ele também fornece um `MediaSource` interface para definir a interação entre o Kinesis Video Streams e seu hardware e software proprietários.

Praticamente qualquer coisa pode ser uma fonte de mídia. Por exemplo, você pode usar uma fonte de mídia de câmera ou uma fonte de mídia de microfone. As fontes de mídia não se limitam somente a fontes de áudio e vídeo. Por exemplo, os logs de dados podem ser arquivos de texto, mas eles ainda podem ser enviados como um streaming de dados. Você também pode transmitir simultaneamente dados de várias câmeras no seu telefone.

Para obter dados de qualquer uma dessas fontes, você pode implementar a interface `MediaSource`. Essa interface habilita cenários adicionais para os quais não fornecemos suporte integrado. Por exemplo, você pode optar por enviar o seguinte para o Kinesis Video Streams:

- Um stream de dados de diagnóstico (por exemplo, registros de aplicativos e eventos)
- Dados de câmera com tecnologia de infravermelho, RADARs ou câmeras de profundidade

O Kinesis Video Streams não fornece implementações integradas para dispositivos de produção de mídia, como câmeras. Para extrair dados desses dispositivos, você deve implementar o código, criando assim sua própria implementação de fonte de mídia personalizada. Você pode então registrar explicitamente suas fontes de mídia personalizadas com `KinesisVideoClient`, que carrega os dados para o Kinesis Video Streams.

O Kinesis Video Streams Producer Client está disponível para aplicativos Java e Android. Para ter mais informações, consulte [Usar a Biblioteca de produtores Java](#) e [Utilização da biblioteca de produtores Java](#).

Biblioteca do Kinesis Video Streams Producer

A Kinesis Video Streams Producer Library está contida no Kinesis Video Streams Producer Client. A biblioteca também está disponível para uso direto para aqueles que desejam uma integração mais profunda com o Kinesis Video Streams. Ela permite a integração de dispositivos com sistemas operacionais proprietários, pilhas de rede ou recursos limitados no dispositivo.

A Kinesis Video Streams Producer Library implementa a máquina de estado para streaming para o Kinesis Video Streams. Ele fornece ganchos de retorno de chamada, que exigem que você forneça sua própria implementação de transporte e manipule explicitamente cada mensagem de e para o serviço.

Você pode optar por usar a Kinesis Video Streams Producer Library diretamente pelos seguintes motivos:

- O dispositivo no qual você deseja executar o aplicativo não possui uma máquina virtual Java.
- Você quer escrever o código do aplicativo em linguagens diferentes de Java.
- Você quer reduzir a sobrecarga em seu código e limitá-la ao nível mínimo de abstração, devido a limitações como memória e capacidade de processamento.

Atualmente, a Kinesis Video Streams Producer Library está disponível para aplicativos Android, C, C++ e Java. Para obter mais informações, consulte os idiomas suportados a seguir. **Tópicos relacionados.**

Tópicos relacionados

[Usar a Biblioteca de produtores Java](#)

[Utilização da biblioteca de produtores Java](#)

[Usar a Biblioteca de produtores C++](#)

[Usar a Biblioteca de produtores C](#)

[Uso do C++ Producer SDK no Raspberry Pi](#)

Usar a Biblioteca de produtores Java

Você pode usar a biblioteca Java Producer fornecida pelo Amazon Kinesis Video Streams para escrever código de aplicativo com configuração mínima e enviar dados de mídia de um dispositivo para um stream de vídeo do Kinesis.

Execute as etapas a seguir para integrar seu código ao Kinesis Video Streams para que seu aplicativo possa começar a transmitir dados para seu stream de vídeo do Kinesis:

1. Crie uma instância do objeto `KinesisVideoClient`.
2. Crie um objeto `MediaSource` fornecendo informações de fonte de mídia. Por exemplo, ao criar uma fonte de mídia de câmera, você fornece informações como a identificação da câmera e a especificação da codificação usada pela câmera.

Quando quiser iniciar o streaming, você precisará criar uma fonte de mídia personalizada.

3. Registre a origem da mídia com o `KinesisVideoClient`.

Depois de registrar a fonte de mídia com o `KinesisVideoClient`, sempre que os dados estiverem disponíveis com a fonte de mídia, ele chamará `KinesisVideoClient` com os dados.

Procedimento: usar o Producer SDK em Java

Esse procedimento demonstra como usar o cliente Java Producer do Kinesis Video Streams em seu aplicativo Java para enviar dados para seu stream de vídeo do Kinesis.

Essas etapas não exigem que você tenha uma fonte de mídia, como uma câmera ou um microfone. Em vez disso, para fins de teste, o código gera quadros de amostra que consistem em uma série de bytes. Você pode usar o mesmo padrão de codificação ao enviar dados de mídia de fontes reais, como câmeras e microfones.

Este procedimento inclui as seguintes etapas:

- [Fazer download e configurar o código](#)
- [Escrever e examinar o código](#)
- [Executar e verificar o código](#)

Pré-requisitos

- No código de exemplo, você fornece credenciais especificando um perfil que você configurou em seu arquivo de perfil de AWS credenciais. Primeiro, configure seu perfil de credenciais (se ainda não tiver feito isso). Para obter mais informações, consulte [Configurar AWS credenciais e região para desenvolvimento](#) no AWS SDK for Java.

Note

O exemplo Java usa um `SystemPropertiesCredentialsProvider` objeto para obter suas credenciais. O provedor recupera essas credenciais das propriedades de sistema Java `aws.accessKeyId` e `aws.secretKey`. Você define essas propriedades de sistema no seu ambiente de desenvolvimento Java. Para obter informações sobre como configurar as propriedades de sistema Java, consulte a documentação do seu ambiente de desenvolvimento integrado (IDE) específico.

- Você `NativeLibraryPath` deve conter seu `KinesisVideoProducerJNI` arquivo, disponível em <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>. A extensão do nome do arquivo para esse arquivo depende do seu sistema operacional:
 - `KinesisVideoProducerJNI.so` para Linux
 - `KinesisVideoProducerJNI.dylib` para macOS

- KinesisVideoProducerJNI.dll para Windows

Note

Bibliotecas pré-criadas para macOS, Ubuntu, Windows e Raspbian estão disponíveis `src/main/resources/lib` [em amazon-kinesis-video-streams](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java) [https://github.com/aws-labs/- .git. producer-sdk-java](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java) Para outros ambientes, compile o [Biblioteca de produtores C++](#).

Etapa 1: Baixe e configure o código da Java Producer Library

Nesta seção do procedimento da Biblioteca de produtores Java, você faz download do código de exemplo Java, importa o projeto no Java IDE e configura os locais da biblioteca.

Para obter os pré-requisitos e outros detalhes desse exemplo, consulte [Utilização da biblioteca de produtores do Java](#).

1. Crie um diretório e, em seguida, clone o código-fonte de exemplo do GitHub repositório.

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

2. Abra o ambiente de desenvolvimento integrado (IDE) Java que você usa (por exemplo, [Eclipse](#) ou [JetBrains IntelliJ IDEA](#)) e importe o projeto Apache Maven que você baixou:
 - In IntelliJ IDEA: Escolha Import. Navegue até o arquivo `pom.xml` na raiz do pacote que você obteve por download.
 - No Eclipse: escolha File, Import, Maven e Existing Maven Projects. Em seguida, navegue até o diretório `kinesis-video-java-demo`.

Para obter mais informações, consulte a documentação do seu IDE.

3. O código de exemplo Java usa as AWS credenciais atuais. Para usar um perfil de credenciais diferente, localize o seguinte código em `DemoAppMain.java`:

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

Altere o código para o seguinte:

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

Para obter mais informações, consulte [ProfileCredentialsProvider](#) na AWS SDK for Java referência.

Próxima etapa

[the section called “Etapa 2: escrever e examinar o código”](#)

Etapa 2: escrever e examinar o código

Nesta seção do [procedimento da Java Producer Library](#), você escreve e examina o código de exemplo Java que você baixou na seção anterior.

O aplicativo de teste Java ([DemoAppMain](#)) mostra o padrão de codificação a seguir:

- Crie uma instância de `KinesisVideoClient`.
- Crie uma instância de `MediaSource`.
- Registre o `MediaSource` com o cliente.
- Inicie o streaming. `MediaSource` inicie o e ele começa a enviar dados para o cliente.

As seguintes seções fornecem detalhes.

Criação de uma instância do `KinesisVideoClient`

Você cria um objeto `KinesisVideoClient` chamando a operação `createKinesisVideoClient`.

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

Para `KinesisVideoClient` fazer chamadas de rede, ele precisa de credenciais para autenticar. Você passa em uma instância de `SystemPropertiesCredentialsProvider`, que lê `AWSCredentials` para o perfil padrão no arquivo de credenciais:

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

Criação de uma instância do `MediaSource`

Para enviar bytes para seu stream de vídeo do Kinesis, você deve produzir os dados. O Amazon Kinesis Video Streams `MediaSource` fornece a interface, que representa a fonte de dados.

Por exemplo, a biblioteca Java do Kinesis Video Streams `ImageFileMediaSource` fornece a implementação `MediaSource` da interface. Essa classe só lê dados de uma série de arquivos de mídia em vez de um stream de vídeo do Kinesis, mas você pode usá-la para testar o código.

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

Registrando o `MediaSource` com o cliente

Registre a fonte de mídia que você criou com o `KinesisVideoClient` para que ele identifique o cliente (e, em seguida, envie os dados para ele).

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

Iniciando a fonte de mídia

Inicie a fonte de mídia para que ela possa começar a gerar dados e enviá-los ao cliente.

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

Próxima etapa

[the section called “Etapa 3: executar e verificar o código”](#)

Etapa 3: executar e verificar o código

Para executar o equipamento de teste Java para a [Java Producer Library](#), faça o seguinte.

1. Escolha DemoAppMain.
2. Escolha Executar, Executar 'DemoAppMain'.
3. Adicione suas credenciais aos argumentos de JVM do aplicativo:
 - Para AWS credenciais não temporárias: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
 - Para AWS credenciais temporárias: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"
4. Faça login AWS Management Console e abra o console do [Kinesis Video Streams](#).

Na página Manage Streams, selecione o streaming.

5. O vídeo de exemplo será reproduzido no player incorporado. Provavelmente será necessário esperar um tempo curto (até dez segundos em condições típicas de processador e de largura de banda) enquanto os quadros são acumulados antes do vídeo ser exibido.

O exemplo de código cria um streaming. Conforme `MediaSource` no código é iniciado, ele começa a enviar quadros de amostra para o `KinesisVideoClient`. Em seguida, o cliente envia os dados para o seu stream de vídeo do Kinesis.

Utilização da biblioteca de produtores Java

Você pode usar a Android Producer Library fornecida pelo Amazon Kinesis Video Streams para escrever código de aplicativo, com configuração mínima, para enviar dados de mídia de um dispositivo Android para um stream de vídeo do Kinesis.

Execute as etapas a seguir para integrar seu código ao Kinesis Video Streams para que seu aplicativo possa começar a transmitir dados para seu stream de vídeo do Kinesis:

1. Crie uma instância do objeto `KinesisVideoClient`.

2. Crie um objeto `MediaSource` fornecendo informações de fonte de mídia. Por exemplo, ao criar uma fonte de mídia de câmera, você fornece informações como a identificação da câmera e a especificação da codificação usada pela câmera.

Quando quiser iniciar o streaming, você precisará criar uma fonte de mídia personalizada.

Procedimento: Usar o SDK de produtores do Android

Este procedimento demonstra como usar o cliente Android Producer do Kinesis Video Streams em seu aplicativo Android para enviar dados para seu stream de vídeo do Kinesis.

Este procedimento inclui as seguintes etapas:

- [the section called “Pré-requisitos”](#)
- [the section called “Etapa 1: Fazer download e configurar o código ”](#)
- [the section called “Etapa 2: Examine o código”](#)
- [the section called “Etapa 3: executar e verificar o código”](#)

Pré-requisitos

- Recomendamos o [Android Studio](#) para examinar, editar e executar o código do aplicativo. Recomendamos usar a versão estável mais recente.
- No código de amostra, você fornece as credenciais do Amazon Cognito.

Siga esses procedimentos para configurar um grupo de usuários e um grupo de identidades do Amazon Cognito.

- [Configurar um grupo de usuários](#)
- [Configurar um pool de identidades](#)

Configurar um grupo de usuários

Para configurar um grupo de usuários

1. Faça login no [console do Amazon Cognito](#) e verifique se a região está correta.
2. Na navegação à esquerda, escolha Grupos de usuários.

3. Na seção Grupos de usuários, escolha Criar grupo de usuários.
4. Complete as seguintes seções:
 - a. Etapa 1: Configurar a experiência de login - Na seção Opções de login do grupo de usuários do Cognito, selecione as opções apropriadas.

Escolha Próximo.
 - b. Etapa 2: Configurar os requisitos de segurança - Selecione as opções apropriadas.

Escolha Próximo.
 - c. Etapa 3: Configurar a experiência de inscrição - Selecione as opções apropriadas.

Escolha Próximo.
 - d. Etapa 4: Configurar a entrega de mensagens - Selecione as opções apropriadas.

No campo de seleção de função do IAM, selecione uma função existente ou crie uma nova função.

Escolha Próximo.
 - e. Etapa 5: integre seu aplicativo - Selecione as opções apropriadas.

No campo Cliente inicial do aplicativo, escolha Cliente confidencial.

Escolha Próximo.
 - f. Etapa 6: Revisar e criar - Revise suas seleções das seções anteriores e escolha Criar grupo de usuários.
5. Na página Grupos de usuários, selecione o pool que você acabou de criar.

Copie o ID do grupo de usuários e anote isso para mais tarde. No `awsconfiguration.json` arquivo, isso é `CognitoUserPool.Default.PoolId`.
6. Selecione a guia Integração de aplicativos e vá até a parte inferior da página.
7. Na seção Lista de clientes do aplicativo, escolha o nome do cliente do aplicativo que você acabou de criar.

Copie o ID do cliente e anote isso para mais tarde. No `awsconfiguration.json` arquivo, isso é `CognitoUserPool.Default.AppClientId`.
8. Mostre o segredo do cliente e anote isso para mais tarde. No `awsconfiguration.json` arquivo, isso é `CognitoUserPool.Default.AppClientSecret`.

Configurar um pool de identidades

Para configurar um grupo de identidades

1. Faça login no [console do Amazon Cognito](#) e verifique se a região está correta.
2. Na navegação à esquerda, escolha Pools de identidade.
3. Selecione Criar banco de identidades.
4. Configure o banco de identidades.
 - a. Etapa 1: Configurar a confiança do grupo de identidades - Conclua as seguintes seções:
 - Acesso do usuário - Selecione Acesso autenticado
 - Fontes de identidade autenticadas - Selecione o grupo de usuários do Amazon Cognito

Escolha Próximo.

- b. Etapa 2: Configurar permissões - Na seção Função autenticada, preencha os seguintes campos:
 - Função do IAM - selecione Criar uma nova função do IAM
 - Nome da função do IAM - insira um nome e anote-o para uma etapa posterior.

Escolha Próximo.

- c. Etapa 3: Conectar provedores de identidade - Na seção Detalhes do grupo de usuários, preencha os seguintes campos:
 - ID do grupo de usuários - Selecione o grupo de usuários que você criou anteriormente.
 - ID do cliente do aplicativo - Selecione o ID do cliente do aplicativo que você criou anteriormente.

Escolha Próximo.

- d. Etapa 4: Configurar propriedades - Digite um nome no campo Nome do pool de identidades.


Escolha Próximo.

- e. Etapa 5: Revise e crie - revise suas seleções em cada uma das seções e selecione Criar grupo de identidades.

5. Na página Conjuntos de identidades, selecione seu novo grupo de identidades.

Copie o ID do grupo de identidades e anote isso para mais tarde. No `awsconfiguration.json` arquivo, isso é `CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Atualize as permissões para a função do IAM.
 - a. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
 - b. Na navegação à esquerda, escolha Funções.
 - c. Encontre e selecione a função que você criou acima.

 Note

Use a barra de pesquisa, se necessário.

- d. Selecione a política de permissões anexada.

Selecione Edit (Editar).

- e. Selecione a guia JSON e substitua a política pela seguinte:

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

Escolha Próximo.

- f. Selecione a caixa ao lado de Definir esta nova versão como padrão, se ela ainda não estiver selecionada.

Selecione Save Changes (Salvar alterações).

Etapa 1: Baixe e configure o código da Android Producer Library

Nesta seção do procedimento da Biblioteca de produtores do Android, faça o download do código de exemplo do Android e abra o projeto no Android Studio.

Para obter os pré-requisitos e outros detalhes deste exemplo, consulte [Utilização da Biblioteca de produtores do Android](#).

1. Crie um diretório e, em seguida, clone o AWS Mobile SDK for Android do GitHub repositório.

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

2. Abra o [Android Studio](#).
3. Na tela de abertura, escolha Open an existing Android Studio project.
4. Navegue até o diretório `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` e escolha OK.
5. Abra o arquivo `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json`.

No `CredentialsProvider` nó, forneça o ID do grupo de identidades do procedimento Para configurar um grupo de identidades na seção [Pré-requisitos](#) e forneça seu Região da AWS (por exemplo,). **us-west-2**

No `CognitoUserPool` nó, forneça o segredo do cliente do aplicativo, o ID do cliente do aplicativo e o ID do pool no procedimento Para configurar um grupo de usuários na seção [Pré-requisitos](#) e forneça seu Região da AWS (por exemplo,). **us-west-2**

6. Seu arquivo `awsconfiguration.json` será semelhante ao seguinte:

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
```

```
        "PoolId": "us-west-2:01234567-89ab-cdef-0123-456789abcdef",
        "Region": "us-west-2"
    }
},
"IdentityManager": {
    "Default": {}
},
"CognitoUserPool": {
    "Default": {
        "AppClientSecret": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnopqrstuvwxyz",
        "AppClientId": "0123456789abcdefghijklmnopqrstuvwxyz",
        "PoolId": "us-west-2_qRsTuVwXy",
        "Region": "us-west-2"
    }
}
}
```

7. Atualize o `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` com sua região (no exemplo a seguir, está definido como `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 obter informações sobre Região da AWS constantes, consulte [Regiões](#).

Próxima etapa

[the section called “Etapa 2: Examine o código”](#)

Etapa 2: Examine o código

Nesta seção do [Procedimento da Biblioteca de produtores do Android](#), você examina o código de exemplo.

O aplicativo de teste do Android (`AmazonKinesisVideoDemoApp`) mostra o padrão de codificação a seguir:

- Crie uma instância de `KinesisVideoClient`.
- Crie uma instância de `MediaSource`.
- Inicie o streaming. Inicie o `MediaSource`, e ele começa a enviar dados para o cliente.

As seguintes seções fornecem detalhes.

Criação de uma instância do `KinesisVideoClient`

Você cria um objeto `KinesisVideoClient` chamando a operação [createKinesisVideoClient](#).

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

Para `KinesisVideoClient` fazer chamadas de rede, ele precisa de credenciais para autenticar. Você passa uma instância de `AWSCredentialsProvider`, que lê suas credenciais do Amazon Cognito do `awsconfiguration.json` arquivo que você modificou na seção anterior.

Criação de uma instância do `MediaSource`

Para enviar bytes para seu stream de vídeo do Kinesis, você deve produzir os dados. O Amazon Kinesis Video Streams [MediaSource](#) fornece a interface, que representa a fonte de dados.

Por exemplo, a biblioteca Android do Kinesis Video Streams [AndroidCameraMediaSource](#) fornece a implementação `MediaSource` da interface. Essa classe lê dados de uma das câmeras do dispositivo.

No exemplo de código a seguir (do arquivo [fragment/StreamConfigurationFragment.java](#)), é criada a configuração para a origem da mídia:

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

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

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

```

        .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
            .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
            .withIsEncoderHardwareAccelerated(
                mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
            .withFrameRate(FRAMERATE_20)
            .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
            .withEncodingBitRate(BITRATE_384_KBPS)
            .withCameraOrientation(-
                mCamerasDropdown.getSelectedItem().getCameraOrientation())
            .withNalAdaptationFlags(StreamInfo.NalAdaptationFlags.NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS)
            .withIsAbsoluteTimecode(false));
    }

```

No exemplo de código a seguir (do arquivo [fragment/StreamingFragment.java](#)), é criada a origem da mídia:

```

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

```

Iniciando a fonte de mídia

Inicie a fonte de mídia para que ele possa começar a gerar dados e enviá-los ao cliente. O exemplo de código a seguir é do arquivo [fragment/StreamingFragment.java](#):

```

mCameraMediaSource.start();

```

Próxima etapa

[the section called “Etapa 3: executar e verificar o código”](#)

Etapa 3: executar e verificar o código

Para executar o exemplo de aplicativo do Android para a [Biblioteca de produtores do Android](#), faça o seguinte:

1. Conecte a um dispositivo Android.
2. Escolha Run, Run... e selecione Edit configurations....

- Escolha o ícone de adição (+), aplicativo Android. No campo Name (Nome), insira **AmazonKinesisVideoDemoApp**. No menu suspenso Módulo, escolha AmazonKinesisVideoDemoApp Escolha OK.
- Selecione Run, Run.
- Na tela Select a Deployment Target, escolha seu dispositivo conectado e selecione OK.
- No AWSKinesisVideoDemoApp aplicativo no dispositivo, escolha Criar nova conta.
- Insira valores para USERNAME, Password, Given name, Email address e Phone number, e a seguir escolha Sign up.

 Note


Esses valores têm as seguintes restrições:

- Password: deve conter letras maiúsculas e minúsculas, números e caracteres especiais. Você pode alterar essas restrições na sua página de grupo de usuários no console do [Amazon Cognito](#).
- Email address: deve ser um endereço válido para que você possa receber um código de confirmação.
- Phone number: deve estar no seguinte formato: **+<Country code><Number>**, por exemplo, **+12065551212**.

- Insira o código que você recebe por e-mail e escolha Confirmar. Escolha OK.
- Na próxima página, mantenha os valores padrão e escolha Stream.
- Faça login AWS Management Console e abra o console do [Kinesis Video](#) Streams na região Oeste dos EUA (Oregon).

Na página Manage Streams, selecione demo-stream.

- O vídeo de streaming é reproduzido no player incorporado. Provavelmente será necessário esperar um tempo curto (até dez segundos em condições típicas de processador e de largura de banda) enquanto os quadros são acumulados antes do vídeo ser exibido.

 Note

Se a tela do dispositivo girar (por exemplo, de horizontal para vertical), o aplicativo interromperá o streaming de vídeo.

O exemplo de código cria um streaming. Conforme `MediaSource` no código é iniciado, ele começa a enviar quadros da câmera para o `KinesisVideoClient`. Em seguida, o cliente envia os dados para um stream de vídeo do Kinesis chamado `demo-stream`.

Usar a Biblioteca de produtores C++

Você pode usar a biblioteca C++ `Producer` fornecida pelo Amazon Kinesis Video Streams para escrever código de aplicativo para enviar dados de mídia de um dispositivo para um stream de vídeo do Kinesis.

Modelo de objeto

A biblioteca C++ fornece os seguintes objetos para gerenciar o envio de dados para um stream de vídeo do Kinesis:

- `KinesisVideoProducer`: contém informações sobre sua fonte de mídia e AWS credenciais e mantém retornos de chamada para relatar eventos do Kinesis Video Streams.
- `KinesisVideoStream`: representa o stream de vídeo do Kinesis. Contém informações sobre os parâmetros do stream de vídeo, como nome, período de retenção de dados e tipo de conteúdo de mídia.

Colocando mídia no stream

Você pode usar os métodos fornecidos pela biblioteca C++ (por exemplo, `PutFrame`) para colocar dados no `KinesisVideoStream` objeto. Em seguida, ela gerencia o estado interno dos dados, o que pode incluir as seguintes tarefas:

- Realizar autenticação.
- Verificar se há latência de rede. Se a latência for muito alta, a biblioteca poderá reduzir quadros.
- Controlar status do streaming em andamento.

Interfaces de retorno de chamada

Esta camada expõe um conjunto de interfaces de retorno de chamada, que permitem a comunicação com a camada do aplicativo. Essas interfaces de retorno de chamada incluem o seguinte:

- Interface de retorno de chamada de serviço (`CallbackProvider`): a biblioteca invoca eventos obtidos por meio dessa interface quando cria um fluxo, obtém uma descrição do fluxo e exclui um fluxo.
- Interface de eventos de armazenamento baixo ou estado de cliente pronto (`ClientCallbackProvider`): a biblioteca invoca eventos nessa interface quando o cliente está pronto ou quando detecta o risco de ficar sem armazenamento ou com pouca memória disponível.
- Interface de retorno de chamada de eventos do streaming (`StreamCallbackProvider`): a biblioteca invoca eventos nessa interface quando ocorrem eventos de streaming, como streaming que entra no estado pronto, quadros perdidos ou erros de streaming.

O Kinesis Video Streams fornece implementações padrão para essas interfaces. Você também pode fornecer sua própria implementação personalizada, por exemplo, se precisar de uma lógica de rede personalizada ou quiser expor uma condição de armazenamento insuficiente à interface do usuário.

Para obter mais informações sobre retornos de chamadas nas Bibliotecas de produtores, consulte [Retornos de chamada do SDK do produtor](#).

Procedimento: usar o Producer SDK em C++

Esse procedimento demonstra como usar o cliente e as fontes de mídia do Kinesis Video Streams em um aplicativo C++ para enviar dados para seu stream de vídeo do Kinesis.

Este procedimento inclui as seguintes etapas:

- [Etapa 1: Fazer download e configurar o código](#)
- [Etapa 2: Escrever e examinar o código](#)
- [Etapa 3: Executar e verificar o código](#)

Pré-requisitos

- Credenciais: no código de exemplo, você fornece credenciais especificando um perfil que você configurou em seu arquivo de perfil de AWS credenciais. Primeiro, configure seu perfil de credenciais (se ainda não tiver feito isso).

Para obter mais informações, consulte [Configurar AWS credenciais e região para desenvolvimento](#).

- Integração do armazenamento de certificados: a Biblioteca de Produtores do Kinesis Video Streams deve estabelecer confiança com o serviço que ela chama. Isso é feito por meio da validação das autoridades de certificação (CAs) no repositório público de certificados. Em modelos baseados em Linux, esse armazenamento está localizado no diretório `/etc/ssl/`.

Faça download do certificado do seguinte local para o armazenamento de certificados:

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

- Instale as seguintes dependências de compilação para o macOS:
 - [Autoconf 2.69](#) (Licença GPLv3+/ Autoconf: GNU GPL versão 3 ou mais recente)
 - [CMake 3.7 ou 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select versão 2347)
 - Java Development Kit (JDK) (para compilação de JNI do Java)
 - [Lib-Pkg](#)
- Instale as seguintes dependências de compilação para o Ubuntu:
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`
 - configuração do pacote: `sudo apt install pkg-config`
 - OpenJDK: `sudo apt install openjdk-8-jdk`

Note

Isso só é necessário se você estiver criando a Java Native Interface (JNI).

- Defina a `JAVA_HOME` variável de ambiente: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Próxima etapa

[Etapa 1: Fazer download e configurar o código da Biblioteca de produtores C++](#)

Etapa 1: Baixe e configure o código da Biblioteca do Produtor de C++

Para obter informações sobre como baixar e configurar a C++ Producer Library, consulte [Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin e JNI](#).

Para obter os pré-requisitos e obter mais informações sobre esse exemplo, consulte [Usando a biblioteca de produtores de C++](#).

Próxima etapa

[Etapa 2: escrever e examinar o código](#)

Etapa 2: escrever e examinar o código

Nesta seção do [Procedimento da Biblioteca de produtores C++](#), você examinará o código na bateria de teste de C++ (`tst/ProducerTestFixture.h` e outros arquivos). Você fez download desse código na seção anterior.

O exemplo de C++ Platform Independent mostra o seguinte padrão de codificação:

- Crie uma instância de `KinesisVideoProducer` para acessar o Kinesis Video Streams.
- Crie uma instância de `KinesisVideoStream`. Isso cria um stream de vídeo do Kinesis em seu Conta da AWS caso ainda não exista um stream com o mesmo nome.
- Chame `putFrame` em `KinesisVideoStream` para cada quadro de dados (à medida que ele se tornar disponível) para enviá-lo ao streaming.

As seções a seguir fornecem mais informações sobre esse padrão de codificação.

Criação de uma instância do `KinesisVideoProducer`

Você cria um objeto `KinesisVideoProducer` chamando o método `KinesisVideoProducer::createSync`. O exemplo a seguir cria `KinesisVideoProducer` no arquivo `ProducerTestFixture.h`:

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

```
move(credential_provider_,
     defaultRegion_);
```

O método `createSync` usa os seguintes parâmetros:

- Um objeto `DeviceInfoProvider`, que retorna um objeto `DeviceInfo` contendo informações sobre o dispositivo ou a configuração de armazenamento.

Note

Você deve configurar o tamanho do armazenamento de conteúdo usando o parâmetro `deviceInfo.storageInfo.storageSize`. Seus streamings de conteúdo compartilham o armazenamento de conteúdo. Para determinar o requisito de tamanho do storage, multiplique o tamanho médio do quadro pelo número de quadros armazenados para a duração máxima de todos os streamings. Em seguida, multiplique por 1,2 para considerar a desfragmentação. Por exemplo, suponha que o aplicativo tem a seguinte configuração:

- Três streamings
- 3 minutos de duração máxima
- Cada streaming é de 30 quadros por segundo (FPS)
- Cada quadro tem 10.000 KB de tamanho

O requisito de armazenamento de conteúdo para esse aplicativo é 3 (streams) * 3 (minutos) * 60 (segundos em um minuto) * 10000 (kb) * 1,2 (permissão de desfragmentação) = 194,4 Mb ~ 200 Mb.

- Um objeto `ClientCallbackProvider`, que retorna ponteiros de função que relatam eventos específicos do cliente.
- Um objeto `StreamCallbackProvider`, que retorna ponteiros de função que são chamados de volta quando ocorrem eventos específicos de streaming.
- Um `CredentialProvider` objeto que fornece acesso às variáveis de ambiente de AWS credenciais.
- O Região da AWS ("us-west-2"). O endpoint de serviço é determinado a partir da região.

Criação de uma instância do `KinesisVideoStream`

Você cria um objeto `KinesisVideoStream` chamando o método `KinesisVideoProducer::CreateStream` com um parâmetro `StreamDefinition`. O exemplo

cria o `KinesisVideoStream` no arquivo `ProducerTestFixture.h` com o tipo de trilha como vídeo e com o ID da trilha 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));
```

O objeto `StreamDefinition` tem os seguintes campos:

- Nome do stream.
- Período de retenção de dados.
- Tags do streaming. Essas tags podem ser usadas por aplicativos de consumo para encontrar o streaming correto ou obter mais informações sobre o streaming. As tags também podem ser visualizadas no AWS Management Console.
- AWS KMS chave de criptografia para o stream. Para obter mais informações, consulte Como [usar a criptografia do lado do servidor com o Kinesis Video Streams](#).
- Tipo de streaming. Atualmente, o único valor válido é `STREAMING_TYPE_REALTIME`.
- Tipo de conteúdo de mídia.
- Latência de mídia. Esse valor não é usado atualmente e deve ser definido como 0.
- Duração da reprodução de cada fragmento.
- Escala de código de hora da mídia.
- Se a mídia usa ou não fragmentação de keyframe.
- Se a mídia usa ou não códigos de hora.
- Se a mídia usa ou não horas absolutas de fragmento.

Adicionar uma faixa de áudio ao stream de vídeo do Kinesis

Você pode adicionar detalhes da faixa de áudio a uma definição de fluxo de faixa de vídeo usando o método `addTrack` do `StreamDefinition`:

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

O método `addTrack` requer os seguintes parâmetros:

- ID da faixa (como uma para áudio). Esse valor deve ser exclusivo e diferente de zero.
- Nome da faixa definido pelo usuário (por exemplo, “áudio” para a faixa de áudio).
- ID do codec para essa faixa (por exemplo, para a faixa de áudio “A_AAC”).
- Tipo de faixa (por exemplo, use o valor enum de `MKV_TRACK_INFO_TYPE_AUDIO` para áudio).

Se tiver dados privados do codec para a trilha de áudio, você poderá transmiti-los ao chamar a função `addTrack`. Você também pode enviar os dados privados do codec depois de criar o `KinesisVideoStream` objeto enquanto chama o método `start` in `KinesisVideoStream`.

Colocando um quadro no stream de vídeo do Kinesis

Você insere mídia no stream de vídeo do Kinesis usando `KinesisVideoStream::putFrame`, passando um `Frame` objeto que contém o cabeçalho e os dados de mídia. O exemplo chama `putFrame` no arquivo `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

O exemplo anterior do C++ Producer envia um buffer de dados de teste. Em uma situação real, você precisa obter o buffer e o tamanho do quadro dos dados do quadro em uma fonte de mídia (como uma câmera).

O objeto `Frame` tem os seguintes campos:

- Índice de quadros. Este deve ser um valor incremental monotônico.
- Sinalizadores associados ao quadro. Por exemplo, se o codificador fosse configurado para produzir um keyframe, esse quadro receberia o sinalizador `FRAME_FLAG_KEY_FRAME`.
- Carimbo de data/hora de decodificação.
- Data e hora da apresentação.
- Duração do quadro (para unidades de 100 ns).
- Tamanho do quadro em bytes.
- Dados de quadro.

Para obter mais informações sobre o formato do quadro, consulte Modelo de dados do [Kinesis Video Streams](#).

Colocando um `KinesisVideoFrame` em uma faixa específica de `KinesisVideoStream`

Você pode usar a `PutFrameHelper` classe para colocar dados do quadro em uma faixa específica. Primeiro, chame o `getFrameData Buffer` para obter um ponteiro para um dos buffers pré-alocados para preencher os dados. `KinesisVideoFrame` Em seguida, você pode chamar o `putFrameMulti Track` para enviar o valor booleano `KinesisVideoFrame` junto com o valor booleano para indicar o tipo de dados do quadro. Use `true` (verdadeiro) se forem dados de vídeo ou `false` (falso) se o quadro tiver dados de áudio. O método `putFrameMulti Track` usa um mecanismo de enfileiramento para garantir que os fragmentos MKV mantenham registros de data e hora de quadros monotonicamente

crecentes e que quaisquer dois fragmentos não se sobreponham. Por exemplo, o timestamp MKV do primeiro quadro de um fragmento deve sempre ser maior do que o timestamp MKV do último quadro do fragmento anterior.

O `PutFrameHelper` tem os seguintes campos:

- Número máximo de quadros de áudio na fila.
- Número máximo de quadros de vídeo na fila.
- Tamanho a ser alocado para um único quadro de áudio.
- Tamanho a ser alocado para um único quadro de vídeo.

Métricas e registro de métricas

O `Producer SDK` em C++ inclui a funcionalidade de métricas e registro em log de métricas.

Você pode usar as operações `getKinesisVideoMetrics` e da `getKinesisVideoStreamMetrics` API para recuperar informações sobre o Kinesis Video Streams e seus streams ativos.

O código a seguir é do arquivo `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);
```

O objeto `PClientMetrics` preenchido por `getKinesisVideoMetrics` contém as seguintes informações:

- `contentStoreSize`: o tamanho geral em bytes do armazenamento de conteúdo (a memória usada para armazenar dados de streaming).
- `contentStoreAvailableTamanho`: a memória disponível no armazenamento de conteúdo, em bytes.
- `contentStoreAllocatedTamanho`: A memória alocada no armazenamento de conteúdo.
- `totalContentViewsTamanho`: a memória total usada para a visualização do conteúdo. A visualização do conteúdo é uma série de índices de informações no armazenamento de conteúdo.
- `totalFrameRate`: o número agregado de quadros por segundo em todos os fluxos ativos.
- `totalTransferRate`: o total de bits por segundo (bps) enviados em todos os fluxos.

O objeto `PStreamMetrics` preenchido por `getKinesisVideoStreamMetrics` contém as seguintes informações:

- `currentViewDuration`: a diferença em unidades de 100 ns entre a cabeça da visualização do conteúdo (quando os quadros são codificados) e a posição atual (quando os dados do quadro são enviados para o Kinesis Video Streams).
- `overallViewDuration`: a diferença em unidades de 100 ns entre a parte superior da visualização do conteúdo (quando os quadros são codificados) e a parte final (quando os quadros são apagados da memória, seja porque o espaço total alocado para a visualização do conteúdo foi excedido ou porque uma `PersistedAck` mensagem é recebida do Kinesis Video Streams e os quadros conhecidos por persistirem são liberados).
- `currentViewSize`: o tamanho em bytes da visualização do conteúdo do cabeçalho (quando os quadros são codificados) até a posição atual (quando os quadros são enviados para o Kinesis Video Streams).
- `overallViewSize`: o tamanho total em bytes da visualização do conteúdo.
- `currentFrameRate`: a última taxa medida do stream, em quadros por segundo.
- `currentTransferRate`: a última taxa medida do fluxo, em bytes por segundo.

Desgaste

Se quiser enviar os bytes restantes em um buffer e aguardar o ACK, você poderá usar o `stopSync`:

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

Ou você pode chamar stop para encerrar o streaming:

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

Após interromper o streaming, você pode liberá-lo chamando a seguinte API:

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

Próxima etapa

[the section called “Etapa 3: executar e verificar o código”](#)

Etapa 3: executar e verificar o código

Para executar e verificar o código do [procedimento da C++ Producer Library](#), consulte as seguintes instruções específicas do sistema operacional:

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

Você pode monitorar o tráfego em seu stream observando as métricas associadas ao seu stream no CloudWatch console da Amazon, como `PutMedia.IncomingBytes`.

Usando o C++ Producer SDK como um plug-in do GStreamer

O [GStreamer](#) é uma estrutura de mídia popular usada por várias câmeras e fontes de vídeo para criar canais de mídia personalizados combinando plug-ins modulares. O plug-in Kinesis Video Streams GStreams simplifica a integração do seu pipeline de mídia existente do GStreamer com o Kinesis Video Streams.

Para obter informações sobre como usar o C++ Producer SDK como um plug-in do GStreamer, consulte [Exemplo: Plugin GStreamer do SDK do Kinesis Video Streams Producer - kvssink](#).

Usando o C++ Producer SDK como um plug-in do GStreamer em um contêiner Docker

O [GStreamer](#) é uma estrutura de mídia popular usada por várias câmeras e fontes de vídeo para criar canais de mídia personalizados combinando plug-ins modulares. O plug-in Kinesis Video Streams GStreams simplifica a integração do seu pipeline de mídia existente do GStreamer com o Kinesis Video Streams.

Além disso, o uso do [Docker](#) para criar o pipeline do GStreamer padroniza o ambiente operacional do Kinesis Video Streams, o que simplifica a criação e a execução do aplicativo.

Para obter informações sobre como usar o C++ Producer SDK como um plug-in do GStreamer em um contêiner do Docker, consulte [Execute o elemento GStreamer em um contêiner Docker](#).

Usando a geração de registros com o C++ Producer SDK

Você configura o registro em log para aplicativos do C++ Producer SDK no arquivo `kvs_log_configuration` na pasta `kinesis-video-native-build`.

O exemplo a seguir mostra a primeira linha do arquivo de configuração padrão que configura o aplicativo para gravar entradas em nível de DEBUG no AWS Management Console:

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

Você pode definir o nível de registro em log como INFO para log menos detalhado.

Para configurar o aplicativo para gravar entradas de registro em um arquivo de log, atualize a primeira linha do arquivo para o seguinte:

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

Isso configura o aplicativo para gravar entradas de log no `kvs.log` na pasta `kinesis-video-native-build/log`.

Para alterar a localização do arquivo de log, atualize a linha a seguir com o novo caminho:

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

Note

Se o registro em log for gravado em nível de DEBUG, o arquivo de log poderá usar o espaço de armazenamento disponível no dispositivo rapidamente.

Usar a Biblioteca de produtores C

Você pode usar a biblioteca C Producer fornecida pelo Amazon Kinesis Video Streams para escrever código de aplicativo para enviar dados de mídia de um dispositivo para um stream de vídeo do Kinesis.

Modelo de objeto

[A Biblioteca de Produtores C do Kinesis Video Streams é baseada em um componente comum chamado Platform Independent Codebase \(PIC\), que está GitHub disponível em <https://github.com/aws-labs/-pic>. \[amazon-kinesis-video-streams\]\(https://github.com/aws-labs/-pic\)](https://github.com/aws-labs/-pic) O PIC contém lógica de negócios independente de plataforma para os componentes fundamentais. A Kinesis Video Streams C Producer Library combina o PIC com uma camada adicional de API que permite retornos de chamada e eventos específicos do cenário e da plataforma. A Biblioteca de Produtores C do Kinesis Video Streams tem os seguintes componentes criados com base no PIC:

- Provedores de informações do dispositivo — expõe a `DeviceInfo` estrutura que pode ser fornecida diretamente à API PIC. Você pode configurar um conjunto de provedores, incluindo um provedor otimizado para cenários de aplicativos que pode otimizar o armazenamento de conteúdo com base no número e nos tipos de fluxos que seu aplicativo manipula e na quantidade de buffer necessária configurada com base na quantidade de RAM disponível.
- Provedor de informações de streaming — expõe a `StreamInfo` estrutura que pode ser fornecida diretamente à API PIC. Há um conjunto de provedores que são específicos para os tipos de aplicativos e os tipos comuns de cenários de streaming. Isso inclui provedores como vídeo, áudio e multifaixas de áudio e vídeo. Cada um desses cenários tem padrões que você pode personalizar de acordo com os requisitos do seu aplicativo.
- Provedor de retorno de chamada — expõe a `ClientCallbacks` estrutura que pode ser fornecida diretamente à API PIC. Isso inclui um conjunto de provedores de retorno de chamada para rede (retornos de chamada de API baseados em CURL), autorização (API de AWS credenciais) e repetição de streaming em retornos de chamada com erros. A API do Callback Provider usa vários argumentos para ser configurada, como as informações Região da AWS de autorização. Isso é

feito usando certificados de IoT ou usando AWS AccessKeyId, SecretKey, ou. SessionToken Você pode aprimorar o provedor de retorno de chamada com retornos de chamada personalizados se o seu aplicativo precisar de processamento adicional de um determinado retorno de chamada para obter alguma lógica específica do aplicativo.

- **FrameOrderCoordinator**— Ajuda a gerenciar a sincronização de áudio e vídeo para cenários com várias faixas. Ele tem um comportamento padrão, que você pode personalizar para lidar com a lógica específica do seu aplicativo. Ele também simplifica o empacotamento de metadados do quadro na estrutura do PIC Frame antes de enviá-lo para a API PIC de camada inferior. Para cenários que não são de várias trilhas, esse componente é uma passagem para a API putFrame do PIC.

A biblioteca C fornece os seguintes objetos para gerenciar o envio de dados para um streaming de vídeo do Kinesis:

- **KinesisVideoClient**— Contém informações sobre seu dispositivo e mantém retornos de chamada para relatar eventos do Kinesis Video Streams.
- **KinesisVideoStream**— representa informações sobre os parâmetros do stream de vídeo, como nome, período de retenção de dados e tipo de conteúdo de mídia.

Colocando mídia no stream

Você pode usar os métodos fornecidos pela biblioteca C (por exemplo, PutKinesisVideoFrame) para colocar dados no KinesisVideoStream objeto. Em seguida, ela gerencia o estado interno dos dados, o que pode incluir as seguintes tarefas:

- Realizar autenticação.
- Verificar se há latência de rede. Se a latência for muito alta, a biblioteca poderá reduzir quadros.
- Controlar status do streaming em andamento.

Procedimento: usar o C Producer SDK

Esse procedimento demonstra como usar o cliente e as fontes de mídia do Kinesis Video Streams em um aplicativo C para enviar quadros de vídeo codificados em H.264 para o stream de vídeo do Kinesis.

Este procedimento inclui as seguintes etapas:

- [Etapa 1: Baixe o código da C Producer Library](#)
- [Etapa 2: escrever e examinar o código](#)
- [Etapa 3: executar e verificar o código](#)

Pré-requisitos

- Credenciais — No código de exemplo, você fornece credenciais especificando um perfil que você configurou em seu arquivo de perfil de AWS credenciais. Primeiro, configure seu perfil de credenciais (se ainda não tiver feito isso).

Para obter mais informações, consulte [Configurar AWS credenciais e região para desenvolvimento](#).

- Integração do armazenamento de certificados — A Biblioteca de Produtores do Kinesis Video Streams deve estabelecer confiança com o serviço que ela chama. Isso é feito por meio da validação das autoridades de certificação (CAs) no repositório público de certificados. Em modelos baseados em Linux, esse armazenamento está localizado no diretório `/etc/ssl/`.

Faça download do certificado do seguinte local para o armazenamento de certificados:

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

- Instale as seguintes dependências de compilação para o macOS:
 - [Autoconf 2.69](#) (Licença GPLv3+/ Autoconf: GNU GPL versão 3 ou mais recente)
 - [CMake 3.7 ou 3.8](#)
 - [Pkg-Config](#)
 - xCode (macOS) / clang / gcc (xcode-select versão 2347)
 - Java Development Kit (JDK) (para compilação de JNI do Java)
 - [Lib-Pkg](#)
- Instale as seguintes dependências de compilação para o Ubuntu:
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`
 - configuração do pacote: `sudo apt install pkg-config`
 - OpenJDK: `sudo apt install openjdk-8-jdk`

- Defina a JAVA_HOME variável de ambiente: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Próxima etapa

[Etapa 1: Baixe o código da C Producer Library](#)

Etapa 1: Baixe o código da C Producer Library

Nesta seção, você fará download das bibliotecas de baixo nível. Para obter os pré-requisitos e outros detalhes desse exemplo, consulte [Utilização da biblioteca de produtores C](#).

1. Crie um diretório e, em seguida, clone o código-fonte de exemplo do GitHub repositório.

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

Note

Se você perder a execução do `git clone` com `--recursive`, execute `git submodule update --init` no diretório `amazon-kinesis-video-streams-producer-c/open-source`. Você também deve instalar `pkg-config`, `CMake` e um ambiente de compilação.

Para obter mais informações, consulte `README.md` em <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git>.

2. Abra o código no ambiente de desenvolvimento integrado (IDE) de sua preferência (por exemplo, [Eclipse](#)).

Próxima etapa

[Etapa 2: escrever e examinar o código](#)

Etapa 2: escrever e examinar o código

Nesta seção, você examina o código do aplicativo de amostra `KvsVideoOnlyStreamingSample.c` na `samples` pasta do repositório <https://github.com/aws-labs/>

[amazon-kinesis-video-streams -producer-c](#) em. GitHub Você fez download desse código na etapa anterior. Este exemplo demonstra como usar a C Producer Library para enviar quadros de vídeo codificados em H.264 dentro da pasta para `samples/h264SampleFrames` seu stream de vídeo do Kinesis.

Este aplicativo de exemplo tem três seções:

- Inicialização e configuração:
 - Como inicializar e configurar o pipeline de mídia específicos da plataforma.
 - Inicializando e configurando `KinesisVideoClient` o pipeline, definindo `KinesisVideoStream` os retornos de chamada, integrando a autenticação específica do cenário, extraíndo e enviando dados privados do codec e colocando o stream no estado `READY`.
- Loop principal:
 - Obter o quadro do pipeline de mídia com os timestamps e os sinalizadores.
 - Enviando o quadro para o `KinesisVideoStream`
- Desgaste:
 - Parando (sincronizando) `KinesisVideoStream`, liberando `KinesisVideoStream`, `KinesisVideoClient` liberando.

Este aplicativo de exemplo executa as seguintes tarefas:

- Chame a API `createDefaultDeviceInfo` para criar o objeto `deviceInfo` que contém informações sobre o dispositivo ou a configuração de armazenamento.

```
// 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;
```

- Chame a API `createRealtimeVideoStreamInfoProvider` para criar o objeto `StreamInfo`.

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

- Chame a `createDefaultCallbacksProviderWithAwsCredentials` API para criar o provedor de retornos de chamada padrão com base em AWS credenciais estáticas.

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

- Chame a `createKinesisVideoClient` API para criar o `KinesisVideoClient` objeto que contém informações sobre o armazenamento do seu dispositivo e mantém retornos de chamada para relatar eventos do Kinesis Video Streams.

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

- Chame a API `createKinesisVideoStreamSync` para criar o objeto `KinesisVideoStream`.

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

- Configure um quadro de amostra e chame a API `PutKinesisVideoFrame` para enviar o quadro para o 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;
}
```

- **Desgaste:**

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

Próxima etapa

[Etapa 3: executar e verificar o código](#)

Etapa 3: executar e verificar o código

Para executar e verificar o código do [Procedimento da Biblioteca de produtores C](#), faça o seguinte:

1. Execute os seguintes comandos para criar um build diretório no [SDK C baixado](#) e inicie a cmake partir dele:

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

Você pode passar as opções a seguir para `cmake ..`

- `-DBUILD_DEPENDENCIES`- se deve construir, dependendo das bibliotecas da fonte.
- `-DBUILD_TEST=TRUE`- criar testes unitários e de integração. Pode ser útil para confirmar o suporte para seu dispositivo.

```
./tst/webrtc_client_test
```

- `-DCODE_COVERAGE`- habilitar relatórios de cobertura.
- `-DCOMPILER_WARNINGS`- habilite todos os avisos do compilador.
- `-DADDRESS_SANITIZER`- construa com AddressSanitizer.
- `-DMEMORY_SANITIZER`- construa com MemorySanitizer.
- `-DTHREAD_SANITIZER`- construa com ThreadSanitizer.
- `-DUNDEFINED_BEHAVIOR_SANITIZER`- construa com UndefinedBehaviorSanitizer.
- `-DALIGNED_MEMORY_MODEL` – compilação apenas para dispositivos de modelo de memória alinhada O padrão é OFF.

2. Navegue até o build diretório que você acabou de criar com a etapa anterior e execute `make` para criar o WebRTC C SDK e as amostras fornecidas.

```
make
```

3. O aplicativo de amostra `kinesis_video_cproducer_video_only_sample` envia quadros de vídeo codificados em h.264 dentro da pasta para o Kinesis `samples/h264SampleFrames` Video Streams. O comando a seguir envia os quadros de vídeo em um loop por dez segundos para o Kinesis Video Streams:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Se você quiser enviar quadros codificados em H.264 de outra pasta (por exemplo, `MyH264FramesFolder`), execute a amostra com os seguintes argumentos:

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

4. Para habilitar os logs detalhados, defina os C-defines `HEAP_DEBUG` e `LOG_STREAMING` removendo os comentários das linhas apropriadas em `CMakeList.txt`.

Você pode monitorar o progresso do pacote de teste na saída de depuração no IDE. Você também pode monitorar o tráfego em seu stream observando as métricas associadas ao seu stream no CloudWatch console da Amazon, como `PutMedia.IncomingBytes`.

Note

O console não exibe os dados como um streaming de vídeo porque o equipamento de teste envia somente quadros de bytes vazios.

Uso do C++ Producer SDK no Raspberry Pi

O Raspberry Pi é um computador pequeno e barato que pode ser usado para ensinar e aprender habilidades básicas de programação de computadores. Este tutorial descreve como você pode configurar e usar o SDK do Amazon Kinesis Video Streams C++ Producer em um dispositivo Raspberry Pi. As etapas também incluem como verificar a instalação por meio do aplicativo de demonstração GStreamer.

Tópicos

- [Pré-requisitos](#)
- [Crie um usuário do IAM com permissão para gravar no Kinesis Video Streams](#)
- [Conecte seu Raspberry Pi à sua rede Wi-Fi](#)
- [Conecte-se remotamente ao seu Raspberry Pi](#)
- [Configurar a câmera Raspberry Pi](#)
- [Pré-requisitos de instalação do software](#)

- [Baixe e crie o SDK do Kinesis Video Streams C++ Producer](#)
- [Transmita vídeo para sua transmissão de vídeo do Kinesis e assista à transmissão ao vivo](#)

Pré-requisitos

Antes de configurar o C++ Producer SDK no Raspberry Pi, verifique se você tem os seguintes pré-requisitos:

- Um dispositivo Raspberry Pi com a seguinte configuração:
 - Versão da placa: 3 Model B ou posterior.
 - Um módulo de câmera conectado.
 - Um cartão SD com uma capacidade de pelo menos 8 GB.
 - O sistema operacional Raspbian (versão do kernel 4.9 ou posterior) instalado. Você pode baixar a imagem mais recente do sistema operacional Raspberry Pi (anteriormente chamado de Raspbian) no site do [Raspberry Pi](#). Siga as instruções do Raspberry Pi para [instalar a imagem obtida por download em um cartão SD](#).
- E Conta da AWS com um stream de vídeo do Kinesis. Para obter mais informações, consulte [Conceitos básicos do Kinesis Video Streams](#).

Note

O C++ Producer SDK usa a região Oeste dos EUA (Oregon) (`us-west-2`) por padrão. Para usar o padrão, Região da AWS crie seu stream de vídeo do Kinesis na região Oeste dos EUA (Oregon).

Para usar uma região diferente para seu stream de vídeo do Kinesis, faça o seguinte:

- Defina a seguinte variável de ambiente para sua região (por exemplo, `us-east-1`):

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

Crie um usuário do IAM com permissão para gravar no Kinesis Video Streams

Se você ainda não tiver feito isso, configure um usuário AWS Identity and Access Management (IAM) com permissões para gravar em um stream de vídeo do Kinesis.

Esses procedimentos têm como objetivo ajudá-lo a começar rapidamente a usar um par de chaves de AWS acesso. Os dispositivos podem usar certificados X.509 para se conectar. AWS IoT Consulte [the section called “Controlando o acesso aos recursos do Kinesis Video Streams usando AWS IoT”](#) para obter mais informações sobre como configurar seu dispositivo para usar a autenticação baseada em certificado.

1. Faça login AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No menu de navegação à esquerda, escolha Users (Usuários).
3. Para criar um usuário, selecione Add user (Adicionar usuário).
4. Forneça um Nome do usuário para o usuário, como **kinesis-video-raspberry-pi-producer**.
5. Em Access type (Tipo de acesso), selecione Programmatic Access (Acesso programático).
6. Selecione Next: Permissions (Próximo: permissões).
7. Em Definir permissões para kinesis-video-raspberry-pi-producer, escolha Anexar políticas existentes diretamente.
8. Escolha Criar política. A página Create policy (Criar política) é aberta em uma nova guia do navegador da web.
9. Selecione a guia JSON.
10. Copie a política JSON a seguir e cole-a na área de texto. Essa política dá permissão ao usuário para criar e gravar dados nos streams de vídeo do Kinesis.

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

11. Escolha Revisar política.
12. Forneça um Nome para sua política, como **kinesis-video-stream-write-policy**.
13. Escolha Criar política.
14. Retorne à guia Add user (Adicionar usuário) no navegador e selecione Refresh (Atualizar).
15. Na caixa de pesquisa, digite o nome da política que você criou.
16. Marque a caixa de seleção ao lado da sua nova política na lista.
17. Escolha Próximo: revisar.
18. Escolha Criar usuário.
19. O console exibe a Access key ID (ID de chave de acesso) para o novo usuário. Selecione Show (Mostrar) para exibir a Secret access key (Chave de acesso secreta). Registre esses valores; eles são necessários quando o aplicativo é configurado.

Conecte seu Raspberry Pi à sua rede Wi-Fi

Você pode usar o Raspberry Pi no modo dedicado, ou seja, sem uma chave, monitor ou cabo de rede anexados. Se você estiver usando um monitor e um teclado anexados, prossiga para [Configurar a câmera Raspberry Pi](#).

1. No computador, crie um arquivo chamado `wpa_supplicant.conf`.
2. Copie o texto a seguir e cole-o no `wpa_supplicant.conf` arquivo:

```
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"
}
```

Substitua os valores `ssid` e `psk` pelas informações da sua rede Wi-Fi.

3. Copie o arquivo `wpa_supplicant.conf` no cartão SD. Ele deve ser copiado na raiz do volume boot.

4. Insira o cartão SD no Raspberry Pi e ligue o dispositivo. Isso conecta a rede Wi-Fi e o SSH é habilitado.

Conecte-se remotamente ao seu Raspberry Pi

É possível se conectar remotamente ao Raspberry Pi no modo dedicado. Se você estiver usando o Raspberry Pi com um monitor ou teclado conectado, prossiga para [Configurar a câmera Raspberry Pi](#).

1. Antes de se conectar remotamente ao dispositivo Raspberry Pi, execute uma das ações a seguir para determinar seu endereço IP:
 - Se tiver acesso ao roteador Wi-fi da rede, verifique os dispositivos Wi-Fi conectados. Localize o dispositivo chamado Raspberry Pi para encontrar o endereço IP do seu dispositivo.
 - Se não tiver acesso ao roteador Wi-fi da rede, você pode usar outro software para localizar os dispositivos nessa rede. O [Fing](#) é um aplicativo popular disponível para dispositivos Android e iOS. Você pode usar a versão gratuita desse aplicativo para encontrar os endereços IP dos dispositivos na sua rede.
2. Quando você sabe o endereço IP do dispositivo Raspberry Pi, é possível usar qualquer aplicativo de terminal para conexão.
 - No macOS ou Linux, use ssh:

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

- No Windows, use [PuTTY](#), um cliente SSH gratuito para Windows.

Para obter uma nova instalação do Raspbian, o nome de usuário é **pi** e a senha é **raspberry**. Recomendamos que você [altere a senha padrão](#).

Configurar a câmera Raspberry Pi

Siga estas etapas para configurar a câmera Raspberry Pi para enviar vídeo do dispositivo para um stream de vídeo do Kinesis.

1. Abra um editor para atualizar o arquivo `modules` com o seguinte comando:

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

2. Adicione a seguinte linha ao final do arquivo, se ainda não houver uma:

```
bcm2835-v4l2
```

3. Salve o arquivo e saia do editor (Ctrl-X).
4. Reinicie o Raspberry Pi:

```
sudo reboot
```

5. Ao reiniciar o dispositivo, conecte-o novamente por meio do aplicativo de terminal se estiver se conectando remotamente.
6. Aberto `raspi-config`:

```
sudo raspi-config
```

7. Escolha Opções de interface, Câmera antiga. Em versões mais antigas do sistema operacional Raspbian, essa opção de menu pode estar em Opções de interface, Câmera.

Habilite a câmera se ainda não estiver habilitada e reinicie se solicitado.

8. Verifique se a câmera está funcionando digitando o seguinte comando:

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

Se sua câmera estiver configurada corretamente, esse comando captura uma imagem da câmera, a salva em um arquivo chamado `test.jpg` e exibe mensagens informativas.

Pré-requisitos de instalação do software

O C++ Producer SDK exige que você instale os pré-requisitos de software a seguir no Raspberry Pi.

1. Atualize a lista de pacotes e instale as bibliotecas necessárias para criar o SDK. Digite os seguintes comandos:

```
sudo apt update
sudo apt install -y \
```

```
automake \  
build-essential \  
cmake \  
git \  
gstreamer1.0-plugins-base-apps \  
gstreamer1.0-plugins-bad \  
gstreamer1.0-plugins-good \  
gstreamer1.0-plugins-ugly \  
gstreamer1.0-tools \  
gstreamer1.0-omx-generic \  
libcurl4-openssl-dev \  
libgstreamer1.0-dev \  
libgstreamer-plugins-base1.0-dev \  
liblog4cplus-dev \  
libssl-dev \  
pkg-config
```

2. Copie o arquivo PEM a seguir em `/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
```

Baixe e crie o SDK do Kinesis Video Streams C++ Producer

Você pode baixar e criar o SDK do Kinesis Video Streams C++ Producer usando o procedimento a seguir. Essa abordagem leva mais tempo para compilar, dependendo de conectividade de rede e da velocidade do processador.

1. Faça o download do SDK . Digite:

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

2. Prepare um diretório de compilação. Digite:

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

3. Crie o SDK e os aplicativos de amostra. Dependendo do modelo do Raspberry Pi em que você está construindo, pode levar várias horas para ser executado pela primeira vez:

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=FALSE
make
```

Transmita vídeo para sua transmissão de vídeo do Kinesis e assista à transmissão ao vivo

1. Para executar o aplicativo de amostra, você precisa das seguintes informações:
 - O nome do streaming que você criou na seção [Pré-requisitos](#).
 - As credenciais da conta (ID de chave de acesso e chave de acesso secreta) que você criou em [Crie um usuário do IAM com permissão para gravar no Kinesis Video Streams](#).
2. Execute o aplicativo de amostra usando os comandos a seguir. Substitua os espaços reservados por valores para seu ambiente.

```
export GST_PLUGIN_PATH=Directory Where You Cloned the SDK/amazon-kinesis-video-streams-producer-sdk-cpp/build
export AWS_DEFAULT_REGION=AWS Region i.e. us-east-1
export AWS_ACCESS_KEY_ID=Access Key ID
export AWS_SECRET_ACCESS_KEY=Secret Access Key
./kvs_gstreamer_sample Your Stream Name
```

3. Se o aplicativo de amostra sair com um `library not found` erro, digite o comando a seguir para verificar se o projeto está vinculado corretamente às dependências de código aberto:

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

4. Abra o console do [Kinesis Video Streams](#).
5. Escolha o Stream name (Nome do streaming) do streaming que você criou.

O streaming de vídeo enviado do Raspberry Pi aparecerá no console.

Quando o stream está sendo reproduzido, você pode experimentar os seguintes recursos do console do Kinesis Video Streams:

- Na seção Video preview (Visualização de vídeo), use os controles de navegação para retroceder ou avançar rapidamente pelo streaming.

- Na seção Stream info (Informações de streaming), observe o codec, a resolução e a taxa de bits do streaming. Os valores da resolução e da taxa de bits estão com definições baixas propositalmente no Raspberry Pi para minimizar o uso da largura de banda para este tutorial. Para ver as CloudWatch métricas da Amazon que estão sendo criadas para seu stream, escolha View stream metrics in CloudWatch.
- Em Data retention period (Período de retenção de dados), observe que o streaming de vídeo é retido por um dia. Você pode editar esse valor e defini-lo como No data retention (Sem retenção de dados) ou definir um valor de um dia a vários anos.

Em criptografia do lado do servidor, observe que seus dados estão sendo criptografados em repouso usando uma chave mantida pelo AWS Key Management Service (AWS KMS).

Referência do Producer SDK

Esta seção contém limites, códigos de erro e outras informações de referência para o [Bibliotecas do Kinesis Video Streams Producer](#).

Tópicos

- [Limites do SDK do produtor](#)
- [Referência de código de erro](#)
- [Referência de sinalizadores de adaptação do Network Abstraction Layer \(NAL\)](#)
- [Estruturas do SDK do produtor](#)
- [Estruturas de transmissão de vídeo do Kinesis](#)
- [Retornos de chamada do SDK do produtor](#)

Limites do SDK do produtor

A seguinte tabela contém os limites atuais para valores no [Bibliotecas de produtores](#).

Note

Antes de definir esses valores, você deve validar suas entradas. O SDK não valida esses limites, e ocorrerá um erro em tempo de execução se os limites forem excedidos.

Valor	Limite	Observações
Contagem máxima de streamings	128	O número máximo de streamings que um objeto de produtor pode criar. Trata-se de um limite flexível (você pode solicitar um aumento). Isso garante que o produtor não crie fluxos de forma recursiva acidentalmente.
Tamanho máximo de nome de dispositivo	128 caracteres	
Contagem máxima de tag	50 por streaming	
Tamanho máximo de nome de streaming	256 caracteres	
Tamanho mínimo de armazenamento	10 MiB = $10 * 1024 * 1024$ bytes	
Tamanho máximo de armazenamento	10 GiB = $10 * 1024 * 1024 * 1024$ bytes	
Tamanho máximo de caminho de diretório raiz	4,096 caracteres	
Tamanho máximo de informações de autenticação	10,000 bytes	
Tamanho máximo de string de URI	10 mil caracteres	
Tamanho máximo de nome de tag	128 caracteres	
Tamanho máximo de valor de tag	1,024 caracteres	

Valor	Limite	Observações
Período mínimo de token de segurança	30 segundos	
Período de carência de token de segurança	40 minutos	Se a duração especificada for maior, ela será limitada a esse valor.
Período de retenção	0 ou superior a uma hora	0 indica que não há retenção.
Duração mínima de cluster	1 segundo	O valor é especificado em unidades de 100 ns, o que é o padrão do SDK.
Duração máxima de cluster	30 segundos	O valor é especificado em unidades de 100 ns, o que é o padrão do SDK. A API de back-end pode impor uma duração menor do cluster.
Tamanho máximo de fragmento	50 MB	Para ter mais informações, consulte Cotas do serviço Kinesis Video Streams .
Duração máxima de fragmento	20 segundos	Para ter mais informações, consulte Cotas do serviço Kinesis Video Streams .
Duração máxima de conexão	45 minutos	O back-end encerra a conexão após esse período. O SDK gira o token e estabelece uma nova conexão nesse tempo.
Tamanho máximo de segmento de ACK	1,024 caracteres	Tamanho máximo de segmento da confirmação enviada para a função de analisador de ACK.

Valor	Limite	Observações
Tamanho máximo da string de tipo de conteúdo.	128 caracteres	
Tamanho máximo de string de ID do codec	32 caracteres	
Tamanho máximo de string de nome do controle	32 caracteres	
Tamanho máximo de dados privados do codec	1 MiB = 1 * 1024 * 1024 bytes	
Tamanho mínimo de valor de escala de código de hora	100 ns	O valor mínimo de escala de código de hora para representar os time stamps dos quadros no cluster do MKV resultante. O valor é especificado em incrementos de 100 ns, o que é o padrão do SDK.
Tamanho máximo de valor de escala de código de hora	1 segundo	O valor máximo de escala de código de hora para representar os time stamps dos quadros no cluster do MKV resultante. O valor é especificado em incrementos de 100 ns, o que é o padrão do SDK.
Contagem mínima de itens de exibição do conteúdo	10	
Duração mínima de buffer	20 segundos	O valor é especificado em incrementos de 100 ns, o que é o padrão do SDK.
Tamanho máximo de versão de atualização	128 caracteres	

Valor	Limite	Observações
Tamanho máximo de ARN	1024 caracteres	
Tamanho máximo de sequência de fragmento	128 caracteres	
Período máximo de retenção	10 anos	

Referência de código de erro

Esta seção contém informações sobre código de status e erro para o [Bibliotecas de produtores](#).

Para obter informações sobre soluções para problemas comuns, consulte [Solução de problemas do Kinesis Video Streams](#).

Tópicos

- [Erros e códigos de status retornados por PutFrame Callbacks - Código Independente de Plataforma \(PIC\)](#)
- [Erros e códigos de status retornados por retornos de PutFrame chamada - biblioteca de produtores C](#)

Erros e códigos de status retornados por PutFrame Callbacks - Código Independente de Plataforma (PIC)

As seções a seguir contêm informações de erro e status que são retornadas por retornos de chamada para a PutFrame operação dentro do Código Independente de Plataforma (PIC).

Tópicos

- [Códigos de erro e status retornados pela biblioteca cliente](#)
- [Códigos de erro e status retornados pela biblioteca de duração](#)
- [Códigos de erro e status retornados pela biblioteca comum](#)
- [Códigos de erro e status retornados pela biblioteca de pilhas](#)
- [Códigos de erro e status retornados pela biblioteca MKVGen](#)
- [Códigos de erro e status retornados pela biblioteca Trace](#)

- [Códigos de erro e status retornados pela biblioteca Utils](#)
- [Códigos de erro e status retornados pela biblioteca View](#)

Códigos de erro e status retornados pela biblioteca cliente

A tabela a seguir contém informações de erro e status que são retornadas por métodos na biblioteca do Kinesis Client Video Streams.

Código	Message	Descrição	Ação recomendada
0x52000001	STATUS_MAX_STREAM_COUNT	A contagem máxima de streamings foi atingida.	Especifique uma contagem máxima de streamings maior em DeviceInfo conforme especificado em Limites do SDK do produtor .
0x52000002	STATUS_MIN_STREAM_COUNT	Erro de contagem mínima de streamings.	Especifique o número máximo de fluxos maiores que zero em DeviceInfo.
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	Tamanho de nome de dispositivo inválido.	Consulte o tamanho máximo do nome do dispositivo em caracteres especificado em Limites do SDK do produtor .
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	Versão da estrutura DeviceInfo inválida.	Especifique a versão atual da estrutura.
0x52000005	STATUS_MAX_TAG_COUNT	A contagem máxima de tag foi atingida.	Consulte a contagem máxima atual de tags especificada em Limites do SDK do produtor .

Código	Message	Descrição	Ação recomendada
0x52000006	STATUS_DE VICE_FING ERPRINT_LENGTH		
0x52000007	STATUS_IN VALID_CAL LBACKS_VERSION	Versão da estrutura Callbacks inválida.	Especifique a versão atual da estrutura.
0x52000008	STATUS_IN VALID_STR EAM_INFO_VERSION	Versão da estrutura StreamInfo inválida.	Especifique a versão atual da estrutura.
0x52000009	STATUS_IN VALID_STR EAM_NAME_LENGTH	Tamanho de nome de streaming inválido.	Consulte o tamanho máximo do nome do stream em caracteres especificado em Limites do SDK do produtor .
0x5200000a	STATUS_IN VALID_STO RAGE_SIZE	Um tamanho de armazenamento inválido foi especific ado.	O tamanho de armazenamento em bytes deve estar dentro dos limites especificados em Limites do SDK do produtor .
0x5200000b	STATUS_IN VALID_R00 T_DIRECTO RY_LENGTH	Tamanho de string de diretório raiz inválido.	Consulte o comprimen to máximo do caminho do diretório raiz especific ado em Limites do SDK do produtor .
0x5200000c	STATUS_IN VALID_SPI LL_RATIO	Proporção de vazamento inválida.	Expresse a taxa de derramamento como uma porcentagem de 0 a 100.

Código	Message	Descrição	Ação recomendada
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Versão da estrutura StorageInfo inválida.	Especifique a versão atual da estrutura.
0x5200000e	STATUS_INVALID_STREAM_STATE	O streaming está em um estado que não permite a operação atual.	Geralmente, esse erro ocorre quando o SDK não consegue atingir o estado necessário para realizar a operação solicitada. Por exemplo, ele ocorre quando há falha na chamada de API GetStreamingEndpoint e o aplicativo cliente a ignora, continuando a colocar quadros no streaming.
0x5200000f	STATUS_SERVICE_CALLBACK_MISSING	A estrutura Callbacks tem pontos de entrada de função ausentes para algumas funções obrigatórias.	Verifique se os retornos de chamada obrigatórios estão implementados no aplicativo cliente. Esse erro é exposto somente para clientes de Código Independente de Plataforma (PIC). C++ e outros wrappers de nível superior atendem a essas chamadas.

Código	Message	Descrição	Ação recomendada
0x52000010	STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR	Não autorizado.	Verifique o token de segurança, o certificado, a integração e a expiração do token de segurança. Verifique se o token tem os direitos associados corretos a ele. Para os aplicativos de amostra do Kinesis Video Streams, verifique se a variável de ambiente está definida corretamente.
0x52000011	STATUS_DESCRIBESTREAM_CALL_FAILED	Falha da API DescribeStream .	Esse erro será retornado após a falha de repetição da API DescribeStream . O cliente PIC retorna esse erro depois de parar de tentar novamente.
0x52000012	STATUS_INVALID_DESCRIPTOR_RESPONSE	Estrutura DescribeStreamResponse inválida.	A estrutura que foi passada ao DescribeStreamResultEvent é nula ou contém itens inválidos, como um nome de recurso da Amazon (ARN) nulo.

Código	Message	Descrição	Ação recomendada
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	O streaming está sendo excluído.	Uma falha de API foi causada pelo streaming sendo excluído. Verifique se nenhum outro processo está tentando excluir o fluxo enquanto o fluxo está em uso.
0x52000014	STATUS_SERVICE_CALL_INVALID_ARGUMENT_ERROR	Argumentos inválidos foram especificados para a chamada de serviço.	O back-end retorna esse erro quando um argumento de chamada de serviço não é válido ou quando o SDK encontra um erro que não consegue interpretar.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	O dispositivo não foi encontrado.	Verifique se o dispositivo não foi excluído durante o uso.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	O dispositivo não foi provisionado.	Verifique se o dispositivo foi provisionado.

Código	Message	Descrição	Ação recomendada
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	Recurso genérico não encontrado e retornado do serviço.	Esse erro ocorre quando o serviço não consegue localizar o recurso (por exemplo, um streaming). Ele pode ter significados diferentes em contextos diferentes, mas a causa provável é o uso de APIs antes que o streaming seja criado. O uso do SDK confirma que o stream foi criado primeiro.
0x52000018	STATUS_INVALID_AUTH_LEN	Tamanho de informações de autenticação inválido.	Consulte os valores atuais que são especificados em Limites do SDK do produtor .
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	Falha na chamada da API <code>CreateStream</code> .	Consulte a string do erro para obter informações mais detalhadas sobre o motivo da falha na operação.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	Falha na chamada <code>GetStreamingToken</code> .	Consulte a string do erro para obter informações mais detalhadas sobre o motivo da falha na operação.
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	Falha na chamada da API <code>GetStreamingEndpoint</code> .	Consulte a string do erro para obter informações mais detalhadas sobre o motivo da falha na operação.

Código	Message	Descrição	Ação recomendada
0x5200002c	STATUS_INVALID_URI_LEN	Um tamanho de string de URI inválido foi retornado da API <code>GetStreamingEndpoint</code> .	Consulte os valores máximos atuais que são especificados em Limites do SDK do produtor .
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	Falha na chamada da API <code>PutMedia</code> .	Consulte a string do erro para obter informações mais detalhadas sobre o motivo da falha na operação.

Código	Message	Descrição	Ação recomendada
0x5200002e	STATUS_STREAM_OUT_OF_MEMORY	Não há memória no armazenamento de conteúdo.	O armazenamento de conteúdo é compartilhado entre os streamings e deve ter capacidade suficiente para armazenar as durações máximas de todos os streamings + ~20% (representando a desfragmentação). É importante não exceder o armazenamento. Escolha valores para a duração máxima por streaming que correspondam ao tamanho de armazenamento cumulativo e às tolerâncias de latência. Recomendamos eliminar os quadros quando eles caem da janela de visualização do conteúdo, em vez de simplesmente serem colocados (pressão na memória do armazenamento de conteúdo). Isso ocorre porque a eliminação dos quadros inicia os retornos de chamada de notificação de pressão do fluxo. Com isso, o aplicativo pode ajustar os componentes de mídia de upstream (como o

Código	Message	Descrição	Ação recomendada
			codificador) para diminuir a taxa de bits, soltar os quadros e agir apropriadamente.
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	Atualmente, não há mais dados disponíveis para um streaming.	Trata-se de um possível resultado válido quando a pipeline de mídia produz mais lentamente do que o thread de redes consome os quadros a serem enviados ao serviço. Clientes de nível superior (por exemplo, C++, Java ou Android) não veem esse aviso porque ele é tratado internamente.
0x52000030	STATUS_INVALID_TAG_VERSION	Versão da estrutura Tag inválida.	Especifique a versão atual da estrutura.
0x52000031	STATUS_SERVICE_UNKNOWN_ERROR	Um erro desconhecido ou genérico foi retornado da pilha de redes.	Consulte os logs para obter informações mais detalhadas.
0x52000032	STATUS_SERVICE_RESOURCE_IN_USE_ERROR	Recurso em uso.	Retornado do serviço. Para obter mais informações, consulte a Referência da API Kinesis Video Streams.

Código	Message	Descrição	Ação recomendada
0x52000033	STATUS_SERVICE_CLIENT_LIMIT_ERROR	Limite do cliente.	Retornado do serviço. Para obter mais informações, consulte a Referência da API Kinesis Video Streams.
0x52000034	STATUS_SERVICE_DEVICE_LIMIT_ERROR	Limite do dispositivo.	Retornado do serviço. Para obter mais informações, consulte a Referência da API Kinesis Video Streams.
0x52000035	STATUS_SERVICE_STREAM_LIMIT_ERROR	Limite do streaming.	Retornado do serviço. Para obter mais informações, consulte a Referência da API Kinesis Video Streams.
0x52000036	STATUS_SERVICE_RESOURCE_DELETED_ERROR	O recurso foi excluído ou está sendo excluído.	Retornado do serviço. Para obter mais informações, consulte a Referência da API Kinesis Video Streams.
0x52000037	STATUS_SERVICE_TIMEOUT_ERROR	A chamada de serviço expirou.	Chamar uma API de serviço específica resultou no esgotamento do tempo. Verifique se você tem uma conexão de rede válida. O PIC repetirá a operação automaticamente.

Código	Message	Descrição	Ação recomendada
0x52000038	STATUS_STREAM_READY_CALLBACK_FAILED	Notificação de streaming pronto.	Essa notificação é enviada do PIC para o cliente indicando que o streaming assíncrono foi criado.
0x52000039	STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	Tags inválidas foram especificadas.	A contagem de tags não é zero, mas as tags estão vazias. Verifique se as tags estão especificadas ou se a contagem é zero.
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	Versão da estrutura StreamDescription inválida.	Especifique a versão atual da estrutura.
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	Tamanho de nome de tag inválido.	Consulte os limites para o nome da tag que são especificados em Limites do SDK do produtor .
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	Tamanho de valor de tag inválido.	Consulte os limites para o valor da tag que são especificados em Limites do SDK do produtor .
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	Falha na API TagResource .	Falha na chamada da API TagResource . Verifique se há uma conexão de rede válida. Consulte os logs para obter mais informações sobre a falha.

Código	Message	Descrição	Ação recomendada
0x5200003e	STATUS_INVALID_CUSTOM_DATA	Dados personalizados inválidos chamando APIs do PIC.	Dados personalizados inválidos foram especificados em uma chamada para as APIs do PIC. Isso só pode ocorrer nos clientes que usam diretamente o PIC.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	Estrutura CreateStreamResponse inválida.	A estrutura ou seus campos de membro são inválidos (ou seja, o ARN é nulo ou maior do que o que está especificado em Limites do SDK do produtor).
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	Falha na autenticação do cliente.	O PIC não conseguiu obter as informações de autenticação adequadas (AccessKeyId ou SecretAccessKey) após várias tentativas. Verifique a integração de autenticação. Os aplicativos de amostra usam as variáveis de ambiente para inserir informações da credencial na Biblioteca de produtores C++.

Código	Message	Descrição	Ação recomendada
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	Falha na obtenção de chamada de token de segurança.	Essa situação pode ocorrer com os clientes que usam diretamente o PIC. Após várias tentativas, a chamada falha e apresenta esse erro.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	Erro de provisionamento.	O provisionamento não está implementado.
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	Falha ao criar o cliente produtor.	Um erro genérico retornado pelo PIC após várias tentativas quando há falha na criação do cliente.
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	Falha ao levar o cliente produtor ao um estado PRONTO.	Retornada pela máquina de estado do PIC se este falhar ao levar ao estado PRONTO. Consulte os logs para obter mais informações sobre a causa raiz.
0x52000045	STATUS_TAG_CLIENT_CALL_FAILED	Falha no TagResource do cliente produtor.	Falha na chamada da API TagResource do cliente produtor. Consulte os logs para obter mais informações sobre a causa raiz.

Código	Message	Descrição	Ação recomendada
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	Falha na criação de dispositivo/produtor.	Os SDKs de nível superior (por exemplo, C++ ou Java) ainda não implementam a API de criação do dispositivo ou do produtor. Clientes que usam o PIC diretamente podem indicar uma falha usando a notificação do resultado.
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	O time stamp da ACK recebida não está na visualização.	Isso ocorre se o quadro correspondente à ACK recebida estiver fora da janela de exibição de conteúdo. Geralmente, isso ocorre quando a entrega da ACK é lenta. Isso pode ser interpretado como um aviso ou uma indicação de que o downlink está lento.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	Versão da estrutura FragmentAck inválida.	Especifique a versão atual da estrutura FragmentAck .

Código	Message	Descrição	Ação recomendada
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	Expiração do token de segurança inválido.	A expiração do token de segurança deve ter um timestamp absoluto no futuro que seja maior do que o timestamp atual, com um período de carência. Para obter informações sobre os limites do período de carência, consulte Limites do SDK do produtor .
0x5200004a	STATUS_END_OF_STREAM	Indicador de término de streaming (EOS).	Na chamada da API <code>GetStreamData</code> , ele indica que a sessão de processamento de upload atual chegou ao fim. Isso ocorre se a sessão terminar ou houver erros ou se o token da sessão expirou e a sessão está sendo girada.
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	Nome de streaming duplicado.	Streamings diferentes não podem ter o mesmo nome. Escolha um nome exclusivo para cada streaming.

Código	Message	Descrição	Ação recomendada
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	Período de retenção inválido.	Um período de retenção inválido é especificado na estrutura <code>StreamInfo</code> . Para obter informações sobre o intervalo de valores válidos para o período de retenção, consulte Limites do SDK do produtor .
0x5200004d	STATUS_INVALID_ACK_KEY_START	Inválido <code>FragmentAck</code> .	Falha ao analisar a string de ACK do fragmento. Indicador de início de chave inválido. A string de ACK do fragmento pode estar danificada. Ela pode se corrigir e o erro pode ser tratado como um aviso.
0x5200004e	STATUS_INVALID_ACK_DUPLICATE_KEY_NAME	Inválido <code>FragmentAck</code> .	Falha ao analisar a string de ACK do fragmento. Várias chaves têm o mesmo nome. A string de ACK do fragmento pode estar danificada. Ela pode se corrigir e o erro pode ser tratado como um aviso.

Código	Message	Descrição	Ação recomendada
0x5200004f	STATUS_INVALID_ACK_VALUE_START	Inválido FragmentAck .	Falha ao analisar a string de ACK do fragmento por causa de um indicador de início de valor de chave inválido. A string de ACK do fragmento pode estar danificada. Ela pode se corrigir, e o erro pode ser tratado como um aviso.
0x52000050	STATUS_INVALID_ACK_VALUE_END	Inválido FragmentAck .	Falha ao analisar a string de ACK do fragmento por causa de um indicador de fim de valor de chave inválido. A string de ACK do fragmento pode estar danificada. Ela pode se corrigir e o erro pode ser tratado como um aviso.
0x52000051	STATUS_INVALID_ACK_TYPE	Inválido FragmentAck .	Falha ao analisar a string de ACK do fragmento porque um tipo ACK inválido foi especificado.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	O streaming foi interrompido.	O streaming foi interrompido, mas um quadro ainda está sendo colocado no streaming.
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Versão da estrutura StreamMetrics inválida.	Especifique a versão atual da estrutura StreamMetrics .

Código	Message	Descrição	Ação recomendada
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Versão da estrutura <code>ClientMetrics</code> inválida.	Especifique a versão atual da estrutura <code>ClientMetrics</code> .
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	Falha na inicialização de produtor para atingir um estado PRONTO.	Falha ao atingir o estado PRONTO durante a inicialização do cliente produtor. Consulte os logs para obter mais informações.
0x52000056	STATUS_STATE_MACHINE_STATE_NOT_FOUND	Erro de máquina de estado interno.	Não é um erro publicamente visível.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	O tipo de ACK inválido é especificado na estrutura <code>FragmentAck</code> .	A estrutura <code>FragmentAck</code> deve conter tipos de ACK definidos no cabeçalho público.
0x52000058	STATUS_INVALID_STREAM_READY_STATE	Erro de transição de máquina de estado interno.	Não é um erro publicamente visível.
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	O objeto de streaming foi liberado após a liberação do produtor.	Houve uma tentativa de liberar um objeto de streaming após a liberação do objeto do produtor. Isso só pode ocorrer em clientes que usam diretamente o PIC.

Código	Message	Descrição	Ação recomendada
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	Erro de armazenamento interno.	Um erro interno indicando que o tamanho real da alocação do armazenamento de conteúdo é menor que o tamanho do quadro e do fragmento empacotados.
0x5200005b	STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION	Erro de armazenamento interno.	O tamanho armazenado da alocação na exibição do conteúdo é maior que o tamanho da alocação no armazenamento do conteúdo.
0x5200005c	STATUS_ACK_ERR_STREAM_READ_ERROR	ACK de erro de leitura de streaming .	Um erro que o ACK retornou do back-end indicando um erro de leitura ou análise do fluxo. Isso costuma ocorrer quando o back-end não consegue recuperar o streaming. Em geral, o restreaming automático pode corrigir esse erro.

Código	Message	Descrição	Ação recomendada
0x5200005d	STATUS_AK_ERR_FRAGMENT_SIZE_REACHED	O tamanho de fragmento máximo foi atingido.	O tamanho de fragmento máximo em bytes é definido em Limites do SDK do produtor . Esse erro indica que há quadros muito grandes ou que não há quadros-chave para criar fragmentos de tamanho gerenciável. Verifique as configurações do codificador e verifique se os quadros-chave estão sendo produzidos corretamente. Em caso de streamings que têm densidade muito alta, configure o codificador para produzir fragmentos com menores durações de modo a gerenciar o tamanho máximo.

Código	Message	Descrição	Ação recomendada
0x5200005e	STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED	A duração de fragmento máxima foi atingida.	A duração de fragmento máxima é definida em Limites do SDK do produtor . Esse erro indica que há quadros muito baixos por segundo ou que não há quadros-chave para criar fragmentos de duração gerenciável. Verifique as configurações do codificador e verifique se os quadros-chave estão sendo produzidos adequadamente em intervalos regulares.
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	A duração de conexão máxima foi atingida.	O Kinesis Video Streams impõe a duração máxima da conexão, conforme especificado no Limites do SDK do produtor . O Producer SDK gira automaticamente o stream ou o token antes que o máximo seja atingido. Os clientes que usam o SDK não devem receber esse erro.
0x52000060	STATUS_ACK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC	Os códigos de hora não estão crescendo monotonicamente.	O Producer SDK impõe registros de data e hora, portanto, os clientes que usam o SDK não devem receber esse erro.

Código	Message	Descrição	Ação recomendada
0x52000061	STATUS_AC K_ERR_MUL TI_TRACK_MKV	Vários controles no MKV.	O Producer SDK impõe fluxos de faixa única, portanto, os clientes que usam o SDK não devem receber esse erro.
0x52000062	STATUS_AC K_ERR_INV ALID_MKV_DATA	Dados do MKV inválidos.	O analisador de MKV de back-end encontrou um erro analisando o streaming. Os clientes que usam o SDK podem encontrar esse erro se o stream for corrompido na transição. Isso também pode acontecer se as pressões do buffer forcem o SDK a descartar quadros traseiros que são parcialmente transmitidos. No último caso, recomendamos que você reduza o FPS e a resolução, aumente a taxa de compactação ou (se houver uma rede “intermitente”) permita maior armazenamento de conteúdo e duração do buffer para acomodar as pressões temporárias.

Código	Message	Descrição	Ação recomendada
0x52000063	STATUS_AC K_ERR_INV ALID_PROD UCER_TIMESTAMP	Time stamp do produtor inválido.	O serviço retorna essa ACK de erro se o relógio do produtor tiver um grande desvio no futuro. SDKs de nível superior (por exemplo, Java ou C++) usam uma versão do relógio do sistema para atender ao retorno de chamada da hora atual do PIC. Verifique se o relógio do sistema está ajustado corretamente. Os clientes que usam o PIC diretamente devem verificar se suas funções de retorno de chamada retornam o timestamp correto.
0x52000064	STATUS_AC K_ERR_STR EAM_NOT_ACTIVE	Streaming inativo.	Uma chamada de API de back-end foi realizada enquanto o streaming não estava em estado "Ativo". Isso ocorre quando o cliente cria o streaming e continua a enviar imediatamente quadros para ele. O SDK lida com esse cenário por meio da máquina de estado e do mecanismo de recuperação.

Código	Message	Descrição	Ação recomendada
0x52000065	STATUS_AK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS erro de acesso negado.	Retornado quando a conta não tem acesso à chave especificada.
0x52000066	STATUS_AK_ERR_KMS_KEY_DISABLED	AWS KMS a chave está desativada.	A chave especificada foi desativada.
0x52000067	STATUS_AK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS erro de validação da chave.	Erro de validação genérico. Para obter mais informações, consulte a Referência da API do AWS Key Management Service .
0x52000068	STATUS_AK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key indisponível.	A chave está indisponível. Para obter mais informações, consulte a Referência da API do AWS Key Management Service .
0x52000069	STATUS_AK_ERR_KMS_KEY_INVALID_USAGE	Uso inválido da chave KMS.	O não AWS KMS key está configurado para ser usado nesse contexto. Para obter mais informações, consulte a Referência da API do AWS Key Management Service .
0x5200006a	STATUS_AK_ERR_KMS_KEY_INVALID_STATE	AWS KMS estado inválido.	Para obter mais informações, consulte a Referência da API do AWS Key Management Service .

Código	Message	Descrição	Ação recomendada
0x5200006b	STATUS_ACK_ERR_KMS_KEY_NOT_FOUND	A chave KMS não foi encontrada.	A chave não foi encontrada. Para obter mais informações, consulte a Referência da API do AWS Key Management Service .
0x5200006c	STATUS_ACK_ERR_STREAM_DELETED	O streaming foi ou está sendo excluído.	O streaming está sendo excluído por outro aplicativo ou por meio do AWS Management Console.
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	Erro interno.	Erro interno de serviço genérico.
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARCHIVAL_ERROR	Erro de arquivamento de fragmento.	Retornado quando o serviço não persiste por muito tempo e indexa o fragmento. Embora seja raro, ele pode ocorrer por várias razões. Por padrão, o SDK tentará enviar o fragmento.
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	Erro desconhecido.	O serviço retornou um erro desconhecido.
0x52000070	STATUS_MISSING_ERR_ACK_ID	Informações de ACK ausentes.	O analisador de ACK concluiu a análise, mas as informações de <code>FragmentAck</code> estão ausentes.

Código	Message	Descrição	Ação recomendada
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	Tamanho de segmento de ACK inválido.	Uma string de segmento de ACK com um tamanho inválido foi especificada para o analisador de ACK. Para ter mais informações, consulte Limites do SDK do produtor .
0x52000074	STATUS_MAX_FRAGMENT_METADATA_COUNT	O número máximo de itens de metadados foi adicionado a um fragmento.	Um stream de vídeo do Kinesis pode adicionar até 10 itens de metadados a um fragmento, seja adicionando um item não persistente a um fragmento ou adicionando um item persistente à fila de metadados. Para ter mais informações, consulte Usando metadados de streaming com o Kinesis Video Streams .
0x52000075	STATUS_ACK_FRAGMENT_METADATA_LIMIT_REACHED	Um limite (contagem máxima de metadados, tamanho do nome dos metadados ou tamanho do valor dos metadados) foi atingido.	O Producer SDK limita o número e o tamanho dos itens de metadados. Esse erro não ocorre a menos que os limites no código do SDK do Produtor sejam alterados. Para ter mais informações, consulte Usando metadados de streaming com o Kinesis Video Streams .

Código	Message	Descrição	Ação recomendada
0x52000076	STATUS_BLOCKING_INTERRUPTED_STREAM_TERMINATED	Não implementado.	
0x52000077	STATUS_INVALID_METADATA_NAME	O nome dos metadados não é válido.	O nome dos metadados não pode começar com a string "AWS". Se esse erro ocorrer, o item de metadados não será adicionado ao fragmento ou à fila de metadados. Para ter mais informações, consulte Usando metadados de streaming com o Kinesis Video Streams .
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	O quadro final de um fragmento está em um estado inválido.	O final do fragmento não deve ser enviado em um fluxo non-key-frame fragmentado.
0x52000079	STATUS_TRACK_INFO_MISSING	Faltam informações da trilha.	O número da faixa deve ser maior que zero e corresponder ao ID da faixa.
0x5200007a	STATUS_MAX_TRACK_COUNT_EXCEEDED	A contagem máxima da trilha foi excedida.	Você pode ter no máximo três faixas por stream.

Código	Message	Descrição	Ação recomendada
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	O tempo de retenção do modo de streaming offline está definido como zero.	O tempo de retenção do modo de streaming offline não deve ser definido como zero.
0x5200007c	STATUS_ACK_TRACK_NUMBER_MISMATCH	O número da trilha do ACK de erro é incompatível.	
0x5200007d	STATUS_ACK_FRAMES_MISSING_FOR_TRACK	Faltam quadros para uma trilha.	
0x5200007e	STATUS_ACK_MORE_THAN_ALLOWED_TRACKS_FOUND	O número máximo permitido de trilhas foi excedido.	
0x5200007f	STATUS_UPLOAD_HANDLE_ABORTED	O processamento do upload foi anulado.	
0x52000080	STATUS_INVALID_CERT_PATH_LENGTH	Tamanho do caminho do certificado inválido.	
0x52000081	STATUS_DUPLICATE_TRACK_ID_FOUND	Encontrado ID de trilha duplicado.	

Código	Message	Descrição	Ação 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_MAXIMUM_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUBLISH_MEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		

Código	Message	Descrição	Ação recomendada
0x52000089	STATUS_NO N_ALIGNED _HEAP_WIT H_IN_CONT ENT_STORE _ALLOCATORS		
0x5200008a	STATUS_MU LTIPLE_CO NSECUTIVE_EOFR		
0x5200008b	STATUS_DU PLICATE_S TREAM_EVENT_TYPE		
0x5200008c	STATUS_ST REAM_NOT_STARTED		
0x5200008d	STATUS_IN VALID_IMA GE_PREFIX_LENGTH		
0x5200008e	STATUS_IN VALID_IMA GE_METADA TA_KEY_LENGTH		
0x5200008f	STATUS_IN VALID_IMA GE_METADA TA_VALUE_LENGTH		

Códigos de erro e status retornados pela biblioteca de duração

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na `Duration` biblioteca.

Código	Message
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

Códigos de erro e status retornados pela biblioteca comum

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na Common biblioteca.

Note

Esses códigos de informações de erro e status são comuns a muitas APIs.

Código	Código sem zeros iniciais	Message	Descrição
0x00000001	0x1	STATUS_NULL_ARG	NULO foi transmitido para um argumento obrigatório.
0x00000002	0x2	STATUS_INVALID_ARG	Um valor inválido foi especificado para um argumento.
0x00000003	0x3	STATUS_INVALID_ARG_LEN	Um tamanho de argumento inválido foi especificado.
0x00000004	0x4	STATUS_NOT_ENOUGH_MEMORY	Não foi possível alocar memória suficiente.
0x00000005	0x5	STATUS_BUFFER_TOO_SMALL	O tamanho do buffer especificado é muito pequeno.

Código	Código sem zeros iniciais	Message	Descrição
0x00000006	0x6	STATUS_UNEXPECTED_EOF	Um final de arquivo inesperado foi atingido.
0x00000007	0x7	STATUS_FORMAT_ERROR	Um formato inválido foi encontrado.
0x00000008	0x8	STATUS_INVALID_HANDLE_ERROR	Valor de processamento inválido.
0x00000009	0x9	STATUS_OPEN_FILE_FAILED	Falha ao abrir um arquivo.
0x0000000a	0xa	STATUS_READ_FILE_FAILED	Falha ao ler um arquivo.
0x0000000b	0xb	STATUS_WRITE_TO_FILE_FAILED	Falha ao gravar em um arquivo.
0x0000000c	0xc	STATUS_INTERNAL_ERROR	Um erro interno que não costuma ocorrer e que pode indicar um bug de SDK ou de API de serviço.
0x0000000d	0xd	STATUS_INVALID_OPERATION	Houve uma operação inválida ou a operação não é permitida.
0x0000000e	0xe	STATUS_NOT_IMPLEMENTED	O recurso não está implementado.

Código	Código sem zeros iniciais	Message	Descrição
0x0000000f	0xf	STATUS_OPERATION_TIMED_OUT	A operação expirou.
0x00000010	0x10	STATUS_NOT_FOUND	Um recurso necessário não foi encontrado.
0x00000011	0x11	STATUS_CREATE_THREAD_FAILED	Falha ao criar um tópico.
0x00000012	0x12	STATUS_THREAD_NOT_ENOUGH_RESOURCES	Recursos insuficientes para criar outro encadeamento ou foi encontrado um limite imposto pelo sistema no número de encadeamentos.
0x00000013	0x13	STATUS_THREAD_INVALID_ARG	Atributos de encadeamento inválidos especificados ou outro encadeamento já está aguardando para se juntar a esse encadeamento.
0x00000014	0x14	STATUS_THREAD_PERMISSIONS	Sem permissão para definir a política de agendamento e os parâmetros especificados nos atributos do thread.

Código	Código sem zeros iniciais	Message	Descrição
0x00000015	0x15	STATUS_TH READ_DEAD LOCKED	Um impasse foi detectado ou o encadeamento de junção específica o encadeamento de chamada.
0x00000016	0x16	STATUS_TH READ_DOES _NOT_EXIST	Não foi possível encontrar o tópico com o ID do tópico especificado.
0x00000017	0x17	STATUS_JO IN_THREAD _FAILED	Um erro desconhecido ou genérico foi retornado da operação de junção de encadeamentos.
0x00000018	0x18	STATUS_WA IT_FAILED	Excedeu o tempo máximo de espera pela variável condicional.
0x00000019	0x19	STATUS_CA NCEL_THRE AD_FAILED	Um erro desconhecido ou genérico foi retornado da operação de cancelamento do thread.
0x0000001a	0x1a	STATUS_TH READ_IS_N OT_JOINABLE	A operação de junção de encadeamento é solicitada em um encadeamento não juntável.

Código	Código sem zeros iniciais	Message	Descrição
0x0000001b	0x1b	STATUS_DETACH_THREAD_FAILED	Um erro desconhecido ou genérico foi retornado da operação de desanexação do thread.
0x0000001c	0x1c	STATUS_THREAD_READ_ATTRIBUTES_INIT_FAILED	Falha ao inicializar o objeto de atributos do thread.
0x0000001d	0x1d	STATUS_THREAD_READ_ATTRIBUTES_SET_STACK_SIZE_FAILED	Falha ao definir o tamanho da pilha para o objeto de atributos do encadeamento.
0x0000001e	0x1e	STATUS_MEMORY_NOT_FREED	Usado apenas nos testes. Indica que nem toda a memória solicitada foi liberada.

Códigos de erro e status retornados pela biblioteca de pilhas

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na Heap biblioteca.

Código	Message	Descrição
0x10000001	STATUS_HEAP_FLAGS_ERROR	Uma combinação inválida de sinalizadores foi especificada.
0x10000002	STATUS_HEAP_NOT_INITIALIZED	Uma operação foi tentada antes de o heap ter sido inicializado.

Código	Message	Descrição
0x10000003	STATUS_HEAP_CORRUPTED	O heap foi corrompido ou a faixa de guarda (no modo de depuração) foi substituída. Um estouro do buffer no código do cliente pode causar a corrupção do heap.
0x10000004	STATUS_HEAP_VRAM_LIB_MISSING	A biblioteca do usuário ou do modo kernel da VRAM (RAM de vídeo) não pode ser carregada ou está ausente. Verifique se a plataforma subjacente oferece suporte a alocações de VRAM.
0x10000005	STATUS_HEAP_VRAM_LIB_REOPEN	Falha ao abrir a biblioteca de VRAM.
0x10000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	Falha ao carregar a exportação da função INIT.
0x10000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	Falha ao carregar a exportação da função ALLOC.
0x10000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	Falha ao carregar a exportação da função FREE.
0x10000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	Falha ao carregar a exportação da função LOCK.
0x1000000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	Falha ao carregar a exportação da função UNLOCK.
0x1000000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	Falha ao carregar a exportação da função UNINIT.

Código	Message	Descrição
0x1000000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	Falha ao carregar a exportação da função GETMAX.
0x1000000d	STATUS_HEAP_DIRECT_MEM_INIT	Falha ao inicializar o grupo de heap principal no heap híbrido.
0x1000000e	STATUS_HEAP_VRAM_INIT_FAILED	Falha na inicialização dinâmica de VRAM.
0x1000000f	STATUS_HEAP_LIBRARY_FREE_FAILED	Falha ao desalocar e liberar a biblioteca de VRAM.
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	Falha na alocação de VRAM.
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	Falha na liberação de VRAM.
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	Falha no mapeamento de VRAM.
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	Falha no desmapeamento de VRAM.
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	Falha na desinicialização dinâmica de VRAM.
0x10000015	STATUS_INVALID_ALLLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

Códigos de erro e status retornados pela biblioteca MKVGen

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na MKVGen biblioteca.

Código	Message	Descrição/Ação recomendada
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Membros inválidos da estrutura de dados Frame. Certifique-se de que os dados de duração, tamanho e quadro sejam válidos e estejam dentro dos limites especificados em Limites do SDK do produtor .
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	Time stamp do quadro inválido. O PTS (time stamp de apresentação) e o DTS (time stamp de decodificação) calculados são superiores ou iguais ao time stamp do quadro inicial do fragmento. Essa é uma indicação de um possível reinício de pipeline de mídia ou um problema de estabilidade de decodificador. Para obter informações sobre a solução de problemas, consulte Erro: "falha ao enviar quadro ao cliente de vídeo do Kinesis"
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	Uma duração de fragmento inválida foi especificada. Para ter mais informações, consulte Limites do SDK do produtor .

Código	Message	Descrição/Ação recomendada
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	Tamanho de string de tipo de conteúdo inválido. Para ter mais informações, consulte Limites do SDK do produtor .
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	Houve uma tentativa de codificar um número que é grande demais para ser representado no formato EBML (Extensible Binary Meta Language). Isso não deve ser exposto aos clientes do SDK.
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	Tamanho de string de ID do codec inválido. Para ter mais informações, consulte Limites do SDK do produtor .
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	Tamanho de string de nome do controle inválido. Para ter mais informações, consulte Limites do SDK do produtor .
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	Tamanho de dados privados do codec inválido. Para ter mais informações, consulte Limites do SDK do produtor .
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	Os dados privados do codec (CPD) são NULL, enquanto o tamanho do CPD é maior que zero.

Código	Message	Descrição/Ação recomendada
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	Valor de escala do código de hora inválido. Para ter mais informações, consulte Limites do SDK do produtor .
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	O código de hora do quadro é maior do que o máximo. Para ter mais informações, consulte Limites do SDK do produtor .
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	O código de hora de quadro máximo foi atingido. O formato MKV usa 16 bits assinados para representar o código de hora relativo do quadro no início do cluster. O erro é gerado se o timecode do quadro não puder ser representado. Esse erro indica que uma seleção de escala de código de hora está incorreta ou que a duração do cluster está muito longa, o que representa que o código de hora do quadro excede o espaço assinado de 16 bits.

Código	Message	Descrição/Ação recomendada
0x3200000d	STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA	Um código de início de Annex-B inválido foi encontrado. Por exemplo, o sinalizador de adaptação de Annex-B foi especificado e o código encontra uma sequência inicial inválida superior a três zeros. Um formato Annex-B deve ter uma sequência de "prevenção de emulação" para ignorar uma sequência de três ou mais zeros no bytestream. Para obter mais informações, consulte a especificação de MPEG. Para obter informações sobre esse erro no Android, consulte Erro STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) no Android .
0x3200000e	STATUS_MKV_INVALID _AVCC_NALU_IN_FRAM E_DATA	Embalagem AVCC NALU inválida quando o sinalizador AVCC de adaptação é especificado. Verifique se o bytestream está em um formato AVCC válido. Para obter mais informações, consulte a especificação de MPEG.

Código	Message	Descrição/Ação recomendada
0x3200000f	STATUS_MKV_BOTH_AN NEXB_AND_AVCC_SPEC IFIED	Tanto a adaptação dos NALUs do AVCC quanto do Anexo B foram especificados. Especifique qualquer um ou nenhum deles.
0x32000010	STATUS_MKV_INVALID _ANNEXB_NALU_IN_CPD	Formato Annex-B inválido dos CPD quando o sinalizador de adaptação de Annex-B é especificado. Verifique se o CPD está no formato válido do Anexo B. Se não estiver, remova o sinalizador de adaptação do Anexo B do CPD.
0x32000011	STATUS_MKV_PTS_DTS _ARE_NOT_SAME	O Kinesis Video Streams impõe que o PTS (timestamp de apresentação) e o DTS (timestamp de decodificação) sejam os mesmos para os quadros iniciais do fragmento. São os quadros-chave que iniciam o fragmento.
0x32000012	STATUS_MKV_INVALID _H264_H265_CPD	Falha ao analisar dados privados do codec H264/H265.
0x32000013	STATUS_MKV_INVALID _H264_H265_SPS_WID TH	Falha ao extrair a largura dos dados privados do codec.
0x32000014	STATUS_MKV_INVALID _H264_H265_SPS_HEI GHT	Falha ao extrair a altura dos dados privados do codec.

Código	Message	Descrição/Ação recomendada
0x32000015	STATUS_MKV_INVALID_H264_H265_SPS_NALU	NALU H264/H265 SPS inválido.
0x32000016	STATUS_MKV_INVALID_BIH_CPD	Formato de cabeçalho de informações de bitmap inválido nos dados privados do codec.
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	Contagem de unidades da Camada de abstração de rede (NALU – Network Abstraction Layer units) de Codificação de Vídeo de Alta Eficiência (HEVC – High Efficiency Video Coding) inválida.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	Formato de HEVC inválido.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	Faltam NALUs de HEVC no Conjunto de parâmetros de sequência (SPS – Sequence Parameter Set).
0x3200001a	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	Tamanho de NALU do SPS de HEVC inválido.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	IDC do formato do Chroma inválido.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	SPS reservado para HEVC inválido

Código	Message	Descrição/Ação recomendada
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	Tamanho mínimo AnnexBb do valor beta privado do codec. Para H264, esse valor deve ser igual ou maior que 11. Para H265, esse valor deve ser igual ou maior que 15.
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Faltam dados privados de codec em NALUs do Annex-B.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Beta privada de codec em NALUs do Annex-B inválida.
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	Tamanho de nome de tag inválido. O valor válido é maior que zero e menor que 128.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	Tamanho de valor de tag inválido. O valor válido é maior que zero e menor que 256.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	Tags de estado do gerador inválidas.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	Índice de frequência de amostragem de dados privados do codec AAC inválido.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	Configuração do canal de dados privados do codec AAC inválida.
0x32000025	STATUS_MKV_INVALID_AAC_CPD	Dados privados do codec AAC inválidos.

Código	Message	Descrição/Ação recomendada
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	Informações da trilha não encontradas.
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	UUID do segmento inválido.
0x32000028	STATUS_MKV_INVALID_TRACK_UID	UID da trilha inválido.
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 erro e status retornados pela biblioteca Trace

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na Trace biblioteca.

Código	Message
0x10100001	STATUS_MIN_PROFILER_BUFFER

Códigos de erro e status retornados pela biblioteca Utils

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na Utils biblioteca.

Código	Message
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED

Código	Message
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	Message
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 erro e status retornados pela biblioteca View

A tabela a seguir contém informações de erro e status que são retornadas pelos métodos na View biblioteca.

Código	Message	Descrição
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	Uma contagem de itens de exibição de conteúdo inválida foi especificada. Para ter mais informações, consulte Limites do SDK do produtor .
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	Uma duração de exibição de conteúdo inválida foi especificada. Para ter mais informações, consulte Limites do SDK do produtor .

Código	Message	Descrição
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	Foi feita uma tentativa de ultrapassar a posição de cabeça.
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	Um índice inválido foi especificado.

Código	Message	Descrição
0x30000005	STATUS_CONTENT_VIEW_INVALID_TIMESTAMP	<p>Houve um time stamp inválido ou uma sobreposição de time stamp. O timestamp de decodificação do quadro deve ser maior ou igual ao timestamp do quadro anterior, mais a duração do quadro anterior: $\text{DTS}(n) \geq \text{DTS}(n-1) + \text{Duration}(n-1)$. Em geral, esse erro indica um codificador "instável". O codificador produz uma intermitência de quadros codificados, e seus time stamps são menores do que as durações entre quadros. Ou o streaming é configurado para usar time stamps do SDK, e os quadros são enviados mais rapidamente do que as durações de quadros. Para ajudar em uma "oscilação" no codificador, especifique uma duração de quadros menor na estrutura <code>StreamInfo.StreamCaps</code>. Por exemplo, se o fluxo for de 25 FPS, a duração de cada quadro será de 40 ms. No entanto, para lidar com a "instabilidade" do codificador, recomendamos que você use metade da duração do quadro (20 ms). Alguns streamings</p>

Código	Message	Descrição
		exigem controle mais preciso sobre o tempo para detecção de erros.
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	Um tamanho de dados de itens de exibição de conteúdo inválido foi especificado.

Erros e códigos de status retornados por retornos de PutFrame chamada - biblioteca de produtores C

A seção a seguir contém informações de erro e status que são retornadas por retornos de chamada para a PutFrame operação na biblioteca do produtor C.

Código	Message	Descrição	Ação recomendada
0x15000001	STATUS_STOP_CALLBACK_CHAIN	O retorno de chamada foi interrompido.	
0x15000002	STATUS_MAXIMUM_CALLBACK_CHAIN	O valor máximo da cadeia de retorno de chamada foi atingido.	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	Versão da estrutura PlatformCallbacks inválida.	Especifique a versão atual da estrutura.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Versão da estrutura ProducerCallbacks inválida.	Especifique a versão atual da estrutura.

Código	Message	Descrição	Ação recomendada
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Versão da estrutura StreamCallbacks inválida.	Especifique a versão atual da estrutura.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Versão da estrutura AuthCallbacks inválida.	Especifique a versão atual da estrutura.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	Versão da estrutura ApiCallbacks inválida.	Especifique a versão atual da estrutura.
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Versão da estrutura AwsCredentials inválida.	Especifique a versão atual da estrutura.
0x15000009	STATUS_MAX_REQUEST_HEADER_COUNT	A contagem máxima de cabeçalho de solicitação foi atingida.	
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	O tamanho máximo do nome de cabeçalho de solicitação foi atingido.	

Código	Message	Descrição	Ação recomendada
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	O tamanho máximo de valor de cabeçalho de solicitação foi atingido.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	JSON de retorno inválido para uma chamada de API.	
0x1500000d	STATUS_CURL_INIT_FAILED	Falha na inicialização Curl.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Falha na inicialização curl lib.	
0x1500000f	STATUS_INVALID_DESCRIBE_STREAM_RETURN_JSON	JSON de retorno inválido para DescribeStream	
0x15000010	STATUS_HMAC_GENERATION_ERROR	Erro de geração de HMAC.	
0x15000011	STATUS_IOT_FAILED	Falha na autorização de IoT.	
0x15000012	STATUS_MAX_ROLE_ALIAS_LEN_EXCEEDED	O tamanho máximo do alias da função foi atingido.	Especifique um alias de tamanho menor.

Código	Message	Descrição	Ação recomendada
0x15000013	STATUS_MAX_USER_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED	O tamanho máximo do postfix do nome do agente foi atingido.	
0x15000014	STATUS_MAX_CUSTOM_USER_AGENT_LEN_EXCEEDED	O tamanho máximo do agente de usuário do cliente foi atingido.	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	Tamanho do agente de usuário inválido.	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	Período de armazenamento em cache de endpoint inválido.	Especifique um período de armazenamento em cache menor do que 24 horas.
0x15000017	STATUS_IOT_EXPIRATION_OCCURRED_IN_PAST	O timestamp de expiração da IoT ocorre no passado.	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	A análise da expiração da IoT falhou.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNCTION		

Código	Message	Descrição	Ação 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_MAXIMUM_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		

Código	Message	Descrição	Ação 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		
0x16000001	STATUS_CURL_PERFORMANCE_FAILED		

Código	Message	Descrição	Ação recomendada
0x16000002	STATUS_IO T_INVALID _RESPONSE_LENGTH		
0x16000003	STATUS_IO T_NULL_AWS_CREDS		
0x16000004	STATUS_IO T_INVALID _URI_LEN		
0x16000005	STATUS_TI MESTAMP_S TRING_UNR ECOGNIZED_FORMAT		

Referência de sinalizadores de adaptação do Network Abstraction Layer (NAL)

Esta seção contém informações sobre sinalizadores disponíveis para a enumeração do `StreamInfo.NalAdaptationFlags`.

O [stream elementar](#) em um aplicativo pode estar no formato Annex-B ou AVCC:

- O formato Annex-B delimita as [NALUs \(unidades de camada de abstração de rede\)](#) com dois bytes de zeros, seguidos de um ou três de zeros e do número 1 (chamado de código inicial; por exemplo, 00000001).
- O formato AVCC também encapsula as NALUs, mas toda NALU é precedida de um valor que indica seu tamanho (normalmente, quatro bytes).

Muitos codificadores produzem o formato de streaming de bits Annex-B. Alguns processadores de fluxo de bits de nível superior (como um mecanismo de reprodução ou o player [Media Source Extensions \(MSE\) no AWS Management Console](#)) usam o formato AVCC para seus quadros.

Os dados privados do codec (CPD), que são SPS/PPS (Sequence Parameter Set/Picture Parameter Set) para o codec H.264, também podem estar no formato Annex-B ou AVCC. No entanto, no caso de CPD, os formatos são diferentes dos descritos anteriormente.

Os sinalizadores informam o SDK para ajustar o NALUs para AVCC ou Annex-B para dados de quadros e CPD da seguinte forma:

Sinalizador	Adaptação
NAL_ADAPTATION_FLAG_NONE	Sem adaptação.
NAL_ADAPTATION_ANNEXB_NALS	Adapte as NALUs do Anexo B às NALUs AVCC.
NAL_ADAPTATION_AVCC_NALS	Adapte as NALUs AVCC às NALUs do Anexo B.
NAL_ADAPTATION_ANNEXB_CPD_NALS	Adapte os NALUs do Anexo B para os dados privados do codec para os NALUs no formato AVCC.
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	Adapte os NALUs do Anexo B para o codec e enquadre os dados privados em NALUs no formato AVCC.

Para obter mais informações sobre tipos de NALU, consulte a Seção 1.3: Tipos de unidade de camada de abstração de rede no [RFC 3984](#).

Estruturas do SDK do produtor

Esta seção inclui informações sobre estruturas que você pode usar para fornecer dados ao objeto Kinesis Video Streams Producer.

Tópicos

- [DeviceInfo/DefaultDeviceInfoProvider](#)

- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

Os `DefaultDeviceInfoProvider` objetos `DeviceInfo` controlam o comportamento do objeto `Kinesis Video Streams Producer`.

Campos de membros

- `versão` — Um valor inteiro usado para garantir que a versão correta da estrutura seja usada com a versão atual da base de código. A versão atual é especificada usando a macro `DEVICE_INFO_CURRENT_VERSION`.
- `nome` — O nome legível por humanos para o dispositivo.
- `Contagem de tags/tags` — Não usado atualmente.
- `StreamCount` — O número máximo de streams que o dispositivo pode suportar. Inicialmente, isso pré-aloca o armazenamento dos ponteiros aos objetos de streaming, mas os objetos de streaming reais são criados posteriormente. O padrão é 16 streamings, mas isso pode ser alterado no arquivo `DefaultDeviceInfoProvider.cpp`.
- `storageInfo`: um objeto que descreve a configuração do armazenamento principal. Para ter mais informações, consulte [StorageInfo](#).

StorageInfo

Especifica a configuração do armazenamento principal para o `Kinesis Video Streams`.

A implementação padrão é baseada em uma implementação de heap rápido de baixa fragmentação, que é otimizada para streaming. Ela usa o alocador `MEMALLOC`, que pode ser substituído em uma plataforma determinada. Algumas plataformas têm alocação de memória virtual sem suporte à alocação com páginas físicas. À medida que a memória for usada, as páginas virtuais serão baseadas nas páginas físicas. Isso resultará em pressão de memória insuficiente em todo o sistema quando o armazenamento for subutilizado.

Calcule o tamanho do armazenamento padrão com base na fórmula a seguir. O `DefragmentationFactor` deve ser definido como 1,2 (20 por cento).

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

No exemplo a seguir, um dispositivo tem streamings de áudio e de vídeo. O streaming de áudio tem 512 amostras por segundo, com uma amostra média de 100 bytes. O streaming de vídeo tem 25 quadros por segundo, com uma média de 10.000 bytes. Cada fluxo tem 3 minutos de duração do buffer.

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

Se o dispositivo tiver mais memória disponível, recomendamos que você adicione mais memória ao armazenamento para evitar fragmentação severa.

Verifique se o tamanho do armazenamento é adequado para acomodar os buffers completos de todos os fluxos com alta complexidade de codificação (quando o tamanho do quadro é maior devido ao alto movimento) ou quando a largura de banda é baixa. Se o produtor atingir a pressão da memória, ele emitirá retornos de chamada de chamada de pressão de estouro de armazenamento (`StorageOverflowPressureFunc`). No entanto, quando não há memória disponível no armazenamento de conteúdo, ela descarta o quadro que está sendo enviado para o Kinesis Video Streams com um `STATUS_STORE_OUT_OF_MEMORY = 0x5200002e` e erro (`err`). Para ter mais informações, consulte [Códigos de erro e status retornados pela biblioteca cliente](#). Isso também poderá ocorrer se as confirmações (ACKs) do aplicativo não estiverem disponíveis ou se as ACKs persistidas forem atrasadas. Nesse caso, os buffers são preenchidos até a capacidade de “duração do buffer” antes que os quadros anteriores comecem a ser eliminados.

Campos de membros

- `version` — Um valor inteiro usado para garantir que a versão correta da estrutura seja usada com a versão atual da base de código.
- `StorageType` — `DEVICE_STORAGE_TYPE` uma enumeração que especifica o suporte subjacente e a implementação do armazenamento. Atualmente, o único valor suportado é `DEVICE_STORAGE_TYPE_IN_MEM`. Uma implementação futura oferecerá suporte ao `DEVICE_STORAGE_TYPE_HYBRID_FILE` indicando que o armazenamento retorna para o armazenamento de conteúdo baseado em arquivos.
- `StorageSize` — O tamanho do armazenamento em bytes a ser pré-alocado. A alocação mínima é de 10 MB, e a máxima é de 10 GB. (Isso será alterado na implementação futura do armazenamento de conteúdo com base em arquivos.)
- `SpillRatio` — Um valor inteiro que representa a porcentagem do armazenamento a ser alocado do tipo de armazenamento direto de memória (RAM), em oposição ao armazenamento secundário de estouro (armazenamento de arquivos). Não utilizado no momento.

- `rootDirectory`: o caminho para o diretório onde o armazenamento de conteúdo baseado em arquivos está localizado. Não utilizado no momento.

Estruturas de transmissão de vídeo do Kinesis

Você pode usar as estruturas a seguir para fornecer dados a uma instância de um stream de vídeo do Kinesis.

Tópicos

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

StreamDefinition/StreamInfo

O objeto `StreamDefinition` na camada C++ encapsula o objeto `StreamInfo` no Platform Independent Code (PIC - Código independente de plataforma) e fornece alguns valores padrão no construtor.

Campos de membros

Campo	Tipo de dados	Descrição	Valor padrão
<code>stream_name</code>	<code>string</code>	Nome de stream opcional. Para obter mais informações sobre o tamanho do nome do streaming, consulte Limites do SDK do produtor . Cada stream deve ter um nome exclusivo.	Se nenhum nome for especificado, ele será gerado aleatoriamente.
<code>retention_period</code>	<code>duration<uint64_t, ratio<3600>></code>	O período de retenção do streaming em segundos.	3.600 (uma hora)

Campo	Tipo de dados	Descrição	Valor padrão
		Especificar 0 indica que não há retenção.	
tags	<code>const map<string, string>*</code>	Um mapa de pares de chave-valor que contém as informações do usuário. Se o streaming já tiver um conjunto de tags, as novas tags serão acrescentadas ao conjunto de tags existente.	Sem tags
kms_key_id	<code>string</code>	O ID da AWS KMS chave a ser usado para criptografar o fluxo. Para ter mais informações, consulte Proteção de dados no Kinesis Video Streams .	O nome da chave KMS padrão (<code>aws/kinesis-video</code>).
tipo de streaming	Enumeração <code>STREAMING_TYPE</code>	O único valor aceito é <code>STREAMING_TYPE_REALTIME</code> .	
tipo de conteúdo	<code>string</code>	O formato do conteúdo do stream. O console do Kinesis Video Streams pode reproduzir conteúdo <code>video/h264</code> no formato.	<code>video/h264</code>

Campo	Tipo de dados	Descrição	Valor padrão
latência máxima	<code>duration<uint64_t, milli></code>	A latência máxima em milissegundos para o stream. O retorno de chamada de pressão de latência de streaming (se especificado) será chamado quando a duração do buffer exceder esse intervalo. A especificação de 0 indica que nenhum retorno de pressão de latência de stream será chamado.	<code>milliseconds::zero()</code>

Campo	Tipo de dados	Descrição	Valor padrão
fragment_duration	duration<uint64_t>	<p>A duração do fragmento desejada, em segundos.</p> <p>Esse valor é usado juntamente com o valor de <code>key_frame_fragmentation</code>. Se esse valor for <code>false</code>, o Kinesis Video Streams gerará fragmentos em um quadro-chave após o término dessa duração.</p> <p>Por exemplo, um streaming de áudio Advanced Audio Coding (AAC - Codificação de áudio avançada) tem cada quadro como um quadro chave. A especificação de <code>key_frame_fragmentation = false</code> fará com que a fragmentação aconteça em um quadro chave após o término desse período, resultando em fragmentos de dois segundos.</p>	2

Campo	Tipo de dados	Descrição	Valor padrão
escala de código de hora	<code>duration<uint64_t, milli></code>	<p>A escala de timecode de MKV em milissegundos, que especifica a granularidade dos timecodes dos quadros no cluster do MKV. O código de hora do quadro do MKV sempre é relativo ao início do cluster. O MKV usa um valor de 16 bits (0 a 32767) assinado para representar o timecode dentro do cluster (fragmento). Verifique se o timecode do quadro pode ser representado com a escala de timecode fornecida. O valor padrão da escala do timecode de 1 ms garante que o maior quadro que pode ser representado seja 32767 ms \approx 32 segundos. Isso está acima da duração máxima de fragmentos especificada em Cotas do serviço Kinesis Video</p>	1

Campo	Tipo de dados	Descrição	Valor padrão
		Streams , que é 10 segundos.	
key_frame_fragmentation	bool	Se fragmentos devem ser produzidos em um quadro chave. Se true, o SDK produzirá um início de fragmento sempre que houver um quadro chave. Se false, o Kinesis Video Streams espera pelo menos e produz um novo fragmento <code>fragment_duration</code> no quadro-chave que o segue.	true

Campo	Tipo de dados	Descrição	Valor padrão
<code>frame_timecodes</code>	<code>bool</code>	Determina se usará códigos de hora de quadros ou gerará carimbos de hora usando o retorno de hora atual. Muitos codificadores não produzem time stamps com os quadros. Portanto, especificar <code>false</code> esse parâmetro garante que os quadros tenham um registro de data e hora à medida que são inseridos no Kinesis Video Streams.	<code>true</code>

Campo	Tipo de dados	Descrição	Valor padrão
<code>absolute_fragment_times</code>	<code>bool</code>	O Kinesis Video Streams usa o MKV como mecanismo de empacotamento subjacente. A especificação do MKV é rigorosa quanto aos timecodes de quadros serem relativos ao início do cluster (fragmento). No entanto, os timecodes do cluster podem ser absolutos ou relativos em relação à hora de início do streaming . Se os time stamps forem relativos, a chamada de API do serviço PutMedia usará o time stamp opcional de início do streaming e ajustará os time stamps do cluster. O serviço sempre armazena os fragmentos com seus carimbos de data e hora absolutos.	<code>true</code>
<code>fragment_acks</code>	<code>bool</code>	Se deve receber ACKs (confirmações) de fragmentos de nível de aplicativo.	<code>true</code> , o que significa que o SDK receberá os ACKs e agirá adequadamente.

Campo	Tipo de dados	Descrição	Valor padrão
<code>restart_on_error</code>	<code>bool</code>	Determina se será reiniciado em erros específicos.	<code>true</code> , o que significa que o SDK tentará reiniciar o streaming se ocorrerem erros.
<code>recalculate_metrics</code>	<code>bool</code>	Se as métricas devem ser recalculadas. Cada chamada para recuperar as métricas pode recalculá-las para obter o valor "em execução" mais recente, o que pode criar um pequeno impacto na CPU. Talvez seja necessário o definir isso como <code>false</code> em dispositivos de ocupação de espaço/energia extremamente baixos para poupar os ciclos da CPU. Caso contrário, não recomendamos o <code>false</code> uso desse valor.	<code>true</code>

Campo	Tipo de dados	Descrição	Valor padrão
<code>nal_adaptation_flags</code>	<code>uint32_t</code>	Especifica os sinalizadores de adaptação da Network Abstraction Layer unit (NALU, unidade da Camada de abstração de rede). Se o bitstream for codificado em H.264, ele poderá ser processado como bruto ou empacotado em NALUs. Esses são em formato Annex-B ou AVCC. A maioria dos produtores e consumidores de streams elementares (codificadores e decodificadores de leitura) usa o formato Annex-B porque ele tem vantagens, como recuperação de erros. Os sistemas de nível superior usam o formato AVCC, que é o formato padrão para MPEG, HLS, DASH e assim por diante. A reprodução do console usa MSE (media source extensions -	O padrão é adaptar o formato Annex-B ao formato AVCC para os dados do quadro e para os dados privados do codec.

Campo	Tipo de dados	Descrição	Valor padrão
		<p>extensões de fonte de mídia) do navegador para decodificar e reproduzir o streaming que usa o formato AVCC. Para H.264 (e M-JPEG e H.265), o SDK fornece recursos de adaptação.</p> <p>Muitos streams elementares estão no seguinte formato. Neste exemplo, Ab é o código de início do Annex-B (001 ou 0001).</p> <div data-bbox="829 1079 1149 1476" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)... Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)</pre> </div> <p>No caso do H.264, os dados privados do codec (CPD) estão nos parâmetros SPS (conjunto de parâmetros de sequência) e</p>	

Campo	Tipo de dados	Descrição	Valor padrão
		<p>PPS (conjunto de parâmetros de imagem) e podem ser adaptados ao formato AVCC. A menos que o pipeline de mídia ofereça o CPD separadamente, o aplicativo pode extrair o CPD do quadro. Ele pode fazer isso procurando o primeiro quadro IDR (que deve conter o SPS e o PPS), extrair os dois NALUs (que estão <code>Ab(Sps)</code> <code>Ab(Pps)</code>) e configurá-lo no CPD. <code>StreamDefinition</code></p> <p>Para ter mais informações, consulte Sinalizadores de adaptação do NAL.</p>	
<code>frame_rate</code>	<code>uint32_t</code>	A taxa de quadros esperada. Esse valor é usado para calcular melhor as necessidades de armazenamento em buffer.	25

Campo	Tipo de dados	Descrição	Valor padrão
avg_bandwidth_bps	uint32_t	A largura de banda média esperada para o stream. Esse valor é usado para calcular melhor as necessidades de armazenamento em buffer.	4 * 1024 * 1024

Campo	Tipo de dados	Descrição	Valor padrão
<code>buffer_duration</code>	<code>duration<uint64_t></code>	A duração do buffer de streaming, em segundos. O SDK mantém os quadros no armazenamento de conteúdo por até <code>buffer_duration</code> , após o qual os quadros anteriores são eliminados à medida que a janela avança. Se o quadro que está sendo eliminado não tiver sido enviado para o back-end, o retorno de chamada do quadro descartado será chamado. Se a duração atual do buffer for maior que <code>max_latency</code> , o retorno de chamada de pressão de latência do streaming será chamado. O buffer é ajustado para o início do próximo fragmento quando a ACK do fragmento persistido é recebida. Isso indica que o conteúdo foi persistido de forma durável na	120

Campo	Tipo de dados	Descrição	Valor padrão
		nuvem, de forma que o armazenamento do conteúdo no dispositivo local não é mais necessário.	

Campo	Tipo de dados	Descrição	Valor padrão
replay_duration	duration<uint64_t>	A duração, em segundos, para reverter o leitor atual para reproduzir durante um erro, se a reinicialização estiver ativada. A reversão é interrompida no início do buffer (caso o streaming tenha acabado de iniciar ou a ACK persistida tiver ocorrido). A reversão tenta atingir um quadro chave que indica o início de um fragmento. Se o erro “causando a reinicialização” não for indicativo de um host inativo (o host ainda está ativo e contém os dados do quadro em seus buffers internos), a reversão é interrompida no último quadro ACK recebido. Em seguida, ele avança para o próximo quadro chave, pois o fragmento inteiro já está armazenado na memória do host.	40

Campo	Tipo de dados	Descrição	Valor padrão
connection_staleness	duration<uint64_t>	O tempo, em segundos, após o qual o retorno de chamada de inatividade do stream é chamado se o SDK não receber o ACK de buffer. Isso indica que os quadros estão sendo enviados do dispositivo, mas o back-end não os reconhece. Essa condição indica uma conexão rompida no salto intermediário ou no load balancer.	30
codec_id	string	O ID codec da trilha do MKV.	V_MPEG4/ISO/AVC
track_name	string	O nome da trilha do MKV.	kinesis_video

Campo	Tipo de dados	Descrição	Valor padrão
codecPrivateData	unsigned char*	O buffer de codec private data (CPD - dados privados do codec). Se o pipeline de mídia tiver as informações sobre o CPD antes do início do stream, ele poderá ser definido em <code>StreamDefinition.codecPrivateData</code> . Os bits são copiados, e o buffer pode ser reutilizado ou liberado depois que a chamada <code>createStreaming</code> for criada. No entanto, se os dados não estiverem disponíveis quando o fluxo for criado, eles poderão ser definidos em uma das sobrecargas da <code>KinesisVideoStream.start(cpd)</code> função.	null
codecPrivateDataTamanho	uint32_t	O tamanho do buffer dos dados privados do codec.	0

ClientMetrics

O ClientMetrics objeto é preenchido por meio de uma chamada `getKinesisVideoMetrics`.

Campos de membros

Campo	Tipo de dados	Descrição
<code>version</code>	UINT32	A versão da estrutura, definida na macro <code>CLIENT_METRICS_CURRENT_VERSION</code> .
<code>contentStoreSize</code>	UINT64	O tamanho total em bytes do armazenamento de conteúdo. Esse é o valor especificado em <code>DeviceInfo.StorageInfo.storageSize</code> .
<code>contentStoreAvailableTamanho</code>	UINT64	Tamanho atual do armazenamento disponível em bytes.
<code>contentStoreAllocatedTamanho</code>	UINT64	Tamanho atual alocado. Os tamanhos alocados mais os tamanhos disponíveis devem ser um pouco menores que o tamanho total do armazenamento, devido à contabilidade interna e à implementação do armazenamento de conteúdo.
<code>totalContentViewsTamanho</code>	UINT64	O tamanho da memória alocada para todas as visualizações de conteúdo de todos os streams. Isso não é contabilizado em relação ao tamanho do armazenamento. Essa memória é alocada usando a macro <code>MEMALLOC</code> ,

Campo	Tipo de dados	Descrição
		que pode ser substituída para fornecer um alocador personalizado.
totalFrameRate	UINT64	A taxa de quadros total observada em todos os streams.
totalTransferRate	UINT64	A taxa de stream total observada em bytes por segundo em todos os streams.

StreamMetrics

O `StreamMetrics` objeto é preenchido por meio de uma chamada `getKinesisVideoMetrics`.

Campos de membros

Campo	Tipo de dados	Descrição
version	UINT32	A versão da estrutura, definida na macro <code>STREAM_METRICS_CURRENT_VERSION</code> .
currentViewDuration	UINT64	A duração dos quadros acumulados. No caso de rede rápida, essa duração é zero ou a duração do quadro (à medida que o quadro está sendo transmitido). Se a duração for maior do que a <code>max_latency</code> especificada no <code>StreamDefinition</code> , o retorno de chamada de latência do stream será

Campo	Tipo de dados	Descrição
		chamado se for especificado. A duração é especificado em unidades de 100ns, que é a unidade de tempo padrão para a camada PIC.
overallViewDuration	UINT64	A duração geral da visualização. Se o stream for configurado sem ACKs ou persistência, esse valor aumentará à medida que os quadros forem inseridos no stream de vídeo do Kinesis e se tornarão iguais aos <code>buffer_duration</code> do <code>StreamDefinition</code> . Quando os ACKs são habilitados e o ACK persistente é recebido, o buffer é reduzido para o próximo quadro-chave. Isso ocorre porque o timestamp ACK indica o início de todo o fragmento. A duração é especificada em unidades de 100 ns, que é a unidade de tempo padrão para a camada PIC.
currentViewSize	UINT64	O tamanho em bytes do buffer atual.
overallViewSize	UINT64	O tamanho geral da visualização em bytes.
currentFrameRate	UINT64	A taxa de quadros observada para o stream atual.

Campo	Tipo de dados	Descrição
currentTransferRate	UINT64	A taxa de transferência observada em bytes por segundo para o stream atual.

Retornos de chamada do SDK do produtor

As classes e métodos no SDK do Amazon Kinesis Video Streams Producer não mantêm seus próprios processos. Em vez disso, eles usam as chamadas e eventos da função de entrada para programar retornos de chamada para se comunicar com o aplicativo.

Há dois padrões de retorno de chamada que o aplicativo pode usar para interagir com o SDK:

- [CallbackProvider](#)— Esse objeto expõe cada retorno de chamada do componente de código independente da plataforma (PIC) para o aplicativo. Esse padrão permite funcionalidade total, mas também significa que a implementação deve tratar todos os métodos de API pública e assinaturas na camada do C++.
- [StreamCallbackProvider](#) e [ClientCallbackProvider](#) — Esses objetos expõem os retornos de chamada específicos do stream e do cliente, e a camada C++ do SDK expõe o restante dos retornos de chamada. Esse é o padrão de retorno preferencial para interagir com o Producer SDK.

O diagrama a seguir ilustra o modelo de objeto dos objetos de retorno de chamada:

No diagrama anterior, `DefaultCallbackProvider` deriva de `CallbackProvider` (que expõe todos os retornos de chamada no PIC) e contém `StreamCallbackProvider` e `ClientCallbackProvider`.

Este tópico contém as seguintes seções:

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks estrutura](#)
- [Implementações de retorno de chamada para repetir o streaming](#)

ClientCallbackProvider

O objeto `ClientCallbackProvider` expõe as funções de retorno no nível do cliente. Os detalhes das funções são descritos na seção [ClientCallbacks](#).

Métodos de retorno de chamada:

- `getClientReadyCallback`— Relata um estado pronto para o cliente.
- `getStorageOverflowPressureCallback`— Relata sobrecarga ou pressão de armazenamento. Esse retorno de chamada é feito quando a utilização do armazenamento fica abaixo do valor de `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD`, que é de 5% do tamanho total do armazenamento. Para ter mais informações, consulte [StorageInfo](#).

StreamCallbackProvider

O objeto `StreamCallbackProvider` expõe as funções de retorno no nível do streaming.

Métodos de retorno de chamada:

- `getDroppedFragmentReportCallback`: relata um fragmento descartado.
- `getDroppedFrameReportCallback`— Relata um quadro perdido.
- `getFragmentAckReceivedCallback`— Relata que um fragmento ACK foi recebido para o fluxo.
- `getStreamClosedCallback`— Relata uma condição de riacho fechado.
- `getStreamConnectionStaleCallback`— Relata uma condição de conexão obsoleta. Nessa condição, o produtor está enviando dados para o serviço, mas não está recebendo confirmações.
- `getStreamDataAvailableCallback`— Relata que os dados estão disponíveis no stream.
- `getStreamErrorReportCallback`— Relata uma condição de erro de transmissão.
- `getStreamLatencyPressureCallback`— Relata uma condição de latência do stream, que ocorre quando o tamanho do buffer acumulado é maior que o `max_latency` valor. Para ter mais informações, consulte [StreamDefinition/StreamInfo](#).
- `getStreamReadyCallback`: —Relata uma condição de prontidão do fluxo.
- `getStreamUnderflowReportCallback`— Relata uma condição de subfluxo do riacho. Essa função não é usada atualmente e está reservada para uso futuro.

Para obter o código-fonte de `StreamCallbackProvider`, consulte [StreamCallbackProvider.h](#).

ClientCallbacks estrutura

A estrutura de `ClientCallbacks` contém os pontos de entrada da função de retorno de chamada que o PIC chama quando ocorrerem eventos específicos. A estrutura também contém informações sobre a versão no campo `CALLBACKS_CURRENT_VERSION` e um campo `customData` de dados definidos pelo usuário que é retornado com funções de retorno de chamada individuais.

O aplicativo cliente pode usar um ponteiro `this` para o campo `custom_data` para mapear as funções de membro para as funções `ClientCallback` estáticas em tempo de execução, conforme mostrado no seguinte código de exemplo:

```
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

Função	Descrição	Tipo
<code>CreateDeviceFunc</code>	Não implementado no back-end atualmente. Essa chamada falha quando chamada do Java ou do C++. Outros clientes executam inicialização específica à plataforma.	API de back-end
<code>CreateStreamFunc</code>	Chamada quando o streaming é criado.	API de back-end
<code>DescribeStreamFunc</code>	Chamada quando <code>DescribeStream</code> é chamado.	API de back-end
<code>GetStreamingEndpointFunc</code>	Chamada quando <code>GetStreamingEndpoint</code> é chamado.	API de back-end

Função	Descrição	Tipo
<code>GetStreamingTokenFunc</code>	Chamada quando <code>GetStreamingToken</code> é chamado.	API de back-end
<code>PutStreamFunc</code>	Chamada quando <code>PutStream</code> é chamado.	API de back-end
<code>TagResourceFunc</code>	Chamada quando <code>TagResource</code> é chamado.	API de back-end
<code>CreateMutexFunc</code>	Cria um mutex de sincronização.	Sincronização
<code>FreeMutexFunc</code>	Libera o mutex.	Sincronização
<code>LockMutexFunc</code>	Bloqueia o mutex de sincronização.	Sincronização
<code>TryLockMutexFunc</code>	Tenta bloquear o mutex. Não implementado no momento.	Sincronização
<code>UnlockMutexFunc</code>	Desbloqueia o mutex.	Sincronização
<code>ClientReadyFunc</code>	Chamada quando o cliente entra em estado de pronto.	Notificação
<code>DroppedFrameReportFunc</code>	Relata quando um quadro é descartado.	Notificação
<code>DroppedFragmentReportFunc</code>	Relata quando um fragmento é descartado. Essa função não é usada atualmente e está reservada para uso futuro.	Notificação

Função	Descrição	Tipo
<code>FragmentAckReceiveFunc</code>	Chamada quando uma ACK (armazenamento em buffer, recebida, persistida e erro) de fragmento é recebida.	Notificação
<code>StorageOverflowPressureFunc</code>	Chamada quando a utilização do armazenamento fica abaixo do valor de <code>STORAGE_PRESSURE_NOTIFICATION_THRESHOLD</code> , que é definido como 5% do tamanho total do armazenamento.	Notificação
<code>StreamClosedFunc</code>	Chamada quando os últimos bits dos quadros restantes são transmitidos.	Notificação
<code>StreamConnectionStaleFunc</code>	Chamada quando o streaming entra em um estado de conexão obsoleta. Nessa condição, o produtor está enviando dados ao serviço, mas não está recebendo confirmações.	Notificação
<code>StreamDataAvailableFunc</code>	Chamada quando os dados do streaming estão disponíveis.	Notificação
<code>StreamErrorReportFunc</code>	Chamada quando ocorre um erro de streaming. O PIC fecha automaticamente o streaming sob essa condição.	Notificação

Função	Descrição	Tipo
<code>StreamLatencyPressureFunc</code>	Chamada quando o streaming entra em uma condição de latência, que é quando o tamanho do buffer acumulado é maior que o valor de <code>max_latency</code> . Para ter mais informações, consulte StreamDefinition/StreamInfo .	Notificação
<code>StreamReadyFunc</code>	Chamada quando o streaming entra em estado de pronto.	Notificação
<code>StreamUnderflowReportFunc</code>	Essa função não é usada atualmente e está reservada para uso futuro.	Notificação
<code>DeviceCertToTokenFunc</code>	Retorna o certificado da conexão como um token.	Integração de plataformas
<code>GetCurrentTimeFunc</code>	Retorna a hora atual.	Integração de plataformas
<code>GetDeviceCertificateFunc</code>	Retorna o certificado do dispositivo. Essa função não é usada atualmente e está reservada para uso futuro.	Integração de plataformas
<code>GetDeviceFingerprintFunc</code>	Retorna a impressão digital do dispositivo. Essa função não é usada atualmente e está reservada para uso futuro.	Integração de plataformas
<code>GetRandomNumberFunc</code>	Retorna um número aleatório entre 0 e <code>RAND_MAX</code> .	Integração de plataformas

Função	Descrição	Tipo
GetSecurityTokenFunc	Retorna o token de segurança que é passado para as funções que se comunicam com a API de back-end. A implementação pode especificar os <code>AccessKeyId</code> , <code>SecretKeyId</code> e o token de sessão serializados.	Integração de plataformas
LogPrintFunc	Registra em log uma linha de texto com a tag e o nível do log. Para ter mais informações, consulte <code>PlatformUtils.h</code> .	Integração de plataformas

Para as funções de integração de plataformas na tabela anterior, o último parâmetro é uma estrutura `ServiceCallContext` que tem os seguintes campos:

- `version`: a versão do struct.
- `callAfter`: um horário absoluto após o qual chamar a função.
- `timeout`: o tempo limite da operação em 100 unidades de nanossegundo.
- `customData`: um valor definido pelo usuário a ser passado de volta para o cliente.
- `pAuthInfo`: as credenciais para a chamada. Para obter mais informações, consulte a estrutura (`__AuthInfo`) a seguir.

As informações de autorização são fornecidas usando a estrutura `__AuthInfo`, que podem ser credenciais serializadas ou um token de autenticação específico ao provedor. Esta estrutura tem os seguintes campos:

- `version`: a versão da estrutura de `__AuthInfo`.
- `type`: um valor de `AUTH_INFO_TYPE` que define o tipo de credencial (certificado ou token de segurança).
- `data`: uma matriz de bytes que contém as informações de autenticação.

- `size`: o tamanho do parâmetro `data`.
- `expiration`: a expiração das credenciais em 100 unidades de nanossegundo.

Implementações de retorno de chamada para repetir o streaming

O Producer SDK do Kinesis Video fornece o status de streaming por meio de funções de retorno de chamada. Recomendamos que você implemente os seguintes mecanismos de retorno de chamada para se recuperar de quaisquer problemas momentâneos de rede encontrados durante o streaming.

- Callback de pressão de latência de stream — esse mecanismo de callback é iniciado quando o SDK encontra uma condição de latência de stream. Isso acontece quando o tamanho do buffer acumulado é maior que o valor de `MAX_LATENCY`. Quando o streaming é criado, o aplicativo de streaming define `MAX_LATENCY` para o valor padrão de 60 segundos. A implementação típica para esse retorno de chamada é a redefinição da conexão. Você pode usar o exemplo de implementação em <https://github.com/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.c> conforme necessário. Observe que não há opção de armazenar os quadros não entregues devido à interrupção da rede em um armazenamento secundário para preenchimento.
- Retorno de chamada de inatividade do stream — esse retorno de chamada é iniciado quando o produtor pode enviar dados para o serviço Amazon Kinesis Data Streams (uplink), mas não consegue recuperar as confirmações (ACK em buffer) a tempo (o padrão é 60 segundos). Dependendo das configurações de rede, o retorno de chamada de pressão de latência do fluxo ou o retorno de chamada de inatividade do fluxo, ou ambos, podem ser iniciados. Semelhante à implementação de nova tentativa do retorno de chamada de pressão de latência do streaming, a implementação típica é a redefinição da conexão e o início de uma nova conexão para streaming. Você pode usar o exemplo de implementação em <https://github.com/awslabs/amazon-kinesis-video-streams-connection-state-machine-producer-c/blob/master/src/source/> conforme necessário.
- Retorno de chamada de erro de transmissão - esse retorno de chamada é iniciado quando o SDK encontra um tempo limite na conexão de rede ou outros erros durante a chamada para as chamadas do serviço da API KVS.
- Retorno de chamada de quadro perdido - esse retorno de chamada é iniciado quando o tamanho do armazenamento está cheio devido à baixa velocidade da rede ou a um erro de transmissão. Se a velocidade da rede resultar em queda de quadros, você poderá aumentar o tamanho do armazenamento, reduzir o tamanho do quadro do vídeo ou a taxa de quadros para corresponder à velocidade da rede.

Biblioteca de análise de streaming de vídeo do Kinesis

A Kinesis Video Stream Parser Library é um conjunto de ferramentas que você pode usar em aplicativos Java para consumir os dados MKV em um stream de vídeo do Kinesis.

A biblioteca inclui as seguintes ferramentas:

- [StreamingMkvReader](#): esta classe lê os elementos do MKV especificados em um streaming de vídeo.
- [FragmentMetadataVisitor](#): esta classe recupera metadados de fragmentos (elementos de mídia) e de trilhas (streams de dados individuais contendo informações de mídia, como áudio ou legendas).
- [OutputSegmentMerger](#): esta classe mescla fragmentos ou partes consecutivas em um streaming de vídeo.
- [KinesisVideoExample](#): este é um aplicativo de exemplo que mostra como usar a Biblioteca de Análise de Streams de Vídeo Kinesis.

A biblioteca também inclui testes que mostram como as ferramentas são usadas.

Procedimento: Usar a biblioteca de análise de streaming de vídeo Kinesis

Este procedimento inclui as seguintes etapas:

- [the section called “Etapa 1: Baixe e configure o código”](#).
- [the section called “Etapa 2: escrever e examinar o código”](#).
- [the section called “Etapa 3: executar e verificar o código”](#).

Pré-requisitos

Você deve ter o seguinte para examinar e usar a Biblioteca de Análise de Streams de Vídeo da Kinesis:

- Uma conta da Amazon Web Services (AWS). Se você ainda não tem um Conta da AWS, veja [the section called “Inscreva-se para um Conta da AWS”](#).

- [Um ambiente de desenvolvimento integrado \(IDE\) Java, como Eclipse Java Neon ou JetBrains IntelliJ Idea.](#)
- Java 11, como o [Amazon Corretto](#) 11.

Etapa 1: Baixe e configure o código

Nesta seção, você vai fazer download da biblioteca Java, testar o código e importar o projeto para o IDE do Java.

Para ver os pré-requisitos e outros detalhes sobre este procedimento, consulte [Biblioteca do analisador de streaming](#).

1. Crie um diretório e clone o código-fonte da biblioteca do GitHub repositório (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>).

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Abra o Java IDE que você está usando (por exemplo, [Eclipse](#) ou [IntelliJ IDEA](#)) e importe o projeto Apache Maven que você baixou:
 - No Eclipse: escolha File, Import, Maven, Existing Maven Projects e navegue até a pasta `kinesis-video-streams-parser-lib`.
 - No IntelliJ Idea: escolha Import. Navegue até o arquivo `pom.xml` na raiz do pacote que você obteve por download.

Para obter mais informações, consulte a documentação relacionada ao IDE.

Próxima etapa

[the section called “Etapa 2: escrever e examinar o código”.](#)

Etapa 2: escrever e examinar o código

Nesta seção, você examinará a biblioteca Java e o código de teste e aprenderá a usar as ferramentas da biblioteca no seu próprio código.

A biblioteca de análise de streaming de vídeo do Kinesis contém as seguintes ferramentas:

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

StreamingMkvReader

Essa classe lê os elementos do MKV especificados a partir de um streaming em uma forma sem bloqueio.

O exemplo de código a seguir (de `FragmentMetadataVisitorTest`) mostra como criar e usar `StreamingMkvReader` para recuperar `MkvElement` objetos de um streaming de entrada chamado `inputStream`:

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

FragmentMetadataVisitor

Essa classe recupera metadados para fragmentos (elementos de mídia) e rastreia fluxos de dados individuais contendo informações de mídia, como dados privados do codec, largura ou altura de pixels.

O código de exemplo a seguir (do arquivo `FragmentMetadataVisitorTest`) mostra como usar `FragmentMetadataVisitor` para recuperar dados de um 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++;
                }
            }
        }
    }
}

```

O exemplo anterior mostra o padrão de codificação a seguir:

- Crie uma classe `FragmentMetadataVisitor` para analisar os dados e uma [StreamingMkvReader](#) para fornecer os dados.
- Em cada `MkvElement` no streaming, teste se os metadados são do tipo `SIMPLEBLOCK`.
- Se forem, recupere o `MkvDataElement` do `MkvElement`.
- Recupere o `Frame` (dados de mídia) do `MkvDataElement`.
- Recupere os `MkvTrackMetadata` do `Frame` do `FragmentMetadataVisitor`.
- Recupere e verifique os seguintes dados dos objetos `Frame` e `MkvTrackMetadata`:

- O número de controle.
- A altura do pixel de quadros.
- A profundidade do pixel de quadros.
- O ID do codec usado para codificar o quadro.
- Que este quadro foi recebido em ordem. Verifique se o número da faixa do quadro anterior, se presente, é menor que o do quadro atual.

Para usar `FragmentMetadataVisitor` no seu projeto, passe os objetos `MkvElement` para o visitante usando o método `accept`:

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

Esta classe combina metadados de diferentes trilhas no streaming em um streaming com um segmento único.

O exemplo de código a seguir (do arquivo `FragmentMetadataVisitorTest`) mostra como usar `OutputSegmentMerger` para mesclar metadados de uma matriz de byte chamada `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
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetadata().getTypeInfo()))
{
    MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
    Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
    Assert.assertTrue(frame.getFrameData().limit() > 0);
    MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
    assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
}
}
```

O exemplo anterior mostra o padrão de codificação a seguir:

- Crie um [FragmentMetadataVisitor](#) para recuperar os metadados do stream.
- Criar um streaming de saída para receber os metadados mesclados.
- Crie um `OutputSegmentMerger`, passando o `ByteArrayOutputStream`.
- Crie um `CompositeMkvElementVisitor` que contenha os dois visitantes.
- Crie um `InputStream` que aponte para o arquivo especificado.
- Mescle cada elemento nos dados de entrada no streaming de saída.

KinesisVideoExample

Este é um aplicativo de exemplo que mostra como usar a Biblioteca de análise de streaming de vídeo Kinesis.

Essa classe executa as seguintes operações:

- Cria um stream de vídeo do Kinesis. Se um stream com o nome fornecido já existe, o stream é excluído e recriado.
- Chamadas [PutMedia](#) para transmitir fragmentos de vídeo para o stream de vídeo do Kinesis.
- Chamadas [GetMedia](#) para transmitir fragmentos de vídeo do stream de vídeo do Kinesis.
- Usa uma [StreamingMkvReader](#) para analisar os fragmentos retornados no stream, e usa um [FragmentMetadataVisitor](#) para registrar os fragmentos.

Excluir e recriar o stream

O exemplo de código a seguir (do `StreamOps.java` arquivo) exclui um determinado stream de vídeo do Kinesis:

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

O exemplo de código a seguir (do `StreamOps.java` arquivo) cria um stream de vídeo do Kinesis com o nome especificado:

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

Ligue PutMedia

O exemplo de código a seguir (do `PutMediaWorker.java` arquivo) faz chamadas [PutMedia](#) no stream:

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

Ligue GetMedia

O exemplo de código a seguir (do `GetMediaWorker.java` arquivo) faz chamadas [GetMedia](#) no stream:

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

Analise o resultado GetMedia

Esta seção descreve como usar [StreamingMkvReader](#), [FragmentMetadataVisitor](#) e [CompositeMkvElementVisitor](#) para analisar, salvar no arquivo e registrar os dados retornados de `GetMedia`.

Leia a saída de GetMedia com StreamingMkvReader

O exemplo de código a seguir (do `GetMediaWorker.java` arquivo) cria um [StreamingMkvReader](#) e o usa para analisar o resultado da [GetMedia](#) operação:

```
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);
}
```

No exemplo de código anterior, o [StreamingMkvReader](#) recupera objetos `MkvElement` da carga do resultado `GetMedia`. Na próxima seção, os elementos são passados para um [FragmentMetadataVisitor](#).

Recupere fragmentos com FragmentMetadataVisitor

Os exemplos de código a seguir (dos arquivos `KinesisVideoExample.java` e `StreamingMkvReader.java`) criam um [FragmentMetadataVisitor](#). Os objetos `MkvElement` iteradas pelo [StreamingMkvReader](#) são passados para o visitante usando o 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 elementos e gravação em um arquivo

O exemplo de código a seguir (do arquivo `KinesisVideoExample.java`) cria os seguintes objetos e os retorna como parte do valor de retorno da função `GetMediaProcessingArguments`:

- Um `LogVisitor` (uma extensão de `MkvElementVisitor`) que grava no log do sistema.
- Um `OutputStream` que grava os dados de entrada em um MKV arquivo.
- Um `BufferedOutputStream` que acumula dados delimitados para o `OutputStream`.
- Um [the section called “OutputSegmentMerger”](#) que combina elementos consecutivos no resultado de `GetMedia` com os mesmos dados de controle e EBML.
- Um `CompositeMkvElementVisitor` que compõe o [FragmentMetadataVisitor](#), [the section called “OutputSegmentMerger”](#), e `LogVisitor` em um único elemento 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);
```

Os argumentos de processamento de mídia são então passados para `oGetMediaWorker`, que por sua vez é passado para `oExecutorService`, que executa o trabalhador em um encadeamento separado:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getMkvElementVisitor());
executorService.submit(getMediaWorker);
```

Próxima etapa

[the section called “Etapa 3: executar e verificar o código”](#)

Etapa 3: executar e verificar o código

A Kinesis Video Stream Parser Library contém ferramentas que devem ser usadas em seus próprios projetos. O projeto contém testes de unidade das ferramentas que você pode executar para verificar a instalação.

Os seguintes testes de unidade estão incluídos na biblioteca:

- mkv
 - ElementSizeAndOffsetVisitorTest
 - MkvValueTest
 - StreamingMkvReaderTest
- utilitários
 - FragmentMetadataVisitorTest
 - OutputSegmentMergerTest

Exemplos do Amazon Kinesis Video Streams

Os exemplos de código a seguir demonstram como trabalhar com a API Kinesis Video Streams:

Exemplos: envio de dados para o Kinesis Video Streams

- [Exemplo: Plugin GStreamer do SDK do Kinesis Video Streams Producer - kvssink](#): mostra como criar o SDK do Kinesis Video Streams Producer para usar como destino do GStreamer.
- [Execute o elemento GStreamer em um contêiner Docker](#): mostra como usar uma imagem pré-criada do Docker para enviar vídeo do Protocolo de Streaming em Tempo Real (RTSP) de uma câmera IP para o Kinesis Video Streams.
- [Exemplo: streaming de uma fonte RTSP](#): mostra como criar sua própria imagem do Docker e enviar vídeo RTSP de uma câmera IP para o Kinesis Video Streams.
- [Exemplo: envio de dados para o Kinesis Video Streams PutMedia usando a API](#): mostra como usar o para enviar dados [Usar a Biblioteca de produtores Java](#) para o Kinesis Video Streams que já estão em formato de contêiner (MKV) usando a API. [PutMedia](#)

Exemplos: recuperação de dados do Kinesis Video Streams

- [KinesisVideoExample](#): mostra como analisar e registrar fragmentos de vídeo usando a Biblioteca de Analisadores do Kinesis Video Streams.
- [Exemplo: análise e renderização de fragmentos do Kinesis Video Streams: mostra como analisar e renderizar fragmentos de stream de vídeo do Kinesis usando o JCodec e o JFrame.](#)

Exemplos: reprodução de dados de vídeo

- [Exemplo: Use HLS em HTML e JavaScript](#): mostra como recuperar uma sessão de streaming HLS para um stream de vídeo do Kinesis e reproduzi-la em uma página da web.

Pré-requisitos

- No código de exemplo, você fornece credenciais especificando um perfil definido no seu arquivo de perfil de AWS credenciais ou fornecendo credenciais nas propriedades do sistema Java do seu ambiente de desenvolvimento integrado (IDE). Se você ainda não fez isso, primeiro configure suas

credenciais. Para obter mais informações, consulte [Configurar AWS credenciais e região para desenvolvimento](#).

- Recomendamos que você use um Java IDE para visualizar e executar o código, como um dos seguintes:
 - [Eclipse Java Neon](#)
 - [JetBrains IntelliJ IDEA](#)

Exemplo: Plugin GStreamer do SDK do Kinesis Video Streams Producer - kvssink

Este tópico descreve como criar o SDK do Amazon Kinesis Video Streams Producer para usar como um plug-in do GStreamer.

Tópicos

- [Baixe, crie e configure o elemento GStreamer](#)
- [Execute o elemento GStreamer](#)
- [Exemplo de comandos de inicialização do GStreamer](#)
- [Execute o elemento GStreamer em um contêiner Docker](#)
- [Referência de parâmetros do elemento GStreamer](#)

O [GStreamer](#) é uma estrutura de mídia popular usada por várias câmeras e fontes de vídeo para criar canais de mídia personalizados combinando plug-ins modulares. O plug-in Kinesis Video Streams GStreams simplifica a integração do seu pipeline de mídia existente do GStreamer com o Kinesis Video Streams. Depois de integrar o GStreamer, você pode transmitir vídeo de uma webcam ou câmera Real Time Streaming Protocol (RTSP) para o Kinesis Video Streams para reprodução em tempo real ou posterior, armazenamento e análise adicional.

O plug-in GStreamer gerencia automaticamente a transferência do seu stream de vídeo para o Kinesis Video Streams encapsulando a funcionalidade fornecida pelo SDK do Kinesis Video Streams Producer em um elemento coletor do GStreamer, `kvssink`. A estrutura do GStreamer fornece um ambiente gerenciado padrão para construir o fluxo de mídia de um dispositivo, como uma câmera ou outra fonte de vídeo para processamento adicional, renderização ou armazenamento.

O pipeline do GStreamer normalmente consiste no link entre uma fonte (câmera de vídeo) e o elemento coletor (um player para renderizar o vídeo ou um armazenamento para recuperação off-

line). Neste exemplo, você usa o elemento Producer SDK como um coletor, ou mídia de destino, para sua fonte de vídeo (webcam ou câmera IP). O elemento de plug-in que encapsula o SDK então envia o stream de vídeo para o Kinesis Video Streams.

Este tópico descreve como construir um pipeline de mídia do GStreamer capaz de transmitir vídeo de uma fonte de vídeo, como uma webcam ou stream RTSP, normalmente conectado por meio de estágios intermediários de codificação (usando a codificação H.264) ao Kinesis Video Streams. Quando seu stream de vídeo está disponível como um stream de vídeo do Kinesis, você pode usar a Biblioteca do [Analisador do Kinesis Video Streams para processamento, reprodução, armazenamento ou análise](#) adicionais do seu stream de vídeo.

Baixe, crie e configure o elemento GStreamer

O exemplo do plug-in GStreamer está incluído no SDK do Kinesis Video Streams C++ Producer. Para obter informações sobre os pré-requisitos e o download do SDK, consulte [Etapa 1: Baixe e configure o código da Biblioteca do Produtor de C++](#).

Você pode criar o coletor GStreamer para o Producer SDK como uma biblioteca dinâmica no macOS, Ubuntu, Raspberry Pi ou Windows. O plug-in do GStreamer está localizado no diretório `build`. Para carregar este plugin, ele deve estar no seu `GST_PLUGIN_PATH`. Execute o seguinte comando:

```
export GST_PLUGIN_PATH=`pwd`/build
```

Note

No macOS, você só pode fazer streaming de vídeo de uma câmera de rede ao executar o GStreamer em um contêiner do Docker. O streaming de vídeo de uma câmera USB no macOS em um contêiner do Docker não é suportado.

Execute o elemento GStreamer

Para executar o GStreamer com o elemento SDK do Kinesis Video Streams Producer como coletor, use o comando `gst-launch-1.0`. Use elementos upstream que sejam apropriados para o plug-in GStreamer usar. Por exemplo, [v4l2src](#) para dispositivos v4l2 em sistemas Linux ou [rtspsrc](#) para

dispositivos RTSP. Especifique `kvssink` como o coletor (destino final do pipeline) para enviar vídeo ao Producer SDK.

Além de [fornecer credenciais](#) e [fornecer uma região](#), o `kvssink` elemento tem o seguinte parâmetro obrigatório:

- `stream-name`— O nome do destino Kinesis Video Streams.

Para obter informações sobre os parâmetros opcionais `kvssink`, consulte [Referência de parâmetros do elemento GStreamer](#).

Para obter as informações mais recentes sobre os plug-ins e parâmetros do GStreamer, consulte Plugins do [GStreamer](#). Você também pode usar `gst-inspect-1.0` seguido pelo nome de um elemento ou plug-in do GStreamer para imprimir suas informações e verificar se elas estão disponíveis em seu dispositivo:

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

Se a construção `kvssink` falhar ou o `GST_PLUGIN_PATH` não estiver configurado corretamente, sua saída será semelhante a esta:

```
No such element or plugin 'kvssink'
```

Exemplo de comandos de inicialização do GStreamer

Os exemplos a seguir demonstram como usar o plug-in `kvssink` GStreamer para transmitir vídeo de diferentes tipos de dispositivos.

Exemplo 1: Transmite vídeo de uma câmera RTSP no Ubuntu

O comando a seguir cria um pipeline do GStreamer no Ubuntu que transmite de uma câmera RTSP na rede, usando o plug-in do GStreamer do [rtspsrc](#):

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !  
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```


Exemplo 2: codifique e transmita vídeo de uma câmera USB no Ubuntu

O comando a seguir cria um pipeline do GStreamer no Ubuntu que codifica o stream de uma câmera USB no formato H.264 e o transmite para o Kinesis Video Streams. Este exemplo usa o plug-in [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"
```

Exemplo 3: Transmita vídeo pré-codificado de uma câmera USB no Ubuntu

O comando a seguir cria um pipeline do GStreamer no Ubuntu que transmite vídeo que a câmera já codificou no formato H.264 para o Kinesis Video Streams. Este exemplo usa o plug-in [v4l2src](#) GStreamer.

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

Exemplo 4: Transmita vídeo de uma câmera de rede no macOS

O comando a seguir cria um pipeline do GStreamer no macOS que transmite vídeo para o Kinesis Video Streams a partir de uma câmera de rede. Este exemplo usa o plug-in do GStreamer do [rtspsrc](#).

```
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"
```

Exemplo 5: Transmita vídeo de uma câmera de rede no Windows

O comando a seguir cria um pipeline do GStreamer no Windows que transmite vídeo para o Kinesis Video Streams a partir de uma câmera de rede. Este exemplo usa o plug-in do GStreamer do [rtspsrc](#).

```
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"
```

Exemplo 6: Transmita vídeo de uma câmera no Raspberry Pi

O comando a seguir cria um pipeline do GStreamer no Raspberry Pi que transmite vídeo para o Kinesis Video Streams. Este exemplo usa o plug-in [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 !  
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"
```

Exemplo 7: Transmita áudio e vídeo no Raspberry Pi e no Ubuntu

Veja como [executar o comando gst-launch-1.0 para iniciar o streaming de áudio e vídeo no Raspberry-Pi e Ubuntu](#).

Exemplo 8: Transmita áudio e vídeo de fontes de dispositivos no macOS

Veja como [executar o comando gst-launch-1.0 para iniciar o streaming de áudio e vídeo no MacOS](#).

Exemplo 9: Carregue o arquivo MKV que contém áudio e vídeo

Veja como [executar o comando gst-launch-1.0 para fazer upload do arquivo MKV que contém áudio e vídeo](#). Você precisará de um arquivo de teste MKV com mídia codificada em h.264 e AAC.

Execute o elemento GStreamer em um contêiner Docker

O Docker é uma plataforma para desenvolvimento, implantação e execução de aplicativos usando contêineres. Usar o Docker para criar o pipeline do GStreamer padroniza o ambiente operacional do Kinesis Video Streams, o que simplifica a criação e o uso do aplicativo.

Para instalar e configurar o Docker, consulte o seguinte:

- [Instruções para download do Docker](#)
- [Conceitos básicos do Docker](#)

Depois de instalar o Docker, você pode baixar o SDK do Kinesis Video Streams C++ Producer (e o plug-in GStreamer) do Amazon Elastic Container Registry usando um dos comandos fornecidos abaixo. `docker pull`

Para executar o GStreamer com o elemento SDK do Kinesis Video Streams Producer como coletor em um contêiner do Docker, faça o seguinte:

Tópicos

- [Autentique seu cliente Docker](#)
- [Fazer download da imagem de docker para Ubuntu, macOS, Windows ou Raspberry Pi](#)
- [Execute a imagem do Docker](#)

Autentique seu cliente Docker

Autentique o cliente do Docker para o registro do Amazon ECR do qual você pretende extrair a imagem. Você deve obter tokens de autenticação para cada registro usado. Os tokens são válidos por 12 horas. Para obter mais informações, consulte [Autenticação de Registro](#) no campo do Manual do Usuário do Amazon Elastic Container Registry.

Example : autenticar com o Amazon ECR

Para se autenticar com o Amazon ECR, copie e cole o seguinte comando conforme mostrado.

```
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
```

Se for bem-sucedido, a saída imprimirá `Login Succeeded`.

Fazer download da imagem de docker para Ubuntu, macOS, Windows ou Raspberry Pi

Faça download da imagem do Docker no ambiente do Docker usando um dos comandos a seguir, dependendo do sistema operacional:

Fazer download da imagem do Docker para Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Fazer download da imagem do Docker para macOS

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

Fazer download da imagem de docker para Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

Fazer download da imagem do Docker para Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

Para verificar se a imagem foi adicionada com êxito, execute o seguinte comando:

```
docker images
```

Execute a imagem do Docker

Use um dos comandos a seguir para executar a imagem do Docker, dependendo do sistema operacional:

Execute a imagem do Docker no 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
```

Execute a imagem do Docker no macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

Execute a imagem do Docker no 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
```

Execute a imagem do Docker no 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
```

O Docker inicia o contêiner e apresenta um prompt de comando para usar comandos dentro do contêiner.

No contêiner, defina as variáveis de ambiente usando o seguinte 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
```

Comece a transmitir kvssink usando o `gst-launch-1.0` para executar um pipeline apropriado para seu dispositivo e fonte de vídeo. Por exemplo, oleodutos, consulte [Exemplo de comandos de inicialização do GStreamer](#).

Referência de parâmetros do elemento GStreamer

Para enviar vídeo para o SDK C++ do Amazon Kinesis Video Streams Producer, kvssink você especifica como coletor ou destino final do pipeline. Essa referência fornece informações sobre o kvssink obrigatório e os parâmetros opcionais. Para ter mais informações, consulte [the section called “Plugin do GStreamer - kvssink”](#).

Tópicos

- [the section called “Forneça credenciais para kvssink”](#)
- [the section called “Forneça uma região para kvssink”](#)
- [the section called “kvssinkparâmetros opcionais”](#)

Forneça credenciais para kvssink

Para permitir que o elemento kvssink GStreamer faça solicitações AWS, forneça AWS credenciais para que ele use ao chamar o serviço Amazon Kinesis Video Streams. A cadeia de fornecedores de credenciais procura as credenciais na seguinte ordem:

1. AWS IoT credenciais

Para configurar AWS IoT as credenciais, consulte [the section called “Controlando o acesso aos recursos do Kinesis Video Streams usando AWS IoT”](#).

O valor do `iot-credentials` parâmetro deve começar com `iot-certificate`, e ser seguido por uma lista separada por vírgulas dos seguintes pares *chave = valor*.

Chave	Obrigatório	Descrição
<code>ca-path</code>	Sim	<p>Caminho do arquivo para o certificado CA usado para estabelecer confiança com o serviço de back-end por meio do TLS.</p> <p>Example</p> <p>Exemplo: <code>/file/path/to/certificate.pem</code></p>
<code>cert-path</code>	Sim	<p>Caminho do arquivo para o certificado X.509.</p> <p>Example</p> <p>Exemplo: <code>/file/path/to/certificateID -certificate.pem.crt</code></p>
<code>endpoint</code>	Sim	<p>O endpoint do provedor de endpoint de AWS IoT Core credenciais para sua conta. AWS Consulte o Guia AWS IoT do desenvolvedor.</p>

Chave	Obrigatório	Descrição
		<p>Example</p> <p>Exemplo: <i>credential-account-specific-prefix</i> .credentials.iot. <i>aws-region</i> .amazonaws.com</p>
key-path	Sim	<p>Caminho do arquivo para a chave privada usada no par de chaves pública/privada.</p> <p>Example</p> <p>Exemplo: <i>/file/path/to/certificateID-private.pem.key</i></p>
role-aliases	Sim	<p>O nome do alias da função que aponta para a função do AWS IAM a ser usada ao se conectar AWS IoT Core.</p> <p>Example</p> <p>Exemplo: <i>KvsCameraIoTRoleAlias</i></p>
iot-thing-name	Não	<p>iot-thing-name é opcional. Se não iot-thing-name for fornecido, o valor do stream-name parâmetro será usado.</p> <p>Example</p> <p>Exemplo: <i>kvs_example_camera</i></p>

Example

Exemplo:

```
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. Variáveis de ambiente

Para ter credenciais de kvssink uso do ambiente, defina as seguintes variáveis de ambiente:

Nome da variável de ambiente	Obrigatório	Descrição
AWS_ACCESS_KEY_ID	Sim	A chave de AWS acesso usada para acessar o Amazon Kinesis Video Streams.
AWS_SECRET_ACCESS_KEY	Sim	A chave AWS secreta associada à chave de acesso.
AWS_SESSION_TOKEN	Não	Especifica o valor do token de sessão necessário se você usar credenciais de segurança temporárias diretamente das AWS STS operações.

Configurar a variável de ambiente altera o valor usado até o final da sua sessão de shell ou até que você defina a variável como um valor diferente. Para tornar as variáveis persistentes em futuras sessões, defina-as no script de inicialização do shell.

3. **access-key**, **secret-key** parâmetros

Para especificar as credenciais diretamente como um kvssink parâmetro, defina os seguintes parâmetros:

kvssinkNome do parâmetro	Obrigatório	Descrição
access-key	Sim	A chave de AWS acesso usada para acessar o Amazon Kinesis Video Streams.
secret-key	Sim	A chave AWS secreta associada à chave de acesso.
session-token	Não	Especifica o valor do token de sessão necessário se você usar credenciais de segurança temporárias diretamente das AWS STS operações.

Example

Usando credenciais estáticas:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

Example

Usando credenciais temporárias:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

4. Arquivo de credenciais

Important

Se você selecionou um dos métodos anteriores, não poderá usar o `credential-file` kvssink parâmetro.

Nome do parâmetro	Obrigatório	Descrição
<code>credential-file</code>	Sim	Caminho para o arquivo de texto contendo credenciais em um formato específico.

O arquivo de texto deve conter credenciais em um dos seguintes formatos:

- CREDENCIAIS *YourAccessKeyYourSecretKey*
- *Expiração de CREDENCIAIS YourAccessKeyYourSecretKeySessionToken*

Example

Exemplo: Seu *credentials.txt* arquivo está localizado em `/home/ubuntu` e contém o seguinte:

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

Para usá-lo em `kvssink`, digite:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
credential-file="/home/ubuntu/credentials.txt"
```

Note

O tempo de expiração deve ser de pelo menos $5 + 30 + 3 = 38$ segundos no futuro. O período de carência é definido como a `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` variável em [IotCredentialProvider.h](#). Se as credenciais estiverem muito próximas da expiração quando você começar `kvssink`, você receberá o código `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION` de erro.

Important

`kvssink` não modifica o arquivo de credenciais. Se você estiver usando credenciais temporárias, o arquivo de credenciais deverá ser atualizado por uma fonte externa antes do prazo de expiração menos o período de carência.


Forneça uma região para **kvssink**

A seguir está a ordem de pesquisa da região:



1. `AWS_DEFAULT_REGION`Na variável de ambiente é revisada primeiro. Se estiver definida, essa região será usada para configurar o cliente.
2. `aws-region`o parâmetro é revisado a seguir. Se estiver definida, essa região será usada para configurar o cliente.
3. Se nenhum dos métodos anteriores foi usado, o `kvssink` padrão é. `us-west-2`

kvssinkparâmetros opcionais

O elemento `kvssink` tem os seguintes parâmetros opcionais. Para mais informações sobre esses parâmetros, consulte [Estruturas de transmissão de vídeo do Kinesis](#).

Parâmetro	Descrição	Unidade/tipo	Padrão
<code>stream-name</code>	<p>O nome do stream de vídeo do Amazon Kinesis de destino.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Se nenhum nome de fluxo for especificado, o nome padrão do fluxo será usado: "DEFAULT_STREAM". Se um stream com esse nome padrão ainda não</p> </div>		

Parâmetro	Descrição	Unidade/tipo	Padrão
	existir, ele será criado.		
<code>absolute-fragment-times</code>	Se devem ser usadas horas absolutas de fragmentos.	Booleano	true
<code>access-key</code>	<p>A chave de AWS acesso usada para acessar o Kinesis Video Streams.</p> <p>Você deve ter AWS as credenciais definidas ou fornecer esse parâmetro. Para fornecer essas informações, digite o seguinte:</p> <pre>export AWS_ACCESS_KEY_ID=</pre>		
<code>avg-bandwidth-bps</code>	A largura de banda média esperada para o stream.	Bits por segundo	4194304


Parâmetro	Descrição	Unidade/tipo	Padrão
<code>aws-region</code>	<p>O Região da AWS para usar.</p> <div data-bbox="472 352 792 1287"><p> Note</p><p>Você também pode fornecer à região a variável de <code>AWS_DEFAULT_REGION</code> ambiente. As variáveis de ambiente têm precedência se a variável de ambiente e os parâmetros <code>kvssink</code> estiverem definidos.</p></div> <div data-bbox="472 1354 792 1864"><p> Important</p><p>A região será usada como padrão, <code>us-west-2</code> se não for especificada de outra forma.</p></div>	String	<code>"us-west-2"</code>

Parâmetro	Descrição	Unidade/tipo	Padrão
<code>buffer-duration</code>	A duração do buffer de streaming.	Segundos	120
<code>codec-id</code>	O ID do codec do streaming.	String	"V_MPEG4/ISO/AVC"
<code>connection-staleness</code>	O tempo após o qual o retorno de chamada de inatividade do stream é chamado.	Segundos	60
<code>content-type</code>	O tipo do conteúdo do streaming.	String	"video/h264"
<code>fragment-acks</code>	Se devem ser usadas ACKs de fragmentos.	Booleano	true
<code>fragment-duration</code>	A duração do fragmento desejada.	Milissegundos	2000
<code>framerate</code>	A taxa de quadros esperada.	Quadros por segundo	25
<code>frame-timecodes</code>	Determina se usará códigos de hora de quadros ou gerará carimbos de hora usando o retorno de hora atual.	Booleano	true
<code>key-frame-fragmentation</code>	Se fragmentos devem ser produzidos em um quadro chave.	Booleano	true
<code>log-config</code>	O caminho da configuração do log.	String	"../kvs_log_configuration"

Parâmetro	Descrição	Unidade/tipo	Padrão
max-latency	A latência máxima do streaming.	Segundos	60
recalculate-metrics	Se as métricas devem ser recalculadas.	Booleano	true
replay-duration	A duração para reverter o leitor atual para reprodução durante um erro se a reinicialização estiver habilitada.	Segundos	40
restart-on-error	Se uma reinicialização deve ser executada quando ocorre um erro.	Booleano	true
retention-period	A duração do streaming é preservada.	Horas	2
rotation-period	O período de rotação de chaves. Para obter mais informações, consulte AWS KMS Chaves rotativas .	Segundos	3600

Parâmetro	Descrição	Unidade/tipo	Padrão
secret-key	<p>A chave AWS secreta usada para acessar o Kinesis Video Streams.</p> <p>Você deve ter AWS as credenciais definidas ou fornecer esse parâmetro.</p> <pre>export AWS_SECRET_ACCESS_KEY=</pre>		
session-token	<p>Especifica o valor do token de sessão necessário se você usar credenciais de segurança temporárias diretamente das AWS STS operações.</p>		
storage-size	<p>O tamanho do armazenamento do dispositivo em mebibyte (MiB). Para obter informações sobre como configurar o armazenamento de dispositivos, consulte StorageInfo.</p>	Mebibyte (MiB)	128

Parâmetro	Descrição	Unidade/tipo	Padrão
<code>streaming-type</code>	O tipo do streaming . Os valores válidos são: <ul style="list-style-type: none">• 0: tempo real• 1: quase tempo real (não compatível atualmente)• 2: off-line	Enum <code>GstKvsSinkStreamingType</code>	0: tempo real
<code>timecode-scale</code>	A escala do timecode do MKV.	Milissegundos	1
<code>track-name</code>	O nome da trilha do MKV.	String	"kinesis_video"

Parâmetro	Descrição	Unidade/tipo	Padrão
<code>iot-certificate</code>	<p>AWS IoT credenciais a serem usadas no kvssink elemento.</p> <p><code>iot-certificate</code> aceita as seguintes chaves e valores:</p> <div data-bbox="472 621 792 1320" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note</p> <p>O <code>iot-thing-name</code> é opcional. Se não <code>iot-thing-name</code> for fornecido, o valor do <code>stream-name</code> parâmetro será usado.</p> </div> <ul style="list-style-type: none"> • <code>endpoint=iotcredentialsproviderendpoint</code> • <code>cert-path=/localdirectorypath/to/certificate</code> • <code>key-path=/localdir</code> 	String	Nenhum

Parâmetro	Descrição	Unidade/tipo	Padrão
	<p>ectorypath / to/private/ key</p> <ul style="list-style-type: none"> • ca-path=/ localdir ectorypath/ to/ca-cert • role-aliases =role-aliases • iot-thing-name=YourIotThingName 		

Exemplo: envio de dados para o Kinesis Video Streams PutMedia usando a API

Este exemplo demonstra como usar a [PutMedia API](#). Mostra como enviar dados que já estão em formato de contêiner (MKV). Se seus dados precisarem ser reunidos em um formato de contêiner antes do envio (por exemplo, se você estiver reunindo dados de vídeo da câmera em quadros), consulte [Bibliotecas do Kinesis Video Streams Producer](#).

Note

A PutMedia operação está disponível somente nos SDKs de C++ e Java. Isso se deve ao gerenciamento full-duplex de conexões, fluxo de dados e confirmações. Não é compatível com outros idiomas.

Este exemplo inclui as seguintes etapas:

- [Etapa 1: Baixe e configure o código](#)
- [Etapa 2: escrever e examinar o código](#)
- [Etapa 3: executar e verificar o código](#)

Etapa 1: Baixe e configure o código

Siga as etapas para baixar o código de exemplo Java, importar o projeto em seu Java IDE, configurar os locais da biblioteca e configurar o código para usar suas AWS credenciais.

1. Crie um diretório e clone o código-fonte de exemplo do GitHub repositório. O exemplo de PutMedia é parte da [Biblioteca de produtores Java](#).

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

2. Abra o Java IDE que você está usando (por exemplo, [Eclipse](#) ou [IntelliJ IDEA](#)) e importe o projeto Apache Maven que você baixou:
 - No Eclipse: selecione File (Arquivo), Import (Importar), Maven (Maven), Existing Maven Projects (Projetos existentes do Maven) e navegue até a raiz do pacote obtido por download. Selecione o arquivo pom.xml.
 - No IntelliJ Idea: escolha Import. Navegue até o arquivo pom.xml na raiz do pacote que você obteve por download.

Para obter mais informações, consulte a documentação relacionada ao IDE.

3. Atualize o projeto para que o IDE possa localizar as bibliotecas que você importou.
 - Para IntelliJ IDEA, faça o seguinte:
 - a. Abra o menu de contexto (clique com o botão direito do mouse) da lib do projeto e escolha Add as library.
 - b. Escolha Arquivo e, em seguida, escolha Estrutura do projeto.
 - c. Em Project Settings, escolha Modules.
 - d. Na guia Sources (Fontes), defina Language Level (Nível de linguagem) como **7** ou um número superior.
 - Para Eclipse, faça o seguinte:
 - a. Abra o menu contextual (clique direito) no projeto e escolha Properties, Java Build Path, Source. Então, faça o seguinte:
 1. Na guia Source, clique duas vezes em Native library location.
 2. No assistente Native Library Folder Configuration, escolha Workspace.
 3. Na caixa de seleção Native Library Folder, escolha o diretório lib no projeto.

- b. Abra o menu de contexto (clique com o botão direito do mouse) do projeto e escolha Properties. Então, faça o seguinte:
 1. Na guia Libraries, escolha Add Jars.
 2. No assistente JAR selection, escolha tudo .jars no diretório lib do projeto.

Etapa 2: escrever e examinar o código

O exemplo de API PutMedia () mostra o padrão de codificação a seguir:PutMediaDemo

Tópicos

- [Crie o PutMediaClient](#)
- [Transmitir mídia e pausar o thread](#)

Os exemplos de código nesta seção são da classe PutMediaDemo.

Crie o PutMediaClient

A criação do PutMediaClient objeto requer os seguintes parâmetros:

- O URI endpoint PutMedia.
- Um InputStream apontando para o arquivo MKV para streaming.
- Nome do streaming. Este exemplo usa o mesmo streaming criado em [Usar a Biblioteca de produtores Java](#) (my-stream). Para usar outro streaming, altere o seguinte parâmetro:

```
private static final String STREAM_NAME="my-stream";
```

Note

O exemplo PutMedia da API não cria um stream. Você deve criar um stream usando o aplicativo de teste do [Usar a Biblioteca de produtores Java](#), o console do Kinesis Video Streams ou o AWS CLI

- O time stamp atual.
- O tipo de código de hora. O exemplo usa RELATIVE, indicando que o time stamp é relativo ao início do contêiner.

- Um objeto `AWSKinesisVideoV4Signer` que verifica se os pacotes recebidos foram enviados pelo remetente autorizado.
- A largura de banda máxima de upstream em Kbps.
- Um objeto `AckConsumer` para confirmações de recebimento do pacote.

O código a seguir cria o 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();
```

Transmitir mídia e pausar o thread

Após a criação do cliente, a amostra inicia o streaming assíncrono com `putMediaInBackground`. Em seguida, o thread principal é pausado com `latch.await` até que `AckConsumer` retorne. Nesse momento, o cliente é fechado.

```
/* start streaming video in a background thread */
```

```
client.putMediaInBackground();

/* wait for request/response to complete */
latch.await();

/* close the client */
client.close();
```

Etapa 3: executar e verificar o código

Para executar o exemplo da API PutMedia, faça o seguinte:

1. Crie um stream chamado `my-stream` no console do Kinesis Video Streams ou usando o. AWS CLI
2. Altere seu diretório de trabalho para o diretório do Java Producer SDK:

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Compile o SDK do Java e o aplicativo de demonstração:

```
mvn package
```

4. Crie um nome de arquivo temporário no diretório `/tmp`:

```
jar_files=$(mktemp)
```

5. Crie uma string de classpath de dependências do repositório local para um arquivo:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. Defina o valor da variável de ambiente `LD_LIBRARY_PATH` da seguinte forma:

```
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. Execute a demonstração na linha de comando da seguinte forma, fornecendo suas AWS credenciais:

```
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 o console do [Kinesis Video Streams](#) e escolha seu stream na página Gerenciar Streams. O vídeo é reproduzido no painel Video Preview.

Exemplo: streaming de uma fonte RTSP

O [Biblioteca de produtores C++](#) contém uma definição para um contêiner [Docker](#) que se conecta a uma câmera de rede RTSP (Real-Time Streaming Protocol). O uso do Docker padroniza o ambiente operacional do Kinesis Video Streams, o que simplifica a criação e o uso do aplicativo.

O procedimento a seguir demonstra como configurar e usar o aplicativo do RTSP de demonstração.

Tópicos

- [Vídeos de tutorial](#)
- [Pré-requisitos](#)
- [Crie a imagem do Docker](#)
- [Execute o aplicativo de exemplo RTSP](#)

Vídeos de tutorial

Este vídeo mostra como configurar um Raspberry Pi para enviar feeds RTSP para a AWS nuvem e para o Amazon Kinesis Video Streams. Isso é uma end-to-end demonstração.

Este vídeo demonstra como capturar imagens de um feed para usar a visão computacional e o Amazon Rekognition para processar as imagens e enviar alertas.

Pré-requisitos

Para executar o aplicativo de exemplo RTSP do Kinesis Video Streams, você deve ter o seguinte:

- Docker: para obter informações sobre como instalar e usar o Docker, consulte os links a seguir:
 - [Instruções para download do Docker](#)
 - [Conceitos básicos do Docker](#)

- Fonte de câmera de rede RTSP: para obter mais informações sobre as câmeras recomendadas, consulte [Requisitos do sistema](#).

Crie a imagem do Docker

Primeiro, crie a imagem do Docker na qual o aplicativo de demonstração será executado.

1. Clone o repositório de demonstrações do Amazon Kinesis Video Streams.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Mude para o diretório que contém o Dockerfile. Nesse caso, é o diretório [docker-rtsp](#).

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. Use o comando a seguir para criar a imagem do Docker. Esse comando cria a imagem e a marca como `rtspdockertest`.

```
docker build -t rtspdockertest .
```

4. Execute `docker images` e pesquise o ID da imagem marcado com `rtspdockertest`.

Por exemplo, no exemplo de saída abaixo, `IMAGE ID` é `54f0d65f69b2`.

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB

Você precisará disso em uma etapa posterior.

Execute o aplicativo de exemplo RTSP

Você pode executar o aplicativo de exemplo RTSP de dentro ou de fora do contêiner Docker. Siga as instruções apropriadas abaixo.

Tópicos

- [Dentro do contêiner Docker](#)
- [Fora do contêiner Docker](#)

Dentro do contêiner Docker

Execute o aplicativo de exemplo RTSP

1. Inicie o contêiner Docker do Amazon Kinesis Video Streams usando o seguinte comando:

```
docker run -it YourImageId /bin/bash
```

2. Para iniciar o aplicativo de amostra, forneça suas AWS credenciais, o nome do stream de vídeo do Amazon Kinesis e a URL da câmera de rede RTSP.

Important

Se você estiver usando credenciais temporárias, você também precisará fornecer suas `AWS_SESSION_TOKEN`. Veja o segundo exemplo abaixo.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId  
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId  
export AWS_DEFAULT_REGION=YourAWSRegion  
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Credenciais temporárias:

```
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. Faça login AWS Management Console e abra o console do [Kinesis Video Streams](#).

Veja o stream.

4. Para sair do contêiner Docker, feche a janela do terminal ou digite `exit`.

Fora do contêiner Docker

De fora do contêiner do Docker, use o seguinte 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"
```

Exemplo: análise e renderização de fragmentos do Kinesis Video Streams

O [Biblioteca do analisador de streaming](#) contém um aplicativo de demonstração chamado `KinesisVideoRendererExample` que demonstra a análise e a renderização de fragmentos de stream de vídeo do Amazon Kinesis. O exemplo usa [JCodec](#) para decodificar os quadros codificados em H.264 que são recebidos por meio do aplicativo [Exemplo: Plugin GStreamer do SDK do Kinesis Video Streams Producer - kvssink](#). Após a decodificação do quadro por meio de `JCodec`, a imagem visível é renderizada por meio de [JFrame](#).

O exemplo mostra como:

- Recupere quadros de um stream de vídeo do Kinesis usando `GetMedia` a API e renderize o stream para visualização.
- Visualize o conteúdo de vídeo dos streams em um aplicativo personalizado em vez de usar o console do Kinesis Video Streams.

Você também pode usar as classes neste exemplo para visualizar o conteúdo do stream de vídeo do Kinesis que não está codificado como H.264, como um stream de arquivos JPEG que não precisam ser decodificados antes de serem exibidos.

O procedimento a seguir demonstra como configurar e usar o aplicativo de demonstração `Renderizador`.

Pré-requisitos

Para examinar e usar a biblioteca de exemplo `Renderizador`, você precisa fazer o seguinte:

- Uma conta da Amazon Web Services (AWS). Se você ainda não tiver uma AWS conta, consulte [Introdução ao Kinesis Video Streams](#).
- [Um ambiente de desenvolvimento integrado \(IDE\) Java, como Eclipse Java Neon ou JetBrains IntelliJ Idea](#).

Executando o exemplo do renderizador

1. Crie um diretório e, em seguida, clone o código-fonte de exemplo do GitHub repositório.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Abra o IDE do Java que você está usando (por exemplo, [Eclipse](#) ou [IntelliJ IDEA](#)) e importe o projeto Apache Maven que você obteve por download:
 - No Eclipse: escolha File, Import, Maven e Existing Maven Projects. Navegue até o diretório `kinesis-video-streams-parser-lib`.
 - No IntelliJ Idea: escolha Import. Navegue até o arquivo `pom.xml` na raiz do pacote que você obteve por download.

Note

Se IntelliJ não puder encontrar suas dependências, você pode ter de fazer o seguinte:

- Limpar compilação: escolha File (Arquivo), Settings (Configurações), Build, Execution, Deployment (Compilação, Execução, Implantação), Compiler (Compilador). Verifique se a opção Limpar diretório de saída na reconstrução está selecionada e escolha Construir, Construir projeto.
- Reimportar o projeto: Abra o menu de contexto (clique com botão direito) no projeto e escolha Maven, Reimport (Reimportar).

Para obter mais informações, consulte a documentação relacionada ao IDE.

3. No Java IDE, abra `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest`.
4. Remova a diretiva `@Ignore` do arquivo.
5. Atualize o `.stream` parâmetro com o nome do seu stream de vídeo do Kinesis.
6. Execute o teste do `KinesisVideoRendererExample`.

Como funciona

O aplicativo de exemplo faz o seguinte:

- [Envio de dados do MKV](#)

- [Analisando fragmentos de MKV em quadros](#)
- [Decodificando e exibindo o quadro](#)

Envio de dados do MKV

O exemplo envia amostras de dados MKV do `rendering_example_video.mkv` arquivo, usando `PutMedia` para enviar dados de vídeo para um stream chamado `render-example-stream`.

O aplicativo cria um `PutMediaWorker`:

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

Para obter mais informações sobre a classe `PutMediaWorker`, consulte [Ligue PutMedia](#) na documentação [Biblioteca do analisador de streaming](#).

Analisando fragmentos de MKV em quadros

O exemplo recupera e analisa os fragmentos do MKV do streaming por meio de um `GetMediaWorker`:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getFrameVisitor());
executorService.submit(getMediaWorker);
```

Para obter mais informações sobre a classe `GetMediaWorker`, consulte [Ligue GetMedia](#) na documentação [Biblioteca do analisador de streaming](#).

Decodificando e exibindo o quadro

O exemplo decodifica e exibe o quadro usando [JFrame](#).

O seguinte exemplo de código vem da classe `KinesisVideoFrameViewer`, que estende `JFrame`:

```
public void setImage(BufferedImage bufferedImage) {  
    image = bufferedImage;  
    repaint();  
}
```

A imagem é exibida como uma instância de [java.awt.image. BufferedImage](#). Para obter exemplos que mostram como trabalhar com `BufferedImage`, consulte [Leitura/carregamento de uma imagem](#).

Monitoramento dos Amazon Kinesis Video Streams

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho do Amazon Kinesis Video Streams AWS e de suas soluções. Recomendamos coletar dados de monitoramento de todas as partes da sua AWS solução para ajudá-lo a depurar uma falha multiponto, caso ocorra. Antes de começar a monitorar o Amazon Kinesis Video Streams, recomendamos que você crie um plano de monitoramento que inclua respostas às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

Depois de definir suas metas de monitoramento e criar seu plano de monitoramento, a próxima etapa é estabelecer uma linha de base para o desempenho normal do Amazon Kinesis Video Streams em seu ambiente. Você deve medir o desempenho do Amazon Kinesis Video Streams em vários momentos e sob diferentes condições de carga. Ao monitorar o Amazon Kinesis Video Streams, armazene um histórico dos dados de monitoramento que você coletou. Você pode comparar o desempenho atual do Amazon Kinesis Video Streams com esses dados históricos para ajudá-lo a identificar padrões normais de desempenho e anomalias de desempenho e criar métodos para resolver problemas que possam surgir.

Tópicos

- [Monitorando as métricas do Amazon Kinesis Video Streams com CloudWatch](#)
- [Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch](#)
- [Registro de chamadas de API do Amazon Kinesis Video Streams com AWS CloudTrail](#)

Monitorando as métricas do Amazon Kinesis Video Streams com CloudWatch

Você pode monitorar um stream de vídeo do Kinesis usando a Amazon CloudWatch, que coleta e processa dados brutos do Amazon Kinesis Video Streams em métricas legíveis e quase em tempo real. Essas estatísticas são registradas por um período de 15 meses para que você possa acessar informações históricas e ter uma melhor perspectiva sobre o desempenho de seu aplicativo ou serviço da web.

No console do [Amazon Kinesis Video](#) Streams, você CloudWatch pode visualizar as métricas de um stream de vídeo do Amazon Kinesis de duas maneiras:

- Na página Dashboard (Painel), selecione a guia Video streams (Streamings de vídeo) na seção Account-level metrics for Current Region (Métricas no nível da conta para a região atual).
- Selecione a guia Monitoring (Monitoramento) na página de detalhes do streaming de vídeo.


O Amazon Kinesis Video Streams fornece as seguintes métricas:


Métrica	Descrição
ArchivedFragmentsConsumed.Media	<p>O número de pontos de cota de mídia fragmentada que foram consumidos por todas as APIs. Para obter uma explicação sobre o conceito de pontos de cota, consulte the section called “Cotas de fragmentação de metadados e de mídia fragmentada”.</p> <p>Unidades: contagem</p>
ArchivedFragmentsConsumed.Metadata	<p>O número de pontos de cota de metadados de fragmentos que foram consumidos por todas as APIs. Para obter uma explicação sobre o conceito de pontos de cota, consulte the section called “Cotas de fragmentação de metadados e de mídia fragmentada”.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>PutMedia.Requests</code>	<p>O número de solicitações de PutMedia API para um determinado stream.</p> <p>Unidades: contagem</p>
<code>PutMedia.IncomingBytes</code>	<p>O número de bytes recebidos como parte PutMedia do fluxo.</p> <p>Unidade: bytes</p>
<code>PutMedia.IncomingFragments</code>	<p>O número de fragmentos completos recebidos como parte do PutMedia fluxo.</p> <p>Unidades: contagem</p>
<code>PutMedia.IncomingFrames</code>	<p>O número de quadros completos recebidos como parte PutMedia do stream.</p> <p>Unidades: contagem</p>
<code>PutMedia.ActiveConnections</code>	<p>O número total de conexões com o serviço host.</p> <p>Unidades: contagem</p>
<code>PutMedia.ConnectionErrors</code>	<p>Os erros ao estabelecer PutMedia a conexão para o stream.</p> <p>Unidades: contagem</p>
<code>PutMedia.FragmentIngestionLatency</code>	<p>A diferença de tempo entre quando o primeiro e o último bytes de um fragmento são recebidos pelo Amazon Kinesis Video Streams.</p> <p>Unidade: milissegundos</p>


Métrica	Descrição
<code>PutMedia.FragmentPersistLatency</code>	<p>O tempo decorrido a partir do momento em que os dados completos do fragmento são recebidos e arquivados.</p> <p>Unidades: contagem</p>
<code>PutMedia.Latency</code>	<p>A diferença de tempo entre a solicitação e a resposta HTTP InletService ao estabelecer a conexão.</p> <p>Unidades: contagem</p>
<code>PutMedia.BufferingAckLatency</code>	<p>A diferença de tempo entre quando o primeiro byte de um novo fragmento é recebido pelo Amazon Kinesis Video Streams e quando o Buffering ACK é enviado para o fragmento.</p> <p>Unidade: milissegundos</p>
<code>PutMedia.ReceivedAckLatency</code>	<p>A diferença de tempo entre quando o último byte de um novo fragmento é recebido pelo Amazon Kinesis Video Streams e quando o ACK recebido é enviado para o fragmento.</p> <p>Unidade: milissegundos</p>
<code>PutMedia.PersistedAckLatency</code>	<p>A diferença de tempo entre o momento em que o último byte de um novo fragmento é recebido pelo Amazon Kinesis Video Streams e o momento em que o ACK persistente é enviado para o fragmento.</p> <p>Unidade: milissegundos</p>
<code>PutMedia.ErrorAckCount</code>	<p>O número de ACKs de erro enviados PutMedia durante a execução do stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>PutMedia.Success</code>	<p>1 para cada fragmento gravado com êxito; 0 para cada fragmento com falha. O valor médio dessa métrica indica quantos fragmentos completos e válidos são enviados.</p> <p>Unidades: contagem</p>
<code>GetMedia.Requests</code>	<p>O número de solicitações de <code>GetMedia</code> API para um determinado stream.</p> <p>Unidades: contagem</p>
<code>GetMedia.OutgoingBytes</code>	<p>O número total de bytes enviados do serviço como parte da <code>GetMedia</code> API para um determinado stream.</p> <p>Unidade: bytes</p>
<code>GetMedia.OutgoingFragments</code>	<p>O número de fragmentos enviados durante a execução <code>GetMedia</code> do stream.</p> <p>Unidades: contagem</p>
<code>GetMedia.OutgoingFrames</code>	<p>O número de quadros enviados durante <code>GetMedia</code> um determinado fluxo.</p> <p>Unidades: contagem</p>
<code>GetMedia.MillisBehindNow</code>	<p>A diferença de horário entre o timestamp atual do servidor e o timestamp do servidor do último fragmento enviado.</p> <p>Unidade: milissegundos</p>
<code>GetMedia.ConnectionErrors</code>	<p>O número de conexões que não foram estabelecidas com êxito.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>GetMedia.Success</code>	<p>1 para cada fragmento enviado com êxito; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p> </div> <p>Unidades: contagem</p>
<code>GetMediaForFragmentList.OutgoingBytes</code>	<p>O número total de bytes enviados do serviço como parte da <code>GetMediaForFragmentList</code> API para um determinado stream.</p> <p>Unidade: bytes</p>
<code>GetMediaForFragmentList.OutgoingFragments</code>	<p>O número total de fragmentos enviados do serviço como parte da <code>GetMediaForFragmentList</code> API para um determinado stream.</p> <p>Unidades: contagem</p>
<code>GetMediaForFragmentList.OutgoingFrames</code>	<p>O número total de quadros enviados do serviço como parte da <code>GetMediaForFragmentList</code> API para um determinado stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>GetMediaForFragmentList.Requests</code>	<p>O número de solicitações de <code>GetMediaForFragmentList</code> API para um determinado stream.</p> <p>Unidades: contagem</p>
<code>GetMediaForFragmentList.Success</code>	<p>1 para cada fragmento enviado com êxito; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 606 1507 1016"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>ListFragments.Latency</code>	<p>A latência da <code>ListFragments</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>ListFragments.Requests</code>	<p>O número de solicitações de <code>ListFragments</code> API para um determinado stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>ListFragments.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1508 758"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetHLSStreamingSessionURL.Latency</code>	<p>A latência da <code>GetHLSStreamingSessionURL</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetHLSStreamingSessionURL.Requests</code>	<p>O número de solicitações de <code>GetHLSStreamingSessionURL</code> API para um determinado stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>GetHLSStreamingSessionURL.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1507 758"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetHLSMasterPlaylist.Latency</code>	<p>A latência da <code>GetHLSMasterPlaylist</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetHLSMasterPlaylist.Requests</code>	<p>O número de solicitações de <code>GetHLSMasterPlaylist</code> API para um determinado stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>GetHLSMasterPlaylist.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 350 1510 760"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetHLSMediaPlaylist.Latency</code>	<p>A latência da <code>GetHLSMediaPlaylist</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetHLSMediaPlaylist.Requests</code>	<p>O número de solicitações de <code>GetHLSMediaPlaylist</code> API para um determinado stream.</p> <p>Unidades: contagem</p>


Métrica	Descrição
<code>GetHLSMediaPlaylist.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1507 758"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetMP4InitFragment.Latency</code>	<p>A latência da <code>GetMP4InitFragment</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetMP4InitFragment.Requests</code>	<p>O número de solicitações de <code>GetMP4InitFragment</code> API para um determinado stream.</p> <p>Unidades: contagem</p>

Métrica	Descrição
<code>GetMP4InitFragment.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1510 760" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetMP4MediaFragment.Latency</code>	<p>A latência da <code>GetMP4MediaFragment</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetMP4MediaFragment.Requests</code>	<p>O número de solicitações de <code>GetMP4MediaFragment</code> API para um determinado stream.</p> <p>Unidades: contagem</p>

Métrica	Descrição
<code>GetMP4MediaFragment.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1507 758" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetMP4MediaFragment.OutgoingBytes</code>	<p>O número total de bytes enviados do serviço como parte da <code>GetMP4MediaFragment</code> API para um determinado stream.</p> <p>Unidade: bytes</p>
<code>GetTSFragment.Latency</code>	<p>A latência da <code>GetTSFragment</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetTSFragment.Requests</code>	<p>O número de solicitações de <code>GetTSFragment</code> API para um determinado stream.</p> <p>Unidades: contagem</p>

Métrica	Descrição
<code>GetTSFfragment.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p> </div> <p>Unidades: contagem</p>
<code>GetTSFfragment.OutgoingBytes</code>	<p>O número total de bytes enviados do serviço como parte da <code>GetTSFfragment</code> API para um determinado stream.</p> <p>Unidade: bytes</p>
<code>GetDASHStreamingSessionURL.Latency</code>	<p>A latência da <code>GetDASHStreamingSessionURL</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetDASHStreamingSessionURL.Requests</code>	<p>O número de solicitações de <code>GetDASHStreamingSessionURL</code> API para um determinado stream.</p> <p>Unidades: contagem</p>

Métrica	Descrição
<code>GetDASHStreamingSessionURL.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 352 1507 758"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetDASHManifest.Latency</code>	<p>A latência da <code>GetDASHManifest</code> API exige o nome do stream fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetDASHManifest.Requests</code>	<p>O número de solicitações de <code>GetDASHManifest</code> API para um determinado stream.</p> <p>Unidades: contagem</p>

Métrica	Descrição
<code>GetDASHManifest.Success</code>	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div data-bbox="748 350 1508 760" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p></div> <p>Unidades: contagem</p>
<code>GetClip.Latency</code>	<p>A latência da GetClip API exige o nome do stream de vídeo fornecido.</p> <p>Unidade: milissegundos</p>
<code>GetClip.Requests</code>	<p>O número de solicitações de GetClip API para um determinado stream de vídeo.</p> <p>Unidades: contagem</p>

Métrica	Descrição
GetClip.Success	<p>1 para cada solicitação bem-sucedida; 0 para cada falha. O valor médio indica a taxa de sucesso.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>As falhas incluem erros 400 (usuário) e 500 (sistema). Para obter mais informações sobre como habilitar um resumo de solicitações e respostas, incluindo IDs de AWS solicitação, consulte Registro de resumo de solicitação/resposta.</p> </div> <p>Unidades: contagem</p>
GetClip.OutgoingBytes	<p>O número total de bytes enviados do serviço como parte da GetClip API para um determinado stream de vídeo.</p> <p>Unidade: bytes</p>

CloudWatch orientação de métricas

CloudWatch as métricas podem ajudar a encontrar respostas para as seguintes perguntas:

Tópicos

- [Os dados estão chegando ao serviço Amazon Kinesis Video Streams?](#)
- [Por que os dados não estão sendo ingeridos com sucesso pelo serviço Amazon Kinesis Video Streams?](#)
- [Por que os dados não podem ser lidos do serviço Amazon Kinesis Video Streams na mesma taxa em que estão sendo enviados pelo produtor?](#)
- [Por que não há nenhum vídeo no console ou por que o vídeo está sendo reproduzido com atraso?](#)
- [Qual o motivo do atraso na leitura de dados em tempo real e por que o cliente está ficando para trás em relação ao início do streaming?](#)

- [O cliente está lendo dados do stream de vídeo do Kinesis e em que taxa?](#)
- [Por que o cliente não consegue ler os dados do stream de vídeo do Kinesis?](#)

Os dados estão chegando ao serviço Amazon Kinesis Video Streams?

Métricas relevantes:

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

Itens de ação:

- Se houver uma queda nessas métricas, verifique se seu aplicativo ainda está enviando dados para o serviço.
- Verifique a largura de banda da rede. Se a largura de banda da rede for insuficiente, pode ser que esteja diminuindo a taxa segundo a qual o serviço está recebendo os dados.

Por que os dados não estão sendo ingeridos com sucesso pelo serviço Amazon Kinesis Video Streams?

Métricas relevantes:

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

Itens de ação:

- Se houver um aumento em `PutMedia.ConnectionErrors`, veja a resposta HTTP e os códigos de erro recebidos pelo cliente produtor para ver quais erros estão ocorrendo ao estabelecer a conexão.

- Se houver uma queda `PutMedia.Success` ou um aumento `PutMedia.ErrorAckCount`, veja o código de erro de `ack` nas respostas de `ack` enviadas pelo serviço para ver por que a ingestão de dados está falhando. Para obter mais informações, consulte [AckErrorCode.Values](#).

Por que os dados não podem ser lidos do serviço Amazon Kinesis Video Streams na mesma taxa em que estão sendo enviados pelo produtor?

Métricas relevantes:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Itens de ação:

- Se houver uma queda nessas métricas, verifique a largura de banda da rede de suas conexões. As conexões com baixa largura de banda podem fazer com que os dados alcancem o serviço a uma velocidade mais baixa.

Por que não há nenhum vídeo no console ou por que o vídeo está sendo reproduzido com atraso?

Métricas relevantes:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Itens de ação:

- Se houver um aumento `PutMedia.FragmentIngestionLatency` ou uma queda `PutMedia.IncomingFragments`, verifique a largura de banda da rede e se os dados ainda estão sendo enviados.

- Se houver uma lista `PutMedia.Success`, verifique os códigos de erro do Rack. Para obter mais informações, consulte [AckErrorCode.Values](#).
- Se houver um aumento em `PutMedia.FragmentPersistLatency` ou `ListFragments.Latency`, é provável que você esteja enfrentando um problema de serviço. Se a condição persistir por um longo período de tempo, verifique com seu contato de atendimento ao cliente se há algum problema com seu serviço.

Qual o motivo do atraso na leitura de dados em tempo real e por que o cliente está ficando para trás em relação ao início do streaming?

Métricas relevantes:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Itens de ação:

- Se houver um aumento `GetMedia.ConnectionErrors`, o consumidor pode estar ficando para trás na leitura do stream, devido às frequentes tentativas de se reconectar ao stream. Examine a resposta HTTP/códigos de erro retornado para a solicitação `GetMedia`.
- Se houver uma queda `GetMedia.Success`, é provável que o serviço não consiga enviar os dados ao consumidor, o que resultaria na queda da conexão, e na reconexão dos consumidores, o que resultaria em um atraso no início da transmissão.
- Se houver um aumento `GetMedia.MillisBehindNow`, analise seus limites de largura de banda para ver se você está recebendo os dados em uma taxa mais lenta devido à menor largura de banda.

O cliente está lendo dados do stream de vídeo do Kinesis e em que taxa?

Métricas relevantes:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`

- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

Itens de ação:

- Essas métricas indicam a taxa na qual os dados arquivados e em tempo real estão sendo lidos.

Por que o cliente não consegue ler os dados do stream de vídeo do Kinesis?

Métricas relevantes:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Itens de ação:

- Se houver um aumento de `GetMedia.ConnectionErrors`, veja a resposta HTTP e os códigos de erro retornados pela `GetMedia` solicitação. Para obter mais informações, consulte [AckErrorCode.Values](#).
- Se você estiver tentando ler os dados mais recentes ou ativos, verifique `PutMedia.IncomingBytes` se há dados entrando no stream para o serviço enviar aos consumidores.
- Se houver uma queda de `GetMedia.Success` ou `GetMediaForFragmentList.Success`, provavelmente é porque o serviço não consegue enviar os dados ao consumidor. Se a condição persistir por um longo período de tempo, verifique com seu contato de atendimento ao cliente se há algum problema com seu serviço.

Monitorando o Amazon Kinesis Video Streams Edge Agent com CloudWatch


Você pode monitorar o Amazon Kinesis Video Streams Edge Agent usando a CloudWatch Amazon, que coleta e processa dados brutos em métricas legíveis, quase em tempo real. Essas estatísticas são registradas por um período de 15 meses. Com essas informações históricas, você pode ter uma perspectiva melhor sobre o desempenho da sua aplicação web ou do serviço Amazon Kinesis Video Streams Edge Agent.

Para visualizar as métricas, faça o seguinte:


1. Faça login no AWS Management Console e abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação à esquerda, em Métricas, selecione Todas as métricas.
3. Escolha a guia Procurar e, em seguida, selecione o namespace EdgeRuntimeAgent personalizado.

O Amazon Kinesis Video Streams Edge Agent publica as seguintes métricas sob o namespace: EdgeRuntimeAgent

Dimensões	Estado	Descrição
Nome do stream, RecordJob	Running	<p>Publica continuamente quando RecordJob está em execução.</p> <p>Unidades: Nenhuma. "1" é publicado enquanto RecordJob estiver nesse estado.</p>
	FatalError	<p>Publica se houver um erro RecordJob fatal.</p> <p>Unidades: Nenhuma. "1" é publicado uma vez, quando esse evento ocorre.</p>

 **Note**

Consulte os registros para obter informações adicionais.

Dimensões	Estado	Descrição
	Concluído	<p>Publica quando um <code>RecordJob</code> é concluído.</p> <p>Unidades: Nenhuma. "1" é publicado uma vez, quando esse evento ocorre.</p>
Nome do stream, <code>UploadJob</code>	Running	<p>Publica continuamente quando <code>UploadJob</code> está em execução.</p> <p>Unidades: Nenhuma. "1" é publicado enquanto <code>UploadJob</code> estiver nesse estado.</p>
	FatalError	<p>Publica os erros <code>UploadJob</code> fatais.</p> <p>Unidades: Nenhuma. "1" é publicado uma vez, quando esse evento ocorre.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Consulte os registros para obter informações adicionais.</p> </div>
	Concluído	<p>Publica quando o <code>UploadJob</code> é concluído.</p> <p>Unidades: Nenhuma. "1" é publicado uma vez, quando esse evento ocorre.</p>
Nome do fluxo	<code>PercentageSpaceUsed</code>	<p>Essa é a porcentagem usada do espaço total alocado nas configurações do Amazon Kinesis Video Streams Edge Agent para mídia de gravação. Consulte the section called "LocalSizeConfig" Para mais informações.</p> <p>Unidades: Porcentagem (escala 0—1).</p>

Dimensões	Estado	Descrição
Nome da coisa	Vivo	<p>Publica a cada minuto a partir do Amazon Kinesis Video Streams Edge Agent, independentemente de qualquer configuração em execução nele.</p> <p>Isso pode ser usado para entender se o Amazon Kinesis Video Streams Edge Agent está ativo e pronto para aceitar configurações.</p> <p>Unidades: Nenhuma. "1" é publicado a cada minuto.</p>
	RecordJobs.HealthyJobCount	<p>Contagem total de trabalhos de gravação em execução e programados no Amazon Kinesis Video Streams Edge Agent.</p> <p>Unidades: Contagem.</p>
	UploadJobs.HealthyJobCount	<p>Contagem total de trabalhos de upload em execução e programados no Amazon Kinesis Video Streams Edge Agent.</p> <p>Unidades: Contagem.</p>
	RecordJobs.UnhealthyJobCount	<p>Contagem total de trabalhos de registro com erro no momento.</p> <p>Unidades: Contagem.</p>
	UploadJobs.UnhealthyJobCount	<p>Contagem total de trabalhos de upload com erro no momento.</p> <p>Unidades: Contagem.</p>
	RecordJobs.RunningJobCount	<p>Contagem total de trabalhos de registro em execução ativa.</p> <p>Unidades: Contagem.</p>
	UploadJobs.RunningJobCount	<p>Contagem total de trabalhos de upload em execução ativa.</p> <p>Unidades: Contagem.</p>

Dimensões	Estado	Descrição
	RecordJobs.EdgeConfigCount	Contagem total de configurações de registro em andamento no Amazon Kinesis Video Streams Edge Agent. Unidades: Contagem.
	UploadJobs.EdgeConfigCount	Contagem total de configurações de upload em andamento no Amazon Kinesis Video Streams Edge Agent. Unidades: Contagem.

CloudWatch orientação de métricas para o Amazon Kinesis Video Streams Edge Agent

CloudWatch as métricas podem ser úteis para encontrar respostas para as seguintes perguntas:

Tópicos

- [O Amazon Kinesis Video Streams Edge Agent tem espaço suficiente para gravar?](#)
- [O Amazon Kinesis Video Streams Edge Agent está vivo?](#)
- [Existem empregos insalubres?](#)
- [Algum trabalho precisa de intervenção externa?](#)

O Amazon Kinesis Video Streams Edge Agent tem espaço suficiente para gravar?

Métricas relevantes: PercentageSpaceUsed

Ação: Nenhuma ação é necessária.

O Amazon Kinesis Video Streams Edge Agent está vivo?

Métricas relevantes: Alive

Ação: Se em algum momento você parar de receber essa métrica, isso significa que o Amazon Kinesis Video Streams Edge Agent encontrou um ou mais dos seguintes:

- Um problema de tempo de execução do aplicativo: memória ou outra restrição de recursos, bug e assim por diante

- O AWS IoT dispositivo em que o agente está executando foi desligado, travado ou encerrado
- O AWS IoT dispositivo não tem conectividade de rede

Existem empregos insalubres?

Métricas relevantes:

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

Ação: inspecione os registros e procure a `FatalError` métrica.

- Se a `FatalError` métrica estiver presente, um erro fatal foi encontrado e você precisará reiniciar manualmente o trabalho. Inspecione os registros e corrija o problema antes de usar `StartEdgeConfigurationUpdate` para reiniciar manualmente o trabalho.
- Se a `FatalError` métrica não estiver presente, um erro transitório (não fatal) foi encontrado e o Amazon Kinesis Video Streams Edge Agent está tentando realizar o trabalho novamente.

Note

Para que o agente tente novamente um trabalho com erro fatal, use [the section called “StartEdgeConfigurationUpdate”](#)

Algum trabalho precisa de intervenção externa?

Métricas relevantes:

- `PercentageSpaceUsed`— Se isso exceder um determinado valor, o trabalho de gravação será pausado e retomado somente quando houver espaço disponível (quando a mídia sair da retenção). Você pode enviar uma configuração atualizada com uma superior `MaxLocalMediaSizeInMB` para atualizar o trabalho imediatamente.
- `RecordJob.FatalError/UploadJob.FatalError`— Investigue os registros do agente e envie a configuração novamente para que o trabalho seja retomado.

Ação: faça uma chamada de API com a configuração para reiniciar trabalhos que encontrem esse problema.

Registro de chamadas de API do Amazon Kinesis Video Streams com AWS CloudTrail

O Amazon Kinesis Video Streams AWS CloudTrail trabalha com, um serviço que fornece um registro das ações realizadas por um usuário, uma função AWS service (Serviço da AWS) ou um no Amazon Kinesis Video Streams. CloudTrail captura todas as chamadas de API para o Amazon Kinesis Video Streams como eventos. As chamadas capturadas incluem chamadas do console do Amazon Kinesis Video Streams e chamadas de código para as operações da API do Amazon Kinesis Video Streams. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amazon Kinesis Video Streams. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Kinesis Video Streams, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o [Guia AWS CloudTrail do usuário](#).

Amazon Kinesis Video Streams e CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando uma atividade de evento suportada ocorre no Amazon Kinesis Video Streams, essa atividade é registrada CloudTrail em um evento junto AWS com outros eventos de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos do Amazon Kinesis Video Streams, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros Serviços da AWS para analisar e agir com base nos dados do evento coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- [Visão Geral para Criar uma Trilha](#)

- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

O Amazon Kinesis Video Streams oferece suporte ao registro das seguintes ações como CloudTrail eventos em arquivos de log:

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado
- Se a solicitação foi feita por outro AWS service (Serviço da AWS).

Para obter mais informações, consulte o elemento [CloudTrailUserIdentity](#).

Exemplo: entradas do arquivo de log do Amazon Kinesis Video Streams

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contém uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte, e inclui

informações sobre a ação solicitada, data e hora da ação, parâmetros de solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a [CreateStream](#)ação.

```
{
  "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": "b7acfd0-6ca9-4ee1-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"  
  }  
]  
}
```

Cotas do serviço Kinesis Video Streams

O Kinesis Video Streams tem as seguintes cotas de serviço:

Important

As cotas de serviço a seguir são flexíveis, que podem ser atualizadas com o envio de um ticket de suporte, ou rígidas [h], que não podem ser aumentadas. Você verá [s] e [h] ao lado da cota de serviço individual nas tabelas abaixo.

Controle as cotas de serviço de API do plano


A seção a seguir descreve as cotas de serviço para APIs do plano de controle. TPS significa transações por segundo.

Quando um limite de solicitações no nível da conta ou do recurso é atingido, um é lançado.

`ClientLimitExceededException`

Controle as cotas de serviço de API do plano

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
the section called "CreateStream"	50 TPS [s]	10.000 streams por conta (s) nas regiões Leste dos EUA (Norte da Virgínia) e Oeste dos EUA (Oregon). 5000 streams por conta (s) em todas	N/D	Essa API pode ser chamada por Dispositivos, CLIs, acesso orientado por SDK e pelo console. Somente uma chamada de API é bem-sucedida se o streaming ainda não existir.

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
		<p>as outras regiões suportadas.</p> <div data-bbox="591 478 792 1850" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note</p> <p>Esse limite pode ser aumentado para até 100.000 (ou mais) streams por conta [s]. Faça login no AWS Management Console em https://console.aws.amazon.com/</p> </div>		

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
		<p>envie um caso de aumento de limite de serviço para o Kinesis Video Streams solicitar um aumento desse limite.</p>		
the section called “DeleteEdgeConfiguration”	10 TPS [h]	N/D	1 TPS [h]	
the section called “DeleteStream”	50 TPS [h]	N/D	5 TPS [h]	

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
the section called “Describe EdgeConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “Describe ImageGenerationConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “Describe MappedResourceConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “Describe NotificationConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “Describe Stream”	300 TPS [h]	N/D	5 TPS [h]	

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
the section called “GetDataEndpoint”	300 TPS [h]	N/D	5 TPS [h]	Chamado a cada 45 minutos para atualizar o token de streaming para a maioria dos casos de uso de PutMedia/GetMedia. A captura de endpoints de dados será segura se o aplicativo recarregá-los em caso de falha.
the section called “ListEdgeAgentConfigurations”	50 TPS [h]	N/D	N/D	
the section called “ListStreams”	50 TPS [h]	N/D	N/D	
the section called “ListTagsForStream”	50 TPS [h]	N/D	5 TPS [h]	
the section called “StartEdgeConfigurationUpdate”	10 TPS [h]	N/D	1 TPS [h]	
the section called “TagStream”	50 TPS [h]	N/D	5 TPS [h]	

API	Limite da conta: Solicitação	Limite da conta: Streams	Limite do nível de streaming	Exceções e notas relevantes
the section called “UntagStream”	50 TPS [h]	N/D	5 TPS [h]	
the section called “UpdateDataRetention”	50 TPS [h]	N/D	5 TPS [h]	
the section called “UpdateImageGenerationConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “UpdateNotificationConfiguration”	50 TPS [h]	N/D	5 TPS [h]	
the section called “UpdateStream”	50 TPS [h]	N/D	5 TPS [h]	

Cotas de serviços de API de mídia e mídia arquivada

A seção a seguir descreve as cotas de serviço para APIs de mídia e mídia arquivada.

Quando um limite de solicitações no nível da conta ou do recurso é atingido, um é lançado.

`ClientLimitExceededException`

Quando um limite de nível de conexão é atingido, é lançada uma

`ConnectionLimitExceededException`.


Os seguintes erros ou confirmações são lançados quando um limite de fragmento é atingido:


- Um `ACK MIN_FRAGMENT_DURATION_REACHED` é retornado para um fragmento abaixo da duração mínima.
- Uma confirmação `MAX_FRAGMENT_DURATION_REACHED` é retornado para um fragmento acima da duração máxima.
- Uma confirmação `MAX_FRAGMENT_SIZE` é retornada para um fragmento acima do tamanho máximo de dados.
- Uma exceção `FragmentLimitExceeded` é lançada se o limite de um fragmento for atingido em uma operação `GetMediaForFragmentList`.

Cotas de serviço da API de plano de dados

API	Limite do nível de streaming	Limite de nível de conexão	Limite de largura de banda	Limite de nível de fragmento	Exceções e notas relevantes
the section called "PutMedia"	5 TPS [h]	1 [s]	12,5 MB/segundo ou 100 Mbps [s] por stream	<ul style="list-style-type: none"> • Duração mínima de fragmento : 1 segundo [h] • Duração máxima do fragmento : 20 segundos [h] 	Uma solicitação <code>PutMedia</code> típica contém dados para vários segundos, resultando em um TPS por streaming menor. Se houver várias conexões simultâneas que excedam as cotas, a última conexão será aceita.

API	Limite do nível de streaming	Limite de nível de conexão	Limite de largura de banda	Limite de nível de fragmento	Exceções e notas relevantes
				<ul style="list-style-type: none"> • Tamanho máximo de fragmento : 50 MB [h] • Número máximo de trilhas: 3 [s] • Máximo de fragmentos enviados por segundo: 5 [h] • Limite máximo de metadados do fragmento : 10 tags [h] 	
the section called “GetClip”	N/D	N/D	Limite de tamanho de 100 MB [h]	Número máximo de fragmentos: 200 [h]	

API	Limite do nível de streaming	Limite de nível de conexão	Limite de largura de banda	Limite de nível de fragmento	Exceções e notas relevantes
the section called “GetDASHStreamingSessionURL”	25 FPS [h]	N/D	N/D	N/D	
the section called “GetHLSStreamingSessionURL”	25 FPS [h]	N/D	N/D	N/D	
the section called “GetImages”	N/D	N/D	100 MB [h]	N/D	<p>O número máximo de imagens por solicitação é 100 [h].</p> <div data-bbox="1133 1083 1508 1543" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>O valor mínimo <code>SamplingInterval</code> é de 200 milissegundos (ms), ou seja, 5 imagens por segundo.</p> </div>

API	Limite do nível de streaming	Limite de nível de conexão	Limite de largura de banda	Limite de nível de fragmento	Exceções e notas relevantes
the section called “GetMedia”	5 TPS [h]	3 [s]	25 MB/s ou 200 Mbps [s]	Máximo de fragmentos enviados por segundo: 6 [h]	<p>Um cliente consumidor exclusivo não deve precisar de mais do que dois ou três TPS porque, depois que a conexão for estabelecida, o aplicativo deverá ler continuamente.</p> <p>Se um fragmento típico tiver aproximadamente 5 MB, esse limite significa aproximadamente 75 MBps por stream de vídeo do Kinesis. Esse streaming teria uma taxa de bits de saída de 2 vezes a taxa de bits de entrada máxima dos streamings.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>GetMedia não é usado para reprodução de HLS/DASH.</p> </div>

API	Limite do nível de streaming	Limite de nível de conexão	Limite de largura de banda	Limite de nível de fragmento	Exceções e notas relevantes
the section called “GetMediaForFragmentList”	N/D	5 [s]	25 MB/s ou 200 Mbps [s]	Número máximo de fragmentos: 1.000 [h]	Cinco aplicativos de consumo baseados em fragmentos podem ser invocados simultaneamente. GetMediaForFragmentList Outras conexões de são rejeitadas.

Cotas de serviço da API de protocolo de reprodução de vídeo

API	Limite de nível de sessão	Limite de nível de fragmento
Obtenha o Dash ManifestPlaylist	5 TPS [h]	Número máximo de fragmentos por playlist: 5000 [h]
Obtenha HLS MasterPlaylist	5 TPS [h]	N/D
Obtenha HLS MediaPlaylist	5 TPS [h]	Número máximo de fragmentos por playlist: 5000 [h]
Obtenha MP4 InitFragment	5 TPS [h]	N/D
Obtenha MP4 MediaFragment	20 FPS [h]	N/D
GetTSFragment	20 FPS [h]	N/D

Cotas de fragmentação de metadados e de mídia fragmentada

As [APIs do Kinesis Video Streams para acessar](#) mídias arquivadas são limitadas com base no número de fragmentos solicitados e não no número de chamadas de API. As APIs têm uma taxa limitada pelo número de metadados de fragmentos e pelo número de mídias de fragmentos solicitadas. Os metadados do fragmento e as cotas de mídia do fragmento são aplicados por

fluxo. Em outras palavras, as solicitações de metadados ou mídias de fragmentos em um stream não se aplicam às cotas de outro stream. No entanto, em um determinado stream, cada cota é compartilhada entre várias APIs. Isso significa que, para um determinado stream, as solicitações de fragmentos em diferentes APIs consomem da mesma cota. Quando os metadados do fragmento ou a cota de mídia do fragmento de um stream são excedidos, a API retorna a `ClientLimitExceededException`. As tabelas a seguir mostram como as APIs consomem de cada um dos dois tipos de cota. Para a segunda coluna dessas tabelas, suponha que, se um stream tiver uma cota N, isso significa que as APIs têm N pontos para consumir desse tipo de cota para esse stream. A `GetClip` API aparece nas duas tabelas.

Fragmente o consumo da cota de metadados

API	Número de pontos de cota consumidos por solicitação	Cota compartilhada (N)
the section called “ListFragments”	Valor do <code>MaxResults</code> parâmetro	10000 pontos de cota por segundo, por stream [h]
the section called “GetClip”	Número de fragmentos no clipe resultante	
GetHLSMediaPlaylist	Valor do <code>MaxMediaPlaylistFragmentResults</code> parâmetro	
GetDASHManifest	Valor do <code>MaxManifestFragmentResults</code> parâmetro	
the section called “GetImages”	Valor de 400 + número máximo de imagens solicitadas	

Fragmentar o consumo da cota de mídia

API	Número de pontos de cota consumidos por solicitação	Cota compartilhada (N)
the section called “GetMediaForFragmentList”	Número de fragmentos no parâmetro Fragments	500 pontos de cota por segundo, por stream [h]
the section called “GetClip”	Número de fragmentos no clipe resultante	
GetMP4MediaFragment	1	
GetTSFragment	1	
the section called “GetImages”	Número máximo de imagens solicitadas	

Por exemplo, com uma cota de 500 fragmentos de mídia por segundo, os seguintes padrões de chamada para um determinado stream são suportados:

- 5 solicitações por segundo GetClip com 100 fragmentos em cada clipe.
- 100 solicitações por segundo GetClip com 5 fragmentos em cada clipe.
- 2 solicitações por segundo para GetClip 100 fragmentos em cada clipe e 3 solicitações por segundo para GetMediaForFragmentList cada clipe.
- 400 solicitações por segundo para GetMP4MediaFragment e 100 solicitações por segundo para GetTSFragment.

Essas cotas têm uma implicação importante em relação ao número de sessões HLS e MPEG-DASH que podem ser suportadas por stream. Não há limite para o número de sessões HLS e DASH que podem ser usadas pelos reprodutores de mídia em um determinado momento. Portanto, é importante que o aplicativo de reprodução não permita que muitas sessões sejam usadas simultaneamente. Os dois exemplos a seguir descrevem como determinar o número de sessões de reprodução simultâneas que podem ser suportadas:

Exemplo 1: transmissão ao vivo

Em um cenário de transmissão ao vivo com HLS com fragmentos de 1 segundo de duração, uma faixa de áudio e vídeo e `MaxMediaPlaylistFragmentResults` configurada para cinco, um reprodutor de mídia normalmente faz duas chamadas `GetHLSMediaPlaylist` por segundo. Uma chamada é para os metadados de vídeo mais recentes e outra para os metadados de áudio correspondentes. As duas chamadas consomem cinco pontos de cota de metadados de fragmentos cada. Ele também faz duas chamadas `GetMP4MediaFragment` por segundo: uma chamada para o vídeo mais recente e outra para o áudio correspondente. Cada chamada consome um único fragmento de token de mídia, portanto, dois tokens são consumidos no total.

Nesse cenário, até 250 sessões de reprodução simultâneas podem ser suportadas. Com 250 sessões, esse cenário consome 2.500 pontos de cota de metadados de fragmento por segundo (bem abaixo da cota de 10.000) e 500 pontos de cota de mídia de fragmentos por segundo.

Exemplo 2: reprodução sob demanda

Em um cenário de reprodução sob demanda de um evento passado com MPEG-DASH, uma faixa de áudio e vídeo `MaxManifestFragmentResults` definida como 1.000, um media player normalmente liga uma `GetDASHManifest` vez no início da sessão (consumindo 1.000 pontos de cota de metadados de fragmento) e chama a uma taxa de até 5 vezes por segundo (consumindo 5 pontos de cota de mídia de fragmento) até que todos os fragmentos sejam carregados. `GetMP4MediaFragment` Nesse cenário, até 10 novas sessões podem ser iniciadas por segundo (diretamente na cota de 10.000 fragmentos de metadados por segundo) e até 100 sessões podem carregar ativamente a mídia fragmentada a uma taxa de 5 por segundo (diretamente na cota de 500 fragmentos de mídia por segundo).

Você pode usar `ArchivedFragmentsConsumed.Metadata` e `ArchivedFragmentsConsumed.Media` monitorar o consumo de metadados de fragmentos e pontos de cota de mídia fragmentada, respectivamente. Para obter informações sobre monitoramento, consulte [Monitorar](#).

Fragmentar cotas de metadados

As cotas de serviço a seguir se aplicam à adição de metadados de fragmentos a fragmentos em um stream de vídeo do Kinesis:

- É possível colocar até 10 itens de metadados no início de um fragmento.
- O nome de metadados do fragmento pode ter até 128 bytes.
- O valor de metadados do fragmento pode ter até 256 bytes.

- O nome de um metadado de fragmento não pode começar com a string ""AWS. Se um item de metadados assim é adicionado, o método `putFragmentMetadata` no PIC retornará um erro `STATUS_INVALID_METADATA_NAME` (código de erro `0x52000077`). O aplicativo pode, então, ignorar o erro (o PIC não adiciona o item de metadados) ou responder ao erro.

Tags de transmissão

Esses pares de chave-valor de metadados se aplicam ao recurso geral do Kinesis Video Streams, não aos fragmentos individuais contidos em um stream de vídeo do Kinesis.

Cada stream de vídeo do Kinesis suporta até 50 tags.

Consulte [the section called "TagStream"](#) as restrições sobre as chaves e valores das tags de stream.

Solução de problemas do Kinesis Video Streams

Use as informações a seguir para solucionar problemas comuns encontrados com o Amazon Kinesis Video Streams.

Tópicos

- [Problemas gerais](#)
- [Problemas de API](#)
- [Problemas de HLS](#)
- [Problemas de Java](#)
- [Problemas com a Biblioteca do Produtor](#)
- [Problemas na biblioteca do Stream Parser](#)
- [Problemas de rede](#)

Problemas gerais

Esta seção descreve problemas gerais que você pode encontrar ao trabalhar com o Kinesis Video Streams.

Problemas

- [Latência muito alta](#)

Latência muito alta

A latência pode ser causada pela duração dos fragmentos enviados ao serviço Kinesis Video Streams. Uma forma de reduzir a latência entre o produtor e o serviço é configurar o pipeline de mídia para produzir durações de fragmento mais curtas.

Para reduzir o número de quadros enviados em cada fragmento, reduza o seguinte valor `emkinesis_video_gstreamer_sample_app.cpp`:

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512, NULL);
```

Note

As latências são superiores no navegador do Mozilla Firefox devido à implementação interna da renderização de vídeo.

Problemas de API

Esta seção descreve os problemas de API que você pode encontrar ao trabalhar com o Kinesis Video Streams.

Problemas

- [Erro: "opções desconhecidas"](#)
- [Erro: "Unable to determine service/operation name to be authorized" \(Não foi possível determinar o nome do serviço/operação a ser autorizado\)](#)
- [Erro: "Failed to put a frame in the stream" \(Falha ao inserir um quadro no streaming\)](#)
- [Erro: "O serviço fechou a conexão antes do recebimento final AckEvent"](#)
- [Erro: "STATUS_STORE_OUT_OF_MEMORY"](#)

Erro: "opções desconhecidas"

GetMedia e GetMediaForFragmentList podem apresentar falha com o seguinte erro:

```
Unknown options: <filename>.mkv
```

Esse erro ocorre se você configurou o AWS CLI com um output tipo dejson. Reconfigure o AWS CLI com o tipo de saída padrão (none). Para obter informações sobre como configurar o AWS CLI, consulte [configure](#) na Referência de AWS CLI Comandos.

Erro: "Unable to determine service/operation name to be authorized" (Não foi possível determinar o nome do serviço/operação a ser autorizado)

GetMedia pode falhar com o erro a seguir:

```
Unable to determine service/operation name to be authorized
```

Esse erro pode ocorrer se o endpoint não for especificado corretamente. Ao obter o endpoint, inclua o seguinte parâmetro na `GetDataEndpoint` chamada, dependendo da API a ser chamada:

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

Erro: "Failed to put a frame in the stream" (Falha ao inserir um quadro no streaming)

`PutMedia` pode falhar com o erro a seguir:

```
Failed to put a frame in the stream
```

Esse erro pode ocorrer se não houver conectividade ou permissões para o serviço. Execute o seguinte no AWS CLI e verifique se as informações do stream podem ser recuperadas:

```
aws kinesismedia describe-stream --stream-name StreamName --endpoint https://
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Se a chamada falhar, consulte [Solução de problemas de AWS CLI erros](#) para obter mais informações.

Erro: "O serviço fechou a conexão antes do recebimento final AckEvent"

`PutMedia` pode falhar com o erro a seguir:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was
received
```

Esse erro pode ocorrer se `PushbackInputStream` for implementado inadequadamente. Verifique se os `unread()` métodos foram implementados corretamente.

Erro: "STATUS_STORE_OUT_OF_MEMORY"

`PutMedia` pode falhar com o erro a seguir:

```
The content store is out of memory.
```


Esse erro ocorre quando o armazenamento de conteúdo não está alocado com dimensões suficientes. Para aumentar o tamanho do armazenamento de conteúdo, aumente o valor de `StorageInfo.storageSize`. Para ter mais informações, consulte [StorageInfo](#).

Problemas de HLS

Se o stream de vídeo não for reproduzido corretamente, consulte [the section called “Solução de problemas de HLS”](#).

Problemas de Java

Esta seção descreve como solucionar problemas comuns de Java encontrados ao trabalhar com o Kinesis Video Streams.

Problemas

- [Habilitar logs do Java](#)

Habilitar logs do Java

Para solucionar problemas com amostras e bibliotecas de Java, é útil ativar e examinar os registros de depuração. Para habilitar logs de depuração, faça o seguinte:

1. Adicione `log4j` ao arquivo `pom.xml` no nó `dependencies`:

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. No diretório `target/classes`, crie um arquivo denominado `log4j.properties` com o conteúdo a seguir:

```
# 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
```

Os logs de depuração são impressos no console do IDE.

Problemas com a Biblioteca do Produtor

Esta seção descreve problemas que você pode encontrar ao trabalhar com o [Bibliotecas de produtores](#).

Problemas

- [Não é possível compilar o Producer SDK](#)
- [O streaming de vídeo não aparece no console](#)
- [Erro: "O token de segurança incluído na solicitação é inválido" ao fazer streaming de dados usando o aplicativo de demonstração GStreamer](#)
- [Erro: "falha ao enviar quadro ao cliente de vídeo do Kinesis"](#)
- [O aplicativo GStreamer é interrompido com a mensagem "streaming interrompido, motivo não negociado" no OS X.](#)
- [Erro: "Falha ao alocar heap" ao criar o cliente de vídeo Kinesis na demonstração do GStreamer no Raspberry Pi](#)
- [Erro: "Instrução ilegal" ao executar demonstração do GStreamer no Raspberry Pi](#)
- [A câmera não carrega no Raspberry Pi](#)
- [Não é possível encontrar a câmera no macOS High Sierra](#)
- [Arquivo jni.h não encontrado durante a compilação no macOS High Sierra](#)
- [Erros de Curl durante a execução do aplicativo de demonstração GStreamer](#)
- [Declaração de time stamp/intervalo no tempo de execução no Raspberry Pi](#)
- [Declaração em gst_value_set_fraction_range_full no Raspberry Pi](#)
- [Erro STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA \(0x3200000d\) no Android](#)
- [A duração máxima do fragmento foi atingida \(erro\)](#)
- [Erro "Invalid thing name passed" ao usar a autorização da IoT](#)

Não é possível compilar o Producer SDK

Verifique se as bibliotecas necessárias estão no caminho. Para verificá-las, use o seguinte comando:

```
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
```

O streaming de vídeo não aparece no console

Para exibir o fluxo de vídeo no console, ele deve ser codificado no formato AvCC usando H.264. Se o stream não for exibido, verifique o seguinte:

- Seus [Sinalizadores de adaptação do NAL](#) estão definidos como `NAL_ADAPTATION_ANNEXB_NALS` | `NAL_ADAPTATION_ANNEXB_CPD_NALS` se o stream original estiver no formato Annex-B. Este é o valor padrão no construtor `StreamDefinition`.
- Você está fornecendo os dados privados do codec corretamente. Para H.264, esse é o conjunto de parâmetros de sequência (SPS) e o conjunto de parâmetros de imagem (PPS). Dependendo de sua origem de mídia, esses dados podem ser recuperados da origem de mídia separadamente ou codificados no quadro.

Muitos streams elementares estão no seguinte formato, em que Ab é o código de início de Annex-B (001 ou 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)
```

O CPD (Codec Private Data), se o H.264 estiver no fluxo como SPS e PPS, pode ser adaptado ao formato AvCC. A menos que o pipeline de mídia forneça o CPD separadamente, o aplicativo pode extrair o CPD do quadro procurando o primeiro quadro Idr (que deve conter o SPS e o PPS), extrair os dois NALUs (que serão Ab (Sps) Ab (Pps)) e configurá-lo no CPD in. `StreamDefinition`

Erro: "O token de segurança incluído na solicitação é inválido" ao fazer streaming de dados usando o aplicativo de demonstração GStreamer

Se esse erro ocorrer, isso significa que existe um problema com suas credenciais. Verifique o seguinte:

- Se estiver usando credenciais temporárias, você precisa especificar o token da sessão.
- Verifique se suas credenciais temporárias expiraram.
- Verifique se você tem os direitos apropriados configurados.
- No macOS, verifique se você tem credenciais armazenadas em cache no Keychain.

Erro: "falha ao enviar quadro ao cliente de vídeo do Kinesis"

Esse erro ocorre, pois os time stamps não estão definidos corretamente no streaming de origem. Faça o seguinte:

- Use o exemplo de SDK mais recente, que pode ter uma atualização que corrige o problema.
- Defina o fluxo de alta qualidade para uma taxa de bits mais alta e corrija qualquer instabilidade no fluxo de origem, se a câmera suportar isso.

O aplicativo GStreamer é interrompido com a mensagem "streaming interrompido, motivo não negociado" no OS X.

O streaming pode ser interrompido no OS X com a seguinte mensagem:

```
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)
```

Uma possível solução alternativa para isso é remover os parâmetros de taxa de quadros da `gst_caps_new_simple` chamada em: `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",
```

```
1, 30, 1,  
"width", GST_TYPE_INT_RANGE, 320, 1920,  
"height", GST_TYPE_INT_RANGE, 240, 1080,  
"framerate", GST_TYPE_FRACTION_RANGE, 0,  
NULL);
```

Erro: "Falha ao alocar heap" ao criar o cliente de vídeo Kinesis na demonstração do GStreamer no Raspberry Pi

O aplicativo de exemplo GStreamer tenta alocar 512 MB de RAM, quantidade que talvez você não tenha disponível no sistema. Você pode reduzir essa alocação, diminuindo o seguinte valor em `KinesisVideoProducer.cpp`:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

Erro: "Instrução ilegal" ao executar demonstração do GStreamer no Raspberry Pi

Se você encontrar o seguinte erro ao executar a demonstração do GStreamer, verifique se você compilou o aplicativo para a versão correta do seu dispositivo. (Por exemplo, verifique se você não está compilando para o Raspberry Pi 3 quando estiver executando o Raspberry Pi 2.)

```
INFO - Initializing curl.  
Illegal instruction
```

A câmera não carrega no Raspberry Pi

Para verificar se a câmera está carregada, execute o seguinte:

```
ls /dev/video*
```

Se nada for encontrado, execute o seguinte:

```
vcgencmd get_camera
```

A saída deve ser semelhante à seguinte:

```
supported=1 detected=1
```

Se o driver não detectar a câmera, faça o seguinte:

1. Verifique a configuração da câmera física e se ela está conectada corretamente.
2. Execute os comandos a seguir para atualizar o firmware:

```
sudo rpi-update
```

3. Reinicie o dispositivo.
4. Execute os comandos a seguir para carregar o driver:

```
sudo modprobe bcm2835-v4l2
```

5. Verifique se a câmera foi detectada:

```
ls /dev/video*
```

Não é possível encontrar a câmera no macOS High Sierra

No macOS High Sierra, o aplicativo de demonstração não consegue encontrar a câmera quando existe mais de uma câmera disponível.

Arquivo jni.h não encontrado durante a compilação no macOS High Sierra

Para resolver esse erro, atualize sua instalação do Xcode para a versão mais recente.

Erros de Curl durante a execução do aplicativo de demonstração GStreamer

Para resolver erros de curl durante a execução do aplicativo de demonstração GStreamer, copie [este arquivo de certificado](#) em `/etc/ssl/cert.pem`.

Declaração de time stamp/intervalo no tempo de execução no Raspberry Pi

Se uma declaração de time stamp/intervalo ocorrer no tempo de execução, atualize o firmware e reinicie o dispositivo:

```
sudo rpi-update  
$ sudo reboot
```

Declaração em `gst_value_set_fraction_range_full` no Raspberry Pi

A seguinte declaração será exibida se o serviço `uv4l` estiver em execução:

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
denominator_end) < 0' failed
```

Se isso ocorrer, interrompa o serviço `uv4l` e reinicie o aplicativo.

Erro `STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA` (0x3200000d) no Android

O seguinte erro será exibido se o [Sinalizadores de adaptação do NAL](#) estiver incorreto para o streaming de mídia:

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Se esse erro ocorrer, forneça o sinalizador `.withNALAdaptationFlags` correto para a mídia (por exemplo, `NAL_ADAPTATION_ANNEXB_CPD_NALS`). Forneça esse sinalizador na seguinte linha do [Biblioteca de produtores do Android](#) :

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/AmazonKinesisVideoDemoApp.java#L169>
[StreamConfigurationFragment](#)

A duração máxima do fragmento foi atingida (erro)

Esse erro ocorre quando um fragmento de mídia em um streaming excede o limite máximo de duração do fragmento. Veja o limite máximo de duração do fragmento na [the section called “Cotas de serviços de API de mídia e mídia arquivada”](#) seção.

Para resolver esse problema, tente o seguinte:

- Se você estiver usando uma webcam/câmera USB, execute uma das seguintes ações:
 - Se você estiver usando fragmentação baseada em quadros-chave, configure o codificador para fornecer quadros-chave em 10 segundos.
 - Se você não estiver usando a fragmentação baseada em quadros-chave, ao definir o stream em [Etapa 2: escrever e examinar o código](#), defina o limite máximo de duração do fragmento para um valor inferior a 10 segundos.

- Se você estiver usando codificadores de software (como x264) no pipeline do GStreamer, você pode definir o key-int-max atributo para um valor em 10 segundos. Por exemplo, key-int-max defina como 60, com fps definido como 30, para ativar quadros-chave a cada 2 segundos.
- Se você estiver usando uma câmera RPI, defina o atributo keyframe-interval para ser inferior a 10 segundos.
- Se você estiver usando uma câmera IP (RTSP), defina o tamanho do GOP para 60.

Erro "Invalid thing name passed" ao usar a autorização da IoT

Para evitar esse erro (HTTP Error 403: Response: {"message":"Invalid thing name passed"}) ao usar credenciais de IoT para autorização, certifique-se de que o valor de stream-name (um parâmetro obrigatório do kvssink elemento) seja idêntico ao valor de iot-thingname. Para ter mais informações, consulte [Referência de parâmetros do elemento GStreamer](#).

Problemas na biblioteca do Stream Parser

Esta seção descreve problemas que você pode encontrar ao trabalhar com o [Biblioteca do analisador de streaming](#).

Problemas

- [Não é possível acessar um quadro único do streaming](#)
- [Erro na decodificação de fragmentos](#)

Não é possível acessar um quadro único do streaming

Para acessar um único quadro de uma fonte de streaming em seu aplicativo consumidor, verifique se o stream contém os dados privados corretos do codec. Para obter mais informações sobre o formato dos dados em um stream, consulte [Modelo de dados](#).

[Para saber como usar dados privados do codec para acessar um quadro, consulte o seguinte arquivo de teste no GitHub site: KinesisVideoRendererExampleTest.java](#)

Erro na decodificação de fragmentos

Se seus fragmentos não estão codificados corretamente em um formato H.264 e no nível ao qual o navegador oferece suporte, você pode ver o seguinte erro ao executar seu streaming no console:

Fragment Decoding Error

There was an error decoding the video data. Verify that the stream contains valid H.264 content

Se isso ocorrer, verifique se:

- A resolução dos quadros corresponde com a resolução especificada nos dados privados do codec.
- O perfil H.264 e o nível dos quadros codificados correspondem com o nível e o perfil especificado nos dados privados do codec.
- O navegador é compatível com a combinação de perfil/nível. A maioria dos navegadores atuais são compatíveis com todas as combinações de nível e perfil.
- Os time stamps são precisos e estão na ordem correta, e nenhum time stamp duplicado está sendo criado.
- O aplicativo está codificando os dados do quadro usando o formato H.264.

Problemas de rede

Se você encontrar erros de conexão, como “Tempo limite de conexão” ou “Falha na conexão”, ao tentar se conectar ao Kinesis Video Streams, isso pode ser devido a restrições de intervalo de endereços IP na sua configuração de rede.

[Se sua configuração tiver restrições de intervalo de endereços IP para o Kinesis Video Streams, atualize sua configuração de rede para permitir os intervalos de endereços IP do Kinesis Video Streams.](#)

Para obter mais informações, consulte [Intervalos de AWS IP](#). Para ser notificado quando os intervalos de IP mudarem, siga o [procedimento de assinatura](#).

Histórico de documento do Amazon Kinesis Video Streams

A tabela a seguir descreve as mudanças importantes na documentação desde a última versão do Amazon Kinesis Video Streams.

- Versão mais recente da API: 29-11-2017
- Última atualização da documentação: 27 de junho de 2023

Alteração	Descrição	Data
Conexão de ponta a nuvem do agente Edge do Amazon Kinesis Video Streams	Lançamento de novos recursos. Para ter mais informações, consulte Agente Edge .	27 de junho de 2023
Primeiros passos: enviar dados para um stream de vídeo do Kinesis	Tutorial básico para enviar dados de mídia de uma câmera para um stream de vídeo do Kinesis. Para ter mais informações, consulte Enviar dados para um stream de vídeo do Amazon Kinesis .	21 de janeiro de 2019
Metadados de streaming	Você pode usar o Producer SDK para incorporar metadados em um stream de vídeo do Kinesis. Para ter mais informações, consulte Usando metadados de streaming com o Kinesis Video Streams .	28 de setembro de 2018
Registro em log do C++ Producer SDK	É possível configurar o registro em log para aplicativos do C++ Producer SDK. Para ter mais informações,	18 de julho de 2018

Alteração	Descrição	Data
	consulte Usando a geração de registros com o C++ Producer SDK .	
Streaming de vídeo HLS	Agora você pode ver um stream de vídeo do Kinesis usando HTTP Live Streaming . Para ter mais informações, consulte Reprodução do Kinesis Video Streams .	13 de julho de 2018
Streaming de uma fonte RTSP	Exemplo de aplicativo para Kinesis Video Streams que é executado em um contêiner Docker e transmite vídeo de uma fonte RTSP. Para ter mais informações, consulte RTSP e Docker .	20 de junho de 2018
Plug-in do GStreamer para o C++ Producer SDK	Mostra como criar o Biblioteca de produtores C++ para uso como um destino do GStreamer. Para ter mais informações, consulte Plugin do GStreamer - kvssink .	15 de junho de 2018
Documentação de referência dos retornos de chamadas do Producer SDK	Documentação de referência sobre os retornos de chamadas usados pelo Bibliotecas do Kinesis Video Streams Producer . Para ter mais informações, consulte Retornos de chamada do SDK do produtor .	12 de junho de 2018

Alteração	Descrição	Data
Requisitos do sistema	Documentação para os requisitos de memória e armazenamento para dispositivos Producer e SDK. Para ter mais informações, consulte Requisitos de sistema do Kinesis Video Streams .	30 de maio de 2018
CloudTrail apoio	Documentação para uso CloudTrail para monitorar o uso da API. Para ter mais informações, consulte Registro de chamadas de API do Amazon Kinesis Video Streams com AWS CloudTrail .	24 de maio de 2018
Documentação de referência sobre estruturas do Producer SDK	Documentação de referência sobre as estruturas usadas pelo Bibliotecas do Kinesis Video Streams Producer . Para obter mais informações, consulte Estruturas do SDK do produtor e Estruturas de transmissão de vídeo do Kinesis .	7 de maio de 2018
Documentação de exemplo do Renderizador	Documentação do aplicativo de exemplo Renderer, que mostra como decodificar e exibir quadros de um stream de vídeo do Kinesis. Para ter mais informações, consulte Exemplo: análise e renderização de fragmentos do Kinesis Video Streams .	15 de março de 2018

Alteração	Descrição	Data
Documentação de referência Limites do Producer SDK	Informações sobre os limites para operações no Biblioteca de produtores C++ . Para ter mais informações, consulte Limites do SDK do produtor .	13 de março de 2018
Monitoramento	Informações sobre o monitoramento de métricas e chamadas de API do Kinesis Video Streams usando Amazon e. CloudWatch AWS CloudTrail Para ter mais informações, consulte Monitoramento dos Amazon Kinesis Video Streams .	5 de fevereiro de 2018
Referência de sinalizadores de adaptação do Network Abstraction Layer (NAL)	Informações sobre a configuração de sinalizadores de adaptação do NAL ao consumir streaming de vídeo. Para ter mais informações, consulte Sinalizadores de adaptação do NAL .	15 de janeiro de 2018
Suporte do Android para streaming de vídeo	O Kinesis Video Streams agora oferece suporte ao streaming de vídeo de dispositivos Android. Para ter mais informações, consulte Biblioteca de produtores do Android .	12 de janeiro de 2018

Alteração	Descrição	Data
Documentação de exemplo do Kinesis Video	Documentação do aplicativo de exemplo do Kinesis Video, que mostra como usar o Biblioteca de análise de streaming de vídeo do Kinesis em um aplicativo. Para ter mais informações, consulte KinesisVideoExample .	9 de janeiro de 2018
Documentação liberada do Kinesis Video Streams	Esta é a versão inicial do Guia do Desenvolvedor do Amazon Kinesis Video Streams.	29 de novembro de 2017

Referência da API

As seções desse nó contêm a documentação de referência da API. Use o sumário no painel esquerdo para acessar as diferentes seções de referência da API.

Ações

As seguintes ações são suportadas pelo Amazon Kinesis Video Streams:

- [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](#)

As seguintes ações são suportadas pela Amazon Kinesis Video Streams Media:

- [GetMedia](#)
- [PutMedia](#)

As seguintes ações são suportadas pela mídia arquivada do Amazon Kinesis Video Streams:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

As seguintes ações são suportadas pelos canais de sinalização de vídeo do Amazon Kinesis:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

As seguintes ações são suportadas pelo Amazon Kinesis Video WebRTC Storage:

- [JoinStorageSession](#)

Amazon Kinesis Video Streams

As seguintes ações são suportadas pelo Amazon Kinesis Video Streams:

- [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

Serviço: Amazon Kinesis Video Streams

Cria um canal de sinalização.

CreateSignalingChannel é uma operação de natureza assíncrona.

Sintaxe da Solicitação

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelName": "string",
  "ChannelType": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelName

Um nome para o canal de sinalização que você está criando. Deve ser exclusivo para cada Conta da AWS Região da AWS e.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Exigido: Sim

ChannelType

Um tipo do canal de sinalização que você está criando. Atualmente, o único tipo de canal com suporte é SINGLE_MASTER.

Tipo: strings

Valores Válidos: SINGLE_MASTER | FULL_MESH

Obrigatório: não

SingleMasterConfiguration

Uma estrutura contendo a configuração para o tipo de SINGLE_MASTER canal.

Tipo: objeto [SingleMasterConfiguration](#)

Obrigatório: Não

Tags

Um conjunto de tags (pares de valores-chave) que você deseja associar a esse canal.

Tipo: matriz de objetos [Tag](#)

Membros da Matriz: número mínimo de 0 itens. Número máximo de 50 itens.

Obrigatório: Não

Sintaxe da Resposta

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

{
  "ChannelARN": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

ChannelARN

O Amazon Resource Name (ARN) do canal criado.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

AccountChannelLimitExceededException

Você atingiu o limite máximo de canais de sinalização ativos para isso Conta da AWS nesta região.

Código de Status HTTP: 400

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

TagsPerResourceExceededLimitException

Você excedeu o limite de tags que você pode associar ao recurso. Um stream de vídeo do Kinesis pode suportar até 50 tags.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

CreateStream

Serviço: Amazon Kinesis Video Streams

Cria um novo stream de vídeo do Kinesis.

Quando você cria um novo fluxo, o Kinesis Video Streams atribui a ele um número de versão.

Quando você altera os metadados do fluxo, o Kinesis Video Streams atualiza a versão.

CreateStream é uma operação de natureza assíncrona.

Para saber mais sobre como o serviço funciona, consulte [Como funciona](#).

Você deve ter permissões para a ação `KinesisVideo:CreateStream`.

Sintaxe da Solicitação

```
POST /createStream HTTP/1.1
Content-type: application/json

{
  "DataRetentionInHours": number,
  "DeviceName": "string",
  "KmsKeyId": "string",
  "MediaType": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[DataRetentionInHours](#)

O número de horas em que você deseja reter os dados no fluxo. O Kinesis Video Streams retém os dados em um armazenamento de dados associado ao fluxo.

O valor padrão é 0, indicando que o fluxo não persiste dados.

Quando o `DataRetentionInHours` valor é 0, os consumidores ainda podem consumir os fragmentos que permanecem no buffer do host do serviço, que tem um limite de tempo de retenção de 5 minutos e um limite de memória de retenção de 200 MB. Os fragmentos são removidos do buffer quando um dos limites é atingido.

Tipo: inteiro

Intervalo válido: valor mínimo de 0.

Obrigatório: não

DeviceName

O nome do dispositivo que está gravando no stream.

Note

Na implementação atual, o Kinesis Video Streams não usa esse nome.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: não

KmsKeyId

O ID da chave AWS Key Management Service (AWS KMS) que você deseja que o Kinesis Video Streams use para criptografar os dados do stream.

Se nenhum ID de chave for especificado, a chave padrão gerenciada pelo Kinesis Video () `AWS/kinesisvideo` será usada.

Para obter mais informações, consulte [DescribeKey](#).

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 2.048.

Padrão: . +

Obrigatório: não

MediaType

O tipo de mídia do stream. Os consumidores do fluxo podem usar essas informações ao processar o fluxo. Para obter mais informações sobre os tipos de mídia, consulte [Tipos de mídia](#). Se você optar por especificar o `MediaType`, consulte [Requisitos de nomenclatura](#) para obter diretrizes.

Exemplos de valores válidos incluem "vídeo/h264" e "vídeo/h264, áudio/aac".

Esse parâmetro é opcional; o valor padrão é `null` (ou vazio em JSON).

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `[\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*`

Obrigatório: não

StreamName

Um nome para o stream que você está criando.

O nome do stream é um identificador do stream e deve ser exclusivo para cada conta e região.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Exigido: Sim

Tags

Uma lista de tags a serem associadas ao fluxo especificado. Cada tag é um par de valores-chave (o valor é opcional).

Tipo: mapa de string para string

Entradas do mapa: número máximo de 50 itens.

Restrições de Tamanho de Chave: Tamanho mínimo de 1. O tamanho máximo é 128.

Padrão da chave: `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `[\p{L}\p{Z}\p{N}_.:/+\\-@]*`

Obrigatório: Não

Sintaxe da Resposta

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

{
  "StreamARN": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccountStreamLimitExceededException

O número de streams criados para a conta é muito alto.

Código de Status HTTP: 400

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

DeviceStreamLimitExceededException

Não implementado.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

InvalidDeviceException

Não implementado.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

TagsPerResourceExceededLimitException

Você excedeu o limite de tags que você pode associar ao recurso. Um stream de vídeo do Kinesis pode suportar até 50 tags.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DeleteEdgeConfiguration

Serviço: Amazon Kinesis Video Streams

Uma API assíncrona que exclui a configuração de borda existente de um stream, bem como a mídia correspondente do Edge Agent.

Quando você invoca essa API, o status de sincronização é definido como `DELETING`. Um processo de exclusão é iniciado, no qual as tarefas de borda ativas são interrompidas e todas as mídias são excluídas do dispositivo de borda. O tempo de exclusão varia, dependendo da quantidade total de mídia armazenada. Se o processo de exclusão falhar, o status de sincronização mudará para `DELETE_FAILED`. Você precisará tentar novamente a exclusão.

Quando o processo de exclusão for concluído com êxito, a configuração de borda não estará mais acessível.

Note

Essa API não está disponível na região AWS África (Cidade do Cabo), `af-south-1`.

Sintaxe da Solicitação

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo de vídeo. Especifique o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual excluir a configuração de borda. Especifique o `StreamName` ou `StreamARN`.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

StreamEdgeConfigurationNotFoundException

A exceção é renderizada quando o Amazon Kinesis Video Stream não consegue encontrar a configuração de borda do stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DeleteSignalingChannel

Serviço: Amazon Kinesis Video Streams

Exclui um canal de sinalização especificado. DeleteSignalingChannel é uma operação assíncrona. Se você não especificar a versão atual do canal, a versão mais recente será excluída.

Sintaxe da Solicitação

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O Amazon Resource Name (ARN) do canal de sinalização que você deseja excluir.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Exigido: Sim

CurrentVersion

A versão atual do canal de sinalização que você deseja excluir. Você pode obter a versão atual invocando as operações de ListSignalingChannels API DescribeSignalingChannel ou.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para qual stream determinado canal é mapeado.

2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

VersionMismatchException

A versão do stream que você especificou não é a versão mais recente. Para obter a versão mais recente, use a [DescribeStream](#) API.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DeleteStream

Serviço: Amazon Kinesis Video Streams

Exclui um stream de vídeo do Kinesis e os dados contidos no stream.

Esse método marca o fluxo para exclusão e torna os dados no fluxo inacessíveis imediatamente.

Para garantir que você tenha a versão mais recente do stream antes de excluí-lo, você pode especificar a versão do stream. O Kinesis Video Streams atribui uma versão a cada stream. Quando você atualiza um stream, o Kinesis Video Streams atribui um novo número de versão. Para obter a versão mais recente do stream, use a `DescribeStream` API.

Essa operação exige permissão para a ação `KinesisVideo:DeleteStream`.

Sintaxe da Solicitação

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "StreamARN": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

CurrentVersion

Opcional: a versão do stream que você deseja excluir.

Especifique a versão como uma proteção para garantir que você esteja excluindo o stream correto. Para obter a versão do stream, use a `DescribeStream` API.

Se não for especificado, somente o `CreationTime` é verificado antes de excluir o fluxo.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Obrigatório: não

StreamARN

O Amazon Resource Name (ARN) do stream que você deseja excluir.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

VersionMismatchException

A versão do stream que você especificou não é a versão mais recente. Para obter a versão mais recente, use a [DescribeStream](#) API.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeEdgeConfiguration

Serviço: Amazon Kinesis Video Streams

Descreve a configuração de borda de um stream que foi definida usando a `StartEdgeConfigurationUpdate` API e o status mais recente das tarefas de gravador e carregador do agente de borda. Use essa API para obter o status da configuração e determinar se a configuração está sincronizada com o Edge Agent. Use essa API para avaliar a integridade do Edge Agent.

Note

Essa API não está disponível na região AWS África (Cidade do Cabo), `af-south-1`.

Sintaxe da Solicitação

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do . Especifique o `StreamName` ou `StreamARN` o.

Tipo: `string`

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo cuja configuração de borda você deseja atualizar. Especifique o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

CreationTime

O carimbo de data/hora em que a configuração de borda de um riacho foi criada pela primeira vez.

Tipo: Timestamp

[EdgeAgentStatus](#)

Um objeto que contém os detalhes de status mais recentes das tarefas de gravador e carregador de um agente de ponta. Use essas informações para determinar a integridade atual de um agente de ponta.

Tipo: objeto [EdgeAgentStatus](#)

[EdgeConfig](#)

Uma descrição da configuração de borda do stream que será usada para sincronizar com o componente Edge Agent IoT Greengrass. O componente Edge Agent será executado em um dispositivo Hub IoT configurado em suas instalações.

Tipo: objeto [EdgeConfig](#)

[FailedStatusDetails](#)

Uma descrição do status de falha gerado.

Tipo: string

[LastUpdatedTime](#)

O carimbo de data/hora em que a configuração de borda de um stream foi atualizada pela última vez.

Tipo: Timestamp

[StreamARN](#)

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

[StreamName](#)

O nome do fluxo a partir do qual a configuração de borda foi atualizada.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

SyncStatus

O status mais recente da atualização da configuração de borda.

Tipo: strings

Valores Válidos: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

StreamEdgeConfigurationNotFoundException

A exceção é renderizada quando o Amazon Kinesis Video Stream não consegue encontrar a configuração de borda do stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeImageGenerationConfiguration

Serviço: Amazon Kinesis Video Streams

Obtém o `ImageGenerationConfiguration` para um determinado stream de vídeo do Kinesis.

Sintaxe da Solicitação

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Amazon Resource Name (ARN) do stream de vídeo do Kinesis do qual recuperar a configuração de geração de imagem. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual recuperar a configuração de geração de imagem. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[ImageGenerationConfiguration](#)

A estrutura que contém as informações necessárias para a entrega de imagens do Kinesis Video Stream (KVS). Se essa estrutura for nula, a configuração será excluída do fluxo.

Tipo: objeto [ImageGenerationConfiguration](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)

- [AWS SDK para Ruby V3](#)

DescribeMappedResourceConfiguration

Serviço: Amazon Kinesis Video Streams

Retorna as informações mais atuais sobre o fluxo. O `streamName` ou `streamARN` deve ser fornecido na entrada.

Sintaxe da Solicitação

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

MaxResults

O número máximo de resultados a serem retornados na resposta.

Tipo: inteiro

Intervalo válido: valor fixo de 1.

Obrigatório: não

NextToken

O token a ser fornecido em sua próxima solicitação para obter outro lote de resultados.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Obrigatório: não

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

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

{
  "MappedResourceConfigurationList": [
    {
      "ARN": "string",
      "Type": "string"
    }
  ],
  "NextToken": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[MappedResourceConfigurationList](#)

Uma estrutura que encapsula ou contém as propriedades de configuração do armazenamento de mídia.

Tipo: matriz de objetos [MappedResourceConfigurationListItem](#)

Membros da Matriz: número mínimo de 0 itens. Número máximo de 1 item.

[NextToken](#)

O token que foi usado na NextToken solicitação para buscar o próximo conjunto de resultados.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: [a-zA-Z0-9+/=]*

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeMediaStorageConfiguration

Serviço: Amazon Kinesis Video Streams

Retorna as informações mais atuais sobre o canal. Especifique o `ChannelName` ou `ChannelARN` na entrada.

Sintaxe da Solicitação

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[ChannelARN](#)

O Amazon Resource Name (ARN) do canal.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

[ChannelName](#)

O nome do canal.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

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

{
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

MediaStorageConfiguration

Uma estrutura que encapsula ou contém as propriedades de configuração do armazenamento de mídia.

Tipo: objeto [MediaStorageConfiguration](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeNotificationConfiguration

Serviço: Amazon Kinesis Video Streams

Obtém o `NotificationConfiguration` para um determinado stream de vídeo do Kinesis.

Sintaxe da Solicitação

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Amazon Resource Name (ARN) do stream de vídeo do Kinesis de onde você deseja recuperar a configuração de notificação. Você deve especificar o `StreamName` ou o `StreamArn`.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual recuperar a configuração da notificação. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

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

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  }
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

NotificationConfiguration

A estrutura que contém as informações necessárias para as notificações. Se a estrutura for nula, a configuração será excluída do fluxo.

Tipo: objeto [NotificationConfiguration](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeSignalingChannel

Serviço: Amazon Kinesis Video Streams

Retorna as informações mais atuais sobre o canal de sinalização. Você deve especificar o nome ou o Amazon Resource Name (ARN) do canal que você deseja descrever.

Sintaxe da Solicitação

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O ARN do canal de sinalização que você deseja descrever.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

ChannelName

O nome do canal de sinalização que você deseja descrever.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

ChannelInfo

Uma estrutura que encapsula os metadados e as propriedades do canal de sinalização especificado.

Tipo: objeto [ChannelInfo](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

DescribeStream

Serviço: Amazon Kinesis Video Streams

Retorna as informações mais atuais sobre o fluxo especificado. Você deve especificar o `StreamName` ou `StreamARN` o.

Sintaxe da Solicitação

```
POST /describeStream HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

StreamInfo

Um objeto que descreve o fluxo.

Tipo: objeto [StreamInfo](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetDataEndpoint

Serviço: Amazon Kinesis Video Streams

Obtém um endpoint para um fluxo especificado para leitura ou gravação. Use esse endpoint em seu aplicativo para ler o fluxo especificado (usando as `GetMediaForFragmentList` operações `GetMedia` ou) ou gravar nele (usando a `PutMedia` operação).

Note

O endpoint retornado não tem o nome da API anexado. O cliente precisa adicionar o nome da API ao endpoint retornado.

Na solicitação, especifique o fluxo por `StreamName` ou `StreamARN`.

Sintaxe da Solicitação

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json
```

```
{
  "APIName": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

APIName

O nome da ação da API para a qual obter um endpoint.

Tipo: strings

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

Obrigatório: Sim

StreamARN

O Amazon Resource Name (ARN) do stream para o qual você deseja obter o endpoint. Você deve especificar esse parâmetro ou um StreamName na solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Obrigatório: não

StreamName

O nome do stream para o qual você deseja obter o endpoint. Você deve especificar esse parâmetro ou um StreamARN na solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

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

{
  "DataEndpoint": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

DataEndpoint

O valor do endpoint. Para ler dados do stream ou gravar dados nele, especifique esse endpoint em seu aplicativo.

Tipo: string

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetSignalingChannelEndpoint

Serviço: Amazon Kinesis Video Streams

Fornecer um endpoint para o canal de sinalização especificado para enviar e receber mensagens. Essa API usa o parâmetro `SingleMasterChannelEndpointConfiguration` de entrada, que consiste nas `Role` propriedades `Protocols` e.

`Protocols` é usado para determinar o mecanismo de comunicação. Por exemplo, se você especificar `WSS` como protocolo, essa API produzirá um endpoint seguro de websocket. Se você especificar `HTTPS` como protocolo, essa API gerará um endpoint `HTTPS`. Se você especificar `WEBRTC` como protocolo, mas o canal de sinalização não estiver configurado para ingestão, você receberá o erro. `InvalidArgumentException`

`Role` determina as permissões de mensagens. Uma `MASTER` função faz com que essa API gere um endpoint que um cliente pode usar para se comunicar com qualquer um dos espectadores no canal. Uma `VIEWER` função faz com que essa API gere um endpoint que um cliente pode usar para se comunicar somente com um `MASTER`.

Sintaxe da Solicitação

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "SingleMasterChannelEndpointConfiguration": {
    "Protocols": [ "string" ],
    "Role": "string"
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O Amazon Resource Name (ARN) do canal de sinalização para o qual você deseja obter um endpoint.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

SingleMasterChannelEndpointConfiguration

Uma estrutura contendo a configuração do endpoint para o tipo de SINGLE_MASTER canal.

Tipo: objeto [SingleMasterChannelEndpointConfiguration](#)

Obrigatório: Não

Sintaxe da Resposta

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

{
  "ResourceEndpointList": [
    {
      "Protocol": "string",
      "ResourceEndpoint": "string"
    }
  ]
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[ResourceEndpointList](#)

Uma lista de pontos finais para o canal de sinalização especificado.

Tipo: matriz de objetos [ResourceEndpointListItem](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListEdgeAgentConfigurations

Serviço: Amazon Kinesis Video Streams

Retorna uma matriz de configurações de borda associadas ao Agente de Borda especificado.

Na solicitação, você deve especificar o Edge AgentHubDeviceArn.

Note

Essa API não está disponível na região AWS África (Cidade do Cabo), af-south-1.

Sintaxe da Solicitação

```
POST /listEdgeAgentConfigurations HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "HubDeviceArn": "string",
  "MaxResults": number,
  "NextToken": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

HubDeviceArn

A “coisa da Internet das Coisas (IoT)” Arn of the edge agent.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+

Exigido: Sim

MaxResults

O número máximo de configurações de borda a serem retornadas na resposta. O padrão é 5.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 10.

Obrigatório: não

NextToken

Se você especificar esse parâmetro, quando o resultado de uma `ListEdgeAgentConfigurations` operação for truncado, a chamada retornará o `NextToken` na resposta. Para obter outro lote de configurações de borda, forneça esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[EdgeConfigs](#)

Uma descrição da configuração de borda de um único fluxo.

Tipo: matriz de objetos [ListEdgeAgentConfigurationsEdgeConfig](#)

[NextToken](#)

Se a resposta for truncada, a chamada retornará esse elemento com um determinado token. Para obter o próximo lote de configurações de borda, use esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: [a-zA-Z0-9+/=]*

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)

- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListSignalingChannels

Serviço: Amazon Kinesis Video Streams

Retorna uma matriz de `ChannelInfo` objetos. Cada objeto descreve um canal de sinalização. Para recuperar somente os canais que satisfazem uma condição específica, você pode especificar a `ChannelNameCondition`

Sintaxe da Solicitação

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelNameCondition

Opcional: retorna somente os canais que satisfazem uma condição específica.

Tipo: objeto [ChannelNameCondition](#)

Obrigatório: Não

MaxResults

O número máximo de canais a serem retornados na resposta. O padrão é 500.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 10000.

Obrigatório: não

NextToken

Se você especificar esse parâmetro, quando o resultado de uma `ListSignalingChannels` operação for truncado, a chamada retornará o `NextToken` na resposta. Para obter outro lote de canais, forneça esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[ChannelInfoList](#)

Uma matriz de objetos `ChannelInfo`.

Tipo: matriz de objetos [ChannelInfo](#)

[NextToken](#)

Se a resposta for truncada, a chamada retornará esse elemento com um token. Para obter o próximo lote de streams, use esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListStreams

Serviço: Amazon Kinesis Video Streams

Retorna uma matriz de `StreamInfo` objetos. Cada objeto descreve um fluxo. Para recuperar somente fluxos que satisfaçam uma condição específica, você pode especificar a `StreamNameCondition`

Sintaxe da Solicitação

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string",
  "StreamNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

MaxResults

O número máximo de fluxos a serem retornados na resposta. O padrão é 10.000.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 10000.

Obrigatório: não

NextToken

Se você especificar esse parâmetro, quando o resultado de uma `ListStreams` operação for truncado, a chamada retornará o `NextToken` na resposta. Para obter outro lote de streams, forneça esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: [a-zA-Z0-9+/=]*

Obrigatório: não

StreamNameCondition

Opcional: retorna somente fluxos que satisfazem uma condição específica. Atualmente, você pode especificar somente o prefixo do nome de um stream como condição.

Tipo: objeto [StreamNameCondition](#)

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[NextToken](#)

Se a resposta for truncada, a chamada retornará esse elemento com um token. Para obter o próximo lote de streams, use esse token em sua próxima solicitação.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: [a-zA-Z0-9+/=]*

[StreamInfoList](#)

Uma matriz de objetos `StreamInfo`.

Tipo: matriz de objetos [StreamInfo](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListTagsForResource

Serviço: Amazon Kinesis Video Streams

Retorna uma lista de tags associadas ao canal de sinalização especificado.

Sintaxe da Solicitação

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "ResourceARN": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[NextToken](#)

Se você especificar esse parâmetro e o resultado de uma `ListTagsForResource` chamada for truncado, a resposta incluirá um token que você poderá usar na próxima solicitação para buscar o próximo lote de tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Obrigatório: não

[ResourceARN](#)

O Amazon Resource Name (ARN) do canal de sinalização para o qual você deseja listar as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

Sintaxe da Resposta

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[NextToken](#)

Se você especificar esse parâmetro e o resultado de uma `ListTagsForResource` chamada for truncado, a resposta incluirá um token que você poderá usar na próxima solicitação para buscar o próximo conjunto de tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

[Tags](#)

Um mapa de chaves e valores de tag associados ao canal de sinalização especificado.

Tipo: mapa de string para string

Entradas do mapa: número máximo de 50 itens.

Restrições de Tamanho de Chave: Tamanho mínimo de 1. O tamanho máximo é 128.

Padrão da chave: `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `[\p{L}\p{Z}\p{N}_.:/+\\-@]*`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListTagsForStream

Serviço: Amazon Kinesis Video Streams

Retorna uma lista de tags associadas ao fluxo especificado.

Na solicitação, você deve especificar o `StreamName` ou `StreamARN` o.

Sintaxe da Solicitação

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[NextToken](#)

Se você especificar esse parâmetro e o resultado de uma `ListTagsForStream` chamada for truncado, a resposta incluirá um token que você poderá usar na próxima solicitação para buscar o próximo lote de tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Obrigatório: não

[StreamARN](#)

O Amazon Resource Name (ARN) do stream para o qual você deseja listar as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream para o qual você deseja listar as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

NextToken

Se você especificar esse parâmetro e o resultado de uma `ListTags` chamada for truncado, a resposta incluirá um token que você poderá usar na próxima solicitação para buscar o próximo conjunto de tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Tamanho máximo de 512.

Padrão: `[a-zA-Z0-9+/=]*`

Tags

Um mapa de chaves e valores de tag associados ao fluxo especificado.

Tipo: mapa de string para string

Entradas do mapa: número máximo de 50 itens.

Restrições de Tamanho de Chave: Tamanho mínimo de 1. O tamanho máximo é 128.

Padrão da chave: `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `[\p{L}\p{Z}\p{N}_.:/+\\-@]*`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

InvalidResourceFormatException

O formato do `StreamARN` é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

StartEdgeConfigurationUpdate

Serviço: Amazon Kinesis Video Streams

Uma API assíncrona que atualiza a configuração de borda existente de um stream. O Kinesis Video Stream sincronizará a configuração de borda do stream com o componente Edge Agent IoT Greengrass que é executado em um dispositivo IoT Hub, configurado em suas instalações. O tempo de sincronização pode variar e depende da conectividade do dispositivo hub. Eles SyncStatus serão atualizados assim que a configuração de borda for reconhecida e sincronizada com o Edge Agent.

Se essa API for invocada pela primeira vez, uma nova configuração de borda será criada para o stream e o status de sincronização será definido como SYNCING. Você precisará esperar que o status de sincronização alcance um estado terminal, como: ou IN_SYNC_FAILED, antes de usar essa API novamente. Se você invocar essa API durante o processo de sincronização, um ResourceInUseException será lançado. A conectividade da configuração de borda do stream e do Edge Agent será testada novamente por 15 minutos. Após 15 minutos, o status passará para o SYNC_FAILED estado.

Para mover uma configuração de borda de um dispositivo para outro, use [DeleteEdgeConfiguration](#) para excluir a configuração de borda atual. Em seguida, você pode invocar StartEdgeConfigurationUpdate com um ARN de dispositivo de hub atualizado.

Note

Essa API não está disponível na região AWS África (Cidade do Cabo), af-south-1.

Sintaxe da Solicitação

```
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"
}

```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

EdgeConfig

Os detalhes da configuração de borda necessários para invocar o processo de atualização.

Tipo: objeto [EdgeConfig](#)

Obrigatório: Sim

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do . Especifique o StreamName ou StreamARN o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo cuja configuração de borda você deseja atualizar. Especifique o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
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 Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

CreationTime

O timestamp no qual a configuração de borda de um riacho foi criada pela primeira vez.

Tipo: Timestamp

EdgeConfig

Uma descrição da configuração de borda do stream que será usada para sincronizar com o componente Edge Agent IoT Greengrass. O componente Edge Agent será executado em um dispositivo Hub IoT configurado em suas instalações.

Tipo: objeto [EdgeConfig](#)

FailedStatusDetails

Uma descrição do status de falha gerado.

Tipo: string

LastUpdatedTime

A data e hora em que a configuração de borda de um stream foi atualizada pela última vez.

Tipo: Timestamp

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

O nome do fluxo a partir do qual a configuração de borda foi atualizada.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

SyncStatus

O status de sincronização atual da configuração de borda do stream. Quando você invoca essa API, o status de sincronização é definido como o SYNCING estado. Use a `DescribeEdgeConfiguration` API para obter o status mais recente da configuração de borda.

Tipo: strings

Valores Válidos: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NoDataRetentionException

A retenção de dados do Stream em horas é igual a zero.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para qual stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

TagResource

Serviço: Amazon Kinesis Video Streams

Adiciona uma ou mais tags a um canal de sinalização. Uma tag é um par de valores-chave (o valor é opcional) que você pode definir e atribuir aos AWS recursos. Se você especificar uma tag que já existe, o valor da tag será substituído pelo valor especificado na solicitação. Para obter mais informações, consulte [Usando tags de alocação de custos](#) no Guia do AWS Billing and Cost Management usuário do Cost Management.

Sintaxe da Solicitação

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[ResourceARN](#)

O Amazon Resource Name (ARN) do canal de sinalização ao qual você deseja adicionar tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

Tags

Uma lista de tags a serem associadas ao canal de sinalização especificado. Cada tag é um par de chave/valor.

Tipo: Matriz de objetos [Tag](#)

Membros da Matriz: Número mínimo de 1 item. Número máximo de 50 itens.

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

TagsPerResourceExceededLimitException

Você excedeu o limite de tags que você pode associar ao recurso. Um stream de vídeo do Kinesis pode suportar até 50 tags.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

TagStream

Serviço: Amazon Kinesis Video Streams

Adiciona uma ou mais tags a um stream. Uma tag é um par de valores-chave (o valor é opcional) que você pode definir e atribuir aos AWS recursos. Se você especificar uma tag que já existe, o valor da tag será substituído pelo valor especificado na solicitação. Para obter mais informações, consulte [Usando tags de alocação de custos](#) no Guia do AWS Billing and Cost Management usuário do Cost Management.

Você deve fornecer o StreamName ou StreamARN o.

Essa operação exige permissão para a ação KinesisVideo:TagStream.

Um stream de vídeo do Kinesis pode suportar até 50 tags.

Sintaxe da Solicitação

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O nome de recurso da Amazon (ARN) do recurso ao qual você deseja adicionar a tag ou as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream ao qual você deseja adicionar a tag ou as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: não

Tags

Uma lista de tags a serem associadas ao fluxo especificado. Cada tag é um par de valores-chave (o valor é opcional).

Tipo: mapa de string para string

Entradas do mapa: número máximo de 50 itens.

Restrições de Tamanho de Chave: Tamanho mínimo de 1. O tamanho máximo é 128.

Padrão da chave: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

InvalidResourceFormatException

O formato do `StreamARN` é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

TagsPerResourceExceededLimitException

Você excedeu o limite de tags que você pode associar ao recurso. Um stream de vídeo do Kinesis pode suportar até 50 tags.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UntagResource

Serviço: Amazon Kinesis Video Streams

Remove uma ou mais tags de um canal de sinalização. Na solicitação, especifique somente uma chave ou chaves de tag; não especifique o valor. Se você especificar uma chave de tag que não existe, ela será ignorada.

Sintaxe da Solicitação

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "TagKeyList": [ "string" ]
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ResourceARN

O Amazon Resource Name (ARN) do canal de sinalização do qual você deseja remover as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

TagKeyList

Uma lista das chaves das tags que você deseja remover.

Tipo: Matriz de strings

Membros da Matriz: Número mínimo de 1 item. Número máximo de 50 itens.

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: $^([\p{L}\p{Z}\p{N}_\p{.}:/=+\-@]^\ast)\$$

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UntagStream

Serviço: Amazon Kinesis Video Streams

Remove uma ou mais tags de um stream. Na solicitação, especifique somente uma chave ou chaves de tag; não especifique o valor. Se você especificar uma chave de tag que não existe, ela será ignorada.

Na solicitação, você deve fornecer o `StreamName` ou `StreamARN`.

Sintaxe da Solicitação

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "TagKeyList": [ "string" ]
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

StreamARN

O Amazon Resource Name (ARN) do stream do qual você deseja remover as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream do qual você deseja remover as tags.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

[TagKeyList](#)

Uma lista das chaves das tags que você deseja remover.

Tipo: Matriz de strings

Membros da Matriz: Número mínimo de 1 item. Número máximo de 50 itens.

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: ^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)\$

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

`InvalidResourceFormatException`

O formato do `StreamARN` é inválido.

Código de Status HTTP: 400

`NotAuthorizedException`

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

`ResourceNotFoundException`

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateDataRetention

Serviço: Amazon Kinesis Video Streams

Aumenta ou diminui o período de retenção de dados do stream de acordo com o valor que você especificar. Para indicar se você deseja aumentar ou diminuir o período de retenção de dados, especifique o `Operation` parâmetro no corpo da solicitação. Na solicitação, você deve especificar o `StreamName` ou `StreamARN`.

Essa operação exige permissão para a ação `KinesisVideo:UpdateDataRetention`.

A alteração do período de retenção de dados afeta os dados no fluxo da seguinte forma:

- Se o período de retenção de dados for aumentado, os dados existentes serão retidos pelo novo período de retenção. Por exemplo, se o período de retenção de dados for aumentado de uma hora para sete horas, todos os dados existentes serão retidos por sete horas.
- Se o período de retenção de dados for reduzido, os dados existentes serão retidos pelo novo período de retenção. Por exemplo, se o período de retenção de dados for reduzido de sete horas para uma hora, todos os dados existentes serão retidos por uma hora e todos os dados com mais de uma hora serão excluídos imediatamente.

Sintaxe da Solicitação

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DataRetentionChangeInHours": number,
  "Operation": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

CurrentVersion

A versão do stream cujo período de retenção você deseja alterar. Para obter a versão, chame a API `DescribeStream` ou a `ListStreams` API.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: `[a-zA-Z0-9]+`

Exigido: Sim

DataRetentionChangeInHours

O número de horas para ajustar a retenção atual. O valor especificado é adicionado ou subtraído do valor atual, dependendo do `operation`

O valor mínimo para retenção de dados é 0 e o valor máximo é 87600 (dez anos).

Tipo: inteiro

Intervalo válido: valor mínimo de 1.

Obrigatório: Sim

Operation

Indica se você deseja aumentar ou diminuir o período de retenção.

Tipo: strings

Valores Válidos: `INCREASE_DATA_RETENTION` | `DECREASE_DATA_RETENTION`

Obrigatório: Sim

StreamARN

O Amazon Resource Name (ARN) do stream cujo período de retenção você deseja alterar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream cujo período de retenção você deseja alterar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

VersionMismatchException

A versão do stream que você especificou não é a versão mais recente. Para obter a versão mais recente, use a [DescribeStream](#) API.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)

- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateImageGenerationConfiguration

Serviço: Amazon Kinesis Video Streams

Atualiza StreamInfo os ImageProcessingConfiguration campos e.

Sintaxe da Solicitação

```
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"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ImageGenerationConfiguration

A estrutura que contém as informações necessárias para a entrega das imagens do KVS. Se a estrutura for nula, a configuração será excluída do fluxo.

Tipo: objeto [ImageGenerationConfiguration](#)

Obrigatório: Não

[StreamARN](#)

O Amazon Resource Name (ARN) do stream de vídeo do Kinesis de onde você deseja atualizar a configuração de geração de imagens. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

[StreamName](#)

O nome do fluxo a partir do qual atualizar a configuração de geração de imagem. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NoDataRetentionException

A retenção de dados do Stream em horas é igual a zero.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateMediaStorageConfiguration

Serviço: Amazon Kinesis Video Streams

Associa a `SignalingChannel` a um stream para armazenar a mídia. Há dois modos de sinalização que você pode especificar:

- Se `StorageStatus` estiver ativado, os dados serão armazenados no formato `StreamARN` fornecido. Para que a ingestão de WebRTC funcione, o stream deve ter a retenção de dados ativada.
- Se `StorageStatus` estiver desativado, nenhum dado será armazenado e o `StreamARN` parâmetro não será necessário.

Important

Se `StorageStatus` estiver ativado, as conexões diretas peer-to-peer (visualizador principal) não ocorrerão mais. Os colegas se conectam diretamente à sessão de armazenamento. Você deve chamar a `JoinStorageSession` API para acionar o envio de uma oferta de SDP e estabelecer uma conexão entre um par e a sessão de armazenamento.

Sintaxe da Solicitação

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O Amazon Resource Name (ARN) do canal.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

MediaStorageConfiguration

Uma estrutura que encapsula ou contém as propriedades de configuração do armazenamento de mídia.

Tipo: objeto [MediaStorageConfiguration](#)

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NoDataRetentionException

A retenção de dados do Stream em horas é igual a zero.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para qual stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateNotificationConfiguration

Serviço: Amazon Kinesis Video Streams

Atualiza as informações de notificação de um stream.

Sintaxe da Solicitação

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

NotificationConfiguration

A estrutura que contém as informações necessárias para as notificações. Se a estrutura for nula, a configuração será excluída do fluxo.

Tipo: objeto [NotificationConfiguration](#)

Obrigatório: Não

StreamARN

O Amazon Resource Name (ARN) do stream de vídeo do Kinesis de onde você deseja atualizar a configuração de notificação. Você deve especificar o StreamName ou StreamARN o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo a partir do qual atualizar a configuração da notificação. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NoDataRetentionException

A retenção de dados do Stream em horas é igual a zero.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)

- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateSignalingChannel

Serviço: Amazon Kinesis Video Streams

Atualiza o canal de sinalização existente. Essa é uma operação assíncrona e leva tempo para ser concluída.

Se o `MessageTtlSeconds` valor for atualizado (aumentado ou reduzido), ele só se aplicará às novas mensagens enviadas por esse canal após a atualização. As mensagens existentes ainda expiraram de acordo com o `MessageTtlSeconds` valor anterior.

Sintaxe da Solicitação

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  }
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O Amazon Resource Name (ARN) do canal de sinalização que você deseja atualizar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

CurrentVersion

A versão atual do canal de sinalização que você deseja atualizar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Exigido: Sim

SingleMasterConfiguration

A estrutura que contém a configuração do SINGLE_MASTER tipo de canal de sinalização que você deseja atualizar.

Tipo: objeto [SingleMasterConfiguration](#)

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de status HTTP: 401

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para qual stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

VersionMismatchException

A versão do stream que você especificou não é a versão mais recente. Para obter a versão mais recente, use a [DescribeStream](#) API.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

UpdateStream

Serviço: Amazon Kinesis Video Streams

Atualiza os metadados do stream, como o nome do dispositivo e o tipo de mídia.

Você deve fornecer o nome do stream ou o Amazon Resource Name (ARN) do stream.

Para garantir que você tenha a versão mais recente do stream antes de atualizá-lo, você pode especificar a versão do stream. O Kinesis Video Streams atribui uma versão a cada stream. Quando você atualiza um stream, o Kinesis Video Streams atribui um novo número de versão. Para obter a versão mais recente do stream, use a `DescribeStream` API.

`UpdateStream` é uma operação assíncrona e leva tempo para ser concluída.

Sintaxe da Solicitação

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DeviceName": "string",
  "MediaType": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

CurrentVersion

A versão do stream cujos metadados você deseja atualizar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Exigido: Sim

DeviceName

O nome do dispositivo que está gravando no stream.

Note

Na implementação atual, o Kinesis Video Streams não usa esse nome.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

MediaType

O tipo de mídia do stream. Use `MediaType` para especificar o tipo de conteúdo que o stream contém para os consumidores do stream. Para obter mais informações sobre os tipos de mídia, consulte [Tipos de mídia](#). Se você optar por especificar `oMediaType`, consulte [Requisitos de nomenclatura](#).

Para reproduzir vídeo no console, você deve especificar o tipo de vídeo correto. Por exemplo, se o vídeo no stream for H.264, especifique como o `video/h264` `MediaType`

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: [\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*

Obrigatório: não

StreamARN

O ARN do stream cujos metadados você deseja atualizar.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream cujos metadados você deseja atualizar.

O nome do stream é um identificador do stream e deve ser exclusivo para cada conta e região.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceInUseException

Quando a entrada `StreamARN` ou `ChannelARN` a entrada já estiverem mapeadas para um recurso diferente do Kinesis Video Stream, ou se a `StreamARN` entrada `ChannelARN` fornecida não estiver no status Ativo, tente uma das seguintes opções: `CLOUD_STORAGE_MODE`

1. A `DescribeMediaStorageConfiguration` API para determinar para onde o stream determinado canal é mapeado.
2. A `DescribeMappedResourceConfiguration` API para determinar o canal para o qual o determinado fluxo é mapeado.
3. A `DescribeSignalingChannel` API `DescribeStream` ou para determinar o status do recurso.

Código de Status HTTP: 400

ResourceNotFoundException

O Amazon Kinesis Video Streams não consegue encontrar o stream que você especificou.

Código de Status HTTP: 404

VersionMismatchException

A versão do stream que você especificou não é a versão mais recente. Para obter a versão mais recente, use a [DescribeStream](#) API.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

Mídia do Amazon Kinesis Video Streams

As seguintes ações são suportadas pelo Amazon Kinesis Video Streams Media:

- [GetMedia](#)
- [PutMedia](#)

GetMedia

Serviço: Amazon Kinesis Video Streams Media

Use essa API para recuperar conteúdo de mídia de um stream de vídeo do Kinesis. Na solicitação, você identifica o nome do stream ou o Amazon Resource Name (ARN) do stream e a parte inicial. Em seguida, o Kinesis Video Streams retorna um stream de partes em ordem por número de fragmentos.

Note

Você deve primeiro chamar a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `GetMedia` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.

Quando você coloca dados de mídia (fragmentos) em um stream, o Kinesis Video Streams armazena cada fragmento de entrada e metadados relacionados no que é chamado de “fragmento”. Para obter mais informações, consulte [PutMedia](#). A `GetMedia` API retorna um fluxo desses fragmentos a partir do fragmento que você especifica na solicitação.

Os seguintes limites se aplicam ao usar a `GetMedia` API:

- Um cliente pode ligar `GetMedia` até cinco vezes por segundo por stream.
- O Kinesis Video Streams envia dados de mídia a uma taxa de até 25 megabytes por segundo (ou 200 megabits por segundo) durante uma sessão. `GetMedia`

Note

O código de status da resposta `GetMedia` HTTP será retornado imediatamente, mas a leitura da carga útil da resposta HTTP atingirá o tempo limite após 3 segundos se não houver fragmentos ingeridos disponíveis para reprodução.

Note

Se um erro for gerado após a invocação de uma API de mídia do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.
- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS, a equipe de suporte poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção Erros na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
  "StartSelector": {
    "AfterFragmentNumber": "string",
    "ContinuationToken": "string",
    "StartSelectorType": "string",
    "StartTimestamp": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[StartSelector](#)

Identifica a parte inicial a ser obtida do fluxo especificado.

Tipo: objeto [StartSelector](#)

Obrigatório: Sim

[StreamARN](#)

O ARN do stream de onde você deseja obter o conteúdo de mídia. Se você não especificar `streamARN`, deverá especificar `streamName`.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

[StreamName](#)

O nome do stream de vídeo do Kinesis de onde você deseja obter o conteúdo de mídia. Se você não especificar `streamName`, deverá especificar `streamARN`.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

A resposta retorna os cabeçalhos HTTP a seguir.

ContentType

O tipo de conteúdo da mídia solicitada.

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[a-zA-Z0-9_\.\-]+$`

A resposta retorna as informações a seguir como corpo HTTP.

Payload

A carga que o Kinesis Video Streams retorna é uma sequência de partes do stream especificado. Para obter mais informações sobre os fragmentos, consulte [PutMedia](#). Os trechos que o Kinesis Video Streams retorna na chamada também incluem as seguintes tags adicionais de GetMedia Matroska (MKV):

- `AWS_KINESISVIDEO_CONTINUATION_TOKEN` (string UTF-8) - Caso sua `GetMedia` chamada seja encerrada, você pode usar esse token de continuação em sua próxima solicitação para obter a próxima parte em que a última solicitação foi encerrada.
- `AWS_KINESISVIDEO_MILLIS_BEHIND_NOW` (string UTF-8) - Os aplicativos clientes podem usar esse valor de tag para determinar a distância entre o fragmento retornado na resposta e o fragmento mais recente no stream.
- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - Número do fragmento retornado no bloco.
- `AWS_KINESISVIDEO_SERVER_TIMESTAMP` - Timestamp do servidor do fragmento.
- `AWS_KINESISVIDEO_PRODUCER_TIMESTAMP` - Carimbo de data/hora do produtor do fragmento.

As seguintes tags estarão presentes se ocorrer um erro:

- `AWS_KINESISVIDEO_ERROR_CODE` - Descrição da string de um erro que causou a interrupção. `GetMedia`
- `AWS_KINESISVIDEO_ERROR_ID`: código inteiro do erro.

Os códigos de erro são os seguintes:

- 3002 - Erro ao gravar no stream
- 4000 - O fragmento solicitado não foi encontrado
- 4500 - Acesso negado para a chave KMS do stream
- 4501 - A chave KMS do stream está desativada

- 4502 - Erro de validação na chave KMS do stream
- 4503 - A chave KMS especificada no fluxo não está disponível
- 4504 - Uso inválido da chave KMS especificada no fluxo
- 4505 - Estado inválido da chave KMS especificada no fluxo
- 4506 - Não é possível encontrar a chave KMS especificada no fluxo
- 5000 - Erro interno

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

ConnectionLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de conexões de clientes.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

InvalidEndpointException

O chamador usou o endpoint errado para gravar dados em um stream. Ao receber essa exceção, o usuário deve chamar `GetDataEndpoint` com `APIName` set to `PUT_MEDIA` e usar o endpoint from response para invocar a próxima `PutMedia` chamada.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar uma operação em determinado stream ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

Código de status: 404, O fluxo com o nome fornecido não existe.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

PutMedia

Serviço: Amazon Kinesis Video Streams Media

Use essa API para enviar dados de mídia para um stream de vídeo do Kinesis.

Note

Você deve primeiro chamar a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `PutMedia` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.

Na solicitação, você usa os cabeçalhos HTTP para fornecer informações de parâmetros, por exemplo, nome do stream, timestamp e se o valor do timestamp é absoluto ou relativo a quando o produtor começou a gravar. Você usa o corpo da solicitação para enviar os dados de mídia. O Kinesis Video Streams suporta somente o formato de contêiner Matroska (MKV) para enviar dados de mídia usando essa API.

Você tem as seguintes opções para enviar dados usando essa API:

- Envie dados de mídia em tempo real: por exemplo, uma câmera de segurança pode enviar quadros em tempo real à medida que os gera. Essa abordagem minimiza a latência entre a gravação de vídeo e os dados enviados pela rede. Isso é chamado de produtor contínuo. Nesse caso, um aplicativo consumidor pode ler o stream em tempo real ou quando necessário.
- Envie dados de mídia off-line (em lotes): por exemplo, uma câmera corporal pode gravar vídeo por horas e armazená-lo no dispositivo. Posteriormente, quando você conecta a câmera à porta de encaixe, a câmera pode iniciar uma `PutMedia` sessão para enviar dados para um stream de vídeo do Kinesis. Nesse cenário, a latência não é um problema.

Ao usar essa API, observe as seguintes considerações:

- Você precisa especificar o `streamName` ou o `streamARN`, mas não os dois.
- Para poder reproduzir a mídia no console ou via HLS, a faixa 1 de cada fragmento deve conter vídeo codificado em h.264, o codec nos metadados do fragmento deve ser “V_MPEG/ISO/AVC” e os metadados do fragmento devem incluir dados privados do codec h.264 formatado em AVCC. Opcionalmente, a faixa 2 de cada fragmento deve conter áudio codificado em AAC, o codec nos metadados do fragmento deve ser “A_AAC” e os metadados do fragmento devem incluir dados privados do codec AAC.

- Talvez você ache mais fácil usar uma única PutMedia sessão de longa duração e enviar um grande número de fragmentos de dados de mídia na carga útil. Para cada fragmento recebido, o Kinesis Video Streams envia uma ou mais confirmações. Possíveis considerações de rede podem fazer com que você não receba todas essas confirmações à medida que elas são geradas.
- Você pode escolher várias PutMedia sessões consecutivas, cada uma com menos fragmentos, para garantir que você receba todas as confirmações do serviço em tempo real.

Note

Se você enviar dados para o mesmo stream em várias PutMedia sessões simultâneas, os fragmentos de mídia serão intercalados no stream. Você deve se certificar de que isso esteja OK no cenário do seu aplicativo.

Os seguintes limites se aplicam ao usar a PutMedia API:

- Um cliente pode ligar PutMedia até cinco vezes por segundo por stream.
- Um cliente pode enviar até cinco fragmentos por segundo por stream.
- O Kinesis Video Streams lê dados de mídia a uma taxa de até 12,5 MB/segundo, ou 100 Mbps durante uma sessão. PutMedia

Observe as seguintes restrições. Nesses casos, o Kinesis Video Streams envia a confirmação de erro na resposta.

- Fragmentos com códigos de tempo que se estendem por mais do que o limite máximo permitido e que contêm mais de 50 MB de dados não são permitidos.
- Fragmentos contendo mais de três faixas não são permitidos. Cada quadro em cada fragmento deve ter o mesmo número de faixa de uma das faixas definidas no cabeçalho do fragmento. Além disso, cada fragmento deve conter pelo menos um quadro para cada faixa definida no cabeçalho do fragmento.
- Cada fragmento deve conter pelo menos um quadro para cada faixa definida nos metadados do fragmento.
- O carimbo de data/hora do quadro mais antigo em um fragmento deve ser posterior ao carimbo de data/hora do quadro mais recente no fragmento anterior.

- Um fluxo MKV contendo mais de um segmento MKV ou contendo elementos MKV não permitidos (como `track*`) também resulta na confirmação de erro.

O Kinesis Video Streams armazena cada fragmento de entrada e metadados relacionados no que é chamado de “fragmento”. Os metadados do fragmento incluem o seguinte:

- Os cabeçalhos MKV fornecidos no início da solicitação `PutMedia`
- Os seguintes metadados específicos do Kinesis Video Streams para o fragmento:
 - `server_timestamp`- Registro de data e hora em que o Kinesis Video Streams começou a receber o fragmento.
 - `producer_timestamp`- Timestamp, quando o produtor começou a gravar o fragmento. O Kinesis Video Streams usa três informações recebidas na solicitação para calcular esse valor.
 - O valor do timecode do fragmento recebido no corpo da solicitação junto com o fragmento.
 - Dois cabeçalhos de solicitação: `producerStartTimeStamp` (quando o produtor começou a gravar) e `fragmentTimeCodeType` (se o timecode do fragmento na carga é absoluto ou relativo).

Em seguida, o Kinesis Video Streams `producer_timestamp` calcula o fragmento da seguinte forma:

Se `fragmentTimeCodeType` for relativo, então

`producer_timestamp` = `producerStartTimeStamp` + código de tempo do fragmento

Se `fragmentTimeCodeType` for absoluto, então

`producer_timestamp` = timecode do fragmento (convertido em milissegundos)

- Número de fragmento exclusivo atribuído pelo Kinesis Video Streams.

Note

Quando você faz a `GetMedia` solicitação, o Kinesis Video Streams retorna um stream desses trechos. O cliente pode processar os metadados conforme necessário.

Note

Essa operação está disponível somente para o AWS SDK for Java. Ele não é compatível com AWS SDKs para outros idiomas.

Note

O Kinesis Video Streams não analisa e valida os dados privados do codec durante a ingestão e o arquivamento por meio da API. PutMedia O KVS extrai e valida as informações necessárias dos dados privados do codec para o empacotamento de fragmentos MPEG-TS e MP4 ao consumir o fluxo por meio das APIs HLS.

Note

Se um erro for gerado após a invocação de uma API de mídia do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.
- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS, a equipe de suporte poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção Erros na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
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

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

FragmentTimecodeType

Você passa este valor como o cabeçalho HTTP `x-amzn-fragment-timecode-type`.

Indica se os timecodes nos fragmentos (carga útil, corpo da solicitação HTTP) são absolutos ou relativos a `producerStartTimestamp`. O Kinesis Video Streams usa essas informações para `producer_timestamp` calcular o fragmento recebido na solicitação, conforme descrito na visão geral da API.

Valores Válidos: ABSOLUTE | RELATIVE

Obrigatório: Sim

ProducerStartTimestamp

Você passa este valor como o cabeçalho HTTP `x-amzn-producer-start-timestamp`.

Esse é o timestamp do produtor no qual o produtor começou a gravar a mídia (não o timestamp dos fragmentos específicos na solicitação).

StreamARN

Você passa este valor como o cabeçalho HTTP `x-amzn-stream-arn`.

Nome de recurso da Amazon (ARN) do stream de vídeo do Kinesis em que você deseja gravar o conteúdo de mídia. Se você não especificar `streamARN`, deverá especificar `streamName`.

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

Você passa este valor como o cabeçalho HTTP `x-amzn-stream-name`.

Nome do stream de vídeo do Kinesis em que você deseja gravar o conteúdo de mídia. Se você não especificar `streamName`, deverá especificar `streamARN`.

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Corpo da Solicitação

A solicitação aceita os dados binários a seguir.

Payload

O conteúdo de mídia a ser gravado no stream de vídeo do Kinesis. Na implementação atual, o Kinesis Video Streams suporta somente o formato de contêiner Matroska (MKV) com um único segmento MKV. Um segmento pode conter um ou mais clusters.

Note

Cada cluster MKV é mapeado para um fragmento de stream de vídeo do Kinesis. Qualquer que seja a duração do cluster que você escolher, se tornará a duração do fragmento.

Sintaxe da Resposta

```
HTTP/1.1 200
```

```
Payload
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.


A resposta retorna as informações a seguir como corpo HTTP.

Payload

Depois que o Kinesis Video Streams PutMedia recebe com sucesso uma solicitação, o serviço valida os cabeçalhos da solicitação. Em seguida, o serviço começa a ler a carga e primeiro envia uma resposta HTTP 200.

Em seguida, o serviço retorna um fluxo contendo uma série de objetos JSON (`Acknowledgmentobjects`) separados por novas linhas. As confirmações são recebidas na mesma conexão na qual os dados de mídia são enviados. Pode haver muitas confirmações para uma `PutMedia` solicitação. Cada um `Acknowledgment` consiste nos seguintes pares de valores-chave:

- `AckEventType`- Tipo de evento que a confirmação representa.
 - **Armazenamento em buffer:** o Kinesis Video Streams começou a receber o fragmento. O Kinesis Video Streams envia a primeira confirmação de buffering quando o primeiro byte do fragmento de dados é recebido.
 - **Recebido:** o Kinesis Video Streams recebeu o fragmento inteiro. Se você não configurou o stream para manter os dados, o produtor pode parar de armazenar o fragmento em buffer ao receber essa confirmação.
 - **Persistente:** o Kinesis Video Streams persistiu no fragmento (por exemplo, no Amazon S3). Você receberá essa confirmação se tiver configurado o fluxo para persistir os dados. Depois de receber essa confirmação, o produtor pode parar de armazenar o fragmento em buffer.
 - **Erro:** o Kinesis Video Streams encontrou um erro ao processar o fragmento. Você pode revisar o código de erro e determinar o próximo curso de ação.
 - **Inativo:** a `PutMedia` sessão está em andamento. No entanto, o Kinesis Video Streams não está recebendo dados no momento. O Kinesis Video Streams envia essa confirmação periodicamente por até 30 segundos após os últimos dados recebidos. Se nenhum dado for recebido em 30 segundos, o Kinesis Video Streams fechará a solicitação.

 Note

Esse reconhecimento pode ajudar o produtor a determinar se a `PutMedia` conexão está ativa, mesmo que não esteja enviando dados.

- `FragmentTimecode`- Fragmente o timecode para o qual a confirmação é enviada.

O elemento pode estar ausente se `AckEventType` estiver inativo.

- `FragmentNumber`- Número do fragmento gerado pelo Kinesis Video Streams para o qual a confirmação é enviada.
- `ErrorId` `ErrorCode` - Se `AckEventType` for `Error`, esse campo fornecerá o código de erro correspondente. A seguir está a lista de IDs de erro e seus códigos de erro e mensagens de erro correspondentes:

- 4000 - STREAM_READ_ERROR - Erro ao ler o fluxo de dados.
- 4001 - MAX_FRAGMENT_SIZE_REACHED - O tamanho do fragmento é maior que o limite máximo, 50 MB, permitido.
- 4002 - MAX_FRAGMENT_DURATION_REACHED - A duração do fragmento é maior que o limite máximo permitido.
- 4003 - MAX_CONNECTION_DURATION_REACHED - A duração da conexão é maior que o limite máximo permitido.
- 4004 - FRAGMENT_TIMECODE_LESSER_THAN_PREVIOUS - O timecode do fragmento é menor que o timecode anterior (em uma chamada, você não pode enviar fragmentos fora de ordem). PutMedia
- 4005 - MORE_THAN_ALLOWED_TRACKS_FOUND - Mais de uma faixa é encontrada no MKV. (obsoleto)
- 4006 - INVALID_MKV_DATA - Falha ao analisar o fluxo de entrada como formato MKV válido.
- 4007 - INVALID_PRODUCER_TIMESTAMP - Carimbo de data/hora do produtor inválido.
- 4008 - STREAM_NOT_ACTIVE - O fluxo não existe mais (excluído).
- 4009 - FRAGMENT_METADATA_LIMIT_REACHED - Limite de metadados do fragmento atingido. Consulte a seção [Limites](#) do guia do desenvolvedor.
- 4010 - TRACK_NUMBER_MISMATCH - O número da faixa em um quadro MKV não corresponde às faixas no cabeçalho MKV.
- 4011 - FRAMES_MISSING_FOR_TRACK - O fragmento não continha nenhum quadro para pelo menos uma das faixas no cabeçalho MKV.
- 4012 - INVALID_FRAGMENT_METADATA - O nome dos metadados do fragmento não pode começar com a string. AWS_
- 4500 - KMS_KEY_ACCESS_DENIED - O acesso à chave KMS especificada do stream é negado.
- 4501 - KMS_KEY_DISABLED - A chave KMS especificada do stream está desativada.
- 4502 - KMS_KEY_VALIDATION_ERROR - A validação da chave KMS especificada do stream falhou.
- 4503 - KMS_KEY_UNAVAILABLE - A chave KMS especificada do stream não está disponível.
- 4504 - KMS_KEY_INVALID_USAGE - Uso inválido da chave KMS especificada do stream.

- 4505 - KMS_KEY_INVALID_STATE - A chave KMS especificada do stream está em um estado inválido.
- 4506 - KMS_KEY_NOT_FOUND - A chave KMS especificada do stream não foi encontrada.
- 5000 - INTERNAL_ERROR - Erro interno do serviço.
- 5001 - ARCHIVAL_ERROR - O Kinesis Video Streams não conseguiu manter fragmentos no armazenamento de dados.

Note

O produtor, ao enviar a carga útil de uma PutMedia solicitação de longa duração, deve ler a resposta para receber confirmações. Um produtor pode receber partes de confirmações ao mesmo tempo, devido ao buffer em um servidor proxy intermediário. Um produtor que deseja receber confirmações em tempo hábil pode enviar menos fragmentos em cada solicitação. PutMedia

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

ConnectionLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de conexões de clientes.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

InvalidEndpointException

O chamador usou o endpoint errado para gravar dados em um stream. Ao receber essa exceção, o usuário deve chamar `GetDataEndpoint` com `APIName` set to `PUT_MEDIA` e usar o endpoint from response para invocar a próxima `PutMedia` chamada.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar uma operação em determinado stream ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

Código de status: 404, O fluxo com o nome fornecido não existe.

Código de Status HTTP: 404

Exemplos

Formato de reconhecimento

O formato da confirmação é o seguinte:

```
{
  Acknowledgement : {
    "EventType": enum,
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
    "ErrorId" : String
  }
}
```

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

Mídia arquivada do Amazon Kinesis Video Streams

As ações a seguir são suportadas pela mídia arquivada do Amazon Kinesis Video Streams:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

GetClip

Serviço: Amazon Kinesis Video Streams Archived Media

Faz o download de um arquivo MP4 (clipe) contendo a mídia arquivada sob demanda do stream de vídeo especificado durante o intervalo de tempo especificado.

Os parâmetros StreamArn StreamName e StreamArn são opcionais, mas você deve especificar o StreamName ou o StreamArn ao invocar essa operação de API.

Note

Você deve primeiro chamar a GetDataEndpoint API para obter um endpoint. Em seguida, envie as GetClip solicitações para esse endpoint usando o parâmetro `--endpoint-url`.

Um stream de vídeo do Amazon Kinesis tem os seguintes requisitos para fornecer dados por meio de MP4:

- [Requisitos de faixa de reprodução de vídeo](#).
- A retenção de dados deve ser maior que 0.
- A trilha de vídeo de cada fragmento deve conter dados privados de codec na Advanced Video Coding (AVC – Codificação de vídeo avançada) para H.264, e HEVC para formato H.265. Para obter mais informações, consulte [Especificação de MPEG-4 ISO/IEC 14496-15](#). Para obter informações sobre a adaptação de dados de streaming para um formato específico, consulte [Sinalizadores de adaptação do NAL](#).
- A trilha de áudio (se houver) de cada fragmento deve conter os dados privados de codec no formato AAC ([especificação de AAC ISO/IEC 13818-7](#)) ou o [formato MS Wave](#).

Você pode monitorar a quantidade de dados enviados monitorando a CloudWatch métrica da GetClip.OutgoingBytes Amazon. Para obter informações sobre como usar o Kinesis Video Streams CloudWatch para monitorar o Kinesis Video Streams, [consulte Monitoramento do Kinesis Video Streams](#). [Para obter informações sobre preços, consulte Preços e preços do Amazon Kinesis Video Streams AWS](#). Aplicam-se taxas de saída AWS de dados.

Important

Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de codificação,

que são necessárias para decodificar adequadamente o fragmento. Não há suporte para alterações de CPD entre os fragmentos de destino do clipe resultante. O CPD deve permanecer consistente na mídia consultada, caso contrário, um erro será retornado.

Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. Um erro será retornado se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.

Sintaxe da Solicitação

```
POST /getClip HTTP/1.1
Content-type: application/json

{
  "ClipFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[ClipFragmentSelector](#)

O intervalo de tempo do clipe solicitado e a origem dos carimbos de data/hora.

Tipo: objeto [ClipFragmentSelector](#)

Obrigatório: Sim

[StreamARN](#)

O Amazon Resource Name (ARN) do stream para o qual recuperar o clipe de mídia.

Você deve especificar o StreamName ou o StreamArn.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: Não

[StreamName](#)

O nome do stream para o qual recuperar o clipe de mídia.

Você deve especificar o StreamName ou o StreamArn.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200  
Content-Type: ContentType
```

```
Payload
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

A resposta retorna os cabeçalhos HTTP a seguir.

ContentType

O tipo de conteúdo da mídia no clipe solicitado.

Restrições de comprimento: tamanho mínimo de 1. O tamanho máximo é 128.

Padrão: `^[a-zA-Z0-9_\. \-]+$`

A resposta retorna as informações a seguir como corpo HTTP.

Payload

Arquivo MP4 tradicional que contém o clipe de mídia do stream de vídeo especificado. A saída conterá os primeiros 100 MB ou os primeiros 200 fragmentos do timestamp de início especificado. Para obter mais informações, consulte Limites do [Kinesis Video Streams](#).

Erros

Para obter informações sobre os erros comuns retornados pelas ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

InvalidCodecPrivateDataException

Os dados privados do codec em pelo menos uma das faixas do stream de vídeo não são válidos para essa operação.

Código de Status HTTP: 400

InvalidMediaFrameException

Um ou mais quadros no clipe solicitado não puderam ser analisados com base no codec especificado.

Código de Status HTTP: 400

MissingCodecPrivateDataException

Nenhum dado privado do codec foi encontrado em pelo menos uma das faixas do stream de vídeo.

Código de Status HTTP: 400

NoDataRetentionException

GetImagesfoi solicitado para um fluxo que não retém dados (ou seja, tem um `DataRetentionInHours` de 0).

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

GetImagesgerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

GetHLSStreamingSessionURLe GetDASHStreamingSessionURL gere esse erro se uma sessão com um `PlaybackMode` de `ON_DEMAND` ou `LIVE_REPLAY` for solicitada para um fluxo que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um `PlaybackMode` of `LIVE` for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

UnsupportedStreamMediaTypeException

O tipo da mídia (por exemplo, vídeo h.264 ou h.265 ou áudio AAC ou G.711) não pôde ser determinado a partir dos IDs de codec das faixas no primeiro fragmento de uma sessão de

reprodução. O ID do codec para a faixa 1 deve ser V_MPEG/ISO/AVC e, opcionalmente, o ID do codec para a faixa 2 deve ser. A_AAC

Código de Status HTTP: 400

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [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 PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetDASHStreamingSessionURL

Serviço: Amazon Kinesis Video Streams Archived Media

Recupera um URL MPEG Dynamic Adaptive Streaming over HTTP (DASH) para o stream. Em seguida, você pode abrir o URL em um reprodutor de mídia para visualizar o conteúdo do stream.


Tanto os parâmetros `StreamName` quanto os `StreamARN` parâmetros são opcionais, mas você deve especificar o `StreamName` ou o `StreamARN` ao invocar essa operação de API.

Um stream de vídeo do Amazon Kinesis tem os seguintes requisitos para fornecer dados por meio do MPEG-DASH:

- [Requisitos de faixa de reprodução de vídeo](#).
- A retenção de dados deve ser maior que 0.
- A trilha de vídeo de cada fragmento deve conter dados privados de codec na Advanced Video Coding (AVC – Codificação de vídeo avançada) para H.264, e HEVC para formato H.265. Para obter mais informações, consulte [Especificação de MPEG-4 ISO/IEC 14496-15](#). Para obter informações sobre a adaptação de dados de streaming para um formato específico, consulte [Sinalizadores de adaptação do NAL](#).
- A trilha de áudio (se houver) de cada fragmento deve conter os dados privados de codec no formato AAC ([especificação de AAC ISO/IEC 13818-7](#)) ou o [formato MS Wave](#).

O procedimento a seguir mostra como usar o MPEG-DASH com o Kinesis Video Streams:

1. Chame a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `GetDASHStreamingSessionURL` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.
2. Recupere a URL do MPEG-DASH usando `GetDASHStreamingSessionURL`. O Kinesis Video Streams cria uma sessão de streaming MPEG-DASH para ser usada para acessar o conteúdo em um stream usando o protocolo MPEG-DASH. `GetDASHStreamingSessionURL` retorna uma URL autenticada (que inclui um token de sessão criptografado) para o manifesto MPEG-DASH da sessão (o recurso raiz necessário para streaming com MPEG-DASH).

 Note

Não compartilhe nem armazene esse token onde uma entidade não autorizada possa acessá-lo. O token fornece acesso ao conteúdo do stream. Proteja o token com as mesmas medidas que você usa com suas AWS credenciais.

A mídia disponibilizada por meio do manifesto consiste somente no stream, no intervalo de tempo e no formato solicitados. Nenhum outro dado de mídia (como quadros fora da janela solicitada ou taxas de bits alternativas) é disponibilizado.

3. Forneça a URL (contendo o token de sessão criptografado) do manifesto MPEG-DASH para um media player que suporte o protocolo MPEG-DASH. O Kinesis Video Streams disponibiliza o fragmento de inicialização e os fragmentos de mídia por meio do URL do manifesto. O fragmento de inicialização contém os dados privados do codec para o stream e outros dados necessários para configurar o decodificador e o renderizador de vídeo ou áudio. Os fragmentos de mídia contêm quadros de vídeo codificados ou amostras de áudio codificadas.
4. O media player recebe o URL autenticado e solicita metadados de stream e dados de mídia normalmente. Quando o media player solicita dados, ele chama as seguintes ações:
 - GetDashManifest: recupera um manifesto MPEG DASH, que contém os metadados da mídia que você deseja reproduzir.
 - GetMP4InitFragment: recupera o fragmento de inicialização do MP4. O media player normalmente carrega o fragmento de inicialização antes de carregar qualquer fragmento de mídia. Esse fragmento contém os átomos fytp "" e "moov" MP4 e os átomos secundários necessários para inicializar o decodificador do media player.


O fragmento de inicialização não corresponde a um fragmento em um stream de vídeo do Kinesis. Ele contém apenas os dados privados do codec para o stream e a respectiva faixa, que o reproduzidor de mídia precisa para decodificar os quadros de mídia.

- GetMP4MediaFragment: recupera fragmentos de mídia MP4. Esses fragmentos contêm os átomos de MP4 moof "" e mdat "" seus átomos filhos, contendo os quadros de mídia do fragmento codificado e seus registros de data e hora.

 Important


Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de

codificação, que são necessárias para decodificar adequadamente o fragmento. As alterações de CPD não são suportadas durante uma sessão de streaming. O CPD deve permanecer consistente na mídia consultada.

 Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. O streaming falhará se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.


Os dados recuperados com essa ação são faturáveis. Consulte [Pricing](#) para obter detalhes.

 Note

Para restrições que se aplicam às sessões MPEG-DASH, consulte [Limites do Kinesis Video Streams](#).

Você pode monitorar a quantidade de dados que o media player consome monitorando a CloudWatch métrica da `GetMP4MediaFragment.OutgoingBytes` Amazon. Para obter informações sobre como CloudWatch monitorar o Kinesis Video Streams, [consulte Monitoramento do Kinesis Video Streams](#). [Para obter informações sobre preços, consulte Preços e preços do Amazon Kinesis Video Streams AWS](#). Aplicam-se cobranças para sessões de HLS e AWS dados de saída.

Para obter mais informações sobre o HLS, consulte [HTTP Live Streaming](#) no [site de desenvolvedores da Apple](#).

 Important

Se um erro for gerado após a invocação de uma API de mídia arquivada do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.

- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS à equipe de suporte, poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção Erros na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
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"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[DASHFragmentSelector](#)

O intervalo de tempo do fragmento solicitado e a origem dos carimbos de data/hora.

Esse parâmetro é necessário se `PlaybackMode` for `ON_DEMAND` ou `LIVE_REPLAY`. Esse parâmetro é opcional se `PlaybackMode` for `LIVE`. Se `PlaybackMode` for `LIVE`, o `FragmentSelectorType` pode ser definido, mas o `TimestampRange` deve ser definido. Se `PlaybackMode` for `ON_DEMAND` ou `LIVE_REPLAY`, ambos `FragmentSelectorType` e `TimestampRange` devem ser definidos.

Tipo: objeto [DASHFragmentSelector](#)

Obrigatório: Não

[DisplayFragmentNumber](#)

Os fragmentos são identificados no arquivo de manifesto com base em seu número de sequência na sessão. Se `DisplayFragmentNumber` estiver definido como `ALWAYS`, o número do fragmento do Kinesis Video Streams será adicionado a cada elemento S no arquivo de manifesto com o nome do atributo "kvs:fn". Esses números de fragmentos podem ser usados para registro ou para uso com outras APIs (por exemplo, `GetMedia` e `GetMediaForFragmentList`). É necessário um reprodutor de mídia MPEG-DASH personalizado para aproveitar esses atributos personalizados.

O valor padrão é `NEVER`.

Tipo: string

Valores Válidos: `ALWAYS` | `NEVER`

Obrigatório: Não

[DisplayFragmentTimestamp](#)

De acordo com a especificação MPEG-DASH, a hora do relógio de parede dos fragmentos no arquivo de manifesto pode ser derivada usando atributos no próprio manifesto. No entanto, normalmente, os reprodutores de mídia compatíveis com MPEG-DASH não lidam adequadamente com as lacunas na linha do tempo da mídia. O Kinesis Video Streams ajusta a linha do tempo da mídia no arquivo de manifesto para permitir a reprodução de mídia com descontinuidades. Portanto, a hora do relógio de parede derivada do arquivo de manifesto pode ser imprecisa. Se `DisplayFragmentTimestamp` estiver definido como `ALWAYS`, o timestamp preciso do fragmento será adicionado a cada elemento S no arquivo de manifesto com o nome

do atributo “kvs:ts”. É necessário um reprodutor de mídia MPEG-DASH personalizado para aproveitar esse atributo personalizado.

O valor padrão é NEVER. Quando [DASHFragmentSelector](#) for SERVER_TIMESTAMP, os timestamps serão os timestamps de início do servidor. Da mesma forma, quando [DASHFragmentSelector](#) for PRODUCER_TIMESTAMP, os carimbos de data/hora serão os carimbos de data e hora de início do produtor.

Tipo: sequências

Valores Válidos: ALWAYS | NEVER

Obrigatório: Não

[Expires](#)

O tempo em segundos até a sessão solicitada expirar. Esse valor pode estar entre 300 (5 minutos) e 43200 (12 horas).

Quando uma sessão expira, nenhuma nova chamada para `GetDashManifest`, `GetMP4InitFragment`, ou `GetMP4MediaFragment` pode ser feita para essa sessão.

O padrão é 300 (5 minutos).

Tipo: número inteiro

Intervalo válido: valor mínimo de 300. Valor máximo de 43200.

Obrigatório: Não

[MaxManifestFragmentResults](#)

O número máximo de fragmentos retornados no manifesto MPEG-DASH.

Quando o `PlaybackMode` é LIVE, os fragmentos mais recentes são retornados até esse valor. Quando o `PlaybackMode` é ON_DEMAND, os fragmentos mais antigos são retornados, até esse número máximo.

Quando há um número maior de fragmentos disponíveis em um manifesto MPEG-DASH ao vivo, os reprodutores de vídeo geralmente armazenam o conteúdo em buffer antes de iniciar a reprodução. Aumentar o tamanho do buffer aumenta a latência da reprodução, mas diminui a probabilidade de que o rebuffer ocorra durante a reprodução. Recomendamos que um manifesto MPEG-DASH ativo tenha no mínimo 3 fragmentos e no máximo 10 fragmentos.

O padrão é 5 fragmentos se PlaybackMode for LIVE ou LIVE_REPLAY e 1.000 se PlaybackMode for ON_DEMAND.

O valor máximo de 1.000 fragmentos corresponde a mais de 16 minutos de vídeo em streams com fragmentos de 1 segundo e mais de 2 horas e meia de vídeo em streams com fragmentos de 10 segundos.

Tipo: longo

Faixa válida: valor mínimo de 1. Valor máximo de 5000.

Obrigatório: Não

PlaybackMode

Seja para recuperar dados ao vivo, reproduzidos ao vivo ou arquivados sob demanda.

Os recursos dos três tipos de sessões incluem o seguinte:

- **LIVE**: para sessões desse tipo, o manifesto MPEG-DASH é atualizado continuamente com os fragmentos mais recentes à medida que eles se tornam disponíveis. Recomendamos que o media player recupere um novo manifesto em um intervalo de um segundo. Quando esse tipo de sessão é reproduzido em um reprodutor de mídia, a interface do usuário normalmente exibe uma notificação “ao vivo”, sem controle manual para escolher a posição na janela de reprodução a ser exibida.

Note

No LIVE modo, os fragmentos mais recentes disponíveis são incluídos em um manifesto MPEG-DASH, mesmo que haja uma lacuna entre os fragmentos (ou seja, se um fragmento estiver ausente). Uma lacuna como essa pode fazer com que um reprodutor de mídia pare ou cause um salto na reprodução. Nesse modo, os fragmentos não são adicionados ao manifesto MPEG-DASH se forem mais antigos do que o fragmento mais recente na playlist. Se o fragmento ausente ficar disponível após a adição de um fragmento subsequente ao manifesto, o fragmento mais antigo não será adicionado e a lacuna não será preenchida.

- **LIVE_REPLAY**: para sessões desse tipo, o manifesto MPEG-DASH é atualizado da mesma forma que é atualizado para o LIVE modo, exceto pelo fato de começar com a inclusão de fragmentos de um determinado horário de início. Em vez de fragmentos serem adicionados à medida que são ingeridos, os fragmentos são adicionados à medida que a duração do próximo

fragmento decorre. Por exemplo, se os fragmentos na sessão durarem dois segundos, um novo fragmento será adicionado ao manifesto a cada dois segundos. Esse modo é útil para poder iniciar a reprodução a partir da detecção de um evento e continuar transmitindo ao vivo a mídia que ainda não foi ingerida no momento da criação da sessão. Esse modo também é útil para transmitir mídia previamente arquivada sem ser limitado pelo limite de 1.000 fragmentos no ON_DEMAND modo.

- **ON_DEMAND**: para sessões desse tipo, o manifesto MPEG-DASH contém todos os fragmentos da sessão, até o número especificado em `MaxManifestFragmentResults`. O manifesto deve ser recuperado somente uma vez para cada sessão. Quando esse tipo de sessão é reproduzido em um reprodutor de mídia, a interface do usuário normalmente exibe um controle de depuração para escolher a posição na janela de reprodução a ser exibida.

Em todos os modos de reprodução, se houver `FragmentSelectorTypePRODUCER_TIMESTAMP`, e se houver vários fragmentos com o mesmo carimbo de data e hora de início, o fragmento que tiver o maior número de fragmentos (ou seja, o fragmento mais novo) será incluído no manifesto MPEG-DASH. Os outros fragmentos não estão incluídos. Fragmentos com registros de data e hora diferentes, mas com durações sobrepostas, ainda estão incluídos no manifesto MPEG-DASH. Isso pode levar a um comportamento inesperado no media player.

O padrão é LIVE.

Tipo: String

Valores Válidos: LIVE | LIVE_REPLAY | ON_DEMAND

Obrigatório: Não

StreamARN

O Amazon Resource Name (ARN) do stream para o qual recuperar a URL do manifesto MPEG-DASH.

Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: Não

StreamName

O nome do fluxo para o qual recuperar a URL do manifesto MPEG-DASH.

Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

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

{
  "DASHStreamingSessionURL": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

DASHStreamingSessionURL

A URL (contendo o token da sessão) que um media player pode usar para recuperar o manifesto MPEG-DASH.

Tipo: string

Erros

Para obter informações sobre os erros comuns retornados pelas ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

InvalidCodecPrivateDataException

Os dados privados do codec em pelo menos uma das faixas do stream de vídeo não são válidos para essa operação.

Código de Status HTTP: 400

MissingCodecPrivateDataException

Nenhum dado privado do codec foi encontrado em pelo menos uma das faixas do stream de vídeo.

Código de Status HTTP: 400

NoDataRetentionException

GetImagesfoi solicitado para um fluxo que não retém dados (ou seja, tem um `DataRetentionInHours` de 0).

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

GetImagesgerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

GetHLSStreamingSessionURL e GetDASHStreamingSessionURL gere esse erro se uma sessão com um PlaybackMode de ON_DEMAND ou LIVE_REPLAY for solicitada para um fluxo que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um PlaybackMode of LIVE for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

UnsupportedStreamMediaTypeException

O tipo da mídia (por exemplo, vídeo h.264 ou h.265 ou áudio AAC ou G.711) não pôde ser determinado a partir dos IDs de codec das faixas no primeiro fragmento de uma sessão de reprodução. O ID do codec para a faixa 1 deve ser V_MPEG/ISO/AVC e, opcionalmente, o ID do codec para a faixa 2 deve ser. A_AAC

Código de Status HTTP: 400

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [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 PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetHLSStreamingSessionURL

Serviço: Amazon Kinesis Video Streams Archived Media

Recupera um URL HTTP Live Streaming (HLS) para o stream. Em seguida, você pode abrir o URL em um navegador ou reprodutor de mídia para visualizar o conteúdo do stream.

Tanto os parâmetros `StreamName` quanto os `StreamARN` parâmetros são opcionais, mas você deve especificar o `StreamName` ou o `StreamARN` ao invocar essa operação de API.


Um stream de vídeo do Amazon Kinesis tem os seguintes requisitos para fornecer dados por meio do HLS:

- [Requisitos de faixa de reprodução de vídeo](#).
- A retenção de dados deve ser maior que 0.
- [A trilha de vídeo de cada fragmento deve conter dados privados do codec na Codificação Avançada de Vídeo \(AVC\) para o formato H.264 ou HEVC para o formato H.265 \(especificação MPEG-4 ISO/IEC 14496-15\)](#). Para obter informações sobre a adaptação de dados de streaming para um formato específico, consulte [Sinalizadores de adaptação do NAL](#).
- A faixa de áudio (se presente) de cada fragmento deve conter dados privados do codec no formato AAC ([especificação AAC ISO/IEC 13818-7](#)).

As sessões HLS do Kinesis Video Streams contêm fragmentos no formato MPEG-4 fragmentado (também chamado de fMP4 ou CMAF) ou no formato MPEG-2 (também chamado de partes TS, que a especificação HLS também suporta). Para obter mais informações sobre os tipos de fragmentos HLS, consulte a especificação [HLS](#).

O procedimento a seguir mostra como usar o HLS com o Kinesis Video Streams:

1. Chame a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `GetHLSStreamingSessionURL` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.
2. Recupere o URL do HLS usando `GetHLSStreamingSessionURL`. O Kinesis Video Streams cria uma sessão de streaming HLS para ser usada para acessar o conteúdo em um stream usando o protocolo HLS. `GetHLSStreamingSessionURL` retorna uma URL autenticada (que inclui um token de sessão criptografado) para a playlist principal do HLS da sessão (o recurso raiz necessário para streaming com HLS).

 Note

Não compartilhe nem armazene esse token onde uma entidade não autorizada possa acessá-lo. O token fornece acesso ao conteúdo do stream. Proteja o token com as mesmas medidas que você usaria com suas AWS credenciais.

A mídia disponibilizada por meio da playlist consiste somente no stream, no intervalo de tempo e no formato solicitados. Nenhum outro dado de mídia (como quadros fora da janela solicitada ou taxas de bits alternativas) é disponibilizado.

3. Forneça o URL (contendo o token de sessão criptografado) da playlist principal do HLS para um reprodutor de mídia compatível com o protocolo HLS. O Kinesis Video Streams disponibiliza a playlist de mídia HLS, o fragmento de inicialização e os fragmentos de mídia por meio da URL principal da playlist. O fragmento de inicialização contém os dados privados do codec para o stream e outros dados necessários para configurar o decodificador e o renderizador de vídeo ou áudio. Os fragmentos de mídia contêm quadros de vídeo codificados em H.264 ou amostras de áudio codificadas em AAC.
4. O reprodutor de mídia recebe o URL autenticado e solicita metadados de stream e dados de mídia normalmente. Quando o media player solicita dados, ele chama as seguintes ações:
 - `GetHLSMasterPlaylist`: recupera uma playlist principal do HLS, que contém um URL para a `GetHLSMediaPlaylist` ação de cada faixa e metadados adicionais para o reprodutor de mídia, incluindo taxa de bits e resolução estimadas.
 - `GetHLSMediaPlaylist`: recupera uma playlist de mídia HLS, que contém uma URL para acessar o fragmento de inicialização do MP4 com a `GetMP4InitFragment` ação e URLs para acessar os fragmentos de mídia MP4 com as ações. `GetMP4MediaFragment` A playlist de mídia do HLS também contém metadados sobre o stream que o player precisa para reproduzi-lo, como se `PlaybackMode` é `LIVE` ou `ON_DEMAND` A playlist de mídia HLS geralmente é estática para sessões com um `PlaybackType` de `ON_DEMAND`. A playlist de mídia do HLS é atualizada continuamente com novos fragmentos para sessões com um de. `PlaybackType` `LIVE` Há uma playlist de mídia HLS distinta para a faixa de vídeo e a faixa de áudio (se aplicável) que contém URLs de mídia MP4 para a faixa específica.
 - `GetMP4InitFragment`: recupera o fragmento de inicialização do MP4. Normalmente, o reprodutor de mídia carrega o fragmento de inicialização antes de carregar qualquer fragmento de mídia. Esse fragmento contém os átomos `fytp` "" e `moov` MP4 e os átomos secundários necessários para inicializar o decodificador do media player.

O fragmento de inicialização não corresponde a um fragmento em um stream de vídeo do Kinesis. Ele contém apenas os dados privados do codec para o stream e a respectiva faixa, que o reprodutor de mídia precisa para decodificar os quadros de mídia.

- `GetMP4MediaFragment`: recupera fragmentos de mídia MP4. Esses fragmentos contêm os átomos de MP4 `moov`""` e mdat`""` seus átomos filhos, contendo os quadros de mídia do fragmento codificado e seus registros de data e hora.`

Note

Os dados privados do codec (CPD) contidos em cada fragmento contêm informações de inicialização específicas do codec, como taxa de quadros, resolução e perfil de codificação, que são necessárias para decodificar adequadamente o fragmento. Tanto para TS quanto para MP4, as alterações de CPD são suportadas durante uma sessão de streaming. Portanto, os fragmentos em uma sessão podem ter informações diferentes no CPD sem interromper a reprodução. Para cada sessão de streaming, apenas 500 alterações de CPD são permitidas.

Important

Não há suporte para alterações na faixa. As faixas devem permanecer consistentes em toda a mídia consultada. O streaming falhará se os fragmentos no stream mudarem de ter apenas vídeo para ter áudio e vídeo, ou se uma faixa de áudio AAC for alterada para uma faixa de áudio A-Law.

Os dados recuperados com essa ação são faturáveis. Para obter mais informações, consulte [Definição de preço do](#) .

- `getTSFragment`: recupera fragmentos MPEG TS contendo dados de inicialização e de mídia para todas as faixas no fluxo.

Note

Se `ContainerFormat forMPEG_TS`, essa API será usada em vez de `GetMP4InitFragment` e `GetMP4MediaFragment` para recuperar a mídia de stream.

Os dados recuperados com essa ação são faturáveis. Para obter mais informações, consulte os preços do [Kinesis Video Streams](#).

O URL de uma sessão de streaming não deve ser compartilhado entre jogadores. O serviço pode limitar uma sessão se vários reprodutores de mídia a estiverem compartilhando. Para saber os limites de conexão, consulte Limites do [Kinesis Video Streams](#).

Você pode monitorar a quantidade de dados que o media player consome monitorando a CloudWatch métrica da `GetMP4MediaFragment.OutgoingBytes` Amazon. Para obter informações sobre como usar o Kinesis Video Streams CloudWatch para monitorar o Kinesis Video Streams, consulte [Monitoramento do Kinesis Video Streams](#). [Para obter informações sobre preços, consulte Preços e preços do Amazon Kinesis Video Streams AWS](#). Aplicam-se cobranças para sessões de HLS e AWS dados de saída.

Veja os exemplos de reprodução de vídeo no guia de documentação: [Use o AWS CLI para recuperar um URL de sessão de streaming HLS](#) e [Exemplo: Use HLS em HTML e JavaScript](#)

Para obter mais informações sobre o HLS, consulte [HTTP Live Streaming](#) no [site de desenvolvedores da Apple](#).

Important

Se um erro for gerado após a invocação de uma API de mídia arquivada do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.
- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS, a equipe de suporte poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção Erros na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
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"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

[ContainerFormat](#)

Especifica qual formato deve ser usado para empacotar a mídia. A especificação do formato do FRAGMENTED_MP4 contêiner empacota a mídia em fragmentos MP4 (fMP4 ou CMAF). Essa é a embalagem recomendada porque a sobrecarga de embalagem é mínima. A outra opção de formato de contêiner é MPEG_TS. O HLS oferece suporte a fragmentos MPEG TS desde que foi lançado e, às vezes, é o único pacote suportado em reprodutores HLS mais antigos. O MPEG

TS normalmente tem uma sobrecarga de embalagem de 5 a 25%. Isso significa que o MPEG TS normalmente requer de 5 a 25% mais largura de banda e custo do que o FMP4.

O padrão é FRAGMENTED_MP4.

Tipo: String

Valores Válidos: FRAGMENTED_MP4 | MPEG_TS

Obrigatório: Não

DiscontinuityMode

Especifica quando os sinalizadores que marcam descontinuidades entre fragmentos são adicionados às playlists de mídia.

Os reprodutores de mídia geralmente criam uma linha do tempo do conteúdo de mídia a ser reproduzido, com base nos registros de data e hora de cada fragmento. Isso significa que, se houver alguma sobreposição ou lacuna entre os fragmentos (como é normal se estiver [HLSFragmentSelector](#) definido como SERVER_TIMESTAMP), a linha do tempo do media player também terá pequenas lacunas entre os fragmentos em alguns lugares e substituirá os quadros em outros lugares. Lacunas na linha do tempo do reprodutor de mídia podem fazer com que a reprodução pare e as sobreposições podem fazer com que a reprodução fique instável. Quando há sinalizadores de descontinuidade entre os fragmentos, espera-se que o reprodutor de mídia redefina a linha do tempo, resultando na reprodução do próximo fragmento imediatamente após o fragmento anterior.

Os seguintes modos são suportados:

- ALWAYS: um marcador de descontinuidade é colocado entre cada fragmento na playlist de mídia do HLS. É recomendável usar um valor de ALWAYS se os carimbos de data/hora do fragmento não forem precisos.
- NEVER: nenhum marcador de descontinuidade é colocado em nenhum lugar. É recomendável usar um valor de NEVER para garantir que NEVER a linha do tempo do reprodutor de mídia seja mapeada com mais precisão para os carimbos de data/hora do produtor.
- ON_DISCONTINUITY: um marcador de descontinuidade é colocado entre fragmentos que têm uma lacuna ou sobreposição de mais de 50 milissegundos. Para a maioria dos cenários de reprodução, é recomendável usar um valor de ON_DISCONTINUITY que a linha do tempo do reprodutor de mídia seja redefinida somente quando houver um problema significativo com a linha do tempo da mídia (por exemplo, um fragmento ausente).

O padrão é ALWAYS quando [HLSFragmentSelector](#) está definido como e NEVER quando está definido como PRODUCER_TIMESTAMP. SERVER_TIMESTAMP

Tipo: sequências

Valores Válidos: ALWAYS | NEVER | ON_DISCONTINUITY

Obrigatório: Não

[DisplayFragmentTimestamp](#)

Especifica quando os carimbos de data e hora de início do fragmento devem ser incluídos na playlist de mídia do HLS. Normalmente, os reprodutores de mídia relatam a posição do indicador de reprodução como um tempo em relação ao início do primeiro fragmento na sessão de reprodução. No entanto, quando os carimbos de data e hora de início são incluídos na lista de reprodução de mídia do HLS, alguns reprodutores de mídia podem relatar o indicador de reprodução atual como um horário absoluto com base nos carimbos de data/hora do fragmento. Isso pode ser útil para criar uma experiência de reprodução que mostre aos espectadores a hora do relógio de parede da mídia.

O padrão é NEVER. Quando [HLSFragmentSelector](#) for SERVER_TIMESTAMP, os carimbos de data/hora serão os carimbos de data e hora de início do servidor. Da mesma forma, quando [HLSFragmentSelector](#) for PRODUCER_TIMESTAMP, os carimbos de data/hora serão os carimbos de data e hora de início do produtor.

Tipo: sequências

Valores Válidos: ALWAYS | NEVER

Obrigatório: Não

[Expires](#)

O tempo em segundos até a sessão solicitada expirar. Esse valor pode estar entre 300 (5 minutos) e 43200 (12 horas).

Quando uma sessão expira, nenhuma nova chamada para `GetHLSMasterPlaylist`, `GetHLSMediaPlaylist`, `GetMP4InitFragment`, `GetMP4MediaFragment`, ou `GetTSFragment` pode ser feita para essa sessão.

O padrão é 300 (5 minutos).

Tipo: número inteiro

Intervalo válido: valor mínimo de 300. Valor máximo de 43200.

Obrigatório: Não

[HLSFragmentSelector](#)

O intervalo de tempo do fragmento solicitado e a origem dos carimbos de data/hora.

Esse parâmetro é necessário se `PlaybackMode` for `ON_DEMAND` ou `LIVE_REPLAY`.

Esse parâmetro é opcional se `PlaybackMode` for `LIVE`. Se `PlaybackMode` for `LIVE`, o

`FragmentSelectorType` pode ser definido, mas o não `TimestampRange` deve ser definido.

Se `PlaybackMode` for `ON_DEMAND` ou `LIVE_REPLAY`, ambos `FragmentSelectorType` e

`TimestampRange` devem ser definidos.

Tipo: objeto [HLSFragmentSelector](#)

Obrigatório: Não

[MaxMediaPlaylistFragmentResults](#)

O número máximo de fragmentos que são retornados nas playlists de mídia do HLS.

Quando o `PlaybackMode` é `LIVE`, os fragmentos mais recentes são retornados até esse valor.

Quando o `PlaybackMode` é `ON_DEMAND`, os fragmentos mais antigos são retornados, até esse número máximo.

Quando há um número maior de fragmentos disponíveis em uma playlist de mídia HLS ao vivo, os reprodutores de vídeo geralmente armazenam o conteúdo em buffer antes de iniciar a reprodução. Aumentar o tamanho do buffer aumenta a latência da reprodução, mas diminui a probabilidade de que o rebuffer ocorra durante a reprodução. Recomendamos que uma playlist de mídia HLS ao vivo tenha no mínimo 3 fragmentos e no máximo 10 fragmentos.

O padrão é 5 fragmentos se `PlaybackMode` for `LIVE` ou `LIVE_REPLAY` e 1.000 se `PlaybackMode` for `ON_DEMAND`.

O valor máximo de 5.000 fragmentos corresponde a mais de 80 minutos de vídeo em streams com fragmentos de 1 segundo e mais de 13 horas de vídeo em streams com fragmentos de 10 segundos.

Tipo: longo

Faixa válida: valor mínimo de 1. Valor máximo de 5000.

Obrigatório: Não

PlaybackMode

Seja para recuperar dados ao vivo, reproduzidos ao vivo ou arquivados sob demanda.

Os recursos dos três tipos de sessões incluem o seguinte:

- **LIVE**: para sessões desse tipo, a playlist de mídia do HLS é atualizada continuamente com os fragmentos mais recentes à medida que são disponibilizados. Recomendamos que o reprodutor de mídia recupere uma nova playlist em um intervalo de um segundo. Quando esse tipo de sessão é reproduzido em um reprodutor de mídia, a interface do usuário normalmente exibe uma notificação “ao vivo”, sem controle manual para escolher a posição na janela de reprodução a ser exibida.

Note

No LIVE modo, os fragmentos mais recentes disponíveis são incluídos em uma playlist de mídia HLS, mesmo que haja uma lacuna entre os fragmentos (ou seja, se um fragmento estiver ausente). Uma lacuna como essa pode fazer com que um reprodutor de mídia pare ou cause um salto na reprodução. Nesse modo, os fragmentos não são adicionados à lista de reprodução de mídia do HLS se forem mais antigos que o fragmento mais recente na lista de reprodução. Se o fragmento ausente ficar disponível após a adição de um fragmento subsequente à playlist, o fragmento mais antigo não será adicionado e a lacuna não será preenchida.

- **LIVE_REPLAY**: para sessões desse tipo, a playlist de mídia do HLS é atualizada da mesma forma que é atualizada para o LIVE modo, exceto pelo fato de começar com a inclusão de fragmentos de um determinado horário de início. Em vez de fragmentos serem adicionados à medida que são ingeridos, os fragmentos são adicionados à medida que a duração do próximo fragmento decorre. Por exemplo, se os fragmentos da sessão durarem dois segundos, um novo fragmento será adicionado à playlist de mídia a cada dois segundos. Esse modo é útil para poder iniciar a reprodução a partir da detecção de um evento e continuar transmitindo ao vivo a mídia que ainda não foi ingerida no momento da criação da sessão. Esse modo também é útil para transmitir mídia previamente arquivada sem ser limitado pelo limite de 1.000 fragmentos no ON_DEMAND modo.
- **ON_DEMAND**: para sessões desse tipo, a playlist de mídia HLS contém todos os fragmentos da sessão, até o número especificado em `MaxMediaPlaylistFragmentResults`. A playlist deve ser recuperada somente uma vez para cada sessão. Quando esse tipo de sessão é

reproduzido em um reprodutor de mídia, a interface do usuário normalmente exibe um controle de depuração para escolher a posição na janela de reprodução a ser exibida.

Em todos os modos de reprodução, se `FragmentSelectorType` houver `PRODUCER_TIMESTAMP`, e se houver vários fragmentos com o mesmo carimbo de data e hora de início, o fragmento que tiver o maior número de fragmentos (ou seja, o fragmento mais novo) será incluído na playlist de mídia do HLS. Os outros fragmentos não estão incluídos. Fragmentos com registros de data e hora diferentes, mas com durações sobrepostas, ainda estão incluídos na lista de reprodução de mídia do HLS. Isso pode levar a um comportamento inesperado no media player.

O padrão é `LIVE`.

Tipo: `String`

Valores Válidos: `LIVE` | `LIVE_REPLAY` | `ON_DEMAND`

Obrigatório: Não

[StreamARN](#)

O Amazon Resource Name (ARN) do stream para o qual recuperar a URL da playlist principal do HLS.

Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: `string`

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: Não

[StreamName](#)

O nome do stream para o qual recuperar o URL da playlist principal do HLS.

Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: `string`

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

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

{
  "HLSStreamingSessionURL": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[HLSStreamingSessionURL](#)

O URL (contendo o token da sessão) que um reprodutor de mídia pode usar para recuperar a playlist principal do HLS.

Tipo: string

Erros

Para obter informações sobre os erros comuns retornados pelas ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

InvalidCodecPrivateDataException

Os dados privados do codec em pelo menos uma das faixas do stream de vídeo não são válidos para essa operação.

Código de Status HTTP: 400

MissingCodecPrivateDataException

Nenhum dado privado do codec foi encontrado em pelo menos uma das faixas do stream de vídeo.

Código de Status HTTP: 400

NoDataRetentionException

GetImagesfoi solicitado para um fluxo que não retém dados (ou seja, tem um `DataRetentionInHours` de 0).

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

GetImagesgerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

GetHLSStreamingSessionURLe GetDASHStreamingSessionURL gere esse erro se uma sessão com um `PlaybackMode` de `ON_DEMAND` ou `LIVE_REPLAY` for solicitada para um fluxo que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um `PlaybackMode` de `LIVE` for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

UnsupportedStreamMediaTypeException

O tipo da mídia (por exemplo, vídeo h.264 ou h.265 ou áudio AAC ou G.711) não pôde ser determinado a partir dos IDs de codec das faixas no primeiro fragmento de uma sessão de

reprodução. O ID do codec para a faixa 1 deve ser V_MPEG/ISO/AVC e, opcionalmente, o ID do codec para a faixa 2 deve ser. A_AAC

Código de Status HTTP: 400

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [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 PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetImages

Serviço: Amazon Kinesis Video Streams Archived Media

Recupera uma lista de imagens correspondentes a cada registro de data e hora para um determinado intervalo de tempo, intervalo de amostragem e configuração de formato de imagem.

Note

Você deve primeiro chamar a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `GetImages` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.

[Requisitos de faixa de reprodução de vídeo.](#)

Sintaxe da Solicitação

```
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
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

EndTimeStamp

A data e hora de término da variedade de imagens a serem geradas. Se o intervalo de tempo entre `StartTimeStamp` e `EndTimeStamp` for superior a 300 segundos `StartTimeStamp`, você receberá um `IllegalArgumentException`.

Tipo: Timestamp

Obrigatório: Sim

Format

O formato que será usado para codificar a imagem.

Tipo: strings

Valores Válidos: JPEG | PNG

Obrigatório: Sim

FormatConfig

A lista de uma estrutura de pares de valores-chave que contém parâmetros extras que podem ser aplicados quando a imagem é gerada. A `FormatConfig` chave é `JPEGQuality`, que indica a chave de qualidade JPEG a ser usada para gerar a imagem. O `FormatConfig` valor aceita ints de 1 a 100. Se o valor for 1, a imagem será gerada com menos qualidade e a melhor compressão. Se o valor for 100, a imagem será gerada com a melhor qualidade e menos compressão. Se nenhum valor for fornecido, o valor padrão da `JPEGQuality` chave será definido como 80.

Tipo: mapa de string para string

Entradas do mapa: número máximo de um item.

Chaves válidas: `JPEGQuality`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `^[a-zA-Z_0-9]+`

Obrigatório: não

HeightPixels

A altura da imagem de saída usada em conjunto com o `WidthPixels` parâmetro. Quando os `WidthPixels` parâmetros `HeightPixels` e parâmetros forem fornecidos, a imagem será esticada para se ajustar à proporção especificada. Se somente o `HeightPixels` parâmetro for fornecido, sua proporção original será usada para calcular a `WidthPixels` proporção. Se nenhum parâmetro for fornecido, o tamanho original da imagem será retornado.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 2160.

Obrigatório: não

ImageSelectorType

A origem dos registros de data e hora do servidor ou do produtor a serem usados para gerar as imagens.

Tipo: strings

Valores Válidos: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Obrigatório: Sim

MaxResults

O número máximo de imagens a serem retornadas pela API.

Note

O limite padrão é de 25 imagens por resposta da API. Fornecer um valor `MaxResults` maior que esse resultará em um tamanho de página de 25. Quaisquer resultados adicionais serão paginados.

Tipo: longo

Faixa válida: valor mínimo de 1. Valor máximo de 100.

Obrigatório: não

NextToken

Um token que especifica por onde começar a pagnar o próximo conjunto de imagens. Isso é `GetImages:NextToken` de uma resposta previamente truncada.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 4.096.

Padrão: `[a-zA-Z0-9+/\]{0,2}`

Obrigatório: não

SamplingInterval

O intervalo de tempo em milissegundos (ms) no qual as imagens precisam ser geradas a partir do fluxo. O valor mínimo que pode ser fornecido é 200 ms (5 imagens por segundo). Se o intervalo do timestamp for menor que o intervalo de amostragem, a imagem do `startTimestamp` será retornada, se disponível.

Tipo: inteiro

Obrigatório: Sim

StartTimestamp

O ponto de partida a partir do qual as imagens devem ser geradas. Isso `StartTimestamp` deve estar dentro de um intervalo inclusivo de registros de data e hora para que uma imagem seja retornada.

Tipo: Timestamp

Obrigatório: Sim

StreamARN

O Amazon Resource Name (ARN) do stream do qual recuperar as imagens. Você deve especificar o `StreamName` ou `StreamARN` o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual recuperar as imagens. Você deve especificar o StreamName ou StreamARN o.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

WidthPixels

A largura da imagem de saída usada em conjunto com o HeightPixels parâmetro. Quando os HeightPixels parâmetros WidthPixels e parâmetros forem fornecidos, a imagem será esticada para se ajustar à proporção especificada. Se somente o WidthPixels parâmetro for fornecido ou se somente o HeightPixels for fornecido, um ValidationException será lançado. Se nenhum parâmetro for fornecido, o tamanho original da imagem do stream será retornado.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 3840.

Obrigatório: Não

Sintaxe da Resposta

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

{
  "Images": [
    {
      "Error": "string",
      "ImageContent": "string",
      "TimeStamp": number
    }
  ],
}
```

```
"NextToken": "string"  
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

Images

A lista de imagens geradas a partir do stream de vídeo. Se não houver mídia disponível para o carimbo de data/hora especificado, o NO_MEDIA erro será listado na saída. Se ocorrer um erro enquanto a imagem estiver sendo gerada, MEDIA_ERROR ele será listado na saída como a causa da imagem ausente.

Tipo: matriz de objetos [Image](#)

NextToken

O token criptografado que foi usado na solicitação para obter mais imagens.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 4.096.

Padrão: [a-zA-Z0-9+/\]= {0, 2}

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

GetImagesgerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

GetHLSStreamingSessionURL e GetDASHStreamingSessionURL gere esse erro se uma sessão com um PlaybackMode de ON_DEMAND ou LIVE_REPLAY for solicitada para um fluxo que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um PlaybackMode of LIVE for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

GetMediaForFragmentList

Serviço: Amazon Kinesis Video Streams Archived Media

Obtém mídia para uma lista de fragmentos (especificada pelo número do fragmento) dos dados arquivados em um stream de vídeo do Amazon Kinesis.

Note

Você deve primeiro chamar a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `GetMediaForFragmentList` solicitações para esse endpoint usando o parâmetro [--endpoint-url](#).

Para ver os limites, consulte Limites do [Kinesis Video Streams](#).

Important

Se um erro for gerado após a invocação de uma API de mídia arquivada do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.
- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS, a equipe de suporte poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção Erros na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
POST /getMediaForFragmentList HTTP/1.1
Content-type: application/json
```

```
{
  "Fragments": [ "string" ],
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

Fragments

Uma lista dos números de fragmentos para os quais recuperar a mídia. Você recupera esses valores com [ListFragments](#).

Tipo: Matriz de strings

Membros da Matriz: Número mínimo de 1 item. Número máximo de 1.000 itens.

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[0-9]+$`

Exigido: Sim

StreamARN

O Amazon Resource Name (ARN) do stream do qual recuperar a mídia fragmentada. Especifique esse parâmetro ou o `StreamName` parâmetro.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual recuperar a mídia fragmentada. Especifique esse parâmetro ou o `StreamARN` parâmetro.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

A resposta retorna os cabeçalhos HTTP a seguir.

ContentType

O tipo de conteúdo da mídia solicitada.

Restrições de Tamanho: Tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[a-zA-Z0-9_\.\-]+$`

A resposta retorna as informações a seguir como corpo HTTP.

Payload

A carga que o Kinesis Video Streams retorna é uma sequência de partes do stream especificado. Para obter informações sobre os fragmentos, consulte [PutMedia](#). Os trechos que o Kinesis Video Streams retorna na chamada também incluem as seguintes tags adicionais de `GetMediaForFragmentList` Matroska (MKV):

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - Número do fragmento retornado no bloco.
- `AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP` - Carimbo de data/hora do fragmento no lado do servidor.
- `AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP` - Carimbo de data/hora do fragmento no lado do produtor.

As seguintes tags serão incluídas se ocorrer uma exceção:

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - O número do fragmento que gerou a exceção.
- `AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE` - O código inteiro do erro.
- `AWS_KINESISVIDEO_EXCEPTION_MESSAGE` - Uma descrição em texto da exceção.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

`GetImages` gerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

GetHLSStreamingSessionURL e GetDASHStreamingSessionURL gere esse erro se uma sessão com um PlaybackMode de ON_DEMAND ou LIVE_REPLAY for solicitada para um fluxo que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um PlaybackMode of LIVE for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

ListFragments

Serviço: Amazon Kinesis Video Streams Archived Media

Retorna uma lista de [Fragment](#) objetos do intervalo especificado de fluxo e timestamp nos dados arquivados.

A listagem de fragmentos acaba sendo consistente. Isso significa que, mesmo que o produtor receba uma confirmação de que um fragmento persiste, o resultado pode não ser retornado imediatamente de uma solicitação para `ListFragments`. No entanto, os resultados geralmente estão disponíveis em menos de um segundo.

Note

Você deve primeiro chamar a `GetDataEndpoint` API para obter um endpoint. Em seguida, envie as `ListFragments` solicitações para esse endpoint usando o parâmetro `--endpoint-url`.

Important

Se um erro for gerado após a invocação de uma API de mídia arquivada do Kinesis Video Streams, além do código de status HTTP e do corpo da resposta, ele incluirá as seguintes informações:

- `x-amz-ErrorType` Cabeçalho HTTP — contém um tipo de erro mais específico, além do que o código de status HTTP fornece.
- `x-amz-RequestId` Cabeçalho HTTP — se você quiser relatar um problema AWS, a equipe de suporte poderá diagnosticar melhor o problema se receber o ID da solicitação.

Tanto o código de status HTTP quanto o `ErrorType` cabeçalho podem ser utilizados para tomar decisões programáticas sobre se os erros podem ser repetidos e sob quais condições, além de fornecer informações sobre quais ações o programador cliente pode precisar realizar para tentar novamente com sucesso.

Para obter mais informações, consulte a seção [Erros](#) na parte inferior deste tópico, bem como [Erros comuns](#).

Sintaxe da Solicitação

```
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"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

FragmentSelector

Descreve o intervalo do timestamp e a origem do timestamp para o intervalo de fragmentos a serem retornados.

Note

Isso só é necessário quando o NextToken não é passado na API.

Tipo: objeto FragmentSelector

Obrigatório: Não

MaxResults

O número total de fragmentos a serem devolvidos. Se o número total de fragmentos disponíveis for maior que o valor especificado em `max-results`, um [ListFragments: NextToken](#) será fornecido na saída que você pode usar para retomar a paginação.

O valor padrão é 100.

Tipo: longo

Faixa válida: valor mínimo de 1. Valor máximo de 1.000.

Obrigatório: não

NextToken

Um token para especificar onde iniciar a paginação. Este é o [ListFragments: NextToken](#) de uma resposta previamente truncada.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 4.096.

Padrão: `[a-zA-Z0-9+/\]=\{0,2\}`

Obrigatório: não

StreamARN

O Amazon Resource Name (ARN) do stream do qual recuperar uma lista de fragmentos. Especifique esse parâmetro ou o `StreamName` parâmetro.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do fluxo do qual recuperar uma lista de fragmentos. Especifique esse parâmetro ou o `StreamARN` parâmetro.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Sintaxe da Resposta

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

{
  "Fragments": [
    {
      "FragmentLengthInMilliseconds": number,
      "FragmentNumber": "string",
      "FragmentSizeInBytes": number,
      "ProducerTimestamp": number,
      "ServerTimestamp": number
    }
  ],
  "NextToken": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

Fragments

Uma lista de [Fragment](#) objetos arquivados do stream que atendem aos critérios do seletor. Os resultados não estão em uma ordem específica, mesmo entre páginas.

Se não houver fragmentos no fluxo que atendam aos critérios do seletor, uma lista vazia será retornada.

Tipo: matriz de objetos [Fragment](#)

NextToken

Se a lista retornada for truncada, a operação retornará esse token para ser usado na recuperação da próxima página de resultados. Esse valor é `null` quando não há mais resultados a serem retornados.

Tipo: `string`

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 4.096.

Padrão: `[a-zA-Z0-9+/\]+= {0, 2}`

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu um limite. Tente fazer a ligação mais tarde. Para obter informações sobre limites, consulte Limites do [Kinesis Video Streams](#).

Código de Status HTTP: 400

InvalidArgumentException

Um parâmetro especificado excede suas restrições, não é suportado ou não pode ser usado.

Código de Status HTTP: 400

NotAuthorizedException

Código de status: 403, O chamador não está autorizado a realizar uma operação no stream fornecido ou o token expirou.

Código de status HTTP: 401

ResourceNotFoundException

`GetImages` gerará esse erro quando o Kinesis Video Streams não conseguir encontrar o stream que você especificou.

`GetHLSStreamingSessionURL` e `GetDASHStreamingSessionURL` gere esse erro se uma sessão com um `PlaybackMode` de `ON_DEMAND` ou `LIVE_REPLAY` for solicitada para um fluxo

que não tem fragmentos dentro do intervalo de tempo solicitado, ou se uma sessão com um PlaybackMode of LIVE for solicitada para um fluxo que não tem fragmentos nos últimos 30 segundos.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

Amazon Kinesis Video Streams

As seguintes ações são suportadas pelos canais de sinalização de vídeo do Amazon Kinesis:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

GetIceServerConfig

Serviço: Amazon Kinesis Video Signaling Channels

Observação: antes de usar essa API, você deve chamar a `GetSignalingChannelEndpoint` API para solicitar o endpoint HTTPS. Em seguida, você especifica o endpoint e a região na sua solicitação de `GetIceServerConfig` API.

Obtém as informações de configuração do servidor Interactive Connectivity Establishment (ICE), incluindo URIs, nome de usuário e senha, que podem ser usados para configurar a conexão WebRTC. O componente ICE usa essas informações de configuração para configurar a conexão WebRTC, incluindo a autenticação com o servidor de retransmissão Traversal Using Relays around NAT (TURN).

TURN é um protocolo usado para melhorar a conectividade dos peer-to-peer aplicativos. Ao fornecer um serviço de retransmissão baseado em nuvem, o TURN garante que uma conexão possa ser estabelecida mesmo quando um ou mais pares não conseguem uma conexão direta. peer-to-peer Para obter mais informações, consulte [Uma API REST para acesso aos serviços TURN](#).

Você pode invocar essa API para estabelecer um mecanismo de fallback caso algum dos pares não consiga estabelecer uma peer-to-peer conexão direta por meio de um canal de sinalização. Você deve especificar o Amazon Resource Name (ARN) do seu canal de sinalização para invocar essa API.

Sintaxe da Solicitação

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ClientId": "string",
  "Service": "string",
  "Username": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O ARN do canal de sinalização a ser usado para a peer-to-peer conexão entre os pares configurados.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Exigido: Sim

ClientId

Identificador exclusivo para o espectador. Deve ser exclusivo dentro do canal de sinalização.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: não

Service

Especifica o serviço desejado. Atualmente, TURN é o único valor válido.

Tipo: strings

Valores Válidos: TURN

Obrigatório: não

Username

Uma ID de usuário opcional a ser associada às credenciais.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Sintaxe da Resposta

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

{
  "IceServerList": [
    {
      "Password": "string",
      "Ttl": number,
      "Uris": [ "string" ],
      "Username": "string"
    }
  ]
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

[IceServerList](#)

A lista de objetos de informações do servidor ICE.

Tipo: matriz de objetos [IceServer](#)

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

Sua solicitação foi restringida porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

`InvalidClientException`

O cliente especificado é inválido.

Código de Status HTTP: 400

`NotAuthorizedException`

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

`ResourceNotFoundException`

O recurso especificado não foi encontrado.

Código de Status HTTP: 404

`SessionExpiredException`

Se a sessão do cliente tiver expirado. Depois que o cliente estiver conectado, a sessão será válida por 45 minutos. O cliente deve se reconectar ao canal para continuar enviando/recebendo mensagens.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)

- [AWS SDK para Ruby V3](#)

SendAlexaOfferToMaster

Serviço: Amazon Kinesis Video Signaling Channels

Note

Antes de usar essa API, você deve chamar a `GetSignalingChannelEndpoint` API para obter um endpoint. Em seguida, você especifica o endpoint e a região na sua solicitação de `SendAlexaOfferToMaster` API.

Essa API permite que você conecte dispositivos habilitados para WebRTC com dispositivos de exibição Alexa. Quando invocado, ele envia a oferta do Alexa Session Description Protocol (SDP) para o peer principal. A oferta é entregue assim que o mestre é conectado ao canal de sinalização especificado. Essa API retorna a resposta SDP do mestre conectado. Se o mestre não estiver conectado ao canal de sinalização, as solicitações de reentrega serão feitas até que a mensagem expire.

Sintaxe da Solicitação

```
POST /v1/send-alex-a-offer-to-master HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "MessagePayload": "string",
  "SenderClientId": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

ChannelARN

O Amazon Resource Name (ARN) do canal de sinalização pelo qual a Alexa e o peer principal se comunicam.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Exigido: Sim

MessagePayload

O SDP codificado em base64 oferece conteúdo.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 10000.

Padrão: [a-zA-Z0-9+/=]+

Exigido: Sim

SenderClientId

O identificador exclusivo do cliente remetente.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Exigido: Sim

Sintaxe da Resposta

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

{
  "Answer": "string"
}
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

Answer

O conteúdo da resposta SDP codificada em base64.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 10000.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

ClientLimitExceededException

Sua solicitação foi restringida porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

NotAuthorizedException

O chamador não está autorizado a realizar essa operação.

Código de status HTTP: 401

ResourceNotFoundException

O recurso especificado não foi encontrado.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

Amazon Kinesis WebRTC Streams

As seguintes ações são suportadas pelo Amazon Kinesis Video WebRTC Storage:

- [JoinStorageSession](#)

JoinStorageSession

Serviço: Amazon Kinesis Video WebRTC Storage

Note

Antes de usar essa API, você deve chamar a `GetSignalingChannelEndpoint` API para solicitar o endpoint WebRTC. Em seguida, você especifica o endpoint e a região na sua solicitação de `JoinStorageSession` API.

Participe da sessão WebRTC contínua de vídeo unidirecional e/ou áudio multidirecional como um dispositivo de produção de vídeo para um canal de entrada. Se não houver uma sessão existente para o canal, uma nova sessão de streaming precisa ser criada e o Amazon Resource Name (ARN) do canal de sinalização deve ser fornecido.

Atualmente, para o `SINGLE_MASTER` tipo, um dispositivo produtor de vídeo é capaz de ingerir mídia de áudio e vídeo em um stream. Somente dispositivos produtores de vídeo podem participar da sessão e gravar mídia.

Important

Atualmente, as faixas de áudio e vídeo são necessárias para a ingestão do WebRTC.

Requisitos atuais:

- Faixa de vídeo: H.264
- Faixa de áudio: Opus

O vídeo ingerido resultante no stream de vídeo do Kinesis terá os seguintes parâmetros: vídeo H.264 e áudio AAC.

Depois que um participante principal negociar uma conexão por meio do WebRTC, a sessão de mídia ingerida será armazenada no stream de vídeo do Kinesis. Vários espectadores podem então reproduzir mídia em tempo real por meio de nossas APIs de reprodução.

Você também pode usar os HLS recursos existentes do Kinesis Video Streams, DASH como reprodução, [GetImages](#) geração de imagens e muito mais, com mídia WebRTC ingerida.

Note

Atualmente, não há suporte para entrega de imagens e notificações do S3.

Note

Suponha que somente um cliente de dispositivo produtor de vídeo possa ser associado a uma sessão do canal. Se mais de um cliente ingressar na sessão de um canal específico como dispositivo de produção de vídeo, a solicitação mais recente do cliente terá precedência.

Informações adicionais

- Idempotente - Essa API não é idempotente.
- Comportamento de nova tentativa: isso é contabilizado como uma nova chamada de API.
- Chamadas simultâneas - Chamadas simultâneas são permitidas. Uma oferta é enviada uma vez por cada chamada.

Sintaxe da Solicitação

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json
```

```
{
  "channelArn": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

channelArn

O nome do recurso da Amazon (ARN) do canal de sinalização.

Tipo: string

Padrão: `^arn:(aws[a-zA-Z-]*) :kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+$`

Exigido: Sim

Sintaxe da Resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação tiver êxito, o serviço enviará de volta uma resposta HTTP 200 com um corpo HTTP vazio.

Erros

Para obter informações sobre os erros comuns a todas as ações, consulte [Erros comuns](#).

AccessDeniedException

Você não tem as permissões necessárias para realizar essa operação.

Código de Status HTTP: 403

ClientLimitExceededException

O Kinesis Video Streams limitou a solicitação porque você excedeu o limite permitido de chamadas de clientes. Tente fazer a ligação mais tarde.

Código de Status HTTP: 400

InvalidArgumentException

O valor desse parâmetro de entrada é inválido.

Código de Status HTTP: 400

ResourceNotFoundException

O recurso especificado não foi encontrado.

Código de Status HTTP: 404

Consulte Também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK para Go v2](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para JavaScript V3](#)
- [AWS SDK para PHP V3](#)
- [AWS SDK para Python](#)
- [AWS SDK para Ruby V3](#)

Tipos de dados

Os seguintes tipos de dados são compatíveis com o Amazon Kinesis Video Streams:

- [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](#)

Os seguintes tipos de dados são compatíveis com o Amazon Kinesis Video Streams Media:

- [StartSelector](#)

Os seguintes tipos de dados são compatíveis com a mídia arquivada do Amazon Kinesis Video Streams:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Os seguintes tipos de dados são compatíveis com os canais de sinalização de vídeo do Amazon Kinesis:

- [IceServer](#)

Os seguintes tipos de dados são compatíveis com o Amazon Kinesis Video WebRTC Storage:

Amazon Kinesis Video Streams

Os seguintes tipos de dados são compatíveis com o Amazon Kinesis Video Streams:

- [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

Serviço: Amazon Kinesis Video Streams

Uma estrutura que encapsula os metadados e as propriedades de um canal de sinalização.

Conteúdo

ChannelARN

O nome do recurso da Amazon (ARN) do canal de sinalização.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

ChannelName

O nome do canal de sinalização.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: não

ChannelStatus

Status atual do canal de sinalização.

Tipo: strings

Valores Válidos: `CREATING | ACTIVE | UPDATING | DELETING`

Obrigatório: não

ChannelType

O tipo do canal de sinalização.

Tipo: strings

Valores Válidos: SINGLE_MASTER | FULL_MESH

Obrigatório: não

CreationTime

A hora em que o canal de sinalização foi criado.

Tipo: Timestamp

Obrigatório: não

SingleMasterConfiguration

Uma estrutura que contém a configuração para o tipo de SINGLE_MASTER canal.

Tipo: objeto [SingleMasterConfiguration](#)

Obrigatório: Não

Version

A versão atual do canal de sinalização.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ChannelNameCondition

Serviço: Amazon Kinesis Video Streams

Um parâmetro de entrada opcional para a `ListSignalingChannels` API. Quando esse parâmetro é especificado durante a invocação `ListSignalingChannels`, a API retorna somente os canais que atendem a uma condição especificada em `ChannelNameCondition`.

Conteúdo

ComparisonOperator

Um operador de comparação. Atualmente, você só pode especificar o `BEGINS_WITH` operador, que encontra canais de sinalização cujos nomes começam com um determinado prefixo.

Tipo: strings

Valores Válidos: `BEGINS_WITH`

Obrigatório: não

ComparisonValue

Um valor a ser comparado.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

DeletionConfig

Serviço: Amazon Kinesis Video Streams

Os detalhes de configuração necessários para excluir a conexão do stream do Edge Agent.

Conteúdo

DeleteAfterUpload

O booleano valor usado para indicar se você deseja ou não marcar a mídia para exclusão depois de carregada na nuvem do Kinesis Video Stream. Os arquivos de mídia podem ser excluídos se algum dos valores da configuração de exclusão estiver definido como `true`, por exemplo `EdgeRetentionInHours`, quando o limite para o ou o `MaxLocalMediaSizeInMB` for atingido.

Como o valor padrão está definido como `true`, configure a programação do carregador de forma que os arquivos de mídia não sejam excluídos antes de serem inicialmente enviados para a AWS nuvem.

Tipo: booleano

Obrigatório: não

EdgeRetentionInHours

O número de horas em que você deseja reter os dados no fluxo no Edge Agent. O valor padrão do tempo de retenção é de 720 horas, o que significa 30 dias.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 720.

Obrigatório: não

LocalSizeConfig

O valor do tamanho local necessário para excluir a configuração de borda.

Tipo: objeto [LocalSizeConfig](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

EdgeAgentStatus

Serviço: Amazon Kinesis Video Streams

Um objeto que contém os detalhes de status mais recentes das tarefas de gravador e carregador de um agente de ponta. Use essas informações para determinar a integridade atual de um agente de ponta.

Conteúdo

LastRecorderStatus

O status mais recente de um trabalho de gravação do Stream's Edge.

Tipo: objeto [LastRecorderStatus](#)

Obrigatório: Não

LastUploaderStatus

O status mais recente de um trabalho do stream's edge to cloud uploader.

Tipo: objeto [LastUploaderStatus](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

EdgeConfig

Serviço: Amazon Kinesis Video Streams

Uma descrição da configuração de borda do stream que será usada para sincronizar com o componente Edge Agent IoT Greengrass. O componente Edge Agent será executado em um dispositivo Hub IoT configurado em suas instalações.

Conteúdo

HubDeviceArn

A “coisa da Internet das Coisas (IoT)” Arn do stream.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Exigido: Sim

RecorderConfig

A configuração do gravador consiste nos `MediaSourceConfig` detalhes locais, que são usados como credenciais para acessar os arquivos de mídia locais transmitidos na câmera.

Tipo: objeto [RecorderConfig](#)

Obrigatório: Sim

DeletionConfig

A configuração de exclusão é composta pelos detalhes do tempo de retenção (`EdgeRetentionInHours`) e da configuração de tamanho local (`LocalSizeConfig`) usados para fazer a exclusão.

Tipo: objeto [DeletionConfig](#)

Obrigatório: Não

UploaderConfig

A configuração do carregador contém os `ScheduleExpression` detalhes usados para agendar trabalhos de upload dos arquivos de mídia gravados do Edge Agent para um Kinesis Video Stream.

Tipo: objeto [UploaderConfig](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ImageGenerationConfiguration

Serviço: Amazon Kinesis Video Streams

A estrutura que contém as informações necessárias para a entrega das imagens do KVS. Se for nula, a configuração será excluída do fluxo.

Conteúdo

DestinationConfig

A estrutura que contém as informações necessárias para entregar imagens a um cliente.

Tipo: objeto [ImageGenerationDestinationConfig](#)

Obrigatório: Sim

Format

O formato de imagem aceito.

Tipo: strings

Valores Válidos: JPEG | PNG

Obrigatório: Sim

ImageSelectorType

A origem dos registros de data e hora do servidor ou do produtor a serem usados para gerar as imagens.

Tipo: strings

Valores Válidos: SERVER_TIMESTAMP | PRODUCER_TIMESTAMP

Obrigatório: Sim

SamplingInterval

O intervalo de tempo em milissegundos (ms) no qual as imagens precisam ser geradas a partir do fluxo. O valor mínimo que pode ser fornecido é 200 ms. Se o intervalo do timestamp for menor que o intervalo de amostragem, a imagem do StartTimestamp será retornada, se disponível.

Tipo: inteiro

Intervalo válido: valor mínimo de 3000. Valor máximo de 20000.

Obrigatório: Sim

Status

Indica se a `ContinuousImageGenerationConfigurations` API está ativada ou desativada.

Tipo: strings

Valores Válidos: ENABLED | DISABLED

Obrigatório: Sim

FormatConfig

A lista de uma estrutura de pares de valores-chave que contém parâmetros extras que podem ser aplicados quando a imagem é gerada. A `FormatConfig` chave é a `JPEGQuality`, que indica a chave de qualidade JPEG a ser usada para gerar a imagem. O `FormatConfig` valor aceita ints de 1 a 100. Se o valor for 1, a imagem será gerada com menos qualidade e a melhor compressão. Se o valor for 100, a imagem será gerada com a melhor qualidade e menos compressão. Se nenhum valor for fornecido, o valor padrão da `JPEGQuality` chave será definido como 80.

Tipo: mapa de string para string

Entradas do mapa: número máximo de um item.

Chaves válidas: `JPEGQuality`

Restrições de tamanho do valor: tamanho mínimo de 0. O tamanho máximo é 256.

Padrão de valor: `^[a-zA-Z_0-9]+`

Obrigatório: não

HeightPixels

A altura da imagem de saída usada em conjunto com o `WidthPixels` parâmetro. Quando os `WidthPixels` parâmetros `HeightPixels` e parâmetros forem fornecidos, a imagem será esticada para se ajustar à proporção especificada. Se somente o `HeightPixels` parâmetro for fornecido, sua proporção original será usada para calcular a `WidthPixels` proporção. Se nenhum parâmetro for fornecido, o tamanho original da imagem será retornado.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 2160.

Obrigatório: não

WidthPixels

A largura da imagem de saída usada em conjunto com o HeightPixels parâmetro. Quando os HeightPixels parâmetros WidthPixels e parâmetros forem fornecidos, a imagem será esticada para se ajustar à proporção especificada. Se somente o WidthPixels parâmetro for fornecido, sua proporção original será usada para calcular a HeightPixels proporção. Se nenhum parâmetro for fornecido, o tamanho original da imagem será retornado.

Tipo: inteiro

Intervalo válido: valor mínimo de 1. Valor máximo de 3840.

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ImageGenerationDestinationConfig

Serviço: Amazon Kinesis Video Streams

A estrutura que contém as informações necessárias para entregar imagens a um cliente.

Conteúdo

DestinationRegion

A AWS região do bucket do S3 em que as imagens serão entregues. Isso `DestinationRegion` deve corresponder à região em que o riacho está localizado.

Tipo: string

Restrições de comprimento: comprimento mínimo de 9. Comprimento máximo de 14.

Padrão: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Exigido: Sim

Uri

O Uniform Resource Identifier (URI) que identifica onde as imagens serão entregues.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 255.

Padrão: `^[a-zA-Z_0-9]+:(//)?(^[^/]+)/?([^*]*)$`

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

LastRecorderStatus

Serviço: Amazon Kinesis Video Streams

O status mais recente de um trabalho de gravação do Stream's Edge.

Conteúdo

JobStatusDetails

Uma descrição do status mais recente de um trabalho de gravador.

Tipo: String

Obrigatório: não

LastCollectedTime

O registro de data e hora em que o trabalho do gravador foi executado pela última vez e a mídia armazenada no disco local.

Tipo: Timestamp

Obrigatório: não

LastUpdatedTime

A data e hora em que o status do gravador foi atualizado pela última vez.

Tipo: Timestamp

Obrigatório: não

RecorderStatus

O status do trabalho mais recente do gravador.

Tipo: strings

Valores Válidos: SUCCESS | USER_ERROR | SYSTEM_ERROR

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

LastUploaderStatus

Serviço: Amazon Kinesis Video Streams

O status mais recente de um trabalho do stream's edge to cloud uploader.

Conteúdo

JobStatusDetails

Uma descrição do status mais recente de um trabalho de upload.

Tipo: String

Obrigatório: não

LastCollectedTime

O registro de data e hora em que o trabalho do carregador foi executado pela última vez e a mídia coletada na nuvem.

Tipo: Timestamp

Obrigatório: não

LastUpdatedTime

A data e hora em que o status do carregador foi atualizado pela última vez.

Tipo: Timestamp

Obrigatório: não

UploaderStatus

O status do trabalho mais recente do carregador.

Tipo: strings

Valores Válidos: SUCCESS | USER_ERROR | SYSTEM_ERROR

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ListEdgeAgentConfigurationsEdgeConfig

Serviço: Amazon Kinesis Video Streams

Uma descrição da configuração de borda de um único fluxo.

Conteúdo

CreationTime

O carimbo de data/hora em que o stream criou pela primeira vez a configuração de borda.

Tipo: Timestamp

Obrigatório: não

EdgeConfig

Uma descrição da configuração de borda do stream que será usada para sincronizar com o componente Edge Agent IoT Greengrass. O componente Edge Agent será executado em um dispositivo Hub IoT configurado em suas instalações.

Tipo: objeto [EdgeConfig](#)

Obrigatório: Não

FailedStatusDetails

Uma descrição do status de falha gerado.

Tipo: String

Obrigatório: não

LastUpdatedTime

O carimbo de data e hora da última atualização da configuração do edge.

Tipo: Timestamp

Obrigatório: não

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: não

SyncStatus

O status de sincronização atual da configuração de borda do stream.

Tipo: strings

Valores Válidos: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

LocalSizeConfig

Serviço: Amazon Kinesis Video Streams

Os detalhes da configuração que incluem o tamanho máximo da mídia (`MaxLocalMediaSizeInMB`) que você deseja armazenar para um stream no Edge Agent, bem como a estratégia que deve ser usada (`StrategyOnFullSize`) quando o tamanho máximo de um stream for atingido.

Conteúdo

MaxLocalMediaSizeInMB

O tamanho máximo geral da mídia que você deseja armazenar para um stream no Edge Agent.

Tipo: inteiro

Intervalo válido: valor mínimo de 64. Valor máximo de 2000000.

Obrigatório: não

StrategyOnFullSize

A estratégia a ser executada quando o `MaxLocalMediaSizeInMB` limite de um stream é atingido.

Tipo: strings

Valores Válidos: `DELETE_OLDEST_MEDIA` | `DENY_NEW_MEDIA`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

MappedResourceConfigurationListItem

Serviço: Amazon Kinesis Video Streams

Uma estrutura que encapsula ou contém as propriedades de configuração do armazenamento de mídia.

Conteúdo

ARN

O Amazon Resource Name (ARN) do recurso Kinesis Video Stream, associado ao stream.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

Type

O tipo do recurso associado ao stream de vídeo do kinesis.

Tipo: String

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

MediaSourceConfig

Serviço: Amazon Kinesis Video Streams

Os detalhes de configuração que consistem nas credenciais necessárias (`MediaUriSecretArneMediaUriType`) para acessar os arquivos de mídia que são transmitidos para a câmera.

Conteúdo

MediaUriSecretArn

O ARN do AWS Secrets Manager para o nome de usuário e senha da câmera ou a localização de um arquivo de mídia local.

Tipo: string

Restrições de tamanho: tamanho mínimo 20. Tamanho máximo de 2.048.

Padrão: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

Exigido: Sim

MediaUriType

O tipo Uniform Resource Identifier (URI). O `FILE_URI` valor pode ser usado para transmitir arquivos de mídia local.

Note

O Preview suporta apenas o formato URI da fonte de `RTSP_URI` mídia.

Tipo: strings

Valores Válidos: `RTSP_URI` | `FILE_URI`

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

MediaStorageConfiguration

Serviço: Amazon Kinesis Video Streams

Uma estrutura que encapsula ou contém as propriedades de configuração do armazenamento de mídia.

- Se `StorageStatus` estiver ativado, os dados serão armazenados no formato `StreamARN` fornecido. Para que a ingestão de WebRTC funcione, o stream deve ter a retenção de dados ativada.
- Se `StorageStatus` estiver desativado, nenhum dado será armazenado e o `StreamARN` parâmetro não será necessário.

Conteúdo

Status

O status da configuração de armazenamento de mídia.

Tipo: strings

Valores Válidos: ENABLED | DISABLED

Obrigatório: Sim

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

NotificationConfiguration

Serviço: Amazon Kinesis Video Streams

Use essa API para configurar as notificações do Amazon Simple Notification Service (Amazon SNS) para quando os fragmentos ficarem disponíveis em um stream. Se esse parâmetro for nulo, a configuração será excluída do fluxo.

Consulte [Notificações no Kinesis Video Streams](#) para obter mais informações.

Conteúdo

DestinationConfig

As informações de destino necessárias para entregar uma notificação a um cliente.

Tipo: objeto [NotificationDestinationConfig](#)

Obrigatório: Sim

Status

Indica se uma configuração de notificação está ativada ou desativada.

Tipo: sequências

Valores Válidos: ENABLED | DISABLED

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

NotificationDestinationConfig

Serviço: Amazon Kinesis Video Streams

A estrutura que contém as informações necessárias para entregar uma notificação a um cliente.

Conteúdo

Uri

O Uniform Resource Identifier (URI) que identifica onde as imagens serão entregues.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 255.

Padrão: `^[a-zA-Z_0-9]+:(//)?(^[^/]+)/?([^*]*)$`

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

RecorderConfig

Serviço: Amazon Kinesis Video Streams

A configuração do gravador consiste nos `MediaSourceConfig` detalhes locais que são usados como credenciais para acessar os arquivos de mídia locais transmitidos na câmera.

Conteúdo

MediaSourceConfig

Os detalhes de configuração que consistem nas credenciais necessárias (`MediaUriSecretArneMediaUriType`) para acessar os arquivos de mídia transmitidos para a câmera.

Tipo: objeto [MediaSourceConfig](#)

Obrigatório: Sim

ScheduleConfig

A configuração que consiste em `ScheduleExpression` e os `DurationInMinutes` detalhes que especificam o agendamento para gravar de uma câmera ou arquivo de mídia local no Edge Agent. Se o `ScheduleExpression` atributo não for fornecido, o Edge Agent sempre será configurado para o modo de gravação.

Tipo: objeto [ScheduleConfig](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ResourceEndpointListItem

Serviço: Amazon Kinesis Video Streams

Um objeto que descreve o endpoint do canal de sinalização retornado pela API.

`GetSignalingChannelEndpoint`

O endpoint do servidor de mídia corresponderá ao WEBRTC Protocolo.

Conteúdo

Protocol

O protocolo do canal de sinalização retornado pela `GetSignalingChannelEndpoint` API.

Tipo: strings

Valores Válidos: WSS | HTTPS | WEBRTC

Obrigatório: não

ResourceEndpoint

O endpoint do canal de sinalização retornado pela API. `GetSignalingChannelEndpoint`

Tipo: String

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ScheduleConfig

Serviço: Amazon Kinesis Video Streams

Essa API permite que você especifique a duração que a câmera, ou arquivo de mídia local, deve gravar no Edge Agent. O ScheduleConfig consiste nos DurationInMinutes atributos ScheduleExpression e.

Se não ScheduleConfig for fornecido noRecorderConfig, o Edge Agent sempre estará configurado para o modo de gravação.

Se o não ScheduleConfig for fornecido noUploaderConfig, o Edge Agent fará o upload em intervalos regulares (a cada 1 hora).

Conteúdo

DurationInSeconds

A duração total da gravação da mídia. Se o ScheduleExpression atributo for fornecido, o DurationInSeconds atributo também deverá ser especificado.

Tipo: inteiro

Faixa válida: valor mínimo de 60. Valor máximo de 3600.

Obrigatório: Sim

ScheduleExpression

A expressão Quartz cron que cuida do agendamento de trabalhos para gravar da câmera ou arquivo de mídia local no Edge Agent. Se o não ScheduleExpression for fornecido para oRecorderConfig, o Edge Agent sempre estará configurado para o modo de gravação.

Para obter mais informações sobre o Quartz, consulte a página [Tutorial do Cron Trigger](#) para entender as expressões válidas e seu uso.

Tipo: string

Restrições de tamanho: tamanho mínimo de 11. Comprimento máximo de 100.

Padrão: [^\n]{11,100}

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

SingleMasterChannelEndpointConfiguration

Serviço: Amazon Kinesis Video Streams

Um objeto que contém a configuração do endpoint para o tipo de SINGLE_MASTER canal.

Conteúdo

Protocols

Essa propriedade é usada para determinar a natureza da comunicação por esse canal de SINGLE_MASTER sinalização. Se WSS for especificada, essa API retornará um endpoint de websocket. Se HTTPS for especificada, essa API retornará um HTTPS endpoint.

Tipo: Matriz de strings

Membros da Matriz: número mínimo de 1 item. Número máximo de 5 itens.

Valores Válidos: WSS | HTTPS | WEBRTC

Obrigatório: não

Role

Essa propriedade é usada para determinar as permissões de mensagens nesse canal de SINGLE_MASTER sinalização. Se MASTER for especificada, essa API retornará um endpoint que um cliente pode usar para receber ofertas e enviar respostas para qualquer um dos espectadores desse canal de sinalização. Se VIEWER for especificada, essa API retornará um endpoint que um cliente pode usar somente para enviar ofertas para outro MASTER cliente nesse canal de sinalização.

Tipo: strings

Valores Válidos: MASTER | VIEWER

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)

- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

SingleMasterConfiguration

Serviço: Amazon Kinesis Video Streams

Uma estrutura que contém a configuração para o tipo de SINGLE_MASTER canal.

Conteúdo

MessageTtlSeconds

O período de tempo (em segundos) em que um canal de sinalização retém as mensagens não entregues antes de serem descartadas. Use [UpdateSignalingChannel](#) para atualizar esse valor.

Tipo: inteiro

Faixa válida: valor mínimo de 5. Valor máximo de 120.

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

StreamInfo

Serviço: Amazon Kinesis Video Streams

Um objeto que descreve um stream de vídeo do Kinesis.

Conteúdo

CreationTime

Um registro de data e hora que indica quando o fluxo foi criado.

Tipo: Timestamp

Obrigatório: não

DataRetentionInHours

Por quanto tempo o fluxo retém os dados, em horas.

Tipo: inteiro

Intervalo válido: valor mínimo de 0.

Obrigatório: não

DeviceName

O nome da pilha que está associado ao fluxo.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

KmsKeyId

O ID da chave AWS Key Management Service (AWS KMS) que o Kinesis Video Streams usa para criptografar dados no stream.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 2.048.

Padrão: . +

Obrigatório: não

MediaType

O MediaType do fluxo.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `[\w\-\.\.]+/[\w\-\.\.]+(, [\w\-\.\.]+/[\w\-\.\.]+)*`

Obrigatório: não

Status

O status do stream.

Tipo: strings

Valores Válidos: CREATING | ACTIVE | UPDATING | DELETING

Obrigatório: não

StreamARN

O Nome de recurso da Amazon (ARN) do fluxo do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 1.024.

Padrão: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Obrigatório: não

StreamName

O nome do stream do .

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

Version

A versão do fluxo.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 64.

Padrão: [a-zA-Z0-9]+

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

StreamNameCondition

Serviço: Amazon Kinesis Video Streams

Especifica a condição que os fluxos devem satisfazer para serem retornados quando você lista os fluxos (consulte a `ListStreams` API). Uma condição tem uma operação de comparação e um valor. Atualmente, você pode especificar somente o `BEGINS_WITH` operador, que encontra fluxos cujos nomes começam com um determinado prefixo.

Conteúdo

ComparisonOperator

Um operador de comparação. Atualmente, você pode especificar somente o `BEGINS_WITH` operador, que encontra fluxos cujos nomes começam com um determinado prefixo.

Tipo: strings

Valores Válidos: `BEGINS_WITH`

Obrigatório: não

ComparisonValue

Um valor a ser comparado.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: `[a-zA-Z0-9_.-]+`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Tag

Serviço: Amazon Kinesis Video Streams

Um par de chave e valor associado ao canal de sinalização especificado.

Conteúdo

Key

A chave da tag associada ao canal de sinalização especificado.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Exigido: Sim

Value

O valor da tag associada ao canal de sinalização especificado.

Tipo: string

Restrições de tamanho: o tamanho mínimo é 0. O tamanho máximo é 256.

Padrão: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Exigido: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

UploaderConfig

Serviço: Amazon Kinesis Video Streams

A configuração que consiste em `ScheduleExpression` e os `DurationInMinutes` detalhes que especificam o agendamento para gravar de uma câmera ou arquivo de mídia local no Edge Agent. Se o não `ScheduleConfig` for fornecido no `UploaderConfig`, o Edge Agent fará o upload em intervalos regulares (a cada 1 hora).

Conteúdo

ScheduleConfig

A configuração que consiste em `ScheduleExpression` e os `DurationInMinutes` detalhes que especificam o agendamento para gravar de uma câmera ou arquivo de mídia local no Edge Agent. Se não `ScheduleConfig` for fornecido aqui `UploaderConfig`, o Edge Agent fará o upload em intervalos regulares (a cada 1 hora).

Tipo: objeto [ScheduleConfig](#)

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Mídia do Amazon Kinesis Video Streams

Os seguintes tipos de dados são compatíveis com o Amazon Kinesis Video Streams Media:

- [StartSelector](#)

StartSelector

Serviço: Amazon Kinesis Video Streams Media

Identifica a parte no stream de vídeo do Kinesis em que você deseja que a `GetMedia` API comece a retornar dados de mídia. Você tem as seguintes opções para identificar o fragmento inicial:

- Escolha o pedaço mais recente (ou mais antigo).
- Identifique um pedaço específico. Você pode identificar um fragmento específico fornecendo um número de fragmento ou um registro de data e hora (servidor ou produtor).
- Os metadados de cada fragmento incluem um token de continuação como uma tag Matroska (MKV) (). `AWS_KINESISVIDEO_CONTINUATION_TOKEN` Se sua `GetMedia` solicitação anterior foi encerrada, você poderá usar esse valor de tag na próxima `GetMedia` solicitação. A API então começa a retornar partes começando de onde a última API terminou.

Conteúdo

StartSelectorType

Identifica o fragmento no stream de vídeo do Kinesis de onde você deseja começar a obter os dados.

- `AGORA` - Comece com a parte mais recente da transmissão.
- `MAIS ANTIGO` - Comece com a primeira parte disponível na transmissão.
- `FRAGMENT_NUMBER` - Comece com o fragmento após um fragmento específico. Você também deve especificar o parâmetro `AfterFragmentNumber`.
- `PRODUCER_TIMESTAMP` ou `SERVER_TIMESTAMP` - Comece com o fragmento contendo um fragmento com o timestamp especificado do produtor ou servidor. Você especifica o carimbo de data/hora adicionando. `StartTimeStamp`
- `CONTINUATION_TOKEN` - Leia usando o token de continuação especificado.

Note

Se você escolher `NOW`, `EARLIEST` ou `CONTINUATION_TOKEN` como `startSelectorType`, você não fornecerá nenhuma informação adicional no. `startSelector`

Tipo: strings

Valores Válidos: `FRAGMENT_NUMBER` | `SERVER_TIMESTAMP` | `PRODUCER_TIMESTAMP` | `NOW` | `EARLIEST` | `CONTINUATION_TOKEN`

Obrigatório: Sim

AfterFragmentNumber

Especifica o número do fragmento de onde você deseja que a `GetMedia` API comece a devolver os fragmentos.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[0-9]+$`

Obrigatório: não

ContinuationToken

Token de continuação que o Kinesis Video Streams retornou na resposta anterior. `GetMedia` A `GetMedia` API então começa com o fragmento identificado pelo token de continuação.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[a-zA-Z0-9_\.\-]+$`

Obrigatório: não

StartTimestamp

Um valor de timestamp. Esse valor é necessário se você escolher o `PRODUCER_TIMESTAMP` ou o `SERVER_TIMESTAMP` como o `startSelectorType`. Em seguida, a `GetMedia` API começa com o fragmento que contém o fragmento que tem o carimbo de data/hora especificado.

Tipo: Timestamp

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Mídia arquivada do Amazon Kinesis Video Streams

Os seguintes tipos de dados são suportados pela mídia arquivada do Amazon Kinesis Video Streams:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

ClipFragmentSelector

Serviço: Amazon Kinesis Video Streams Archived Media

Descreve o intervalo do timestamp e a origem do timestamp de uma série de fragmentos.

Os fragmentos que têm carimbos de data e hora duplicados do produtor são desduplicados. Isso significa que, se os produtores estiverem produzindo um fluxo de fragmentos com carimbos de data e hora do produtor que sejam aproximadamente iguais à hora real do relógio, o clipe conterá todos os fragmentos dentro do intervalo de timestamp solicitado. Se alguns fragmentos forem ingeridos no mesmo intervalo de tempo e em momentos muito diferentes, somente a coleção de fragmentos ingerida mais antiga será retornada.

Conteúdo

FragmentSelectorType

A origem dos timestamps a serem usados (servidor ou produtor).

Tipo: strings

Valores Válidos: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

Obrigatório: Sim

TimestampRange

O intervalo de timestamps a serem retornados.

Tipo: objeto [ClipTimestampRange](#)

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

ClipTimestampRange

Serviço: Amazon Kinesis Video Streams Archived Media

O intervalo de registros de data e hora para os quais os fragmentos devem ser retornados.

Conteúdo

EndTimeStamp

O fim do intervalo de registro de data e hora da mídia solicitada.

Esse valor deve estar dentro de 24 horas do especificado `StartTimeStamp` e deve ser posterior ao `StartTimeStamp` valor. Se for `FragmentSelectorType` para a solicitação `SERVER_TIMESTAMP`, esse valor deve estar no passado.

Esse valor é inclusivo. O `EndTimeStamp` é comparado ao timestamp (inicial) do fragmento. Fragmentos que começam antes do `EndTimeStamp` valor e continuam depois dele são incluídos na sessão.

Tipo: Timestamp

Obrigatório: Sim

StartTimeStamp

O timestamp inicial na faixa de timestamps para os quais os fragmentos devem ser retornados.

Somente fragmentos que começam exatamente em ou depois `StartTimeStamp` são incluídos na sessão. Fragmentos que começam antes `StartTimeStamp` e continuam depois dela não são incluídos na sessão. Em caso `FragmentSelectorType` `SERVER_TIMESTAMP` afirmativo, `StartTimeStamp` deve ser posterior à cabeceira do riacho.

Tipo: Timestamp

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)

- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

DASHFragmentSelector

Serviço: Amazon Kinesis Video Streams Archived Media

Contém o intervalo de registros de data e hora da mídia solicitada e a origem dos carimbos de data e hora.

Conteúdo

FragmentSelectorType

A fonte dos registros de data e hora da mídia solicitada.

Quando `FragmentSelectorType` está definido como `PRODUCER_TIMESTAMP` e [GetDash StreamingSession URL: PlaybackMode](#) é `ON_DEMAND` ou `LIVE_REPLAY`, o primeiro fragmento ingerido com um timestamp do produtor dentro do especificado [FragmentSelector: TimestampRange](#) é incluído na playlist de mídia. Além disso, os fragmentos com carimbos de data/hora do produtor dentro do `TimestampRange` ingerido imediatamente após o primeiro fragmento (até o [StreamingSessionGetDash URL: value](#)) são incluídos.

MaxManifestFragmentResults

Os fragmentos que têm carimbos de data e hora duplicados do produtor são desduplicados. Isso significa que, se os produtores estiverem produzindo um fluxo de fragmentos com registros de data e hora do produtor que sejam aproximadamente iguais ao horário real do relógio, o manifesto MPEG-DASH conterá todos os fragmentos dentro do intervalo de timestamp solicitado. Se alguns fragmentos forem ingeridos no mesmo intervalo de tempo e em momentos muito diferentes, somente a coleção de fragmentos ingerida mais antiga será retornada.

Quando `FragmentSelectorType` está definido como `PRODUCER_TIMESTAMP` e [GetDash StreamingSession URL: PlaybackMode](#) is `LIVE`, os carimbos de data/hora do produtor são usados nos fragmentos MP4 e para desduplicação. Mas os fragmentos ingeridos mais recentemente com base nos registros de data e hora do servidor estão incluídos no manifesto MPEG-DASH. Isso significa que, mesmo que os fragmentos ingeridos no passado tenham registros de data e hora do produtor com valores atuais, eles não serão incluídos na playlist de mídia do HLS.

O padrão é `SERVER_TIMESTAMP`.

Tipo: String

Valores Válidos: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Obrigatório: não

TimestampRange

O início e o fim do intervalo de timestamp para a mídia solicitada.

Esse valor não deve estar presente se PlaybackType estiver LIVE.

Tipo: objeto [DASHTimestampRange](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

DASHTimestampRange

Serviço: Amazon Kinesis Video Streams Archived Media

O início e o fim do intervalo de timestamp para a mídia solicitada.

Esse valor não deve estar presente se PlaybackType estiver LIVE.

Os valores contidos DASHTimestampRange são inclusivos. Fragmentos que começam exatamente na hora de início ou depois dela são incluídos na sessão. Fragmentos que começam antes da hora de início e continuam após ela não são incluídos na sessão.

Conteúdo

EndTimeStamp

O fim do intervalo de registro de data e hora da mídia solicitada. Esse valor deve estar dentro de 24 horas do especificado StartTimestamp e deve ser posterior ao StartTimestamp valor.

Se for FragmentSelectorType para a solicitação SERVER_TIMESTAMP, esse valor deve estar no passado.

O EndTimestamp valor é obrigatório para o ON_DEMAND modo, mas opcional para o LIVE_REPLAY modo. Se o não EndTimestamp estiver definido para o LIVE_REPLAY modo, a sessão continuará a incluir fragmentos recém-ingeridos até que a sessão expire.

Note

Esse valor é inclusivo. O EndTimestamp é comparado ao timestamp (inicial) do fragmento. Fragmentos que começam antes do EndTimestamp valor e continuam depois dele são incluídos na sessão.

Tipo: Timestamp

Obrigatório: não

StartTimestamp

O início do intervalo de timestamp para a mídia solicitada.

Se o DASHTimestampRange valor for especificado, o StartTimestamp valor será obrigatório.

Somente fragmentos que começam exatamente em ou depois `StartTimestamp` são incluídos na sessão. Fragmentos que começam antes `StartTimestamp` e continuam depois dela não são incluídos na sessão. Em caso `FragmentSelectorType SERVER_TIMESTAMP` afirmativo, `StartTimestamp` deve ser posterior à cabeceira do riacho.

Tipo: `Timestamp`

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Fragment

Serviço: Amazon Kinesis Video Streams Archived Media

Representa um segmento de vídeo ou outros dados delimitados por tempo.

Conteúdo

FragmentLengthInMilliseconds

A duração da reprodução ou outro valor de tempo associado ao fragmento.

Tipo: longo

Obrigatório: não

FragmentNumber

O identificador exclusivo do fragmento. Esse valor aumenta monotonicamente com base na ordem de ingestão.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. O tamanho máximo é 128.

Padrão: `^[0-9]+$`

Obrigatório: não

FragmentSizeInBytes

O tamanho total do fragmento, incluindo informações sobre o fragmento e os dados de mídia contidos.

Tipo: longo

Obrigatório: não

ProducerTimestamp

O timestamp do produtor correspondente ao fragmento, em milissegundos.

Tipo: Timestamp

Obrigatório: não

ServerTimestamp

O timestamp do AWS servidor correspondente ao fragmento, em milissegundos.

Tipo: Timestamp

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

FragmentSelector

Serviço: Amazon Kinesis Video Streams Archived Media

Descreve o intervalo do timestamp e a origem do timestamp de uma série de fragmentos.

Somente fragmentos com um carimbo de data/hora de início maior ou igual à hora de início especificada e menor ou igual à hora de término são retornados. Por exemplo, se um fluxo contiver fragmentos com os seguintes carimbos de data e hora de início:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

Um intervalo seletor de fragmentos com hora de início de 00:00:01 e hora de término de 00:00:04 retornaria os fragmentos com horários de início de 00:00:02 e 00:00:04.

Conteúdo

FragmentSelectorType

A origem dos timestamps a serem usados (servidor ou produtor).

Tipo: strings

Valores Válidos: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

Obrigatório: Sim

TimestampRange

O intervalo de timestamps a serem retornados.

Tipo: objeto [TimestampRange](#)

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

HLSFragmentSelector

Serviço: Amazon Kinesis Video Streams Archived Media

Contém o intervalo de registros de data e hora da mídia solicitada e a origem dos carimbos de data e hora.

Conteúdo

FragmentSelectorType

A fonte dos registros de data e hora da mídia solicitada.

Quando `FragmentSelectorType` está definido como `PRODUCER_TIMESTAMP` e [GetHLS StreamingSession URL: PlaybackMode](#) é `ON_DEMAND` ou `LIVE_REPLAY`, o primeiro fragmento ingerido com um timestamp do produtor dentro do especificado [FragmentSelector: TimestampRange](#) é incluído na playlist de mídia. Além disso, os fragmentos com carimbos de data e hora do produtor dentro do `TimestampRange` ingerido imediatamente após o primeiro fragmento (até o URL do [StreamingSessionGetHLS](#): valor) são incluídos. `MaxMediaPlaylistFragmentResults`

Os fragmentos que têm carimbos de data e hora duplicados do produtor são desduplicados. Isso significa que, se os produtores estiverem produzindo um fluxo de fragmentos com carimbos de data/hora do produtor que sejam aproximadamente iguais à hora real do relógio, as playlists de mídia do HLS conterão todos os fragmentos dentro do intervalo de timestamp solicitado. Se alguns fragmentos forem ingeridos no mesmo intervalo de tempo e em momentos muito diferentes, somente a coleção de fragmentos ingerida mais antiga será retornada.

Quando `FragmentSelectorType` está definido como `PRODUCER_TIMESTAMP` e [GetHLS StreamingSession URL: PlaybackMode isLIVE](#), os carimbos de data/hora do produtor são usados nos fragmentos MP4 e para desduplicação. Mas os fragmentos ingeridos mais recentemente com base nos registros de data e hora do servidor estão incluídos na playlist de mídia do HLS. Isso significa que, mesmo que os fragmentos ingeridos no passado tenham registros de data e hora do produtor com valores atuais, eles não serão incluídos na playlist de mídia do HLS.

O padrão é `SERVER_TIMESTAMP`.

Tipo: String

Valores Válidos: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Obrigatório: não

TimestampRange

O início e o fim do intervalo de timestamp para a mídia solicitada.

Esse valor não deve estar presente se PlaybackType estiver LIVE.

Tipo: objeto [HLSTimestampRange](#)

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

HLSTimestampRange

Serviço: Amazon Kinesis Video Streams Archived Media

O início e o fim do intervalo de timestamp para a mídia solicitada.

Esse valor não deve estar presente se PlaybackType estiver LIVE.

Conteúdo

EndTimeStamp

O fim do intervalo de registro de data e hora da mídia solicitada. Esse valor deve estar dentro de 24 horas do especificado StartTimestamp e deve ser posterior ao StartTimestamp valor.

Se for FragmentSelectorType para a solicitação SERVER_TIMESTAMP, esse valor deve estar no passado.

O EndTimestamp valor é obrigatório para o ON_DEMAND modo, mas opcional para o LIVE_REPLAY modo. Se o não EndTimestamp estiver definido para o LIVE_REPLAY modo, a sessão continuará a incluir fragmentos recém-ingeridos até que a sessão expire.

Note

Esse valor é inclusivo. O EndTimestamp é comparado ao timestamp (inicial) do fragmento. Fragmentos que começam antes do EndTimestamp valor e continuam depois dele são incluídos na sessão.

Tipo: Timestamp

Obrigatório: não

StartTimestamp

O início do intervalo de timestamp para a mídia solicitada.

Se o HLSTimestampRange valor for especificado, o StartTimestamp valor será obrigatório.

Somente fragmentos que começam exatamente em ou depois StartTimestamp são incluídos na sessão. Fragmentos que começam antes StartTimestamp e continuam depois dela não são incluídos na sessão. Em caso FragmentSelectorType SERVER_TIMESTAMP afirmativo, StartTimestamp deve ser posterior à cabeceira do riacho.

Tipo: Timestamp

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Image

Serviço: Amazon Kinesis Video Streams Archived Media

Uma estrutura que contém o `TimestampError`, `ImageContent` e.

Conteúdo

Error

A mensagem de erro exibida quando a imagem do carimbo de data/hora fornecido não foi extraída devido a um erro que não pode ser testado. Um erro será retornado se:

- Não existe mídia para o especificado `Timestamp`.
- A mídia do tempo especificado não permite que uma imagem seja extraída. Nesse caso, a mídia é somente de áudio ou a mídia incorreta foi ingerida.

Tipo: strings

Valores Válidos: `NO_MEDIA` | `MEDIA_ERROR`

Obrigatório: não

ImageContent

Um atributo do `Image` objeto codificado em Base64.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 6291456.

Obrigatório: não

TimeStamp

Um atributo do `Image` objeto usado para extrair uma imagem do stream de vídeo. Esse campo é usado para gerenciar lacunas nas imagens ou para entender melhor a janela de paginação.

Tipo: Timestamp

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

TimestampRange

Serviço: Amazon Kinesis Video Streams Archived Media

O intervalo de registros de data e hora para os quais os fragmentos devem ser retornados.

Conteúdo

EndTimeStamp

O timestamp final na faixa de timestamps para os quais os fragmentos devem ser retornados.

Tipo: Timestamp

Obrigatório: Sim

StartTimeStamp

O timestamp inicial na faixa de timestamps para os quais os fragmentos devem ser retornados.

Tipo: Timestamp

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Amazon Kinesis Video Streams

Os seguintes tipos de dados são suportados pelos canais de sinalização de vídeo do Amazon Kinesis:

- [IceServer](#)

IceServer

Serviço: Amazon Kinesis Video Signaling Channels

Uma estrutura para os dados de conexão do servidor ICE.

Conteúdo

Password

Uma senha para fazer login no servidor ICE.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: não

Ttl

O período de tempo, em segundos, durante o qual o nome de usuário e a senha são válidos.

Tipo: inteiro

Intervalo válido: valor mínimo de 30. Valor máximo de 86.400.

Obrigatório: não

Uris

Uma matriz de URIs, no formato especificado no [I-D. petithuguenin-behave-turn-uris](#) especificação. Esses URIs fornecem os diferentes endereços e/ou protocolos que podem ser usados para acessar o servidor TURN.

Tipo: matriz de strings

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Obrigatório: não

Username

Um nome de usuário para fazer login no servidor ICE.

Tipo: string

Restrições de tamanho: tamanho mínimo 1. Comprimento máximo de 256.

Padrão: [a-zA-Z0-9_.-]+

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos AWS SDKs específicos da linguagem, consulte o seguinte:

- [AWS SDK for C++](#)
- [AWS SDK para Java V2](#)
- [AWS SDK para Ruby V3](#)

Amazon Kinesis WebRTC Streams

Os seguintes tipos de dados são suportados pelo Amazon Kinesis Video WebRTC Storage:

Erros comuns

Esta seção lista os erros comuns às ações de API de todos os serviços da AWS. Para saber os erros específicos de uma ação de API para esse serviço, consulte o tópico sobre a ação de API em questão.

AccessDeniedException

Você não tem acesso suficiente para executar essa ação.

Código de status HTTP: 400

IncompleteSignature

A assinatura da solicitação não segue os padrões da AWS.

Código de status HTTP: 400

InternalFailure

O processamento da solicitação falhou por causa de um erro, uma exceção ou uma falha desconhecida.

Código de status HTTP: 500

InvalidAction

A ação ou operação solicitada é inválida. Verifique se a ação foi digitada corretamente.

Código de status HTTP: 400

InvalidClientTokenId

O certificado X.509 ou o ID de chave de acesso da AWS fornecido não existe em nossos registros.

Código de status HTTP: 403

NotAuthorized

Você não tem permissão para realizar esta ação.

Código de status HTTP: 400

OptInRequired

O ID da chave de acesso da AWS precisa de uma assinatura do serviço.

Código de status HTTP: 403

RequestExpired

A solicitação atingiu o serviço mais de 15 minutos após a data na solicitação ou mais de 15 minutos após a data de expiração da solicitação (como para URLs predeterminados), ou a data na solicitação está a mais de 15 minutos no futuro.

Código de status HTTP: 400

ServiceUnavailable

Falha na solicitação devido a um erro temporário do servidor.

Código de status HTTP: 503

ThrottlingException

A solicitação foi negada devido à limitação da solicitação.

Código de status HTTP: 400

ValidationError

A entrada não atende às restrições especificadas por um serviço da AWS.

Código de status HTTP: 400

Parâmetros gerais

A lista a seguir contém os parâmetros que todas as ações usam para assinar solicitações do Signature versão 4 com uma string de consulta. Todos os parâmetros específicos de uma ação são listados no tópico para a ação. Para obter mais informações sobre o Signature versão 4, consulte [Solicitações de AWS API de assinatura](#) no Guia do usuário do IAM.

Action

A ação a ser executada.

Tipo: string

Obrigatório: sim

Version

A versão da API para a qual a solicitação foi escrita, expressa no formato AAAA-MM-DD.

Tipo: string

Obrigatório: sim

X-Amz-Algorithm

O algoritmo de hash que foi usado para criar a assinatura da solicitação.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Valores válidos: AWS4-HMAC-SHA256

Obrigatório: Condicional

X-Amz-Credential

O valor de escopo da credencial, uma string que inclui a sua chave de acesso, a data, a região visada, o serviço que está sendo solicitado e uma sequência de encerramento ("aws4_request"). O valor é expresso no seguinte formato: `chave_aceso/AAAAMMDD/região/serviço/aws4_request`.

Para obter mais informações, consulte [Criar uma solicitação deAWS API assinada](#) no Guia do usuário do IAM.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

X-Amz-Date

A data usada para criar a assinatura. O formato deve ser o formato básico ISO 8601 (AAAAMMDD'T'HHMMSS'Z'). Por exemplo, a data/hora a seguir é um valor X-Amz-Date válido: `20120325T120000Z`.

Condição: X-Amz-Date é opcional para todas as solicitações e pode ser usado para substituir a data usada para assinar solicitações. Se o cabeçalho Date (Data) for especificado no formato básico ISO 8601, o valor X-Amz-Date não será necessário. Quando X-Amz-Date é usado, sempre substitui o valor do cabeçalho Date (Data). Para obter mais informações, consulte [Elementos de uma assinatura de solicitação deAWS API](#) no Guia do usuário do IAM.

Tipo: string

Obrigatório: Condicional

X-Amz-Security-Token

O token de segurança temporário que foi obtido por meio de uma chamada paraAWS Security Token Service (AWS STS). Para obter uma lista de serviços que são compatíveis com credenciais de segurança temporárias doAWS STS, vá para [Serviços da AWSque funcionam com o IAM](#) no Manual do usuário do IAM.

Condição: se estiver usando credenciais de segurança temporárias doAWS STS, você deverá incluir o token de segurança.

Tipo: string

Obrigatório: Condicional

X-Amz-Signature

Especifica a assinatura com codificação hexadecimal que foi calculada com base na string a ser assinada e na chave de assinatura derivada.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

X-Amz-SignedHeaders

Especifica todos os cabeçalhos HTTP que foram incluídos como parte da solicitação canônica. Para obter mais informações sobre como especificar cabeçalhos assinados, consulte [Criar uma solicitação deAWS API assinada](#) no Guia do usuário do IAM.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.