



Panduan Developer

# Amazon Kinesis Video Streams



# Amazon Kinesis Video Streams: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

---

# Table of Contents

Apa itu Kinesis Video Streams? .....	1
Ketersediaan Wilayah .....	2
Apakah Anda pengguna pertama kali Kinesis Video Streams? .....	3
Persyaratan Sistem .....	5
Persyaratan kamera .....	5
Sistem operasi yang diuji .....	6
Persyaratan penyimpanan SDK .....	6
Cara Kerjanya .....	7
Pustaka API dan Produser .....	8
Kinesis Video Streams API .....	8
Pustaka produser .....	11
Pemutaran Video .....	11
Persyaratan pemutaran .....	12
Pemutaran video dengan HLS .....	14
Pemutaran Video dengan MPEG-DASH .....	25
Menggunakan Metadata Streaming .....	29
Menambahkan metadata ke aliran video Kinesis .....	30
Mengonsumsi metadata yang disematkan dalam aliran video Kinesis .....	31
Batas metadata streaming .....	33
Model Data .....	33
Elemen header aliran .....	33
Alirkan data trek .....	39
Elemen header bingkai .....	40
Data bingkai MKV .....	41
Memulai .....	42
Menyiapkan akun .....	42
Mendaftar untuk Akun AWS .....	43
Membuat pengguna administratif .....	43
Buat Akun AWS kunci .....	44
Buat aliran video Kinesis .....	44
Buat aliran video menggunakan konsol .....	45
Buat aliran video menggunakan AWS CLI .....	45
Kirim data ke aliran video Amazon Kinesis .....	46
Membangun SDK dan sampel .....	46

Jalankan sampel untuk mengunggah media ke Kinesis Video Streams .....	49
Tinjau objek pengakuan .....	50
Konsumsi data media .....	51
Lihat media di konsol .....	51
Konsumsi data media menggunakan HLS .....	51
Agen Edge .....	52
Operasi API Agen Edge Streams Amazon Kinesis Video Streams .....	53
Memantau Agen Edge Streams Video Amazon Kinesis .....	53
Terapkan dalam mode non- AWS IoT Greengrass .....	53
1. Instal dependensi .....	54
2. Buat sumber daya untuk URL RTSP kamera IP Anda .....	55
3. Membuat kebijakan izin IAM .....	57
4. Membuat peran IAM .....	59
5. Buat alias AWS IoT peran .....	61
6. Buat AWS IoT kebijakan .....	61
7. Buat AWS IoT sesuatu dan dapatkan AWS IoT Core kredensial .....	63
8. Bangun dan jalankan Agen Edge Amazon Kinesis Video Streams .....	66
9. (Opsional) Instal CloudWatch agen .....	77
10. (Opsional) Jalankan Agen Edge Streams Amazon Kinesis Video .....	80
Menyebarkan ke AWS IoT Greengrass .....	82
1. Buat instance Ubuntu .....	83
2. Siapkan perangkat AWS IoT Greengrass inti .....	84
3. Buat sumber daya untuk URL RTSP kamera IP Anda .....	85
4. Tambahkan izin ke peran TES .....	87
5. Instal komponen Secret Manager .....	90
6. Terapkan Agen Edge Streams Video Amazon Kinesis di perangkat .....	93
7. (Opsional) Instal komponen pengelola AWS IoT Greengrass log .....	101
Pertanyaan yang Sering Diajukan .....	105
Sistem operasi apa yang didukung Amazon Kinesis Video Streams Edge Agent? .....	105
Apakah Agen Edge Amazon Kinesis Video Streams mendukung media H.265? .....	105
Apakah Agen Edge Streams Video Amazon Kinesis berfungsi di AL2? .....	106
Bagaimana saya bisa menjalankan beberapa aliran di dalam AWS IoT benda atau perangkat? .....	106
Bagaimana saya bisa mengedit StartEdgeConfigurationUpdate setelah dikirim? .....	106
Apakah Anda memiliki contoh umumScheduleConfigs? .....	106
Apakah ada batas aliran maksimum? .....	107

Bagaimana cara memulai kembali pekerjaan yang telah salah? .....	107
Bagaimana cara memantau kesehatan Agen Tepi Amazon Kinesis Video Streams saya? ...	108
Streaming video melalui VPC .....	109
Informasi tambahan .....	109
Prosedur titik akhir VPC .....	110
Citra .....	112
Memulai dengan GetImages .....	112
Memulai pengiriman Amazon S3 .....	113
UpdateImageGenerationConfiguration .....	113
DescribeImageGenerationConfiguration .....	115
Tag MKV produser .....	116
Menambahkan tag metadata di Producer SDK menggunakanPutEventMetaData .....	117
Batas .....	117
Metadata objek S3 .....	117
S3 objek jalan (gambar) .....	118
Rekomendasi URI Amazon S3 untuk melindungi dari pembatasan .....	118
Notifikasi .....	120
UpdateNotificationConfiguration .....	120
DescribeNotificationConfiguration .....	120
Tag MKV Produser .....	116
Sintaks untuk tag MKV Produser .....	116
Batas tag MKV .....	121
.....	121
.....	121
Payload topik Amazon SNS .....	121
Melihat pesan Amazon SNS Anda .....	122
Keamanan .....	124
Perlindungan Data .....	125
Apa itu enkripsi sisi server untuk Kinesis Video Streams? .....	125
Biaya, Wilayah, dan pertimbangan kinerja .....	125
Bagaimana cara memulai dengan enkripsi sisi server? .....	126
Membuat dan menggunakan kunci KMS buatan pengguna .....	127
Izin untuk menggunakan kunci KMS buatan pengguna .....	128
Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan IAM .....	129
Sintaksis kebijakan .....	130
Tindakan untuk Kinesis Video Streams .....	131

Nama Sumber Daya Amazon (ARN) untuk Kinesis Video Streams .....	131
Memberikan akses akun IAM lainnya ke aliran video Kinesis .....	132
Contoh kebijakan .....	132
Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan AWS IoT .....	134
AWS IoT ThingName sebagai nama aliran .....	135
AWS IoT CertificateId sebagai nama aliran .....	142
Gunakan AWS IoT kredensi untuk melakukan streaming ke nama aliran hard-coded .....	143
Memantau .....	144
Validasi Kepatuhan .....	144
Ketangguhan .....	146
Keamanan Infrastruktur .....	146
Praktik Terbaik Keamanan .....	146
Terapkan akses hak akses paling rendah .....	147
Gunakan IAM role .....	147
Gunakan CloudTrail untuk memantau panggilan API .....	147
Pustaka Produser .....	149
Klien Produser Aliran Video Kinesis .....	149
Perpustakaan Produser Aliran Video Kinesis .....	150
Topik terkait .....	151
Perpustakaan Produser Java .....	151
Prosedur: Menggunakan Java Producer SDK .....	151
Langkah 1: Unduh dan Konfigurasi Kode .....	153
Langkah 2: Tulis dan periksa kodenya .....	154
Langkah 3: Jalankan dan verifikasi kode .....	156
Pustaka Produser Android .....	156
Prosedur: Menggunakan Android Producer SDK .....	157
Prasyarat .....	157
Langkah 1: Unduh dan Konfigurasi Kode .....	161
Langkah 2: Periksa kodenya .....	162
Langkah 3: Jalankan dan verifikasi kode .....	164
Pustaka Produser C++ .....	166
Model objek .....	166
Menempatkan media ke dalam aliran .....	166
Antarmuka Panggilan Balik .....	166
Prosedur: Menggunakan C++ Produser SDK .....	167
Langkah 1: Mengunduh dan Mengkonfigurasi Kode .....	170

Langkah 2: Tulis dan periksa kode .....	170
Langkah 3: Jalankan dan verifikasi kode .....	177
Menggunakan C++ Producer SDK sebagai plugin GStreamer .....	177
Menggunakan C++ Producer SDK sebagai plugin GStreamer dalam wadah Docker .....	177
Menggunakan Pencatatan .....	178
Perpustakaan Produser .....	179
Model objek .....	179
Menempatkan media ke dalam aliran .....	180
Prosedur: Menggunakan SDK Produser C .....	180
Langkah 1: Mengunduh Kode .....	183
Langkah 2: Tulis dan periksa kode .....	183
Langkah 3: Jalankan dan verifikasi kode .....	186
SDK Produser C ++ di Raspberry Pi .....	188
Prasyarat .....	189
Buat pengguna IAM dengan izin untuk menulis ke Kinesis Video Streams .....	189
Bergabunglah dengan Raspberry Pi Anda ke jaringan Wi-Fi Anda .....	191
Connect dari jarak jauh ke Raspberry Pi .....	192
Konfigurasi kamera Raspberry Pi .....	192
Instal prasyarat perangkat lunak .....	193
Unduh dan buat Kinesis Video Streams C++ Producer SDK .....	194
Streaming video ke streaming video Kinesis Anda dan lihat streaming langsung .....	195
Referensi .....	196
Batas SDK produsen .....	196
Referensi kode kesalahan .....	199
Bendera Adaptasi NAL .....	252
Struktur Produsen .....	253
Struktur Aliran .....	256
Callback .....	276
Perpustakaan Parser .....	284
Prosedur: Menggunakan Perpustakaan Parser video Kinesis .....	284
Prasyarat .....	284
Langkah 1: Mengunduh dan mengatur kode .....	285
Langkah selanjutnya .....	285
Langkah 2: Tulis dan periksa kode .....	285
StreamingMkvReader .....	286
FragmentMetadataVisitor .....	286

OutputSegmentMerger .....	288
KinesisVideoExample .....	289
Langkah selanjutnya .....	292
Langkah 3: Jalankan dan verifikasi kode .....	292
Contoh-contoh .....	294
Contoh: Mengirim data ke Kinesis Video Streams .....	294
Contoh: Mengambil data dari Kinesis Video Streams .....	294
Contoh: Memutar kembali data video .....	294
Prasyarat .....	295
GStreamer .....	295
Unduh, buat, dan konfigurasi elemen GStreamer .....	296
Jalankan elemen GStreamer .....	297
Luncurkan Perintah .....	297
Jalankan elemen GStreamer dalam wadah Docker .....	300
Referensi Parameter .....	303
PutMedia API .....	316
Langkah 1: Unduh dan konfigurasi kode .....	317
Langkah 2: Tulis dan periksa kodenya .....	318
Langkah 3: Jalankan dan verifikasi kode .....	320
RTSP dan Docker .....	321
Prasyarat .....	321
Bangun gambar Docker .....	321
Jalankan aplikasi contoh RTSP .....	322
Penyaji .....	323
Prasyarat .....	324
Menjalankan contoh penyaji .....	324
Cara Kerjanya .....	325
SageMaker .....	327
Prasyarat .....	328
Membuat aplikasi .....	329
Memantau aplikasi .....	331
Memperluas aplikasi .....	333
Membersihkan aplikasi .....	334
Memantau .....	335
Memantau metrik dengan CloudWatch .....	335
CloudWatch panduan metrik .....	351



Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch .....	355
CloudWatch panduan metrik untuk Agen Edge Streams Amazon Kinesis Video Streams ....	358
Pencatatan Panggilan API dengan CloudTrail .....	360
Amazon Kinesis Video Streams dan CloudTrail .....	361
Contoh: Entri file log Amazon Kinesis Video Streams .....	362
Kuota .....	366
Kuota layanan API bidang kontrol .....	366
Kuota layanan API media dan media arsip .....	371
Kuota fragmen-metadana dan fragmen-media .....	375
Kuota metadana fragmen .....	378
Tag Stream .....	378
Pemecahan Masalah .....	379
Memecahkan masalah umum .....	379
Latensi terlalu tinggi .....	379
Memecahkan masalah API .....	380
Kesalahan: “Opsinya tidak dikenal” .....	380
Kesalahan: “Tidak dapat menentukan nama layanan/operasi yang akan diotorisasi” .....	380
Kesalahan: “Gagal menempatkan bingkai di aliran” .....	381
Kesalahan: “Layanan menutup koneksi sebelum final AckEvent diterima” .....	381
Kesalahan: “STATUS_STORE_OUT_OF_MEMORY” .....	381
Memecahkan masalah HLS .....	382
Memecahkan masalah Java .....	382
Mengaktifkan log Java .....	382
Memecahkan masalah Pustaka Produser .....	383
Tidak dapat mengkompilasi SDK Produser .....	384
Aliran video tidak muncul di konsol .....	384
Kesalahan: “Token keamanan yang disertakan dalam permintaan tidak valid” saat streaming data menggunakan aplikasi demo GStreamer .....	384
Kesalahan: “Gagal mengirimkan bingkai ke klien Video Kinesis” .....	385
Aplikasi GStreamer berhenti dengan pesan “streaming berhenti, alasan tidak dinegosiasikan” di OS X .....	385
Kesalahan: “Gagal mengalokasikan heap” saat membuat Klien Video Kinesis di demo GStreamer di Raspberry Pi .....	386
Kesalahan: “Instruksi ilegal” saat menjalankan demo GStreamer di Raspberry Pi .....	386
Kamera gagal dimuat di Raspberry Pi .....	386
Kamera tidak dapat ditemukan di macOS High Sierra .....	387

file jni.h tidak ditemukan saat mengkompilasi di macOS High Sierra .....	387
Kesalahan keriting saat menjalankan aplikasi demo GStreamer .....	387
Pernyataan stempel waktu/rentang saat runtime di Raspberry Pi .....	387
Pernyataan tentang gst_value_set_fraction_range_full pada Raspberry Pi .....	387
Kesalahan STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) di Android .....	388
Durasi fragmen maksimum tercapai kesalahan .....	388
Kesalahan “Nama benda tidak valid lulus” saat menggunakan otorisasi IoT .....	389
Memecahkan masalah Stream Parser Library .....	389
Tidak dapat mengakses satu frame dari aliran .....	389
Kesalahan decoding fragmen .....	389
Riwayat Dokumen .....	391
Referensi API .....	396
Tindakan .....	396
Amazon Kinesis Video Streams .....	397
Arus Amazon Kinesis Video Streams .....	520
Arus Amazon Kinesis Video Streams .....	537
Saluran Pensinyalan Video Amazon Kinesis .....	583
Penyimpanan WebRTC Amazon .....	592
Tipe Data .....	596
Amazon Kinesis Video Streams .....	597
Arus Amazon Kinesis Video Streams .....	641
Arus Amazon Kinesis Video Streams .....	644
Amazon Kinesis Video .....	662
Penyimpanan Video WebRTC Amazon .....	664
Kesalahan Umum .....	664
Parameter Umum .....	666
.....	dclxix

# Apa itu Kinesis Video Streams?

Anda dapat menggunakan Amazon Kinesis Video Streams, yang dikelola Layanan AWS sepenuhnya, untuk melakukan streaming video langsung dari perangkat ke perangkat, atau membuat aplikasi AWS Cloud untuk pemrosesan video real-time atau analitik video berorientasi batch.

Kinesis Video Streams tidak hanya penyimpanan untuk data video. Anda dapat menggunakannya untuk menonton streaming video Anda secara real time saat diterima di cloud. Anda dapat memantau streaming langsung Anda di AWS Management Console, atau mengembangkan aplikasi pemantauan Anda sendiri yang menggunakan pustaka API Kinesis Video Streams untuk menampilkan video langsung.

Anda dapat menggunakan Kinesis Video Streams untuk menangkap sejumlah besar data video langsung dari jutaan sumber, termasuk smartphone, kamera keamanan, webcam, kamera yang disematkan di mobil, drone, dan sumber lainnya. Anda juga dapat mengirim data non-video, berseri waktu seperti data audio, citra termal, data kedalaman, dan data RADAR. Saat streaming video langsung dari sumber-sumber ini ke aliran video Kinesis, Anda dapat membangun aplikasi untuk mengakses data frame-by-frame, secara real time untuk pemrosesan latensi rendah. Kinesis Video Streams adalah sumber-agnostik. Anda dapat melakukan streaming video dari webcam komputer menggunakan [GStreamer](#) perpustakaan, atau dari kamera di jaringan Anda menggunakan protokol streaming real-time (RTSP).

Anda juga dapat mengonfigurasi aliran video Kinesis untuk menyimpan data media secara tahan lama selama periode retensi yang ditentukan. Kinesis Video Streams secara otomatis menyimpan data ini dan mengenkripsinya saat istirahat. Selain itu, indeks waktu Kinesis Video Streams menyimpan data berdasarkan stempel waktu produsen dan stempel waktu konsumsi. Anda dapat membangun aplikasi yang memproses data video secara berkala, atau Anda dapat membuat aplikasi yang memerlukan akses satu kali ke data historis untuk berbagai kasus penggunaan.

Aplikasi kustom Anda, real-time atau berorientasi batch, dapat berjalan di instans Amazon EC2. Aplikasi ini dapat memproses data menggunakan open source, algoritma pembelajaran mendalam, atau menggunakan aplikasi pihak ketiga yang terintegrasi dengan Kinesis Video Streams.

Manfaat menggunakan Kinesis Video Streams meliputi:

- **Connect dan streaming dari jutaan perangkat** — Anda dapat menggunakan Kinesis Video Streams untuk menghubungkan dan mengalirkan video, audio, dan data lainnya dari jutaan perangkat mulai dari smartphone konsumen, drone, dan kamera dasbor. Anda dapat menggunakan pustaka

produser Kinesis Video Streams untuk mengonfigurasi perangkat Anda dan melakukan streaming secara andal secara real time, atau sebagai unggahan media. after-the-fact

- Menyimpan, mengenkripsi, dan mengindeks data secara tahan lama — Anda dapat mengonfigurasi aliran video Kinesis Anda untuk menyimpan data media secara tahan lama selama periode retensi kustom. Kinesis Video Streams juga menghasilkan indeks atas data yang disimpan berdasarkan stempel waktu buatan produser atau sisi layanan. Aplikasi Anda dapat mengambil data tertentu dalam aliran menggunakan indeks waktu.
- Fokus pada pengelolaan aplikasi alih-alih infrastruktur — Kinesis Video Streams tanpa server, jadi tidak ada infrastruktur untuk disiapkan atau dikelola. Anda tidak perlu khawatir tentang penerapan, konfigurasi, atau penskalaan elastis infrastruktur yang mendasarinya karena aliran data Anda dan jumlah aplikasi yang dikonsumsi bertambah dan menyusut. Kinesis Video Streams secara otomatis melakukan semua administrasi dan pemeliharaan yang diperlukan untuk mengelola aliran, sehingga Anda dapat fokus pada aplikasi, bukan infrastruktur.
- Membangun aplikasi real-time dan batch pada aliran data — Anda dapat menggunakan Kinesis Video Streams untuk membangun aplikasi real-time kustom yang beroperasi pada streaming data langsung, dan membuat aplikasi batch atau satu kali yang beroperasi pada data tahan lama tanpa persyaratan latensi yang ketat. Anda dapat membangun, menyebarkan, dan mengelola aplikasi kustom: open source (Apache MxNet, OpenCV), homegrown, atau solusi pihak ketiga menggunakan untuk memproses dan menganalisis aliran Anda. AWS Marketplace Anda dapat menggunakan Kinesis Get Video Streams API untuk membangun beberapa aplikasi bersamaan yang memproses data secara real-time atau berorientasi batch.
- Streaming data dengan lebih aman — Kinesis Video Streams mengenkripsi semua data saat mengalir melalui layanan dan saat data tetap ada. Kinesis Video Streams memberlakukan enkripsi berbasis Transport Layer Security (TLS) pada streaming data dari perangkat, dan mengenkripsi semua data saat istirahat menggunakan (). AWS Key Management Service AWS KMS Selain itu, Anda dapat mengelola akses ke data Anda menggunakan AWS Identity and Access Management (IAM).
- Bayar saat Anda pergi — Untuk informasi lebih lanjut, lihat [AWS Pricing Calculator](#).

## Ketersediaan Wilayah

Amazon Kinesis Video Streams tersedia di wilayah berikut:

Nama Wilayah	AWSKode Wilayah
Timur AS (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
AS Barat (Oregon)	as-barat-2
Afrika (Cape Town)	af-selatan-1
Asia Pasifik (Hong Kong)	ap-timur-1
Asia Pasifik (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-sentral-1
China (Beijing)	cn-north-1
Eropa (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Amerika Selatan (Sao Paulo)	sa-east-1

## Apakah Anda pengguna pertama kali Kinesis Video Streams?

Jika Anda adalah pengguna pertama kali Kinesis Video Streams, kami sarankan Anda membaca bagian berikut secara berurutan:

1. [Kinesis Video Streams: Cara kerjanya](#)— Untuk mempelajari tentang konsep Kinesis Video Streams.
2. [Memulai dengan Amazon Kinesis Video Streams](#)— Untuk mengatur akun Anda dan menguji Kinesis Video Streams.
3. [Perpustakaan Produser Aliran Video Kinesis](#)— Untuk mempelajari tentang membuat aplikasi produser Kinesis Video Streams.
4. [Perpustakaan Parser aliran video](#)— Untuk mempelajari tentang memproses frame data yang masuk dalam aplikasi konsumen Kinesis Video Streams.
5. [Contoh Aliran Video Amazon Kinesis](#)— Untuk melihat lebih banyak contoh tentang apa yang dapat Anda lakukan dengan Kinesis Video Streams.

# Persyaratan sistem Kinesis Video Streams

Bagian berikut berisi persyaratan perangkat keras, perangkat lunak, dan penyimpanan untuk Amazon Kinesis Video Streams.

Topik

- [Persyaratan kamera](#)
- [Sistem operasi yang diuji](#)
- [Persyaratan penyimpanan SDK](#)

## Persyaratan kamera

Kamera yang digunakan untuk menjalankan Kinesis Video Streams Producer SDK dan sampel memiliki persyaratan memori berikut:

- Tampilan konten SDK membutuhkan memori 16 MB.
- Konfigurasi default aplikasi sampel adalah 128 MiB memori. Nilai ini sesuai untuk produsen yang memiliki konektivitas jaringan yang baik dan tidak ada persyaratan untuk buffering tambahan. Jika konektivitas jaringan buruk dan lebih banyak buffering diperlukan, Anda dapat menghitung kebutuhan memori per detik buffering dengan mengalikan frame rate per detik dengan ukuran memori frame. Untuk informasi lebih lanjut tentang mengalokasikan memori, lihat [StorageInfo](#)

Sebaiknya gunakan kamera USB atau RTSP (Real Time Streaming Protocol) yang menyandikan data menggunakan H.264 karena ini menghilangkan beban kerja pengkodean dari CPU.

Saat ini, aplikasi demo tidak mendukung User Datagram Protocol (UDP) untuk streaming RTSP. Kemampuan ini akan ditambahkan di masa depan.

Producer SDK mendukung jenis kamera berikut:

- Kamera web.
- Kamera USB.
- Kamera dengan pengkodean H.264 (lebih disukai).
- Kamera tanpa pengkodean H.264.

- Modul kamera Raspberry Pi. Ini lebih disukai untuk perangkat Raspberry Pi karena terhubung ke GPU untuk transfer data video, sehingga tidak ada overhead untuk pemrosesan CPU.
- Kamera RTSP (jaringan). Kamera ini lebih disukai karena aliran video sudah dikodekan dengan H.264.

## Sistem operasi yang diuji

Kami telah menguji kamera web dan kamera RTSP dengan perangkat dan sistem operasi berikut:

- Mac mini
  - Sierra tinggi
- MacBook Laptop Pro
  - Sierra (10.12)
  - Kapitan (10.11)
- Laptop HP yang menjalankan Ubuntu 16.04
- Ubuntu 17.10 (wadah Docker)
- Raspberry Pi 3

## Persyaratan penyimpanan SDK

Instalasi [Perpustakaan Produser Aliran Video Kinesis](#) memiliki persyaratan penyimpanan minimum 170 MB dan persyaratan penyimpanan yang disarankan 512 MB.



# Kinesis Video Streams: Cara kerjanya

## Topik

- [Kinesis Video Streams API dan pustaka produser mendukung](#)
- [Pemutaran Kinesis Video Streams](#)
- [Menggunakan metadata streaming dengan Kinesis Video Streams](#)
- [Model data Kinesis Video Streams](#)

Anda dapat menggunakan Amazon Kinesis Video Streams, yang dikelola sepenuhnya Layanan AWS, untuk melakukan streaming video langsung dari perangkat ke AWS Cloud dan menyimpannya dengan tahan lama. Anda kemudian dapat membuat aplikasi Anda sendiri untuk pemrosesan video real-time atau melakukan analisis video berorientasi batch.

Diagram berikut memberikan gambaran umum tentang cara kerja Kinesis Video Streams.

Diagram menunjukkan interaksi antara komponen-komponen berikut:

- Produser - Sumber apa pun yang menempatkan data ke dalam aliran video Kinesis. Seorang produser dapat berupa perangkat penghasil video apa pun, seperti kamera keamanan, kamera yang dikenakan di tubuh, kamera ponsel cerdas, atau kamera dasbor. Produser juga dapat mengirim data non-video, seperti umpan audio, gambar, atau data RADAR.

Seorang produser tunggal dapat menghasilkan satu atau lebih aliran video. Misalnya, kamera video dapat mendorong data video ke satu aliran video Kinesis dan data audio ke yang lain.

- Perpustakaan Kinesis Video Streams Produser — Satu set perangkat lunak dan pustaka yang dapat Anda instal dan konfigurasi di perangkat Anda. Anda dapat menggunakan pustaka ini untuk terhubung dengan aman dan streaming video dengan cara yang berbeda, termasuk secara real time, setelah buffering selama beberapa detik, atau sebagai after-the-fact unggahan media.
- Aliran video Kinesis - Sumber daya yang dapat Anda gunakan untuk mengangkut data video langsung, menyimpannya secara opsional, dan membuat data tersedia untuk dikonsumsi baik secara real time maupun secara batch atau satu kali. Dalam konfigurasi tipikal, aliran video Kinesis hanya memiliki satu produser yang menerbitkan data ke dalamnya.

Aliran dapat membawa audio, video, dan aliran data yang dikodekan waktu serupa, seperti umpan penginderaan kedalaman, umpan RADAR, dan banyak lagi. Anda membuat aliran video Kinesis menggunakan AWS Management Console atau pemrograman menggunakan SDK. AWS

Beberapa aplikasi independen dapat mengonsumsi aliran video Kinesis secara parallel.

- Konsumen - Mendapatkan data, seperti fragmen dan bingkai, dari aliran video Kinesis untuk melihat, memproses, atau menganalisisnya. Umumnya konsumen ini disebut aplikasi Kinesis Video Streams. Anda dapat menulis aplikasi yang mengonsumsi dan memproses data di Kinesis Video Streams secara real time, atau setelah data disimpan dan diindeks waktu ketika pemrosesan latensi rendah tidak diperlukan. Anda dapat membuat aplikasi konsumen tersebut untuk dijalankan di instans Amazon EC2.
- [Perpustakaan Parser aliran video](#)- Memungkinkan aplikasi Kinesis Video Streams untuk mendapatkan media secara andal dari aliran video Kinesis dengan cara latensi rendah. Selain itu, ini mem-parsing batas bingkai di media sehingga aplikasi dapat fokus pada pemrosesan dan analisis frame itu sendiri.

## Kinesis Video Streams API dan pustaka produser mendukung

Kinesis Video Streams menyediakan API bagi Anda untuk membuat dan mengelola aliran dan membaca atau menulis data media ke dan dari stream. Konsol Kinesis Video Streams, selain fungsionalitas administrasi, juga mendukung live dan video-on-demand playback. Kinesis Video Streams juga menyediakan serangkaian pustaka produser yang dapat Anda gunakan dalam kode aplikasi Anda untuk mengekstrak data dari sumber media Anda dan mengunggah ke aliran video Kinesis Anda.

Topik

- [Kinesis Video Streams API](#)
- [Pustaka produser](#)

## Kinesis Video Streams API

Kinesis Video Streams menyediakan API untuk membuat dan mengelola Kinesis Video Streams. Ini juga menyediakan API untuk membaca dan menulis data media ke aliran, sebagai berikut:

- **API Produser** - Kinesis Video Streams menyediakan PutMedia API untuk menulis data media ke aliran video Kinesis. Dalam PutMedia permintaan, produser mengirimkan aliran fragmen media. Fragmen adalah urutan bingkai mandiri. Bingkai milik fragmen seharusnya tidak memiliki ketergantungan pada bingkai apa pun dari fragmen lain. Untuk informasi selengkapnya, lihat [PutMedia](#).

Saat fragmen tiba, Kinesis Video Streams memberikan nomor fragmen unik, dalam urutan penambahannya dalam urutan penambahannya. Ini juga menyimpan stempel waktu sisi produser dan sisi server untuk setiap fragmen, sebagai metadata khusus Kinesis Video Streams.

- **API Konsumen** — Konsumen dapat menggunakan API berikut untuk mendapatkan data dari stream:
  - **GetMedia**- Saat menggunakan API ini, konsumen harus mengidentifikasi fragmen awal. API kemudian mengembalikan fragmen dalam urutan penambahannya ke aliran (dalam urutan penambahannya dengan urutan penambahannya dengan nomor fragmen). Data media dalam fragmen dikemas ke dalam format terstruktur seperti [Matroska](#) (MKV). Untuk informasi selengkapnya, lihat [GetMedia](#).

#### Note

GetMedia tahu di mana fragmennya (diarsipkan di penyimpanan data atau tersedia secara real time). Misalnya, jika GetMedia menentukan bahwa fragmen awal diarsipkan, ia mulai mengembalikan fragmen dari penyimpanan data. Ketika harus mengembalikan fragmen baru yang belum diarsipkan, GetMedia beralih ke pembacaan fragmen dari buffer aliran dalam memori.

Ini adalah contoh dari konsumen berkelanjutan, yang memproses fragmen dalam urutan bahwa mereka dicerna oleh sungai.

GetMedia memungkinkan aplikasi pemrosesan video gagal atau ketinggalan, dan kemudian catch tanpa upaya tambahan. Dengan menggunakan GetMedia, aplikasi dapat memproses data yang diarsipkan di penyimpanan data, dan saat aplikasi menyusul, GetMedia terus memberi makan data media secara real time saat tiba.

- **GetMediaFromFragmentList**(dan **ListFragments**) - Aplikasi pemrosesan Batch dianggap konsumen offline. Konsumen offline dapat memilih untuk secara eksplisit mengambil fragmen media atau rentang video tertentu dengan menggabungkan dan API. **ListFragments** **GetMediaFromFragmentList** **ListFragments** dan **GetMediaFromFragmentList**

memungkinkan aplikasi untuk mengidentifikasi segmen video untuk rentang waktu atau rentang fragmen tertentu, dan kemudian mengambil fragmen tersebut baik secara berurutan atau parallel untuk diproses. Pendekatan ini cocok untuk suite MapReduce aplikasi, yang harus dengan cepat memproses sejumlah besar data secara parallel.

Misalnya, anggaplah konsumen ingin memproses fragmen video satu hari. Konsumen akan melakukan hal berikut:

1. Dapatkan daftar fragmen dengan memanggil `ListFragments` API dan menentukan rentang waktu untuk memilih koleksi fragmen yang diinginkan.

API mengembalikan metadata dari semua fragmen dalam rentang waktu yang ditentukan. Metadata menyediakan informasi seperti nomor fragmen, stempel waktu sisi produser dan sisi server, dan sebagainya.

2. Ambil daftar metadata fragmen dan mengambil fragmen, dalam urutan apapun. Misalnya, untuk memproses semua fragmen untuk hari itu, konsumen dapat memilih untuk membagi daftar menjadi subdaftar dan meminta pekerja (misalnya, beberapa instans Amazon EC2) mengambil fragmen secara parallel menggunakan `GetMediaFromFragmentList`, dan memprosesnya secara parallel.

Diagram berikut menunjukkan aliran data untuk fragmen dan potongan selama panggilan API ini.

Ketika produser mengirim `PutMedia` permintaan, ia mengirimkan metadata media dalam payload, dan kemudian mengirimkan urutan fragmen data media. Setelah menerima data, Kinesis Video Streams menyimpan data media yang masuk sebagai potongan Kinesis Video Streams. Setiap potongan terdiri atas hal berikut:

- Salinan metadata media
- Sebuah fragmen
- Metadata khusus Kinesis Video Streams-spesifik; misalnya, nomor fragmen dan stempel waktu sisi server dan sisi produser

Saat konsumen meminta metadata media, Kinesis Video Streams mengembalikan aliran potongan, dimulai dengan nomor fragmen yang Anda tentukan dalam permintaan.

Jika Anda mengaktifkan persistensi data untuk streaming, setelah menerima fragmen pada aliran, Kinesis Video Streams juga menyimpan salinan fragmen ke penyimpanan data.

## Pustaka produser

Setelah Anda membuat aliran video Kinesis, Anda dapat mulai mengirim data ke streaming. Dalam kode aplikasi Anda, Anda dapat menggunakan pustaka ini untuk mengekstrak data dari sumber media Anda dan mengunggah ke aliran video Kinesis Anda. Untuk informasi selengkapnya tentang pustaka produser yang tersedia, lihat [Perpustakaan Produser Aliran Video Kinesis](#).

## Pemutaran Kinesis Video Streams

Anda dapat melihat aliran video Kinesis menggunakan metode berikut:

- **GetMedia**— Anda dapat menggunakan GetMedia API untuk membangun aplikasi Anda sendiri untuk memproses Kinesis Video Streams. GetMedia adalah API real-time dengan latensi rendah. Untuk membuat pemain yang menggunakan GetMedia, Anda harus membangunnya sendiri. Untuk informasi tentang cara mengembangkan aplikasi yang menampilkan aliran video Kinesis menggunakan GetMedia, lihat [Perpustakaan Parser](#)
- **HLS** — [HTTP Live Streaming \(HLS\)](#) adalah protokol komunikasi streaming media berbasis HTTP standar industri. Anda dapat menggunakan HLS untuk melihat aliran video Kinesis, baik untuk pemutaran langsung atau untuk melihat video yang diarsipkan.

Anda dapat menggunakan HLS untuk pemutaran langsung. Latensi biasanya antara 3-5 detik, tetapi bisa antara 1-10 detik, tergantung pada kasus penggunaan, pemain, dan kondisi jaringan. Anda dapat menggunakan pemutar pihak ketiga (seperti [Video.js](#) atau [Google Shaka Player](#)) untuk menampilkan aliran video dengan menyediakan URL sesi streaming HLS, baik secara terprogram maupun manual. Anda juga dapat memutar video dengan memasukkan URL sesi streaming HLS di bilah Lokasi browser [Apple Safari](#) atau [Microsoft Edge](#).

- **MPEG-DASH** — [Dynamic Adaptive Streaming over HTTP \(DASH\)](#), juga dikenal sebagai MPEG-DASH, adalah protokol streaming bitrate adaptif yang memungkinkan streaming konten media berkualitas tinggi melalui internet yang dikirim dari server web HTTP konvensional.

Anda dapat menggunakan MPEG-DASH untuk pemutaran langsung. Latensi biasanya antara 3-5 detik, tetapi bisa antara 1-10 detik, tergantung pada kasus penggunaan, pemain, dan kondisi jaringan. Anda dapat menggunakan pemutar pihak ketiga (seperti [dash.js](#) atau [Google Shaka Player](#)) untuk menampilkan aliran video dengan menyediakan URL sesi streaming MPEG-DASH, baik secara terprogram maupun manual.

- **GetClip**— Anda dapat menggunakan **GetClip** API untuk mengunduh klip (dalam file MP4) yang berisi media yang diarsipkan dan sesuai permintaan dari aliran video yang ditentukan selama rentang waktu yang ditentukan. Untuk informasi selengkapnya, lihat Referensi [GetClip](#) API.

## Topik

- [Persyaratan trek pemutaran video](#)
- [Pemutaran video dengan HLS](#)
- [Pemutaran Video dengan MPEG-DASH](#)

## Persyaratan trek pemutaran video

Amazon Kinesis Video Streams mendukung media yang dikodekan dalam berbagai format. Jika aliran video Kinesis Anda menggunakan format yang tidak didukung oleh salah satu dari empat API yang tercantum di bawah ini, gunakan [GetMedia](#) atau [GetMediaForFragmentList](#), karena tidak memiliki batasan jenis trek.

## Topik

- [GetClip persyaratan](#)
- [Persyaratan StreamingSession URL GetDash](#)
- [Persyaratan StreamingSession URL GetHLS](#)
- [GetImages persyaratan](#)

## GetClip persyaratan

Untuk informasi selengkapnya tentang API ini, lihat [GetClip](#).

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.264 video	V_MPEG/ISO/AVC	T/A	T/A
H.264 video	V_MPEG/ISO/AVC	Audio AAC	A_AAC
H.264 video	V_MPEG/ISO/AVC	Audio G.711 (hanya A-Law)	A_MS/ACM
H.265 video	V_MPEGH/ISO/HEVC	T/A	T/A

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

## Persyaratan StreamingSession URL GetDash

Untuk informasi selengkapnya tentang API ini, lihat [GetDASHStreamingSessionURL](#).

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.264 video	V_MPEG/ISO/AVC	T/A	T/A
H.264 video	V_MPEG/ISO/AVC	Audio AAC	A_AAC
H.264 video	V_MPEG/ISO/AVC	Audio G.711 (hanya A-Law)	A_MS/ACM
H.264 video	V_MPEG/ISO/AVC	Audio G.711 (hanya U-Law)	A_MS/ACM
Audio AAC	A_AAC	T/A	T/A
H.265 video	V_MPEGH/ISO/HEVC	T/A	T/A
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

## Persyaratan StreamingSession URL GetHLS

Untuk informasi selengkapnya tentang API ini, lihat [GetHLSStreamingSessionURL](#).

### HLS Mp4

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.264 video	V_MPEG/ISO/AVC	T/A	T/A
H.264 video	V_MPEG/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	T/A	T/A

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.265 video	V_MPEGH/ISO/HEVC	T/A	T/A
H.265 video	V_MPEGH/ISO/HEVC	Audio AAC	A_AAC

## HLS TS

Track 1 deskripsi	Lacak 1 ID codec	Deskripsi trek 2	Lacak 2 ID codec
H.264 video	V_MPEG/ISO/AVC	T/A	T/A
H.264 video	V_MPEG/ISO/AVC	Audio AAC	A_AAC
Audio AAC	A_AAC	T/A	T/A

## GetImages persyaratan

Untuk informasi selengkapnya tentang API ini, lihat [GetImages](#).

### Note

GetImagesMedia harus berisi trek video di trek 1.

## Pemutaran video dengan HLS

[HTTP Live Streaming \(HLS\)](#) adalah protokol komunikasi streaming media berbasis HTTP standar industri. Anda dapat menggunakan HLS untuk melihat aliran video Kinesis, baik untuk pemutaran langsung atau untuk melihat video yang diarsipkan.

Anda dapat menggunakan HLS untuk pemutaran langsung. Latensi biasanya antara 3-5 detik, tetapi bisa antara 1-10 detik, tergantung pada kasus penggunaan, pemain, dan kondisi jaringan. Anda dapat menggunakan pemutar pihak ketiga (seperti [Video.js](#) atau [Google Shaka Player](#)) untuk menampilkan aliran video dengan menyediakan URL sesi streaming HLS, baik secara terprogram maupun manual. Anda juga dapat memutar video dengan memasukkan URL sesi streaming HLS di bilah Lokasi browser [Apple Safari](#) atau [Microsoft Edge](#).



[Untuk melihat aliran video Kinesis menggunakan HLS, pertama buat sesi streaming menggunakan URL GetHLS. StreamingSession](#) Tindakan ini mengembalikan URL (berisi token sesi) untuk mengakses sesi HLS. Anda kemudian dapat menggunakan URL di pemutar media atau aplikasi mandiri untuk menampilkan aliran.

**⚠ Important**

Tidak semua media yang dikirim ke Kinesis Video Streams dapat diputar ulang melalui HLS. Lihat [the section called “GetHLSStreamingSessionURL”](#) untuk persyaratan pengunggahan khusus.

## Topik

- [Gunakan AWS CLI untuk mengambil URL sesi streaming HLS](#)
- [Contoh: Gunakan HLS dalam HTML dan JavaScript](#)
- [Memecahkan masalah HLS](#)

## Gunakan AWS CLI untuk mengambil URL sesi streaming HLS

Ikuti prosedur di bawah ini AWS CLI untuk menggunakan URL sesi streaming HLS untuk aliran video Kinesis.

Untuk petunjuk penginstalan, lihat [Panduan AWS Command Line Interface Pengguna](#). Setelah instalasi, [konfigurasi AWS CLI](#) dengan kredensial dan wilayah.

Atau, buka AWS CloudShell terminal, yang telah AWS CLI diinstal dan dikonfigurasi. Lihat [Panduan AWS CloudShell Pengguna](#) untuk informasi selengkapnya.

Ambil titik akhir URL HLS untuk aliran video Kinesis Anda.

1. Ketik berikut ini ke terminal:

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

Anda akan menerima respons yang terlihat seperti ini:

```
{
```

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

2. Buat permintaan URL sesi streaming HLS ke titik akhir yang dikembalikan.

### Live

Untuk pemutaran langsung, daftar putar media HLS terus diperbarui dengan media terbaru saat tersedia. Saat Anda memainkan jenis sesi ini di pemutar media, antarmuka pengguna biasanya menampilkan pemberitahuan "langsung", tanpa kontrol scrubber untuk memilih posisi di jendela pemutaran yang akan ditampilkan.

Pastikan Anda mengunggah media ke aliran ini saat menjalankan perintah ini.

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

### Live replay

Untuk pemutaran ulang langsung, pemutaran dimulai dari waktu mulai yang ditentukan. Daftar putar media HLS juga terus diperbarui dengan media terbaru saat tersedia. Sesi akan terus menyertakan media yang baru dicerna sampai sesi berakhir, atau sampai waktu akhir yang ditentukan, mana yang lebih dulu. Mode ini berguna untuk dapat memulai pemutaran dari saat suatu peristiwa terdeteksi dan melanjutkan media streaming langsung yang belum tertelan pada saat pembuatan sesi.

Tentukan stempel waktu mulai.

Untuk contoh ini, kami menggunakan waktu Unix Epoch dalam format detik. Lihat bagian [Stempel Waktu](#) di Panduan AWS Command Line Interface Pengguna untuk informasi selengkapnya tentang pemformatan stempel waktu.

Lihat [UnixTime.org](https://unixtime.org) untuk alat konversi.

- 1708471800 sama dengan 20 Februari 2024 3:30:00 PM GMT- 08:00

Dalam contoh ini, kami tidak menentukan stempel waktu akhir, artinya sesi akan terus menyertakan media yang baru dicerna hingga sesi berakhir.

Panggil `GetHLSStreamingSessionURL` API dengan mode `LIVE_REPLAY` pemutaran dan Pilih [Fragmen HLS](#) yang ditentukan.

```
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

Untuk pemutaran sesuai permintaan, daftar putar media HLS berisi media yang ditentukan oleh pemilih fragmen HLS. Ketika jenis sesi ini diputar di pemutar media, antarmuka pengguna biasanya menampilkan kontrol scrubber untuk memilih posisi di jendela pemutaran untuk ditampilkan.

Untuk membuat URL untuk bagian aliran tertentu, pertama-tama tentukan stempel waktu awal dan akhir.

Untuk contoh ini, kami menggunakan waktu Unix Epoch dalam format detik. Lihat bagian [Stempel Waktu](#) di Panduan AWS Command Line Interface Pengguna untuk informasi selengkapnya tentang pemformatan stempel waktu.

Lihat [UnixTime.org](https://unixtime.org) untuk alat konversi.

- 1708471800 sama dengan 20 Februari 2024 3:30:00 PM GMT- 08:00
- 1708471860 sama dengan 20 Februari 2024 3:31:00 PM GMT- 08:00

Panggil `GetHLSStreamingSessionURL` API dengan mode `ON_DEMAND` pemutaran dan Pilih [Fragmen HLS](#) yang ditentukan.

```
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 \  

```



- [hls.js](#)

Lihat [kode contoh lengkap](#) dan [halaman web yang dihosting](#) di GitHub.

Kode berjalan melalui topik:

- [Impor AWS SDK JavaScript untuk browser](#)
- [Siapkan klien Kinesis Video Streams](#)
- [Ambil titik akhir untuk pemutaran HLS](#)
- [Siapkan klien media yang diarsipkan Kinesis Video Streams](#)
- [Ambil URL sesi streaming HLS](#)
- [Menampilkan aliran HLS di halaman web](#)

Impor AWS SDK JavaScript untuk browser

Di halaman web, sertakan tag skrip berikut untuk mengimpor AWS SDK untuk JavaScript v2 ke dalam proyek.

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

Untuk informasi selengkapnya, lihat [AWS SDK untuk JavaScript](#) dokumentasi.

Siapkan klien Kinesis Video Streams

Untuk mengakses streaming video dengan HLS, pertama-tama buat dan konfigurasi klien Kinesis Video Streams. Lihat [Menyetel Kredensial di Browser Web](#) untuk metode autentikasi lainnya.

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

Aplikasi mengambil nilai-nilai yang diperlukan dari kotak input pada halaman HTML.

## Ambil titik akhir untuk pemutaran HLS

Gunakan klien Kinesis Video Streams untuk [the section called “GetDataEndpoint”](#) memanggil API guna mengambil titik akhir.

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

Kode ini menyimpan titik akhir dalam `hlsDataEndpoint` variabel.

## Siapkan klien media yang diarsipkan Kinesis Video Streams

Dalam konfigurasi klien untuk klien media yang diarsipkan Kinesis Video Streams, tentukan titik akhir yang Anda peroleh pada langkah sebelumnya.

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

## Ambil URL sesi streaming HLS

Gunakan klien media yang diarsipkan Kinesis Video Streams untuk [the section called “GetHLSStreamingSessionURL”](#) memanggil API guna mengambil URL pemutaran HLS.

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

## Menampilkan aliran HLS di halaman web

Saat Anda memiliki URL sesi streaming HLS, berikan ke pemutar video. Metode untuk memberikan URL ke pemutar video khusus untuk pemutar yang digunakan.

### Video.js

Lakukan hal berikut untuk mengimpor [Video.js](#) dan kelas CSS-nya ke dalam skrip browser kami:

```
<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>
```

Buat elemen video HTML untuk menampilkan video:

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

Mengatur URL HLS sebagai sumber elemen video HTML:

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

### Shaka

Lakukan hal berikut untuk mengimpor [pemutar Google Shaka](#) ke skrip browser kami:

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

Buat elemen video HTML untuk menampilkan video:

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

Buat pemutar Shaka yang menentukan elemen video dan panggil metode pemuatan:

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

## hls.js

Lakukan hal berikut untuk mengimpor [hls.js](#) ke skrip browser kami:

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

Buat elemen video HTML untuk menampilkan video:

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

Buat pemutar hls.js, berikan URL HLS, dan beri tahu untuk bermain:

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

## Memecahkan masalah HLS

Bagian ini menjelaskan masalah yang mungkin Anda temui saat menggunakan HTTP Live Streaming (HLS) dengan Kinesis Video Streams.

### Masalah

- [Mengambil URL sesi streaming HLS berhasil, tetapi pemutaran gagal di pemutar video](#)
- [Latensi terlalu tinggi antara produser dan pemain](#)

Mengambil URL sesi streaming HLS berhasil, tetapi pemutaran gagal di pemutar video

Situasi ini terjadi ketika Anda berhasil mengambil URL sesi streaming HLS menggunakan `getHLSStreamingSessionURL`, tetapi video gagal diputar kembali ketika URL diberikan ke pemutar video.



Untuk mengatasi masalah ini, coba yang berikut ini:

- Tentukan apakah aliran video diputar kembali di konsol Kinesis Video Streams. Pertimbangkan kesalahan apa pun yang ditampilkan konsol.
- Jika durasi fragmen kurang dari satu detik, tingkatkan menjadi satu detik. Jika durasi fragmen terlalu pendek, layanan mungkin membatasi pemutar karena terlalu sering membuat permintaan untuk fragmen video.
- Verifikasi bahwa setiap URL sesi streaming HLS hanya digunakan oleh satu pemain. Jika lebih dari satu pemain menggunakan URL sesi streaming HLS tunggal, layanan mungkin menerima terlalu banyak permintaan dan membatasi mereka.
- Verifikasi bahwa pemain Anda mendukung semua opsi yang Anda tentukan untuk sesi streaming HLS. Coba kombinasi nilai yang berbeda untuk parameter berikut:
  - `ContainerFormat`
  - `PlaybackMode`
  - `FragmentSelectorType`
  - `DiscontinuityMode`
  - `MaxMediaPlaylistFragmentResults`

Beberapa pemutar media (seperti HTML5 dan pemutar seluler) biasanya hanya mendukung HLS dengan format wadah fMP4. Pemutar media lain (seperti Flash dan pemutar khusus) mungkin hanya mendukung HLS dengan format wadah MPEG TS. Kami merekomendasikan bereksperimen dengan `ContainerFormat` parameter untuk memulai pemecahan masalah.

- Verifikasi bahwa setiap fragmen memiliki jumlah trek yang konsisten. Verifikasi bahwa fragmen dalam aliran tidak berubah antara memiliki trek audio dan video dan hanya trek video. Juga verifikasi bahwa pengaturan encoder (resolusi dan frame rate) tidak berubah di antara fragmen di setiap trek.

### Latensi terlalu tinggi antara produser dan pemain

Situasi ini terjadi ketika latensi terlalu tinggi dari saat video ditangkap hingga saat diputar di pemutar video.

Video diputar kembali melalui HLS berdasarkan per-fragmen. Oleh karena itu, latensi tidak boleh kurang dari durasi fragmen. Latensi juga mencakup waktu yang dibutuhkan untuk buffering dan transfer data. Jika solusi Anda memerlukan latensi kurang dari satu detik, pertimbangkan untuk menggunakan `GetMedia` API sebagai gantinya.

Anda dapat menyesuaikan parameter berikut untuk mengurangi latensi keseluruhan, tetapi menyesuaikan parameter ini juga dapat mengurangi kualitas video atau meningkatkan laju rebuffering.

- **Durasi fragmen** — Durasi fragmen adalah jumlah video antar divisi dalam aliran yang dikendalikan oleh frekuensi keyframe yang dihasilkan oleh encoder video. Nilai yang disarankan adalah satu detik. Memiliki durasi fragmen yang lebih pendek berarti lebih sedikit waktu yang dihabiskan untuk menunggu fragmen selesai sebelum mengirimkan data video ke layanan. Fragmen yang lebih pendek juga lebih cepat untuk diproses oleh layanan. Namun, jika durasi fragmen terlalu pendek, probabilitas meningkat bahwa pemain akan kehabisan konten dan harus berhenti dan menyangga konten. Jika durasi fragmen kurang dari 500 milidetik, produsen mungkin membuat terlalu banyak permintaan, menyebabkan layanan mencekatnya.
- **Bitrate** — Streaming video dengan bitrate lebih rendah membutuhkan waktu lebih sedikit untuk membaca, menulis, dan mengirimkan. Namun, streaming video dengan bitrate lebih rendah biasanya memiliki kualitas video yang lebih rendah.
- **Jumlah fragmen dalam daftar putar media** — Pemutar yang sensitif terhadap latensi hanya boleh memuat fragmen terbaru dalam daftar putar media. Sebagian besar pemain mulai dari fragmen paling awal sebagai gantinya. Dengan mengurangi jumlah fragmen dalam daftar putar, Anda mengurangi pemisahan waktu antara fragmen sebelumnya dan baru. Dengan ukuran playlist yang lebih kecil, fragmen dapat dilewati selama pemutaran jika ada penundaan dalam menambahkan fragmen baru ke daftar putar, atau jika ada penundaan pemain mendapatkan daftar putar yang diperbarui. Sebaiknya gunakan 3-5 fragmen, dan gunakan pemutar yang dikonfigurasi untuk memuat hanya fragmen terbaru dari daftar putar.
- **Ukuran buffer pemutar** — Sebagian besar pemutar video memiliki durasi buffer minimum yang dapat dikonfigurasi, biasanya dengan default 10 detik. Untuk latensi terendah, Anda dapat mengatur nilai ini menjadi 0 detik. Namun, melakukannya berarti pemain menolak jika ada penundaan menghasilkan fragmen karena pemain tidak akan memiliki buffer untuk menyerap penundaan.
- **Player “catch up”** - Pemutar video biasanya tidak secara otomatis menangkap pemutaran ke bagian depan buffer video jika buffer terisi, seperti ketika fragmen yang tertunda menyebabkan backlog fragmen diputar. Pemain khusus dapat menghindari hal ini dengan menjatuhkan bingkai, atau meningkatkan kecepatan pemutaran (misalnya, menjadi 1.1x) untuk mengejar bagian depan buffer. Hal ini menyebabkan pemutaran menjadi berombak atau peningkatan kecepatan saat pemain mengejar, dan rebuffering mungkin lebih sering karena ukuran buffer tetap pendek.

## Pemutaran Video dengan MPEG-DASH

[Untuk melihat aliran video Kinesis menggunakan MPEG-DASH, pertama-tama Anda membuat sesi streaming menggunakan URL GetDash. StreamingSession](#) Tindakan ini mengembalikan URL (berisi token sesi) untuk mengakses sesi MPEG-DASH. Anda kemudian dapat menggunakan URL di pemutar media atau aplikasi mandiri untuk menampilkan aliran.

Aliran video Amazon Kinesis memiliki persyaratan berikut untuk menyediakan video melalui MPEG-DASH:

- Untuk persyaratan trek pemutaran video streaming, lihat [the section called “URL GetDash StreamingSession”](#).
- Retensi data harus lebih besar dari 0.
- Trek video dari setiap fragmen harus berisi data pribadi codec dalam Advanced Video Coding (AVC) untuk format H.264 dan HEVC untuk format H.265. Untuk informasi lebih lanjut, lihat [spesifikasi MPEG-4 ISO/IEC 14496-15](#). Untuk informasi tentang mengadaptasi data aliran ke format tertentu, lihat [Bendera Adaptasi NAL](#).
- [Track audio \(jika ada\) dari setiap fragmen harus berisi data pribadi codec dalam format AAC \(spesifikasi AAC ISO/IEC 13818-7\) atau format MS Wave.](#)

### Contoh: Menggunakan MPEG-DASH dalam HTML dan JavaScript

Contoh berikut menunjukkan cara mengambil sesi streaming MPEG-DASH untuk aliran video Kinesis dan memutarinya kembali di halaman web. Contoh menunjukkan cara memutar video di pemain berikut:

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

### Topik

- [Mengatur Klien Kinesis Video Streams untuk Pemutaran MPEG-DASH](#)
- [Ambil Titik Akhir Konten yang Diarsipkan Kinesis Video Streams untuk Pemutaran MPEG-DASH](#)
- [Ambil URL Sesi Streaming MPEG-DASH](#)
- [Tampilkan Video Streaming dengan Pemutaran MPEG-DASH](#)
- [Contoh Selesai](#)

## Mengatur Klien Kinesis Video Streams untuk Pemutaran MPEG-DASH

Untuk mengakses video streaming dengan MPEG-DASH, pertama-tama buat dan konfigurasi klien Kinesis Video Streams (untuk mengambil titik akhir layanan) dan klien media yang diarsipkan (untuk mengambil sesi streaming MPEG-DASH). Aplikasi mengambil nilai-nilai yang diperlukan dari kotak input pada halaman 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);
```

### Ambil Titik Akhir Konten yang Diarsipkan Kinesis Video Streams untuk Pemutaran MPEG-DASH

Setelah klien dimulai, ambil titik akhir konten yang diarsipkan Kinesis Video Streams sehingga Anda dapat mengambil URL sesi streaming MPEG-DASH sebagai berikut:

```
// 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);
```

### Ambil URL Sesi Streaming MPEG-DASH

Saat Anda memiliki titik akhir konten yang diarsipkan, panggil API [StreamingSessionURL GetDash untuk mengambil URL](#) sesi streaming MPEG-DASH sebagai berikut:

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

## Tampilkan Video Streaming dengan Pemutaran MPEG-DASH

Saat Anda memiliki URL sesi streaming MPEG-DASH, berikan ke pemutar video. Metode untuk memberikan URL ke pemutar video khusus untuk pemutar yang Anda gunakan.

Contoh kode berikut menunjukkan cara memberikan URL sesi streaming ke pemutar [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');
}
```

Contoh kode berikut menunjukkan cara menyediakan URL sesi streaming ke pemutar [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');
}
```

Contoh Selesai

Anda dapat [mengunduh atau melihat kode contoh lengkap](#) pada GitHub.

# Menggunakan metadata streaming dengan Kinesis Video Streams

Anda dapat menggunakan Amazon Kinesis Video Streams Producer SDK untuk menyematkan metadata pada tingkat fragmen individual dalam aliran video Kinesis. Metadata dalam Kinesis Video Streams adalah pasangan kunci-nilai yang bisa berubah. Anda dapat menggunakannya untuk mendeskripsikan konten fragmen, menyematkan pembacaan sensor terkait yang harus ditransfer bersama dengan fragmen yang sebenarnya, atau memenuhi kebutuhan khusus lainnya. Metadata tersedia sebagai bagian dari operasi [the section called “GetMedia”](#) atau [the section called “GetMediaForFragmentList”](#) API. Ini disimpan bersama dengan fragmen untuk seluruh durasi periode retensi aliran. Aplikasi yang Anda konsumsi dapat membaca, memproses, dan bereaksi berdasarkan metadata menggunakan. [Perpustakaan Parser aliran video](#)

Ada dua mode di mana metadata dapat disematkan dengan fragmen dalam aliran:

- Non-persisten — Anda dapat membubuhkan metadata pada satu kali, atau basis ad hoc ke fragmen dalam aliran, berdasarkan kriteria khusus bisnis yang telah terjadi. Contohnya adalah kamera pintar yang mendeteksi gerakan dan menambahkan metadata ke fragmen terkait yang berisi gerakan sebelum mengirim fragmen ke aliran video Kinesis. Anda dapat menerapkan metadata ke fragmen dalam format berikut: `Motion = true`
- Persistent — Anda dapat membubuhkan metadata ke fragmen berturut-turut dan berurutan dalam aliran berdasarkan kebutuhan yang berkelanjutan. Contohnya adalah kamera pintar yang mengirimkan koordinat lintang dan bujur saat ini yang terkait dengan semua fragmen yang dikirimkannya ke aliran video Kinesis. Anda dapat menerapkan metadata ke semua fragmen dalam format berikut: `Lat = 47.608013N , Long = -122.335167W`

Anda dapat menempelkan metadata di kedua mode ini ke fragmen yang sama secara bersamaan, berdasarkan kebutuhan aplikasi Anda. Metadata yang disematkan mungkin mencakup objek yang terdeteksi, aktivitas yang dilacak, koordinat GPS, atau data kustom lainnya yang ingin Anda kaitkan dengan fragmen dalam aliran. Metadata dikodekan sebagai pasangan string kunci-nilai.

## Topik

- [Menambahkan metadata ke aliran video Kinesis](#)
- [Mengonsumsi metadata yang disematkan dalam aliran video Kinesis](#)
- [Batas metadata streaming](#)

## Menambahkan metadata ke aliran video Kinesis

Metadata yang Anda tambahkan ke aliran video Kinesis dimodelkan sebagai tag MKV, yang diimplementasikan sebagai pasangan nilai kunci.

Metadata dapat bersifat sementara, seperti untuk menandai peristiwa dalam aliran, atau persisten, seperti untuk mengidentifikasi fragmen di mana peristiwa tertentu sedang berlangsung. Item metadata persisten tetap ada, dan diterapkan ke setiap fragmen berturut-turut, hingga dibatalkan.

### Note

Item metadata yang ditambahkan menggunakan API penandaan tingkat aliran [Pustaka Produser](#) berbeda dari API penandaan tingkat aliran yang diimplementasikan dengan, dan. [the section called "TagStream"](#) [the section called "UntagStream"](#) [the section called "ListTagsForStream"](#)

## Streaming metadata API

Anda dapat menggunakan operasi berikut di Producer SDK untuk mengimplementasikan metadata streaming.

### FOTO

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

### SDK Produser C ++

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

### SDK Produsen Java

Anda dapat menggunakan Java Producer SDK, untuk menambahkan metadata ke penggunaan: `MediaSource MediaSourceSink.onCodecPrivateData`



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

## Metadata persisten dan nonpersisten

Untuk metadata nonpersisten, Anda dapat menambahkan beberapa item metadata dengan nama yang sama. Producer SDK mengumpulkan item metadata dalam antrian metadata hingga ditambahkan ke fragmen berikutnya. Antrian metadata dihapus saat item metadata diterapkan ke aliran. Untuk mengulang metadata, panggil `putKinesisVideoFragmentMetadata` atau `putFragmentMetadata` lagi.

Untuk metadata persisten, Producer SDK mengumpulkan item metadata dalam antrian metadata dengan cara yang sama seperti untuk metadata nonpersisten. Namun, item metadata tidak dihapus dari antrian saat ditambahkan ke fragmen berikutnya.

Memanggil `putKinesisVideoFragmentMetadata` atau `putFragmentMetadata` dengan `persistent` set untuk `true` memiliki perilaku berikut:

- Memanggil API menempatkan item metadata dalam antrian. Metadata ditambahkan sebagai tag MKV ke setiap fragmen saat item berada dalam antrian.
- Memanggil API dengan nama yang sama dan nilai yang berbeda sebagai item metadata yang ditambahkan sebelumnya akan menimpa item tersebut.
- Memanggil API dengan nilai kosong akan menghapus (membatalkan) item metadata dari antrian metadata.

## Mengonsumsi metadata yang disematkan dalam aliran video Kinesis

Untuk menggunakan metadata dalam aliran video Kinesis, gunakan implementasi:

### 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");
        }
    }
}

```

Antarmuka ini ditemukan di [FragmentMetadataVisitor](#) kelas di [Perpustakaan Parser aliran video](#).

FragmentMetadataVisitorKelas berisi implementasiMkvTagProcessor:

```

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();
    }
}

```

KinesisVideoRendererExampleKelas berisi contoh bagaimana menggunakanBasicMkvTagProcessor. Dalam contoh berikut, a BasicMkvTagProcessor ditambahkan ke MediaProcessingArguments aplikasi:

```

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

```

BasicMkvTagProcessor.processMetode ini disebut ketika metadata fragmen tiba. Anda dapat mengambil metadata yang terakumulasi dengan GetTags Untuk mengambil satu item metadata, panggil dulu clear untuk menghapus metadata yang dikumpulkan, lalu ambil kembali item metadata.

## Batas metadata streaming

Lihat [the section called “Kuota metadata fragmen”](#) untuk informasi selengkapnya tentang batasan yang berlaku untuk menambahkan metadata streaming ke aliran video Kinesis

## Model data Kinesis Video Streams

[Perpustakaan Parser](#) Mengirim [Pustaka Produser](#) dan menerima data video dalam format yang mendukung penyematan informasi bersama data video. Format ini didasarkan pada spesifikasi Matroska (MKV).

[Format MKV](#) adalah spesifikasi terbuka untuk data media. Semua pustaka dan contoh kode dalam Panduan Pengembang Amazon Kinesis Video Streams mengirim atau menerima data dalam format MKV.

[Perpustakaan Produser Aliran Video Kinesis](#) Menggunakan `StreamDefinition` dan `Frame` jenis untuk menghasilkan header aliran MKV, header bingkai, dan data bingkai.

Untuk informasi tentang spesifikasi MKV lengkap, lihat Spesifikasi [Matroska](#).

Bagian berikut menjelaskan komponen data berformat MKV yang dihasilkan oleh [Pustaka Produser C++](#)

### Topik

- [Elemen header aliran](#)
- [Alirkan data trek](#)
- [Elemen header bingkai](#)
- [Data bingkai MKV](#)

## Elemen header aliran

Elemen header MKV berikut digunakan oleh `StreamDefinition` (didefinisikan dalam `StreamDefinition.h`).

Elemen	Deskripsi	Nilai-nilai khas
<code>stream_name</code>	Sesuai dengan nama aliran video Kinesis.	aliran saya

Elemen	Deskripsi	Nilai-nilai khas
retensi_periode	Durasi, dalam beberapa jam, data streaming dipertahankan oleh Kinesis Video Streams. 0 Tentukan aliran yang tidak menyimpan data.	24
tag	Pengumpulan nilai kunci data pengguna. Data ini ditampilkan di AWS Management Console dan dapat dibaca oleh aplikasi klien untuk memfilter atau mendapatkan informasi tentang aliran.	
kms_key_id	Jika ada, AWS KMS kunci yang ditentukan pengguna digunakan untuk mengenkripsi data pada aliran. Jika tidak ada, data dienkripsi oleh kunci yang disediakan Kinesis ( <code>aws/kinesis-video</code> ).	01234567-89ab-cdef-0123-456789ab
streaming_type	Saat ini, satu-satunya jenis streaming yang valid adalah <code>STREAMING_TYPE_REALTIME</code> .	<code>STREAMING_TYPE_REALTIME</code>
content_type	Jenis konten yang ditentukan pengguna. Untuk streaming data video untuk diputar di konsol, jenis konten harus <code>video/h264</code> .	Video/h264
max_latensi	Nilai ini saat ini tidak digunakan dan harus disetel ke 0.	0

Elemen	Deskripsi	Nilai-nilai khas
fragment_duration	Perkiraan berapa lama fragmen Anda seharusnya, yang digunakan untuk optimasi. Durasi fragmen aktual ditentukan oleh data streaming.	2
timecode_scale	Menunjukkan skala yang digunakan oleh stempel waktu bingkai. Defaultnya adalah 1 milidetik. Menentukan 0 juga menetapkan nilai default 1 milidetik. Nilai ini bisa antara 100 nanodetik dan 1 detik.  Untuk informasi lebih lanjut, lihat <a href="#">TimecodeScaled</a> di dokumentasi Matroska.	
key_frame_fragmentasi	Jika <code>true</code> , aliran memulai cluster baru saat keyframe diterima.	benar
frame_timecodes	Jika <code>true</code> , Kinesis Video Streams menggunakan nilai stempel waktu presentasi (pts) dan decoding time stamp (dts) dari frame yang diterima. Jika <code>false</code> , Kinesis Video Streams mencap frame saat diterima dengan nilai waktu yang dihasilkan sistem.	benar

Elemen	Deskripsi	Nilai-nilai khas
<code>absolute_fragment_time</code>	Jika <code>true</code> , kode waktu cluster ditafsirkan menggunakan waktu absolut (misalnya, dari jam sistem produsen). Jika <code>false</code> , kode waktu cluster ditafsirkan sebagai relatif terhadap waktu mulai aliran.	benar
<code>fragment_acks</code>	Jika <code>true</code> , tanda terima kasih (ACK) dikirim saat Kinesis Video Streams menerima data. ACK dapat diterima menggunakan <code>KinesisVideoStreamFragmentAck</code> atau <code>KinesisVideoStreamParseFragmentAck</code> callback.	benar
<code>restart_on_error</code>	Menunjukkan apakah aliran harus melanjutkan transmisi setelah kesalahan aliran dinaikkan.	benar
<code>nal_adaptation_flags</code>	Menunjukkan apakah adaptasi NAL (Network Abstraction Layer) atau data pribadi codec hadir dalam konten. Bendera yang valid termasuk <code>NAL_ADAPTATION_ANNEXB_NALS</code> dan <code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code> .	<code>NAL_ADAPTATION_ANNEXB_NALS</code>

Elemen	Deskripsi	Nilai-nilai khas
frame_rate	Perkiraan frame rate konten. Nilai ini digunakan untuk optimasi; frame rate aktual ditentukan oleh laju data yang masuk. Menentukan 0 menetapkan default dari. 24	24
avg_bandwidth_bps	Perkiraan bandwidth konten, dalam Mbps. Nilai ini digunakan untuk optimasi; tingkat aktual ditentukan oleh bandwidth data yang masuk. Misalnya, untuk aliran video resolusi 720 p yang berjalan pada 25 FPS, Anda dapat mengharapkan bandwidth rata-rata menjadi 5 Mbps.	5
buffer_duration	Durasi konten yang akan di-buffer pada produser. Jika latensi jaringan rendah, nilai ini dapat dikurangi. Jika latensi jaringan tinggi, meningkatkan nilai ini mencegah frame dijatuhkan sebelum dapat dikirim, karena alokasi gagal memasukkan bingkai ke buffer yang lebih kecil.	

Elemen	Deskripsi	Nilai-nilai khas
replay_duration	Jumlah waktu aliran data video “diputar ulang” jika koneksi terputus. Nilai ini bisa nol jika frame yang hilang karena kehilangan koneksi tidak menjadi perhatian. Nilainya dapat ditingkatkan jika aplikasi yang dikonsumsi dapat menghapus frame yang berlebihan. Nilai ini harus kurang dari durasi buffer, jika tidak durasi buffer digunakan.	
connection_staleness	Durasi koneksi dipertahankan ketika tidak ada data yang diterima.	
codec_id	Codec yang digunakan oleh konten. Untuk informasi lebih lanjut, lihat <a href="#">CodeCID dalam spesifikasi Matroska</a> .	V_MPEG2
track_name	Nama trek yang ditentukan pengguna.	my_track



Elemen	Deskripsi	Nilai-nilai khas
codecPrivateData	Data yang disediakan oleh encoder digunakan untuk memecahkan kode data frame, seperti lebar bingkai dan tinggi dalam piksel, yang dibutuhkan oleh banyak konsumen hilir. Di <a href="#">Perpustakaan Produser C++</a> , gMkvTrackVideoBits array MkvStatics.cpp termasuk lebar piksel dan tinggi untuk bingkai.	
codecPrivateDataUkuran	Ukuran data dalam codecPrivateData parameter.	
track_type	Jenis trek untuk aliran.	MKV_TRACK_INFO_TYPE_AUDIO atau MKV_TRACK_INFO_TYPE_VIDEO
segment_uuid	Uuid segmen yang ditentukan pengguna (16 byte).	
default_track_id	Nomor unik bukan nol untuk trek.	1

## Alirkan data trek

Elemen trek MKV berikut digunakan oleh `StreamDefinition` (didefinisikan dalam `StreamDefinition.h`).

Elemen	Deskripsi	Nilai Khas
track_name	Nama trek yang ditentukan pengguna. Misalnya, "audio" untuk trek audio.	audio
codec_id	Codec id untuk trek. Misalnya, "A_AAC" untuk trek audio.	A_AAC
cpd	Data yang disediakan oleh encoder digunakan untuk memecahkan kode data frame. Data ini dapat mencakup lebar bingkai dan tinggi dalam piksel, yang dibutuhkan oleh banyak konsumen hilir. Di <a href="#">C++ Producer Library</a> , gMkvTrack VideoBits array MkvStatics di.cpp menyertakan lebar piksel dan tinggi untuk frame.	
cpd_size	Ukuran data dalam codecPrivateData parameter.	
track_type	Jenis trek. Misalnya, Anda dapat menggunakan nilai enum MKV_TRACK_INFO_TYPE_AUDIO untuk audio.	MKV_TRACK_INFO_TYPE_AUDIO

## Elemen header bingkai

Elemen header MKV berikut digunakan oleh Frame (didefinisikan dalam KinesisVideoPic paket, dimkvgen/Include.h):

- Indeks Bingkai: Nilai yang meningkat secara monoton.

- Bendera: Jenis bingkai. Nilai yang valid adalah sebagai berikut:
  - `FRAME_FLAGS_NONE`
  - `FRAME_FLAG_KEY_FRAME`: Jika `key_frame_fragmentation` diatur pada aliran, bingkai kunci memulai fragmen baru.
  - `FRAME_FLAG_DISCARDABLE_FRAME`: Memberitahu decoder bahwa ia dapat membuang frame ini jika decoding lambat.
  - `FRAME_FLAG_INVISIBLE_FRAME`: Durasi blok ini adalah 0.
- Decoding Timestamp: Stempel waktu kapan bingkai ini diterjemahkan. Jika frame sebelumnya bergantung pada frame ini untuk decoding, stempel waktu ini mungkin lebih awal dari frame sebelumnya. Nilai ini relatif terhadap awal fragmen.
- Stempel Waktu Presentasi: Stempel waktu kapan bingkai ini ditampilkan. Nilai ini relatif terhadap awal fragmen.
- Durasi: Durasi pemutaran frame.
- Ukuran: Ukuran data frame dalam byte

## Data bingkai MKV

Data di `frame.frameData` mungkin hanya berisi data media untuk frame, atau mungkin berisi informasi header bersarang lebih lanjut, tergantung pada skema pengkodean yang digunakan. Untuk ditampilkan di AWS Management Console, data harus dikodekan dalam codec [H.264](#), tetapi Kinesis Video Streams dapat menerima aliran data berseri waktu dalam format apa pun.

# Memulai dengan Amazon Kinesis Video Streams

Bagian ini menjelaskan cara melakukan tugas-tugas berikut di Amazon Kinesis Video Streams:

- Siapkan Akun AWS dan buat administrator, jika Anda belum melakukannya.
- Buat aliran video Kinesis.
- Kirim data ke aliran video Kinesis dari kamera Anda dan lihat media di konsol.

Jika Anda baru mengenal Amazon Kinesis Video Streams, sebaiknya Anda membaca terlebih dahulu. [Kinesis Video Streams: Cara kerjanya](#)

## Note

Mengikuti sampel Memulai tidak akan dikenakan biaya apa pun kepada Anda Akun AWS. Untuk biaya data di Wilayah Anda, lihat Harga [Amazon Kinesis Video Streams](#).

## Topik

- [Menyiapkan akun](#)
- [Buat aliran video Kinesis](#)
- [Kirim data ke aliran video Amazon Kinesis](#)
- [Konsumsi data media](#)

## Menyiapkan akun

Sebelum Anda menggunakan Amazon Kinesis Video Streams untuk pertama kalinya, selesaikan tugas-tugas berikut.

## Topik

- [Mendaftar untuk Akun AWS](#)
- [Membuat pengguna administratif](#)
- [Buat Akun AWS kunci](#)

## Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.
2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS mengirimkan Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

## Membuat pengguna administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In .

2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuk, lihat [Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root \(konsol\) Anda](#) di Panduan Pengguna IAM.

## Membuat pengguna administratif

### 1. Aktifkan Pusat Identitas IAM.

Untuk mendapatkan petunjuk, silakan lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center .

### 2. Di Pusat Identitas IAM, berikan akses administratif ke sebuah pengguna administratif.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM](#) di Panduan AWS IAM Identity Center Pengguna.

## Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal AWS akses](#) di Panduan AWS Sign-In Pengguna.

## Buat Akun AWS kunci

Anda akan memerlukan Akun AWS Kunci untuk mengakses Amazon Kinesis Video Streams secara terprogram.

Untuk membuat Akun AWS Kunci, lakukan hal berikut:

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Pengguna di bilah navigasi, dan pilih pengguna Administrator.
3. Pilih tab Security credentials, dan pilih Create access key.
4. Rekam ID kunci Akses. Pilih Tampilkan di bawah Kunci akses rahasia. Rekam kunci akses Rahasia.

## Buat aliran video Kinesis

Bagian ini menjelaskan cara membuat aliran video Kinesis.

Bagian ini berisi prosedur berikut:

- [the section called “Buat aliran video menggunakan konsol”](#)
- [the section called “Buat aliran video menggunakan AWS CLI”](#)

## Buat aliran video menggunakan konsol

1. Buka konsol di <https://console.aws.amazon.com/kinesisvideo/home>.
2. Pada halaman Streaming video, pilih Buat aliran video.
3. Pada halaman Create a new video stream, masukkan *YourStreamName* nama stream. Biarkan tombol konfigurasi Default dipilih.
4. Pilih Buat aliran video.
5. Setelah Amazon Kinesis Video Streams membuat streaming, tinjau detail YourStreamName di halaman.

## Buat aliran video menggunakan AWS CLI

1. Verifikasi bahwa Anda telah AWS CLI menginstal dan mengkonfigurasi. Untuk informasi lebih lanjut, lihat [AWS Command Line Interface](#) dokumentasi.
2. Jalankan Create-Stream perintah berikut di AWS CLI:

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

Responsnya akan terlihat mirip dengan yang berikut:

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

# Kirim data ke aliran video Amazon Kinesis

Bagian ini menjelaskan cara mengirim data media dari kamera ke aliran video Kinesis yang Anda buat di bagian sebelumnya. Bagian ini menggunakan [Pustaka Produser C++](#) sebagai [GStreamer](#) plugin.

Untuk mengirim media dari berbagai perangkat pada berbagai sistem operasi, tutorial ini menggunakan [perpustakaan](#) produser Kinesis Video Streams C++ dan GStreamer, kerangka media open-source yang menstandarisasi akses ke kamera dan sumber media lainnya.

## Topik

- [Membangun SDK dan sampel](#)
- [Jalankan sampel untuk mengunggah media ke Kinesis Video Streams](#)
- [Tinjau objek pengakuan](#)

## Membangun SDK dan sampel

Anda dapat membangun SDK dan sampel di komputer Anda atau di AWS Cloud9. Ikuti prosedur yang sesuai di bawah ini.

### Build on your computer

Gunakan instruksi dalam [file readme](#) untuk membangun pustaka produser dan aplikasi sampel.

Hal ini mencakup:

- Menginstal dependensi
- Mengkloning repositori
- Menggunakan CMake untuk menghasilkan makefile
- Membangun file biner menggunakan make

### Build in AWS Cloud9

Ikuti prosedur ini untuk mengunggah ke Kinesis Video AWS Cloud9 Streams di. Anda tidak perlu mengunduh apa pun ke komputer Anda.

1. Di AWS Management Console, terbuka [AWS Cloud9](#).




Pilih Buat lingkungan.

2. Pada layar Create environment, lengkapi yang berikut ini:
  - Nama - Ketik nama untuk lingkungan baru Anda.
  - Platform - Pilih Ubuntu Server 22.04 LTS.

Anda dapat meninggalkan bidang lain dengan pilihan default.

3. Ketika lingkungan telah dibuat, pilih Buka di kolom Cloud9 IDE.

Di area tengah bawah layar, Anda lihat. `Admin:~/environment $` Ini adalah terminal AWS Cloud9 (Amazon EC2).

 Note

Jika Anda secara tidak sengaja menutup terminal, pilih Jendela, Terminal Baru.

Jalankan perintah berikut di terminal untuk mengubah volume menjadi 20 GiB.

- a. Unduh skrip .

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

- b. Berikan izin eksekusi skrip.

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

- c. Jalankan penulisan.

```
./resize_volume.sh
```

4. Ambil informasi terbaru tentang semua perangkat lunak yang dapat Anda instal atau perbarui melalui Advanced Packaging Tool (APT).

Perintah ini tidak memperbarui perangkat lunak itu sendiri, tetapi memastikan sistem Anda tahu apa versi terbaru yang tersedia.

```
sudo apt-get update
```

5. Instal dependensi SDK produsen 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. Gunakan git untuk mengkloning SDK produsen C ++.

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

7. Siapkan direktori build.

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

8. Gunakan CMake untuk menghasilkan makefile.

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

Akhir dari output yang diharapkan terlihat seperti berikut:

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

9. Gunakan make untuk mengkompilasi SDK dan contoh aplikasi, serta membangun executable akhir.

```
make
```

Akhir dari output yang diharapkan terlihat seperti berikut:

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

```
[100%] Built target kvs_gstreamer_file_uploader_sample
```

10. Konfirmasikan file sampel telah dibuat. Daftar file di direktori saat ini:

```
ls
```

Konfirmasikan bahwa file-file berikut ada:

- `kvs_gstreamer_sample`
- `libgstkvssink.so`

## Jalankan sampel untuk mengunggah media ke Kinesis Video Streams

Aplikasi sampel tidak mendukung kredensial IMDS. Di terminal Anda, ekspor AWS kredensial untuk pengguna atau peran IAM Anda dan wilayah tempat aliran Anda berada.

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

Jika Anda menggunakan AWS kredensial sementara, ekspor juga token sesi Anda:

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

Unduh contoh video.mp4 untuk diunggah ke Kinesis Video Streams.

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

Jalankan perintah berikut dengan nama aliran yang Anda buat sebelumnya. Jika Anda belum membuat streaming, lihat [the section called “Buat aliran video Kinesis”](#).

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

Sample video from GStreamer

Gunakan perintah berikut untuk menghasilkan video menggunakan GStreamer.

Beri tahu GStreamer di mana menemukan plugin kvssink GStreamer. Di direktori build Anda, tentukan path ke folder yang berisi `libgstkvssink.so` file.

Dari direktori build Anda, jalankan perintah berikut:

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

Pipeline GStreamer ini menghasilkan streaming video uji langsung dengan pola pengujian standar yang berjalan pada 10 frame per detik dengan resolusi 640x480 piksel. Hamparan ditambahkan menampilkan waktu dan tanggal sistem saat ini. Video kemudian dikodekan ke dalam format H.264 dan keyframe dihasilkan paling banyak setiap 10 frame, menghasilkan durasi fragmen (juga dikenal sebagai ukuran sekelompok gambar (GoP)) 1 detik. kvssink mengambil aliran video yang dikodekan H.264, mengemasnya ke dalam format wadah Matroska (MKV), dan mengunggahnya ke aliran video Kinesis Anda.

Jalankan perintah berikut:

```
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"
```

Untuk menghentikan pipa GStreamer, pilih jendela terminal dan tekan CTRL+C.

#### Note

Untuk informasi lebih lanjut tentang menggunakan plugin GStreamer untuk melakukan streaming video dari aliran RTSP dari kamera, atau dari kamera USB, lihat. [Contoh: Plugin SDK GStreamer Produser Kinesis Video Streamer](#)

## Tinjau objek pengakuan

Selama pengunggahan, Kinesis Video Streams akan mengirim objek pengakuan kembali ke klien yang melakukan unggahan. Anda akan melihat ini dicetak dalam output perintah. Contohnya terlihat seperti berikut:

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

Jika pengakuan itu PERSISTED, itu EventType berarti Kinesis Video Streams telah lama menyimpan dan mengenkripsi potongan media ini untuk pengambilan, analisis, dan penyimpanan jangka panjang.

Untuk informasi lebih lanjut tentang ucapan terima kasih, lihat [the section called “PutMedia”](#)

## Konsumsi data media

Anda dapat menggunakan data media dengan melihatnya di konsol, atau dengan membuat aplikasi yang membaca data media dari aliran menggunakan Hypertext Live Streaming (HLS).

### Lihat media di konsol

Di tab browser lain, buka file AWS Management Console. Di Dasbor Kinesis Video Streams, pilih Streaming video.

Pilih nama aliran Anda dalam daftar aliran. Gunakan bilah pencarian, jika perlu.

Perluas bagian Pemutaran media. Jika video masih diunggah, itu akan ditampilkan. Jika unggahan telah selesai, pilih panah kiri ganda.

## Konsumsi data media menggunakan HLS

Anda dapat membuat aplikasi klien yang mengkonsumsi data dari aliran video Kinesis menggunakan HLS. Untuk informasi tentang membuat aplikasi yang mengkonsumsi data media menggunakan HLS, lihat [the section called “Pemutaran Video”](#)

# Agen Tepi Streaming Video Amazon Kinesis

Amazon Kinesis Video Streams menawarkan cara yang efisien dan hemat biaya untuk terhubung ke kamera IP di tempat pelanggan. Dengan Agen Edge Amazon Kinesis Video Streams, Anda dapat merekam dan menyimpan video secara lokal dari kamera dan mengalirkan video ke cloud pada jadwal yang ditentukan pelanggan untuk penyimpanan, pemutaran, dan pemrosesan analitis jangka panjang.

## Note

Untuk mengakses Agen Tepi Amazon Kinesis Video Streams, [lengkapi](#) formulir singkat ini.

Anda dapat mengunduh Agen Edge Amazon Kinesis Video Streams dan menerapkannya di perangkat komputasi edge lokal. Anda juga dapat dengan mudah menerapkannya di kontainer Docker yang berjalan di instans Amazon EC2. Setelah penerapan, Anda dapat menggunakan Amazon Kinesis Video Streams API untuk memperbarui perekaman video dan konfigurasi pengunggahan cloud. Fitur ini bekerja dengan kamera IP apa pun yang dapat melakukan streaming melalui protokol RTSP. Itu tidak memerlukan penerapan firmware tambahan ke kamera.

Kami menawarkan instalasi berikut untuk Agen Edge Amazon Kinesis Video Streams:

- Sebagai AWS IoT Greengrass V2 komponen: Anda dapat menginstal Agen Tepi Amazon Kinesis Video Streams AWS IoT Greengrass sebagai komponen AWS IoT Greengrass pada perangkat bersertifikat apa pun. Untuk mempelajari selengkapnya AWS IoT Greengrass, lihat [Panduan AWS IoT Greengrass Version 2 Pengembang](#).
- Aktif AWS Snowball Edge: Anda dapat menjalankan Agen Edge Amazon Kinesis Video Streams di perangkat Snowball Edge. Untuk mempelajari lebih lanjut, lihat [Panduan Pengembang AWS Snowball Edge](#).
- Pada AWS IoT penerapan asli: Anda dapat menginstal Agen Edge Amazon Kinesis Video Streams secara native pada instans komputasi apa pun. Edge SDK digunakan [AWS IoT Core](#) untuk mengelola edge melalui file. [the section called "Amazon Kinesis Video Streams"](#)

Untuk memulai dengan Agen Edge Amazon Kinesis Video Streams, lanjutkan dengan prosedur yang sesuai di bawah ini.

Topik

- [Operasi API Agen Edge Streams Amazon Kinesis Video Streams](#)
- [Memantau Agen Edge Streams Video Amazon Kinesis](#)
- [Jalankan Agen Edge Streams Video Amazon Kinesis dalam mode non-mode AWS IoT Greengrass](#)
- [Terapkan Agen Edge Streams Video Amazon Kinesis ke AWS IoT Greengrass](#)
- [FAQ Agen Edge Streams Video Kinesis Amazon](#)

## Operasi API Agen Edge Streams Amazon Kinesis Video Streams

Gunakan operasi API berikut untuk mengonfigurasi Agen Edge Amazon Kinesis Video Streams:

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

## Memantau Agen Edge Streams Video Amazon Kinesis

Untuk memantau Agen Edge Amazon Kinesis Video Streams, lihat. [the section called “Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch”](#)

## Jalankan Agen Edge Streams Video Amazon Kinesis dalam mode non-mode AWS IoT Greengrass

Ikuti langkah-langkah berikut untuk menjalankan AWS IoT Agen Edge Amazon Kinesis Video Streams dengan MQTT sebagai penerapan mandiri.

Topik

- [Langkah 1: Instal dependensi yang diperlukan pada perangkat](#)
- [Langkah 2: Buat Amazon Kinesis Video AWS Secrets Manager Streams dan sumber daya untuk URL RTSP kamera IP Anda](#)
- [Langkah 3: Buat kebijakan izin IAM](#)
- [Langkah 4: Buat peran IAM](#)
- [Langkah 5: Buat alias AWS IoT peran](#)

- [Langkah 6: Buat AWS IoT kebijakan](#)
- [Langkah 7: Buat AWS IoT sesuatu dan dapatkan kredensyal untuk AWS IoT Core](#)
- [Langkah 8: Bangun dan jalankan Agen Edge Amazon Kinesis Video Streams](#)
- [Langkah 9: \(Opsional\) Instal CloudWatch agen di perangkat](#)
- [Langkah 10: \(Opsional\) Jalankan Agen Edge Amazon Kinesis Video Streams sebagai proses asli](#)

## Langkah 1: Instal dependensi yang diperlukan pada perangkat

### Note

Untuk daftar sistem operasi yang didukung, lihat [the section called “Sistem operasi apa yang didukung Amazon Kinesis Video Streams Edge Agent?”](#).

### Instal dependensi pada perangkat

1. Untuk menjalankan Agen Edge Amazon Kinesis Video Streams, instal pustaka berikut yang sesuai di perangkat Anda:

#### Ubuntu

Jenis:

```
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

Jenis:

```
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
```

Instal log4cplus-2.1.0 dari sumbernya.

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/  
log4cplus-2.1.0.tar.gz  
tar -xvzf log4cplus-2.1.0.tar.gz  
cd log4cplus-2.1.0 && \  
mkdir build && \  
cd build && \  
cmake .. && \  
sudo make && \  
sudo make install
```

Instal apache-maven-3.9.2 dari sumbernya.

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

### Important

Jika Anda melihat layar yang memberi tahu Anda bahwa beberapa layanan perlu dimulai ulang, tekan Enter untuk memilih Ok.

Untuk informasi tambahan, lihat Panduan [Pengguna Amazon Corretto 11](#).

2. Instal AWS Command Line Interface. Lihat [Menginstal atau memperbarui versi terbaru AWS CLI prosedur](#) di Panduan AWS Command Line Interface Pengguna.

## Langkah 2: Buat Amazon Kinesis Video AWS Secrets Manager Streams dan sumber daya untuk URL RTSP kamera IP Anda

Ikuti prosedur ini untuk membuat aliran dan rahasia yang dibutuhkan. AWS Secrets Manager Lakukan langkah ini terlebih dahulu, karena Anda memerlukan ARN dari sumber daya yang dibuat dalam kebijakan.

## Buat Streaming Video Amazon Kinesis

Buat Amazon Kinesis Video Streams AWS Management Console menggunakan,, atau AWS CLI API.

DiAWS Management Console, buka konsol [Amazon Kinesis Video Streams](#). Pilih Streaming video di navigasi kiri.

Untuk informasi selengkapnya, lihat [the section called “Buat aliran video Kinesis”](#).

## Buat rahasia di AWS Secrets Manager

DiAWS Management Console, buka [AWS Secrets Managerkonsol](#). Pilih Rahasia di navigasi kiri.

Verifikasi bahwa Wilayah yang sesuai dipilih.

1. Pilih Simpan rahasia baru.

a. Langkah 1: Pilih tipe rahasia

- Pilih Jenis rahasia lainnya.
- Di bagian Pasangan Kunci/Nilai, tambahkan pasangan kunci-nilai.

Kunci: MediaURI

### Note

Kuncinya harusMediaURI. Ini peka huruf besar/kecil. Jika Anda salah memasukkannya, aplikasi tidak berfungsi.

Nilai:*Your MediaURI*.

Example

Contoh:rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/  
YourCameraMediaURI.

- Langkah 2: Konfigurasi rahasia. Beri nama rahasia ini. Beri nama apa pun yang Anda inginkan.
- Langkah 3: Konfigurasi rotasi - opsional. Pilih Berikutnya.
- Langkah 4: Tinjau. Pilih Toko.

2. Jika rahasia Anda tidak segera ditampilkan, pilih tombol refresh.

Pilih nama rahasia Anda. Catat ARN Rahasia.

3. Ulangi proses ini untuk setiap MediAuri yang ingin Anda streaming.

#### Note

AWS Jaringan memblokir beberapa sumber RTSP publik. Anda tidak dapat mengakses ini dari dalam instans Amazon EC2 atau jika Anda menjalankan unmanaged saat terhubung ke VPN.

#### Important

URL RTSP kamera Anda harus melakukan streaming video dalam format h.264. Durasi fragmen tidak boleh melebihi batas yang disebutkan dalam [the section called “Batas SDK produsen”](#).

Amazon Kinesis Video Streams Edge Agent hanya mendukung video.

Jalankan `gst-discoverer-1.0 Your RtspUrl` untuk memastikan kamera Anda dapat dijangkau dari perangkat Anda.

Simpan ARN untuk semua aliran dan rahasia yang Anda buat. Anda membutuhkan ini untuk langkah selanjutnya.

## Langkah 3: Buat kebijakan izin IAM

Ikuti prosedur ini untuk membuat kebijakan IAM. Kebijakan izin ini memungkinkan kontrol akses selektif (subset operasi yang didukung) untuk sumber daya. AWS Dalam hal ini, AWS sumber daya adalah aliran video yang Anda inginkan untuk streaming Agen Edge Amazon Kinesis Video Streams. Sumber daya juga mencakup AWS Secrets Manager rahasia yang dapat diambil oleh Agen Edge Amazon Kinesis Video Streams. Untuk informasi selengkapnya, lihat [kebijakan IAM](#).

Membuat kebijakan dengan menggunakan editor kebijakan JSON

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.

Jika ini pertama kalinya Anda memilih Kebijakan, akan muncul halaman Selamat Datang di Kebijakan Terkelola. Pilih Memulai.

3. Di bagian atas halaman, pilih Buat kebijakan.
4. Di bagian Editor kebijakan, pilih opsi JSON.
5. Masukkan dokumen kebijakan JSON berikut:

```
{
  "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

Ganti `arn:aws:kinesisvideo:*:*:stream/streamName1/*` dan `arn:aws:kinesisvideo:*:*:stream/streamName2/*` dengan ARN untuk streaming video, dan ganti `arn:aws:secretsmanager:*:*:secret:*` dengan ARN yang berisi rahasia MediAuri yang Anda buat. [the section called “2. Buat sumber daya untuk URL RTSP kamera IP Anda”](#) Gunakan ARN untuk mengetahui rahasia yang ingin diakses oleh Agen Edge Amazon Kinesis Video Streams.

#### 6. Pilih Berikutnya.

#### Note

Anda dapat beralih antara opsi editor Visual dan JSON kapan saja. Namun, jika Anda melakukan perubahan atau memilih Berikutnya di editor Visual, IAM dapat merestrukturisasi kebijakan Anda untuk mengoptimalkannya bagi editor visual. Untuk informasi selengkapnya, lihat [Restrukturisasi kebijakan](#) di Panduan Pengguna IAM.

7. Pada halaman Tinjau dan buat, masukkan nama Kebijakan dan Deskripsi opsional untuk kebijakan yang Anda buat. Tinjau Izin yang ditentukan dalam kebijakan ini untuk melihat izin yang diberikan oleh kebijakan Anda.
8. Pilih Buat kebijakan untuk menyimpan kebijakan baru Anda.

## Langkah 4: Buat peran IAM

Peran yang Anda buat dalam langkah ini dapat diasumsikan oleh AWS IoT untuk mendapatkan kredensial sementara dari AWS Security Token Service (AWS STS). Ini dilakukan saat melakukan permintaan otorisasi kredensial dari Agen Edge Amazon Kinesis Video Streams.

## Buat peran layanan untuk Amazon Kinesis Video Streams (konsol IAM)

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi konsol IAM, pilih Peran, dan lalu pilih Buat peran.
3. Pilih jenis peran kebijakan kepercayaan kustom dan tempel kebijakan berikut:

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

4. Pilih kotak di samping kebijakan IAM yang Anda buat. [the section called “3. Membuat kebijakan izin IAM”](#)
5. Pilih Berikutnya.
6. Masukkan nama peran atau akhiran nama peran untuk membantu Anda mengidentifikasi tujuan peran ini.

### Example

Contoh: KvsEdgeAgentRole

7. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk peran baru ini.
8. (Opsional) Tambahkan metadata ke peran dengan melampirkan tag sebagai pasangan kunci/nilai.

Untuk informasi selengkapnya tentang penggunaan tag di IAM, lihat [Menandai sumber daya IAM di Panduan Pengguna IAM](#).

9. Tinjau peran, lalu pilih Buat peran.

## Langkah 5: Buat alias AWS IoT peran

Ikuti prosedur ini untuk membuat alias AWS IoT peran untuk peran IAM yang Anda buat. [the section called “4. Membuat peran IAM”](#) Alias peran adalah model data alternatif yang menunjuk ke peran IAM. Permintaan penyedia AWS IoT kredensial harus menyertakan alias peran untuk menunjukkan peran IAM mana yang harus diambil untuk mendapatkan kredensi sementara dari (). AWS Security Token Service AWS STS Untuk informasi selengkapnya, lihat [Cara menggunakan sertifikat untuk mendapatkan token keamanan](#).

### Buat alias AWS IoT peran

1. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Verifikasi bahwa Wilayah yang sesuai dipilih.
3. Pada navigasi kiri, pilih Keamanan dan kemudian pilih Alias Peran.
4. Pilih Buat alias peran.
5. Masukkan nama untuk alias peran Anda.

### Example

Contoh: `KvsEdgeAgentRoleAlias`

6. Di dropdown Peran, pilih peran IAM yang Anda buat. [the section called “4. Membuat peran IAM”](#)
7. Pilih Buat. Pada halaman berikutnya, Anda melihat catatan bahwa alias peran Anda berhasil dibuat.
8. Cari dan pilih alias peran yang baru dibuat. Catat Peran alias ARN. Anda memerlukan ini untuk AWS IoT kebijakan di langkah berikutnya.

## Langkah 6: Buat AWS IoT kebijakan

Ikuti prosedur ini untuk membuat AWS IoT kebijakan yang akan dilampirkan ke sertifikat perangkat. Ini memberikan izin untuk AWS IoT kemampuan dan memungkinkan asumsi alias peran menggunakan sertifikat.

Dengan AWS IoT Core kebijakan, Anda dapat mengontrol akses ke bidang AWS IoT Core data. Bidang AWS IoT Core data terdiri dari operasi yang dapat Anda gunakan untuk melakukan hal berikut:

- Connect ke broker AWS IoT Core pesan
- Mengirim dan menerima pesan MQTT
- Dapatkan atau perbarui bayangan perangkat sesuatu

Untuk informasi selengkapnya, lihat [AWS IoT Corekebijakan](#).

Gunakan editor AWS IoT kebijakan untuk membuat AWS IoT kebijakan

1. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Pada navigasi kiri, pilih Keamanan dan kemudian pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Masukkan nama untuk kebijakan Anda.

#### Example

Contoh nama kebijakan adalah KvsEdgeAccessIoTPolicy.

5. (Opsional) Tambahkan metadata ke kebijakan dengan melampirkan tanda sebagai pasangan nilai kunci.


Untuk informasi selengkapnya tentang penggunaan tag di IAM, lihat [Menandai AWS IoT sumber daya Anda](#) di Panduan AWS IoT CorePengembang.

6. Pilih tab JSON.
7. Rekatkan dokumen kebijakan JSON berikut:

```
{
  "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

Ganti `your-role-alias-arn` dengan ARN alias peran yang Anda buat. [the section called "5. Buat alias AWS IoT peran"](#)

8. Pilih Buat untuk menyimpan pekerjaan Anda.

## Langkah 7: Buat AWS IoT sesuatu dan dapatkan kredensial untuk AWS IoT Core

Pada titik ini Anda telah membuat:

- Kebijakan izin IAM. Lihat [the section called "3. Membuat kebijakan izin IAM"](#).
- Peran IAM, dengan kebijakan izin terlampir. Lihat [the section called "4. Membuat peran IAM"](#).
- Alias AWS IoT peran untuk peran IAM. Lihat [the section called "5. Buat alias AWS IoT peran"](#).
- AWS IoT Kebijakan, saat ini tidak terikat pada AWS sumber daya apa pun. Lihat [the section called "6. Buat AWS IoT kebijakan"](#).

Untuk membuat dan mendaftarkan AWS IoT sesuatu dan mendapatkan kredensial AWS IoT Core akses

1. Daftarkan perangkat sebagai AWS IoT sesuatu dan hasilkan sertifikat X.509 untuk perangkat.
  - a. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
  - b. Pilih wilayah yang sesuai.

- c. Di navigasi kiri, pilih Semua perangkat, lalu pilih Things.
- d. Pilih Buat sesuatu.
- e. Pilih Buat satu hal, lalu pilih Berikutnya.

1. Langkah 1. Tentukan properti benda

Ketik nama untuk barang Anda, lalu pilih Berikutnya.

2. Langkah 2. Konfigurasi sertifikat perangkat

Pilih Auto-generate a new certificate (recommended), lalu pilih Next.

3. Langkah 3. Lampirkan kebijakan ke sertifikat

Cari kebijakan izin yang Anda buat. [the section called “6. Buat AWS IoT kebijakan”](#)

Pilih kotak centang di samping kebijakan Anda dan pilih Buat sesuatu.

- f. Di jendela yang muncul, unduh file-file berikut:

- Sertifikat perangkat. Ini adalah sertifikat X.509.
- File kunci publik
- File kunci pribadi
- Titik akhir layanan kepercayaan Amazon (kunci RSA 2048 bit: Amazon Root CA 1)

Catat lokasi masing-masing file ini untuk langkah selanjutnya.

- g. Pilih Selesai. Di halaman berikutnya, Anda melihat catatan bahwa barang Anda berhasil dibuat.
- h. Transfer file yang diunduh di atas ke AWS IoT barang Anda, jika belum ada.

2. Dapatkan titik akhir penyedia kredensial untuk akun Anda AWS.

### AWS CLI

Jalankan perintah berikut:

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

### AWS Management Console

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

Catat informasi ini untuk langkah selanjutnya.

3. Dapatkan titik akhir data perangkat untuk AWS akun Anda.

## AWS CLI

Jalankan perintah berikut:

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

## AWS Management Console

Lakukan hal-hal berikut:

1. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>.
2. Di navigasi kiri, pilih Pengaturan.
3. Temukan titik akhir data Perangkat.

Catat informasi ini untuk langkah selanjutnya.

4. (Opsional) Verifikasi bahwa sertifikat Anda dibuat dengan benar.

Jalankan perintah berikut untuk memvalidasi bahwa item Anda dihasilkan dengan benar.

```
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
```

Untuk informasi selengkapnya, lihat [Cara menggunakan sertifikat untuk mendapatkan token keamanan](#).

## Langkah 8: Bangun dan jalankan Agen Edge Amazon Kinesis Video Streams

### Bangun dan jalankan Agen Edge Amazon Kinesis Video Streams

1. Unduh tar file menggunakan tautan yang diberikan kepada Anda.

Jika Anda mengisi formulir minat Agen Edge Amazon Kinesis Video Streams, periksa email Anda untuk tautan unduhan. Jika Anda belum mengisi formulir, lengkapi [di sini](#).

2. Verifikasi checksum.
3. Ekstrak binari dan toples di perangkat Anda.

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

Setelah ekstraksi, struktur folder Anda akan terlihat seperti berikut:

```
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**

Nama folder rilis harus diatur dengan cara yang mencerminkan nomor rilis biner terbaru. Misalnya, rilis 1.0.0 akan memiliki nama folder ditetapkan sebagai 1.0.0.

## 4. Bangun jar dependensi.

**Note**

Guci yang disertakan dengan `kvs-edge-agent.tar.gz` tidak memiliki dependensi. Gunakan langkah-langkah berikut untuk membangun pustaka tersebut.

Arahkan ke `kvs-edge-agent` folder yang berisi `pom.xml`.

Ketik `mvn clean package`.

Ini menghasilkan file jar yang berisi dependensi yang dibutuhkan Agen Tepi Amazon Kinesis Video Streams. `kvs-edge-agent/target/libs.jar`

5. Tempatkan `libs.jar` ke dalam folder yang berisi artefak komponen.

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

## 6. Tetapkan variabel lingkungan menggunakan nilai dari langkah sebelumnya. Tabel berikut memberikan deskripsi untuk variabel.

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_REGION	Ya	wilayah yang digunakan.  Contoh: us-barat-2
AWS_IOT_CA_CERT	Ya	Jalur file ke sertifikat CA yang digunakan untuk membangun kepercayaan dengan layanan backend melalui TLS.

Nama Variabel Lingkungan	Diperlukan	Deskripsi
		<p>Contoh: <i>/file/path/to/AmazonRootCA1.pem</i></p>
AWS_IOT_CORE_CERT	Ya	<p>Jalur file ke sertifikat X.509.</p> <p>Contoh: <i>/file/path/to/certificateID-certificate.pem.crt</i></p>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	Ya	<p>Titik akhir <a href="#">penyedia titik akhir AWS IoT Core kredensial</a> untuk akun Anda. AWS</p> <p>Contoh: <i>credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com</i></p>
AWS_IOT_CORE_DATA_ATS_ENDPOINT	Ya	<p><a href="#">Titik akhir pesawat AWS IoT Core data</a> untuk AWS akun Anda.</p> <p>Contoh: <i>data-account-specific-prefix.iot.aws-region.amazonaws.com</i></p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_IOT_CORE_PRIVATE_KEY	Ya	<p>File path ke kunci pribadi yang digunakan dalam public/private key pair. Untuk informasi selengkapnya, lihat <a href="#">Manajemen kunci di AWS IoT</a>.</p> <p>Contoh: <i>/file/path/to/certificateID-private.pem.key</i></p>
AWS_IOT_CORE_ROLE_ALIAS	Ya	<p>Nama alias peran yang menunjuk ke peran AWS IAM untuk digunakan saat menghubungkan ke AWS IoT Core</p> <p>Contoh: <i>kvs-edge-role-alias</i></p>
AWS_IOT_CORE_THING_NAME	Ya	<p>Nama AWS IoT benda yang sedang dijalankan aplikasi.</p> <p>Contoh: <i>my-edge-device-thing</i></p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
GST_PLUGIN_PATH	Ya	<p>Jalur file yang menunjuk ke folder yang berisi pustaka yang bergantung pada <code>IngestorPipelineJNI</code> platform <code>gstkvssink</code> dan. Memungkinkan <code>GStreamer</code> memuat plugin ini. Untuk informasi selengkapnya, lihat <a href="#">Mengunduh, Membangun, dan Mengkonfigurasi Elemen <code>GStreamer</code></a>.</p> <p>Contoh: <code>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgent</i> <i>Version</i> /</code></p>



Nama Variabel Lingkungan	Diperlukan	Deskripsi
LD_LIBRARY_PATH	Ya	<p>Jalur file yang menunjuk ke direktori yang berisi pustaka yang bergantung pada KinesisVideoProducer platform cproducer dan.</p> <p>Contoh: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Tidak	<p>Menentukan apakah Agen Edge Streams Video Amazon Kinesis akan memposting metrik kesehatan pekerjaan ke Amazon CloudWatch</p> <p>Nilai yang diterima: TRUE/FALSE(case insensitive). Default FALSE jika tidak disediakan.</p> <p>Contoh: SALAH</p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_KVS_EDGE_LOG_LEVEL	Tidak	<p>Tingkat pencatatan keluaran Agen Tepi Amazon Kinesis Video Streams.</p> <p>Nilai yang diterima:</p> <ul style="list-style-type: none"><li>• MATI</li><li>• SEMUA</li><li>• MEMATIKAN</li><li>• ERROR</li><li>• PERINGATAN</li><li>• INFO, default, jika tidak disediakan</li><li>• AWAKUTU</li><li>• JEJAK</li></ul> <p>Contoh: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	Tidak	<p>Setelah file log mencapai ukuran ini, rollover akan terjadi.</p> <ul style="list-style-type: none"><li>• Min: 0</li><li>• Maks: 10000</li><li>• Default: 20, jika tidak disediakan</li><li>• Unit: Megabyte (MB)</li></ul> <p>Contoh 5:</p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	Tidak	<p>Jalur file yang mengarah ke direktori tempat log Agen Tepi Amazon Kinesis Video Streams dihasilkan. Default <code>./log</code> jika tidak disediakan.</p> <p>Contoh: <code>/file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Tidak	<p>Jumlah log yang digulung untuk disimpan sebelum dihapus.</p> <ul style="list-style-type: none"><li>• Min: 1</li><li>• Maks: 100</li><li>• Default: 10, jika tidak disediakan</li></ul> <p>Contoh: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Tidak	<p>Jalur file yang menunjuk ke direktori yang direkam media akan ditulis. Default ke direktori saat ini jika tidak disediakan.</p> <p>Contoh: <code>/file/path/</code></p>
GST_DEBUG	Tidak	<p>Menentukan tingkat log GStreamer untuk output. Untuk informasi selengkapnya, lihat dokumentasi <a href="#">GStreamer</a>.</p> <p>Contoh: 0</p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
GST_DEBUG_FILE	Tidak	Menentukan file output dari log debug GStreamer. Jika tidak disetel, log debug mendapatkan output ke kesalahan standar. Untuk informasi selengkapnya, lihat dokumentasi <a href="#">GStreamer</a> .  Contoh: <code>/tmp/gstreamer-logging .log</code>


7. Bersihkan cache GStreamer. Jenis:

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

Untuk informasi selengkapnya, lihat dokumentasi [registri GStreamer](#).

8. Siapkan dan jalankan perintah java. Agen Edge Streams Video Amazon Kinesis menerima argumen berikut:

Nama Properti Java	Diperlukan	Deskripsi
java.library.path	Tidak	Jalur file yang menunjuk ke folder yang berisi pustaka gstkvssink dan IngestorPipelineJNI dependen. Jika tidak disediakan, Agen Edge Amazon Kinesis Video Streams akan mencarinya di direktori saat ini.

 **Important**  
Agen Edge Streams  
Video Amazon

Nama Properti Java

Diperlukan

Deskripsi

Kinesis tidak akan berfungsi dengan benar jika tidak dapat menemukan file-file ini.

Contoh: */file/path/*

Untuk mengatur ini, tambahkan `-Djava-property-name=value` ke perintah java yang digunakan untuk menjalankan jar.

Sebagai contoh:

```
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

Jalankan perintah java di atas dari direktori yang sama seperti *download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*.

## 9. Kirim konfigurasi ke aplikasi menggunakan file. AWS CLI

### a. Buat file baru, *example-edge-configuration*.json.

Tempel kode berikut ke file. Ini adalah konfigurasi sampel yang mencatat setiap hari mulai pukul 9:00:00 hingga 16:59:59 (sesuai dengan waktu sistem di perangkat Anda). AWS IoT Ini juga mengunggah media yang direkam setiap hari dari pukul 19:00:00 hingga 9:59:59 PM.

Untuk informasi selengkapnya, lihat [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. Untuk mengirim file ke Agen Tepi Amazon Kinesis Video Streams, ketik berikut ini di: AWS CLI

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

10. Ulangi langkah sebelumnya untuk setiap streaming untuk Agen Tepi Amazon Kinesis Video Streams.

## Langkah 9: (Opsional) Instal CloudWatch agen di perangkat

### Note

Waspada [CloudWatchkuota](#).

Ikuti prosedur ini untuk menginstal dan mengonfigurasi CloudWatch agen agar secara otomatis mengunggah log yang dihasilkan oleh Agen Edge Amazon Kinesis Video Streams ke CloudWatch

Untuk [prosedur](#) menginstal CloudWatch agen di perangkat Anda, lihat Panduan CloudWatch Pengguna Amazon.

Ketika diminta untuk konfigurasi, pilih salah satu konfigurasi berikut.

### Important

`file_path` Dalam konfigurasi berikut mengasumsikan bahwa lokasi output logging default digunakan.

Jalur file yang digunakan mengasumsikan bahwa Anda menjalankan Agen Tepi Amazon Kinesis Video Streams dari lokasi: `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/`*version*

- Untuk mengonfigurasi CloudWatch agen untuk mengunggah log dan memposting metrik RAM dan CPU perangkat, rekatkan yang berikut ini ke dalam file konfigurasi.

```
{
  "agent": {
    "run_as_user": "ubuntu",
    "metrics_collection_interval": 60
  },
  "metrics": {
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
```

```

    ],
    "append_dimensions": {
      "IotThing": "YourIotThingName"
    }
  },
  "cpu": {
    "resources": [
      "*"
    ],
    "measurement": [
      "usage_active"
    ],
    "totalcpu": true,
    "append_dimensions": {
      "IotThing": "YourIotThingName"
    }
  }
},
"logs": {
  "logs_collected": {
    "files": {
      "collect_list": [
        {
          "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/java_kvs.log",
          "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
          "log_stream_name": "YourIotThingName-java_kvs.log"
        },
        {
          "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_edge.log*",
          "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
          "log_stream_name": "YourIotThingName-cpp_kvs_edge.log"
        },
        {
          "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvs_streams.log*",
          "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
          "log_stream_name": "YourIotThingName-cpp_kvs_streams.log"
        },
        {
          "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/cpp_kvssink.log*",

```





```

    }
  }
}

```

## Langkah 10: (Opsional) Jalankan Agen Edge Amazon Kinesis Video Streams sebagai proses asli

Siapkan Agen Edge Amazon Kinesis Video Streams sebagai layanan systemd.

systemd adalah manajer sistem dan layanan pada perangkat Linux. systemd adalah cara yang disarankan untuk mengelola proses, karena akan memulai ulang Agen Tepi Amazon Kinesis Video Streams jika aplikasi mengalami kesalahan atau perangkat yang menjalankan aplikasi kehilangan daya.

Lakukan hal-hal berikut:

Jalankan Agen Edge Amazon Kinesis Video Streams sebagai proses asli

1. Buat file baru `/etc/systemd/system` dan beri nama `aws.kinesisvideo.edge-runtime-agent.service`.

Tempel yang berikut ini:

```

[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
```

Untuk informasi selengkapnya tentang parameter yang diterima oleh file konfigurasi `systemd` layanan, lihat [dokumentasi](#).

### Note

Tambahkan variabel lingkungan yang diperlukan di . . . lokasi, seperti yang ditentukan dalam [the section called “8. Bangun dan jalankan Agen Edge Amazon Kinesis Video Streams”](#).

2. Muat ulang file layanan untuk menyertakan layanan baru.

Ketik `sudo systemctl daemon-reload`.

3. Mulai layanan.

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

4. Periksa status layanan Agen Edge Amazon Kinesis Video Streams untuk memverifikasi bahwa layanan tersebut sedang berjalan.

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

Berikut ini adalah contoh output yang akan Anda lihat.

```
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. Periksa log untuk kesalahan apa pun.

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

6. Ketik `systemctl --help` untuk daftar lengkap opsi untuk mengelola proses menggunakan `systemctl`.

Berikut ini adalah beberapa perintah umum untuk mengelola Agen Edge Amazon Kinesis Video Streams:

- Untuk memulai ulang, ketik `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`.
- Untuk berhenti, ketik `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- Untuk memulai secara otomatis pada setiap perangkat reboot, ketik `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

## Terapkan Agen Edge Streams Video Amazon Kinesis ke AWS IoT Greengrass

Ikuti langkah-langkah berikut untuk menyebarkan Agen Edge Amazon Kinesis Video Streams AWS IoT Greengrass untuk merekam dan mengunggah media dari kamera IP.

### Topik

- [Langkah 1: Buat instance Ubuntu Amazon EC2](#)
- [Langkah 2: Siapkan perangkat AWS IoT Greengrass V2 inti pada perangkat](#)
- [Langkah 3: Buat Amazon Kinesis Video AWS Secrets Manager Streams dan sumber daya untuk URL RTSP kamera IP Anda](#)
- [Langkah 4: Tambahkan izin ke peran layanan pertukaran token \(TES\)](#)
- [Langkah 5: Instal komponen AWS IoT Greengrass Secret Manager pada perangkat](#)
- [Langkah 6: Menerapkan komponen Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent di perangkat](#)
- [Langkah 7: \(Opsional\) Instal komponen pengelola AWS IoT Greengrass log pada perangkat](#)

## Langkah 1: Buat instance Ubuntu Amazon EC2

Lakukan hal berikut untuk membuat instance Ubuntu Amazon EC2.

Buat instans Ubuntu Amazon EC2

1. Masuk ke AWS Management Console dan buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.

Verifikasi bahwa Wilayah yang sesuai dipilih.

2. Pilih Luncurkan Instans.

Lengkapi bidang berikut:

- Nama - Ketik nama untuk contoh.
- Aplikasi dan Gambar OS (Amazon Machine Image) - Pilih Ubuntu.
- Jenis instans - Pilih t2.large.
- Login pasangan kunci - Buat key pair Anda sendiri.
- Pengaturan jaringan - Pertahankan default.
- Konfigurasi penyimpanan - Tingkatkan volume menjadi 256 GiB.
- Pengaturan lanjutan - Pertahankan default.

3. Luncurkan instance dan SSH ke dalamnya.

Lakukan hal-hal berikut:

1. Pilih Instans di navigasi kiri, lalu pilih ID instans.
  2. Pilih Connect di kanan atas.
  3. Pilih klien SSH dan ikuti instruksi di layar.
  4. Buka terminal dan arahkan ke .pem file yang diunduh (kemungkinan masuk~/Downloads).
  5. Pertama kali Anda mengikuti prosedur ini, Anda akan menerima pesan “Keaslian tuan rumah (...) tidak dapat ditetapkan.” Ketik ya.
4. Instal library sistem untuk membangun Agen Edge Amazon Kinesis Video Streams ke instance.

```
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

Jika Anda melihat layar yang memberi tahu Anda bahwa beberapa layanan perlu dimulai ulang, tekan Enter untuk memilih Ok.

Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon Corretto 11](#).

## Langkah 2: Siapkan perangkat AWS IoT Greengrass V2 inti pada perangkat

Ikuti prosedur ini untuk menginstal perangkat lunak AWS IoT Greengrass inti pada instans Amazon EC2.

Siapkan perangkat AWS IoT Greengrass inti

1. Masuk ke AWS Management Console, <https://console.aws.amazon.com/iot/>.

Verifikasi bahwa Wilayah yang sesuai dipilih.

2. Di navigasi kiri, pilih Perangkat Greengrass, Perangkat inti.
3. Pilih Siapkan satu perangkat inti.
4. Selesaikan langkah-langkah di layar.
  - Langkah 1: Daftarkan perangkat inti Greengrass. Ketik nama untuk perangkat.
  - Langkah 2: Tambahkan ke grup hal untuk menerapkan penerapan berkelanjutan. Pilih Tidak ada grup.
  - Langkah 3: Instal perangkat lunak Greengrass Core. Pilih Linux.
    - Langkah 3.1: Instal Java di perangkat


Java diinstal sebagai bagian dari [the section called "1. Buat instance Ubuntu"](#) Kembali ke langkah itu jika Anda belum menginstal Java.

- Langkah 3.2: Salin AWS kredensial ke perangkat

Buka bash/zsh opsi dan tempel perintah ekspor di instans Amazon EC2.


- Langkah 3.3: Jalankan installer

1. Salin dan jalankan perintah Unduh penginstal dan Jalankan penginstal di instance Ubuntu Amazon EC2.

 Note

Perintah Run the installer akan secara otomatis diperbarui berdasarkan nama yang Anda pilih pada langkah sebelumnya.

2. Catat peran layanan pertukaran token (TES) yang dibuat. Anda membutuhkannya nanti.

 Note

Secara default, peran yang dibuat disebut GreenGrassV2 TokenExchangeRole.

## Langkah 3: Buat Amazon Kinesis Video AWS Secrets Manager Streams dan sumber daya untuk URL RTSP kamera IP Anda

Ikuti prosedur ini untuk membuat aliran dan rahasia yang dibutuhkan. AWS Secrets Manager Lakukan langkah ini terlebih dahulu, karena Anda memerlukan ARN dari sumber daya yang dibuat dalam kebijakan.

### Buat Streaming Video Amazon Kinesis

Buat Amazon Kinesis Video Streams AWS Management Console menggunakan,, atau AWS CLI API.

DiAWS Management Console, buka konsol [Amazon Kinesis Video Streams](#). Pilih Streaming video di navigasi kiri.

Untuk informasi selengkapnya, lihat [the section called “Buat aliran video Kinesis”](#).

### Buat rahasia di AWS Secrets Manager

DiAWS Management Console, buka [AWS Secrets Managerkonsol](#). Pilih Rahasia di navigasi kiri.


Verifikasi bahwa Wilayah yang sesuai dipilih.

1. Pilih Simpan rahasia baru.

a. Langkah 1: Pilih tipe rahasia

- Pilih Jenis rahasia lainnya.
- Di bagian Pasangan Kunci/Nilai, tambahkan pasangan kunci-nilai.

Kunci: MediaURI

 Note

Kuncinya harus MediaURI. Ini peka huruf besar/kecil. Jika Anda salah memasukkannya, aplikasi tidak berfungsi.

Nilai: *Your MediaURI*.


Example

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

- Langkah 2: Konfigurasi rahasia. Beri nama rahasia ini. Beri nama apa pun yang Anda inginkan.
  - Langkah 3: Konfigurasi rotasi - opsional. Pilih Berikutnya.
  - Langkah 4: Tinjau. Pilih Toko.
2. Jika rahasia Anda tidak segera ditampilkan, pilih tombol refresh.

Pilih nama rahasia Anda. Catat ARN Rahasia.

3. Ulangi proses ini untuk setiap MediaURI yang ingin Anda streaming.

 Note

AWS Jaringan memblokir beberapa sumber RTSP publik. Anda tidak dapat mengakses ini dari dalam instans Amazon EC2 atau jika Anda menjalankan unmanaged saat terhubung ke VPN.



**⚠ Important**

URL RTSP kamera Anda harus melakukan streaming video dalam format h.264. Durasi fragmen tidak boleh melebihi batas yang disebutkan dalam [the section called “Batas SDK produsen”](#). Amazon Kinesis Video Streams Edge Agent hanya mendukung video.

Jalankan `gst-discoverer-1.0 Your RtspUrl` untuk memastikan kamera Anda dapat dijangkau dari perangkat Anda.

Simpan ARN untuk semua aliran dan rahasia yang Anda buat. Anda membutuhkan ini untuk langkah selanjutnya.

## Langkah 4: Tambahkan izin ke peran layanan pertukaran token (TES)

Berikan peran layanan pertukaran token (TES) ke perangkat yang mengasumsikan izin untuk melihat rahasia. Ini diperlukan agar AWS Secrets Manager AWS IoT Greengrass komponen berfungsi dengan benar.

Tambahkan izin ke peran TES

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Peran di navigasi kiri dan cari peran TES yang Anda buat sebelumnya dalam proses.
3. Di menu tarik-turun Tambahkan izin, pilih Lampirkan kebijakan.
4. Pilih Buat kebijakan.
5. Gulir ke bawah dan pilih Edit.
6. Di editor kebijakan, pilih JSON dan edit kebijakan.

Ganti kebijakan dengan yang berikut:

**Note**

Ganti `arn:aws:kinesisvideo:*:*:stream/streamName1/*` dan `arn:aws:kinesisvideo:*:*:stream/streamName2/*` dengan ARN untuk aliran yang Anda buat pada langkah sebelumnya.

```
{
  "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. Pada halaman Tambahkan tanda, pilih Berikutnya:
8. Beri nama kebijakan Anda, lalu pilih Buat kebijakan.

Contoh nama kebijakan adalah `KvsEdgeAccessPolicy`.

9. Tutup tab dan kembali ke tab tempat Anda melampirkan kebijakan ke peran TES.


Pilih tombol refresh, lalu cari kebijakan yang baru dibuat.

Pilih kotak centang dan pilih Lampirkan kebijakan.

Di layar berikutnya, Anda melihat catatan yang mengatakan Kebijakan berhasil dilampirkan ke peran.

10. Buat dan lampirkan kebijakan lain, kali ini untuk rahasia Anda.

Ganti kebijakan dengan yang berikut:

 Note

Ganti `arn:aws:secretsmanager:*:*:secret:*` dengan ARN yang berisi rahasia MediAuri yang Anda buat. [the section called “3. Buat sumber daya untuk URL RTSP kamera IP Anda”](#)

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

11. Buat dan lampirkan kebijakan lain, kali ini untuk Amazon CloudWatch metrik. Ganti kebijakan dengan yang berikut:

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

```

```
    "Resource": [
      "*"
    ]
  }
]
```

## Langkah 5: Instal komponen AWS IoT Greengrass Secret Manager pada perangkat

Agan Edge Streams Video Amazon Kinesis AWS IoT Greengrass memerlukan komponen Secret Manager untuk diinstal pada perangkat terlebih dahulu.

Instal komponen Secret Manager

1. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>. Verifikasi bahwa Wilayah yang sesuai dipilih.
2. Di navigasi kiri, pilih Perangkat Greengrass, Deployment.

Pilih penerapan dengan target yang sama dengan hal yang kita buat. [the section called "2. Siapkan perangkat AWS IoT Greengrass inti"](#)

3. Di menu tarik-turun Tindakan di pojok kanan atas, pilih Revisi.


Di pop-up yang muncul, pilih Revise deployment.

4. Lengkapi bagian berikut:

- Langkah 1: Tentukan target. Pilih Berikutnya.
- Langkah 2: Pilih komponen.
  - Verifikasi bahwa komponen AWS.greengrass.cli dipilih. Jangan hapus instalasi komponen ini.
  - Alihkan sakelar Tampilkan hanya komponen yang dipilih dan cari aws.greengrass.SecretManager.
  - Centang kotak di sebelah aws.greengrass.SecretManager, lalu pilih Berikutnya.
- Langkah 3: Konfigurasi komponen. Konfigurasi komponen AWS IoT Greengrass Secret Manager untuk mengunduh rahasia dari dalam AWS IoT Greengrass lingkungan.


Pilih aws.greengrass.SecretManagerkomponen, lalu pilih Configure component.

Di layar yang muncul, perbarui AWS Secrets Manager ARN di kotak Konfigurasi untuk digabungkan.

 Note

Ganti `arn:aws:secretsmanager:*:*:secret:*` dengan ARN rahasia yang Anda buat. [the section called “3. Buat sumber daya untuk URL RTSP kamera IP Anda”](#)

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

 Note

`cloudSecrets` adalah daftar objek dengan kunci `arn`. Untuk informasi selengkapnya, lihat bagian [Konfigurasi manajer rahasia](#) di Panduan AWS IoT Greengrass Version 2 Pengembang.

Setelah selesai, pilih Konfirmasi, lalu pilih Berikutnya.

- Langkah 4: Konfigurasi pengaturan lanjutan. Pilih Selanjutnya.
  - Langkah 5: Tinjau. Pilih Terapkan.
5. Konfirmasikan bahwa AWS Secrets Manager komponen dan izin telah diinstal dengan benar.

Pada instance Ubuntu Amazon EC2, ketik `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` untuk memverifikasi bahwa komponen menerima konfigurasi yang diperbarui.

6. Periksa log AWS IoT Greengrass inti.

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

Tinjau kesalahan penerapan.

Jika ada kesalahan, revisi penerapan untuk menghapus komponen.

```
aws.greengrass.SecretManager
```

Ketik `sudo service greengrass restart` untuk memulai ulang layanan AWS IoT Greengrass inti.

Jika kesalahan penerapan terkait dengan izin yang hilang, tinjau [the section called “4. Tambahkan izin ke peran TES”](#) bagian tersebut untuk memastikan bahwa peran TES memiliki izin yang tepat. Kemudian, ulangi bagian ini.

## Perbarui rahasia pada komponen AWS IoT Greengrass Secret Manager

### Important

Komponen AWS IoT Greengrass Secret Manager mengambil dan menyimpan rahasia hanya ketika penerapan diperbarui.

Untuk memperbarui rahasia pada komponen AWS IoT Greengrass Secret Manager, ikuti langkah sebelumnya 1—6, dengan perubahan berikut.

Langkah 3: Konfigurasi komponen. Konfigurasi komponen AWS IoT Greengrass Secret Manager untuk mengunduh rahasia dari dalam AWS IoT Greengrass lingkungan.

Pilih `aws.greengrass.SecretManager` komponen, lalu pilih `Configure component`.

Di layar yang muncul, tempel `[""]` di kotak `Atur ulang jalur`, dan perbarui `AWS Secrets Manager ARN` di kotak `Konfigurasi` untuk digabungkan.

Untuk informasi selengkapnya, lihat [Mengatur ulang pembaruan](#).

## Langkah 6: Menerapkan komponen Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent di perangkat

Menerapkan komponen Agen Tepi Amazon Kinesis Video Streams di perangkat AWS IoT Greengrass

1. Unduh tar file menggunakan tautan yang disediakan.

Jika Anda mengisi formulir minat Agen Edge Amazon Kinesis Video Streams, periksa email Anda untuk tautan unduhan. Jika Anda belum mengisi formulir, lengkapi [di sini](#).

2. Verifikasi checksum.
3. Ekstrak binari dan toples di perangkat Anda.


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

Setelah ekstraksi, struktur folder Anda akan terlihat seperti berikut:

```
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

Nama folder rilis harus diatur dengan cara yang mencerminkan nomor rilis biner terbaru. Misalnya, rilis 1.0.0 akan memiliki nama folder ditetapkan sebagai 1.0.0.

4. Bangun jar dependensi.

 Note

Guci yang disertakan kvs-edge-agent dengan.tar.gz tidak memiliki dependensi. Gunakan langkah-langkah berikut untuk membangun pustaka tersebut.

Arahkan ke kvs-edge-agent folder yang berisipom.xml.

Ketik `mvn clean package`.

Ini akan menghasilkan file jar yang berisi dependensi yang dibutuhkan Agen Tepi Amazon Kinesis Video Streams. kvs-edge-agent/target/libs.jar

5. Tempatkan libs.jar ke dalam folder yang berisi artefak komponen.

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

6. Opsional. Konfigurasi properti. Agen Edge Streams Video Amazon Kinesis menerima variabel lingkungan berikut dalam mode: AWS IoT Greengrass

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_REGION	Ya	wilayah yang digunakan.  Contoh: us-barat-2  AWS IoT Greengrass perangkat lunak inti secara otomatis menetapkan nilai ini



Nama Variabel Lingkungan	Diperlukan	Deskripsi
GST_PLUGIN_PATH	Ya	<p>untuk Anda. Untuk informasi selengkapnya, lihat topik <a href="#">referensi variabel lingkungan Komponen</a> di Panduan AWS IoT Greengrass Version 2 Pengembang.</p> <p>Jalur file yang menunjuk ke folder yang berisi pustaka yang bergantung pada IngestorPipelineJNI platform gstkvssink dan. Ini memungkinkan GStreamer memuat plugin ini. Untuk informasi selengkapnya, lihat <a href="#">Mengunduh, Membangun, dan Mengkonfigurasi Elemen GStreamer</a>.</p> <p>Contoh: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
LD_LIBRARY_PATH	Ya	<p>Jalur file yang menunjuk ke direktori yang berisi pustaka yang bergantung pada KinesisVideoProducer platform cproducer dan.</p> <p>Contoh: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	Tidak	<p>Menentukan apakah Agen Edge Streams Video Amazon Kinesis akan memposting metrik kesehatan pekerjaan ke Amazon CloudWatch</p> <p>Nilai yang diterima: TRUE/FALSE(case insensitive). Default FALSE jika tidak disediakan.</p> <p>Contoh: SALAH</p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_KVS_EDGE_LOG_LEVEL	Tidak	<p>Tingkat pencatatan keluaran Agen Tepi Amazon Kinesis Video Streams.</p> <p>Nilai yang diterima:</p> <ul style="list-style-type: none"><li>• MATI</li><li>• SEMUA</li><li>• MEMATIKAN</li><li>• ERROR</li><li>• PERINGATAN</li><li>• INFO, default, jika tidak disediakan</li><li>• AWAKUTU</li><li>• JEJAK</li></ul> <p>Contoh: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	Tidak	<p>Setelah file log mencapai ukuran ini, rollover akan terjadi.</p> <ul style="list-style-type: none"><li>• Min: 1</li><li>• Maks: 100</li><li>• Default: 20, jika tidak disediakan</li><li>• Unit: Megabyte (MB)</li></ul> <p>Contoh 5:</p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	Tidak	<p>Jalur file yang mengarah ke direktori tempat log Agen Tepi Amazon Kinesis Video Streams dihasilkan. Default <code>./log</code> jika tidak disediakan.</p> <p>Contoh: <code>/file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	Tidak	<p>Jumlah log yang digulung untuk disimpan sebelum dihapus.</p> <ul style="list-style-type: none"> <li>• Min: 1</li> <li>• Maks: 100</li> <li>• Default: 10, jika tidak disediakan</li> </ul> <p>Contoh: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	Tidak	<p>Jalur file yang menunjuk ke direktori yang direkam media akan ditulis. Default ke direktori saat ini jika tidak disediakan.</p> <p>Contoh: <code>/file/path/</code></p>
GREENGRASS_ROOT_DIRECTORY	Tidak	<p>Jalur file ke direktori AWS IoT Greengrass root.</p> <p>Ini default <code>/greengrass/v2/</code> jika tidak disediakan.</p> <p>Contoh: <code>/file/path/</code></p>

Nama Variabel Lingkungan	Diperlukan	Deskripsi
GST_DEBUG	Tidak	Menentukan tingkat log GStreamer untuk output. Untuk informasi selengkapnya, lihat dokumentasi <a href="#">GStreamer</a> .  Contoh: 0
GST_DEBUG_FILE	Tidak	Menentukan file output dari log debug GStreamer. Jika tidak disetel, log debug mendapatkan output ke kesalahan standar. Untuk informasi selengkapnya, lihat dokumentasi <a href="#">GStreamer</a> .  Contoh: <i>/tmp/gstreamer-logging .log</i>

Buka `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` dan modifikasi skrip run untuk menambahkan salah satu variabel lingkungan sebelumnya.

#### Important

Pastikan skrip run yang dimodifikasi tidak berisi karakter tab apa pun. Perangkat lunak AWS IoT Greengrass inti tidak akan dapat membaca resepnya.

## 7. Menerapkan komponen Agen Tepi Amazon Kinesis Video Streams. AWS IoT Greengrass

Jenis:

```
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"
```

Untuk informasi tambahan, lihat bagian berikut di Panduan AWS IoT Greengrass Version 2 Pengembang:

- [AWS IoT GreengrassPerintah CLI](#)
- [Menyebarkan AWS IoT Greengrass komponen ke perangkat](#)

8. Kirim konfigurasi ke aplikasi menggunakan file. AWS CLI

- a. Buat file baru, *example-edge-configuration.json*.

Tempel kode berikut ke file. Ini adalah konfigurasi sampel yang mencatat setiap hari mulai pukul 9:00:00 hingga 16:59:59 (sesuai dengan waktu sistem di perangkat Anda). AWS IoT Ini juga mengunggah media yang direkam setiap hari dari pukul 19:00:00 hingga 9:59:59 PM.

Untuk informasi selengkapnya, lihat [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. Ketik berikut ini AWS CLI untuk mengirim file ke Agen Edge Amazon Kinesis Video Streams:

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

9. Ulangi langkah sebelumnya untuk setiap streaming untuk Agen Tepi Amazon Kinesis Video Streams.

## Langkah 7: (Opsional) Instal komponen pengelola AWS IoT Greengrass log pada perangkat

### Note

Waspada [CloudWatchkuota](#).

Ikuti prosedur ini untuk mengonfigurasi log Agen Tepi Amazon Kinesis Video Streams untuk CloudWatch mengunggah secara otomatis AWS IoT Greengrass menggunakan komponen pengelola log.

Instal komponen pengelola AWS IoT Greengrass log

1. Konfirmasikan bahwa peran AWS IoT Greengrass perangkat memiliki [izin yang sesuai](#).
  - a. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
  - b. Klik Peran di navigasi kiri.
  - c. Pilih nama peran TES yang dibuat di [the section called “2. Siapkan perangkat AWS IoT Greengrass inti”](#). Gunakan bilah pencarian jika perlu.
  - d. Pilih GreengrassV2TokenExchangeRoleAccess kebijakan.

- e. Pilih tab JSON dan verifikasi bahwa kebijakan terlihat seperti berikut:

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

- f. Jika `GreengrassV2TokenExchangeRoleAccess` kebijakan tidak ada, atau jika beberapa izin yang diperlukan tidak ada, buat kebijakan IAM baru dengan izin ini dan lampirkan ke peran TES yang dibuat. [the section called “2. Siapkan perangkat AWS IoT Greengrass inti”](#)
2. Masuk ke AWS Management Console dan buka AWS IoT Core konsol di <https://console.aws.amazon.com/iot/>. Verifikasi bahwa Wilayah yang sesuai dipilih.
  3. Di navigasi kiri, pilih Perangkat Greengrass, Deployment.  
  
Pilih penerapan dengan target yang sama dengan hal yang Anda buat. [the section called “2. Siapkan perangkat AWS IoT Greengrass inti”](#)
  4. Di pojok kanan atas, pilih Tindakan, lalu pilih Revisi.  
  
Di pop-up yang muncul, pilih Revise deployment.
  5. Lengkapi bagian berikut:
    - a. Langkah 1: Tentukan target. Pilih Berikutnya.
    - b. Langkah 2: Pilih komponen.
      - i. Verifikasi bahwa komponen `AWS.greengrass.cli` dan `aws.greengrass.SecretManager` komponen masih dipilih.



**⚠ Important**

Jangan hapus instalasi komponen ini.

- ii. Alihkan sakelar Tampilkan hanya komponen yang dipilih dan cari `aws.greengrass.LogManager`.
  - iii. Pilih kotak di sebelah `aws.greengrass.LogManager`, lalu pilih Berikutnya.
- c. Langkah 3: Konfigurasi komponen. Konfigurasi komponen pengelola AWS IoT Greengrass log untuk mengunggah log yang dihasilkan oleh Agen Tepi Amazon Kinesis Video Streams.

Pilih `aws.greengrass.LogManager` komponen, lalu pilih `Configure component`.

Di layar yang muncul, tempel konfigurasi pengelola log berikut di kotak Konfigurasi untuk digabungkan.

```
{
  "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` Dalam konfigurasi sebelumnya mengasumsikan bahwa lokasi output logging default digunakan.

#### Note

Untuk informasi selengkapnya tentang masing-masing parameter untuk konfigurasi pengelola log, lihat bagian [Manajer log](#) dari Panduan AWS IoT Greengrass Version 2 Pengembang.

Setelah selesai, pilih Konfirmasi, lalu pilih Berikutnya.

- d. Langkah 4: Konfigurasikan pengaturan lanjutan. Pilih Selanjutnya.
- e. Langkah 5: Tinjau. Pilih Terapkan.
6. Konfirmasikan bahwa komponen pengelola AWS log dan izin telah diinstal dengan benar.
7. Pada instance Ubuntu Amazon EC2, ketik `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` untuk memverifikasi komponen menerima konfigurasi yang diperbarui.
8. Periksa log AWS IoT Greengrass inti.

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

Tinjau kesalahan penerapan.

Jika ada kesalahan, revisi penerapan untuk menghapus komponen.

```
aws.greengrass.LogManager
```

Ketik `sudo service greengrass restart` untuk memulai ulang layanan AWS IoT Greengrass inti.

Jika kesalahan penerapan terkait dengan izin yang hilang, tinjau [the section called “4. Tambahkan izin ke peran TES”](#) untuk memastikan bahwa peran TES memiliki izin yang tepat. Kemudian, ulangi bagian ini.

## FAQ Agen Edge Streams Video Kinesis Amazon

Berikut ini adalah beberapa pertanyaan umum untuk layanan Agen Edge Amazon Kinesis Video Streams.

### Sistem operasi apa yang didukung Amazon Kinesis Video Streams Edge Agent?

Amazon Kinesis Video Streams Edge Agent saat ini mendukung sistem operasi berikut:

Ubuntu

- 22.x
  - AMD64
- 18.x
  - LENGAN

AL2

- amzn2
  - AMD64 amazonlinux:2.0.20210219.0-amd64 (Bola Salju)

### Apakah Agen Edge Amazon Kinesis Video Streams mendukung media H.265?

Amazon Kinesis Video Streams Edge Agent hanya mendukung aliran dasar H.264.

## Apakah Agen Edge Streams Video Amazon Kinesis berfungsi di AL2?

Ya.

## Bagaimana saya bisa menjalankan beberapa aliran di dalam AWS IoT benda atau perangkat?

Kirim yang lain [the section called "StartEdgeConfigurationUpdate"](#) ke Amazon Kinesis Video AWS Secrets Manager Streams/ARN yang sama `HubDeviceArn`, tetapi berbeda.

## Bagaimana saya bisa mengedit **StartEdgeConfigurationUpdate** setelah dikirim?

Kirim yang diperbarui [the section called "StartEdgeConfigurationUpdate"](#) ke yang sama `HubDeviceArn` dengan Amazon Kinesis Video Streams ARN yang sama. Saat aplikasi menerima pesan dari Amazon Kinesis Video Streams, aplikasi akan mengganti konfigurasi sebelumnya untuk streaming tersebut. Perubahan akan terjadi kemudian.

## Apakah Anda memiliki contoh umum **ScheduleConfigs**?

Agen Edge Streams Video Amazon Kinesis menggunakan waktu sistem perangkat yang sedang berjalan.

Deskripsi	ScheduleExpression	DurationInSeconds
Rekaman 24/7, mengunggah setiap jam	(no1 ScheduleConfig)	
9:00:00 AM - 4:59:59 PM setiap hari	0 0 9-16 * *? *	3599
9:00:00 AM - 4:59:59 PM weekdays	0 0 9-16? * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599

Deskripsi	ScheduleExpression	DurationInSeconds
	0 0 9-16 ? * MON,TUE,W ED,THU,FRI *	3599
9:00:00 AM - 16:59:59 PM akhir pekan	0 0 9-16 ? * SAT,SUN *	3599
10:00:00 PM - 23.59:59 PM hari kerja	0 0 22,23 ? * MON-FRI *	3599
9:00:00 AM - 10:00:00 AM setiap hari	0 0 9 * * ? *	3600
16:00:00 - 17:59:59 setiap hari	0 0 16-17 * * ? *	3599

Untuk contoh lainnya, lihat [dokumentasi Kuarsa](#).

## Apakah ada batas aliran maksimum?

Agen Edge Streams Video Amazon Kinesis saat ini memiliki batas keras 16 aliran per perangkat. Gunakan [the section called “DeleteEdgeConfiguration”](#) API untuk menghapus aliran dari perangkat. Memperbarui konfigurasi untuk aliran yang sama menggunakan [the section called “StartEdgeConfigurationUpdate”](#) tidak meningkatkan jumlah aliran perangkat.

## Bagaimana cara memulai kembali pekerjaan yang telah salah?

Jika terjadi kesalahan, Agen Tepi Amazon Kinesis Video Streams akan mencoba memulai ulang pekerjaan. Namun, dengan beberapa kesalahan (seperti kesalahan konfigurasi), Anda harus me-restart pekerjaan secara manual.

Untuk menentukan pekerjaan mana yang perlu dimulai ulang secara manual, lihat FatalErrormetrik di [the section called “Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch”](#)

Kirim ulang [the section called “StartEdgeConfigurationUpdate”](#) untuk memulai ulang pekerjaan untuk streaming.

## Bagaimana cara memantau kesehatan Agen Tepi Amazon Kinesis Video Streams saya?

Lihat informasi yang lebih lengkap di [the section called “Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch”](#).

# Streaming video melalui VPC

Beta ini tersedia dalam pratinjau di Wilayah Eropa (Paris), eu-barat-3. Untuk mengakses komponen-komponen ini dan panduan memulai kami, [kirim email kepada kami](#).

Layanan endpoint VPC Amazon Kinesis Video Streams memungkinkan Anda melakukan streaming dan mengonsumsi video melalui jaringan Amazon tanpa data apa pun melalui internet publik.

Untuk meminta akses, [email kami](#) informasi berikut:

- ID Akun
- Streaming ARN
- ID VPC

## Note

Mungkin diperlukan waktu hingga satu minggu bagi kami untuk menambahkan Anda ke layanan.

Jika Anda belum pernah bekerja dengan titik akhir VPC di masa lalu, tinjau informasi berikut untuk membiasakan diri dengan konsepnya:

- [AWS PrivateLinklatar belakang](#)
- [Panduan memulai VPC](#)

## Informasi tambahan

Setelah Anda ditambahkan ke versi beta, kami akan mengirimkan tautan ke informasi tambahan tentang fitur ini melalui email kepada Anda.

## Prosedur titik akhir VPC

### Kuota

Perbedaan kuota utama adalah:

- Kuota yang lebih rendah untuk semua API bandwidth (2 mbps):
  - PutMedia
  - GetMedia
  - GetMediaForFragmentList
- 10 aliran diizinkan per pelanggan

### Buat titik akhir

Setelah diizinkan terdaftar, Anda akan menerima nama layanan titik akhir VPC untuk Amazon Kinesis Video Streams. Ini akan terlihat seperti `com.amazonaws.region.kinesisvideo`.

Buat [titik akhir VPC antarmuka](#) untuk Amazon Kinesis Video Streams menggunakan Konsol VPC Amazon atau (). AWS Command Line Interface AWS CLI

Dalam AWS CLI, ketik berikut ini:

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

#### Important

Lalu lintas dalam VPC Anda akan menggunakan DNS pribadi untuk merutekan titik akhir. Jika Anda tidak mengaktifkan ini, Anda harus menerapkan logika DNS Anda sendiri. Untuk informasi selengkapnya tentang DNS pribadi, lihat [AWS PrivateLink dokumentasi](#).

Untuk informasi lebih lanjut tentang AWS CLI opsi, lihat [create-vpc-endpoint](#).



## Kontrol akses ke titik akhir

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke Amazon Kinesis Video Streams. Kebijakan menentukan informasi berikut ini:

- kepala sekolah yang dapat melakukan tindakan,
- tindakan yang dapat dilakukan, dan
- sumber daya di mana tindakan dapat dilakukan.

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan dengan titik akhir VPC menggunakan kebijakan titik akhir](#) dalam Panduan. AWS PrivateLink

Berikut ini adalah contoh kebijakan endpoint untuk Amazon Kinesis Video Streams. Saat dilampirkan ke titik akhir, kebijakan ini menolak akses ke PutMedia tindakan yang tercantum untuk semua prinsipal di semua sumber daya.

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

# Gambar di Kinesis Video Streams

Anda dapat menggunakan Amazon Kinesis Video Streams API dan SDK untuk membantu Anda mengekstrak gambar dari streaming video Anda. Anda dapat menggunakan gambar ini untuk aplikasi pemutaran yang disempurnakan seperti thumbnail atau scrubbing yang disempurnakan, atau untuk digunakan dalam pipeline machine learning. Kinesis Video Streams menawarkan ekstraksi gambar sesuai permintaan melalui API, atau ekstraksi gambar otomatis dari tag metadata dalam video yang tertelan.

Untuk informasi tentang penggunaan dukungan terkelola Kinesis Video Streams untuk gambar, lihat:

- [Pembuatan Gambar Sesuai Permintaan \(GetImages\)](#)- API ini memungkinkan pelanggan untuk mengekstrak satu gambar atau beberapa gambar dari video yang disimpan di Kinesis Video Streams.
- [Pembuatan Gambar Otomatis \(Pengiriman S3\)](#) - Konfigurasikan Kinesis Video Streams untuk secara otomatis mengekstrak gambar dari data video secara real time berdasarkan tag dalam video yang diunggah, dan kirimkan gambar ke bucket S3 yang ditentukan pelanggan.

## Topik

- [Memulai dengan GetImages](#)
- [Memulai pengiriman Amazon S3](#)

## Memulai dengan GetImages

Dukungan terkelola untuk Images menyediakan cara yang dikelola sepenuhnya untuk mendapatkan gambar dari data video yang dialirkan dan disimpan di Kinesis Video Streams. Anda dapat menggunakan gambar untuk menjalankan beban kerja machine learning (ML/machine learning) seperti deteksi orang, hewan peliharaan, atau kendaraan. Gambar juga dapat digunakan untuk menambahkan elemen interaktif ke pemutaran, seperti pratinjau gambar untuk acara gerak dan menggosok untuk klip video.

Untuk informasi lebih lanjut tentang Get Images fitur, lihat [GetImages](#) di dalam Amazon Kinesis Video Streaming Media yang Diarsipkan Panduan referensi API.

## Memulai pengiriman Amazon S3

Saat ini, pelanggan menjalankan dan mengelola pipeline transcoding gambar mereka sendiri untuk membuat gambar untuk berbagai keperluan seperti scrubbing, pratinjau gambar, menjalankan model ML pada gambar, dan banyak lagi. Kinesis Video Streams menawarkan kemampuan untuk mentranskode dan mengirimkan gambar. Kinesis Video Streams akan secara otomatis mengekstrak gambar dari data video secara real-time berdasarkan tag, dan mengirimkan gambar ke bucket S3 yang ditentukan pelanggan.

### UpdateImageGenerationConfiguration

Untuk menyiapkan aliran video Kinesis untuk mengaktifkan pembuatan gambar ke Amazon S3:

1. Buat sebuah Ember S3 untuk pembuatan gambar berdasarkan tag yang ditambahkan dalam SDK menggunakan API baru. Perhatikan JENIS S3, yang diperlukan pada langkah berikutnya, saat memperbarui konfigurasi pembuatan gambar untuk aliran.
2. Buat file JSON bernama `update-image-generation-input.json` dengan konten berikut sebagai masukan.

```
{
  "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
  }
}
```

Anda dapat menggunakan AWS CLI untuk memohon [UpdateImageGenerationConfiguration](#) Operasi API untuk menambahkan Amazon S3 ARN yang dibuat sebelumnya dan mengubah statusnya `ENABLED`.

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

#### Permintaan:

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).
  }
}
```

#### Tanggapan:

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

```
}
```

### Note

Diperlukan setidaknya 1 menit untuk memulai alur kerja pembuatan gambar setelah memperbarui konfigurasi pembuatan gambar. Tunggu setidaknya 1 menit sebelum memohonPutMediasetelah panggilan pembaruan.

## DescribeImageGenerationConfiguration

Untuk melihat konfigurasi pembuatan gambar yang sudah ditetapkan untuk stream, pelanggan dapat membuatDescribeImageGenerationConfigurationpermintaan, sebagai berikut.

Permintaan:

```
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
```

```
}
```

Tanggapan:

```
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'
}
}

```

Untuk mempelajari lebih lanjut tentang `DescribeImageGenerationConfiguration` fitur, lihat [DescribeImageGenerationConfiguration](#) di dalam Panduan Pengembang Amazon Kinesis Video Streams.

## Tag MKV produser

Anda dapat menggunakan Kinesis Video Streams Produser SDK untuk menandai fragmen minat tertentu dengan mengekspos operasi API di SDK. Untuk contoh tag, lihat [kode ini](#). Setelah memanggil API ini, SDK akan menambahkan satu set tag MKV yang telah ditentukan bersama dengan data fragmen. Kinesis Video Streams akan mengenali tag MKV khusus ini dan memulai alur kerja pembuatan gambar berdasarkan konfigurasi pemrosesan gambar aliran itu.

Metadata fragmen apa pun yang disediakan bersama dengan tag pembuatan gambar Amazon S3 akan disimpan sebagai metadata Amazon S3.

## Sintaks untuk tag MKV Produser

```

|+ 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

```

## Menambahkan tag metadata di Producer SDK menggunakan `PutEventMetadata`

Yang `PutEventMetadata` fungsi menambahkan file MKV yang terkait dengan suatu peristiwa. `PutEventMetadata` mengambil dua parameter. Parameter pertama adalah peristiwa yang nilainya berasal dari `STREAM_EVENT_TYPE` enum. Parameter kedua, [pStreamEventMetadata](#), bersifat opsional dan dapat digunakan untuk menyertakan metadata tambahan sebagai pasangan kunci-nilai. Ada batas lima pasangan kunci-nilai metadata yang dapat ditambahkan.

### Batas

Tabel berikut mencantumkan batasan yang terkait dengan tag metadata. Jika batas tag metadata dapat disesuaikan, Anda dapat meminta peningkatan melalui manajer akun.

Kuota	Nilai maks	Dapat Disesuaikan
Panjang awalan gambar	256	tidak
Panjang kunci metadata opsional	128	tidak
Panjang nilai metadata opsional	256	tidak
Jumlah maksimum metadata opsional	10	ya

### Metadata objek S3

Secara default, Kinesis Video Streams akan mengatur nomor fragmen, penghasil, dan stempel waktu server gambar yang dihasilkan sebagai metadata objek Amazon S3. Jika ada data fragmen tambahan

yang ditentukan dalam tag MKV, tag tersebut juga akan ditambahkan ke metadata objek Amazon S3. Contoh berikut menunjukkan sintaks yang benar untuk metadata objek 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,
}
```

## S3 objek jalan (gambar)

Daftar berikut menunjukkan format yang benar dari path objek dan menjelaskan setiap elemen dalam path.

Format:

*ImagePrefix\_AkuntID\_StreamName\_ImageTimecode\_RandomID.file-ekstensi*

1. ImagePrefix - Nilai AWS\_KINESISVIDEO\_IMAGE\_PREFIX.
2. AccountID - ID Akun di mana aliran dibuat.
3. StreamName - Nama aliran yang gambar dihasilkan.
4. ImageTimecode - Epoch timecode dalam fragmen di mana gambar dihasilkan.
5. RandomID - GUID Acak.
6. file-extension - JPG atau PNG berdasarkan format gambar yang diminta.

## Rekomendasi URI Amazon S3 untuk melindungi dari pembatasan

Jika Anda menulis ribuan gambar ke Amazon S3, ada risiko pembatasan. Untuk informasi lebih lanjut, lihat [Awalan S3 Menempatkan Batas Permintaan](#).



Awalan Amazon S3 dimulai dengan batas PUT 3.500 permintaan PUT per detik, dan secara bertahap akan meningkat seiring waktu untuk awalan unik. Hindari menggunakan tanggal dan waktu sebagai awalan Amazon S3. Data berkode waktu akan memengaruhi satu awalan pada satu waktu, dan juga akan berubah secara teratur, membatalkan peningkatan skala awalan sebelumnya. Untuk mengaktifkan penskalaan Amazon S3 yang lebih cepat dan konsisten, sebaiknya tambahkan awalan acak, seperti kode hex atau UUID ke URI Tujuan Amazon S3. Misalnya, awalan kode hex secara alami akan membagi permintaan Anda secara acak di antara 16 awalan yang berbeda (awalan untuk setiap karakter hex unik), yang akan memungkinkan 56.000 permintaan PUT per detik setelah Amazon S3 memiliki skala otomatis.

# Pemberitahuan di Kinesis Video Streams

Saat fragmen media tersedia untuk dikonsumsi, Kinesis Video Streams akan memberi tahu pelanggan yang menggunakan notifikasi ( Amazon Simple Notification Service Amazon SNS). Topik berikut menjelaskan cara memulai dengan notifikasi.

## UpdateNotificationConfiguration

Gunakan operasi API ini untuk memperbarui informasi notifikasi untuk aliran. Untuk informasi selengkapnya tentang `UpdateNotificationConfiguration` fitur ini, lihat [UpdateNotificationConfiguration](#) di Panduan Pengembang Amazon Kinesis Video Streams.

### Note

Dibutuhkan setidaknya satu menit untuk memulai notifikasi setelah memperbarui konfigurasi notifikasi. Tunggu setidaknya satu menit sebelum memanggil `PutMedia` setelah panggilan pembaruan.

## DescribeNotificationConfiguration

Gunakan API ini untuk menjelaskan konfigurasi notifikasi yang dilampirkan ke aliran. Untuk informasi selengkapnya tentang `DescribeNotificationConfiguration` fitur ini, lihat [DescribeNotificationConfiguration](#) di Panduan Pengembang Amazon Kinesis Video Streams.

## Tag MKV Produser

Anda dapat menggunakan Kinesis Video Streams Producer SDK untuk menandai fragmen tertentu yang diinginkan dengan mengekspos operasi API di SDK. Lihat contoh cara kerjanya [di bagian kode ini](#). Setelah memanggil API ini, SDK akan menambahkan satu set tag MKV yang telah ditentukan bersama dengan data fragmen. Kinesis Video Streams akan mengenali tag MKV khusus ini dan memulai notifikasi untuk fragmen yang ditandai.

Setiap metadata fragmen yang disediakan bersama dengan tag Notification MKV akan dipublikasikan sebagai bagian dari payload topik Amazon SNS.

## Sintaks untuk tag MKV Produser

```

|+ 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

```

## Batas tag MKV

Tabel berikut mencantumkan batasan yang terkait dengan tag metadata. Jika batas tag metadata dapat disesuaikan, Anda dapat meminta peningkatan melalui manajer akun Anda.

Kuota	Nilai maks	Dapat disesuaikan
Panjang tombol metadata opsional	128	Tidak
Panjang nilai metadata opsional	256	Tidak
Jumlah maksimum metadata opsional	10	Ya

## Payload topik Amazon SNS

Setiap pemberitahuan yang dimulai melalui alur kerja sebelumnya akan mengirimkan muatan topik Amazon SNS, seperti yang ditunjukkan pada contoh berikut. Contoh ini adalah pesan Amazon

SNS yang terjadi setelah mengkonsumsi data notifikasi dari antrian 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
  }
}
```

## Melihat pesan Amazon SNS Anda

Anda tidak dapat membaca pesan langsung dari topik Amazon SNS karena tidak ada API untuk melakukannya. Untuk melihat pesan, berlangganan antrean SQS ke topik SNS, atau pilih tujuan lain yang didukung [Amazon SNS](#). Namun, opsi paling efisien untuk melihat pesan adalah dengan menggunakan Amazon SQS.

## Untuk melihat pesan Amazon SNS Anda menggunakan Amazon SQS

1. Buat antrian [Amazon SQS](#).
2. Dari AWS Management Console, buka topik Amazon SNS yang ditetapkan sebagai tujuan di bawah. `NotificationConfiguration`
3. Pilih Buat Langganan, lalu pilih antrian Amazon SQS yang dibuat pada langkah pertama.
4. Jalankan `PutMedia` sesi dengan konfigurasi Notification diaktifkan dan dengan tag Notification MKV ditambahkan ke fragmen.
5. Pilih antrian Amazon SQS di konsol Amazon SQS, lalu pilih Kirim dan terima pesan untuk antrian Amazon SQS.
6. Polling untuk pesan. Perintah ini harus menampilkan semua notifikasi yang dihasilkan oleh `PutMedia` sesi. Untuk informasi tentang polling, lihat jajak pendapat [pendek dan panjang Amazon SQS](#).

# Keamanan di Amazon Kinesis Video Streams

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda akan mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Efektivitas keamanan kami diuji dan diverifikasi secara rutin oleh auditor pihak ketiga sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk Kinesis Video Streams [AWS](#), lihat [Layanan dalam Lingkup](#) berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor-faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Kinesis Video Streams. Topik berikut menunjukkan cara mengonfigurasi Kinesis Video Streams untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan mempelajari cara menggunakan AWS layanan lain yang dapat membantu Anda memantau dan mengamankan sumber daya Kinesis Video Streams Anda.

## Topik

- [Perlindungan data di Kinesis Video Streams](#)
- [Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan IAM](#)
- [Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan AWS IoT](#)
- [Memantau Aliran Video Amazon Kinesis](#)
- [Validasi Kepatuhan untuk Amazon Kinesis Video Streams](#)
- [Ketahanan dalam Aliran Video Amazon Kinesis](#)
- [Keamanan infrastruktur di Kinesis Video Streams](#)
- [Praktik keamanan terbaik untuk Kinesis Video Streams](#)

# Perlindungan data di Kinesis Video Streams

Anda dapat menggunakan enkripsi sisi server (SSE) menggunakan AWS Key Management Service kunci ( ) untuk memenuhi persyaratan manajemen data yang ketat dengan mengenkripsi data Anda saat istirahat di Amazon Kinesis Video Streams. AWS KMS

## Topik

- [Apa itu enkripsi sisi server untuk Kinesis Video Streams?](#)
- [Biaya, Wilayah, dan pertimbangan kinerja](#)
- [Bagaimana cara memulai dengan enkripsi sisi server?](#)
- [Membuat dan menggunakan kunci KMS buatan pengguna](#)
- [Izin untuk menggunakan kunci KMS buatan pengguna](#)

## Apa itu enkripsi sisi server untuk Kinesis Video Streams?

Enkripsi sisi server adalah fitur di Kinesis Video Streams yang secara otomatis mengenkripsi data sebelum diam dengan menggunakan yang Anda tentukan. AWS KMS Data dienkripsi sebelum ditulis ke lapisan penyimpanan aliran Kinesis Video Streams, dan didekripsi setelah diambil dari penyimpanan. Akibatnya, data Anda selalu dienkripsi saat diam dalam layanan Kinesis Video Streams.

Dengan enkripsi sisi server, produsen dan konsumen aliran video Kinesis Anda tidak perlu mengelola kunci KMS atau operasi kriptografi. Jika retensi data diaktifkan, data Anda secara otomatis dienkripsi saat masuk dan keluar dari Kinesis Video Streams, sehingga data Anda saat istirahat dienkripsi. AWS KMS menyediakan semua kunci yang digunakan oleh fitur enkripsi sisi server. AWS KMS merampingkan penggunaan kunci KMS untuk Kinesis Video Streams yang dikelola AWS oleh, kunci yang ditentukan pengguna yang diimpor ke AWS KMS layanan. AWS KMS

## Biaya, Wilayah, dan pertimbangan kinerja

Ketika Anda menerapkan enkripsi sisi server, Anda tunduk pada penggunaan AWS KMS API dan biaya utama. Tidak seperti AWS KMS kunci khusus, kunci (Default) `aws/kinesis-video` KMS ditawarkan tanpa biaya. Namun, Anda tetap harus membayar biaya penggunaan API yang dikeluarkan Kinesis Video Streams atas nama Anda.

Biaya penggunaan API berlaku untuk setiap kunci KMS, termasuk yang khusus. Skala AWS KMS biaya dengan jumlah kredensial pengguna yang Anda gunakan pada produsen data dan konsumen Anda karena setiap kredensial pengguna memerlukan panggilan API yang unik. AWS KMS

Berikut ini menjelaskan biaya berdasarkan sumber daya:

#### Kunci

- Kunci KMS untuk Kinesis Video Streams yang dikelola AWS oleh (alias `aws/kinesis-video` =) tidak dikenakan biaya.
- Kunci KMS buatan pengguna dikenakan biaya. AWS KMS key Untuk informasi selengkapnya, silakan lihat [Harga AWS Key Management Service](#).

## AWS KMS Penggunaan API

Permintaan API untuk menghasilkan kunci enkripsi data baru atau untuk mengambil kunci enkripsi yang ada meningkat seiring dengan peningkatan lalu lintas, dan dikenakan biaya AWS KMS penggunaan. Untuk informasi selengkapnya, lihat [AWS Key Management Service Harga: Penggunaan](#).

Kinesis Video Streams menghasilkan permintaan kunci bahkan ketika retensi disetel ke 0 (tidak ada retensi).

## Ketersediaan enkripsi sisi server menurut Wilayah

Enkripsi sisi server dari aliran video Kinesis tersedia di semua tempat Kinesis Video Wilayah AWS Streams tersedia.

## Bagaimana cara memulai dengan enkripsi sisi server?

Enkripsi sisi server selalu diaktifkan di Kinesis Video Streams. Jika kunci yang disediakan pengguna tidak ditentukan saat aliran dibuat, kunci default (disediakan oleh Kinesis Video Streams) akan digunakan.

Kunci KMS yang disediakan pengguna harus ditetapkan ke aliran video Kinesis saat dibuat. Anda tidak dapat menetapkan kunci yang berbeda ke aliran menggunakan [UpdateStream](#) API nanti.

Anda dapat menetapkan kunci KMS yang disediakan pengguna ke aliran video Kinesis dengan dua cara:



- Saat membuat aliran video Kinesis di AWS Management Console, tentukan kunci KMS di tab Enkripsi pada halaman Buat aliran video baru.
- Saat membuat aliran video Kinesis menggunakan [CreateStream](#) API, tentukan ID kunci dalam parameter. `KmsKeyId`

## Membuat dan menggunakan kunci KMS buatan pengguna

Bagian ini menjelaskan cara membuat dan menggunakan kunci KMS Anda sendiri alih-alih menggunakan kunci yang dikelola oleh Amazon Kinesis Video Streams.

### Membuat kunci KMS yang dibuat pengguna

Untuk informasi tentang cara membuat kunci Anda sendiri, lihat [Membuat Kunci](#) di Panduan AWS Key Management Service Pengembang. Setelah Anda membuat kunci untuk akun Anda, layanan Kinesis Video Streams mengembalikan kunci ini dalam daftar kunci master KMS.

### Menggunakan kunci KMS yang dibuat pengguna

Setelah izin yang benar diterapkan ke konsumen, produsen, dan administrator Anda, Anda dapat menggunakan kunci KMS kustom sendiri Akun AWS atau yang lain. Akun AWS Semua kunci KMS di akun Anda muncul di daftar Kunci Master KMS di konsol.

Untuk menggunakan kunci KMS khusus yang terletak di akun lain, Anda harus memiliki izin untuk menggunakan kunci tersebut. Anda juga harus membuat aliran menggunakan `CreateStream` API. Anda tidak dapat menggunakan kunci KMS dari akun yang berbeda dalam aliran yang dibuat di konsol.

#### Note

Kunci KMS tidak diakses sampai `GetMedia` operasi `PutMedia` atau dilakukan. Ini memiliki hasil sebagai berikut:

- Jika kunci yang Anda tentukan tidak ada, `CreateStream` operasi berhasil, tetapi `PutMedia` `GetMedia` operasi pada aliran gagal.
- Jika Anda menggunakan kunci yang disediakan (`aws/kinesis-video`), kunci tidak ada di akun Anda sampai yang pertama `PutMedia` atau `GetMedia` operasi dilakukan.

## Izin untuk menggunakan kunci KMS buatan pengguna

Sebelum Anda dapat menggunakan enkripsi sisi server dengan kunci KMS buatan pengguna, Anda harus mengonfigurasi kebijakan kunci KMS untuk memungkinkan enkripsi aliran dan enkripsi dan dekripsi catatan aliran. Untuk contoh dan informasi selengkapnya tentang AWS KMS izin, lihat [Izin AWS KMS API: Tindakan dan Referensi Sumber Daya](#).

### Note

Penggunaan kunci layanan default untuk enkripsi tidak memerlukan penerapan izin IAM khusus.

Sebelum Anda menggunakan kunci KMS buatan pengguna, verifikasi bahwa produsen dan konsumen aliran video Kinesis Anda (prinsip IAM) adalah pengguna dalam kebijakan kunci utama. AWS KMS Jika tidak, menulis dan membaca dari aliran akan gagal, yang pada akhirnya dapat mengakibatkan kehilangan data, pemrosesan tertunda, atau aplikasi yang macet. Anda dapat mengelola izin untuk kunci KMS menggunakan kebijakan IAM. Untuk informasi selengkapnya, lihat [Menggunakan Kebijakan IAM dengan AWS KMS](#).

### Contoh izin produsen

Produser streaming video Kinesis Anda harus memiliki izin: `kms:GenerateDataKey`

```
{
  "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"
```

```
}  
]  
}
```

## Contoh Izin konsumen

Konsumen streaming video Kinesis Anda harus memiliki izin: `kms:Decrypt`

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

## Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan IAM

Anda dapat menggunakan AWS Identity and Access Management (IAM) dengan Amazon Kinesis Video Streams, untuk mengontrol apakah pengguna di organisasi Anda dapat melakukan tugas menggunakan operasi API Kinesis Video Streams tertentu dan apakah mereka dapat menggunakan sumber daya tertentu. AWS

Untuk informasi selengkapnya tentang IAM, lihat berikut ini:

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

- [Memulai](#)
- [Panduan Pengguna IAM](#)

## Daftar Isi

- [Sintaksis kebijakan](#)
- [Tindakan untuk Kinesis Video Streams](#)
- [Nama Sumber Daya Amazon \(ARN\) untuk Kinesis Video Streams](#)
- [Memberikan akses akun IAM lainnya ke aliran video Kinesis](#)
- [Contoh kebijakan untuk Kinesis Video Streams](#)

## Sintaksis kebijakan

kebijakan IAM adalah dokumen JSON yang terdiri dari satu atau beberapa pernyataan. Setiap pernyataan memiliki struktur sebagai berikut:

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

Ada berbagai elemen yang membentuk pernyataan:

- Efek — Efeknya bisa Allow atau Deny. Secara default, pengguna tidak memiliki izin untuk menggunakan sumber daya dan tindakan API, sehingga semua permintaan akan ditolak. izin eksplisit akan menggantikan izin default. penolakan eksplisit akan menggantikan izin apa pun.
- Tindakan — Tindakan adalah tindakan API spesifik yang Anda berikan atau tolak izinnya.
- Sumber daya — Sumber daya yang dipengaruhi oleh tindakan. Untuk menentukan sumber daya dalam pernyataan, Anda harus menggunakan Nama Sumber Daya Amazon (ARN).

- Kondisi - Kondisi bersifat opsional. Syarat-syarat ini dapat digunakan untuk mengendalikan kapan kebijakan Anda berlaku.

Saat Anda membuat dan mengelola kebijakan IAM, kami menyarankan Anda menggunakan [IAM Policy Generator dan IAM Policy Simulator](#).

## Tindakan untuk Kinesis Video Streams

Dalam pernyataan kebijakan IAM, Anda dapat menentukan tindakan API apa pun dari layanan apa pun yang mendukung IAM. Untuk Kinesis Video Streams, gunakan awalan berikut dengan nama aksi API: `kinesisvideo:` Misalnya: `kinesisvideo:CreateStream`, `kinesisvideo:ListStreams`, dan `kinesisvideo:DescribeStream`.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut menggunakan koma seperti berikut:

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard. Misalnya, Anda dapat menentukan semua tindakan yang namanya dimulai dengan kata "Dapatkan" sebagai berikut:

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

Untuk menentukan semua operasi Kinesis Video Streams, gunakan wildcard asterisk (\*) sebagai berikut:

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

Untuk daftar lengkap tindakan API Kinesis Video Streams, [lihat referensi API Kinesis Video Streams](#).

## Nama Sumber Daya Amazon (ARN) untuk Kinesis Video Streams

Setiap pernyataan kebijakan IAM berlaku untuk sumber daya yang Anda tentukan menggunakan ARN.

Gunakan format sumber daya ARN berikut untuk Kinesis Video Streams:

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

Sebagai contoh:

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

Anda bisa mendapatkan ARN dari aliran menggunakan [DescribeStream](#)

## Memberikan akses akun IAM lainnya ke aliran video Kinesis

Anda mungkin perlu memberikan izin ke akun IAM lain untuk melakukan operasi pada aliran video Kinesis. Ikhtisar berikut menjelaskan langkah-langkah umum untuk memberikan akses ke aliran video di seluruh akun:

1. Dapatkan ID akun 12 digit dari akun yang ingin Anda berikan izin untuk melakukan operasi di streaming Anda (misalnya, 111111111111).
2. Buat kebijakan terkelola pada akun yang memiliki aliran yang memungkinkan tingkat akses yang ingin Anda berikan. Misalnya kebijakan untuk sumber daya Kinesis Video Streams [Contoh kebijakan](#), lihat di bagian berikutnya.
3. Buat peran, dan tentukan akun yang ingin Anda berikan izin. Kemudian lampirkan kebijakan yang Anda buat pada langkah sebelumnya.
4. Buat kebijakan terkelola yang memungkinkan AssumeRole tindakan pada peran yang Anda buat di langkah sebelumnya. Misalnya, peran mungkin terlihat seperti berikut:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::123456789012:role/CustomRole"
  }
}
```

Untuk step-by-step petunjuk tentang pemberian akses lintas akun, lihat [Mendelegasikan Akses di Seluruh Akun AWS Menggunakan Peran IAM](#).

## Contoh kebijakan untuk Kinesis Video Streams

Contoh kebijakan berikut menunjukkan bagaimana Anda dapat mengontrol akses pengguna ke Kinesis Video Streams

### Example 1: Izinkan pengguna mendapatkan data dari aliran video Kinesis apa pun

Kebijakan ini memungkinkan pengguna atau grup untuk melakukan `DescribeStream`, `GetDataEndpoint`, `GetMediaListStreams`, dan `ListTagsForStream` operasi pada aliran video Kinesis apa pun. Kebijakan ini sesuai untuk pengguna yang bisa mendapatkan data dari streaming video apa pun.

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

### Example 2: Izinkan pengguna membuat aliran video Kinesis dan menulis data ke dalamnya

Kebijakan ini memungkinkan pengguna atau grup untuk melakukan `CreateStream` dan `PutMedia` operasi. Kebijakan ini sesuai untuk kamera keamanan yang dapat membuat aliran video dan mengirim data ke sana.

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

### Example 3: Izinkan pengguna akses penuh ke semua sumber daya Kinesis Video Streams

Kebijakan ini memungkinkan pengguna atau grup untuk melakukan operasi Kinesis Video Streams pada sumber daya apa pun. Kebijakan ini sesuai untuk administrator.

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

### Example 4: Izinkan pengguna untuk menulis data ke aliran video Kinesis tertentu

Kebijakan ini memungkinkan pengguna atau grup untuk menulis data ke aliran video tertentu. Kebijakan ini sesuai untuk perangkat yang dapat mengirim data ke satu aliran.

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

## Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan AWS IoT

Bagian ini menjelaskan cara mengaktifkan perangkat (misalnya, kamera) untuk mengirim data audio dan video ke satu aliran video Kinesis tertentu saja. Anda dapat melakukan ini dengan menggunakan penyedia AWS IoT kredensial dan peran AWS Identity and Access Management (IAM).



Perangkat dapat menggunakan sertifikat X.509 untuk terhubung AWS IoT menggunakan protokol otentikasi timbal balik TLS. Lainnya Layanan AWS (misalnya, Kinesis Video Streams) tidak mendukung otentikasi berbasis sertifikat, tetapi dapat AWS dipanggil menggunakan kredensial dalam format Signature Version 4. AWS Algoritma Signature Version 4 biasanya mengharuskan penelepon untuk memiliki ID kunci akses dan kunci akses rahasia. AWS IoT memiliki penyedia kredensi yang memungkinkan Anda menggunakan sertifikat X.509 bawaan sebagai identitas perangkat unik untuk mengautentikasi AWS permintaan (misalnya, permintaan ke Kinesis Video Streams). Ini menghilangkan kebutuhan untuk menyimpan ID kunci akses dan kunci akses rahasia di perangkat Anda.

Penyedia kredensi mengautentikasi klien (dalam hal ini, SDK Kinesis Video Streams yang berjalan di kamera yang ingin Anda kirim data ke aliran video) menggunakan sertifikat X.509 dan mengeluarkan token keamanan hak istimewa terbatas sementara. Anda dapat menggunakan token untuk menandatangani dan mengautentikasi AWS permintaan apa pun (dalam hal ini, panggilan ke Kinesis Video Streams). Untuk informasi selengkapnya, lihat [Mengotorisasi Panggilan Langsung ke AWS Layanan](#).

Cara mengautentikasi permintaan kamera Anda ke Kinesis Video Streams ini mengharuskan Anda membuat dan mengonfigurasi peran IAM dan melampirkan kebijakan IAM yang sesuai ke peran tersebut sehingga penyedia kredensi dapat mengambil peran AWS IoT tersebut atas nama Anda.

Untuk informasi selengkapnya AWS IoT, lihat [AWS IoT Core Dokumentasi](#). Untuk informasi selengkapnya tentang tag IAM, lihat [AWS Identity and Access Management \(IAM\)](#).

Topik

- [AWS IoT ThingName sebagai nama aliran](#)
- [AWS IoT CertificateId sebagai nama aliran](#)
- [Gunakan AWS IoT kredensi untuk melakukan streaming ke nama aliran hard-coded](#)

## AWS IoT ThingName sebagai nama aliran

Topik

- [Langkah 1: Buat tipe AWS IoT benda dan AWS IoT benda](#)
- [Langkah 2: Buat peran IAM untuk diasumsikan oleh AWS IoT](#)
- [Langkah 3: Buat dan konfigurasi sertifikat X.509](#)
- [Langkah 4: Uji AWS IoT kredensialnya dengan aliran video Kinesis Anda](#)

- [Langkah 5: Menyebarkan AWS IoT sertifikat dan kredensyal pada sistem file kamera Anda dan streaming data ke aliran video Anda](#)

## Langkah 1: Buat tipe AWS IoT benda dan AWS IoT benda

Dalam AWS IoT, sesuatu adalah representasi dari perangkat tertentu atau entitas logis. Dalam hal ini, AWS IoT sesuatu mewakili aliran video Kinesis Anda yang ingin Anda konfigurasi kontrol akses tingkat sumber daya. Untuk membuat sesuatu, pertama, Anda harus membuat tipe AWS IoT benda. Anda dapat menggunakan tipe AWS IoT benda untuk menyimpan deskripsi dan informasi konfigurasi yang umum untuk semua hal yang terkait dengan jenis hal yang sama.

1. Contoh perintah berikut menciptakan tipe benda `kvs_example_camera`:

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

2. Perintah contoh ini menciptakan `kvs_example_camera_stream` hal dari tipe `kvs_example_camera` benda:

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

## Langkah 2: Buat peran IAM untuk diasumsikan oleh AWS IoT

Peran IAM mirip dengan pengguna, karena peran adalah AWS identitas dengan kebijakan izin yang menentukan apa yang dapat dan tidak dapat dilakukan identitas. AWS Peran dapat diasumsikan oleh siapa saja yang membutuhkannya. Saat Anda mengambil peran, itu memberi Anda kredensyal keamanan sementara untuk sesi peran Anda.

Peran yang Anda buat dalam langkah ini dapat diasumsikan oleh AWS IoT untuk mendapatkan kredensil sementara dari layanan token keamanan (STS) saat melakukan permintaan otorisasi kredensi dari klien. Dalam hal ini, klien adalah Kinesis Video Streams SDK yang berjalan di kamera Anda.

Lakukan langkah-langkah berikut untuk membuat dan mengonfigurasi peran IAM ini:

1. Buat peran IAM.

Contoh perintah berikut menciptakan peran IAM yang disebut `KVSCameraCertificateBasedIAMRole`:

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

Anda dapat menggunakan kebijakan kepercayaan berikut JSON untuk: `iam-policy-document.json`

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

2. Selanjutnya, lampirkan kebijakan izin ke peran IAM yang sebelumnya Anda buat. Kebijakan izin ini memungkinkan kontrol akses selektif (subset operasi yang didukung) untuk sumber daya. AWS Dalam hal ini, AWS sumber daya adalah aliran video yang Anda ingin kamera Anda kirim data. Dengan kata lain, setelah semua langkah konfigurasi selesai, kamera ini akan dapat mengirim data hanya ke aliran video ini.

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

Anda dapat menggunakan kebijakan IAM berikut JSON untuk: `iam-permission-document`

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

```

        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:ThingName}/*"
    }
]
}

```

Perhatikan bahwa kebijakan ini mengotorisasi tindakan yang ditentukan hanya pada aliran video (AWS resource) yang ditentukan oleh placeholder (`${credentials-iot:}`). ThingName Placeholder ini mengambil nilai atribut AWS IoT thing ThingName saat penyedia AWS IoT kredensial mengirimkan nama aliran video dalam permintaan.

3. Selanjutnya, buat Alias Peran untuk peran IAM Anda. Alias peran adalah model data alternatif yang menunjuk ke peran IAM. Permintaan penyedia AWS IoT kredensial harus menyertakan alias peran untuk menunjukkan peran IAM mana yang harus diambil untuk mendapatkan kredensial sementara dari STS.

Perintah contoh berikut menciptakan alias peran yang disebut `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. Sekarang Anda dapat membuat kebijakan yang akan memungkinkan AWS IoT untuk mengambil peran dengan sertifikat (setelah dilampirkan) menggunakan alias peran.

Perintah contoh berikut membuat kebijakan untuk AWS IoT dipanggil `KvsCameraIoTPolicy`.

```

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

```

Anda dapat menggunakan perintah berikut untuk membuat dokumen JSON `iot-policy-document.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

```

### Langkah 3: Buat dan konfigurasi sertifikat X.509

Komunikasi antara perangkat (aliran video Anda) dan AWS IoT dilindungi melalui penggunaan sertifikat X.509.

1. Buat sertifikat yang harus Anda lampirkan kebijakan untuk AWS IoT yang sebelumnya Anda buat.

```

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. Lampirkan kebijakan untuk AWS IoT (KvsCameraIoTPolicy dibuat sebelumnya) ke sertifikat ini.

```

aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
$(jq --raw-output '.certificateArn' certificate)

```

3. Lampirkan AWS IoT thing (kvs\_example\_camera\_stream) Anda ke sertifikat yang baru saja Anda buat:

```

aws --profile default iot attach-thing-principal --thing-name
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
certificate)

```

4. Untuk mengotorisasi permintaan melalui penyedia AWS IoT kredensial, Anda memerlukan titik akhir AWS IoT kredensial, yang unik untuk ID Anda. Akun AWS Anda dapat menggunakan perintah berikut untuk mendapatkan titik akhir AWS IoT kredensial.

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
--output text > iot-credential-provider.txt
```

5. Selain sertifikat X.509 yang dibuat sebelumnya, Anda juga harus memiliki sertifikat CA untuk membangun kepercayaan dengan layanan backend melalui TLS. Anda bisa mendapatkan sertifikat CA menggunakan perintah berikut:

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output
cacert.pem
```

## Langkah 4: Uji AWS IoT kredensialnya dengan aliran video Kinesis Anda

Sekarang Anda dapat menguji AWS IoT kredensial yang telah Anda atur sejauh ini.

1. Pertama, buat aliran video Kinesis yang ingin Anda uji konfigurasi ini.

### Important

Buat streaming video dengan nama yang identik dengan nama AWS IoT benda yang Anda buat di langkah sebelumnya (`kvs_example_camera_stream`).

```
aws kinesismedia create-stream --data-retention-in-hours 24 --stream-name
kvs_example_camera_stream
```

2. Selanjutnya, hubungi penyedia AWS IoT kredensial untuk mendapatkan kredensial sementara:

```
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

Anda dapat menggunakan perintah berikut untuk mendapatkan `IOT_GET_CREDENTIAL_ENDPOINT`:

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

Output JSON berisi AccessKey, SecretKey, dan SessionToken, yang dapat Anda gunakan untuk mengakses Kinesis Video Streams.

- Untuk pengujian, Anda dapat menggunakan kredensial ini untuk menjalankan API Kinesis Video DescribeStream Streams untuk aliran video sampel. `kvs_example_camera_stream`

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json)
aws kinesisvideo describe-stream --stream-name kvs_example_camera_stream
```

## Langkah 5: Menyebarkan AWS IoT sertifikat dan kredensial pada sistem file kamera Anda dan streaming data ke aliran video Anda

### Note

Langkah-langkah di bagian ini menjelaskan pengiriman media ke aliran video Kinesis dari kamera yang menggunakan [the section called “Pustaka Produser C++”](#)

- Salin sertifikat X.509, kunci pribadi, dan sertifikat CA yang dihasilkan pada langkah sebelumnya ke sistem file kamera Anda. Tentukan jalur tempat file ini disimpan, nama alias peran, dan titik akhir AWS IoT kredensial untuk menjalankan `gst-launch-1.0` perintah atau aplikasi sampel Anda.
- Contoh perintah berikut menggunakan otorisasi AWS IoT sertifikat untuk mengirim video ke 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 sebagai nama aliran

Untuk mewakili perangkat Anda (misalnya, kamera Anda) melalui AWS IoT sesuatu, tetapi mengotorisasi nama streaming yang berbeda, maka Anda dapat menggunakan AWS IoT `certificateId` atribut sebagai nama streaming dan memberikan izin Kinesis Video Streams pada streaming menggunakan. AWS IoT Langkah-langkah untuk mencapai ini mirip dengan yang diuraikan sebelumnya, dengan beberapa perubahan.

- Ubah kebijakan izin ke peran IAM Anda (`iam-permission-document.json`) sebagai berikut:

```
{
  "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

Sumber daya ARN menggunakan ID sertifikat sebagai pengganti untuk nama aliran. Izin IAM akan berfungsi saat Anda menggunakan ID sertifikat sebagai nama aliran. Dapatkan ID sertifikat dari sertifikat sehingga Anda dapat menggunakannya sebagai nama aliran dalam mendeskripsikan panggilan API aliran berikut.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Verifikasi perubahan ini menggunakan perintah CLI Describe-stream 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}
```

- Meneruskan CertificateId ke AWS IoT penyedia kredensial dalam [contoh aplikasi di Kinesis Video Streams](#) C++ SDK:

```
credential_provider =
    make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
        cert_path,
        private_key_path,
        role_alias,
        ca_cert_path,
        certificateId);
```

### Note

Perhatikan bahwa Anda meneruskan thingname ke penyedia AWS IoT kredensi. Anda dapat menggunakan getenv untuk meneruskan thingname ke aplikasi demo yang mirip dengan meneruskan atribut lainnya. AWS IoT Gunakan ID sertifikat sebagai nama aliran dalam parameter baris perintah saat Anda menjalankan aplikasi sampel.

## Gunakan AWS IoT kredensi untuk melakukan streaming ke nama aliran hard-coded

Untuk mewakili perangkat Anda (misalnya, kamera Anda) melalui AWS IoT sesuatu, tetapi mengotorisasi streaming ke aliran video Amazon Kinesis tertentu, berikan izin Amazon Kinesis Video Streams pada streaming menggunakan. AWS IoT Prosesnya mirip dengan bagian sebelumnya, dengan beberapa perubahan.

Ubah kebijakan izin ke peran IAM Anda (`iam-permission-document.json`) sebagai berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
```

```
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
}
]
```

Salin sertifikat X.509, kunci pribadi, dan sertifikat CA yang dihasilkan pada langkah sebelumnya ke sistem file kamera Anda.

Tentukan jalur tempat file ini disimpan, nama alias peran, nama AWS IoT benda, dan titik akhir AWS IoT kredensial untuk menjalankan `gst-launch-1.0` perintah atau aplikasi sampel Anda.

Perintah contoh berikut menggunakan otorisasi AWS IoT sertifikat untuk mengirim video ke 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"
```

## Memantau Aliran Video Amazon Kinesis

Kinesis Video Streams menyediakan fungsionalitas pemantauan untuk aliran pengiriman Anda. Untuk informasi selengkapnya, lihat [Memantau](#).


## Validasi Kepatuhan untuk Amazon Kinesis Video Streams

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

## Ketahanan dalam Aliran Video Amazon Kinesis

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pinda saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripada infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

## Keamanan infrastruktur di Kinesis Video Streams

Sebagai layanan terkelola, Amazon Kinesis Video Streams dilindungi AWS oleh prosedur keamanan jaringan global yang dijelaskan [dalam whitepaper Amazon Web Services: Tinjauan Proses Keamanan](#).

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Kinesis Video Streams melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.2 atau versi yang lebih baru. Klien juga harus support suite cipher dengan Perfect Forward Secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern, misalnya Java 7 dan versi yang lebih baru, mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan utama IAM. Selain itu, Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara untuk menandatangani permintaan.

## Praktik keamanan terbaik untuk Kinesis Video Streams

Amazon Kinesis Video Streams menyediakan sejumlah fitur keamanan yang perlu dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

Untuk praktik terbaik keamanan untuk perangkat jarak jauh, lihat [Praktik Terbaik Keamanan untuk Agen Perangkat](#).

## Terapkan akses hak akses paling rendah

Saat memberikan izin, Anda memutuskan siapa yang mendapatkan izin apa untuk sumber daya Kinesis Video Streams mana. Anda memungkinkan tindakan tertentu yang ingin Anda lakukan di sumber daya tersebut. Oleh karena itu, Anda harus memberikan hanya izin yang diperlukan untuk melaksanakan tugas. Menerapkan akses hak istimewa yang terkecil adalah hal mendasar dalam mengurangi risiko keamanan dan dampak yang dapat diakibatkan oleh kesalahan atau niat jahat.

Misalnya, produser yang mengirimkan data ke Kinesis Video Streams `GetStreamingEndpoint` hanya `PutMedia` membutuhkan,, dan. `DescribeStream` Jangan berikan izin aplikasi produser untuk semua tindakan (\*), atau untuk tindakan lain seperti `GetMedia`.

Untuk informasi lebih lanjut, lihat [Apa itu Keistimewaan Terkecil & Mengapa Anda Membutuhkannya?](#)

## Gunakan IAM role

Aplikasi produser dan klien harus memiliki kredensial yang valid untuk mengakses Kinesis Video Streams. Anda tidak boleh menyimpan kredensial AWS secara langsung di aplikasi klien atau bucket Amazon S3. Ini adalah kredensyal jangka panjang yang tidak diputar secara otomatis dan dapat memiliki dampak bisnis yang signifikan jika dikompromikan.

Sebagai gantinya, Anda harus menggunakan peran IAM untuk mengelola kredensyal sementara untuk aplikasi produser dan klien Anda untuk mengakses Kinesis Video Streams. Saat Anda menggunakan peran, Anda tidak perlu menggunakan kredensyal jangka panjang (seperti nama pengguna dan kata sandi atau kunci akses) untuk mengakses sumber daya lain.

Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna IAM:

- [Peran IAM](#)
- [Skenario Umum untuk Peran: Pengguna, Aplikasi, dan Layanan](#)

## Gunakan CloudTrail untuk memantau panggilan API

Kinesis Video Streams AWS CloudTrail berfungsi dengan, layanan yang menyediakan rekaman tindakan yang diambil oleh pengguna, peran, Layanan AWS atau dalam Kinesis Video Streams.

Anda dapat menggunakan informasi yang dikumpulkan oleh CloudTrail untuk menentukan permintaan yang dibuat untuk Kinesis Video Streams, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan rincian tambahan.

Untuk informasi selengkapnya, lihat [the section called “Pencatatan Panggilan API dengan CloudTrail”](#).

# Perpustakaan Produser Aliran Video Kinesis

Library Produser Amazon Kinesis Video Streams adalah sekumpulan pustaka di SDK Produser Kinesis Video Streams. Klien menggunakan pustaka dan SDK untuk membangun aplikasi di perangkat agar terhubung dengan aman ke Kinesis Video Streams, dan streaming data media untuk dilihat di konsol atau aplikasi klien secara real time.

Data media dapat dialirkan dengan cara berikut:

- Secara real time
- Setelah buffering selama beberapa detik
- Setelah media diunggah

Setelah Anda membuat aliran Kinesis Video Streams, Anda dapat mulai mengirim data ke sana. Anda dapat menggunakan SDK untuk membuat kode aplikasi yang mengekstrak data video, yang dikenal sebagai frame, dari sumber media dan mengunggahnya ke Kinesis Video Streams. Aplikasi ini juga disebut sebagaipenghasilaplikasi.

Pustaka Produser berisi komponen-komponen berikut:

- [Klien Produser Aliran Video Kinesis](#)
- [Perpustakaan Produser Aliran Video Kinesis](#)

## Klien Produser Aliran Video Kinesis

Klien Produser Aliran Video Kinesis mencakup satu `KinesisVideoClient` kelas.

Kelas ini mengelola sumber media, menerima data dari sumber, dan mengelola siklus hidup stream saat data mengalir dari sumber media ke Kinesis Video Streams. Ini juga menyediakan `MediaSource` antarmuka untuk menentukan interaksi antara Kinesis Video Streams dan perangkat keras dan perangkat lunak milik Anda.

Sumber media bisa hampir apa saja. Misalnya, Anda dapat menggunakan sumber media kamera atau sumber media mikrofon. Sumber media tidak terbatas pada sumber audio dan video saja. Misalnya, data log mungkin file teks, tetapi mereka masih dapat dikirim sebagai aliran data. Anda juga dapat memiliki beberapa kamera di ponsel Anda yang mengalirkan data secara bersamaan.

Untuk mendapatkan data dari sumber-sumber ini, Anda dapat menerapkan `MediaSource` antarmuka. Antarmuka ini memungkinkan skenario tambahan yang kami tidak menyediakan dukungan bawaan. Misalnya, Anda dapat memilih untuk mengirim yang berikut ini ke Kinesis Video Streams:

- Aliran data diagnostik (misalnya, log aplikasi dan peristiwa)
- Data dari kamera inframerah, radar, atau kamera kedalaman

Kinesis Video Streams tidak menyediakan implementasi bawaan untuk perangkat penghasil media seperti kamera. Untuk mengekstrak data dari perangkat ini, Anda harus menerapkan kode, sehingga membuat implementasi sumber media kustom Anda sendiri. Anda kemudian dapat secara eksplisit mendaftarkan sumber media kustom Anda dengan `KinesisVideoClient`, yang mengunggah data ke Kinesis Video Streams.

Klien Produser Aliran Video Kinesis tersedia untuk aplikasi Java dan Android. Untuk informasi selengkapnya, lihat [Menggunakan Java Producer Library](#) dan [Menggunakan Android Producer Library](#).

## Perpustakaan Produser Aliran Video Kinesis

Kinesis Video Streams Produser Library terkandung dalam Kinesis Video Streams Produser Client. Perpustakaan ini juga tersedia untuk digunakan secara langsung bagi mereka yang menginginkan integrasi yang lebih dalam dengan Kinesis Video Streams. Ini memungkinkan integrasi dari perangkat dengan sistem operasi berpemilik, tumpukan jaringan, atau sumber daya terbatas pada perangkat.

Kinesis Video Streams Produser Library mengimplementasikan mesin status untuk streaming ke Kinesis Video Streams. Ini menyediakan kait callback, yang mengharuskan Anda memberikan implementasi transportasi Anda sendiri dan secara eksplisit menangani setiap pesan yang pergi ke dan dari layanan.

Anda dapat memilih untuk menggunakan Kinesis Video Streams Produser Library secara langsung karena alasan berikut:

- Perangkat tempat Anda ingin menjalankan aplikasi tidak memiliki mesin virtual Java.
- Anda ingin menulis kode aplikasi dalam bahasa selain Java.
- Anda ingin mengurangi jumlah overhead dalam kode Anda dan membatasi ke tingkat minimum abstraksi telanjang, karena keterbatasan seperti memori dan kekuatan pemrosesan.



Saat ini, Kinesis Video Streams Produser Library tersedia untuk aplikasi Android, C, C ++ dan Java. Untuk informasi selengkapnya, lihat bahasa yang didukung di bawah ini [Topik Terkait](#).

## Topik terkait

[Menggunakan Java Producer Library](#)

[Menggunakan Android Producer Library](#)

[Menggunakan Pustaka Produser C ++](#)

[Menggunakan Pustaka Produser C](#)

[Menggunakan C++ Producer SDK pada Raspberry Pi](#)

## Menggunakan Java Producer Library

Anda dapat menggunakan Amazon Kinesis Video Streams yang disediakan Java Producer Library untuk menulis kode aplikasi dengan konfigurasi minimal, untuk mengirim data media dari perangkat ke aliran video Kinesis.

Lakukan langkah-langkah berikut untuk mengintegrasikan kode Anda dengan Kinesis Video Streams sehingga aplikasi Anda dapat memulai streaming data ke aliran video Kinesis Anda:

1. Buat sebuah instance dari `KinesisVideoClient` objek.
2. Buat `MediaSource` objek dengan memberikan informasi sumber media. Misalnya, saat membuat sumber media kamera, Anda memberikan informasi seperti mengidentifikasi kamera dan menentukan pengkodean yang digunakan kamera.

Saat Anda ingin memulai streaming, Anda harus membuat sumber media khusus.

3. Daftarkan sumber media dengan `KinesisVideoClient`.

Setelah Anda mendaftarkan sumber media dengan `KinesisVideoClient`, setiap kali data tersedia dengan sumber media, ia memanggil `KinesisVideoClient` dengan data.

## Prosedur: Menggunakan Java Producer SDK

Prosedur ini menunjukkan cara menggunakan Kinesis Video Streams Java Producer Client di aplikasi Java Anda untuk mengirim data ke aliran video Kinesis Anda.

Langkah-langkah ini tidak mengharuskan Anda untuk memiliki sumber media, seperti kamera atau mikrofon. Sebagai gantinya, untuk tujuan pengujian, kode menghasilkan bingkai sampel yang terdiri dari serangkaian byte. Anda dapat menggunakan pola pengkodean yang sama ketika Anda mengirim data media dari sumber nyata seperti kamera dan mikrofon.

Prosedur ini mencakup langkah-langkah berikut:

- [Unduh dan Konfigurasi Kode](#)
- [Menulis dan Memeriksa Kode](#)
- [Jalankan dan Verifikasi Kode](#)

## Prasyarat

- Dalam kode contoh, Anda memberikan kredensial dengan menentukan profil yang Anda atur dalam file profil AWS kredensial Anda. Jika Anda belum melakukannya, pertama-tama siapkan profil kredensial Anda. Untuk informasi selengkapnya, lihat [Mengatur AWS Kredensial dan Wilayah untuk Pengembangan](#) di AWS SDK for Java

### Note

Contoh Java menggunakan `SystemPropertiesCredentialsProvider` objek untuk mendapatkan kredensial Anda. Penyedia mengambil kredensial ini dari properti sistem `aws.accessKeyId` dan `aws.secretKey` Java. Anda mengatur properti sistem ini di lingkungan pengembangan Java Anda. Untuk informasi tentang cara menyetel properti sistem Java, lihat dokumentasi untuk lingkungan pengembangan terintegrasi (IDE) khusus Anda.

- Anda `NativeLibraryPath` harus berisi `KinesisVideoProducerJNI` file Anda, tersedia di <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>. Ekstensi nama file untuk file ini tergantung pada sistem operasi Anda:
  - `KinesisVideoProducerJNI.so` untuk Linux
  - `KinesisVideoProducerJNI.dylib` untuk macOS
  - `KinesisVideoProducerJNI.dll` untuk Windows

**Note**

Pustaka bawaan untuk macOS, Ubuntu, Windows, dan Raspbian tersedia di `src/main/resources/lib` <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java> Untuk lingkungan lain, kompilasi. [Pustaka Produser C++](#)

## Langkah 1: Unduh dan konfigurasi kode Java Producer Library

Di bagian prosedur Java Producer Library ini, Anda mengunduh kode contoh Java, mengimpor proyek ke IDE Java Anda, dan mengonfigurasi lokasi perpustakaan.

Untuk prasyarat dan detail lainnya tentang contoh ini, lihat [Menggunakan Java Producer Library](#).

1. Buat direktori, lalu kloning kode sumber contoh dari GitHub repositori.

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

2. Buka lingkungan pengembangan terintegrasi Java (IDE) yang Anda gunakan (misalnya, [Eclipse](#) atau [JetBrains IntelliJ IDEA](#)), dan impor proyek Apache Maven yang Anda unduh:
  - Di IntelliJ IDEA: Pilih Impor. Arahkan ke `pom.xml` file di root paket yang diunduh.
  - Di Eclipse: Pilih File, Impor, Maven, Proyek Maven yang Ada. Kemudian arahkan ke `kinesis-video-java-demo` direktori.

Untuk informasi selengkapnya, lihat dokumentasi untuk IDE Anda.

3. Kode contoh Java menggunakan AWS kredensial saat ini. Untuk menggunakan profil kredensial yang berbeda, cari kode berikut di: `DemoAppMain.java`

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

Ubah kode menjadi berikut:

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

Untuk informasi selengkapnya, lihat [ProfileCredentialsProvider](#) di referensi AWS SDK for Java.

## Langkah Selanjutnya

[the section called “Langkah 2: Tulis dan periksa kodenya”](#)

## Langkah 2: Tulis dan periksa kodenya

Di bagian [prosedur Java Producer Library ini](#), Anda menulis dan memeriksa kode contoh Java yang Anda unduh di bagian sebelumnya.

Aplikasi uji Java ([DemoAppMain](#)) menunjukkan pola pengkodean berikut:

- Buat instans `KinesisVideoClient`.
- Buat instans `MediaSource`.
- Daftarkan `MediaSource` dengan klien.
- Mulai streaming. Mulai `MediaSource` dan mulai mengirim data ke klien.

Bagian berikut memberikan perincian.

### Membuat sebuah instance dari `KinesisVideoClient`

Anda membuat `KinesisVideoClient` objek dengan memanggil `createKinesisVideoClient` operasi.

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

KinesisVideoClientUntuk melakukan panggilan jaringan, diperlukan kredensial untuk mengautentikasi. Anda meneruskan instanceSystemPropertiesCredentialsProvider, yang berbunyi AWSCredentials untuk profil default di file kredensial:

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

## Membuat sebuah instance dari MediaSource

Untuk mengirim byte ke aliran video Kinesis Anda, Anda harus menghasilkan data. Amazon Kinesis Video Streams MediaSource menyediakan antarmuka, yang mewakili sumber data.

Misalnya, perpustakaan Kinesis Video Streams Java ImageFileMediaSource menyediakan MediaSource implementasi antarmuka. Kelas ini hanya membaca data dari serangkaian file media daripada aliran video Kinesis, tetapi Anda dapat menggunakannya untuk menguji kode.

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

## Mendaftarkan MediaSource dengan klien

Daftarkan sumber media yang Anda buat dengan KinesisVideoClient sehingga tahu tentang klien (dan kemudian dapat mengirim data ke klien).

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

## Memulai sumber media

Mulai sumber media sehingga dapat mulai menghasilkan data dan mengirimkannya ke klien.

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

## Langkah selanjutnya

[the section called “Langkah 3: Jalankan dan verifikasi kode”](#)

## Langkah 3: Jalankan dan verifikasi kode

Untuk menjalankan Java test harness untuk [Java Producer Library](#), lakukan hal berikut.

1. Pilih DemoAppMain.
2. Pilih Jalankan, Jalankan DemoAppMain ".
3. Tambahkan kredensial Anda ke argumen JVM untuk aplikasi:
  - Untuk AWS kredensi non-sementara: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
  - Untuk AWS kredensi sementara: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"
4. Masuk ke AWS Management Console dan buka konsol [Kinesis Video Streams](#).

Pada halaman Kelola Aliran, pilih aliran Anda.

5. Video sampel akan diputar di pemutar yang disematkan. Anda mungkin perlu menunggu waktu yang singkat (hingga sepuluh detik di bawah kondisi bandwidth dan prosesor biasa) sementara frame menumpuk sebelum video muncul.

Contoh kode membuat aliran. Saat `MediaSource` dalam kode dimulai, ia mulai mengirim bingkai sampel ke `fileKinesisVideoClient`. Klien kemudian mengirimkan data ke aliran video Kinesis Anda.

## Menggunakan Android Producer Library

Anda dapat menggunakan Amazon Kinesis Video Streams yang disediakan Android Producer Library untuk menulis kode aplikasi, dengan konfigurasi minimal, untuk mengirim data media dari perangkat Android ke aliran video Kinesis.

Lakukan langkah-langkah berikut untuk mengintegrasikan kode Anda dengan Kinesis Video Streams sehingga aplikasi Anda dapat memulai streaming data ke aliran video Kinesis Anda:

1. Buat sebuah instance dari `KinesisVideoClient` objek.

2. Buat `MediaSource` objek dengan memberikan informasi sumber media. Misalnya, saat membuat sumber media kamera, Anda memberikan informasi seperti mengidentifikasi kamera dan menentukan pengkodean yang digunakan kamera.

Saat Anda ingin memulai streaming, Anda harus membuat sumber media khusus.

## Prosedur: Menggunakan Android Producer SDK

Prosedur ini menunjukkan cara menggunakan Kinesis Video Streams Android Producer Client di aplikasi Android Anda untuk mengirim data ke aliran video Kinesis Anda.

Prosedur ini mencakup langkah-langkah berikut:

- [the section called “Prasyarat”](#)
- [the section called “Langkah 1: Unduh dan Konfigurasi Kode”](#)
- [the section called “Langkah 2: Periksa kodenya”](#)
- [the section called “Langkah 3: Jalankan dan verifikasi kode”](#)

## Prasyarat

- Kami merekomendasikan [Android Studio](#) untuk memeriksa, mengedit, dan menjalankan kode aplikasi. Kami merekomendasikan menggunakan versi stabil terbaru.
- Dalam kode contoh, Anda memberikan kredensi Amazon Cognito.

Ikuti prosedur ini untuk menyiapkan kumpulan pengguna Amazon Cognito dan kumpulan identitas.

- [Siapkan kumpulan pengguna](#)
- [Siapkan kolam identitas](#)

## Siapkan kumpulan pengguna

Untuk mengatur kumpulan pengguna

1. Masuk ke [konsol Amazon Cognito](#) dan verifikasi bahwa wilayahnya benar.
2. Di navigasi di sebelah kiri pilih Kumpulan pengguna.
3. Di bagian User pool, pilih Create user pool.

#### 4. Lengkapi bagian berikut:

- a. Langkah 1: Konfigurasi pengalaman masuk - Di bagian opsi masuk kumpulan pengguna Cognito, pilih opsi yang sesuai.

Pilih Selanjutnya.

- b. Langkah 2: Konfigurasi persyaratan keamanan - Pilih opsi yang sesuai.

Pilih Selanjutnya.

- c. Langkah 3: Konfigurasi pengalaman pendaftaran - Pilih opsi yang sesuai.

Pilih Selanjutnya.

- d. Langkah 4: Konfigurasi pengiriman pesan - Pilih opsi yang sesuai.

Di bidang pemilihan peran IAM, pilih peran yang ada atau buat peran baru.

Pilih Selanjutnya.

- e. Langkah 5: Integrasikan aplikasi Anda - Pilih opsi yang sesuai.

Di bidang Klien aplikasi awal, pilih Klien rahasia.

Pilih Selanjutnya.

- f. Langkah 6: Tinjau dan buat - Tinjau pilihan Anda dari bagian sebelumnya, lalu pilih Buat kumpulan pengguna.

#### 5. Pada halaman User pool, pilih pool yang baru saja Anda buat.

Salin ID kumpulan Pengguna dan catat ini untuk nanti. Dalam `awsconfiguration.json` file, ini adalah `CognitoUserPool.Default.PoolId`.

#### 6. Pilih tab Integrasi aplikasi dan pergi ke bagian bawah halaman.

#### 7. Di bagian Daftar klien Aplikasi, pilih nama klien Aplikasi yang baru saja Anda buat.

Salin ID Klien dan catat ini untuk nanti. Dalam `awsconfiguration.json` file, ini adalah `CognitoUserPool.Default.AppClientId`.

#### 8. Tampilkan rahasia Klien dan catat ini untuk nanti. Dalam `awsconfiguration.json` file, ini adalah `CognitoUserPool.Default.AppClientSecret`.



## Siapkan kolam identitas


Untuk mengatur kumpulan identitas

1. Masuk ke [konsol Amazon Cognito](#) dan verifikasi bahwa wilayahnya benar.
2. Di navigasi di sebelah kiri pilih Identity pool.
3. Pilih Buat kumpulan identitas.
4. Konfigurasi kumpulan identitas.
  - a. Langkah 1: Konfigurasi kepercayaan kumpulan identitas - Lengkapi bagian berikut:
    - Akses pengguna - Pilih Akses Terotentikasi
    - Sumber identitas yang diautentikasi - Pilih kumpulan pengguna Amazon CognitoPilih Selanjutnya.
  - b. Langkah 2: Konfigurasi izin - Di bagian peran yang diautentikasi, lengkapi bidang berikut:
    - Peran IAM - Pilih Buat peran IAM baru
    - Nama peran IAM - Masukkan nama dan catat untuk langkah selanjutnya.Pilih Selanjutnya.
  - c. Langkah 3: Hubungkan penyedia identitas - Di bagian Rincian kumpulan pengguna, lengkapi bidang berikut:
    - ID kumpulan pengguna - Pilih kumpulan pengguna yang Anda buat sebelumnya.
    - ID klien aplikasi - Pilih ID klien aplikasi yang Anda buat sebelumnya.Pilih Selanjutnya.
  - d. Langkah 4: Konfigurasi properti - Ketik nama di bidang Identity pool name.  
  
Pilih Selanjutnya.
  - e. Langkah 5: Tinjau dan buat - Tinjau pilihan Anda di setiap bagian, lalu pilih Buat kumpulan identitas.
5. Pada halaman Identity pool, pilih kumpulan identitas baru Anda.

Salin ID kumpulan Identitas dan catat ini untuk nanti. Dalam `awsconfiguration.json` file, ini adalah `CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. Perbarui izin untuk peran IAM.

- a. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
- b. Di navigasi di sebelah kiri, pilih Peran.
- c. Temukan dan pilih peran yang Anda buat di atas.

 Note

Gunakan bilah pencarian, jika perlu.

- d. Pilih kebijakan izin terlampir.

Pilih Edit.

- e. Pilih tab JSON dan ganti kebijakan dengan yang berikut:

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

Pilih Selanjutnya.

- f. Pilih kotak di samping Setel versi baru ini sebagai default jika belum dipilih.

Pilih Simpan perubahan.

## Langkah 1: Unduh dan konfigurasi kode Android Producer Library

Di bagian prosedur Android Producer Library ini, Anda mengunduh kode contoh Android dan membuka project di Android Studio.

Untuk prasyarat dan detail lainnya tentang contoh ini, lihat [Menggunakan Android Producer Library](#).

1. Buat direktori, lalu kloning AWS Mobile SDK for Android dari GitHub repositori.

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

2. Buka [Android Studio](#).
3. Di layar pembuka, pilih Buka project Android Studio yang sudah ada.
4. Arahkan ke `aws-sdk-android-samples/AmazonKinesisVideoDemoApp` direktori, dan pilih OK.
5. Buka file `AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json`.

Di `CredentialsProvider` node, berikan ID kumpulan identitas dari prosedur Untuk menyiapkan kumpulan identitas di bagian [Prasyarat](#), dan berikan Wilayah AWS (misalnya,). **us-west-2**

Di `CognitoUserPool` node, berikan rahasia klien App, ID klien App, dan ID Pool dari prosedur Untuk menyiapkan kumpulan pengguna di bagian [Prasyarat](#), dan berikan Wilayah AWS (misalnya,). **us-west-2**

6. `awsconfiguration.json` file Anda akan terlihat mirip dengan yang berikut ini:

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

```
"CognitoUserPool": {
  "Default": {
    "AppClientSecret": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnop",
    "AppClientId": "0123456789abcdefghijklmnop",
    "PoolId": "us-west-2_qRsTuVwXy",
    "Region": "us-west-2"
  }
}
```

7. Perbarui `AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java` dengan Wilayah Anda (dalam contoh berikut, disetel ke `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;
```

Untuk informasi tentang Wilayah AWS konstanta, lihat [Wilayah](#).

## Langkah selanjutnya

[the section called “Langkah 2: Periksa kodenya”](#)

## Langkah 2: Periksa kodenya

Di bagian [prosedur Android Producer Library ini](#), Anda memeriksa kode contoh.

Aplikasi pengujian Android (`AmazonKinesisVideoDemoApp`) menunjukkan pola pengkodean berikut:

- Buat instans `KinesisVideoClient`.
- Buat instans `MediaSource`.
- Mulai streaming. Mulai `MediaSource`, dan mulai mengirim data ke klien.

Bagian berikut memberikan perincian.

## Membuat sebuah instance dari KinesisVideoClient

Anda membuat [KinesisVideoClient](#) objek dengan memanggil [createKinesisVideoClient](#) operasi.

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

KinesisVideoClientUntuk melakukan panggilan jaringan, diperlukan kredensial untuk mengautentikasi. Anda meneruskan instanceAWSCredentialsProvider, yang membaca kredensi Amazon Cognito Anda dari awsconfiguration.json file yang Anda modifikasi di bagian sebelumnya.

## Membuat sebuah instance dari MediaSource

Untuk mengirim byte ke aliran video Kinesis Anda, Anda harus menghasilkan data. Amazon Kinesis Video Streams [MediaSource](#) menyediakan antarmuka, yang mewakili sumber data.

Misalnya, library Android Kinesis Video Streams [AndroidCameraMediaSource](#) menyediakan MediaSource implementasi antarmuka. Kelas ini membaca data dari salah satu kamera perangkat.

Dalam contoh kode berikut (dari [fragment/StreamConfigurationFragment.java](#) file), konfigurasi untuk sumber media dibuat:

```
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));
    }
```

Dalam contoh kode berikut (dari [fragment/StreamingFragment.java](#) file), sumber media dibuat:

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

## Memulai sumber media

Mulai sumber media sehingga dapat mulai menghasilkan data dan mengirimkannya ke klien. Contoh kode berikut adalah dari [fragment/StreamingFragment.java](#) file:

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

## Langkah selanjutnya


[the section called “Langkah 3: Jalankan dan verifikasi kode”](#)

## Langkah 3: Jalankan dan verifikasi kode

Untuk menjalankan aplikasi contoh Android untuk [Android Producer Library](#), lakukan hal berikut.

1. Connect ke perangkat Android.
2. Pilih Jalankan, Jalankan... , dan pilih Edit konfigurasi... .
3. Pilih ikon plus (+), Aplikasi Android. Di bidang Nama, masukkan **AmazonKinesisVideoDemoApp**. Dalam pulldown Modul, pilih. AmazonKinesisVideoDemoApp  
Pilih OK.
4. Pilih Jalankan, Jalankan.
5. Di layar Pilih Target Penerapan, pilih perangkat yang terhubung, dan pilih OK.

6. Dalam AWSKinesisVideoDemoAppaplikasi pada perangkat, pilih Buat akun baru.
7. Masukkan nilai untuk NAMA PENGGUNA, Kata Sandi, Nama yang diberikan, Alamat email, dan Nomor telepon, lalu pilih Daftar.

 Note


Nilai-nilai ini memiliki kendala berikut:

- Kata sandi: Harus berisi huruf besar dan kecil, angka, dan karakter khusus. [Anda dapat mengubah batasan ini di halaman kumpulan Pengguna di konsol Amazon Cognito.](#)
- Alamat email: Harus berupa alamat yang valid sehingga Anda dapat menerima kode konfirmasi.
- Nomor telepon: Harus dalam format berikut: **+<Country code><Number>**, misalnya, **+12065551212**.

8. Masukkan kode yang Anda terima melalui email, dan pilih Konfirmasi. Pilih Ok.
9. Pada halaman berikutnya, pertahankan nilai default, dan pilih Stream.
10. Masuk ke AWS Management Console dan buka konsol [Kinesis Video](#) Streams di Wilayah AS Barat (Oregon).

Pada halaman Kelola Aliran, pilih aliran demo.

11. Video streaming diputar di pemutar yang disematkan. Anda mungkin perlu menunggu waktu yang singkat (hingga sepuluh detik di bawah kondisi bandwidth dan prosesor biasa) sementara frame menumpuk sebelum video muncul.

 Note

Jika layar perangkat berputar (misalnya, dari potret ke lanskap), aplikasi berhenti streaming video.

Contoh kode membuat aliran. Saat kode dimulai, ia mulai mengirim bingkai dari kamera keKinesisVideoClient. MediaSource Klien kemudian mengirimkan data ke aliran video Kinesis bernama demo-stream.

## Menggunakan Pustaka Produser C ++

Anda dapat menggunakan Amazon Kinesis Video Streams yang disediakan C++ Produser Library untuk menulis kode aplikasi guna mengirim data media dari perangkat ke aliran video Kinesis.

### Model objek

Pustaka C ++ menyediakan objek berikut untuk mengelola pengiriman data ke aliran video Kinesis:

- `KinesisVideoProducer`: Berisi informasi tentang sumber media dan AWS kredensi Anda, serta mengelola callback untuk melaporkan peristiwa Kinesis Video Streams.
- `KinesisVideoStream`: Mewakili aliran video Kinesis. Berisi informasi tentang parameter aliran video, seperti nama, periode penyimpanan data, dan jenis konten media.

### Menempatkan media ke dalam aliran

Anda dapat menggunakan C ++ library disediakan metode (misalnya, `PutFrame`) untuk menempatkan data ke dalam `KinesisVideoStream` objek. Perpustakaan kemudian mengelola keadaan internal data, yang dapat mencakup tugas-tugas berikut:

- Melakukan autentikasi.
- Menonton latensi jaringan. Jika latensi terlalu tinggi, perpustakaan mungkin memilih untuk menjatuhkan frame.
- Pelacakan status streaming sedang berlangsung.

### Antarmuka Panggilan Balik

Lapisan ini mengekspos satu set antarmuka callback, yang memungkinkannya untuk berbicara dengan lapisan aplikasi. Antarmuka panggilan balik ini meliputi:

- Antarmuka callback layanan (`CallbackProvider`): Perpustakaan memanggil peristiwa yang diperoleh melalui antarmuka ini saat membuat aliran, memperoleh deskripsi aliran, dan menghapus aliran.
- Status siap-klien atau antarmuka peristiwa penyimpanan rendah (`ClientCallbackProvider`): Perpustakaan memanggil peristiwa pada antarmuka ini ketika klien siap, atau ketika mendeteksi bahwa itu mungkin kehabisan penyimpanan atau memori yang tersedia.



- **Stream event callback interface (StreamCallbackProvider):** Perpustakaan memanggil peristiwa pada antarmuka ini ketika peristiwa aliran terjadi, seperti aliran memasuki status siap, menjatuhkan frame, atau kesalahan aliran.

Kinesis Video Streams menyediakan implementasi default untuk antarmuka ini. Anda juga dapat memberikan implementasi kustom Anda sendiri—misalnya, jika Anda memerlukan logika jaringan khusus atau Anda ingin mengekspos kondisi penyimpanan rendah ke antarmuka pengguna.

Untuk informasi selengkapnya tentang callback di Pustaka Produser, lihat [Callback SDK produser](#)

## Prosedur: Menggunakan C++ Produser SDK

Prosedur ini menunjukkan cara menggunakan klien Kinesis Video Streams dan sumber media dalam aplikasi C++ untuk mengirim data ke aliran video Kinesis Anda.

Prosedur ini mencakup langkah-langkah berikut:

- [Langkah 1: Mengunduh dan Mengkonfigurasi Kode](#)
- [Langkah 2: Tulis dan Periksa Kode](#)
- [Langkah 3: Jalankan dan Verifikasi Kode](#)

## Prasyarat

- **Kredensial:** Dalam kode contoh, Anda memberikan kredensi dengan menentukan profil yang Anda siapkan di file profil kredensi Anda. AWS Jika Anda belum melakukannya, atur profil kredensi Anda.

Untuk informasi selengkapnya, lihat [Menyiapkan AWS Kredensi dan Wilayah untuk Pengembangan](#).

- **Integrasi toko sertifikat:** Perpustakaan Produser Kinesis Video Streams harus membangun kepercayaan dengan layanan yang dipanggilnya. Ini dilakukan dengan memvalidasi otoritas sertifikat (CA) di toko sertifikat publik. Pada model berbasis Linux, toko ini terletak di direktori `/etc/ssl`

Unduh sertifikat dari lokasi berikut ke toko sertifikat Anda:

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

- Instal dependensi build berikut untuk macOS:

- [Autoconf 2.69](#) (Lisensi GPLv3+/ Autoconf: GNU GPL versi 3 atau yang lebih baru)
- [CMake 3.7 atau 3.8](#)
- [Config Pkg](#)
- [Flex 2.5.35 Apple \(flex-31\) atau lebih baru](#)
- [Bison 2.4](#) (Lisensi GNU)
- [Automake 1.15.1](#) (Lisensi GNU)
- GNU Libtool (Apple Inc versi cctools-898)
- XCode (macOS) /dentang/gcc (xcode-pilih versi 2347)
- Java Development Kit (JDK) (untuk kompilasi Java JNI)
- [Lib-Pkg](#)
- Instal dependensi build berikut untuk Ubuntu (respons terhadap perintah versi dipotong):
  - Instal Git: `sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- Instal [CMake](#): `sudo apt-get install cmake`

```
$ cmake --version
cmake version 3.9.1
```

- Instal Libtool: `sudo apt-get install libtool`

```
2.4.6-2
```

- Instal libtool-bin: `sudo apt-get install libtool-bin`

```
$ libtool --version
libtool (GNU libtool) 2.4.6
Written by Gordon Matzigkeit, 1996
```

- Instal GNU Automake: `sudo apt-get install automake`

```
$ automake --version
automake (GNU automake) 1.15
```

- Instal GNU Bison: `sudo apt-get install bison`

```
$ bison -V
bison (GNU Bison) 3.0.4
```

- Instal G ++: `sudo apt-get install g++`

```
g++ --version
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- Instal curl: `sudo apt-get install curl`

```
$ curl --version
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- Instal pkg-config: `sudo apt-get install pkg-config`

```
$ pkg-config --version
0.29.1
```

- Instal Flex: `sudo apt-get install flex`

```
$ flex --version
flex 2.6.1
```

- Pasang OpenJDK: `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- Mengatur variabel JAVA\_HOME lingkungan: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
- Jalankan skrip build: `./install-script`

## Langkah selanjutnya

### [Langkah 1: Unduh dan Konfigurasi Kode Perpustakaan Produser C ++](#)

## Langkah 1: Unduh dan konfigurasi kode Perpustakaan Produser C ++

Untuk informasi tentang cara mengunduh dan mengonfigurasi C++ Producer Library, lihat [Amazon Kinesis Video Streams CPP Producer, GStreamer Plugin](#), dan JNI.

Untuk prasyarat dan informasi selengkapnya tentang contoh ini, lihat [Menggunakan](#) Pustaka Produser C ++.

### Langkah selanjutnya

#### [Langkah 2: Menulis dan memeriksa kode](#)

## Langkah 2: Tulis dan periksa kode

Di bagian [prosedur C ++ Producer Library ini](#), Anda memeriksa kode dalam harness uji C ++ (`tst/ProducerTestFixture.h` dan file lainnya). Anda mengunduh kode ini di bagian sebelumnya.

Platform Independent C ++ contoh menunjukkan pola coding berikut:

- Buat instance `KinesisVideoProducer` untuk mengakses Kinesis Video Streams.
- Buat instans `KinesisVideoStream`. Ini menciptakan aliran video Kinesis di Anda Akun AWS jika aliran dengan nama yang sama belum ada.
- Panggilan `putFrame` `KinesisVideoStream` untuk setiap frame data, karena tersedia, untuk mengirimkannya ke sungai.

Bagian berikut menyediakan informasi selengkapnya tentang pola pengkodean ini.

### Membuat sebuah instance dari `KinesisVideoProducer`

Anda membuat `KinesisVideoProducer` objek dengan memanggil `KinesisVideoProducer::createSync` metode. Contoh berikut menciptakan `KinesisVideoProducer` dalam `ProducerTestFixture.h` file:

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

```
defaultRegion_);
```

`createSyncMetode` ini membawa parameter berikut:

- `DeviceInfoProviderObjek`, yang mengembalikan `DeviceInfo` objek yang berisi informasi tentang perangkat atau konfigurasi penyimpanan.

#### Note

Anda mengkonfigurasi ukuran toko konten Anda menggunakan `deviceInfo.storageInfo.storageSize` parameter. Aliran konten Anda berbagi toko konten. Untuk menentukan kebutuhan ukuran penyimpanan Anda, kalikan ukuran bingkai rata-rata dengan jumlah frame yang disimpan selama durasi maks untuk semua aliran. Kemudian kalikan dengan 1,2 untuk memperhitungkan defragmentasi. Misalnya, anggaplah aplikasi Anda memiliki konfigurasi berikut:

- Tiga aliran
- Durasi maksimum 3 menit
- Setiap aliran adalah 30 frame per detik (FPS)
- Setiap frame berukuran 10.000 KB

Persyaratan toko konten untuk aplikasi ini adalah  $3 \text{ (aliran)} * 3 \text{ (menit)} * 60 \text{ (detik dalam satu menit)} * 10000 \text{ (kb)} * 1.2 \text{ (tunjangan defragmentasi)} = 194,4 \text{ Mb} \sim 200 \text{ Mb}$ .

- Sebuah `ClientCallbackProvider` objek, yang mengembalikan pointer fungsi yang melaporkan peristiwa khusus klien.
- Sebuah `StreamCallbackProvider` objek, yang mengembalikan fungsi pointer yang dipanggil kembali ketika peristiwa aliran spesifik terjadi.
- Sebuah `CredentialProvider` objek, yang menyediakan akses ke variabel lingkungan AWS kredensi.
- Wilayah AWS("us-west-2"). Endpoint layanan ditentukan dari Wilayah.

## Membuat sebuah instance dari `KinesisVideoStream`

Anda membuat `KinesisVideoStream` objek dengan memanggil `KinesisVideoProducer::CreateStream` metode dengan `StreamDefinition` parameter. Contoh membuat `KinesisVideoStream` dalam `ProducerTestFixture.h` file dengan jenis trek sebagai video, dan dengan track id sebagai 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));
```

StreamDefinitionObjek memiliki bidang berikut:

- Nama pengaliran.
- Periode retensi data.
- Tag untuk aliran. Tag ini dapat digunakan oleh aplikasi konsumen untuk menemukan aliran yang benar, atau untuk mendapatkan informasi lebih lanjut tentang aliran. Tag juga dapat dilihat di halamanAWS Management Console.
- AWS KMSkunci enkripsi untuk aliran. Untuk informasi selengkapnya, lihat [Menggunakan Enkripsi Sisi Server dengan Kinesis](#) Video Streams.
- Jenis streaming. Saat ini, satu-satunya nilai yang valid adalahSTREAMING\_TYPE\_REALTIME.
- Jenis konten media.
- Latensi media. Nilai ini saat ini tidak digunakan, dan harus diatur ke 0.
- Durasi pemutaran masing-masing fragmen.
- Skala kode waktu media.
- Apakah media menggunakan fragmentasi bingkai kunci.
- Apakah media menggunakan kode waktu.
- Apakah media menggunakan waktu fragmen absolut.

## Menambahkan trek audio ke aliran video Kinesis

Anda dapat menambahkan detail trek audio ke definisi aliran trek video dengan menggunakan metode addTrack dariStreamDefinition:

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

Metode `addTrack` memerlukan parameter berikut:

- Lacak id (sebagai satu untuk audio). Ini harus unik dan bukan nol nilai.
- Nama trek yang ditentukan pengguna (misalnya, "audio" untuk trek audio).
- Codec id untuk trek ini (misalnya, untuk trek audio "A\_AAC").
- Jenis trek (misalnya, gunakan nilai enum `MKV_TRACK_INFO_TYPE_AUDIO` untuk audio).

Jika Anda memiliki data pribadi codec untuk trek audio, maka Anda dapat meneruskannya saat memanggil fungsi `addTrack`. Anda juga dapat mengirim data pribadi codec setelah membuat `KinesisVideoStream` objek saat memanggil metode `start` di `KinesisVideoStream`.

## Menempatkan bingkai ke aliran video Kinesis

Anda menempatkan media ke dalam aliran video Kinesis menggunakan `KinesisVideoStream::putFrame`, meneruskan `Frame` objek yang berisi header dan data media. Contoh panggilan `putFrame` dalam `ProducerApiTest.cpp` file:

```
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

Contoh C ++ Produser sebelumnya mengirimkan buffer data uji. Dalam aplikasi dunia nyata, Anda harus mendapatkan buffer bingkai dan ukuran dari data bingkai dari sumber media (seperti kamera).

FrameObjek memiliki bidang berikut:

- Indeks bingkai. Ini harus menjadi nilai incrementing monoton.
- Bendera yang terkait dengan frame. Misalnya, jika encoder dikonfigurasi untuk menghasilkan bingkai kunci, frame ini akan diberi FRAME\_FLAG\_KEY\_FRAME flag.
- Decoding stempel waktu.
- Presentasi stempel waktu.
- Durasi frame (untuk 100 ns unit).
- Ukuran frame dalam byte.
- Data bingkai.

Untuk informasi selengkapnya tentang format bingkai, lihat Model [Data Kinesis Video Streams](#).

## Menempatkan KinesisVideoFrame ke dalam trek tertentu KinesisVideoStream

Anda dapat menggunakan PutFrameHelper kelas untuk menempatkan data bingkai ke trek tertentu. Pertama, panggil getFrameData Buffer untuk mendapatkan pointer ke salah satu buffer yang telah dialokasikan sebelumnya untuk mengisi data. KinesisVideoFrame Kemudian, Anda dapat memanggil putFrameMulti Track untuk mengirim KinesisVideoFrame bersama dengan nilai Boolean untuk menunjukkan jenis data frame. Gunakan true jika itu adalah data video atau false jika frame berisi data audio. Metode putFrameMulti Track menggunakan mekanisme antrian untuk memastikan bahwa Fragmen MKV mempertahankan stempel waktu bingkai yang meningkat secara monoton dan dua fragmen tidak tumpang tindih. Misalnya, stempel waktu MKV dari frame pertama fragmen harus selalu lebih besar dari stempel waktu MKV dari frame terakhir dari fragmen sebelumnya.

PutFrameHelperMemiliki bidang berikut:

- Jumlah maksimum bingkai audio dalam antrean.



- Jumlah maksimum bingkai video dalam antrian.
- Ukuran untuk mengalokasikan untuk satu bingkai audio.
- Ukuran untuk mengalokasikan untuk satu bingkai video.

## Pencatatan metrik dan metrik

C++ Produser SDK menyertakan fungsionalitas untuk metrik dan pencatatan metrik.

Anda dapat menggunakan `getKinesisVideoMetrics` dan operasi `getKinesisVideoStreamMetrics` API untuk mengambil informasi tentang Kinesis Video Streams dan aliran aktif Anda.

Kode berikut adalah dari `kinesis-video-pic/src/client/include/com/amazonaws/kinesis/video/client/Include.h` file.

```
/**
 * 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);
```

`PClietMetricsObjek` diisi oleh `getKinesisVideoMetrics` berisi informasi berikut:

- `contentStoreSize`: Ukuran keseluruhan dalam byte penyimpanan konten (memori yang digunakan untuk menyimpan data streaming).

- `contentStoreAvailableUkuran`: Memori yang tersedia di toko konten, dalam byte.
- `contentStoreAllocatedUkuran`: Memori yang dialokasikan di toko konten.
- `totalContentViewsUkuran`: Total memori yang digunakan untuk tampilan konten. Tampilan konten adalah serangkaian indeks informasi di toko konten.
- `totalFrameRate`: Jumlah agregat frame per detik di semua aliran aktif.
- `totalTransferRate`: Total bit per detik (bps) yang dikirim di semua aliran.

`PStreamMetrics` objek diisi oleh `getKinesisVideoStreamMetrics` berisi informasi berikut:

- `currentViewDuration`: Perbedaan dalam unit 100 ns antara kepala tampilan konten (ketika frame dikodekan) dan posisi saat ini (ketika data bingkai dikirim ke Kinesis Video Streams).
- `overallViewDuration`: Perbedaan dalam unit 100 ns antara kepala tampilan konten (ketika frame dikodekan) ke ekor (ketika frame memerah dari memori, baik karena total ruang yang dialokasikan untuk tampilan konten terlampaui, atau karena `PersistedAck` pesan diterima dari Kinesis Video Streams, dan frame yang diketahui bertahan akan memerah).
- `currentViewSize`: Ukuran dalam byte tampilan konten dari kepala (ketika frame dikodekan) ke posisi saat ini (ketika frame dikirim ke Kinesis Video Streams).
- `overallViewSize`: Ukuran total dalam byte tampilan konten.
- `currentFrameRate`: Tingkat diukur terakhir dari aliran, dalam frame per detik.
- `currentTransferRate`: Tingkat diukur terakhir dari aliran, dalam byte per detik.

## Teardown

Jika Anda ingin mengirim byte yang tersisa dalam buffer dan menunggu ACK, Anda dapat menggunakan: `stopSync`

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

Atau Anda dapat menelepon `stop` untuk mengakhiri streaming:

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

Setelah menghentikan streaming, Anda dapat membebaskan streaming melalui menerapkan API berikut:

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

Langkah selanjutnya

[the section called “Langkah 3: Jalankan dan verifikasi kode”](#)

## Langkah 3: Jalankan dan verifikasi kode

Untuk menjalankan dan memverifikasi kode [prosedur C++ Producer Library](#), lihat petunjuk khusus OS berikut ini:

- [Linux](#)
- [macOS](#)
- [Jendela](#)
- [Raspberry Pi](#)

Anda dapat memantau lalu lintas di streaming Anda dengan menonton metrik yang terkait dengan streaming Anda di CloudWatch konsol Amazon, seperti `PutMedia.IncomingBytes`.

## Menggunakan C++ Producer SDK sebagai plugin GStreamer

[GStreamer](#) adalah kerangka media populer yang digunakan oleh beberapa kamera dan sumber video untuk membuat jaringan pipa media khusus dengan menggabungkan plugin modular. Plugin Kinesis Video Streams GStreamer menyederhanakan integrasi pipeline media GStreamer yang ada dengan Kinesis Video Streams.

Untuk informasi tentang penggunaan C++ Producer SDK sebagai plugin GStreamer, lihat. [Contoh: Plugin SDK GStreamer Produser Kinesis Video Streamer](#)

## Menggunakan C++ Producer SDK sebagai plugin GStreamer dalam wadah Docker

[GStreamer](#) adalah kerangka media populer yang digunakan oleh beberapa kamera dan sumber video untuk membuat jaringan pipa media khusus dengan menggabungkan plugin modular. Plugin

Kinesis Video Streams GStreamer menyederhanakan integrasi pipeline media GStreamer yang ada dengan Kinesis Video Streams.

Selain itu, menggunakan [Docker](#) untuk membuat pipeline GStreamer menstandarisasi lingkungan operasi untuk Kinesis Video Streams, yang menyederhanakan membangun dan menjalankan aplikasi.

Untuk informasi tentang penggunaan C++ Producer SDK sebagai plugin GStreamer dalam container Docker, lihat. [Jalankan elemen GStreamer dalam wadah Docker](#)

## Menggunakan logging dengan C++ Produser SDK

Anda mengkonfigurasi pencatatan untuk aplikasi C++ Produser SDK dalam `kvs_log_configuration` file dalam folder. `kinesis-video-native-build`

Contoh berikut menunjukkan baris pertama dari file konfigurasi default, yang mengkonfigurasi aplikasi untuk menulis entri log DEBUG -level ke: AWS Management Console

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

Anda dapat mengatur tingkat logging INFO untuk logging kurang verbose.

Untuk mengkonfigurasi aplikasi untuk menulis entri log ke file log, perbarui baris pertama dalam file ke yang berikut:

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

Ini mengkonfigurasi aplikasi untuk menulis entri log ke `kvs.log` dalam folder. `kinesis-video-native-build/log`

Untuk mengubah lokasi file log, perbarui baris berikut dengan jalur baru:

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

### Note

Jika DEBUG pencatatan level ditulis ke file, file log dapat menggunakan ruang penyimpanan yang tersedia di perangkat dengan cepat.

## Menggunakan Pustaka Produser C

Anda dapat menggunakan Amazon Kinesis Video Streams yang disediakan C Producer Library untuk menulis kode aplikasi guna mengirim data media dari perangkat ke aliran video Kinesis.

### Model objek

[Kinesis Video Streams C Produser Library](https://github.com/aws-labs/-pic/) didasarkan pada komponen umum yang disebut [Platform Independent Codebase \(PIC\)](https://github.com/aws-labs/-pic/), yang tersedia GitHub di <https://github.com/aws-labs/-pic/>. [amazon-kinesis-video-streams](https://github.com/aws-labs/-pic/) PIC berisi logika bisnis platform-independen untuk komponen dasar. Kinesis Video Streams C Producer Library membungkus PIC dengan lapisan API tambahan yang memungkinkan callback dan peristiwa khusus skenario dan platform. Kinesis Video Streams C Producer Library memiliki komponen-komponen berikut yang dibangun di atas PIC:

- Penyedia info perangkat - Mengekspos `DeviceInfo` struktur yang dapat langsung dipasok ke PIC API. Anda dapat mengkonfigurasi serangkaian penyedia, termasuk penyedia aplikasi yang dioptimalkan skenario yang dapat mengoptimalkan penyimpanan konten berdasarkan jumlah dan jenis aliran yang ditangani aplikasi Anda dan jumlah buffering yang diperlukan yang dikonfigurasi berdasarkan jumlah RAM yang tersedia.
- Penyedia info streaming - Mengekspos `StreamInfo` struktur yang dapat langsung dipasok ke PIC API. Ada serangkaian penyedia yang khusus untuk jenis aplikasi dan jenis skenario streaming yang umum. Ini termasuk penyedia seperti video, audio, dan audio dan video multitrack. Masing-masing skenario ini memiliki default yang dapat Anda sesuaikan sesuai dengan kebutuhan aplikasi Anda.
- Penyedia panggilan balik - Mengekspos `ClientCallbacks` struktur yang dapat langsung dipasok ke PIC API. Ini termasuk serangkaian penyedia callback untuk jaringan (Curl berbasis API callback), otorisasi (AWS kredensi API), dan coba lagi streaming pada callback kesalahan. API Penyedia Callback mengambil sejumlah argumen untuk dikonfigurasi, seperti informasi Wilayah AWS dan otorisasi. Ini dilakukan dengan menggunakan sertifikat IoT atau dengan menggunakan `AWSSessionToken`, `AWSCredentials`, atau `AWSSessionToken`. Anda dapat meningkatkan Callback Provider dengan callback kustom jika aplikasi Anda membutuhkan pemrosesan lebih lanjut dari callback tertentu untuk mencapai beberapa logika khusus aplikasi.
- `FrameOrderCoordinator` - Membantu menangani sinkronisasi audio dan video untuk skenario multitrack. Ini memiliki perilaku default, yang dapat Anda sesuaikan untuk menangani logika spesifik aplikasi Anda. Ini juga menyederhanakan kemasan metadata bingkai dalam struktur Bingkai PIC sebelum mengirimkannya ke PIC API lapisan bawah. Untuk skenario non-multitrack, komponen ini adalah pass-through ke PIC `PutFrame` API.

Perpustakaan C menyediakan objek berikut untuk mengelola pengiriman data ke aliran video Kinesis:

- `KinesisVideoClient`- Berisi informasi tentang perangkat Anda dan mengelola callback untuk melaporkan peristiwa Kinesis Video Streams.
- `KinesisVideoStream`- Merupakan informasi tentang parameter aliran video, seperti nama, periode retensi data, dan jenis konten media.

## Menempatkan media ke dalam aliran

Anda dapat menggunakan C library disediakan metode (misalnya, `PutKinesisVideoFrame`) untuk menempatkan data ke dalam `KinesisVideoStream` objek. Perpustakaan kemudian mengelola keadaan internal data, yang dapat mencakup tugas-tugas berikut:

- Melakukan autentikasi.
- Menonton latensi jaringan. Jika latensi terlalu tinggi, perpustakaan mungkin memilih untuk menjatuhkan frame.
- Pelacakan status streaming sedang berlangsung.

## Prosedur: Menggunakan SDK Produser C

Prosedur ini menunjukkan cara menggunakan klien Kinesis Video Streams dan sumber media dalam aplikasi C untuk mengirim bingkai video yang dikodekan H.264 ke aliran video Kinesis Anda.

Prosedur ini mencakup langkah-langkah berikut:

- [Langkah 1: Mengunduh kode Perpustakaan Produser Perpustakaan](#)
- [Langkah 2: Tulis dan periksa kode](#)
- [Langkah 3: Jalankan dan verifikasi kode](#)

## Prasyarat

- Kredensi - Dalam kode contoh, Anda memberikan kredensi dengan menentukan profil yang Anda siapkan di file profil kredensi Anda. AWS Jika Anda belum melakukannya, pertama-tama siapkan profil kredensial Anda.

Untuk informasi selengkapnya, lihat [Menyiapkan AWS Kredensial dan Wilayah untuk Pengembangan](#).

- Integrasi toko sertifikat - Perpustakaan Produser Kinesis Video Streams harus membangun kepercayaan dengan layanan yang disebutnya. Ini dilakukan dengan memvalidasi otoritas sertifikat (CA) di toko sertifikat publik. Pada model berbasis Linux, toko ini terletak di direktori `./etc/ssl`

Unduh sertifikat dari lokasi berikut ke toko sertifikat Anda:

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

- Instal dependensi build berikut untuk macOS:
  - [Autoconf 2.69](#) (Lisensi GPLv3+/ Autoconf: GNU GPL versi 3 atau yang lebih baru)
  - [CMake 3.7 atau 3.8](#)
  - [Config Pkg](#)
  - [Flex 2.5.35 Apple \(flex-31\) atau lebih baru](#)
  - [Bison 2.4](#) (Lisensi GNU)
  - [Automake 1.15.1](#) (Lisensi GNU)
  - GNU Libtool (Apple Inc versi cctools-898)
  - XCode (macOS) /dentang/gcc (xcode-pilih versi 2347)
  - Java Development Kit (JDK) (untuk kompilasi Java JNI)
  - [Lib-Pkg](#)
- Instal dependensi build berikut untuk Ubuntu (respons terhadap perintah versi dipotong):
  - Instal Git: `sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- Instal [CMake](#): `sudo apt-get install cmake`

```
$ cmake --version
cmake version 3.9.1
```

- Instal Libtool: `sudo apt-get install libtool`

```
2.4.6-2
```

- Instal libtool-bin: `sudo apt-get install libtool-bin`

```
$ libtool --version
libtool (GNU libtool) 2.4.6
```

```
Written by Gordon Matzigkeit, 1996
```

- Instal GNU Automake: `sudo apt-get install automake`

```
$ automake --version
automake (GNU automake) 1.15
```

- Instal GNU Bison: `sudo apt-get install bison`

```
$ bison -V
bison (GNU Bison) 3.0.4
```

- Instal G ++: `sudo apt-get install g++`

```
g++ --version
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- Instal curl: `sudo apt-get install curl`

```
$ curl --version
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- Instal pkg-config: `sudo apt-get install pkg-config`

```
$ pkg-config --version
0.29.1
```

- Instal Flex: `sudo apt-get install flex`

```
$ flex --version
flex 2.6.1
```

- Pasang OpenJDK: `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- Mengatur variabel JAVA\_HOME lingkungan: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
- Jalankan skrip build: `./install-script`



## Langkah selanjutnya

### [Langkah 1: Mengunduh kode Perpustakaan Produser Perpustakaan](#)

## Langkah 1: Mengunduh kode Perpustakaan Produser Perpustakaan

Di bagian ini, Anda mengunduh pustaka tingkat rendah. Untuk prasyarat dan detail lainnya tentang contoh ini, lihat [Menggunakan](#) Pustaka Produser C.

1. Buat direktori, dan kemudian kloning kode sumber contoh dari GitHub repositori.

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

#### Note

Jika Anda melewatkan menjalankan `git clone` dengan `--recursive`, jalankan `git submodule update --init` di `amazon-kinesis-video-streams-producer-c/open-source` direktori. Anda juga harus menginstal `pkg-config`, `automake`, `CMake`, dan lingkungan build.

Untuk informasi lebih lanjut, lihat `README.md` di <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git>.

2. Buka kode di lingkungan pengembangan terintegrasi (IDE) pilihan Anda (misalnya, [Eclipse](#)).

## Langkah selanjutnya

### [Langkah 2: Tulis dan periksa kode](#)

## Langkah 2: Tulis dan periksa kode

Di bagian ini, Anda memeriksa kode aplikasi sampel `KvsVideoOnlyStreamingSample.c` di `samples` folder repo <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c> aktif. GitHub Anda mengunduh kode di langkah sebelumnya. Contoh ini menunjukkan cara menggunakan C Producer Library untuk mengirim bingkai video yang dikodekan H.264 di dalam folder `samples/h264SampleFrames` ke aliran video Kinesis Anda.

Aplikasi sampel ini memiliki tiga bagian:

- Inisialisasi dan konfigurasi:
  - Menginisialisasi dan mengkonfigurasi pipeline media khusus platform.
  - Menginisialisasi dan mengkonfigurasi KinesisVideoClient dan KinesisVideoStream untuk pipeline, mengatur callback, mengintegrasikan otentikasi skenario khusus, mengekstraksi dan mengirimkan data pribadi codec, dan mendapatkan aliran ke status READY.
- Loop utama:
  - Mendapatkan frame dari pipeline media dengan stempel waktu dan bendera.
  - Mengirimkan frame ke KinesisVideoStream
- Penurunan air mata:
  - Menghentikan (sinkronisasi)KinesisVideoStream, membebaskanKinesisVideoStream, membebaskanKinesisVideoClient.

Aplikasi contoh ini menyelesaikan tugas berikut:

- Panggil `createDefaultDeviceInfo` API untuk membuat `deviceInfo` objek yang berisi informasi tentang perangkat atau konfigurasi penyimpanan.

```
// 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;
```

- Panggil `createRealtimeVideoStreamInfoProvider` API untuk membuat `StreamInfo` objek.

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

- Panggil `createDefaultCallbacksProviderWithAwsCredentials` API untuk membuat penyedia callback default berdasarkan AWS kredensi statis.

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

- Panggil `createKinesisVideoClient` API untuk membuat `KinesisVideoClient` objek yang berisi informasi tentang penyimpanan perangkat Anda dan mengelola callback untuk melaporkan peristiwa Kinesis Video Streams.

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

- Panggil `createKinesisVideoStreamSync` API untuk membuat `KinesisVideoStream` objek.

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

- Siapkan bingkai sampel dan panggil `PutKinesisVideoFrame` API untuk mengirim bingkai itu ke `KinesisVideoStream` objek.

```
// 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;
}
```

- Penurunan air mata:

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

## Langkah selanjutnya

### [Langkah 3: Jalankan dan verifikasi kode](#)

## Langkah 3: Jalankan dan verifikasi kode

Untuk menjalankan dan memverifikasi kode untuk [prosedur Perpustakaan Produser](#), lakukan hal berikut:

1. Jalankan perintah berikut untuk membuat build direktori di [C SDK yang Anda unduh](#), dan luncurkan cmake darinya:

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

Anda dapat melalui opsi berikut cmake . .

- -DBUILD\_DEPENDENCIES- apakah akan membangun perpustakaan tergantung dari sumber.
- -DBUILD\_TEST=TRUE- membangun unit dan tes integrasi. Mungkin berguna untuk mengonfirmasi dukungan untuk perangkat Anda.

```
./tst/webrtc_client_test
```

- -DCODE\_COVERAGE- Aktifkan pelaporan cakupan.
- -DCOMPILER\_WARNINGS- Aktifkan semua peringatan compiler.
- -DADDRESS\_SANITIZER- membangun denganAddressSanitizer.
- -DMEMORY\_SANITIZER- membangun denganMemorySanitizer.
- -DTHREAD\_SANITIZER- membangun denganThreadSanitizer.
- -DUNDEFINED\_BEHAVIOR\_SANITIZER- membangun denganUndefinedBehaviorSanitizer.
- -DALIGNED\_MEMORY\_MODEL- membangun untuk model memori selaras hanya perangkat. Default-nya adalah OFF.

2. Arahkan ke build direktori yang baru saja Anda buat dengan langkah sebelumnya, dan jalankan make untuk membangun SDK WebRTC C dan sampel yang disediakan.

```
make
```

3. Aplikasi sampel kinesis\_video\_cproducer\_video\_only\_sample mengirimkan bingkai video yang dikodekan h.264 di dalam folder samples/h264SampleFrames ke Kinesis Video Streams. Perintah berikut mengirimkan bingkai video dalam satu lingkaran selama sepuluh detik ke Kinesis Video Streams:

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

Jika Anda ingin mengirim frame yang dikodekan H.264 dari folder lain (misalnya,MyH264FramesFolder), jalankan sampel dengan argumen berikut:

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

4. Untuk mengaktifkan log verbose, tentukan `HEAP_DEBUG` dan `LOG_STREAMING` C-defines dengan menghapus komentar pada baris yang sesuai. `CMakeList.txt`

Anda dapat memantau kemajuan rangkaian di output debug di IDE Anda. Anda juga dapat memantau lalu lintas di streaming Anda dengan menonton metrik yang terkait dengan streaming Anda di CloudWatch konsol Amazon, seperti `PutMedia.IncomingBytes`.

#### Note

Konsol tidak menampilkan data sebagai aliran video karena harness pengujian hanya mengirimkan bingkai byte kosong.

## Menggunakan C++ Producer SDK pada Raspberry Pi

Raspberry Pi adalah komputer kecil dan murah yang dapat digunakan untuk mengajar dan mempelajari keterampilan pemrograman komputer dasar. Tutorial ini menjelaskan bagaimana Anda dapat mengatur dan menggunakan Amazon Kinesis Video Streams C++ Producer SDK pada perangkat Raspberry Pi. Langkah-langkahnya juga mencakup cara memverifikasi instalasi menggunakan aplikasi demo GStreamer.

### Topik

- [Prasyarat](#)
- [Buat pengguna IAM dengan izin untuk menulis ke Kinesis Video Streams](#)
- [Bergabunglah dengan Raspberry Pi Anda ke jaringan Wi-Fi Anda](#)
- [Connect dari jarak jauh ke Raspberry Pi](#)
- [Konfigurasi kamera Raspberry Pi](#)
- [Instal prasyarat perangkat lunak](#)
- [Unduh dan buat Kinesis Video Streams C++ Producer SDK](#)
- [Streaming video ke streaming video Kinesis Anda dan lihat streaming langsung](#)

## Prasyarat

Sebelum Anda mengatur C++ Producer SDK pada Raspberry Pi Anda, pastikan Anda memiliki prasyarat berikut:

- Perangkat Raspberry Pi dengan konfigurasi berikut:
  - Versi papan: 3 Model B atau yang lebih baru.
  - Modul kamera yang terhubung.
  - Kartu SD dengan kapasitas minimal 8 GB.
  - Sistem operasi Raspbian (kernel versi 4.9 atau yang lebih baru) diinstal. Anda dapat mengunduh gambar Raspberry Pi OS (sebelumnya disebut Raspbian) terbaru dari [situs web Raspberry Pi](#). Ikuti petunjuk Raspberry Pi untuk [menginstal gambar yang diunduh pada kartu SD](#).
- Sebuah Akun AWS dengan aliran video Kinesis. Untuk informasi selengkapnya, lihat [Memulai dengan Kinesis Video Streams](#).

### Note

C++ Producer SDK menggunakan Wilayah AS Barat (Oregon) (`us-west-2`) secara default. Untuk menggunakan default, Wilayah AWS buat aliran video Kinesis Anda di Wilayah AS Barat (Oregon).

Untuk menggunakan Wilayah yang berbeda untuk streaming video Kinesis Anda, lakukan salah satu hal berikut:

- Tetapkan variabel lingkungan berikut ke Wilayah Anda (misalnya, `us-east-1`):

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

## Buat pengguna IAM dengan izin untuk menulis ke Kinesis Video Streams

Jika Anda belum melakukannya, siapkan pengguna AWS Identity and Access Management (IAM) dengan izin untuk menulis ke aliran video Kinesis.

Prosedur ini dimaksudkan untuk membantu Anda memulai dengan cepat menggunakan AWS access key pair. Perangkat dapat menggunakan sertifikat X.509 untuk terhubung. AWS IoT lihat [the section called “Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan AWS IoT”](#) untuk

informasi selengkapnya tentang cara mengonfigurasi perangkat agar menggunakan autentikasi berbasis sertifikat.

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di menu navigasi di sebelah kiri, pilih Pengguna.
3. Untuk membuat pengguna baru, pilih Tambah pengguna.
4. Berikan nama Pengguna deskriptif untuk pengguna, seperti **kinesis-video-raspberry-pi-producer**.
5. Di bawah Jenis akses, pilih Akses terprogram.
6. Pilih Selanjutnya: Izin.
7. Di bawah Setel izin untuk kinesis-video-raspberry-pi -produser, pilih Lampirkan kebijakan yang ada secara langsung.
8. Pilih Buat kebijakan. Halaman Buat kebijakan terbuka di tab browser web baru.
9. Pilih tab JSON.
10. Salin kebijakan JSON berikut dan tempelkan ke area teks. Kebijakan ini memberikan izin kepada pengguna Anda untuk membuat dan menulis data ke aliran video Kinesis.

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

11. Pilih Tinjau kebijakan.
12. Berikan nama untuk kebijakan Anda, seperti **kinesis-video-stream-write-policy**.
13. Pilih Buat kebijakan.



14. Kembali ke tab Tambah pengguna di browser Anda, dan pilih Segarkan.
15. Di kotak pencarian, ketikkan nama kebijakan yang Anda buat.
16. Pilih kotak centang di samping kebijakan baru Anda dalam daftar.
17. Pilih Berikutnya: Tinjauan.
18. Pilih Create user (Buat pengguna).
19. Konsol menampilkan ID kunci Akses untuk pengguna baru Anda. Pilih Tampilkan untuk menampilkan kunci akses Rahasia. Catat nilai-nilai ini; mereka diperlukan saat Anda mengkonfigurasi aplikasi.

## Bergabunglah dengan Raspberry Pi Anda ke jaringan Wi-Fi Anda

Anda dapat menggunakan Raspberry Pi dalam mode tanpa kepala, yaitu, tanpa keyboard, monitor, atau kabel jaringan yang terpasang. Jika Anda menggunakan monitor dan keyboard yang terpasang, lanjutkan ke [Konfigurasi kamera Raspberry Pi](#).

1. Di komputer Anda, buat file bernama `wpa_supplicant.conf`.
2. Salin teks berikut dan tempel ke dalam `wpa_supplicant.conf` file:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="Your Wi-Fi SSID"
  scan_ssid=1
  key_mgmt=WPA-PSK
  psk="Your Wi-Fi Password"
}
```

Ganti `ssid` dan `psk` nilai dengan informasi untuk jaringan Wi-Fi Anda.

3. Salin `wpa_supplicant.conf` file ke kartu SD. Itu harus disalin ke root boot volume.
4. Masukkan kartu SD ke Raspberry Pi, dan nyalakan perangkat. Ini bergabung dengan jaringan Wi-Fi Anda, dan SSH diaktifkan.

## Connect dari jarak jauh ke Raspberry Pi

Anda dapat terhubung dari jarak jauh ke Raspberry Pi Anda dalam mode tanpa kepala. Jika Anda menggunakan Raspberry Pi Anda dengan monitor dan keyboard yang terhubung, lanjutkan ke [Konfigurasi kamera Raspberry Pi](#).

1. Sebelum menghubungkan ke perangkat Raspberry Pi Anda dari jarak jauh, lakukan salah satu hal berikut untuk menentukan alamat IP-nya:
  - Jika Anda memiliki akses ke router Wi-Fi jaringan Anda, lihat perangkat Wi-Fi yang terhubung. Temukan perangkat yang diberi nama Raspberry Pi untuk menemukan alamat IP perangkat Anda.
  - Jika Anda tidak memiliki akses ke router Wi-Fi jaringan Anda, Anda dapat menggunakan perangkat lunak lain untuk menemukan perangkat di jaringan Anda. [Fing](#) adalah aplikasi populer yang tersedia untuk perangkat Android dan iOS. Anda dapat menggunakan versi gratis aplikasi ini untuk menemukan alamat IP perangkat di jaringan Anda.
2. Ketika Anda mengetahui alamat IP perangkat Raspberry Pi, Anda dapat menggunakan aplikasi terminal apa pun untuk terhubung.

- Di macOS atau Linux, gunakan: `ssh`

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

- Di Windows, gunakan [PuTTY](#), klien SSH gratis untuk Windows.

Untuk instalasi baru Raspbian, nama pengguna adalah **pi**, dan kata sandinya **raspberry**. Kami menyarankan Anda [mengubah kata sandi default](#).

## Konfigurasi kamera Raspberry Pi

Ikuti langkah-langkah ini untuk mengonfigurasi kamera Raspberry Pi untuk mengirim video dari perangkat ke aliran video Kinesis.

1. Buka editor untuk memperbarui `modules` file dengan perintah berikut:

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

2. Tambahkan baris berikut ke akhir file, jika belum ada:

```
bcm2835-v4l2
```

3. Simpan file dan keluar dari editor (Ctrl-X).
4. Nyalakan ulang Raspberry Pi:

```
$ sudo reboot
```

5. Saat perangkat reboot, sambungkan lagi melalui aplikasi terminal Anda jika Anda terhubung dari jarak jauh.
6. Bukarasp-config:

```
$ sudo raspi-config
```

7. Pilih Opsi Antarmuka, Kamera Lama. Dalam versi lama dari Sistem Operasi Raspbian, opsi menu ini mungkin berada di bawah Opsi Antarmuka, Kamera.

Aktifkan kamera jika belum diaktifkan, dan reboot jika diminta.

8. Verifikasi bahwa kamera bekerja dengan mengetikkan perintah berikut:

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

Jika kamera Anda dikonfigurasi dengan benar, perintah ini menangkap gambar dari kamera, menyimpannya ke file bernama `test.jpg`, dan menampilkan pesan informasi.

## Instal prasyarat perangkat lunak

C++ Producer SDK mengharuskan Anda menginstal prasyarat perangkat lunak berikut pada Raspberry Pi.

1. Perbarui daftar paket dan instal pustaka yang diperlukan untuk membangun SDK. Ketik perintah berikut:

```
$ sudo apt update
$ sudo apt install -y \
  automake \
  build-essential \
  cmake \
  git \
```

```
gststreamer1.0-plugins-base-apps \  
gststreamer1.0-plugins-bad \  
gststreamer1.0-plugins-good \  
gststreamer1.0-plugins-ugly \  
gststreamer1.0-tools \  
gststreamer1.0-omx-generic \  
libcurl4-openssl-dev \  
libgststreamer1.0-dev \  
libgststreamer-plugins-base1.0-dev \  
liblog4cplus-dev \  
libssl-dev \  
pkg-config
```

2. Salin file PEM berikut ke `/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
```

## Unduh dan buat Kinesis Video Streams C++ Producer SDK

Anda dapat mengunduh dan membangun Kinesis Video Streams C++ Producer SDK menggunakan prosedur berikut. Pendekatan ini membutuhkan waktu lebih lama untuk membangun, tergantung pada konektivitas jaringan dan kecepatan prosesor.

1. Unduh SDK. Jenis:

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

2. Siapkan direktori build. Jenis:

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

3. Bangun aplikasi SDK dan Sampel. Tergantung pada model Raspberry Pi yang Anda bangun, mungkin perlu beberapa jam untuk berjalan untuk pertama kalinya:

```
$ cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=FALSE  
$ make
```

## Streaming video ke streaming video Kinesis Anda dan lihat streaming langsung

1. Untuk menjalankan aplikasi sampel, Anda memerlukan informasi berikut:
  - Nama aliran yang Anda buat di [Prasyarat](#) bagian ini.
  - Kredensi akun (ID kunci akses dan kunci akses rahasia) yang Anda buat. [Buat pengguna IAM dengan izin untuk menulis ke Kinesis Video Streams](#)
2. Jalankan aplikasi sampel menggunakan perintah berikut. Ganti placeholder dengan nilai untuk lingkungan Anda.

```
$ 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. Jika aplikasi sampel keluar dengan `library not found` kesalahan, ketik perintah berikut untuk memverifikasi bahwa proyek ditautkan dengan benar ke dependensi sumber terbukanya:

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

4. Buka konsol [Kinesis Video Streams](#).
5. Pilih nama Streaming dari aliran yang Anda buat.

Aliran video yang dikirim dari Raspberry Pi muncul di konsol.

Saat streaming diputar, Anda dapat bereksperimen dengan fitur berikut dari konsol Kinesis Video Streams:

- Di bagian Pratinjau video, gunakan kontrol navigasi untuk memundurkan atau mempercepat aliran.
- Di bagian Stream info, perhatikan codec, resolusi, dan bitrate aliran. Resolusi dan nilai bitrate diatur dengan sengaja rendah pada Raspberry Pi untuk meminimalkan penggunaan bandwidth untuk tutorial ini. Untuk melihat CloudWatch metrik Amazon yang sedang dibuat untuk streaming Anda, pilih Lihat metrik aliran di CloudWatch

- Di bawah periode penyimpanan data, perhatikan bahwa aliran video dipertahankan selama satu hari. Anda dapat mengedit nilai ini dan mengaturnya ke Tidak ada retensi data, atau menetapkan nilai dari satu hari hingga beberapa tahun.

Di bawah enkripsi sisi server, perhatikan bahwa data Anda sedang dienkrpsi saat istirahat menggunakan kunci yang dikelola oleh (). AWS Key Management Service AWS KMS

## Referensi SDK Produser

Bagian ini berisi batasan, kode kesalahan, dan informasi referensi lainnya untuk [Perpustakaan Produser Aliran Video Kinesis](#).

Topik

- [Batas SDK produser](#)
- [Referensi kode kesalahan](#)
- [Referensi bendera adaptasi Network Abstraction Layer \(NAL\)](#)
- [Struktur SDK produser](#)
- [Struktur aliran video Kinesis](#)
- [Callback SDK produser](#)

## Batas SDK produser

Tabel berikut berisi batas saat ini untuk nilai-nilai dalam [Pustaka Produser](#).

### Note

Sebelum menetapkan nilai-nilai ini, Anda harus memvalidasi input Anda. SDK tidak memvalidasi batas ini, dan kesalahan runtime terjadi jika batas terlampaui.

Nilai	Kuota	Catatan
Jumlah aliran maks	128	Jumlah maksimum aliran yang dapat dibuat oleh objek produser. Ini adalah batas

Nilai	Kuota	Catatan
		lunak (Anda dapat meminta kenaikan). Ini menjamin bahwa produsen tidak secara tidak sengaja membuat aliran secara rekursif.
Panjang nama perangkat maks	128 karakter	
Jumlah tag maks	50 per aliran	
Panjang nama aliran maks	256 karakter	
Ukuran penyimpanan min	10 MiB = $10 * 1024 * 1024$ byte	
Ukuran penyimpanan maks	10 GiB = $10 * 1024 * 1024 * 1024$ byte	
Panjang jalur direktori root maks	4.096 karakter	
Panjang info autentikasi maks	10.000 byte	
Panjang string URI maks	10.000 karakter	
Panjang nama tag maks	128 karakter	
Panjang nilai tag maks	1.024 karakter	
Periode token keamanan min	30 detik	
Masa tenggang token keamanan	40 menit	Jika durasi yang ditentukan lebih lama, itu terbatas pada nilai ini.
Periode penahanan	0 atau lebih dari satu jam	0 menunjukkan tidak ada retensi.

Nilai	Kuota	Catatan
Durasi cluster min	1 detik	Nilai ditentukan dalam unit 100 ns, yang merupakan standar SDK.
Durasi cluster maks	30 detik	Nilai ditentukan dalam unit 100 ns, yang merupakan standar SDK. API backend dapat menerapkan durasi cluster yang lebih pendek.
Ukuran fragmen maks	50 MB	Untuk informasi selengkapnya, lihat <a href="#">Kuota layanan Kinesis Video Streams</a> .
Durasi fragmen maks	20 detik	Untuk informasi selengkapnya, lihat <a href="#">Kuota layanan Kinesis Video Streams</a> .
Durasi koneksi maks	45 menit	Backend menutup koneksi setelah waktu ini. SDK memutar token dan membuat koneksi baru dalam waktu ini.
Panjang segmen ACK maks	1.024 karakter	Panjang segmen maksimum dari pengakuan yang dikirim ke fungsi parser ACK.
Panjang string tipe konten maks	128 karakter	
Panjang string ID codec maks	32 karakter	
Panjang string nama trek maks	32 karakter	
Panjang data pribadi codec maks	1 MiB = 1 * 1024 * 1024 byte	



Nilai	Kuota	Catatan
Panjang nilai skala kode waktu min	100 ns	Nilai skala kode waktu minimum untuk mewakili stempel waktu bingkai di cluster MKV yang dihasilkan. Nilai ditentukan dalam penambahan 100 ns, yang merupakan standar SDK.
Panjang nilai skala kode waktu maks	1 detik	Nilai skala timecode maksimum untuk mewakili stempel waktu bingkai di cluster MKV yang dihasilkan. Nilai ditentukan dalam penambahan 100 ns, yang merupakan standar SDK.
Jumlah item tampilan konten min	10	
Durasi buffer min	20 detik	Nilai ditentukan dalam penambahan 100 ns, yang merupakan standar SDK.
Panjang versi pembaruan maks	128 karakter	
Panjang ARN maks	1024 karakter	
Panjang urutan fragmen maks	128 karakter	
Periode retensi maks	10 tahun	

## Referensi kode kesalahan

Bagian ini berisi informasi kode kesalahan dan status untuk [Pustaka Produser](#).

Untuk informasi tentang solusi untuk masalah umum, lihat [Memecahkan Masalah Kinesis Video Streams](#).

### Topik

- [Kesalahan dan kode status yang dikembalikan oleh PutFrame Callback - Platform Independent Code \(PIC\)](#)
- [Kesalahan dan kode status yang dikembalikan oleh PutFrame callback - pustaka produser C](#)

## Kesalahan dan kode status yang dikembalikan oleh PutFrame Callback - Platform Independent Code (PIC)

Bagian berikut berisi informasi kesalahan dan status yang dikembalikan oleh callback untuk PutFrame operasi dalam Platform Independent Code (PIC).

### Topik

- [Kode kesalahan dan status yang dikembalikan oleh pustaka klien](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka durasi](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka umum](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka heap](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka MKVGen](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka Trace](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka Utils](#)
- [Kode kesalahan dan status yang dikembalikan oleh pustaka View](#)

Kode kesalahan dan status yang dikembalikan oleh pustaka klien

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di pustaka Kinesis Client Video Streams.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000001	STATUS_MAX_STREAM_COUNT	Jumlah aliran maksimum tercapai.	Tentukan jumlah aliran maks yang lebih besar DeviceInfo seperti yang ditentukan

Code	Pesan	Deskripsi	Tindakan yang disarankan
			n dalam <a href="#">Batas SDK produsen</a> .
0x52000002	STATUS_MIN_STREAM_COUNT	Kesalahan jumlah aliran minimum.	Tentukan jumlah maksimum aliran yang lebih besar dari nol inDeviceInfo .
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	Panjang nama perangkat tidak valid.	Lihat panjang nama perangkat maksimal dalam karakter yang ditentukan <a href="#">Batas SDK produsen</a> .
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	Versi DeviceInfo struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x52000005	STATUS_MAXIMUM_TAG_COUNT	Jumlah tag maksimum tercapai.	Lihat jumlah tag maks saat ini yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	Versi Callbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	Versi StreamInfo struktur tidak valid.	Tentukan versi struktur yang benar saat ini.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	Panjang nama aliran tidak valid.	Lihat panjang nama aliran maksimum dalam karakter yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200000a	STATUS_INVALID_STORAGE_SIZE	Ukuran penyimpanan yang tidak valid ditentukan.	Ukuran penyimpanan dalam byte harus dalam batas yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200000b	STATUS_INVALID_ROOT_DIRECTORY_LENGTH	Panjang string direktori root tidak valid.	Lihat panjang jalur direktori root maks yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200000c	STATUS_INVALID_SPILL_RATIO	Rasio tumpahan tidak valid.	Nyatakan rasio tumpahan sebagai persentase dari 0-100.
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	Versi StorageInfo struktur tidak valid.	Tentukan versi struktur yang benar saat ini.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200000e	STATUS_INVALID_STREAM_STATE	Aliran berada dalam keadaan yang tidak mengizinkan operasi saat ini.	Paling umum, kesalahan ini terjadi ketika SDK gagal mencapai status yang diperlukan untuk melakukan operasi yang diminta. Misalnya, ini terjadi jika panggilan <code>GetStreamingEndpoint</code> API gagal, dan aplikasi klien mengabaikannya dan terus menempatkan bingkai ke dalam aliran.
0x5200000f	STATUS_SERVICE_CALL_CALLBACKS_MISSING	Callbacks Struktur memiliki titik masuk fungsi yang hilang untuk beberapa fungsi wajib.	Verifikasi bahwa panggilan balik wajib diimplementasikan dalam aplikasi klien. Kesalahan ini hanya diekspos ke klien Platform Independent Code (PIC). C++ dan pembungkus tingkat tinggi lainnya memenuhi panggilan ini.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000010	STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR	Tidak diizinkan.	Verifikasi token keamanan, sertifikat, integrasi token keamanan, dan kedaluwarsa. Verifikasi bahwa token memiliki hak terkait yang benar dengannya. Untuk contoh aplikasi Kinesis Video Streams, verifikasi bahwa variabel lingkungan disetel dengan benar.
0x52000011	STATUS_DESCRIBE_STREAM_CALL_FAILED	DescribeStream Kegagalan API.	Kesalahan ini dikembalikan setelah kegagalan coba lagi DescribeStream API. Klien PIC mengembalikan kesalahan ini setelah berhenti mencoba lagi.
0x52000012	STATUS_INVALID_DESCRIPTOR_RESPONSE	Struktur tidak validDescribeStreamResponse .	Struktur yang diteruskan ke nol atau berisi item yang tidak valid seperti Nama Sumber Daya Amazon null (ARN). DescribeStreamResultEvent
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	Streaming sedang dihapus.	Kegagalan API disebabkan oleh aliran yang dihapus. Verifikasi bahwa tidak ada proses lain yang mencoba menghapus aliran saat aliran sedang digunakan.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000014	STATUS_SERVICE_CALL_INVALID_ARG_ERROR	Argumen tidak valid ditentukan untuk panggilan layanan.	Backend menampilkan kesalahan ini ketika argumen panggilan layanan tidak valid atau ketika SDK menemukan kesalahan yang tidak dapat ditafsirkan.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	Perangkat tidak ditemukan.	Pastikan perangkat tidak dihapus saat digunakan.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	Perangkat tidak disediakan.	Verifikasi bahwa perangkat telah disediakan.
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	Sumber daya generik tidak ditemukan dikembalikan dari layanan.	Kesalahan ini terjadi ketika layanan tidak dapat menemukan sumber daya (misalnya, aliran). Ini mungkin berarti hal yang berbeda dalam konteks yang berbeda, tetapi kemungkinan penyebabnya adalah penggunaan API sebelum aliran dibuat. Menggunakan SDK mengonfirmasi bahwa aliran dibuat terlebih dahulu.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000018	STATUS_INVALID_AUTH_LEN	Panjang info autentikasi tidak valid.	Lihat nilai saat ini yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	Panggilan CreateStream API gagal.	Lihat string kesalahan untuk informasi lebih rinci tentang mengapa operasi gagal.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	GetStreamingToken Panggilan gagal.	Lihat string kesalahan untuk informasi lebih rinci tentang mengapa operasi gagal.
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	Panggilan GetStreamingEndpoint API gagal.	Lihat string kesalahan untuk informasi lebih rinci tentang mengapa operasi gagal.
0x5200002c	STATUS_INVALID_URI_LEN	Panjang string URI yang tidak valid dikembalikan dari API. GetStreamingEndpoint	Lihat nilai maksimum saat ini yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	Panggilan PutMedia API gagal.	Lihat string kesalahan untuk informasi lebih rinci tentang mengapa operasi gagal.



Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200002e	STATUS_STORE_OUT_OF_MEMORY	Toko konten kehabisan memori.	Toko konten dibagi antara aliran dan harus memiliki kapasitas yang cukup untuk menyimpan durasi maksimum untuk semua aliran + ~ 20% (akuntansi untuk defragmentasi). Penting untuk tidak meluap penyimpanan. Pilih nilai untuk durasi maksimum per aliran yang sesuai dengan ukuran penyimpanan kumulatif dan toleransi latensi. Kami merekomendasikan untuk menjatuhkan bingkai saat jatuh dari jendela tampilan konten versus hanya diletakkan (tekanan memori penyimpanan konten). Ini karena menjatuhkan frame akan memulai callback notifikasi tekanan aliran. Kemudian aplikasi dapat menyesuaikan komponen media hulu (seperti encoder) untuk menipiskan bitrate, menjatuhkan bingkai, atau bertindak sesuai dengan itu.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	Tidak ada lagi data yang tersedia saat ini untuk streaming.	Ini adalah hasil potensial yang valid ketika pipa media menghasilkan lebih lambat daripada thread jaringan mengkonsumsi frame yang akan dikirim ke layanan. Klien tingkat yang lebih tinggi (misalnya, C ++, Java, atau Android) tidak melihat peringatan ini karena ditangani secara internal.
0x52000030	STATUS_INVALID_TAG_VERSION	Versi Tag struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x52000031	STATUS_SERVICE_CALL_UNKNOWN_ERROR	Kesalahan yang tidak diketahui atau umum dikembalikan dari tumpukan jaringan.	Lihat log untuk informasi lebih rinci.
0x52000032	STATUS_SERVICE_CALL_RESOURCE_IN_USE_ERROR	Sumber daya yang digunakan.	Kembali dari layanan. Untuk informasi selengkapnya, lihat Referensi API Kinesis Video Streams.
0x52000033	STATUS_SERVICE_CALL_CLIENT_LIMIT_ERROR	Batas klien.	Kembali dari layanan. Untuk informasi selengkapnya, lihat Referensi API Kinesis Video Streams.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000034	STATUS_SERVICE_CALL_DEVICE_LIMIT_ERROR	Batas perangkat.	Kembali dari layanan. Untuk informasi selengkapnya, lihat Referensi API Kinesis Video Streams.
0x52000035	STATUS_SERVICE_CALL_STREAM_LIMIT_ERROR	Batas aliran.	Kembali dari layanan. Untuk informasi selengkapnya, lihat Referensi API Kinesis Video Streams.
0x52000036	STATUS_SERVICE_CALL_RESOURCE_DELETED_ERROR	Sumber daya telah dihapus atau sedang dihapus.	Kembali dari layanan. Untuk informasi selengkapnya, lihat Referensi API Kinesis Video Streams.
0x52000037	STATUS_SERVICE_CALL_TIMEOUT_ERROR	Waktu panggilan layanan habis.	Memanggil API layanan tertentu menghasilkan batas waktu. Verifikasi bahwa Anda memiliki koneksi jaringan yang valid. PIC akan mencoba kembali operasi secara otomatis.
0x52000038	STATUS_STREAM_READ_CALLBACK_FAILED	Streaming pemberitahuan siap.	Pemberitahuan ini dikirim dari PIC ke klien yang menunjukkan bahwa aliran async telah dibuat.
0x52000039	STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	Tag tidak valid ditentukan.	Jumlah tag tidak nol, tetapi tag kosong. Verifikasi bahwa tag ditentukan atau hitungannya nol.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	Versi StreamDescription struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	Panjang nama tag tidak valid.	Lihat batas untuk nama tag yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	Panjang nilai tag tidak valid.	Lihat batas untuk nilai tag yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x5200003d	STATUS_TAG_RESOURCE_STREAM_CALL_FAILED	TagResource API gagal.	Panggilan TagResource API gagal. Periksa koneksi jaringan yang valid. Lihat log untuk informasi lebih lanjut tentang kegagalan.
0x5200003e	STATUS_INVALID_CUSTOM_DATA	Data kustom yang memanggil PIC API tidak valid.	Data kustom tidak valid telah ditentukan dalam panggilan ke PIC API. Ini hanya dapat terjadi pada klien yang langsung menggunakan PIC.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	Struktur tidak validCreateStreamResponse .	Struktur atau bidang anggotanya tidak valid (yaitu, ARN adalah nol atau lebih besar dari yang ditentukan dalam). <a href="#">Batas SDK produsen</a>

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	Autentikasi klien gagal.	PIC gagal mendapatkan informasi autentikasi yang tepat (AccessKey Id atau SecretAccessKey ) setelah sejumlah percobaan ulang. Periksa integrasi otentikasi. Contoh aplikasi menggunakan variabel lingkungan untuk meneruskan informasi kredensi ke C++ Producer Library.
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	Mendapatkan panggilan token keamanan gagal.	Situasi ini dapat terjadi untuk klien yang menggunakan PIC secara langsung. Setelah beberapa kali mencoba lagi, panggilan gagal dengan kesalahan ini.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	Kesalahan penyediaan.	Penyediaan tidak diterapkan.
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	Gagal membuat klien produser.	Kesalahan umum yang dikembalikan oleh PIC setelah sejumlah percobaan ulang saat pembuatan klien gagal.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	Gagal membawa klien produser ke status READY.	Dikembalikan oleh mesin status PIC jika PIC gagal pindah ke status READY. Lihat log untuk informasi lebih lanjut tentang akar penyebabnya.
0x52000045	STATUS_TAG_RESOURCE_CLIENT_CALL_FAILED	TagResource Untuk klien produser gagal.	Panggilan TagResource API gagal untuk klien produser. Lihat log untuk informasi lebih lanjut tentang akar penyebabnya.
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	Pembuatan Perangkat/Produser gagal.	SDK tingkat yang lebih tinggi (misalnya, C++ atau Java) belum mengimplementasikan API pembuatan perangkat atau produser. Klien yang menggunakan PIC secara langsung dapat menunjukkan kegagalan menggunakan notifikasi hasil.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	Stempel waktu ACK yang diterima tidak dalam tampilan.	Kesalahan ini terjadi jika bingkai yang sesuai dengan ACK yang diterima jatuh dari jendela tampilan konten. Umumnya, ini terjadi jika pengiriman ACK lambat. Hal ini dapat diartikan sebagai peringatan dan indikasi bahwa downlink lambat.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	Versi FragmentAck struktur tidak valid.	Tentukan versi FragmentAck struktur yang benar saat ini.
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	Kedaluwarsa token keamanan tidak valid.	Kedaluwarsa token keamanan harus memiliki stempel waktu absolut di masa depan yang lebih besar dari stempel waktu saat ini, dengan masa tenggang. Untuk batas masa tenggang, lihat <a href="#">Batas SDK produsen</a> .

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200004a	STATUS_END_OF_STREAM	Indikator akhir aliran (EOS).	Dalam panggilan <code>GetStreamData</code> API, menunjukkan bahwa sesi penanganan unggahan saat ini telah berakhir. Ini terjadi jika sesi berakhir atau kesalahan, atau jika token sesi telah kedaluwarsa dan sesi sedang diputar.
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	Nama aliran duplikat.	Beberapa aliran tidak dapat memiliki nama aliran yang sama. Pilih nama unik untuk streaming.
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	Periode retensi tidak valid.	Periode retensi yang tidak valid ditentukan dalam struktur <code>StreamInfo</code> . Untuk informasi tentang rentang nilai yang valid untuk periode retensi, lihat <a href="#">Batas SDK produsen</a> .
0x5200004d	STATUS_INVALID_ACK_KEY_START	Tidak valid <code>FragmentAck</code> .	Gagal mengurai string ACK fragmen. Indikator awal kunci tidak valid. String ACK fragmen mungkin rusak. Ini dapat memperbaiki diri sendiri dan kesalahan ini dapat diperlakukan sebagai peringatan.



Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200004e	STATUS_INVALID_ACK_DUPLICATE_KEY_NAME	Tidak validFragmentAck .	Gagal mengurai string ACK fragmen. Beberapa kunci memiliki nama yang sama. String ACK fragmen mungkin rusak. Ini dapat memperbaiki diri sendiri dan kesalahan ini dapat diperlakukan sebagai peringatan.
0x5200004f	STATUS_INVALID_ACK_INVALID_VALUE_START	Tidak validFragmentAck .	Gagal mengurai string ACK fragmen karena indikator awal nilai kunci yang tidak valid. String ACK fragmen mungkin rusak. Itu bisa memperbaiki diri sendiri, dan kesalahan ini dapat diperlakukan sebagai peringatan.
0x52000050	STATUS_INVALID_ACK_INVALID_VALUE_END	Tidak validFragmentAck .	Gagal mengurai string ACK fragmen karena indikator akhir nilai kunci yang tidak valid. String ACK fragmen mungkin rusak. Ini dapat memperbaiki diri sendiri dan kesalahan ini dapat diperlakukan sebagai peringatan.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000051	STATUS_INVALID_PARSED_ACK_TYPE	Tidak validFragmentAck .	Gagal mengurai string ACK fragmen karena tipe ACK yang tidak valid ditentukan.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	Stream dihentikan.	Aliran telah dihentikan, tetapi sebuah bingkai masih dimasukkan ke dalam sungai.
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	Versi StreamMetrics struktur tidak valid.	Tentukan versi StreamMetrics struktur yang benar saat ini.
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	Versi ClientMetrics struktur tidak valid.	Tentukan versi ClientMetrics struktur yang benar saat ini.
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	Inisialisasi produser gagal mencapai status READY.	Gagal mencapai status READY selama inisialisasi klien produser. Lihat log untuk informasi lebih lanjut.
0x52000056	STATUS_MACHINE_STATE_NOT_FOUND	Kesalahan mesin keadaan internal.	Bukan kesalahan yang terlihat publik.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	Jenis ACK tidak valid ditentukan dalam struktur. FragmentAck	FragmentAck Struktur harus berisi tipe ACK yang didefinisikan di header publik.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000058	STATUS_INVALID_STREAM_READY_STATE	Kesalahan transisi mesin keadaan internal.	Bukan kesalahan yang terlihat publik.
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	Objek aliran dibebaskan setelah produser dibebaskan.	Ada upaya untuk membebaskan objek aliran setelah objek produser dibebaskan. Ini hanya dapat terjadi pada klien yang langsung menggunakan PIC.
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	Kesalahan penyimpanan internal.	Kesalahan internal yang menunjukkan bahwa ukuran alokasi aktual dari toko konten lebih kecil dari ukuran bingkai dan fragmen yang dikemas.
0x5200005b	STATUS_VIDEO_ITEM_SIZE_GREATER_THAN_ALLOCATION	Kesalahan penyimpanan internal.	Ukuran alokasi yang disimpan dalam tampilan konten lebih besar dari ukuran alokasi di toko konten.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200005c	STATUS_ACK_ERR_STREAM_READ_ERROR	Streaming kesalahan baca ACK.	Kesalahan yang dikembalikan ACK dari backend yang menunjukkan kesalahan pembacaan atau penguraian aliran. Ini umumnya terjadi ketika backend gagal mengambil aliran. Auto-restreaming biasanya dapat memperbaiki kesalahan ini.
0x5200005d	STATUS_ACK_ERR_FRAGMENT_SIZE_REACHED	Ukuran fragmen maksimum tercapai.	Ukuran fragmen maks dalam byte didefinisikan dalam <a href="#">Batas SDK produsen</a> Kesalahan ini menunjukkan bahwa ada bingkai yang sangat besar, atau tidak ada bingkai kunci untuk membuat fragmen ukuran yang dapat dikelola. Periksa pengaturan encoder dan verifikasi bahwa bingkai kunci sedang diproduksi dengan benar. Untuk aliran yang memiliki kepadatan sangat tinggi, konfigurasi encoder untuk menghasilkan fragmen pada durasi yang lebih kecil untuk mengelola ukuran maksimum.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200005e	STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED	Durasi fragmen maksimum tercapai.	Durasi fragmen maks didefinisikan dalam <a href="#">Batas SDK produsen</a> . Kesalahan ini menunjukkan bahwa ada frame yang sangat rendah per detik atau tidak ada bingkai kunci untuk membuat fragmen durasi yang dapat dikelola. Periksa pengaturan encoder dan verifikasi bahwa bingkai kunci sedang diproduksi dengan benar pada interval reguler.
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	Durasi koneksi maksimum tercapai.	Kinesis Video Streams memberlakukan durasi koneksi maksimal seperti yang ditentukan dalam file. <a href="#">Batas SDK produsen</a> Producer SDK secara otomatis memutar aliran atau token sebelum maksimum tercapai. Klien yang menggunakan SDK seharusnya tidak menerima kesalahan ini.
0x52000060	STATUS_ACK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC	Kode waktu tidak meningkat secara monoton.	Producer SDK memberlakukan stempel waktu, sehingga klien yang menggunakan SDK seharusnya tidak menerima kesalahan ini.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000061	STATUS_AC K_ERR_MUL TI_TRACK_MKV	Beberapa trek di MKV.	Producer SDK memberlakukan stream trek tunggal, sehingga klien yang menggunakan SDK seharusnya tidak menerima kesalahan ini.
0x52000062	STATUS_AC K_ERR_INV ALID_MKV_DATA	Data MKV tidak valid.	Pengurai MKV backend mengalami kesalahan saat mengurai aliran. Klien yang menggunakan SDK mungkin mengalami kesalahan ini jika aliran rusak dalam transisi. Ini juga dapat terjadi jika tekanan buffer memaksa SDK untuk menjatuhkan bingkai ekor yang ditransmisikan sebagian. Dalam kasus terakhir, kami menyarankan Anda mengurangi FPS dan resolusi, meningkatkan rasio kompresi, atau (jika ada jaringan “bursty”) memungkinkan penyimpanan konten yang lebih besar dan durasi buffer untuk mengakomodasi tekanan sementara.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	Stempel waktu produsen tidak valid.	Layanan mengembalikkan kesalahan ACK ini jika jam produsen memiliki penyimpangan besar ke masa depan. SDK tingkat yang lebih tinggi (misalnya, Java atau C++) menggunakan beberapa versi jam sistem untuk memenuhi panggilan balik waktu saat ini dari PIC. Verifikasi bahwa jam sistem diatur dengan benar. Klien yang menggunakan PIC secara langsung harus memverifikasi bahwa fungsi callback mereka mengembalikkan stempel waktu yang benar.
0x52000064	STATUS_ACK_ERR_STREAM_NOT_ACTIVE	Aliran tidak aktif.	Panggilan ke API backend dibuat saat aliran tidak dalam status "Aktif". Ini terjadi ketika klien membuat aliran dan segera terus mendorong bingkai ke dalamnya. SDK menangani skenario ini melalui mesin status dan mekanisme pemulihan.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000065	STATUS_AKERR_KMS_KEY_ACCESS_DENIED	AWS KMS akses ditolak kesalahan.	Dikembalikan ketika akun tidak memiliki akses ke kunci yang ditentukan.
0x52000066	STATUS_AKERR_KMS_KEY_DISABLED	AWS KMS kunci dinonaktifkan.	Kunci yang ditentukan telah dinonaktifkan.
0x52000067	STATUS_AKERR_KMS_KEY_VALIDATION_ERROR	AWS KMS kesalahan validasi kunci.	Kesalahan validasi generik. Untuk informasi lebih lanjut, lihat Referensi API <a href="#">AWS Key Management Service</a> .
0x52000068	STATUS_AKERR_KMS_KEY_UNAVAILABLE	AWS KMS key tidak tersedia.	Kuncinya tidak tersedia. Untuk informasi lebih lanjut, lihat Referensi API <a href="#">AWS Key Management Service</a> .
0x52000069	STATUS_AKERR_KMS_KEY_INVALID_USAGE	Penggunaan kunci KMS tidak valid.	Tidak AWS KMS key dikonfigurasi untuk digunakan dalam konteks ini. Untuk informasi lebih lanjut, lihat Referensi API <a href="#">AWS Key Management Service</a> .
0x5200006a	STATUS_AKERR_KMS_KEY_INVALID_STATE	AWS KMS keadaan tidak valid.	Untuk informasi lebih lanjut, lihat Referensi API <a href="#">AWS Key Management Service</a> .



Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200006b	STATUS_AK K_ERR_KMS _KEY_NOT_FOUND	Kunci KMS tidak ditemukan.	Kuncinya tidak ditemukan . Untuk informasi lebih lanjut, lihat Referensi API <a href="#">AWS Key Management Service</a> .
0x5200006c	STATUS_AK K_ERR_STR EAM_DELETED	Streaming telah atau sedang dihapus.	Aliran sedang dihapus oleh aplikasi lain atau melalui fileAWS Management Console.
0x5200006d	STATUS_AK K_ERR_ACK _INTERNAL_ERROR	Kesalahan internal.	Kesalahan internal layanan generik.
0x5200006e	STATUS_AK K_ERR_FRA GMENT_ARC HIVAL_ERROR	Kesalahan arsip fragmen.	Dikembalikan ketika layanan gagal bertahan lama dan mengindeks fragmen. Meskipun jarang, itu bisa terjadi karena berbagai alasan. Secara default, SDK mencoba mengirim fragmen lagi.
0x5200006f	STATUS_AK K_ERR_UNK NOWN_ACK_ERROR	Kesalahan tidak diketahui.	Layanan mengembalikan kesalahan yang tidak diketahui.
0x52000070	STATUS_MI SSING_ERR_ACK_ID	Informasi ACK hilang.	Parser ACK menyelesaikan penguraian, tetapi FragmentAck informasinya hilang.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	Panjang segmen ACK tidak valid.	String segmen ACK dengan panjang tidak valid ditentukan ke parser ACK. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x52000074	STATUS_MAXIMUM_FRAGMENT_METADATA_COUNT	Jumlah maksimum item metadata telah ditambahkan ke sebuah fragmen.	Aliran video Kinesis dapat menambahkan hingga 10 item metadata ke fragmen, baik dengan menambahkan item nonpersisten ke fragmen, atau dengan menambahkan item persisten ke antrian metadata. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan metadata streaming dengan Kinesis Video Streams</a> .
0x52000075	STATUS_ACK_FRAGMENT_METADATA_LIMIT_REACHED	Batas (jumlah metadata maksimum, panjang nama metadata, atau panjang nilai metadata) telah tercapai.	Producer SDK membatasi jumlah dan ukuran item metadata. Kesalahan ini tidak terjadi kecuali batas dalam kode Producer SDK diubah. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan metadata streaming dengan Kinesis Video Streams</a> .

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000076	STATUS_BLOCKING_INTERRUPTED_STREAM_TERMINATED	Tidak diimplementasikan.	
0x52000077	STATUS_INVALID_METADATA_NAME	Nama metadata tidak valid.	Nama metadata tidak dapat dimulai dengan string ""AWS. Jika kesalahan ini terjadi, item metadata tidak ditambahkan ke fragmen atau antrian metadata. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan metadata streaming dengan Kinesis Video Streams</a> .
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	Akhir dari frame fragmen berada dalam keadaan tidak valid.	Akhir fragmen tidak boleh dikirim dalam aliran yang non-key-frame terfragmentasi.
0x52000079	STATUS_TRACK_INFO_MISSING	Informasi trek tidak ada.	Nomor trek harus lebih besar dari nol dan cocok dengan id trek.
0x5200007a	STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED	Jumlah trek maksimum terlampaui.	Anda dapat memiliki maksimal tiga trek per aliran.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x5200007b	STATUS_OF FLINE_MOD E_WITH_ZE RO_RETENTION	Waktu retensi mode streaming offline diatur ke nol.	Waktu retensi mode streaming offline tidak boleh disetel ke nol.
0x5200007c	STATUS_AC K_ERR_TRA CK_NUMBER _MISMATCH	Nomor trek dari kesalahan ACK tidak cocok.	
0x5200007d	STATUS_AC K_ERR_FRA MES_MISSI NG_FOR_TRACK	Bingkai hilang untuk trek.	
0x5200007e	STATUS_AC K_ERR_MOR E_THAN_AL LOWED_TRA CKS_FOUND	Jumlah trek maksimum yang diizinkan terlampaui.	
0x5200007f	STATUS_UP LOAD_HAN DLE_ABORTED	Pegangan unggahan dibatalkan.	
0x52000080	STATUS_IN VALID_CER T_PATH_LENGTH	Panjang jalur sertifikat tidak valid.	
0x52000081	STATUS_DU PLICATE_T RACK_ID_FOUND	ID trek duplikat ditemukan.	
0x52000082	STATUS_IN VALID_CLI ENT_INFO_VERSION		

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000083	STATUS_INVALID_CLIENT_ID_STRING_LENGTH		
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUMEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x52000089	STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS		
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_METADATA_VALUE_LENGTH		


Kode kesalahan dan status yang dikembalikan oleh pustaka durasi

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di `Duration` perpustakaan.

Code	Pesan
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

Kode kesalahan dan status yang dikembalikan oleh pustaka umum

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di Common perpustakaan.

 Note

Kode informasi kesalahan dan status ini umum untuk banyak API.

Code	Pesan	Deskripsi
0x00000001	STATUS_NULL_ARG	NULL disahkan untuk argumen wajib.
0x00000002	STATUS_INVALID_ARG	Nilai yang tidak valid ditentukan untuk argumen.
0x00000003	STATUS_INVALID_ARG_LEN	Panjang argumen yang tidak valid ditentukan.
0x00000004	STATUS_NOT_ENOUGH_MEMORY	Tidak dapat mengalokasikan cukup memori.
0x00000005	STATUS_BUFFER_TOO_SMALL	Ukuran buffer yang ditentukan terlalu kecil.
0x00000006	STATUS_UNEXPECTED_EOF	Akhir file yang tidak terduga tercapai.
0x00000007	STATUS_FORMAT_ERROR	Format yang tidak valid ditemukan.

Code	Pesan	Deskripsi
0x00000008	STATUS_INVALID_HANDLE_ERROR	Nilai pegangan tidak valid.
0x00000009	STATUS_OPEN_FILE_FAILED	Gagal membuka file.
0x0000000A	STATUS_READ_FILE_FAILED	Gagal membaca dari file.
0x0000000b	STATUS_WRITE_TO_FILE_FAILED	Gagal menulis ke file.
0x0000000c	STATUS_INTERNAL_ERROR	Kesalahan internal yang biasanya tidak terjadi dan mungkin menunjukkan bug SDK atau API layanan.
0x0000000d	STATUS_INVALID_OPERATION	Ada operasi yang tidak valid, atau operasi tidak diizinkan.
0x0000000e	STATUS_NOT_IMPLEMENTED	Fitur ini tidak diimplementasikan.
0x0000000f	STATUS_OPERATION_TIMED_OUT	Waktu operasi habis.
0x00000010	STATUS_NOT_FOUND	Sumber daya yang dibutuhkan tidak ditemukan.
0x00000011	STATUS_CREATE_THREAD_FAILED	Gagal membuat thread.
0x00000012	STATUS_THREAD_NOT_ENOUGH_RESOURCES	Sumber daya yang tidak mencukupi untuk membuat utas lain, atau batas yang diberlakukan sistem pada jumlah utas ditemukan.



Code	Pesan	Deskripsi
0x00000013	STATUS_THREAD_INVALID_ARG	Atribut thread tidak valid ditentukan, atau thread lain sudah menunggu untuk bergabung dengan thread ini.
0x00000014	STATUS_THREAD_PERMISSIONS	Tidak ada izin untuk mengatur kebijakan penjadwalan dan parameter yang ditentukan dalam atribut thread.
0x00000015	STATUS_THREAD_DEADLOCKED	Kebuntuan terdeteksi atau utas yang bergabung menentukan utas panggilan.
0x00000016	STATUS_THREAD_DOES_NOT_EXIST	Tidak dapat menemukan utas dengan ID utas yang ditentukan.
0x00000017	STATUS_JOIN_THREAD_FAILED	Kesalahan yang tidak diketahui atau umum dikembalikan dari operasi gabungan thread.
0x00000018	STATUS_WAIT_FAILED	Melebihi waktu maksimum untuk menunggu variabel bersyarat.
0x00000019	STATUS_CANCEL_THREAD_FAILED	Kesalahan yang tidak diketahui atau umum dikembalikan dari operasi pembatalan utas.
0x0000001A	STATUS_THREAD_IS_NOT_JOINABLE	Operasi gabungan utas diminta pada utas yang tidak dapat disambung.

Code	Pesan	Deskripsi
0x0000001b	STATUS_DETACH_THREAD_FAILED	Kesalahan yang tidak diketahui atau umum dikembalikan dari operasi pelepasan utas.
0x0000001c	STATUS_THREAD_ATTR_INIT_FAILED	Gagal menginisialisasi objek atribut thread.
0x0000001d	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	Gagal mengatur ukuran tumpukan untuk objek atribut thread.
0x0000001e	STATUS_MEMORY_NOT_FREED	Hanya digunakan dalam tes. Menunjukkan bahwa tidak semua memori yang diminta telah dibebaskan.
0x10000015	STATUS_INVALID_ALLLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

Kode kesalahan dan status yang dikembalikan oleh pustaka heap

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di Heap perpustakaan.

Code	Pesan	Deskripsi
0x01000001	STATUS_HEAP_FLAGS_ERROR	Kombinasi bendera yang tidak valid telah ditentukan.

Code	Pesan	Deskripsi
0x01000002	STATUS_HEAP_NOT_INITIALIZED	Sebuah operasi dicoba sebelum tumpukan diinisialisasi.
0x01000003	STATUS_HEAP_CORRUPTED	Tumpukan rusak atau band penjaga (dalam mode debug) ditimpa. Buffer overflow dalam kode klien dapat menyebabkan korupsi heap.
0x01000004	STATUS_HEAP_VRAM_LIB_MISSING	Pustaka mode pengguna atau kernel VRAM (RAM video) tidak dapat dimuat atau hilang. Periksa apakah platform yang mendasarinya mendukung alokasi VRAM.
0x01000005	STATUS_HEAP_VRAM_LIB_REOPEN	Gagal membuka pustaka VRAM.
0x01000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	Gagal memuat ekspor fungsi INIT.
0x01000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	Gagal memuat ekspor ALLOC fungsi.
0x01000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	Gagal memuat ekspor FREE fungsi.
0x01000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	Gagal memuat ekspor LOCK fungsi.
0x0100000A	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	Gagal memuat ekspor UNLOCK fungsi.
0x0100000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	Gagal memuat ekspor UNINIT fungsi.

Code	Pesan	Deskripsi
0x0100000C	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	Gagal memuat ekspor GETMAX fungsi.
0x0100000d	STATUS_HEAP_DIRECT_MEM_INIT	Gagal menginisialisasi kumpulan heap utama di heap hybrid.
0x0100000E	STATUS_HEAP_VRAM_INIT_FAILED	Inisialisasi dinamis VRAM gagal.
0x0100000f	STATUS_HEAP_LIBRARY_FREE_FAILED	Gagal menghapus alokasi dan membebaskan pustaka VRAM.
0x01000010	STATUS_HEAP_VRAM_ALLOC_FAILED	Alokasi VRAM gagal.
0x01000011	STATUS_HEAP_VRAM_FREE_FAILED	Gratis VRAM gagal.
0x01000012	STATUS_HEAP_VRAM_MAP_FAILED	Peta VRAM gagal.
0x01000013	STATUS_HEAP_VRAM_UNMAP_FAILED	Unmap VRAM gagal.
0x01000014	STATUS_HEAP_VRAM_UNINIT_FAILED	Deinisialisasi VRAM gagal.

Kode kesalahan dan status yang dikembalikan oleh pustaka MKVGen

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di MKVGen perpustakaan.

Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Anggota struktur Frame data yang tidak valid. Pastikan bahwa durasi, ukuran, dan data bingkai valid dan berada dalam batas yang ditentukan dalam <a href="#">Batas SDK produsen</a> .
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	Stempel waktu bingkai tidak valid. PTS yang dihitung (stempel waktu presentasi) dan DTS (stempel waktu decoding) lebih besar atau sama dengan stempel waktu dari kerangka awal fragmen. Ini adalah indikasi potensi restart pipa media atau masalah stabilitas encoder. Untuk informasi pemecahan masalah, lihat <a href="#">Kesalahan: “Gagal mengirimkan bingkai ke klien Video Kinesis”</a>
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	Durasi fragmen yang tidak valid telah ditentukan. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	Panjang string tipe konten tidak valid. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	Ada upaya untuk menyandikan nomor yang terlalu besar

Code	Pesan	Deskripsi/ Tindakan yang disarankan
		untuk direpresentasikan dalam format EBMM (Extensible Binary Meta Language). Ini tidak boleh diekspos ke klien SDK.
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	Panjang string ID codec tidak valid. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	Panjang string nama trek tidak valid. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	Panjang data pribadi codec tidak valid. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	Data pribadi codec (CPD) adalah NULL, sedangkan ukuran CPD lebih besar dari nol.
0x3200000A	STATUS_MKV_INVALID_TIMECODE_SCALE	Nilai skala kode waktu tidak valid. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	Kode waktu bingkai lebih besar dari maksimum. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .

Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	Kode waktu bingkai maks tercapai. Format MKV menggunakan 16 bit yang ditandatangani untuk mewakili kode waktu relatif dari frame ke awal cluster. Kesalahan dihasilkan jika kode waktu bingkai tidak dapat direpresentasikan. Kesalahan ini menunjukkan pemilihan skala kode waktu yang buruk atau durasi cluster terlalu panjang, sehingga mewakili kode waktu bingkai meluap ruang 16-bit yang ditandatangani.

Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x3200000d	STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA	Kode awal Lampiran-B yang tidak valid ditemukan . Misalnya, flag adaptasi Annex-B ditentukan dan kode menemukan urutan awal yang tidak valid lebih dari tiga nol. Format Lampiran-B yang valid harus memiliki urutan “pencegahan emulasi” untuk menghindari urutan tiga atau lebih nol dalam bytestream. Untuk informasi lebih lanjut, lihat spesifikasi MPEG. Untuk informasi tentang kesalahan ini di Android, lihat <a href="#">Kesalahan STATUS_MKV_INVALID _ANNEXB_NALU_IN_FR AME_DATA (0x3200000d) di Android</a> .
0x3200000e	STATUS_MKV_INVALID _AVCC_NALU_IN_FRAM E_DATA	Kemasan AVCC NALU tidak valid saat flag AVCC yang mengadaptasi ditentukan. Verifikasi bahwa bytestream dalam format AVCC yang valid. Untuk informasi lebih lanjut, lihat spesifikasi MPEG.
0x3200000f	STATUS_MKV_BOTH_AN NEXB_AND_AVCC_SPEC IFIED	Baik NALU AVCC dan Annex-B yang mengadaptasi ditentukan. Tentukan salah satu, atau tentukan tidak ada.



Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x32000010	STATUS_MKV_INVALID _ANNEXB_NALU_IN_CPD	Format Lampiran-B CPD tidak valid saat flag Annex-B yang mengadaptasi ditentukan. Verifikasi bahwa CPD dalam format Lampiran-B yang valid. Jika tidak, hapus flag adaptasi CPD Annex-B.
0x32000011	STATUS_MKV_PTS_DTS _ARE_NOT_SAME	Kinesis Video Streams memaksa PTS (stempel waktu presentasi) dan DTS (decoding timestamp) menjadi sama untuk frame awal fragmen. Ini adalah bingkai kunci yang memulai fragmen.
0x32000012	STATUS_MKV_INVALID _H264_H265_CPD	Gagal mengurai data pribadi codec H264/H265.
0x32000013	STATUS_MKV_INVALID _H264_H265_SPS_WIDTH	Gagal mengekstrak lebar dari data pribadi codec.
0x32000014	STATUS_MKV_INVALID _H264_H265_SPS_HEIGHT	Gagal mengekstrak ketinggian dari data pribadi codec.
0x32000015	STATUS_MKV_INVALID _H264_H265_SPS_NALU	H264/H265 SPS NALU tidak valid.
0x32000016	STATUS_MKV_INVALID _BIH_CPD	Format header info bitmap tidak valid dalam data pribadi codec.

Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	Jumlah unit Lapisan Abstraksi Jaringan (NALU) Pengkodean Video Efisiensi Tinggi (HEVC) Tidak Valid.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	Format HEVC tidak valid.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	Hilang HEVC NALUS dalam Sequence Parameter Set (SPS).
0x3200001A	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	Ukuran HEVC SPS NALU tidak valid.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	IDC format Chroma tidak valid.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	SPS cadangan HEVC tidak valid.
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	Ukuran nilai beta pribadi AnnexBb codec minimum. Untuk H264, nilai ini harus sama dengan atau lebih besar dari 11. Untuk H265, nilai ini harus sama dengan atau lebih besar dari 15.
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Data pribadi codec hilang di Annex-B NALUS.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Beta pribadi codec tidak valid di NaluS Annex-B.

Code	Pesan	Deskripsi/ Tindakan yang disarankan
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	Panjang nama tag tidak valid. Nilai valid lebih besar dari nol dan kurang dari 128.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	Panjang nilai tag tidak valid. Nilai valid lebih besar dari nol dan kurang dari 256.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	Tag status generator tidak valid.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	Indeks frekuensi pengambilan sampel data pribadi codec AAC tidak valid.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	Konfigurasi saluran data pribadi codec AAC tidak valid.
0x32000025	STATUS_MKV_INVALID_AAC_CPD	Data pribadi codec AAC tidak valid.
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	Lacak informasi tidak ditemukan.
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	UUID segmen tidak valid.
0x32000028	STATUS_MKV_INVALID_TRACK_UID	UID trek tidak valid.
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	

Code	Pesan	Deskripsi/ Tindakan yang disarankan
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	

Kode kesalahan dan status yang dikembalikan oleh pustaka Trace

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di Trace perpustakaan.

Code	Pesan
0x10100001	STATUS_MIN_PROFILER_BUFFER

Kode kesalahan dan status yang dikembalikan oleh pustaka Utils

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di Utils perpustakaan.

Code	Pesan
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW

Code	Pesan
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED
0x40000011	STATUS_UTIL_MAX_TAG_COUNT
0x40000012	STATUS_UTIL_INVALID_TAG_VERSION
0x40000013	STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL
0x40000014	STATUS_UTIL_INVALID_TAG_NAME_LEN
0x40000015	STATUS_UTIL_INVALID_TAG_VALUE_LEN

Code	Pesan
0x4000002a	STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE
0x4000002b	STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED
0x4000002c	STATUS_THREADPOOL_MAX_COUNT
0x4000002d	STATUS_THREADPOOL_INTERNAL_ERROR
0x40100001	STATUS_HASH_KEY_NOT_PRESENT
0x40100002	STATUS_HASH_KEY_ALREADY_PRESENT
0x40100003	STATUS_HASH_ENTRY_ITERATION_ABORT
0x41000001	STATUS_BIT_READER_OUT_OF_RANGE
0x41000002	STATUS_BIT_READER_INVALID_SIZE
0x41100001	STATUS_TIMER_QUEUE_STOP_SCHEDULING
0x41100002	STATUS_INVALID_TIMER_COUNT_VALUE
0x41100003	STATUS_INVALID_TIMER_PERIOD_VALUE
0x41100004	STATUS_MAX_TIMER_COUNT_REACHED
0x41100005	STATUS_TIMER_QUEUE_SHUTDOWN
0x41200001	STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN
0x41200002	STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED

Code	Pesan
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

Kode kesalahan dan status yang dikembalikan oleh pustaka View

Tabel berikut berisi informasi kesalahan dan status yang dikembalikan oleh metode di View perpustakaan.

Code	Pesan	Deskripsi
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	Jumlah item tampilan konten yang tidak valid telah ditentukan. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	Durasi tampilan konten yang tidak valid telah ditentukan. Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	Upaya dilakukan untuk melewati posisi kepala.
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	Indeks yang tidak valid ditentukan.
0x30000005	STATUS_CONTENT_VIEW_INVALID_TIMESTAMP	Ada stempel waktu yang tidak valid atau tumpang tindih stempel waktu. Stempel waktu decoding bingkai harus lebih besar dari atau sama dengan stempel waktu bingkai sebelumnya, ditambah durasi bingkai sebelumnya: $\text{DTS}(n) \geq$

Code	Pesan	Deskripsi
		<p><math>DTS(n-1) + Duration(n-1)</math> Kesalahan ini sering menunjukkan encoder “tidak stabil”. Encoder menghasilkan ledakan frame yang dikodekan, dan stempel waktunya lebih kecil dari durasi intra-frame. Atau aliran dikonfigurasi untuk menggunakan stempel waktu SDK, dan frame dikirim lebih cepat dari durasi frame. Untuk membantu dengan beberapa “jitter” di encoder, tentukan durasi bingkai yang lebih kecil dalam struktur. <code>StreamInfo.StreamCaps</code> Misalnya, jika alirannya 25 FPS, durasi setiap frame adalah 40 ms. Namun, untuk menangani encoder “jitter”, kami sarankan Anda menggunakan setengah dari durasi frame tersebut (20 ms). Beberapa aliran memerlukan kontrol yang lebih tepat atas waktu untuk deteksi kesalahan.</p>
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	Panjang data item tampilan konten yang tidak valid telah ditentukan.



## Kesalahan dan kode status yang dikembalikan oleh PutFrame callback - pustaka produser C

Bagian berikut berisi informasi kesalahan dan status yang dikembalikan oleh callback untuk PutFrame operasi dalam pustaka produser C.

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x15000001	STATUS_STOP_CALLBACK_CHAIN	Rantai callback telah berhenti.	
0x15000002	STATUS_MAXIMUM_CALLBACK_CHAIN	Rantai callback maksimum tercapai.	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	Versi PlatformCallbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	Versi ProducerCallbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	Versi StreamCallbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	Versi AuthCallbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x15000007	STATUS_INVALID_API_CALLBACKS	Versi ApiCallbacks struktur tidak valid.	Tentukan versi struktur yang benar saat ini.

Code	Pesan	Deskripsi	Tindakan yang disarankan
	_CALLBACK S_VERSION		
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	Versi <code>AwsCredentials</code> struktur tidak valid.	Tentukan versi struktur yang benar saat ini.
0x15000009	STATUS_MAX_REQUEST_HEADER_COUNT	Jumlah header permintaan maksimum tercapai.	
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	Panjang nama header permintaan maksimum tercapai.	
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	Panjang nilai header permintaan maksimum tercapai.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	Pengembalian JSON tidak valid untuk panggilan API.	
0x1500000d	STATUS_CURL_INIT_FAILED	Inisialisasi curl gagal.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Inisialisasi curl lib gagal.	
0x1500000f	STATUS_INVALID_DESCRIBE_STREAM_RETURN_JSON	Pengembalian JSON tidak valid untuk <code>DescribeStream</code>	

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x15000010	STATUS_HMAC_GENERATION_ERROR	Kesalahan pembuatan HMAC.	
0x15000011	STATUS_IOT_FAILED	Otorisasi IoT gagal.	
0x15000012	STATUS_MAXIMUM_ALIAS_LENGTH_EXCEEDED	Panjang alias peran maksimum tercapai.	Tentukan panjang alias yang lebih pendek.
0x15000013	STATUS_MAXIMUM_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED	Panjang postfix nama agen maksimum tercapai.	
0x15000014	STATUS_MAXIMUM_CUSTOMER_AGENT_LENGTH_EXCEEDED	Panjang agen pengguna pelanggan maksimum tercapai.	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	Panjang agen pengguna tidak valid.	
0x15000016	STATUS_INVALID_CACHING_ENDPOINT_PERIOD	Periode caching endpoint tidak valid.	Tentukan periode caching yang kurang dari 24 jam.
0x15000017	STATUS_IOT_EXPIRATION_OCCURRED_IN_PAST	Stempel waktu kedaluwarsa IoT terjadi di masa lalu.	

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	Penguraian kedaluwarsa IoT telah gagal.	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNC		
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAXIMUM_IOT_THING_NAME_LENGTH		

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x1500001f	STATUS_IOT_CREATE_CONTENT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		

Code	Pesan	Deskripsi	Tindakan yang disarankan
0x15000028	STATUS_CO NTINUOUS_ RETRY_RES ET_FAILED		
0x16000001	STATUS_CU RL_PERFOR M_FAILED		
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		

## Referensi bendera adaptasi Network Abstraction Layer (NAL)

Bagian ini berisi informasi tentang bendera yang tersedia untuk `StreamInfo.NalAdaptationFlags` enumerasi.

[Aliran dasar](#) dalam aplikasi dapat dalam format Lampiran-B atau AVCC:

- Format Lampiran-B membatasi [NALUS \(unit Network Abstraction Layer\)](#) dengan dua byte nol, diikuti oleh satu atau tiga byte nol, diikuti oleh angka 1 (disebut kode awal, misalnya, 00000001).
- Format AVCC juga membungkus NALU, tetapi setiap NALU didahului oleh nilai yang menunjukkan ukuran NALU (biasanya empat byte).

Banyak encoder menghasilkan format bitstream Annex-B. Beberapa prosesor bitstream tingkat tinggi (seperti mesin pemutaran atau pemutar [Media Source Extensions \(MSE\)](#) diAWS Management Console) menggunakan format AVCC untuk frame mereka.

Data pribadi codec (CPD), yaitu SPS/PPS (Sequence Parameter Set/Picture Parameter Set) untuk codec H.264, juga dapat dalam format Lampiran-B atau AVCC. Namun, untuk CPD, formatnya berbeda dari yang dijelaskan sebelumnya.

Bendera memberi tahu SDK untuk mengadaptasi NALU ke AVCC atau Annex-B untuk data bingkai dan CPD sebagai berikut:

Bendera	Adaptasi	
NAL_ADAPTATION_FLAG_NONE	Tidak ada adaptasi.	
NAL_ADAPTATION_ANNEXB_NALS	Sesuaikan NALU Lampiran-B ke AVCC NALU.	
NAL_ADAPTATION_AVC_NALS	Sesuaikan NALU AVCC ke Nalus Annex-B.	
NAL_ADAPTATION_ANNEXB_CPD_NALS	Adaptasi NALUS Annex-B untuk data pribadi codec ke NALU format AVCC.	
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	Sesuaikan NALUS Annex-B untuk codec dan bingkai data pribadi ke NALU format AVCC.	

Untuk informasi selengkapnya tentang tipe NALU, lihat Bagian 1.3: Jenis Unit Lapisan Abstraksi Jaringan di [RFC 3984](#).

## Struktur SDK produsen

Bagian ini mencakup informasi tentang struktur yang dapat Anda gunakan untuk menyediakan data ke objek Kinesis Video Streams Producer.

## Topik

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

## DeviceInfo/DefaultDeviceInfoProvider

DefaultDeviceInfoProviderObjek DeviceInfodan mengontrol perilaku objek Kinesis Video Streams Producer.

### Bidang anggota

- versi — Nilai integer yang digunakan untuk memastikan bahwa versi struktur yang benar digunakan dengan versi basis kode saat ini. Versi saat ini ditentukan menggunakan `DEVICE_INFO_CURRENT_VERSION` makro.
- nama — Nama yang dapat dibaca manusia untuk perangkat.
- TagCount/tag — Saat ini tidak digunakan.
- StreamCount — Jumlah maksimum aliran yang dapat ditangani perangkat. Ini pra-mengalokasikan penyimpanan untuk pointer ke objek aliran pada awalnya, tetapi objek aliran yang sebenarnya dibuat kemudian. Defaultnya adalah 16 aliran, tetapi Anda dapat mengubah nomor ini dalam `DefaultDeviceInfoProvider.cpp` file.
- StorageInfo: Objek yang menjelaskan konfigurasi penyimpanan utama. Untuk informasi selengkapnya, lihat [StorageInfo](#).

## StorageInfo

Menentukan konfigurasi penyimpanan utama untuk Kinesis Video Streams.

Implementasi default didasarkan pada implementasi heap cepat fragmentasi rendah, yang dioptimalkan untuk streaming. Ini menggunakan `MEMALLOC` pengalokasi, yang dapat ditimpa pada platform tertentu. Beberapa platform memiliki alokasi memori virtual tanpa mendukung alokasi dengan halaman fisik. Saat memori digunakan, halaman virtual didukung oleh halaman fisik. Hal ini menyebabkan tekanan memori rendah pada keseluruhan sistem ketika penyimpanan kurang digunakan.

Hitung ukuran penyimpanan default berdasarkan rumus berikut. `DefragmentationFactor` harus diatur ke 1,2 (20 persen).



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

Dalam contoh berikut, perangkat memiliki aliran audio dan video. Aliran audio memiliki 512 sampel per detik, dengan sampel rata-rata 100 byte. Streaming video memiliki 25 frame per detik, dengan rata-rata 10.000 byte. Setiap aliran memiliki durasi buffer 3 menit.

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

Jika perangkat memiliki lebih banyak memori yang tersedia, kami sarankan Anda menambahkan lebih banyak memori ke penyimpanan untuk menghindari fragmentasi yang parah.

Verifikasi bahwa ukuran penyimpanan cukup untuk mengakomodasi buffer penuh untuk semua aliran pada kompleksitas pengkodean tinggi (ketika ukuran bingkai lebih besar karena gerakan tinggi) atau ketika bandwidth rendah. Jika produsen mencapai tekanan memori, ia memancarkan callback tekanan overflow penyimpanan (`StorageOverflowPressureFunc`). Namun, ketika tidak ada memori yang tersedia di toko konten, itu akan menjatuhkan bingkai yang didorong ke Kinesis Video Streams dengan error `STATUS_STORE_OUT_OF_MEMORY = 0x5200002e` (). Untuk informasi selengkapnya, lihat [Kode kesalahan dan status yang dikembalikan oleh pustaka klien](#). Ini juga dapat terjadi jika pengakuan aplikasi (ACK) tidak tersedia, atau ACK yang bertahan tertunda. Dalam hal ini, buffer mengisi kapasitas “durasi buffer” sebelum frame sebelumnya mulai putus.

## Bidang anggota

- `versi` — Nilai integer yang digunakan untuk memastikan bahwa versi struktur yang benar digunakan dengan versi basis kode saat ini.
- `StorageType` — Pencacahan yang menentukan dukungan dan implementasi penyimpanan `DEVICE_STORAGE_TYPE` yang mendasari. Satu-satunya nilai yang di-support saat ini adalah `DEVICE_STORAGE_TYPE_IN_MEM`. Implementasi future akan mendukung `DEVICE_STORAGE_TYPE_HYBRID_FILE`, menunjukkan bahwa penyimpanan kembali ke toko konten yang didukung file.
- `StorageSize` - Ukuran penyimpanan dalam byte untuk dialokasikan sebelumnya. Alokasi minimum adalah 10 MB, dan alokasi maksimum adalah 10 GB. (Ini akan berubah dengan implementasi penyimpanan konten yang didukung file di masa mendatang.)

- **SpillRatio** — Nilai integer yang mewakili persentase penyimpanan yang akan dialokasikan dari tipe penyimpanan memori langsung (RAM), sebagai lawan dari penyimpanan overflow sekunder (penyimpanan file). Saat ini tidak digunakan.
- **RootDirectory**: Jalur ke direktori tempat penyimpanan konten yang didukung file berada. Saat ini tidak digunakan.

## Struktur aliran video Kinesis

Anda dapat menggunakan struktur berikut untuk menyediakan data ke instance aliran video Kinesis.

Topik

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

### StreamDefinition/StreamInfo

StreamDefinitionObjek di lapisan C ++ membungkus StreamInfo objek dalam kode platform-independen, dan memberikan beberapa nilai default dalam konstruktor.

Bidang anggota

Field	Jenis data	Deskripsi	Nilai default
stream_name	string	Nama aliran opsional. Untuk informasi selengkapnya tentang panjang nama aliran, lihat <a href="#">Batas SDK produsen</a> . Setiap aliran harus memiliki nama yang unik.	Jika tidak ada nama yang ditentukan, nama dihasilkan secara acak.
retensi_periode	duration< uint64_t, ratio<3600>>	Periode retensi untuk aliran, dalam hitungan detik. Menentukan <b>0</b>	3600 (Satu jam)

Field	Jenis data	Deskripsi	Nilai default
		menunjukkan tidak ada retensi.	
tag	<code>const map&lt;string, string&gt;*</code>	Peta pasangan kunci-nilai yang berisi informasi pengguna. Jika aliran sudah memiliki satu set tag, tag baru ditambahkan ke set tag yang ada.	Tidak ada tag
kms_key_id	<code>string</code>	ID AWS KMS kunci yang akan digunakan untuk mengenkripsi aliran. Untuk informasi selengkapnya, lihat <a href="#">Perlindungan data di Kinesis Video Streams</a> .	Kunci KMS default (aws/kinesis-video.)
streaming_type	STREAMING_TYPE pencacahan	Satu-satunya nilai yang di-support adalah STREAMING_TYPE_REALTIME .	
content_type	<code>string</code>	Format konten aliran. Konsol Kinesis Video Streams dapat memutar ulang konten video/h264 dalam format.	video/h264

Field	Jenis data	Deskripsi	Nilai default
max_latensi	<code>duration&lt;uint64_t, milli&gt;</code>	Latensi maksimum dalam milidetik untuk aliran. Panggilan balik tekanan latensi aliran (jika ditentukan) dipanggil ketika durasi buffer melebihi jumlah waktu ini. Menentukan 0 menunjukkan bahwa tidak ada panggilan balik tekanan latensi aliran yang akan dipanggil.	<code>milliseconds::zero()</code>

Field	Jenis data	Deskripsi	Nilai default
fragment_duration	duration<uint64_t>	Durasi fragmen yang Anda inginkan, dalam hitungan detik. Nilai ini digunakan dalam kombinasi dengan key_frame_fragmentation nilai. Jika nilai ini false, Kinesis Video Streams menghasilkan fragmen pada bingkai kunci setelah durasi ini berlalu. Misalnya, aliran audio Advanced Audio Coding (AAC) memiliki setiap frame sebagai bingkai kunci. Menentukan key_frame_fragmentation = false menyebabkan fragmentasi terjadi pada bingkai kunci setelah durasi ini berakhir, menghasilkan fragmen 2 detik.	2

Field	Jenis data	Deskripsi	Nilai default
timecode_scale	duration< uint64_t, milli>	Skala kode waktu MKV dalam milidetik , yang menentukan granularitas kode waktu untuk bingkai dalam cluster MKV. Timecode bingkai MKV selalu relatif terhadap awal cluster. MKV menggunakan nilai 16-bit yang ditandatangani (0-32767) untuk mewakili kode waktu dalam cluster (fragmen). Verifikasi bahwa kode waktu bingkai dapat direpresentasikan dengan skala kode waktu yang diberikan . Nilai skala kode waktu default 1 ms memastikan bahwa bingkai terbesar yang dapat direpresentasikan adalah 32767 ms ~= 32 detik. Ini melebihi durasi fragmen maksimum yang ditentukan dalam <a href="#">Kuota layanan Kinesis Video</a>	1

Field	Jenis data	Deskripsi	Nilai default
		<a href="#">Streams</a> , yaitu 10 detik.	
key_frame_fragmentasi	bool	Apakah akan menghasilkan fragmen pada bingkai kunci. Jika <code>true</code> , SDK menghasilkan awal fragmen setiap kali ada bingkai kunci. Jika <code>false</code> , Kinesis Video Streams menunggu setidaknya <code>key_frame_fragmentation_duration</code> pada bingkai kunci yang mengikutinya.	<code>true</code>

Field	Jenis data	Deskripsi	Nilai default
<code>frame_timecodes</code>	<code>bool</code>	Apakah akan menggunakan kode waktu bingkai atau membuat stempel waktu menggunakan callback waktu saat ini. Banyak encoder tidak menghasilkan stempel waktu dengan bingkai. Jadi menentukan <code>false</code> parameter ini memastikan bahwa frame diberi stempel waktu saat dimasukkan ke Kinesis Video Streams.	<code>true</code>



Field	Jenis data	Deskripsi	Nilai default
<code>absolute_fragment_times</code>	<code>bool</code>	Kinesis Video Streams menggunakan MKV sebagai mekanisme pengemasan yang mendasarinya. Spesifikasi MKV ketat tentang kode waktu bingkai yang relatif terhadap awal cluster (fragmen). Namun, kode waktu cluster dapat bersifat absolut atau relatif terhadap waktu mulai aliran. Jika stempel waktu relatif, panggilan API PutMedia layanan menggunakan stempel waktu mulai aliran opsional dan menyesuaikan stempel waktu cluster. Layanan selalu menyimpan fragmen dengan stempel waktu absolutnya.	<code>true</code>
<code>fragment_acks</code>	<code>bool</code>	Apakah akan menerima ACK fragmen tingkat aplikasi (ucapan terima kasih).	<code>true</code> , artinya SDK akan menerima ACK dan bertindak sesuai dengan itu.

Field	Jenis data	Deskripsi	Nilai default
<code>restart_on_error</code>	<code>bool</code>	Apakah akan memulai ulang pada kesalahan tertentu.	<code>true</code> , artinya SDK mencoba memulai ulang streaming jika terjadi kesalahan.
<code>hitung ulang_metrik</code>	<code>bool</code>	Apakah akan menghitung ulang metrik. Setiap panggilan untuk mengambil metrik dapat menghitung ulang untuk mendapatkan nilai “berjalan” terbaru, yang mungkin menciptakan dampak CPU kecil. Anda mungkin perlu mengatur ini <code>false</code> pada perangkat berdaya/footprint yang sangat rendah untuk menghemat siklus CPU. Jika tidak, kami tidak menyarankan penggunaan <code>false</code> untuk nilai ini.	<code>true</code>

Field	Jenis data	Deskripsi	Nilai default
<code>nal_adaptation_flags</code>	<code>uint32_t</code>	Menentukan flag adaptasi unit Network Abstraction Layer (NALU). Jika bitstream dikodekan H.264, maka dapat diproses sebagai mentah atau dikemas dalam NALU. Itu dalam format Lampiran-B atau AVCC. Sebagian besar produsen dan konsumen aliran dasar (baca encoder dan decoder) menggunakan format Annex-B karena memiliki kelebihan, seperti pemulihan kesalahan. Sistem tingkat yang lebih tinggi menggunakan format AVCC, yang merupakan format default untuk MPEG, HLS, DASH, dan sebagainya. Pemutaran konsol menggunakan MSE browser (ekstensi sumber media) untuk memecahkan kode dan memutar	Defaultnya adalah mengadaptasi format Lampiran-B ke format AVCC untuk data frame dan untuk data pribadi codec.

Field	Jenis data	Deskripsi	Nilai default
		<p>ulang aliran yang menggunakan format AVCC. Untuk H.264 (dan untuk M-JPEG dan H.265), SDK menyediakan kemampuan adaptasi.</p> <p>Banyak aliran dasar dalam format berikut. Dalam contoh ini, Ab adalah kode awal Lampiran-B (001 atau 0001).</p> <div data-bbox="831 886 1149 1285" 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>Dalam kasus H.264, codec private data (CPD) ada dalam parameter SPS (sequence parameter set) dan PPS (picture parameter set), dan dapat disesuaikan dengan format AVCC. Kecuali pipa media memberika</p>	

Field	Jenis data	Deskripsi	Nilai default
		<p>n CPD secara terpisah, aplikasi dapat mengekstrak CPD dari bingkai. Hal ini dapat dilakukan dengan mencari frame IDR pertama (yang harus berisi SPS dan PPS), mengekstrak dua NALU (yangAb(Sps)Ab(Pps) ), dan mengaturnya di CPD di. StreamDefinition</p> <p>Untuk informasi selengkapnya, lihat <a href="#">Bendera Adaptasi NAL</a>.</p>	
frame_rate	uint32_t	Frame rate yang diharapkan. Nilai ini digunakan untuk menghitung kebutuhan buffering dengan lebih baik.	25

Field	Jenis data	Deskripsi	Nilai default
avg_bandwidth_bps	uint32_t	Bandwidth rata-rata yang diharapkan untuk aliran. Nilai ini digunakan untuk menghitung kebutuhan buffering dengan lebih baik.	$4 * 1024 * 1024$

Field	Jenis data	Deskripsi	Nilai default
buffer_duration	duration<uint64_t>	Durasi buffer aliran, dalam hitungan detik. SDK menyimpan bingkai di penyimpanan konten hingga <code>buffer_duration</code> , setelah itu bingkai sebelumnya dijatuhkan saat jendela bergerak maju. Jika frame yang dijatuhkan belum dikirim ke backend, callback frame yang dijatuhkan dipanggil. Jika durasi buffer saat ini lebih besar dari <code>max_latency</code> , maka panggilan balik tekanan latensi aliran dipanggil. Buffer dipangkas ke awal fragmen berikutnya ketika fragmen bertahan ACK diterima. Ini menunjukkan bahwa konten telah bertahan lama di cloud, jadi menyimpan konten di perangkat lokal tidak lagi diperlukan.	120

Field	Jenis data	Deskripsi	Nilai default
replay_duration	duration<uint64_t>	Durasi, dalam hitungan detik, untuk memutar pembaca saat ini ke belakang untuk memutar ulang selama kesalahan jika restart diaktifkan. Rollback berhenti di awal buffer (jika baru saja mulai streaming atau ACK yang bertahan telah datang). Rollback mencoba mendarat di bingkai kunci yang menunjukkan awal fragmen. Jika kesalahan yang 'menyebabkan restart tidak menunjukkan host mati (host masih hidup dan berisi data bingkai dalam buffer internalnya), rollback berhenti pada frame ACK yang terakhir diterima. Kemudian berguling ke depan ke bingkai kunci berikutnya, karena seluruh fragmen sudah disimpan dalam memori host.	40



Field	Jenis data	Deskripsi	Nilai default
connection_staleness	duration<uint64_t>	Waktu, dalam hitungan detik, setelah itu callback staleness aliran dipanggil jika SDK tidak menerima ACK buffering. Ini menunjukkan bahwa frame sedang dikirim dari perangkat, tetapi backend tidak mengakuinya. Kondisi ini menunjukkan koneksi terputus pada hop menengah atau pada penyeimbang beban.	30
codec_id	string	ID codec untuk trek MKV.	V_MPEG4/ISO/AVC
track_name	string	Nama trek MKV.	kinesis_video

Field	Jenis data	Deskripsi	Nilai default
codecPrivateData	unsigned char*	Buffer data pribadi codec (CPD). Jika pipa media memiliki informasi tentang CPD sebelum aliran dimulai, itu dapat diatur. <code>StreamDefinition.codecPrivateData</code> Bit disalin, dan buffer dapat digunakan kembali atau dibebaskan setelah panggilan untuk membuat aliran. Namun, jika data tidak tersedia saat aliran dibuat, itu dapat diatur di salah satu kelebihan <code>KinesisVideoStream.start(cpd)</code> fungsi.	null
codecPrivateDataUkuran	uint32_t	Ukuran buffer data pribadi codec.	0

## ClientMetrics

`ClientMetrics` objek diisi dengan menelepon `getKinesisVideoMetrics`.

## Bidang anggota

Field	Jenis data	Deskripsi
versi	UINT32	Versi struktur, didefinisikan dalam CLIENT_METRICS_CURRENT_VERSION makro.
contentStoreSize	UINT64	Ukuran penyimpanan konten keseluruhan dalam byte. Ini adalah nilai yang ditentukan dalam DeviceInfo.StorageInfo.storageSize .
contentStoreAvailableUkuran	UINT64	Ukuran penyimpanan yang tersedia saat ini dalam byte.
contentStoreAllocatedUkuran	UINT64	Ukuran yang dialokasikan saat ini. Ukuran yang dialokasikan ditambah yang tersedia harus sedikit lebih kecil dari ukuran penyimpanan keseluruhan, karena pembukuan internal dan implementasi toko konten.
totalContentViewsUkuran	UINT64	Ukuran memori yang dialokasikan untuk semua tampilan konten untuk semua aliran. Ini tidak dihitung terhadap ukuran penyimpanan. Memori ini dialokasikan menggunakan MEMALLOC makro, yang dapat ditimpa untuk menyediakan pengalokasi khusus.

Field	Jenis data	Deskripsi
totalFrameRate	UINT64	Total frame rate yang diamati di semua aliran.
totalTransferRate	UINT64	Total laju aliran yang diamati dalam byte per detik di semua aliran.

## StreamMetrics

StreamMetricsObjek diisi dengan menelepongetKinesisVideoMetrics.

Bidang anggota

Field	Jenis data	Deskripsi
versi	UINT32	Versi struktur, didefinisikan dalam STREAM_METRICS_CURRENT_VERSION makro.
currentViewDuration	UINT64	Durasi frame yang terakumulasi. Dalam kasus jaringan cepat, durasi ini adalah nol atau durasi bingkai (saat bingkai sedang ditransmisikan). Jika durasi menjadi lebih panjang dari max_latency yang ditentukan dalamStreamDefinition , panggilan balik latensi aliran dipanggil jika ditentukan. Durasi ditentukan dalam satuan 100 ns, yang merupakan satuan waktu default untuk lapisan PIC.

Field	Jenis data	Deskripsi
overallViewDuration	UINT64	Durasi tampilan keseluruhan. Jika aliran dikonfigurasi tanpa ACK atau persistensi, nilai ini tumbuh saat frame dimasukkan ke dalam aliran video Kinesis dan menjadi sama dengan <code>buffer_duration</code> di <code>StreamDefinition</code> . Ketika ACK diaktifkan dan ACK yang bertahan diterima, buffer dipangkas ke bingkai kunci berikutnya. Ini karena stempel waktu ACK menunjukkan awal dari seluruh fragmen. Durasi ditentukan dalam satuan 100-ns, yang merupakan satuan waktu default untuk lapisan PIC.
currentViewSize	UINT64	Ukuran dalam byte buffer saat ini.
overallViewSize	UINT64	Ukuran tampilan keseluruhan dalam byte.
currentFrameRate	UINT64	Frame rate yang diamati untuk aliran saat ini.
currentTransferRate	UINT64	Kecepatan transfer yang diamati dalam byte per detik untuk aliran saat ini.

## Callback SDK produsen

Kelas dan metode di Amazon Kinesis Video Streams Producer SDK tidak mempertahankan prosesnya sendiri. Sebaliknya, mereka menggunakan panggilan fungsi masuk dan acara untuk menjadwalkan callback untuk berkomunikasi dengan aplikasi.

Ada dua pola callback yang dapat digunakan aplikasi untuk berinteraksi dengan SDK:

- `CallbackProvider`— Objek ini mengekspos setiap callback dari komponen platform-independent code (PIC) ke aplikasi. Pola ini memungkinkan fungsionalitas penuh, tetapi juga berarti bahwa implementasi harus menangani semua metode API publik dan tanda tangan di lapisan C++.
- [StreamCallbackProvider](#) dan [ClientCallbackProvider](#) — Objek ini mengekspos callback khusus aliran dan spesifik klien, dan lapisan C++ SDK mengekspos sisa callback. Ini adalah pola callback yang disukai untuk berinteraksi dengan Producer SDK.

Diagram berikut menggambarkan model objek dari objek callback:

Dalam diagram sebelumnya, `DefaultCallbackProvider` berasal dari `CallbackProvider` (yang mengekspos semua callback di PIC) dan berisi `StreamCallbackProvider` dan `ClientCallbackProvider`.

Topik ini berisi bagian-bagian berikut:

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks struktur](#)
- [Implementasi callback untuk mencoba lagi streaming](#)

### ClientCallbackProvider

`ClientCallbackProvider` objek mengekspos fungsi callback tingkat klien. Rincian fungsi dijelaskan di [ClientCallbacks](#) bagian ini.

Metode callback:

- `getClientReadyCallback`— Melaporkan keadaan siap untuk klien.

- `getStorageOverflowPressureCallback`— Melaporkan limpahan penyimpanan atau tekanan. Callback ini dipanggil ketika pemanfaatan penyimpanan turun di bawah `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD` nilai, yaitu 5 persen dari ukuran penyimpanan keseluruhan. Untuk informasi selengkapnya, lihat [StorageInfo](#).

## StreamCallbackProvider

`StreamCallbackProvider` objek mengekspos fungsi callback tingkat aliran.

Metode callback:

- `getDroppedFragmentReportCallback`: Melaporkan fragmen yang dijatuhkan.
- `getDroppedFrameReportCallback`— Melaporkan bingkai yang jatuh.
- `getFragmentAckReceivedCallback`— Melaporkan bahwa sebuah fragmen ACK diterima untuk streaming.
- `getStreamClosedCallback`— Melaporkan kondisi aliran tertutup.
- `getStreamConnectionStaleCallback`— Melaporkan kondisi koneksi basi. Dalam kondisi ini, produsen mengirim data ke layanan tetapi tidak menerima pengakuan.
- `getStreamDataAvailableCallback`— Melaporkan bahwa data tersedia di aliran.
- `getStreamErrorReportCallback`— Melaporkan kondisi kesalahan aliran.
- `getStreamLatencyPressureCallback`— Melaporkan kondisi latensi aliran, yaitu ketika ukuran buffer akumulasi lebih besar dari `nilaimax_latency`. Untuk informasi selengkapnya, lihat [StreamDefinition/StreamInfo](#).
- `getStreamReadyCallback`: —Melaporkan kondisi siap streaming.
- `getStreamUnderflowReportCallback`— Melaporkan kondisi aliran underflow. Fungsi ini saat ini tidak digunakan dan dicadangkan untuk penggunaan masa depan.

Untuk kode sumbernya `StreamCallbackProvider`, lihat [StreamCallbackProvider.h](#).

## ClientCallbacks struktur

`ClientCallbacks` struktur berisi titik entri fungsi callback yang dipanggil PIC ketika peristiwa tertentu terjadi. Struktur juga berisi informasi versi di `CALLBACKS_CURRENT_VERSION` bidang, dan `customData` bidang untuk data yang ditentukan pengguna yang dikembalikan dengan fungsi callback individual.

Aplikasi klien dapat menggunakan `this` pointer untuk `custom_data` bidang untuk memetakan fungsi anggota ke `ClientCallback` fungsi statis saat runtime, seperti yang ditunjukkan dalam contoh kode berikut:

```
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(...);
```

## Kejadian

Fungsi	Deskripsi	Tipe
<code>CreateDeviceFunc</code>	Saat ini tidak diterapkan di backend. Panggilan ini gagal saat dipanggil dari Java atau C++. Klien lain melakukan inisialisasi khusus platform.	API Backend
<code>CreateStreamFunc</code>	Disebut saat aliran dibuat.	API Backend
<code>DescribeStreamFunc</code>	Disebut saat <code>DescribeStream</code> dipanggil.	API Backend
<code>GetStreamingEndpointFunc</code>	Disebut saat <code>GetStreamingEndpoint</code> dipanggil.	API Backend
<code>GetStreamingTokenFunc</code>	Disebut saat <code>GetStreamingToken</code> dipanggil.	API Backend
<code>PutStreamFunc</code>	Disebut saat <code>PutStream</code> dipanggil.	API Backend
<code>TagResourceFunc</code>	Disebut saat <code>TagResource</code> dipanggil.	API Backend



Fungsi	Deskripsi	Tipe
CreateMutexFunc	Membuat mutex sinkronisasi.	Sinkronisasi
FreeMutexFunc	Membebaskan mutex.	Sinkronisasi
LockMutexFunc	Mengunci mutex sinkronisasi.	Sinkronisasi
TryLockMutexFunc	Mencoba mengunci mutex. Saat ini tidak diimpleme ntasikan.	Sinkronisasi
UnlockMutexFunc	Membuka kunci mutex.	Sinkronisasi
ClientReadyFunc	Dipanggil ketika klien memasuki keadaan siap.	Notifikasi
DroppedFrameReport Func	Melaporkan saat bingkai dijatuhkan.	Notifikasi
DroppedFragmentRep ortFunc	Melaporkan saat fragmen dijatuhkan. Fungsi ini saat ini tidak digunakan dan dicadangkan untuk penggunaa n masa depan.	Notifikasi
FragmentAckReceive dFunc	Dipanggil ketika sebuah fragmen ACK (buffering, received, persisted, dan error) diterima.	Notifikasi

Fungsi	Deskripsi	Tipe
<code>StorageOverflowPressureFunc</code>	Disebut ketika pemanfaatan penyimpanan turun di bawah <code>STORAGE_PRESSURE_NOTIFICATION_THRESHOLD</code> nilai, yang didefinisikan sebagai 5 persen dari ukuran penyimpanan keseluruhan.	Notifikasi
<code>StreamClosedFunc</code>	Disebut ketika bit terakhir dari frame yang tersisa dialirkan.	Notifikasi
<code>StreamConnectionStaleFunc</code>	Dipanggil ketika aliran memasuki status koneksi basi. Dalam kondisi ini, produsen mengirimkan data ke layanan tetapi tidak menerima pengakuan.	Notifikasi
<code>StreamDataAvailableFunc</code>	Dipanggil saat data streaming tersedia.	Notifikasi
<code>StreamErrorReportFunc</code>	Dipanggil ketika terjadi kesalahan aliran. PIC secara otomatis menutup aliran dalam kondisi ini.	Notifikasi
<code>StreamLatencyPressureFunc</code>	Dipanggil ketika aliran memasuki kondisi latensi, yaitu ketika ukuran buffer akumulasi lebih besar dari <code>max_latency</code> . Untuk informasi selengkapnya, lihat <a href="#">StreamDefinition/StreamInfo</a> .	Notifikasi
<code>StreamReadyFunc</code>	Dipanggil ketika aliran memasuki keadaan siap.	Notifikasi

Fungsi	Deskripsi	Tipe
StreamUnderflowReportFunc	Fungsi ini saat ini tidak digunakan dan dicadangkan untuk penggunaan masa depan.	Notifikasi
DeviceCertToTokenFunc	Mengembalikan sertifikat koneksi sebagai token.	Integrasi platform
GetCurrentTimeFunc	Mengembalikan waktu saat ini.	Integrasi platform
GetDeviceCertificateFunc	Mengembalikan sertifikat perangkat. Fungsi ini saat ini tidak digunakan dan dicadangkan untuk penggunaan masa depan.	Integrasi platform
GetDeviceFingerprintFunc	Mengembalikan sidik jari perangkat. Fungsi ini saat ini tidak digunakan dan dicadangkan untuk penggunaan masa depan.	Integrasi platform
GetRandomNumberFunc	Mengembalikan nomor acak antara 0 dan RAND_MAX.	Integrasi platform
GetSecurityTokenFunc	Mengembalikan token keamanan yang diteruskan ke fungsi yang berkomunikasi dengan API backend. Implementasi dapat menentukan serialAccessKeyId, SecretKeyId, dan token sesi.	Integrasi platform

Fungsi	Deskripsi	Tipe
LogPrintFunc	Log baris teks dengan tag dan tingkat log. Untuk informasi selengkapnya, lihat PlatformUtils.h .	Integrasi platform

Untuk fungsi integrasi platform di tabel sebelumnya, parameter terakhir adalah `ServiceCallContext` struktur, yang memiliki bidang berikut:

- `version`: Versi struct.
- `callAfter`: Waktu absolut setelah itu untuk memanggil fungsi.
- `timeout`: Batas waktu operasi dalam 100 unit nanodetik.
- `customData`: Nilai yang ditentukan pengguna untuk diteruskan kembali ke klien.
- `pAuthInfo`: Kredensi untuk panggilan tersebut. Untuk informasi selengkapnya, lihat struktur (`__AuthInfo`) berikut.

Informasi otorisasi disediakan menggunakan `__AuthInfo` struktur, yang dapat berupa kredensial serial atau token otentikasi khusus penyedia. Struktur ini memiliki bidang-bidang berikut:

- `version`: Versi `__AuthInfo` struktur.
- `type`: `AUTH_INFO_TYPE` Nilai yang menentukan jenis kredensial (sertifikat atau token keamanan).
- `data`: Sebuah array byte yang berisi informasi otentikasi.
- `size`: Ukuran data parameter.
- `expiration`: Kedaluwarsa kredensial dalam 100 unit nanodetik.

## Implementasi callback untuk mencoba lagi streaming

Kinesis Video Producer SDK menyediakan status streaming melalui fungsi callback. Kami menyarankan Anda menerapkan mekanisme callback berikut untuk memulihkan dari masalah jaringan sesaat yang dihadapi selama streaming.

- Streaming tekanan latensi callback - mekanisme callback ini akan dimulai saat SDK menemukan kondisi latensi aliran. Ini terjadi ketika ukuran buffer akumulasi lebih besar dari

nilai `MAX_LATENCY`. Saat streaming dibuat, aplikasi streaming menetapkan `MAX_LATENCY` ke nilai default 60 detik. Implementasi khas untuk panggilan balik ini adalah mengatur ulang koneksi. Anda dapat menggunakan implementasi sampel di [https://github.com/aws-labs/amazon-kinesis-video-streams - producer-sdk-cpp /blob/master/ kinesis-video-c-producer /src/source/ StreamLatencyStateMachine .c](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.c) sesuai kebutuhan. Perhatikan bahwa tidak ada opsi untuk menyimpan frame yang tidak terkirim karena pemadaman jaringan ke penyimpanan sekunder untuk pengisian ulang.

- Streaming staleness callback - callback ini akan dimulai ketika produsen dapat mengirim data ke layanan Amazon Kinesis Data Streams (uplink) tetapi tidak bisa mendapatkan pengakuan (buffer ACK) kembali tepat waktu (default adalah 60 detik). Bergantung pada pengaturan jaringan, panggilan balik tekanan latensi aliran atau callback staleness aliran, atau keduanya dapat dimulai. Mirip dengan implementasi coba ulang panggilan balik tekanan latensi aliran, implementasi tipikal adalah mengatur ulang koneksi dan memulai koneksi baru untuk streaming. Anda dapat menggunakan implementasi sampel di [https://github.com/aws-labs/ amazon-kinesis-video-streams ConnectionStaleStateMachine -producer-c/blob/master/src/source/](https://github.com/aws-labs/amazon-kinesis-video-streams-connection-state-machine-producer-c/blob/master/src/source/ConnectionStaleStateMachine-producer-c/blob/master/src/source/) .c sesuai kebutuhan.
- Panggilan balik kesalahan aliran - panggilan balik ini dimulai saat SDK mengalami batas waktu pada koneksi jaringan atau kesalahan lainnya selama panggilan ke panggilan layanan API KVS.
- Panggilan balik bingkai yang dijatuhkan - panggilan balik ini dimulai ketika ukuran penyimpanan penuh baik karena kecepatan jaringan yang lambat atau kesalahan aliran. Jika kecepatan jaringan menghasilkan frame yang turun, Anda dapat meningkatkan ukuran penyimpanan, mengurangi ukuran bingkai video, atau kecepatan bingkai agar sesuai dengan kecepatan jaringan.

# Perpustakaan Parser aliran video

Perpustakaan Parser aliran video Kinesis adalah seperangkat alat yang dapat Anda gunakan dalam aplikasi Java untuk mengkonsumsi data MKV dalam aliran video Kinesis.

Perpustakaan mencakup alat-alat berikut:

- [StreamingMkvReader](#): Kelas ini membaca elemen MKV tertentu dari aliran video.
- [FragmentMetadataVisitor](#): Kelas ini mengambil metadata untuk fragmen (elemen media) dan trek (aliran data individual yang berisi informasi media, seperti audio atau subtitle).
- [OutputSegmentMerger](#): Kelas ini menggabungkan fragmen atau potongan berturut-turut dalam aliran video.
- [KinesisVideoExample](#): Ini adalah contoh aplikasi yang menunjukkan cara menggunakan Perpustakaan Parser aliran video Kinesis.

Pustaka juga menyertakan pengujian yang menunjukkan bagaimana alat digunakan.

## Prosedur: Menggunakan Perpustakaan Parser video Kinesis

Prosedur ini mencakup langkah-langkah berikut:

- [the section called “Langkah 1: Mengunduh dan mengatur kode”](#).
- [the section called “Langkah 2: Tulis dan periksa kode”](#).
- [the section called “Langkah 3: Jalankan dan verifikasi kode”](#).

## Prasyarat

Anda harus memiliki yang berikut untuk memeriksa dan menggunakan Perpustakaan Parser aliran video Kinesis:

- Akun (AWS) Amazon Web Services. Jika Anda belum memiliki Akun AWS, lihat [the section called “Mendaftar untuk Akun AWS”](#).
- Sebuah Java integrated development environment (IDE), seperti [Eclipse Java Neon](#) atau [JetBrainsIntelliJ](#) Idea.

## Langkah 1: Mengunduh dan mengatur kode

Di bagian ini, Anda mengunduh perpustakaan Java dan kode uji, dan mengimpor proyek ke IDE Java Anda.

Untuk prasyarat dan detail lainnya tentang prosedur ini, lihat. [Perpustakaan Parser](#)

1. Buat direktori dan kloning kode sumber perpustakaan dari GitHub repositori (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>).

```
$ git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Buka Java IDE yang Anda gunakan (misalnya, [Eclipse](#) atau [IntelliJ IDEA](#)) dan impor proyek Apache Maven yang Anda unduh:
  - Dalam Eclipse: Pilih File, Impor, Maven, Ada Proyek Maven, dan arahkan ke folder. `kinesis-video-streams-parser-lib`
  - Di IntelliJ Ide: Pilih Impor. Arahkan ke file `pom.xml` di root paket yang diunduh.

Untuk informasi selengkapnya, lihat dokumentasi IDE terkait.

## Langkah selanjutnya

[the section called “Langkah 2: Tulis dan periksa kode”](#).

## Langkah 2: Tulis dan periksa kode

Pada bagian ini, Anda memeriksa perpustakaan Java dan kode uji, dan belajar bagaimana menggunakan alat dari perpustakaan dalam kode Anda sendiri.

Perpustakaan Parser aliran video Kinesis berisi alat-alat berikut:

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

## StreamingMkvReader

Kelas ini membaca elemen MKV tertentu dari sungai dengan cara non-blocking.

Contoh kode berikut (dari `FragmentMetadataVisitorTest`) menunjukkan cara membuat dan menggunakan `StreamingMkvReader` untuk mengambil `MkvElement` objek dari input stream yang disebut `inputStream`:

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

## FragmentMetadataVisitor

Kelas ini mengambil metadata untuk fragmen (elemen media) dan melacak aliran data individual yang berisi informasi media, seperti data pribadi codec, lebar piksel, atau tinggi piksel.

Contoh kode berikut (dari `FragmentMetadataVisitorTest` file) menunjukkan bagaimana menggunakan `FragmentMetadataVisitor` untuk mengambil data dari `MkvElement` objek:

```
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
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
{
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame =
((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
```





Untuk digunakan `FragmentMetadataVisitor` dalam proyek Anda, berikan `MkvElement` objek ke pengunjung menggunakan `accept` metode mereka:

```
mkvElement.get().accept(fragmentVisitor);
```

## OutputSegmentMerger

Kelas ini menggabungkan metadata dari trek yang berbeda dalam aliran ke dalam aliran dengan segmen tunggal.

Contoh kode berikut (dari `FragmentMetadataVisitorTest` file) menunjukkan bagaimana menggunakan `OutputSegmentMerger` untuk menggabungkan metadata track dari array byte yang disebut: `inputBytes`

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

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

OutputSegmentMerger outputSegmentMerger =
    OutputSegmentMerger.createDefault(outputStream);

CompositeMkvElementVisitor compositeVisitor =
    new TestCompositeVisitor(fragmentVisitor, outputSegmentMerger);

final InputStream in = TestResourceUtil.getTestInputStream("output_get_media.mkv");

StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));

while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(compositeVisitor);
        if
(MkvTypeInfos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetadata().getTypeInfo()))
        {
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
            Assert.assertTrue(frame.getFrameData().limit() > 0);
            MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
            assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
        }
    }
}
```

```
}  
}
```

Contoh sebelumnya menunjukkan pola pengkodean berikut:

- Buat [FragmentMetadataVisitor](#) untuk mengambil metadata dari aliran.
- Buat aliran keluaran untuk menerima metadata gabungan.
- Buat `OutputSegmentMerger`, lewat `diByteArrayOutputStream`.
- Buat `CompositeMkvElementVisitor` yang berisi dua pengunjung.
- Buat `InputStream` yang menunjuk ke file yang ditentukan.
- Gabungkan setiap elemen dalam data input ke dalam aliran output.

## KinesisVideoExample

Ini adalah contoh aplikasi yang menunjukkan cara menggunakan Perpustakaan Parser aliran video Kinesis.

Kelas ini melakukan operasi berikut:

- Membuat aliran video Kinesis. Jika aliran dengan nama yang diberikan sudah ada, aliran dihapus dan dibuat ulang.
- Panggilan [PutMedia](#) untuk melakukan streaming fragmen video ke aliran video Kinesis.
- Panggilan [GetMedia](#) untuk melakukan streaming fragmen video dari aliran video Kinesis.
- Menggunakan [StreamingMkvReader](#) untuk mengurai fragmen kembali pada sungai, dan menggunakan [FragmentMetadataVisitor](#) untuk log fragmen.

## Hapus dan buat ulang aliran

Contoh kode berikut (dari `StreamOps.java` file) menghapus aliran video Kinesis yang diberikan:

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

Contoh kode berikut (dari `StreamOps.java` file) membuat aliran video Kinesis dengan nama yang ditentukan:

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

## Panggilan PutMedia

Contoh kode berikut (dari `PutMediaWorker.java` file) panggilan [PutMedia](#) di sungai:

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

## Panggilan GetMedia

Contoh kode berikut (dari `GetMediaWorker.java` file) panggilan [GetMedia](#) di sungai:

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

## Parse hasilnya GetMedia

Bagian ini menjelaskan cara menggunakan [StreamingMkvReader](#), [FragmentMetadataVisitor](#) dan `CompositeMkvElementVisitor` mengurai, menyimpan ke file, dan mencatat data yang dikembalikan. `GetMedia`

Baca output `GetMedia` dengan `StreamingMkvReader`

Contoh kode berikut (dari `GetMediaWorker.java` file) membuat [StreamingMkvReader](#) dan menggunakannya untuk mengurai hasil dari [GetMedia](#) operasi:

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

```
}
```

Dalam contoh kode sebelumnya, [StreamingMkvReader](#) mengambil MKVElement objek dari payload hasil. GetMedia Pada bagian selanjutnya, elemen dilewatkan ke [FragmentMetadataVisitor](#).

Ambil fragmen dengan FragmentMetadataVisitor

Contoh kode berikut (dari KinesisVideoExample.java dan StreamingMkvReader.java file) membuat [FragmentMetadataVisitor](#). MkvElementObjek iterasi oleh kemudian [StreamingMkvReader](#) diteruskan ke pengunjung menggunakan accept metode.

dari *KinesisVideoExample.java*:

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

dari *StreamingMkvReader.java*:

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

Log elemen dan menulis mereka ke file

Contoh kode berikut (dari KinesisVideoExample.java file) menciptakan benda-benda berikut dan mengembalikan mereka sebagai bagian dari nilai kembali GetMediaProcessingArguments fungsi:

- A LogVisitor (perpanjangan dari MkvElementVisitor) yang menulis ke log sistem.
- Sebuah OutputStream yang menulis data yang masuk ke file MKV.
- Sebuah BufferedOutputStream yang buffer data terikat untuk OutputStream
- Sebuah [the section called “OutputSegmentMerger”](#) yang menggabungkan elemen berturut-turut dalam GetMedia hasil dengan track yang sama dan EBMLdata.
- Sebuah CompositeMkvElementVisitor yang menyusun [FragmentMetadataVisitor, the section called “OutputSegmentMerger”](#), dan LogVisitor menjadi pengunjung elemen tunggal.

```
//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);
```

Argumen pemrosesan media kemudian diteruskan ke `GetMediaWorker`, yang pada gilirannya diteruskan ke `ExecutorService`, yang melakukan pekerja pada utas terpisah:

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getMkvElementVisitor());
executorService.submit(getMediaWorker);
```

## Langkah selanjutnya

[the section called “Langkah 3: Jalankan dan verifikasi kode”](#)

## Langkah 3: Jalankan dan verifikasi kode

Perpustakaan Parser aliran video Kinesis berisi alat yang dimaksudkan untuk Anda gunakan dalam proyek Anda sendiri. Proyek ini berisi pengujian unit untuk alat yang dapat Anda jalankan untuk memverifikasi instalasi Anda.

Unit test berikut disertakan dalam pustaka:

- mkv
  - ElementSizeAndOffsetVisitorTest
  - MkvValueTest
  - StreamingMkvReaderTest
- utilitas
  - FragmentMetadataVisitorTest
  - OutputSegmentMergerTest

## Contoh Aliran Video Amazon Kinesis

Contoh kode berikut menunjukkan cara bekerja dengan Kinesis Video Streams API:

### Contoh: Mengirim data ke Kinesis Video Streams

- [Contoh: Plugin SDK GStreamer Produser Kinesis Video Streamer](#): Menunjukkan cara membuat SDK Produser Kinesis Video Streams untuk digunakan sebagai tujuan GStreamer.
- [Jalankan elemen GStreamer dalam wadah Docker](#): Menunjukkan cara menggunakan image Docker yang sudah dibuat sebelumnya untuk mengirim video RTSP dari kamera IP ke Kinesis Video Streams.
- [Contoh: Streaming dari sumber RTSP](#): Menunjukkan cara membuat gambar Docker Anda sendiri dan mengirim video RTSP dari kamera IP ke Kinesis Video Streams.
- [Contoh: Mengirim data ke Kinesis Video PutMedia Streams menggunakan API](#): Menunjukkan cara menggunakan file [Menggunakan Java Producer Library](#) untuk mengirim data ke Kinesis Video Streams yang sudah ada dalam format container (MKV) menggunakan API. [PutMedia](#)

### Contoh: Mengambil data dari Kinesis Video Streams

- [KinesisVideoExample](#): Menunjukkan cara mengurai dan mencatat fragmen video menggunakan Perpustakaan Pengurai Aliran Video Kinesis.
- [Contoh: Mengurai dan merender fragmen Kinesis Video Streams: Menunjukkan cara mengurai dan merender fragmen aliran video Kinesis menggunakan JCodec dan JFrame.](#)
- [the section called “SageMaker”](#): Menampilkan solusi yang menggunakan Amazon SageMaker untuk menentukan kapan objek tertentu muncul dalam aliran video.

### Contoh: Memutar kembali data video

- [Contoh: Gunakan HLS dalam HTML dan JavaScript](#): Menunjukkan cara mengambil sesi streaming HLS untuk aliran video Kinesis dan memutarnya kembali di halaman web.



## Prasyarat

- Dalam kode contoh, Anda memberikan kredensial dengan menentukan profil yang Anda tetapkan dalam file profil AWS kredensial Anda, atau dengan memberikan kredensial di properti sistem Java dari lingkungan pengembangan terintegrasi (IDE) Anda. Jika Anda belum melakukannya, pertamanya siapkan kredensial Anda. Untuk informasi selengkapnya, lihat [Menyiapkan AWS Kredensial dan Wilayah untuk Pengembangan](#).
- Kami menyarankan Anda menggunakan IDE Java untuk melihat dan menjalankan kode, seperti salah satu dari berikut ini:
  - [Gerhana Neon Jawa](#)
  - [JetBrains IDE IntelliJ](#)

## Contoh: Plugin SDK GStreamer Produser Kinesis Video Streamer

Topik ini menjelaskan cara membuat SDK Produser Amazon Kinesis Video Streams untuk digunakan sebagai plugin GStreamer.

Topik

- [Unduh, buat, dan konfigurasi elemen GStreamer](#)
- [Jalankan elemen GStreamer](#)
- [Contoh perintah peluncuran GStreamer](#)
- [Jalankan elemen GStreamer dalam wadah Docker](#)
- [Referensi parameter elemen GStreamer](#)

[GStreamer](#) adalah kerangka media populer yang digunakan oleh beberapa kamera dan sumber video untuk membuat saluran media khusus dengan menggabungkan plugin modular. Plugin Kinesis Video Streams GStreamer merampingkan integrasi saluran media GStreamer Anda yang ada dengan Kinesis Video Streamer. Setelah mengintegrasikan GStreamer, Anda dapat melakukan streaming video dari webcam atau kamera Real Time Streaming Protocol (RTSP) ke Kinesis Video Streams untuk pemutaran, penyimpanan, dan analisis lebih lanjut secara real-time atau nanti.

Plugin GStreamer secara otomatis mengelola transfer streaming video Anda ke Kinesis Video Streams dengan merangkum fungsionalitas yang disediakan oleh Kinesis Video Streams Producer SDK dalam elemen sink GStreamer, `kvssink`. Kerangka kerja GStreamer menyediakan lingkungan

terkelola standar untuk membangun aliran media dari perangkat seperti kamera atau sumber video lainnya untuk pemrosesan, rendering, atau penyimpanan lebih lanjut.

Pipeline GStreamer biasanya terdiri dari tautan antara sumber (kamera video) dan elemen wastafel (baik pemutar untuk merender video, atau penyimpanan untuk pengambilan offline). Dalam contoh ini, Anda menggunakan elemen Producer SDK sebagai wastafel, atau tujuan media, untuk sumber video Anda (webcam atau kamera IP). Elemen plugin yang merangkum SDK kemudian mengirimkan aliran video ke Kinesis Video Streams.

Topik ini menjelaskan cara membangun saluran media GStreamer yang mampu mengalirkan video dari sumber video, seperti kamera web atau aliran RTSP, biasanya terhubung melalui tahap pengkodean menengah (menggunakan pengkodean H.264) ke Kinesis Video Streams. Ketika streaming video Anda tersedia sebagai aliran video Kinesis, Anda dapat menggunakan Perpustakaan Pengurai Stream Video Kinesis untuk pemrosesan lebih lanjut, pemutaran, penyimpanan, atau analisis aliran video Anda.

## Unduh, buat, dan konfigurasi elemen GStreamer

Contoh Plugin GStreamer disertakan dengan Kinesis Video Streams C ++ Producer SDK. Untuk informasi tentang prasyarat dan pengunduhan SDK, lihat. [Langkah 1: Unduh dan konfigurasi kode Perpustakaan Produser C ++](#)

Anda dapat membangun sink Producer SDK GStreamer sebagai pustaka dinamis di macOS, Ubuntu, Raspberry Pi, atau Windows. Plugin GStreamer terletak di direktori `Andabuild`. Untuk memuat plugin ini, itu harus ada di `AndaGST_PLUGIN_PATH`. Jalankan perintah berikut:

```
export GST_PLUGIN_PATH=`pwd`/build
```

### Note

Di macOS, Anda hanya dapat melakukan streaming video dari kamera jaringan saat menjalankan GStreamer dalam wadah Docker. Streaming video dari kamera USB di macOS dalam wadah Docker tidak didukung.

## Jalankan elemen GStreamer

Untuk menjalankan GStreamer dengan elemen Kinesis Video Streams Producer SDK sebagai sink, gunakan perintah `gst-launch-1.0`. Gunakan pengaturan yang sesuai untuk plugin GStreamer untuk digunakan. [Misalnya, v4l2src untuk perangkat v4l2 pada sistem Linux, atau rtspsrc untuk perangkat RTSP](#). Tentukan `kvssink` sebagai wastafel (tujuan akhir pipeline) untuk mengirim video ke Producer SDK.

`kvssink` Elemen memiliki parameter yang diperlukan berikut:

- **stream-name**— Nama aliran video Kinesis tujuan.
- **storage-size**— Ukuran penyimpanan perangkat dalam kilobyte. Untuk informasi tentang mengonfigurasi penyimpanan perangkat, lihat [StorageInfo](#).
- **access-key**— Kunci AWS akses yang digunakan untuk mengakses Kinesis Video Streams. Anda harus memberikan parameter ini atau `credential-path`.
- **secret-key**— Kunci AWS rahasia yang digunakan untuk mengakses Kinesis Video Streams. Anda harus memberikan parameter ini atau `credential-path`.
- **credential-path**— Jalur ke file yang berisi kredensial Anda untuk mengakses Kinesis Video Streams. Untuk informasi selengkapnya tentang memutar kredensial, lihat [Mengelola Kunci Akses untuk Pengguna IAM](#). Anda harus memberikan parameter ini atau `access-key` dan `secret-key`.

Untuk informasi tentang parameter `kvssink` opsional, lihat [Referensi parameter elemen GStreamer](#).

Untuk informasi terbaru tentang plugin dan parameter GStreamer, lihat Plugin [GStreamer, atau gunakan perintah berikut](#) untuk mencantumkan opsi:

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

Jika build gagal atau `GST_PLUGIN_PATH` tidak disetel dengan benar, output Anda akan terlihat mirip dengan ini:

```
No such element or plugin 'kvssink'
```

## Contoh perintah peluncuran GStreamer

Contoh berikut menunjukkan cara menggunakan plugin GStreamer untuk melakukan streaming video dari berbagai jenis perangkat.

## Contoh 1: Streaming video dari kamera RTSP di Ubuntu

[Perintah berikut membuat pipeline GStreamer di Ubuntu yang mengalir dari kamera RTSP jaringan, menggunakan plugin rtspsrc GStreamer:](#)

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

## Contoh 2: Encode dan streaming video dari kamera USB di Ubuntu

Perintah berikut membuat pipeline GStreamer di Ubuntu yang mengkodekan aliran dari kamera USB dalam format H.264, dan mengalirkannya ke Kinesis Video Streams. Contoh ini menggunakan plugin [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"
```

## Contoh 3: Streaming video yang telah disandikan sebelumnya dari kamera USB di Ubuntu

Perintah berikut membuat pipeline GStreamer di Ubuntu yang mengalirkan video yang telah dikodekan kamera dalam format H.264 ke Kinesis Video Streams. Contoh ini menggunakan plugin [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"
```

## Contoh 4: Streaming video dari kamera jaringan di macOS

Perintah berikut membuat pipeline GStreamer di macOS yang mengalirkan video ke Kinesis Video Streams dari kamera jaringan. Contoh ini menggunakan plugin [rtspsrc](#) GStreamer.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-
name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

## Contoh 5: Streaming video dari kamera jaringan di Windows

Perintah berikut membuat pipeline GStreamer di Windows yang mengalirkan video ke Kinesis Video Streams dari kamera jaringan. Contoh ini menggunakan plugin [rtspsrc](#) GStreamer.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! video/x-h264, format=avc, alignment=au ! kvssink stream-
  name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
  key="YourSecretKey" aws-region="YourAWSRegion"
```

## Contoh 6: Streaming video dari kamera di Raspberry Pi

Perintah berikut membuat pipeline GStreamer di Raspberry Pi yang mengalirkan video ke Kinesis Video Streams. Contoh ini menggunakan plugin [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"
```

## Contoh 7: Streaming video dari kamera pada Raspberry Pi dan tentukan Wilayah

Perintah berikut membuat pipeline GStreamer di Raspberry Pi yang mengalirkan video ke Kinesis Video Streams di Wilayah AS Timur (Virginia N.). Contoh ini menggunakan plugin [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"
```

## Contoh 8: Streaming audio dan video di Raspberry Pi dan Ubuntu

Lihat cara [menjalankan perintah gst-launch-1.0 untuk memulai streaming audio dan video di Raspberry-Pi dan Ubuntu](#).

## Contoh 9: Streaming audio dan video di macOS

Lihat cara [menjalankan perintah gst-launch-1.0 untuk memulai streaming audio dan video](#) di macOS.

## Contoh 10: Unggah file MKV yang berisi audio dan video

Lihat cara [menjalankan perintah gst-launch-1.0 untuk mengunggah file MKV yang](#) berisi audio dan video.

## Jalankan elemen GStreamer dalam wadah Docker

Docker adalah platform untuk mengembangkan, menyebarkan, dan menjalankan aplikasi menggunakan kontainer. Menggunakan Docker untuk membuat pipeline GStreamer menstandarisasi lingkungan operasi untuk Kinesis Video Streams, yang merampingkan pembuatan dan penggunaan aplikasi.

Untuk menginstal dan mengkonfigurasi Docker, lihat berikut ini:

- [Instruksi unduhan Docker](#)
- [Memulai dengan Docker](#)

Setelah menginstal Docker, Anda dapat mengunduh Kinesis Video Streams C ++ Producer SDK (dan plugin GStreamer) dari Amazon Elastic Container Registry menggunakan perintah. `docker pull`

Untuk menjalankan GStreamer dengan elemen Kinesis Video Streams Producer SDK sebagai sink di container Docker, lakukan hal berikut:

Topik

- [Otentikasi klien Docker Anda](#)
- [Unduh gambar Docker untuk Ubuntu, macOS, Windows, atau Raspberry Pi](#)
- [Jalankan image Docker](#)

### Otentikasi klien Docker Anda

Autentikasi klien Docker Anda ke registrasi Amazon ECR di mana Anda berniat untuk menarik citra Anda. Anda harus mendapatkan token otentikasi untuk setiap registri yang digunakan. Token berlaku selama 12 jam. Untuk informasi selengkapnya, lihat [Autentikasi Registri](#) dalam Panduan Pengguna Amazon Elastic Container Registry.

## Example : Otentikasi dengan Amazon ECR

```
aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin
https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

Jika berhasil, output akan mencetak Login Succeeded.

## Unduh gambar Docker untuk Ubuntu, macOS, Windows, atau Raspberry Pi

Unduh gambar Docker ke lingkungan Docker Anda menggunakan salah satu perintah berikut, tergantung pada sistem operasi Anda:

### Unduh gambar Docker untuk Ubuntu

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-amazon-linux:latest
```

### Unduh gambar Docker untuk macOS

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-amazon-linux:latest
```

### Unduh gambar Docker untuk Windows

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-
cpp-amazon-windows:latest
```

### Unduh gambar Docker untuk Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-
sdk-cpp-raspberry-pi:latest
```

Untuk memverifikasi bahwa gambar berhasil ditambahkan, gunakan perintah berikut:

```
docker images
```

## Jalankan image Docker

Gunakan salah satu perintah berikut untuk menjalankan image Docker, tergantung pada sistem operasi Anda:

## Jalankan image Docker di 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
```

## Jalankan image Docker di macOS

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

## Jalankan gambar Docker di 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>
```

## Jalankan gambar Docker di Raspberry Pi

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi /bin/bash
```

Docker meluncurkan wadah dan memberi Anda prompt perintah untuk menggunakan perintah di dalam wadah.

Dalam wadah, atur variabel lingkungan menggunakan perintah berikut:

```
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
```

Mulai streaming dari kamera menggunakan `gst-launch-1.0` perintah yang sesuai untuk perangkat Anda.

Untuk contoh menggunakan `gst-launch-1.0` perintah untuk terhubung ke kamera web lokal atau kamera RTSP jaringan, lihat [Luncurkan Perintah](#).



## Referensi parameter elemen GStreamer

Untuk mengirim video ke Amazon Kinesis Video Streams Producer SDK, `kvssink` Anda tentukan sebagai wastafel, atau tujuan akhir pipeline. Referensi ini memberikan informasi tentang parameter `kvssink` yang diperlukan dan opsional. Untuk informasi selengkapnya, lihat [the section called “GStreamer”](#).

### Topik

- [the section called “Memberikan kredensi untuk kvssink”](#)
- [the section called “Menyediakan wilayah untuk kvssink”](#)
- [the section called “kvssinkparameter yang diperlukan”](#)
- [the section called “kvssinkparameter opsional”](#)

## Memberikan kredensi untuk `kvssink`

Untuk mengizinkan elemen `kvssink` GStreamer membuat permintaan AWS, berikan AWS kredensialnya untuk digunakan saat memanggil layanan Amazon Kinesis Video Streams. Rantai penyedia kredensi mencari kredensial dengan urutan sebagai berikut:

### 1. AWS IoT kredensi

Untuk mengatur AWS IoT kredensial, lihat [the section called “Mengontrol akses ke sumber daya Kinesis Video Streams menggunakan AWS IoT”](#)

Nilai `iot-credentials` parameter harus dimulai dengan `iot-certificate`, dan diikuti oleh daftar dipisahkan koma dari pasangan *kunci = nilai* berikut.

Kunci	Diperlukan	Deskripsi
<code>ca-path</code>	Ya	Jalur file ke sertifikat CA yang digunakan untuk membangun kepercayaan dengan layanan backend melalui TLS.  Example  Contoh: <code>/file/path/to/certificate.pem</code>

Kunci	Diperlukan	Deskripsi
cert-path	Ya	<p>Jalur file ke sertifikat X.509.</p> <p>Example</p> <p>Contoh:</p> <pre><i>/file/path/to/certificateID -certificate.pem.crt</i></pre>
endpoint	Ya	<p>Titik akhir penyedia titik akhir AWS IoT Core kredensi untuk akun Anda. AWS Lihat <a href="#">Panduan AWS IoT Pengembangan</a>.</p> <p>Example</p> <p>Contoh: <i>credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com</i></p>
key-path	Ya	<p>File path ke kunci pribadi yang digunakan dalam public/private key pair.</p> <p>Example</p> <p>Contoh:</p> <pre><i>/file/path/to/certificateID -private.pem.key</i></pre>

Kunci	Diperlukan	Deskripsi
<code>role-aliases</code>	Ya	<p>Nama alias peran yang menunjuk ke peran AWS IAM yang akan digunakan saat menghubungkan ke. AWS IoT Core</p> <p>Example</p> <p>Contoh: <i>KvsCameraIoRoleAlias</i></p>
<code>iot-thing-name</code>	Tidak	<p><code>iot-thing-name</code> Itu opsional. Jika <code>iot-thing-name</code> tidak disediakan, nilai <code>stream-name</code> parameter digunakan.</p> <p>Example</p> <p>Contoh: <i>kvs_example_camera</i></p>

## Example

### Contoh:

```
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. Variabel-variabel lingkungan

Untuk memiliki kredensi `kvssink` penggunaan dari lingkungan, atur variabel lingkungan berikut:

Nama Variabel Lingkungan	Diperlukan	Deskripsi
AWS_ACCESS_KEY_ID	Ya	Kunci AWS akses yang digunakan untuk mengakses Amazon Kinesis Video Streams.
AWS_SECRET_ACCESS_KEY	Ya	Kunci AWS rahasia yang terkait dengan kunci akses.
AWS_SESSION_TOKEN	Tidak	Menentukan nilai token sesi yang diperlukan jika Anda menggunakan kredensya I keamanan sementara langsung dari operasi. AWS STS

Menyetel variabel lingkungan mengubah nilai yang digunakan hingga akhir sesi shell Anda, atau sampai Anda menyetel variabel ke nilai yang berbeda. Untuk membuat variabel persisten di seluruh sesi future, atur di skrip startup shell Anda.

### 3. **access-key**, **secret-key** parameter

Untuk menentukan kredensyal secara langsung sebagai `kvssink` parameter, atur parameter berikut:

<b>kvssink</b> Nama Parameter	Diperlukan	Deskripsi
<code>access-key</code>	Ya	Kunci AWS akses yang digunakan untuk mengakses Amazon Kinesis Video Streams.
<code>secret-key</code>	Ya	Kunci AWS rahasia yang terkait dengan kunci akses.
<code>session-token</code>	Tidak	Menentukan nilai token sesi yang diperlukan jika Anda menggunakan kredensya

kvssinkNama Parameter	Diperlukan	Deskripsi
		I keamanan sementara langsung dari operasi. AWS STS

### Example

Menggunakan kredensil statis:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEYEXAMPLE"
```

### Example

Menggunakan kredensil sementara:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
access-key="AKIDEXAMPLE" secret-key="SKEYEXAMPLE" session-token="STEXAMPLE"
```

## 4. File kredensi

### Important

Jika Anda telah memilih salah satu metode sebelumnya, Anda tidak dapat menggunakan `credential-file` kvssink parameter.

kvssinkNama Parameter	Diperlukan	Deskripsi
<code>credential-file</code>	Ya	Path ke file teks yang berisi kredensyal dalam format tertentu.

File teks harus berisi kredensil dalam salah satu format berikut:

- KREDENSIAL *YourAccessKeyYourSecretKey*

- **KEDALUWARSA KREDENSIAL** *YourAccessKeyYourSecretKeySessionToken*

## Example

Contoh: *credentials.txt* File Anda terletak di `/home/ubuntu` dan berisi yang berikut:

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

Untuk menggunakannya `kvssink`, ketik:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"  
credential-file="/home/ubuntu/credentials.txt"
```

### Note

Waktu kedaluwarsa harus setidaknya  $5 + 30 + 3 = 38$  detik di masa depan. Masa tenggang didefinisikan sebagai `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` variabel dalam [IotCredentialProvider.h](#). Jika kredensial terlalu dekat dengan kedaluwarsa saat Anda mulai `kvssink`, Anda menerima kode kesalahan. `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`

### Important

`kvssink` tidak memodifikasi file kredensial. Jika Anda menggunakan kredensi sementara, file kredensial harus diperbarui oleh sumber luar sebelum waktu kedaluwarsa dikurangi masa tenggang.

## Menyediakan wilayah untuk **kvssink**

Berikut ini adalah urutan pencarian wilayah:

1. `AWS_DEFAULT_REGION` variabel lingkungan ditinjau terlebih dahulu. Jika diatur, wilayah itu digunakan untuk mengkonfigurasi klien.
2. `aws-region` parameter ditinjau selanjutnya. Jika diatur, wilayah itu digunakan untuk mengkonfigurasi klien.
3. Jika tidak satu pun dari metode sebelumnya yang digunakan, `kvssink` defaultnya. `us-west-2`

## kvssinkparameter yang diperlukan



Selain memberikan kredensial dan wilayah, kvssink elemen memiliki parameter yang diperlukan berikut:

stream-name- Nama streaming video Amazon Kinesis tujuan.

## kvssinkparameter opsional

kvssinkElemen memiliki parameter opsional berikut. Untuk informasi selengkapnya tentang parameter ini, lihat [Struktur aliran video Kinesis](#).

Parameter	Deskripsi	Unit/Jenis	Default
absolute-fragment-times	Apakah akan menggunakan waktu fragmen absolut.	Boolean	true
access-key	<p>Kunci AWS akses yang digunakan untuk mengakses Kinesis Video Streams.</p> <p>Anda harus memiliki AWS kredensi yang ditetapkan atau memberikan parameter ini. Untuk memberikan informasi ini, ketikkan berikut ini:</p> <pre>\$ export   AWS_ACCES   S_KEY_ID=</pre>		
avg-bandwidth-bps	Bandwidth rata-rata yang diharapkan untuk aliran.	Byte per detik	4194304

Parameter	Deskripsi	Unit/Jenis	Default
<code>aws-region</code>	<p>Wilayah AWS Untuk digunakan.</p> <div data-bbox="472 352 792 1335" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Anda juga dapat menyediakan wilayah dengan variabel <code>AWS_DEFAULT_REGION</code> lingkungan. Variabel lingkungan diutamakan jika variabel lingkungan dan parameter <code>kvssink</code> disetel.</p> </div> <div data-bbox="472 1402 792 1766" style="border: 1px solid #ff9999; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Important</b></p> <p>Wilayah akan default <code>us-west-2</code> jika tidak ditentukan lain.</p> </div>	String	<code>us-west-2</code>
<code>buffer-duration</code>	Durasi buffer aliran.	Detik	180




Parameter	Deskripsi	Unit/Jenis	Default
<code>codec-id</code>	ID codec dari aliran.	String	"V_MPEG4/ISO/AVC"
<code>connection-staleness</code>	Waktu setelahnya, di mana panggilan balik staleness aliran dipanggil.	Detik	60
<code>content-type</code>	Jenis konten aliran.	String	"video/h264"
<code>fragment-acks</code>	Apakah akan menggunakan ACK fragmen.	Boolean	true
<code>fragment-duration</code>	Durasi fragmen yang Anda inginkan.	Milidetik	2000
<code>framerate</code>	Frame rate yang diharapkan.	Bingkai per detik	25
<code>frame-timecodes</code>	Apakah akan menggunakan kode waktu bingkai atau membuat stempel waktu menggunakan callback waktu saat ini.	Boolean	true
<code>key-frame-fragmentation</code>	Apakah akan menghasilkan fragmen pada bingkai kunci.	Boolean	true
<code>log-config</code>	Jalur konfigurasi log.	String	"./kvs_log_configuration"

Parameter	Deskripsi	Unit/Jenis	Default
<code>max-latency</code>	Latensi maksimum untuk streaming.	Detik	60
<code>recalculate-metrics</code>	Apakah akan menghitung ulang metrik.	Boolean	true
<code>replay-duration</code>	Durasi untuk memutar pembaca saat ini ke belakang untuk memutar ulang selama kesalahan jika restart diaktifkan.	Detik	40
<code>restart-on-error</code>	Apakah akan memulai ulang ketika terjadi kesalahan.	Boolean	true
<code>retention-period</code>	Lamanya waktu aliran dipertahankan.	Jam	2
<code>rotation-period</code>	Periode rotasi kunci. Untuk informasi selengkapnya, lihat <a href="#">Memutar Kunci Master Pelanggan</a> .	Detik	2400

Parameter	Deskripsi	Unit/Jenis	Default
secret-key	<p>Kunci AWS rahasia yang digunakan untuk mengakses Kinesis Video Streams.</p> <p>Anda harus memiliki AWS kredensi yang ditetapkan atau memberikan parameter ini.</p> <pre>\$ export   AWS_SECRE T_ACCESS_ KEY_ID=</pre>		
session-token	<p>Menentukan nilai token sesi yang diperlukan jika Anda menggunakan kredensial keamanan sementara langsung dari operasi. AWS STS</p>		
storage-size	<p>Ukuran penyimpanan perangkat dalam MegaBytes (MB). Untuk informasi tentang mengonfigurasi penyimpanan perangkat, lihat <a href="#">StorageInfo</a>.</p>	MegaBytes	128

Parameter	Deskripsi	Unit/Jenis	Default
<code>streaming-type</code>	Jenis streaming. Nilai yang valid meliputi: <ul style="list-style-type: none"><li>• 0: waktu nyata</li><li>• 1: mendekati waktu nyata (saat ini tidak didukung)</li><li>• 2: offline</li></ul>	Enum <code>GstKvsSinkStreamingType</code>	0: waktu nyata
<code>timecode-scale</code>	Skala kode waktu MKV.	Milidetik	1
<code>track-name</code>	Nama trek MKV.	String	"kinesis_video"

Parameter	Deskripsi	Unit/Jenis	Default
<code>iot-certificate</code>	<p>AWS IoT kredensial yang akan digunakan dalam elemen <code>kvssink</code>.</p> <p><code>iot-certificate</code> menerima kunci dan nilai berikut:</p> <div data-bbox="472 621 792 1226" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p><code>iot-thing-name</code> itu opsional. Jika <code>iot-thing-name</code> tidak disediakan, nilai <code>stream-name</code> parameter digunakan.</p> </div> <ul style="list-style-type: none"> <li>• <code>endpoint=iotcredentialsproviderendpoint</code></li> <li>• <code>cert-path=/localdirectorypath/to/certificate</code></li> <li>• <code>key-path=/localdirectorypath /</code></li> </ul>	String	Tidak ada

Parameter	Deskripsi	Unit/Jenis	Default
	<code>to/private/ key</code> <ul style="list-style-type: none"><li><code>ca-path= localdir ectorypath/ to/ca-cert</code></li><li><code>role-alias =role-alias</code></li><li><code>iot-thing- name=YourIotTh ingName</code></li></ul>		

## Contoh: Mengirim data ke Kinesis Video PutMedia Streams menggunakan API

Contoh ini menunjukkan cara menggunakan [PutMediaAPI](#). Ini menunjukkan cara mengirim data yang sudah dalam format wadah (MKV). Jika data Anda harus dirakit ke dalam format kontainer sebelum mengirim (misalnya, jika Anda merakit data video kamera ke dalam bingkai), lihat [Perpustakaan Produser Aliran Video Kinesis](#).

### Note

PutMediaOperasi ini hanya tersedia di C++ dan Java SDK. Hal ini disebabkan manajemen full-duplex koneksi, aliran data, dan pengakuan. Ini tidak didukung dalam bahasa lain.

Contoh ini mencakup langkah-langkah berikut:

- [Langkah 1: Unduh dan konfigurasi kode](#)
- [Langkah 2: Tulis dan periksa kodenya](#)
- [Langkah 3: Jalankan dan verifikasi kode](#)

## Langkah 1: Unduh dan konfigurasi kode

Ikuti langkah-langkah untuk mengunduh kode contoh Java, mengimpor proyek ke IDE Java Anda, mengonfigurasi lokasi perpustakaan, dan mengonfigurasi kode untuk menggunakan AWS kredensial Anda.

1. Buat direktori dan kloning kode sumber contoh dari GitHub repositori. PutMediaContohnya adalah bagian dari [Perpustakaan Produser Java](#).

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

2. Buka IDE Java yang Anda gunakan (misalnya, [Eclipse](#) atau [IntelliJ IDEA](#)), dan impor proyek Apache Maven yang Anda unduh:
  - Di Eclipse: Pilih File, Impor, Maven, Proyek Maven yang Ada, dan arahkan ke root paket yang diunduh. Pilih pom.xml file.
  - Dalam Ide IntelliJ: Pilih Impor. Arahkan ke pom.xml file di root paket yang diunduh.

Untuk informasi selengkapnya, lihat dokumentasi IDE terkait.

3. Perbarui proyek sehingga IDE dapat menemukan pustaka yang Anda impor.
  - Untuk IntelliJ IDEA, lakukan hal berikut:
    - a. Buka menu konteks (klik kanan) untuk direktori lib proyek, dan pilih Tambah sebagai perpustakaan.
    - b. Pilih File, lalu pilih Project Structure.
    - c. Di bawah Pengaturan Proyek, pilih Modul.
    - d. Di tab Sumber, atur Level Bahasa ke **7** atau lebih tinggi.
  - Untuk Eclipse, lakukan hal berikut:
    - a. Buka menu konteks (klik kanan) untuk proyek, dan pilih Properties, Java Build Path, Source. Kemudian, lakukan hal berikut:
      1. Pada tab Sumber, klik dua kali Lokasi perpustakaan asli.
      2. Di wizard Konfigurasi Folder Perpustakaan Asli, pilih Workspace.
      3. Dalam pilihan Folder Perpustakaan Asli, pilih direktori lib dalam proyek.

- b. Buka menu konteks (klik kanan) untuk proyek, dan pilih Properties. Kemudian, lakukan hal berikut:
  1. Pada tab Libraries, pilih Add Jars.
  2. Di wizard pemilihan JAR, pilih semua .jars di lib direktori proyek.

## Langkah 2: Tulis dan periksa kodenya

Contoh PutMedia API (PutMediaDemo) menunjukkan pola pengkodean berikut:

Topik

- [Buat PutMediaClient](#)
- [Streaming media dan jeda utas](#)

Contoh kode di bagian ini berasal dari PutMediaDemo kelas.

### Buat PutMediaClient

Membuat PutMediaClient objek membutuhkan parameter berikut:

- URI untuk PutMedia titik akhir.
- InputStreamMenunjuk ke file MKV untuk streaming.
- Nama stream. Contoh ini menggunakan aliran yang dibuat di [Menggunakan Java Producer Library](#) (my-stream). Untuk menggunakan aliran yang berbeda, ubah parameter berikut:

```
private static final String STREAM_NAME="my-stream";
```

#### Note

Contoh PutMedia API tidak membuat aliran. Anda harus membuat aliran baik dengan menggunakan aplikasi pengujian untuk [Menggunakan Java Producer Library](#), konsol Kinesis Video Streams, atau. AWS CLI

- Stempel waktu saat ini.
- Jenis kode waktu. Contoh menggunakan RELATIVE, menunjukkan bahwa stempel waktu relatif terhadap awal wadah.



- `AWSKinesisVideoV4SignerObjek` yang memverifikasi bahwa paket yang diterima dikirim oleh pengirim yang berwenang.
- Bandwidth hulu maksimum di Kbps.
- `AckConsumerObjek` untuk menerima paket menerima ucapan terima kasih.

Kode berikut menciptakan `PutMediaClient` objek:

```

/* 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();

```

## Streaming media dan jeda utas

Setelah klien dibuat, sampel memulai streaming asinkron dengan `putMediaInBackground` Thread utama kemudian dijeda `latch.await` sampai `AckConsumer` pengembalian, di mana klien ditutup.

```

/* start streaming video in a background thread */
client.putMediaInBackground();

```

```
/* wait for request/response to complete */
latch.await();

/* close the client */
client.close();
```

## Langkah 3: Jalankan dan verifikasi kode

Untuk menjalankan contoh PutMedia API, lakukan hal berikut:

1. Buat aliran bernama `my-stream` di konsol Kinesis Video Streams atau dengan menggunakan file. AWS CLI
2. Ubah direktori kerja Anda ke direktori SDK produser Java:

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/
```

3. Kompilasi Java SDK dan aplikasi demo:

```
mvn package
```

4. Buat nama file sementara di direktori: `/tmp`

```
jar_files=$(mktemp)
```

5. Buat string classpath dependensi dari repositori lokal ke file:

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. Tetapkan nilai variabel `LD_LIBRARY_PATH` lingkungan sebagai berikut:

```
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. Jalankan demo dari baris perintah sebagai berikut, berikan AWS kredensyal Anda:

```
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. Buka konsol [Kinesis Video](#) Streams, dan pilih streaming Anda di halaman Kelola Streams. Video diputar di panel Pratinjau Video.

## Contoh: Streaming dari sumber RTSP

[Pustaka Produser C++](#) ini berisi definisi untuk wadah [Docker](#) yang terhubung ke kamera jaringan RTSP (Real Time Streaming Protocol). Menggunakan Docker menstandarisasi lingkungan operasi untuk Kinesis Video Streams, yang merampingkan pembuatan dan penggunaan aplikasi.

Prosedur berikut menunjukkan cara mengatur dan menggunakan aplikasi demo RTSP.

Topik

- [Prasyarat](#)
- [Bangun gambar Docker](#)
- [Jalankan aplikasi contoh RTSP](#)

## Prasyarat

Untuk menjalankan aplikasi contoh Kinesis Video Streams RTSP, Anda harus memiliki yang berikut:

- Docker: Untuk informasi tentang menginstal dan menggunakan Docker, lihat tautan berikut:
  - [Petunjuk unduhan Docker](#)
  - [Memulai dengan Docker](#)
- Sumber kamera jaringan RTSP: Untuk informasi tentang kamera yang direkomendasikan, lihat [Persyaratan Sistem](#).

## Bangun gambar Docker

Pertama, buat image Docker yang akan dijalankan aplikasi demo di dalamnya.

1. Kloning repositori demo Amazon Kinesis Video Streams.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

- Ubah ke direktori yang berisi Dockerfile. Dalam hal ini, ini adalah direktori [docker-rtsp](#).

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

- Gunakan perintah berikut untuk membangun image Docker. Perintah ini membuat gambar dan menandainya sebagai rtspdockertest.

```
docker build -t rtspdockertest .
```

- Jalankan `docker images` dan cari ID gambar yang ditandai dengan `rtspdockertest`.

Misalnya, dalam output sampel di bawah ini, IMAGE ID adalah `54f0d65f69b2`.

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB

Anda akan membutuhkan ini di langkah selanjutnya.

## Jalankan aplikasi contoh RTSP

Anda dapat menjalankan aplikasi contoh RTSP baik dari dalam atau di luar wadah Docker. Ikuti instruksi yang sesuai di bawah ini.

### Topik

- [Di dalam wadah Docker](#)
- [Di luar wadah Docker](#)

### Di dalam wadah Docker

#### Jalankan aplikasi contoh RTSP

- Mulai container Amazon Kinesis Video Streams Docker menggunakan perintah berikut:

```
docker run -it YourImageId /bin/bash
```

- Untuk memulai aplikasi sampel, berikan AWS kredensial Anda, nama aliran video Amazon Kinesis, dan URL kamera jaringan RTSP.

**⚠ Important**

Jika Anda menggunakan kredensial sementara, Anda juga harus memberikan `AWS_SESSION_TOKEN`. Lihat contoh kedua di bawah ini.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

Kredensi sementara:

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId
export AWS_SESSION_TOKEN=YourSessionToken
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

3. Masuk ke AWS Management Console dan buka konsol [Kinesis Video Streams](#).

Lihat aliran.

4. Untuk keluar dari wadah Docker, tutup jendela terminal atau ketik `exit`.

## Di luar wadah Docker

Dari luar wadah Docker, gunakan perintah berikut:

```
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=YourAWSRegion; ./
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

## Contoh: Mengurai dan merender fragmen Kinesis Video Streams

[Perpustakaan Parser](#) Berisi aplikasi demo bernama `KinesisVideoRendererExample` yang menunjukkan penguraian dan rendering fragmen aliran video Amazon Kinesis. Contoh menggunakan

[JCodec untuk memecahkan kode](#) frame yang dikodekan H.264 yang dicerna menggunakan aplikasi. [Contoh: Plugin SDK GStreamer Produser Kinesis Video Streamer Setelah frame diterjemahkan menggunakan JCodec, gambar yang terlihat dirender menggunakan JFrame.](#)

Contoh ini menunjukkan bagaimana melakukan hal berikut:

- Ambil frame dari aliran video Kinesis menggunakan GetMedia API dan render aliran untuk dilihat.
- Lihat konten video streaming dalam aplikasi kustom alih-alih menggunakan konsol Kinesis Video Streams.

Anda juga dapat menggunakan kelas dalam contoh ini untuk melihat konten aliran video Kinesis yang tidak dikodekan sebagai H.264, seperti aliran file JPEG yang tidak memerlukan decoding sebelum ditampilkan.

Prosedur berikut menunjukkan cara mengatur dan menggunakan aplikasi demo Renderer.

## Prasyarat

Untuk memeriksa dan menggunakan perpustakaan contoh Renderer, Anda harus memiliki yang berikut:

- Akun (AWS) Amazon Web Services. Jika Anda belum memiliki AWS akun, lihat [Memulai dengan Kinesis Video Streams](#).
- [Lingkungan pengembangan terintegrasi Java \(IDE\), seperti Eclipse Java Neon atau JetBrains IntelliJ Idea.](#)

## Menjalankan contoh penyaji

1. Buat direktori, lalu kloning kode sumber contoh dari GitHub repositori.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. Buka IDE Java yang Anda gunakan (misalnya, [Eclipse](#) atau [IntelliJ IDEA](#)), dan impor proyek Apache Maven yang Anda unduh:
  - Di Eclipse: Pilih File, Impor, Maven, Proyek Maven yang Ada. Buka direktori `kinesis-video-streams-parser-lib` tersebut.
  - Dalam Ide IntelliJ: Pilih Impor. Arahkan ke `pom.xml` file di root paket yang diunduh.

**Note**

Jika IntelliJ tidak dapat menemukan dependensi Anda, Anda mungkin harus melakukan hal berikut:

- Build clean: Pilih File, Settings, Build, Execution, Deployment, Compiler. Verifikasi bahwa Clear output direktori pada rebuild dipilih, lalu pilih Build, Build Project.
- Impor ulang proyek: Buka menu konteks (klik kanan) untuk proyek, dan pilih Maven, Import ulang.

Untuk informasi selengkapnya, lihat dokumentasi IDE terkait.

3. Dari IDE Java Anda, bukasrc/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest.
4. Hapus @Ignore arahan dari file.
5. Perbarui .stream parameter dengan nama aliran video Kinesis Anda.
6. Jalankan KinesisVideoRendererExample tes.

## Cara Kerjanya

Contoh aplikasi menunjukkan hal berikut:

- [Mengirim data MKV](#)
- [Mengurai fragmen MKV ke dalam bingkai](#)
- [Decoding dan menampilkan frame](#)

### Mengirim data MKV

Contoh mengirimkan data MKV sampel dari rendering\_example\_video.mkv file, menggunakan PutMedia untuk mengirim data video ke aliran bernama. render-example-stream

Aplikasi ini menciptakanPutMediaWorker:

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
```

```
    inputStream,  
    streamOps.amazonKinesisVideo);  
executorService.submit(putMediaWorker);
```

Untuk informasi tentang `PutMediaWorker` kelas, lihat [Panggilan PutMedia](#) di [Perpustakaan Parser](#) dokumentasi.

## Mengurai fragmen MKV ke dalam bingkai

Contoh kemudian mengambil dan mem-parsing fragmen MKV dari aliran menggunakan: `GetMediaWorker`

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),  
    getCredentialsProvider(),  
    getStreamName(),  
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),  
    streamOps.amazonKinesisVideo,  
    getMediaProcessingArgumentsLocal().getFrameVisitor());  
executorService.submit(getMediaWorker);
```

Untuk informasi lebih lanjut tentang `GetMediaWorker` kelas, lihat [Panggilan GetMedia](#) di [Perpustakaan Parser](#) dokumentasi.

## Decoding dan menampilkan frame

Contoh kemudian menerjemahkan dan menampilkan bingkai menggunakan [JFrame](#).

Contoh kode berikut adalah dari `KinesisVideoFrameViewer` kelas, yang meluas `JFrame`:

```
public void setImage(BufferedImage bufferedImage) {  
    image = bufferedImage;  
    repaint();  
}
```

Gambar ditampilkan sebagai contoh dari [java.awt.image. BufferedImage](#). Untuk contoh yang menunjukkan cara bekerja dengan `BufferedImage`, lihat [Membaca/Memuat Gambar](#).



## Contoh: Mengidentifikasi objek dalam aliran video menggunakan SageMaker

Contoh ini menunjukkan cara membuat solusi yang digunakan [SageMaker](#) untuk mengidentifikasi kapan objek tertentu muncul dalam aliran video Amazon Kinesis. SageMaker adalah platform terkelola bagi pengembang dan ilmuwan data untuk membangun, melatih, dan menerapkan model pembelajaran mesin dengan cepat dan mudah.

Contohnya terdiri dari wadah [Docker](#) yang mencakup fungsionalitas aplikasi, dan [AWS CloudFormation](#) template yang mengotomatiskan penyebaran sumber daya aplikasi. AWS

AWS CloudFormation Template membuat sumber daya berikut:

- Cluster [Amazon Elastic Container Service \(Amazon ECS\)](#) yang menggunakan mesin komputasi [AWS Fargate](#) yang menjalankan perangkat lunak perpustakaan.
- Tabel [Amazon DynamoDB](#) yang memelihara pos pemeriksaan dan status terkait di seluruh pekerja yang menjalankan tugas Fargate.
- [Aliran data Kinesis](#) yang menangkap output inferensi yang dihasilkan dari SageMaker
- [AWS Lambda Fungsi](#) yang mem-parsing output dari SageMaker.
- [AWS Identity and Access Management \(IAM\)](#) sumber daya untuk menyediakan akses lintas layanan.
- CloudWatch Sumber daya [Amazon](#) untuk memantau aplikasi.

Aplikasi ini kompatibel dengan SageMaker titik akhir apa pun yang memproses data. Contoh ini berisi instruksi untuk membuat SageMaker titik akhir yang menggunakan contoh template algoritma identifikasi objek. Anda dapat memodifikasi atau mengganti algoritme berdasarkan kasus penggunaan dan persyaratan aplikasi Anda.

### Topik

- [Prasyarat](#)
- [Membuat aplikasi](#)
- [Memantau aplikasi](#)
- [Memperluas aplikasi](#)
- [Membersihkan aplikasi](#)

## Prasyarat

Contoh aplikasi memiliki prasyarat berikut:

- [SageMaker](#)
- [Aliran video Kinesis](#)
- [Peran Terkait Layanan](#)

## SageMaker

Contoh ini membutuhkan SageMaker notebook. Untuk informasi tentang membuat buku catatan, lihat [Membuat Instans Notebook](#) di Panduan SageMaker Pengembang Amazon. Perhatikan hal berikut saat membuat buku catatan Anda:

- Tambahkan `Amazon_JumpStart_Object_Detection.ipynb` contoh (dari bagian Pengantar Algoritma Amazon di tab SageMaker Contoh konsol Jupyter) ke buku catatan.
- Buat bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3), dan berikan namanya di langkah Prasyarat saat menambahkan contoh.
- Setelah Anda membuat buku catatan, pilih konfigurasi Endpoint pada SageMaker konsol, dan buat catatan nama Endpoint.

## Aliran video Kinesis

Contoh ini memerlukan satu atau beberapa Kinesis Video Streams yang memiliki data video langsung. Untuk informasi tentang membuat aliran video Kinesis dan mengirim data ke sana dari kamera, lihat [GStreamer](#) Catat nama aliran video Kinesis Anda.

## Peran Terkait Layanan

Contoh ini mengharuskan akun Anda memiliki peran terkait layanan untuk operasi Fargate. Baru Akun AWS mengaktifkan peran ini secara default. Jika Anda melihat kesalahan berikut saat membuat aplikasi, Anda harus mengaktifkan peran terkait layanan:

```
Unable to assume the service linked role. Please verify that the ECS service linked role exists
```

Untuk mengaktifkan peran terkait layanan, jalankan perintah berikut:

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

## Membuat aplikasi

Untuk membuat contoh aplikasi, Anda menggunakan AWS CloudFormation dan template yang disediakan.

Untuk digunakan AWS CloudFormation untuk membuat aplikasi

1. Masuk ke AWS Management Console dan buka AWS CloudFormation konsol menggunakan salah satu tautan berikut untuk Anda Wilayah AWS. Tautan meluncurkan tumpukan yang benar untuk Wilayah Anda:
  - [Peluncuran di Wilayah Asia Pasifik \(Sydney\) \(ap-tenggara 2\)](#)
  - [Peluncuran di Wilayah Asia Pasifik \(Tokyo\) \(ap-timur laut-1\)](#)
  - [Peluncuran di Wilayah Eropa \(Frankfurt\) \(eu-central-1\)](#)
  - [Peluncuran di Wilayah Eropa \(Irlandia\) \(eu-barat-1\)](#)
  - [Peluncuran di Wilayah AS Timur \(Virginia N.\) \(us-timur-1\)](#)
  - [Peluncuran di Wilayah AS Barat \(Oregon\) \(us-barat-2\)](#)
2. Pada halaman Create Stack, berikan nilai-nilai berikut:
  - Berikan tumpukan nama unik (misalnya, nama *pengguna* -KVS-)SageMaker.
  - Berikan nama SageMaker endpoint (bukan ARN endpoint) yang Anda buat di bagian sebelumnya.
  - Berikan nama aliran video Kinesis Anda. Jika Anda memiliki lebih dari satu aliran video Kinesis, berikan nama aliran dalam tanda kutip dan pisahkan dengan koma.
  - Simpan sisa pengaturan apa adanya.

Pilih Berikutnya.

3. Pada halaman Opsi, pertahankan pengaturan apa adanya.
4. Pilih kotak centang Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM. Pilih Berikutnya.

AWS CloudFormation membuat aplikasi.

Tabel berikut mencantumkan beberapa parameter yang digunakan oleh kontainer Docker saat Anda membuat tumpukan menggunakan AWS CloudFormation template ini. Nilai-nilai mereka telah ditentukan sebelumnya dalam SSM sumber daya dalam template, tetapi Anda dapat menyesuaikannya sesuai kebutuhan.

Nama sumber daya	Nilai default	Deskripsi
<code>inferenceInterval</code>	6	The sampling ratio for video frames that are sent to the SageMaker endpoint. The default value of 6 means that 1 out of every 6 frames is sent to the SageMaker endpoint.
<code>sageMakerTaskQueueSize</code>	5000	The size of the queue that maintains the pending requests to the SageMaker endpoint. The size of the queue is affected by 'inferenceInterval' and 'sageMakerTaskThreadPoolSize'. If SageMaker inference takes longer, requests are buffered in this queue.
<code>sageMakerTaskThreadPoolSize</code>	20	Number of threads that's used to concurrently carry out SageMaker inference requests.
<code>sageMakerTaskTimeoutInMilli</code>	20000	The maximum duration accepted for a single request (or a retry request) to the SageMaker endpoint.
<code>sageMakerTaskThreadPoolName</code>	SageMakerThreadPool-%d	The name of the threadpool that's sending requests to the SageMaker endpoint.

Untuk menyesuaikan nilai parameter ini, unduh AWS CloudFormation templat dengan memilih URL templat di halaman Buat tumpukan, lalu temukan parameter ini di Params bagian templat yang terlihat seperti ini:

```
Params:
  Type: AWS::SSM::Parameter
  Properties:
    Name:
      Ref: AppName
    Description: "Configuration for SageMaker app"
    Type: String
    Value:
```

```
Fn::Sub: |
  {"streamNames":[${StreamNames}], "tagFilters":
[${TagFilters}], "sageMakerEndpoint": "${SageMakerEndpoint}",
  "endPointAcceptContentType": "${EndPointAcceptContentType}",

  "kdsStreamName": "${Kds}", "inferenceInterval":6, "sageMakerTaskQueueSize":5000,
  "sageMakerTaskThreadPoolSize":20, "sageMakerTaskTimeoutInMilli":20000,
  "sageMakerTaskThreadPoolName": "SageMakerThreadPool-%d"}
```

## Memantau aplikasi

Aplikasi yang dibuat oleh AWS CloudFormation template mencakup CloudWatch dashboard Amazon dan aliran CloudWatch log yang digunakan untuk memantau metrik dan peristiwa aplikasi.

### Dasbor aplikasi

Aplikasi ini mencakup CloudWatch dashboard untuk memantau metrik aplikasi. Untuk melihat dashboard aplikasi, buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/> dan pilih Dasbor di bilah navigasi kiri.

***Pilih dasbor KVS- SageMaker -Driver- - KvsSageMakerIntegration aws-region.***

Dasbor menunjukkan informasi berikut:

- **Metrik Bingkai:** Metrik untuk memproses aliran video, mengirim frame ke SageMaker titik akhir, dan menulis ke aliran data Kinesis yang menghubungkan SageMaker notebook dengan AWS Lambda fungsi yang memproses hasil keluaran inferensi. SageMaker
- **IngestToProcessLatency:** Perbedaan waktu antara kapan bingkai video dicerna ke dalam layanan Kinesis Video Streams dan saat aplikasi menerima bingkai.
- **Total Sewa Saat Ini:** Aplikasi diberikan izin untuk membaca dari aliran video Kinesis menggunakan sewa. Metrik ini menunjukkan jumlah sewa aktif. Aplikasi ini menggunakan satu sewa per aliran video Kinesis, dan satu sewa untuk sinkronisasi antar aliran.
- **Metrik Sinkronisasi Sewa:** Frekuensi dan durasi sinkronisasi sewa izin.
- **LeaseCount per Pekerja:** Distribusi sewa di antara atas SageMaker pekerja.
- **Jumlah Pekerja:** Jumlah SageMaker pekerja yang memproses aliran. Setiap tugas dalam kluster Amazon ECS memiliki satu pekerja yang berjalan. Satu pekerja dapat memproses lebih dari satu aliran.
- **Pemanfaatan Layanan ECS:** Metrik penggunaan untuk cluster Amazon ECS.

- KinesisDataStream: Metrik penggunaan aliran data Kinesis.
- SageMaker: Operasi yang dilakukan oleh SageMaker notebook.
- Lambda: Jumlah dan durasi fungsi Lambda yang memproses output dari notebook. SageMaker

Jika salah satu informasi dalam grafik ini menunjukkan masalah operasional (seperti nilai yang terus meningkat daripada stabil), lihat bagian berikut tentang cara membaca log aplikasi untuk menentukan masalah.

## CloudWatch Log

Aplikasi ini mencakup dua CloudWatch Log:

Topik

- [Log aplikasi](#)
- [Log fungsi Lambda](#)

### Log aplikasi

Anda dapat menggunakan log aplikasi untuk memantau peristiwa aplikasi dan kondisi kesalahan. Anda juga dapat menggunakan log ini jika Anda perlu menghubungi dukungan produk dengan masalah.

Untuk membaca Log Aplikasi

1. Buka konsol Amazon ECS di <https://console.aws.amazon.com/ecs>.
2. Pilih cluster KVS-Sagemaker-Driver.
3. Pilih **nama tumpukan** - SageMakerDriverService layanan di tab Layanan.
4. Pilih tab Log.

Log aplikasi menampilkan peristiwa seperti inisialisasi, konfigurasi, dan aktivitas sewa.

### Log fungsi Lambda

Anda dapat menggunakan log fungsi Lambda untuk melacak identifikasi objek yang berhasil.

Untuk membaca log Lambda

1. Buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda>.

2. Pilih fungsi Lambda untuk aplikasi Anda. Nama fungsi Lambda dalam format berikut:

```
stack-name-LambdaFunction-A1B2C3D4E5F6G
```

3. Pilih panel Monitoring.
4. Pilih Lihat log masuk CloudWatch.

CloudWatch Log untuk aplikasi menunjukkan identifikasi objek yang berhasil dalam aliran video Kinesis dan acara aplikasi lainnya.

## Memperluas aplikasi

Anda dapat menambahkan fungsionalitas kustom ke aplikasi Anda dengan memodifikasi nilai yang Anda berikan di jendela AWS CloudFormation template sebagai berikut:

- `EndPointAcceptContentType`: Anda dapat mengubah nilai ini jika SageMaker titik akhir Anda tidak menerima frame dalam format JPG. Format berikut ini didukung:
  - `image/jpeg`
  - `image/png`
  - `image/bmp`
  - `image/gif`
  - `application/x-image`
- `LambdaFunctionBucket`, `LambdaFunctionKey`: Pengaturan yang disediakan menggunakan AWS Lambda fungsi yang memproses SageMaker output dan menuliskannya ke CloudWatch Log. Jika Anda ingin mengirim SageMaker output ke tempat lain, Anda dapat menyediakan fungsi Lambda Anda sendiri.
- `Filter Tag`: Jika Anda memiliki aliran yang ditandai menggunakan [the section called “TagStream”](#) tindakan, Anda dapat menentukan tag aliran yang ingin Anda proses. Misalnya, jika Anda memiliki dua aliran yang memiliki `Location` kunci dengan nilai `Front` dan `Parking`, Anda akan memfilter untuk hanya menggunakan aliran tersebut menggunakan entri berikut:

```
{"key":"Location","values":["Front","Parking"]}
```

## Membersihkan aplikasi

Setelah Anda selesai dengan aplikasi yang Anda buat untuk tutorial ini, kami sarankan Anda menghapus sumber daya apa pun yang tidak ingin Anda simpan, untuk menghindari biaya yang sedang berlangsung.

1. SageMaker endpoint: Jika Anda membuat SageMaker endpoint untuk tutorial ini daripada menggunakan endpoint yang ada, hapus endpoint. Di panel SageMaker kontrol, pilih Konfigurasi titik akhir. Pilih titik akhir yang Anda buat, dan pilih Tindakan, Hapus. Konfirmasi penghapusan.
2. SageMaker notebook: Di SageMaker konsol, pilih instance Notebook. Pilih buku catatan yang Anda buat, dan pilih Actions, Stop. Ketika buku catatan menunjukkan bahwa Statusnya Dihentikan, pilih Tindakan, Hapus. Konfirmasi penghapusan.

### Note

Untuk informasi selengkapnya tentang membersihkan SageMaker sumber daya, lihat [Membersihkan](#) di [panduan SageMaker pengembang](#).

3. SageMaker kebijakan eksekusi: Di konsol IAM, di panel navigasi, pilih Kebijakan. Pilih kebijakan yang Anda buat untuk tutorial ini. Nama kebijakan ini mirip dengan yang berikut: `AmazonSageMaker-ExecutionPolicy-timestamp`.  
Pilih Tindakan kebijakan, Hapus. Konfirmasi penghapusan.
4. SageMaker peran eksekusi: Di konsol IAM, di panel navigasi, pilih Peran. Pilih peran yang Anda buat untuk tutorial ini. Nama perannya mirip dengan yang berikut: `AmazonSageMaker-ExecutionRole-timestamp`.  
Pilih Hapus peran. Konfirmasi penghapusan.
5. AWS CloudFormation stack: Di AWS CloudFormation konsol, pilih tumpukan yang Anda buat untuk tutorial ini. Pilih Tindakan, Hapus Tumpukan. Konfirmasi penghapusan.
6. Bucket Amazon S3: Di konsol Amazon S3, pilih bucket yang Anda buat untuk menyimpan aset SageMaker. Pilih Hapus. Masukkan nama bucket dan pilih Konfirmasi untuk mengonfirmasi penghapusan.
7. Aliran video Kinesis: Di konsol Kinesis Video Streams, pilih aliran video yang Anda buat untuk aplikasi. Pilih Hapus. Konfirmasi penghapusan.



# Memantau Aliran Video Amazon Kinesis

Pemantauan merupakan bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon Kinesis Video Streams AWS dan solusi Anda. Kami menyarankan untuk mengumpulkan data pemantauan dari semua bagian AWS solusi Anda untuk membantu Anda men-debug kegagalan multi-titik jika terjadi. Sebelum Anda mulai memantau Amazon Kinesis Video Streams, kami sarankan Anda membuat rencana pemantauan yang mencakup jawaban atas pertanyaan-pertanyaan berikut:

- Apa tujuan pemantauan Anda?
- Sumber daya apa yang akan Anda pantau?
- Seberapa sering Anda akan memantau sumber daya ini?
- Alat pemantauan apa yang akan Anda gunakan?
- Siapa yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Setelah Anda menentukan sasaran pemantauan dan membuat rencana pemantauan, langkah selanjutnya adalah menetapkan dasar untuk kinerja Amazon Kinesis Video Streams normal di lingkungan Anda. Anda harus mengukur kinerja Amazon Kinesis Video Streams pada berbagai waktu dan dalam kondisi pemuatan yang berbeda. Saat Anda memantau Amazon Kinesis Video Streams, simpan riwayat data pemantauan yang telah Anda kumpulkan. Anda dapat membandingkan kinerja Amazon Kinesis Video Streams saat ini dengan data historis ini untuk membantu Anda mengidentifikasi pola kinerja normal dan anomali kinerja, serta merancang metode untuk mengatasi masalah yang mungkin timbul.

## Topik

- [Memantau metrik Amazon Kinesis Video Streams dengan CloudWatch](#)
- [Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch](#)
- [Mencatat panggilan API Amazon Kinesis Video Streams dengan AWS CloudTrail](#)

## Memantau metrik Amazon Kinesis Video Streams dengan CloudWatch

Anda dapat memantau aliran video Kinesis menggunakan Amazon CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon Kinesis Video Streams menjadi metrik

hampir real-time yang dapat dibaca. Statistik ini dicatat untuk jangka waktu 15 bulan sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang kinerja aplikasi atau layanan web Anda.

Di konsol [Amazon Kinesis Video Streams](#), Anda CloudWatch dapat melihat metrik untuk streaming video Amazon Kinesis dengan dua cara:

- Di halaman Dasbor, pilih tab Streaming video di bagian Metrik tingkat akun untuk Wilayah Saat Ini.
- Pilih tab Pemantauan di halaman detail aliran video.

Amazon Kinesis Video Streams menyediakan metrik berikut:


Metrik	Deskripsi
ArchivedFragmentsConsumed.Media	Jumlah titik kuota media fragmen yang dikonsumsi oleh semua API. Untuk penjelasan tentang konsep poin kuota, lihat <a href="#">the section called “Kuota fragmen-metadata dan fragmen-media”</a> .  Unit: Hitungan
ArchivedFragmentsConsumed.Metadata	Jumlah titik kuota metadata fragmen yang dikonsumsi oleh semua API. Untuk penjelasan tentang konsep poin kuota, lihat <a href="#">the section called “Kuota fragmen-metadata dan fragmen-media”</a> .  Unit: Hitungan
PutMedia.Requests	Jumlah permintaan PutMedia API untuk aliran tertentu.  Unit: Hitungan
PutMedia.IncomingBytes	Jumlah byte yang diterima sebagai bagian dari PutMedia aliran.  Unit: Byte


Metrik	Deskripsi
<code>PutMedia.IncomingFragments</code>	<p>Jumlah fragmen lengkap yang diterima sebagai bagian dari PutMedia aliran.</p> <p>Unit: Hitungan</p>
<code>PutMedia.IncomingFrames</code>	<p>Jumlah frame lengkap yang diterima sebagai bagian dari PutMedia aliran.</p> <p>Unit: Hitungan</p>
<code>PutMedia.ActiveConnections</code>	<p>Jumlah total koneksi ke host layanan.</p> <p>Unit: Hitungan</p>
<code>PutMedia.ConnectionErrors</code>	<p>Kesalahan saat membuat PutMedia koneksi untuk aliran.</p> <p>Unit: Hitungan</p>
<code>PutMedia.FragmentIngestionLatency</code>	<p>Perbedaan waktu antara saat byte pertama dan terakhir dari sebuah fragmen diterima oleh Amazon Kinesis Video Streams.</p> <p>Unit: Milidetik</p>
<code>PutMedia.FragmentPersistLatency</code>	<p>Waktu yang diambil dari saat data fragmen lengkap diterima dan diarsipkan.</p> <p>Unit: Hitungan</p>
<code>PutMedia.Latency</code>	<p>Perbedaan waktu antara permintaan dan respons HTTP dari InletService saat membuat koneksi.</p> <p>Unit: Hitungan</p>

Metrik	Deskripsi
<code>PutMedia.BufferingAckLatency</code>	<p>Perbedaan waktu antara saat byte pertama fragmen baru diterima oleh Amazon Kinesis Video Streams dan saat Buffering ACK dikirim untuk fragmen.</p> <p>Unit: Milidetik</p>
<code>PutMedia.ReceivedAckLatency</code>	<p>Perbedaan waktu antara saat byte terakhir dari fragmen baru diterima oleh Amazon Kinesis Video Streams dan ketika ACK Diterima dikirim untuk fragmen.</p> <p>Unit: Milidetik</p>
<code>PutMedia.PersistedAckLatency</code>	<p>Perbedaan waktu antara saat byte terakhir dari fragmen baru diterima oleh Amazon Kinesis Video Streams dan ketika ACK Bertahan dikirim untuk fragmen.</p> <p>Unit: Milidetik</p>
<code>PutMedia.ErrorAckCount</code>	<p>Jumlah Error ACK yang dikirim saat melakukan <code>PutMedia</code> streaming.</p> <p>Unit: Hitungan</p>
<code>PutMedia.Success</code>	<p>1 untuk setiap fragmen berhasil ditulis; 0 untuk setiap fragmen yang gagal. Nilai rata-rata metrik ini menunjukkan berapa banyak fragmen lengkap dan valid yang dikirim.</p> <p>Unit: Hitungan</p>
<code>GetMedia.Requests</code>	<p>Jumlah permintaan <code>GetMedia</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
<code>GetMedia.OutgoingBytes</code>	<p>Jumlah total byte yang dikirim keluar dari layanan sebagai bagian dari <code>GetMedia</code> API untuk aliran tertentu.</p> <p>Unit: Byte</p>
<code>GetMedia.OutgoingFragments</code>	<p>Jumlah fragmen yang dikirim saat melakukan <code>GetMedia</code> untuk aliran.</p> <p>Unit: Hitungan</p>
<code>GetMedia.OutgoingFrames</code>	<p>Jumlah frame yang dikirim selama <code>GetMedia</code> pada aliran yang diberikan.</p> <p>Unit: Hitungan</p>
<code>GetMedia.MillisBehindNow</code>	<p>Perbedaan waktu antara stempel waktu server saat ini dan stempel waktu server dari fragmen terakhir yang dikirim.</p> <p>Unit: Milidetik</p>
<code>GetMedia.ConnectionErrors</code>	<p>Jumlah koneksi yang tidak berhasil dibuat.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
GetMedia.Success	<p>1 untuk setiap fragmen berhasil dikirim; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 401 1508 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
GetMediaForFragmentList.OutgoingBytes	<p>Jumlah total byte yang dikirim keluar dari layanan sebagai bagian dari GetMediaForFragmentList API untuk aliran tertentu.</p> <p>Unit: Byte</p>
GetMediaForFragmentList.OutgoingFragments	<p>Jumlah total fragmen yang dikirim keluar dari layanan sebagai bagian dari GetMediaForFragmentList API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>
GetMediaForFragmentList.OutgoingFrames	<p>Jumlah total frame yang dikirim keluar dari layanan sebagai bagian dari GetMediaForFragmentList API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
<code>GetMediaForFragmentList.Requests</code>	<p>Jumlah permintaan <code>GetMediaForFragmentList</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>
<code>GetMediaForFragmentList.Success</code>	<p>1 untuk setiap fragmen berhasil dikirim; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 606 1507 1066" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
<code>ListFragments.Latency</code>	<p>Latensi <code>ListFragments</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>ListFragments.Requests</code>	<p>Jumlah permintaan <code>ListFragments</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
ListFragments.Success	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 394 1508 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
GetHLSStreamingSessionURL.Latency	<p>Latensi GetHLSStreamingSessionURL API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
GetHLSStreamingSessionURL.Requests	<p>Jumlah permintaan GetHLSStreamingSessionURL API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>





Metrik	Deskripsi
<code>GetHLSStreamingSessionURL.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="751 401 1507 856"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
<code>GetHLSMasterPlaylist.Latency</code>	<p>Latensi <code>GetHLSMasterPlaylist</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetHLSMasterPlaylist.Requests</code>	<p>Jumlah permintaan <code>GetHLSMasterPlaylist</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
<code>GetHLSMasterPlaylist.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 401 1508 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
<code>GetHLSMediaPlaylist.Latency</code>	<p>Latensi <code>GetHLSMediaPlaylist</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetHLSMediaPlaylist.Requests</code>	<p>Jumlah permintaan <code>GetHLSMediaPlaylist</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
<code>GetHLSMediaPlaylist.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="751 401 1507 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
<code>GetMP4InitFragment.Latency</code>	<p>Latensi <code>GetMP4InitFragment</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetMP4InitFragment.Requests</code>	<p>Jumlah permintaan <code>GetMP4InitFragment</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>

Metrik	Deskripsi
<code>GetMP4InitFragment.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 394 1508 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
<code>GetMP4MediaFragment.Latency</code>	<p>Latensi <code>GetMP4MediaFragment</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetMP4MediaFragment.Requests</code>	<p>Jumlah permintaan <code>GetMP4MediaFragment</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>


Metrik	Deskripsi
<code>GetMP4MediaFragment.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 401 1507 856"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
<code>GetMP4MediaFragment.OutgoingBytes</code>	<p>Jumlah total byte yang dikirim keluar dari layanan sebagai bagian dari <code>GetMP4MediaFragment</code> API untuk aliran tertentu.</p> <p>Unit: Byte</p>
<code>GetTSFragment.Latency</code>	<p>Latensi <code>GetTSFragment</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetTSFragment.Requests</code>	<p>Jumlah permintaan <code>GetTSFragment</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>

Metrik	Deskripsi
<code>GetTSFragment.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 401 1507 856"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
<code>GetTSFragment.OutgoingBytes</code>	<p>Jumlah total byte yang dikirim keluar dari layanan sebagai bagian dari <code>GetTSFragment</code> API untuk aliran tertentu.</p> <p>Unit: Byte</p>
<code>GetDASHStreamingSessionURL.Latency</code>	<p>Latensi <code>GetDASHStreamingSessionURL</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetDASHStreamingSessionURL.Requests</code>	<p>Jumlah permintaan <code>GetDASHStreamingSessionURL</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>

Metrik	Deskripsi
<code>GetDASHStreamingSessionURL.Success</code>	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="751 401 1507 856"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
<code>GetDASHManifest.Latency</code>	<p>Latensi <code>GetDASHManifest</code> API memanggil nama aliran yang diberikan.</p> <p>Unit: Milidetik</p>
<code>GetDASHManifest.Requests</code>	<p>Jumlah permintaan <code>GetDASHManifest</code> API untuk aliran tertentu.</p> <p>Unit: Hitungan</p>

Metrik	Deskripsi
GetDASHManifest.Success	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="748 401 1508 856"><p> <b>Note</b></p><p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p></div> <p>Unit: Hitungan</p>
GetClip.Latency	<p>Latensi GetClip API memanggil nama streaming video yang diberikan.</p> <p>Unit: Milidetik</p>
GetClip.Requests	<p>Jumlah permintaan GetClip API untuk streaming video tertentu.</p> <p>Unit: Hitungan</p>



Metrik	Deskripsi
GetClip.Success	<p>1 untuk setiap permintaan yang berhasil; 0 untuk setiap kegagalan. Nilai rata-rata menunjukkan tingkat keberhasilan.</p> <div data-bbox="751 401 1507 856" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Kegagalan mencakup 400 kesalahan (pengguna) dan kesalahan 500 (sistem). Untuk informasi selengkapnya tentang mengaktifkan ringkasan permintaan dan tanggapan, termasuk ID AWS permintaan, lihat <a href="#">Pencatatan Ringkasan Permintaan/Respons</a>.</p> </div> <p>Unit: Hitungan</p>
GetClip.OutgoingBytes	<p>Jumlah total byte yang dikirim keluar dari layanan sebagai bagian dari GetClip API untuk streaming video tertentu.</p> <p>Unit: Byte</p>

## CloudWatch panduan metrik

CloudWatch metrik dapat membantu menemukan jawaban atas pertanyaan-pertanyaan berikut:

### Topik

- [Apakah data mencapai layanan Amazon Kinesis Video Streams?](#)
- [Mengapa data tidak berhasil dicerna oleh layanan Amazon Kinesis Video Streams?](#)
- [Mengapa data tidak dapat dibaca dari layanan Amazon Kinesis Video Streams dengan kecepatan yang sama seperti yang dikirim dari produser?](#)
- [Mengapa tidak ada video di konsol, atau mengapa video diputar dengan penundaan?](#)

- [Apa keterlambatan dalam membaca data waktu nyata, dan mengapa klien tertinggal di belakang kepala aliran?](#)
- [Apakah klien membaca data keluar dari aliran video Kinesis, dan pada tingkat berapa?](#)
- [Mengapa klien tidak dapat membaca data dari aliran video Kinesis?](#)

## Apakah data mencapai layanan Amazon Kinesis Video Streams?

Metrik yang relevan:

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

Item tindakan:

- Jika ada penurunan dalam metrik ini, periksa apakah aplikasi Anda masih mengirim data ke layanan.
- Periksa bandwidth jaringan. Jika bandwidth jaringan Anda tidak mencukupi, itu bisa memperlambat laju layanan menerima data.

## Mengapa data tidak berhasil dicerna oleh layanan Amazon Kinesis Video Streams?

Metrik yang relevan:

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

Item tindakan:

- Jika ada peningkatan `PutMedia.ConnectionErrors`, lihat respons HTTP dan kode kesalahan yang diterima oleh klien produsen untuk melihat kesalahan apa yang terjadi saat membuat koneksi.

- Jika ada penurunan `PutMedia.Success` atau peningkatan `PutMedia.ErrorAckCount`, lihat kode kesalahan ack dalam respons ack yang dikirim oleh layanan untuk melihat mengapa konsumsi data gagal. Untuk informasi selengkapnya, lihat [AckErrorCode.Values](#).

Mengapa data tidak dapat dibaca dari layanan Amazon Kinesis Video Streams dengan kecepatan yang sama seperti yang dikirim dari produser?

Metrik yang relevan:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.IncomingBytes`

Item tindakan:

- Jika ada penurunan dalam metrik ini, periksa bandwidth jaringan koneksi Anda. Koneksi bandwidth rendah dapat menyebabkan data mencapai layanan pada tingkat yang lebih rendah.

Mengapa tidak ada video di konsol, atau mengapa video diputar dengan penundaan?

Metrik yang relevan:

- `PutMedia.FragmentIngestionLatency`
- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

Item tindakan:

- Jika ada peningkatan `PutMedia.FragmentIngestionLatency` atau penurunan `PutMedia.IncomingFragments`, periksa bandwidth jaringan dan apakah data masih dikirim.
- Jika ada drop in `PutMedia.Success`, periksa kode kesalahan ack. Untuk informasi selengkapnya, lihat [AckErrorCode.Values](#).

- Jika ada peningkatan `PutMedia.FragmentPersistLatency` atau `ListFragments.Latency`, kemungkinan besar Anda mengalami masalah layanan. Jika kondisi berlanjut untuk jangka waktu yang lama, tanyakan kepada kontak layanan pelanggan Anda untuk melihat apakah ada masalah dengan layanan Anda.

Apa keterlambatan dalam membaca data waktu nyata, dan mengapa klien tertinggal di belakang kepala aliran?

Metrik yang relevan:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

Item tindakan:

- Jika ada peningkatan `GetMedia.ConnectionErrors`, maka konsumen mungkin tertinggal dalam membaca aliran, karena seringnya upaya untuk terhubung kembali ke aliran. Lihatlah kode respon/kesalahan HTTP yang dikembalikan untuk permintaan tersebut. `GetMedia`
- Jika ada penurunan `GetMedia.Success`, kemungkinan karena layanan tidak dapat mengirim data ke konsumen, yang akan mengakibatkan koneksi terputus, dan terhubung kembali dari konsumen, yang akan mengakibatkan konsumen tertinggal di belakang kepala aliran.
- Jika ada peningkatan `GetMedia.MillisBehindNow`, lihat batas bandwidth Anda untuk melihat apakah Anda menerima data pada tingkat yang lebih lambat karena bandwidth yang lebih rendah.

Apakah klien membaca data keluar dari aliran video Kinesis, dan pada tingkat berapa?

Metrik yang relevan:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

Item tindakan:

- Metrik ini menunjukkan tingkat di mana data real-time dan arsip sedang dibaca.

## Mengapa klien tidak dapat membaca data dari aliran video Kinesis?

Metrik yang relevan:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

Item tindakan:

- Jika ada peningkatan `GetMedia.ConnectionErrors`, lihat respons HTTP dan kode kesalahan yang dikembalikan oleh `GetMedia` permintaan. Untuk informasi selengkapnya, lihat [AckErrorCode.Values](#).
- Jika Anda mencoba membaca data terbaru atau langsung, periksa `PutMedia.IncomingBytes` untuk melihat apakah ada data yang masuk ke aliran untuk layanan yang akan dikirim ke konsumen.
- Jika ada penurunan `GetMedia.Success` atau `GetMediaForFragmentList.Success`, kemungkinan karena layanan tidak dapat mengirim data ke konsumen. Jika kondisi berlanjut untuk jangka waktu yang lama, tanyakan kepada kontak layanan pelanggan Anda untuk melihat apakah ada masalah dengan layanan Anda.


## Memantau Agen Edge Streams Video Amazon Kinesis dengan CloudWatch


Anda dapat memantau Agen Edge Amazon Kinesis Video Streams menggunakan CloudWatch Amazon, yang mengumpulkan dan memproses data mentah menjadi metrik hampir real-time yang dapat dibaca. Statistik ini dicatat untuk jangka waktu 15 bulan. Dengan informasi historis ini, Anda dapat memperoleh perspektif yang lebih baik tentang kinerja aplikasi web atau Amazon Kinesis Video Streams Edge Agent.

Untuk melihat metrik, lakukan hal berikut:

1. Masuk ke AWS Management Console dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di navigasi kiri, di bawah Metrik, pilih Semua Metrik.
3. Pilih tab Browse, lalu pilih namespace EdgeRuntimeAgentkhusus.

Amazon Kinesis Video Streams Edge Agent menerbitkan metrik berikut di bawah namespace: EdgeRuntimeAgent

Dimensi	Status	Deskripsi
Nama aliran, RecordJob	Berjalan	Menerbitkan terus menerus saat RecordJob sedang berjalan.  Unit: Tidak ada. "1" diterbitkan selama RecordJob dalam keadaan ini.
	FatalError	Menerbitkan jika kesalahan RecordJob fatal.  Unit: Tidak ada. "1" diterbitkan sekali, ketika peristiwa ini terjadi.  <div style="border: 1px solid #007bff; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b> Lihat log untuk informasi tambahan.</p> </div>
	Selesai	Menerbitkan ketika a RecordJob selesai.  Unit: Tidak ada. "1" diterbitkan sekali, ketika peristiwa ini terjadi.
Nama aliran, UploadJob	Berjalan	Menerbitkan terus menerus saat UploadJob sedang berjalan.  Unit: Tidak ada. "1" diterbitkan selama UploadJob dalam keadaan ini.
	FatalError	Menerbitkan jika kesalahan UploadJob fatal.

Dimensi	Status	Deskripsi
		<p>Unit: Tidak ada. "1" diterbitkan sekali, ketika peristiwa ini terjadi.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b> Lihat log untuk informasi tambahan.</p> </div>
	Selesai	<p>Menerbitkan ketika UploadJob selesai.</p> <p>Unit: Tidak ada. "1" diterbitkan sekali, ketika peristiwa ini terjadi.</p>
Nama Streaming	Percentag eSpaceUsed	<p>Ini adalah persentase yang digunakan dari total ruang yang dialokasikan dalam konfigurasi Amazon Kinesis Video Streams Edge Agent untuk media perekaman. Untuk informasi selengkapnya, lihat <a href="#">the section called "LocalSizeConfig"</a>.</p> <p>Unit: Persentase (skala 0—1).</p>
Nama benda	Hidup	<p>Menerbitkan setiap menit dari Agen Edge Amazon Kinesis Video Streams, terlepas dari konfigurasi apa pun yang berjalan di dalamnya.</p> <p>Ini dapat digunakan untuk memahami apakah Agen Tepi Amazon Kinesis Video Streams masih hidup dan siap menerima konfigurasi.</p> <p>Unit: Tidak ada. "1" diterbitkan setiap menit.</p>
	RecordJobs.Healthy JobCount	<p>Jumlah total pekerjaan rekaman yang berjalan dan terjadwal di Agen Edge Amazon Kinesis Video Streams.</p> <p>Unit: Hitung.</p>
	UploadJobs.Healthy JobCount	<p>Jumlah total pekerjaan pengunggahan yang berjalan dan dijadwalkan di Agen Edge Amazon Kinesis Video Streams.</p> <p>Unit: Hitung.</p>

Dimensi	Status	Deskripsi
	RecordJobs.UnhealthyJobCount	Jumlah total pekerjaan rekor yang saat ini salah. Unit: Hitung.
	UploadJobs.UnhealthyJobCount	Jumlah total pekerjaan unggahan yang saat ini salah. Unit: Hitung.
	RecordJobs.RunningJobCount	Jumlah total pekerjaan rekor yang aktif menjalankan. Unit: Hitung.
	UploadJobs.RunningJobCount	Jumlah total pekerjaan upload yang aktif berjalan. Unit: Hitung.
	RecordJobs.EdgeConfigCount	Jumlah total konfigurasi rekaman dalam proses di Amazon Kinesis Video Streams Edge Agent. Unit: Hitung.
	UploadJobs.EdgeConfigCount	Jumlah total konfigurasi unggahan yang sedang diproses di Amazon Kinesis Video Streams Edge Agent. Unit: Hitung.

## CloudWatch panduan metrik untuk Agen Edge Streams Amazon Kinesis Video Streams

CloudWatch metrik dapat berguna untuk menemukan jawaban atas pertanyaan-pertanyaan berikut:

### Topik

- [Apakah Agen Edge Streams Video Amazon Kinesis memiliki cukup ruang untuk merekam?](#)
- [Apakah Agen Edge Streams Video Amazon Kinesis masih hidup?](#)
- [Apakah ada pekerjaan yang tidak sehat?](#)



- [Apakah ada pekerjaan yang membutuhkan intervensi eksternal?](#)

Apakah Agen Edge Streams Video Amazon Kinesis memiliki cukup ruang untuk merekam?

Metrik yang relevan: `PercentageSpaceUsed`

Tindakan: Tidak ada tindakan yang diperlukan.

Apakah Agen Edge Streams Video Amazon Kinesis masih hidup?

Metrik yang relevan: `Alive`

Tindakan: Jika suatu saat Anda berhenti menerima metrik ini, itu berarti Agen Tepi Amazon Kinesis Video Streams menemukan satu atau beberapa hal berikut:

- Masalah runtime aplikasi: memori atau kendala sumber daya lainnya, bug, dan sebagainya
- AWS IoT Perangkat yang dijalankan agen saat shutdown, crash, atau dihentikan
- AWS IoT Perangkat tidak memiliki konektivitas jaringan

Apakah ada pekerjaan yang tidak sehat?

Metrik yang relevan:

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

Tindakan: Periksa log dan cari `FatalError` metriknya.

- Jika `FatalError` metrik ada, kesalahan fatal ditemukan dan Anda perlu me-restart pekerjaan secara manual. Periksa log dan perbaiki masalah sebelum menggunakan `StartEdgeConfigurationUpdate` untuk memulai ulang pekerjaan secara manual.
- Jika `FatalError` metrik tidak ada, kesalahan sementara (non-fatal) ditemukan dan Amazon Kinesis Video Streams Edge Agent mencoba kembali pekerjaan tersebut.

**Note**

Agar agen mencoba kembali pekerjaan yang salah, gunakan [the section called “StartEdgeConfigurationUpdate”](#)

Apakah ada pekerjaan yang membutuhkan intervensi eksternal?

Metrik yang relevan:

- `PercentageSpaceUsed`— Jika ini melebihi nilai tertentu, pekerjaan rekaman dijeda dan dilanjutkan hanya ketika ruang tersedia (ketika media keluar dari retensi). Anda dapat mengirim konfigurasi yang diperbarui dengan yang lebih tinggi `MaxLocalMediaSizeInMB` untuk segera memperbarui pekerjaan.
- `RecordJob.FatalError/UploadJob.FatalError`— Selidiki log agen dan kirim konfigurasi lagi agar pekerjaan dapat dilanjutkan.

Tindakan: Lakukan panggilan API dengan konfigurasi untuk memulai ulang pekerjaan yang mengalami masalah ini.

## Mencatat panggilan API Amazon Kinesis Video Streams dengan AWS CloudTrail

Amazon Kinesis Video Streams AWS CloudTrail berfungsi dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, Layanan AWS atau di Amazon Kinesis Video Streams. CloudTrail menangkap semua panggilan API untuk Amazon Kinesis Video Streams sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Amazon Kinesis Video Streams dan panggilan kode ke operasi API Amazon Kinesis Video Streams. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Amazon Kinesis Video Streams. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Amazon Kinesis Video Streams, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat [Panduan AWS CloudTrail Pengguna](#).

## Amazon Kinesis Video Streams dan CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas peristiwa yang didukung terjadi di Amazon Kinesis Video Streams, aktivitas tersebut direkam CloudTrail dalam suatu peristiwa bersama AWS dengan peristiwa layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat [Melihat Acara dengan Riwayat CloudTrail Acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amazon Kinesis Video Streams, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi lainnya Layanan AWS untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran umum untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Amazon Kinesis Video Streams mendukung pencatatan tindakan berikut sebagai CloudTrail peristiwa dalam file log:

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

Setiap peristiwa atau entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut:

- Apakah permintaan dibuat dengan root atau kredensi pengguna
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan
- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

Untuk informasi selengkapnya, lihat Elemen [CloudTrailUserIdentity](#).

## Contoh: Entri file log Amazon Kinesis Video Streams

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa merepresentasikan satu permintaan dari sumber apa pun dan menyertakan informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan [CreateStream](#) tindakan.

```
{
  "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"
}
]
```

# Kuota layanan Kinesis Video Streams

Kinesis Video Streams memiliki kuota layanan berikut:

## Important

Kuota layanan berikut adalah soft [s], yang dapat ditingkatkan dengan mengirimkan tiket dukungan, atau hard [h], yang tidak dapat ditingkatkan. Anda akan melihat [s] dan [h] di samping kuota layanan individual pada tabel di bawah ini.

## Kuota layanan API bidang kontrol


Bagian berikut menjelaskan kuota layanan untuk API bidang kontrol. TPS adalah singkatan dari transaksi per detik.

Ketika batas permintaan tingkat akun atau tingkat sumber daya tercapai, a dilemparkan. `ClientLimitExceededException`

Kuota layanan API bidang kontrol

API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
<a href="#">the section called "CreateStream"</a>	50 TPS [s]	10000 aliran per akun di Wilayah AS Timur (Virginia N.) dan AS Barat (Oregon). 5000 aliran per akun di semua Wilayah lain yang didukung.		Perangkat, CLI, akses berbasis SDK, dan konsol semuanya dapat menjalankan API ini. Hanya satu panggilan API yang berhasil jika aliran belum ada.



API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
		<p> <b>Note</b></p> <p>Batas ini dapat ditingkatkan hingga 100.000 (atau lebih) aliran per akun. Masuk ke AWS Management Console at <a href="https://console.aws.amazon.com/">https://console.aws.amazon.com/</a> dan kirimkan <a href="#">kasus peningkatan batas layanan</a> untuk Kinesis</p>		

API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
		Video Streams untuk meminta peningkatan batas ini.		
<a href="#">the section called "DeleteEdgeConfiguration"</a>	10 TPS [h]	N/A	1 TPS [h]	
<a href="#">the section called "DeleteStream"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "DescribeEdgeConfiguration"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "DescribeImageGenerationConfiguration"</a>	50 TPS [h]	N/A	5 TPS [h]	

API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
<a href="#">the section called “Describe MappedResourceConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “Describe NotificationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “Describe Stream”</a>	300 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “GetDataEndpoint”</a>	300 TPS [h]	N/A	5 TPS [h]	Dipanggil setiap 45 menit untuk menyegarkan token streaming untuk sebagian PutMedia besar/kasus GetMedia penggunaan. Caching data endpoint aman jika aplikasi memuat ulang mereka pada kegagalan.
<a href="#">the section called “ListEdgeAgentConfigurations”</a>	50 TPS [h]	N/A	N/A	

API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
<a href="#">the section called "ListStreams"</a>	50 TPS [h]	N/A		
<a href="#">the section called "ListTags ForStream"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "StartEdgeConfigurationUpdate"</a>	10 TPS [h]	N/A	1 TPS [h]	
<a href="#">the section called "TagStream"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "UntagStream"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "UpdateDataRetention"</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called "UpdateImageGenerationConfiguration"</a>	50 TPS [h]	N/A	5 TPS [h]	

API	Batas akun: Permintaan	Batas akun: Streams	Batas tingkat aliran	Pengecualian dan catatan yang relevan
<a href="#">the section called “UpdateNotificationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “UpdateStream”</a>	50 TPS [h]	N/A	5 TPS [h]	

## Kuota layanan API media dan media arsip

Bagian berikut menjelaskan kuota layanan untuk media dan API media yang diarsipkan.

Ketika batas permintaan tingkat akun atau tingkat sumber daya tercapai, a dilemparkan. `ClientLimitExceededException`


Ketika batas tingkat koneksi tercapai, a `ConnectionLimitExceededException` dilemparkan.


Kesalahan atau acks berikut dilemparkan ketika batas tingkat fragmen tercapai:

- `MIN_FRAGMENT_DURATION_REACHED`Ack dikembalikan untuk fragmen di bawah durasi minimum.
- `MAX_FRAGMENT_DURATION_REACHED`Ack dikembalikan untuk fragmen di atas durasi maksimum.
- `MAX_FRAGMENT_SIZE`Ack dikembalikan untuk fragmen di atas ukuran data maksimum.
- `FragmentLimitExceeded`Pengecualian dilemparkan jika batas fragmen tercapai dalam suatu `GetMediaForFragmentList` operasi.

## Kuota layanan API bidang data

API	Batas tingkat aliran	Batas tingkat koneksi	Batas bandwidth	Batas tingkat fragmen	Pengecualian dan catatan yang relevan
<a href="#">the section called "PutMedia"</a>	5 TPS [h]	1 [s]	12,5 MB/detik, atau 100 Mbps per aliran	<ul style="list-style-type: none"> <li>Durasi fragmen minimum: 1 detik [h]</li> <li>Durasi fragmen maksimum 20 detik [h]</li> <li>Ukuran fragmen maksimum 50 MB [h]</li> <li>Jumlah maksimum trek: 3 [s]</li> <li>Fragmen maksimum yang dikirim per detik: 5 [h]</li> <li>Batas metadata fragmen maksimum 10 tag [h]</li> </ul>	PutMediaPermintaan tipikal berisi data selama beberapa detik, menghasilkan TPS per aliran yang lebih rendah. Jika ada beberapa koneksi bersamaan yang melebihi kuota, koneksi terakhir diterima.

API	Batas tingkat aliran	Batas tingkat koneksi	Batas bandwidth	Batas tingkat fragmen	Pengecualian dan catatan yang relevan
<a href="#">the section called “GetClip”</a>	N/A	N/A	Batas ukuran 100 MB [h]	Jumlah fragmen maksimum: 200 [h]	
<a href="#">the section called “GetDASHStreamingSessionURL”</a>	25 TPS [h]	N/A	N/A	N/A	
<a href="#">the section called “GetHLSStreamingSessionURL”</a>	25 TPS [h]	N/A	N/A	N/A	
<a href="#">the section called “GetImages”</a>	N/A	N/A	100 MB [h]	N/A	<p>Jumlah maksimum gambar per permintaan adalah 100 [h].</p> <div data-bbox="1133 1306 1507 1814" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px;"> <p> <b>Note</b></p> <p>Nilai minimum untuk SamplingInterval adalah 200 milidetik (ms), yaitu 5 gambar per detik.</p> </div>

API	Batas tingkat aliran	Batas tingkat koneksi	Batas bandwidth	Batas tingkat fragmen	Pengecualian dan catatan yang relevan
<a href="#">the section called "GetMedia"</a>	5 TPS [h]	3 [s]	25 MB/s atau 200 Mbps [s]	Fragmen maksimum yang dikirim per detik: 6 [h]	<p>Klien konsumen yang unik tidak perlu lebih dari dua atau tiga TPS karena setelah koneksi dibuat, aplikasi harus terus membaca.</p> <p>Jika fragmen tipikal kira-kira 5 MB, batas ini berarti ~75 MBps per aliran video Kinesis. Aliran seperti itu akan memiliki bitrate keluar 2x bitrate masuk maksimum aliran.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>GetMedia tidak digunakan untuk pemutaran HLS/DASH.</p> </div>
<a href="#">the section called "GetMediaForFragmentList"</a>	N/A	5 [s]	25 MB/s atau 200 Mbps [s]	Jumlah fragmen maksimum: 1000 [h]	Lima aplikasi konsumsi berbasis fragmen dapat digunakan secara bersamaan. GetMediaForFragmentList Koneksi lebih lanjut ditolak.



## Kuota layanan API protokol pemutaran video

API	Batas tingkat sesi	Batas tingkat fragmen
<a href="#">GetDash ManifestPlaylist</a>	5 TPS [h]	Jumlah maksimum fragmen per daftar putar: 5000 [h]
<a href="#">GetHL MasterPlaylist</a>	5 TPS [h]	N/A
<a href="#">GetHL MediaPlaylist</a>	5 TPS [h]	Jumlah maksimum fragmen per daftar putar: 5000 [h]
<a href="#">GetMP4 InitFragment</a>	5 TPS [h]	N/A
<a href="#">GetMP4 MediaFragment</a>	20 TPS [h]	N/A
<a href="#">GettsFragment</a>	20 TPS [h]	N/A

## Kuota fragmen-metadana dan fragmen-media

[API Kinesis Video Streams untuk mengakses](#) media yang diarsipkan dibatasi berdasarkan jumlah fragmen yang diminta, bukan jumlah panggilan API. API dibatasi oleh jumlah metadata fragmen dan jumlah media fragmen yang diminta. Metadata fragmen dan kuota media fragmen diterapkan per aliran. Dengan kata lain, permintaan untuk metadata fragmen atau media dalam satu aliran tidak berlaku untuk kuota aliran lain. Namun, dalam aliran tertentu, setiap kuota dibagi di beberapa API. Ini berarti bahwa, untuk aliran tertentu, permintaan fragmen di berbagai API menggunakan kuota yang sama. Ketika metadata fragmen atau kuota media fragmen untuk aliran terlampaui, API akan mengembalikan file. `ClientLimitExceededException` Tabel berikut menunjukkan bagaimana API mengkonsumsi dari masing-masing dari dua jenis kuota. Untuk kolom kedua dalam tabel ini, asumsikan bahwa jika aliran memiliki kuota N, itu berarti API memiliki titik N untuk dikonsumsi dari jenis kuota tersebut untuk aliran tersebut. `GetClipAPI` muncul di kedua tabel.

### Konsumsi kuota metadata fragmen

API	Jumlah poin kuota yang dikonsumsi per permintaan	Kuota bersama (N)
<a href="#">the section called “ListFragments”</a>	Nilai <code>MaxResults</code> parameter	10000 poin kuota per detik, per aliran [h]

API	Jumlah poin kuota yang dikonsumsi per permintaan	Kuota bersama (N)
<a href="#">the section called “GetClip”</a>	Jumlah fragmen dalam klip yang dihasilkan	
<a href="#">GetHLSMediaPlaylist</a>	Nilai MaxMediaPlaylistFragmentResults parameter	
<a href="#">GetDASHManifest</a>	Nilai MaxManifestFragmentResults parameter	
<a href="#">the section called “GetImages”</a>	Nilai 400+jumlah maksimum gambar yang diminta	

#### Konsumsi kuota media fragmen

API	Jumlah poin kuota yang dikonsumsi per permintaan	Kuota bersama (N)
<a href="#">the section called “GetMediaForFragmentList”</a>	Jumlah fragmen dalam parameter Fragmen	500 poin kuota per detik, per aliran [h]
<a href="#">the section called “GetClip”</a>	Jumlah fragmen dalam klip yang dihasilkan	
<a href="#">GetMP4MediaFragment</a>	1	
<a href="#">GetTSFragment</a>	1	
<a href="#">the section called “GetImages”</a>	Jumlah maksimum gambar yang diminta	

Misalnya, dengan kuota 500 media fragmen per detik, pola panggilan berikut untuk aliran tertentu didukung:

- 5 permintaan per detik untuk GetClip dengan 100 fragmen di setiap klip.

- 100 permintaan per detik untuk `GetClip` dengan 5 fragmen di setiap klip.
- 2 permintaan per detik untuk `GetClip` dengan 100 fragmen di setiap klip dan 3 permintaan per detik untuk `GetMediaForFragmentList` di setiap klip.
- 400 permintaan per detik untuk `GetMP4MediaFragment` dan 100 permintaan per detik untuk `GetTSFragment`.

Kuota ini memiliki implikasi penting mengenai jumlah sesi HLS dan MPEG-DASH yang dapat didukung per aliran. Tidak ada batasan jumlah sesi HLS dan DASH yang dapat digunakan oleh pemutar media pada waktu tertentu. Oleh karena itu, penting bahwa aplikasi pemutaran tidak mengizinkan terlalu banyak sesi untuk digunakan secara bersamaan. Dua contoh berikut menjelaskan cara menentukan jumlah sesi pemutaran bersamaan yang dapat didukung:

#### Contoh 1: Streaming langsung

Dalam skenario streaming langsung dengan HLS dengan fragmen durasi 1 detik, trek audio dan video, dan `MaxMediaPlaylistFragmentResults` diatur ke lima, pemutar media biasanya membuat dua panggilan `GetHLSMediaPlaylist` per detik. Satu panggilan adalah untuk metadata video terbaru dan satu lagi untuk metadata audio yang sesuai. Kedua panggilan tersebut mengkonsumsi masing-masing lima poin kuota metadata fragmen. Itu juga membuat dua panggilan `GetMP4MediaFragment` per detik: satu panggilan untuk video terbaru dan satu lagi untuk audio yang sesuai. Setiap panggilan mengkonsumsi satu token media fragmen, sehingga dua token dikonsumsi secara total.

Dalam skenario ini, hingga 250 sesi pemutaran bersamaan dapat didukung. Dengan 250 sesi, skenario ini mengkonsumsi 2.500 poin kuota metadata fragmen per detik (jauh di bawah kuota 10.000) dan 500 poin kuota media fragmen per detik.

#### Contoh 2: Pemutaran sesuai permintaan

Dalam skenario pemutaran sesuai permintaan dari acara sebelumnya dengan MPEG-DASH, trek audio dan video dan `MaxManifestFragmentResults` diatur ke 1.000, pemutar media biasanya menelepon `GetDASHManifest` sekali pada awal sesi (mengkonsumsi 1.000 poin kuota metadata fragmen) dan panggilan `GetMP4MediaFragment` dengan kecepatan hingga 5 kali per detik (mengkonsumsi 5 titik kuota media fragmen) hingga semua fragmen dimuat. Dalam skenario ini, hingga 10 sesi baru dapat dimulai per detik (tepat di 10.000 metadata fragmen per detik kuota), dan hingga 100 sesi dapat secara aktif memuat media fragmen dengan kecepatan 5 per detik (tepat di 500 fragmen media per detik kuota).

Anda dapat menggunakan `ArchivedFragmentsConsumed.Metadata` dan `ArchivedFragmentsConsumed.Media` memantau konsumsi metadata fragmen dan titik kuota media fragmen, masing-masing. Untuk informasi tentang pemantauan, lihat [Memantau](#).

## Kuota metadata fragmen

Kuota layanan berikut berlaku untuk menambahkan metadata fragmen ke fragmen dalam aliran video Kinesis:

- Anda dapat menambahkan hingga 10 item metadata ke sebuah fragmen.
- Nama metadata fragmen dapat mencapai panjang 128 byte.
- Nilai metadata fragmen bisa mencapai 256 byte panjangnya.
- Nama metadata fragmen tidak dapat dimulai dengan string `""`. AWS Jika item metadata tersebut ditambahkan, `putFragmentMetadata` metode dalam PIC mengembalikan `STATUS_INVALID_METADATA_NAME` kesalahan (kode `0x52000077` kesalahan). Aplikasi Anda kemudian dapat mengabaikan kesalahan (PIC tidak menambahkan item metadata), atau menanggapi kesalahan.

## Tag Stream

Pasangan nilai kunci metadata ini berlaku untuk sumber daya Kinesis Video Streams secara keseluruhan, bukan fragmen individual yang terkandung dalam aliran video Kinesis.

Setiap aliran video Kinesis mendukung hingga 50 tag.

Lihat [the section called "TagStream"](#) pembatasan pada kunci dan nilai tag aliran.

# Memecahkan Masalah Kinesis Video Streams

Gunakan informasi berikut untuk memecahkan masalah umum yang dihadapi dengan Amazon Kinesis Video Streams.

## Topik

- [Memecahkan masalah umum](#)
- [Memecahkan masalah API](#)
- [Memecahkan masalah HLS](#)
- [Memecahkan masalah Java](#)
- [Memecahkan masalah Pustaka Produser](#)
- [Memecahkan masalah Stream Parser Library](#)

## Memecahkan masalah umum

Bagian ini menjelaskan masalah umum yang mungkin Anda temui saat bekerja dengan Kinesis Video Streams.

### Masalah

- [Latensi terlalu tinggi](#)

## Latensi terlalu tinggi

Latensi mungkin disebabkan oleh durasi fragmen yang dikirim ke layanan Kinesis Video Streams. Salah satu cara untuk mengurangi latensi antara produser dan layanan adalah dengan mengkonfigurasi pipa media untuk menghasilkan durasi fragmen yang lebih pendek.

Untuk mengurangi jumlah frame yang dikirim di setiap fragmen, kurangi nilai berikut dikinesis\_video\_gstreamer\_sample\_app.cpp:

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512,
NULL);
```

**Note**

Latensi lebih tinggi di browser Mozilla Firefox karena implementasi internal rendering video.

## Memecahkan masalah API

Bagian ini menjelaskan masalah API yang mungkin Anda temui saat bekerja dengan Kinesis Video Streams.

### Masalah

- [Kesalahan: “Opsi tidak dikenal”](#)
- [Kesalahan: “Tidak dapat menentukan nama layanan/operasi yang akan diotorisasi”](#)
- [Kesalahan: “Gagal menempatkan bingkai di aliran”](#)
- [Kesalahan: “Layanan menutup koneksi sebelum final AckEvent diterima”](#)
- [Kesalahan: “STATUS\\_STORE\\_OUT\\_OF\\_MEMORY”](#)

### Kesalahan: “Opsi tidak dikenal”

GetMedia dan GetMediaForFragmentList bisa gagal dengan kesalahan berikut:

```
Unknown options: <filename>.mkv
```

Kesalahan ini terjadi jika Anda mengkonfigurasi AWS CLI dengan output jenis json. Konfigurasi ulang AWS CLI dengan tipe output default (none). Untuk informasi tentang mengonfigurasi AWS CLI, lihat [mengkonfigurasi](#) di Referensi AWS CLI Perintah.

### Kesalahan: “Tidak dapat menentukan nama layanan/operasi yang akan diotorisasi”

GetMedia bisa gagal dengan kesalahan berikut:

```
Unable to determine service/operation name to be authorized
```

Kesalahan ini dapat terjadi jika titik akhir tidak ditentukan dengan benar. Saat Anda mendapatkan titik akhir, pastikan untuk menyertakan parameter berikut dalam `GetDataEndpoint` panggilan, tergantung pada API yang akan dipanggil:

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

## Kesalahan: “Gagal menempatkan bingkai di aliran”

`PutMedia` bisa gagal dengan kesalahan berikut:

```
Failed to put a frame in the stream
```

Kesalahan ini mungkin terjadi jika konektivitas atau izin tidak tersedia untuk layanan. Jalankan yang berikut ini di AWS CLI, dan verifikasi bahwa informasi aliran dapat diambil:

```
aws kinesismedia describe-stream --stream-name StreamName --endpoint https://ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

Jika panggilan gagal, lihat [Pemecahan Masalah AWS CLI Kesalahan](#) untuk informasi selengkapnya.

## Kesalahan: “Layanan menutup koneksi sebelum final `AckEvent` diterima”

`PutMedia` bisa gagal dengan kesalahan berikut:

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was received
```

Kesalahan ini dapat terjadi jika `PushbackInputStream` diterapkan secara tidak benar. Verifikasi bahwa `unread()` metode diterapkan dengan benar.

## Kesalahan: “`STATUS_STORE_OUT_OF_MEMORY`”

`PutMedia` bisa gagal dengan kesalahan berikut:

```
The content store is out of memory.
```

Kesalahan ini terjadi ketika toko konten tidak dialokasikan dengan ukuran yang cukup. Untuk meningkatkan ukuran toko konten, tingkatkan nilai `StorageInfo.storageSize`. Untuk informasi selengkapnya, lihat [StorageInfo](#).

## Memecahkan masalah HLS

Jika streaming video Anda tidak diputar dengan benar, lihat [the section called “Memecahkan masalah HLS”](#).

## Memecahkan masalah Java

Bagian ini menjelaskan cara memecahkan masalah umum Java yang dihadapi saat bekerja dengan Kinesis Video Streams.

Masalah

- [Mengaktifkan log Java](#)

### Mengaktifkan log Java

Untuk memecahkan masalah dengan sampel dan pustaka Java, akan sangat membantu untuk mengaktifkan dan memeriksa log debug. Untuk mengaktifkan log debug, lakukan hal berikut:

1. Tambahkan `log4j` ke `pom.xml` file, di `dependencies` simpul:

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. Di `target/classes` direktori, buat file bernama `log4j.properties` dengan konten berikut:

```
# 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
```

Log debug kemudian mencetak ke konsol IDE.

## Memecahkan masalah Pustaka Produser

Bagian ini menjelaskan masalah yang mungkin Anda temui saat menggunakan [Pustaka Produser](#).

### Masalah

- [Tidak dapat mengkompilasi SDK Produser](#)
- [Aliran video tidak muncul di konsol](#)
- [Kesalahan: “Token keamanan yang disertakan dalam permintaan tidak valid” saat streaming data menggunakan aplikasi demo GStreamer](#)
- [Kesalahan: “Gagal mengirimkan bingkai ke klien Video Kinesis”](#)
- [Aplikasi GStreamer berhenti dengan pesan “streaming berhenti, alasan tidak dinegosiasikan” di OS X](#)
- [Kesalahan: “Gagal mengalokasikan heap” saat membuat Klien Video Kinesis di demo GStreamer di Raspberry Pi](#)
- [Kesalahan: “Instruksi Ilegal” saat menjalankan demo GStreamer di Raspberry Pi](#)
- [Kamera gagal dimuat di Raspberry Pi](#)
- [Kamera tidak dapat ditemukan di macOS High Sierra](#)
- [file jni.h tidak ditemukan saat mengkompilasi di macOS High Sierra](#)
- [Kesalahan keriting saat menjalankan aplikasi demo GStreamer](#)
- [Pernyataan stempel waktu/rentang saat runtime di Raspberry Pi](#)
- [Pernyataan tentang gst\\_value\\_set\\_fraction\\_range\\_full pada Raspberry Pi](#)
- [Kesalahan STATUS\\_MKV\\_INVALID\\_ANNEXB\\_NALU\\_IN\\_FRAME\\_DATA \(0x3200000d\) di Android](#)
- [Durasi fragmen maksimum tercapai kesalahan](#)
- [Kesalahan “Nama benda tidak valid lulus” saat menggunakan otorisasi IoT](#)

## Tidak dapat mengkompilasi SDK Produser

Verifikasi bahwa pustaka yang diperlukan ada di jalur Anda. Untuk memverifikasi ini, gunakan perintah berikut:

```
$ 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
```

## Aliran video tidak muncul di konsol

Untuk menampilkan aliran video Anda di konsol, itu harus dikodekan menggunakan H.264 dalam format AVCC. Jika streaming Anda tidak ditampilkan, verifikasi hal berikut:

- Anda [Bendera Adaptasi NAL](#) disetel ke `NAL_ADAPTATION_ANNEXB_NALS` | `NAL_ADAPTATION_ANNEXB_CPD_NALS` jika aliran asli dalam format Lampiran-B. Ini adalah nilai default dalam `StreamDefinition` konstruktor.
- Anda menyediakan data pribadi codec dengan benar. Untuk H.264, ini adalah set parameter urutan (SPS) dan set parameter gambar (PPS). Bergantung pada sumber media Anda, data ini dapat diambil dari sumber media secara terpisah atau dikodekan ke dalam bingkai.

Banyak aliran dasar dalam format berikut, di mana Ab kode awal Lampiran-B (001 atau 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)
```

CPD (Codec Private Data), jika H.264 berada dalam aliran sebagai SPS dan PPS, dapat disesuaikan dengan format AVCC. Kecuali pipa media memberikan CPD secara terpisah, aplikasi dapat mengekstrak CPD dari frame dengan mencari frame Id pertama (yang harus berisi SPS dan PPS), mengekstrak dua NALU (yang akan menjadi Ab (Sps) Ab (Pps)) dan mengaturnya di CPD di `StreamDefinition`

## Kesalahan: “Token keamanan yang disertakan dalam permintaan tidak valid” saat streaming data menggunakan aplikasi demo GStreamer

Jika kesalahan ini terjadi, ada masalah dengan kredensial Anda. Verifikasi hal berikut:

- Jika Anda menggunakan kredensial sementara, Anda harus menentukan token sesi.

- Verifikasi bahwa kredensial sementara Anda tidak kedaluwarsa.
- Verifikasi bahwa Anda memiliki hak yang tepat.
- Di macOS, verifikasi bahwa Anda tidak memiliki kredensial yang di-cache di Keychain.

## Kesalahan: “Gagal mengirimkan bingkai ke klien Video Kinesis”

Jika kesalahan ini terjadi, stempel waktu tidak diatur dengan benar di aliran sumber. Coba yang berikut ini:

- Gunakan contoh SDK terbaru, yang mungkin memiliki pembaruan yang memperbaiki masalah Anda.
- Atur aliran berkualitas tinggi ke bitrate yang lebih tinggi, dan perbaiki jitter apa pun di aliran sumber jika kamera mendukung melakukannya.

## Aplikasi GStreamer berhenti dengan pesan “streaming berhenti, alasan tidak dinegosiasikan” di OS X

Streaming dapat berhenti di OS X dengan pesan berikut:

```
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)
```

Solusi yang mungkin untuk ini adalah menghapus parameter frame rate dari panggilan `gst_caps_new_simple`: `kinesis_video_gstreamer_sample_app.cpp`

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

## Kesalahan: “Gagal mengalokasikan heap” saat membuat Klien Video Kinesis di demo GStreamer di Raspberry Pi

Contoh aplikasi GStreamer mencoba mengalokasikan 512 MB RAM, yang mungkin tidak tersedia di sistem Anda. Anda dapat mengurangi alokasi ini dengan mengurangi nilai berikut di `KinesisVideoProducer.cpp`:

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

## Kesalahan: “Instruksi ilegal” saat menjalankan demo GStreamer di Raspberry Pi

Jika Anda mengalami kesalahan berikut saat menjalankan demo GStreamer, verifikasi bahwa Anda telah mengkompilasi aplikasi untuk versi perangkat yang benar. (Misalnya, verifikasi bahwa Anda tidak mengkompilasi untuk Raspberry Pi 3 saat Anda menjalankan Raspberry Pi 2.)

```
INFO - Initializing curl.  
Illegal instruction
```

## Kamera gagal dimuat di Raspberry Pi

Untuk memeriksa apakah kamera dimuat, jalankan yang berikut ini:

```
$ ls /dev/video*
```

Jika tidak ada yang ditemukan, jalankan yang berikut:

```
$ v4l2ctl get_camera
```

Outputnya akan serupa dengan yang berikut ini:

```
supported=1 detected=1
```

Jika pengemudi tidak mendeteksi kamera, lakukan hal berikut:

1. Periksa pengaturan kamera fisik dan verifikasi apakah itu terhubung dengan benar.

2. Jalankan yang berikut ini untuk memutakhirkan firmware:

```
$ sudo rpi-update
```

3. Mulai ulang perangkat.

4. Jalankan yang berikut ini untuk memuat driver:

```
$ sudo modprobe bcm2835-v4l2
```

5. Verifikasi bahwa kamera terdeteksi:

```
$ ls /dev/video*
```

## Kamera tidak dapat ditemukan di macOS High Sierra

Di macOS High Sierra, aplikasi demo tidak dapat menemukan kamera jika lebih dari satu kamera tersedia.

## file jni.h tidak ditemukan saat mengkompilasi di macOS High Sierra

Untuk mengatasi kesalahan ini, perbarui instalasi Xcode Anda ke versi terbaru.

## Kesalahan keriting saat menjalankan aplikasi demo GStreamer

Untuk mengatasi kesalahan curl saat Anda menjalankan aplikasi demo GStreamer, salin file [sertifikat ini](#) ke. /etc/ssl/cert.pem

## Pernyataan stempel waktu/rentang saat runtime di Raspberry Pi

Jika pernyataan rentang stempel waktu terjadi saat runtime, perbarui firmware dan mulai ulang perangkat:

```
$ sudo rpi-update  
$ sudo reboot
```

## Pernyataan tentang gst\_value\_set\_fraction\_range\_full pada Raspberry Pi

Pernyataan berikut muncul jika uv4l layanan sedang berjalan:

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,  
denominator_end) < 0' failed
```

Jika ini terjadi, hentikan uv4l layanan dan restart aplikasi.

## Kesalahan STATUS\_MKV\_INVALID\_ANNEXB\_NALU\_IN\_FRAME\_DATA (0x3200000d) di Android

Kesalahan berikut muncul [Bendera Adaptasi NAL](#) jika salah untuk aliran media:

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

Jika kesalahan ini terjadi, berikan `.withNALAdaptationFlags` tanda yang benar untuk media Anda (misalnya, `NAL_ADAPTATION_ANNEXB_CPD_NALS`). Berikan bendera ini di baris berikut [Pustaka Produser Android](#):

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/AmazonKinesisVideoDemoApp.java#L169>  
[StreamConfigurationFragment](#)

## Durasi fragmen maksimum tercapai kesalahan

Kesalahan ini terjadi ketika fragmen media dalam aliran melebihi batas durasi fragmen maksimum. Lihat batas durasi fragmen maksimum di [the section called “Kuota layanan API media dan media arsip”](#) bagian.

Untuk mengatasi masalah ini, coba yang berikut ini:

- Jika Anda menggunakan kamera Webcam/USB, lakukan salah satu hal berikut:
  - Jika Anda menggunakan fragmentasi berbasis bingkai kunci, maka atur encoder untuk menyediakan bingkai kunci dalam waktu 10 detik.
  - Jika Anda tidak menggunakan fragmentasi berbasis bingkai kunci, maka saat menentukan aliran [Langkah 2: Tulis dan periksa kode](#), tetapkan batas durasi fragmen maksimum ke nilai yang kurang dari 10 detik.
  - Jika Anda menggunakan encoder perangkat lunak (seperti x264) di pipeline GStreamer, Anda dapat mengatur `key-int-max` atribut ke nilai dalam waktu 10 detik. Misalnya, atur `key-int-max` ke 60, dengan `fps` disetel ke 30, untuk mengaktifkan bingkai kunci setiap 2 detik.

- Jika Anda menggunakan kamera RPI, atur atribut keyframe-interval menjadi kurang dari 10 detik.
- Jika Anda menggunakan kamera IP (RTSP), atur ukuran GOP ke 60.

## Kesalahan “Nama benda tidak valid lulus” saat menggunakan otorisasi IoT

Untuk menghindari kesalahan ini (HTTP Error 403: Response: {"message":"Invalid thing name passed"}) saat Anda menggunakan kredensial IoT untuk otorisasi, pastikan bahwa nilai stream-name (parameter kvssink elemen yang diperlukan) identik dengan nilai. iot-thingname Untuk informasi selengkapnya, lihat [Referensi parameter elemen GStreamer](#).

## Memecahkan masalah Stream Parser Library

Bagian ini menjelaskan masalah yang mungkin Anda temui saat menggunakan [Perpustakaan Parser](#).

### Masalah

- [Tidak dapat mengakses satu frame dari aliran](#)
- [Kesalahan decoding fragmen](#)

### Tidak dapat mengakses satu frame dari aliran

Untuk mengakses satu frame dari sumber streaming di aplikasi konsumen Anda, verifikasi bahwa streaming Anda berisi data pribadi codec yang benar. Untuk informasi tentang format data dalam aliran, lihat [Model Data](#).

[Untuk mempelajari cara menggunakan data pribadi codec untuk mengakses bingkai, lihat file pengujian berikut di GitHub situs web: KinesisVideoRendererExampleTest.java](#)

### Kesalahan decoding fragmen

Jika fragmen Anda tidak dikodekan dengan benar dalam format H.264 dan level yang didukung browser, Anda mungkin melihat kesalahan berikut saat memutar streaming di konsol:

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264 content
```

Jika ini terjadi, verifikasi yang berikut:

- Resolusi frame cocok dengan resolusi yang ditentukan dalam Data Pribadi Codec.
- Profil H.264 dan level frame yang dikodekan cocok dengan profil dan level yang ditentukan dalam Data Pribadi Codec.
- Browser mendukung kombinasi profil/level. Sebagian besar browser saat ini mendukung semua kombinasi profil dan level.
- Stempel waktu akurat dan dalam urutan yang benar, dan tidak ada cap waktu duplikat yang dibuat.
- Aplikasi Anda menyalin data bingkai menggunakan format H.264.



# Riwayat Dokumen untuk Amazon Kinesis Video Streams

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir Amazon Kinesis Video Streams.

- Versi API terbaru: 2017-11-29
- Update dokumentasi terbaru: 27 Juni 2023

Perubahan	Deskripsi	Tanggal
Amazon Kinesis Video Streams Edge Agent Koneksi Ujung-ke-Cloud	Rilis fitur baru. Untuk informasi selengkapnya, lihat <a href="#">Agen Edge</a> .	27 Juni 2023
Memulai: Mengirim Data ke aliran video Kinesis	Tutorial dasar untuk mengirim data media dari kamera ke aliran video Kinesis. Untuk informasi selengkapnya, lihat <a href="#">Kirim data ke aliran video Amazon Kinesis</a> .	21 Januari 2019
Template perpustakaan untuk integrasi dengan SageMaker	Contoh aplikasi untuk Kinesis Video Streams yang digunakan SageMaker untuk mengidentifikasi kapan objek tertentu muncul dalam aliran video Amazon Kinesis. Untuk informasi selengkapnya, lihat <a href="#">SageMaker</a> .	19 November 2018
Metadata streaming	Anda dapat menggunakan an Produser SDK untuk menyematkan metadata dalam aliran video Kinesis. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan metadata</a>	28 September 2018

Perubahan	Deskripsi	Tanggal
	<a href="#">streaming dengan Kinesis Video Streams.</a>	
Pencatatan SDK Produser C++	Anda dapat mengkonfigurasi logging untuk aplikasi C++ Produser SDK. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan logging dengan C++ Produser SDK.</a>	18 Juli 2018
Streaming video HLS	Sekarang Anda dapat melihat aliran video Kinesis menggunakan HTTP Live Streaming. Untuk informasi selengkapnya, lihat <a href="#">Pemutaran Kinesis Video Streams.</a>	13 Juli 2018
Streaming dari sumber RTSP	Contoh aplikasi untuk Kinesis Video Streams yang berjalan dalam wadah Docker dan streaming video dari sumber RTSP. Untuk informasi selengkapnya, lihat <a href="#">RTSP dan Docker.</a>	20 Juni 2018
Plugin GStreamer SDK Produser C++	Menunjukkan cara membangun <a href="#">Pustaka Produser C++</a> untuk digunakan sebagai tujuan GStreamer. Untuk informasi selengkapnya, lihat <a href="#">GStreamer.</a>	Juni 15, 2018

Perubahan	Deskripsi	Tanggal
Dokumentasi referensi callback SDK produser	Dokumentasi referensi untuk callback yang digunakan oleh <a href="#">Perpustakaan Produser Aliran Video Kinesis</a> Untuk informasi selengkapnya, lihat <a href="#">Callback SDK produser</a> .	12 Juni 2018
Persyaratan sistem	Dokumentasi untuk memori dan penyimpanan persyaratan untuk perangkat produser dan SDK. Untuk informasi selengkapnya, lihat <a href="#">Persyaratan sistem Kinesis Video Streams</a> .	30 Mei 2018
Dukungan CloudTrail	Dokumentasi untuk digunakan CloudTrail untuk memantau penggunaan API. Untuk informasi selengkapnya, lihat <a href="#">Mencatat panggilan API Amazon Kinesis Video Streams dengan AWS CloudTrail</a> .	24 Mei 2018
Dokumentasi referensi struktur SDK produser	Dokumentasi referensi untuk struktur yang digunakan oleh <a href="#">Perpustakaan Produser Aliran Video Kinesis</a> . Untuk informasi selengkapnya, lihat <a href="#">Struktur SDK produser</a> dan <a href="#">Struktur aliran video Kinesis</a> .	7 Mei 2018

Perubahan	Deskripsi	Tanggal
Dokumentasi contoh perender	Dokumentasi untuk aplikasi contoh Renderer, yang menunjukkan cara memecahkan kode dan menampilkan bingkai dari aliran video Kinesis. Untuk informasi selengkapnya, lihat <a href="#">Contoh: Mengurai dan merender fragmen Kinesis Video Streams</a> .	15 Maret 2018
Dokumentasi referensi Batas SDK Produser	Informasi tentang batasan untuk operasi di <a href="#">Pustaka Produser C++</a> . Untuk informasi selengkapnya, lihat <a href="#">Batas SDK produsen</a> .	13 Maret 2018
Memantau	Informasi tentang pemantauan metrik Kinesis Video Streams dan panggilan API menggunakan Amazon dan CloudWatch AWS CloudTrail Untuk informasi selengkapnya, lihat <a href="#">Memantau Aliran Video Amazon Kinesis</a> .	5 Februari 2018
Referensi bendera adaptasi Network Abstraction Layer (NAL)	Informasi tentang pengaturan bendera adaptasi NAL saat mengonsumsi video streaming. Untuk informasi selengkapnya, lihat <a href="#">Bendera Adaptasi NAL</a> .	15 Januari 2018

Perubahan	Deskripsi	Tanggal
Dukungan Android untuk streaming video	Kinesis Video Streams sekarang mendukung streaming video dari perangkat Android. Untuk informasi selengkapnya, lihat <a href="#">Pustaka Produser Android</a> .	12 Januari 2018
Kinesis Video contoh dokumentasi	Dokumentasi untuk aplikasi contoh Kinesis Video, yang menunjukkan bagaimana menggunakan <a href="#">Perpustakaan Parser aliran video</a> dalam aplikasi. Untuk informasi selengkapnya, lihat <a href="#">KinesisVideoExample</a> .	9 Januari 2018
Dokumentasi Kinesis Video Streams dirilis	Ini adalah rilis awal Panduan Pengembang Amazon Kinesis Video Streams.	29 November 2017

# Referensi API

Bagian di bawah node ini berisi dokumentasi Referensi API. Gunakan daftar isi di panel kiri untuk membuka bagian referensi API yang berbeda.

## Tindakan

Tindakan berikut didukung oleh 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](#)

Tindakan berikut didukung oleh Amazon Kinesis Video Streams Media:

- [GetMedia](#)
- [PutMedia](#)

Tindakan berikut didukung oleh Amazon Kinesis Video Streams Media yang Diarsipkan:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

Tindakan berikut didukung oleh Amazon Kinesis Video Signaling Channels:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

Tindakan berikut didukung oleh Amazon Kinesis Video WebRTC Storage:

- [JoinStorageSession](#)

## Amazon Kinesis Video Streams

Tindakan berikut didukung oleh 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

Layanan: Amazon Kinesis Video Streams

Membuat saluran pensinyalan.

CreateSignalingChannel adalah operasi asinkron.

### Minta Sintaks

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelName": "string",
  "ChannelType": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelName

Nama untuk saluran pensinyalan yang Anda buat. Itu harus unik untuk masing-masing Akun AWS dan Wilayah AWS.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Ya

### ChannelType

Jenis saluran pensinyalan yang Anda buat. Saat ini, SINGLE\_MASTER adalah satu-satunya jenis saluran yang didukung.

Jenis: String

Nilai yang Valid: SINGLE\_MASTER | FULL\_MESH

Wajib: Tidak

### SingleMasterConfiguration

Struktur yang berisi konfigurasi untuk jenis SINGLE\_MASTER saluran.

Tipe: Objek [SingleMasterConfiguration](#)

Wajib: Tidak

### Tags

Satu set tag (pasangan nilai kunci) yang ingin Anda kaitkan dengan saluran ini.

Tipe: Array objek [Tag](#)

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "ChannelARN": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

## ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran yang dibuat.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

#### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

#### AccountChannelLimitExceededException

Anda telah mencapai batas maksimum saluran pensinyalan aktif untuk ini Akun AWS di wilayah ini.

Kode Status HTTP: 400

#### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

#### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

#### ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

#### TagsPerResourceExceededLimitException

Anda telah melampaui batas tag yang dapat Anda kaitkan dengan sumber daya. Aliran video Kinesis dapat mendukung hingga 50 tag.

Kode Status HTTP: 400

#### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## CreateStream

Layanan: Amazon Kinesis Video Streams

Membuat aliran video Kinesis baru.

Saat Anda membuat streaming baru, Kinesis Video Streams memberinya nomor versi. Saat Anda mengubah metadata streaming, Kinesis Video Streams memperbarui versi.

CreateStream adalah operasi asinkron.

Untuk informasi tentang cara kerja layanan, lihat [Cara Kerjanya](#).

Anda harus memiliki izin untuk `KinesisVideo:CreateStream` tindakan tersebut.

### Minta Sintaks

```
POST /createStream HTTP/1.1
Content-type: application/json

{
  "DataRetentionInHours": number,
  "DeviceName": "string",
  "KmsKeyId": "string",
  "MediaType": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [DataRetentionInHours](#)

Jumlah jam yang Anda inginkan untuk menyimpan data dalam aliran. Kinesis Video Streams menyimpan data dalam penyimpanan data yang terkait dengan aliran.

Nilai default adalah 0, menunjukkan bahwa aliran tidak mempertahankan data.

Ketika `DataRetentionInHours` nilainya 0, konsumen masih dapat mengkonsumsi fragmen yang tersisa di buffer host layanan, yang memiliki batas waktu retensi 5 menit dan batas memori retensi 200 MB. Fragmen dihapus dari buffer ketika salah satu batas tercapai.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

### DeviceName

Nama perangkat yang menulis ke aliran.

#### Note

Dalam implementasi saat ini, Kinesis Video Streams tidak menggunakan nama ini.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[a-zA-Z0-9_.- ]+`

Wajib: Tidak

### KmsKeyId

ID kunci AWS Key Management Service (AWS KMS) yang ingin Anda gunakan Kinesis Video Streams untuk mengenkripsi data streaming.

Jika tidak ada ID kunci yang ditentukan, default, Kinesis Video-managed key () `AWS/kinesisvideo` digunakan.

Untuk informasi lebih lanjut, lihat [DescribeKey](#).

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2048.

Pola: `.+`

Wajib: Tidak

### MediaType

Jenis media aliran. Konsumen aliran dapat menggunakan informasi ini saat memproses aliran. Untuk informasi selengkapnya tentang jenis media, lihat [Jenis Media](#). Jika Anda memilih untuk menentukan `MediaType`, lihat [Persyaratan Penamaan](#) untuk pedoman.

Contoh nilai yang valid termasuk "video/h264" dan "video/h264, audio/aac".

Parameter ini opsional; nilai default adalah `null` (atau kosong di JSON).

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Wajib: Tidak

### StreamName

Nama untuk aliran yang Anda buat.

Nama aliran adalah pengenal untuk aliran, dan harus unik untuk setiap akun dan wilayah.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Ya

### Tags

Daftar tag untuk dikaitkan dengan aliran yang ditentukan. Setiap tag adalah pasangan kunci-nilai (nilainya opsional).

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 50 item.

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.



Pola Kunci:  $^{\wedge}([\backslash p\{L}\backslash p\{Z}\backslash p\{N}\_ . : / = + \backslash - @ ] * ) \$$

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai:  $[\backslash p\{L}\backslash p\{Z}\backslash p\{N}\_ . : / = + \backslash - @ ] *$

Wajib: Tidak

## Sintaksis Respons

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

{
  "StreamARN": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola:  $arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.--]+/[0-9]+$

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccountStreamLimitExceededException

Jumlah aliran yang dibuat untuk akun terlalu tinggi.

Kode Status HTTP: 400

ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

DeviceStreamLimitExceededException

Tidak diimplementasikan.

Kode Status HTTP: 400

InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

InvalidDeviceException

Tidak diimplementasikan.

Kode Status HTTP: 400

ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

TagsPerResourceExceededLimitException

Anda telah melampaui batas tag yang dapat Anda kaitkan dengan sumber daya. Aliran video Kinesis dapat mendukung hingga 50 tag.

## Kode Status HTTP: 400

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DeleteEdgeConfiguration

Layanan: Amazon Kinesis Video Streams

API asinkron yang menghapus konfigurasi tepi aliran yang ada, serta media terkait dari Edge Agent.

Saat Anda menjalankan API ini, status sinkronisasi disetel keDELETING. Proses penghapusan dimulai, di mana pekerjaan tepi aktif dihentikan dan semua media dihapus dari perangkat tepi. Waktu untuk menghapus bervariasi, tergantung pada jumlah total media yang disimpan. Jika proses penghapusan gagal, status sinkronisasi berubah menjadi DELETE\_FAILED Anda perlu mencoba kembali penghapusan.

Ketika proses penghapusan telah selesai dengan sukses, konfigurasi tepi tidak lagi dapat diakses.

### Note

API ini tidak tersedia di wilayah AWS Afrika (Cape Town), af-selatan-1.

### Minta Sintaks

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran. Tentukan salah satu StreamName atauStreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran untuk menghapus konfigurasi tepi. Tentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

StreamEdgeConfigurationNotFoundException

Pengecualian yang dirender saat Amazon Kinesis Video Stream tidak dapat menemukan konfigurasi tepi aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DeleteSignalingChannel

Layanan: Amazon Kinesis Video Streams

Menghapus saluran pensinyalan tertentu. DeleteSignalingChannel adalah operasi asinkron. Jika Anda tidak menentukan versi saluran saat ini, versi terbaru akan dihapus.

### Minta Sintaks

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan yang ingin Anda hapus.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

#### CurrentVersion

Versi saluran pensinyalan saat ini yang ingin Anda hapus. Anda dapat memperoleh versi saat ini dengan menjalankan operasi DescribeSignalingChannel atau ListSignalingChannels API.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `[a-zA-Z0-9]+`

Diperlukan: Tidak

## Sintaksis Respons

```
HTTP/1.1 200
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:



1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

#### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

#### VersionMismatchException

Versi streaming yang Anda tentukan bukan versi terbaru. Untuk mendapatkan versi terbaru, gunakan [DescribeStreamAPI](#).

Kode Status HTTP: 400

#### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DeleteStream

Layanan: Amazon Kinesis Video Streams

Menghapus aliran video Kinesis dan data yang terkandung dalam aliran.

Metode ini menandai aliran untuk dihapus, dan membuat data dalam aliran tidak dapat diakses segera.

Untuk memastikan bahwa Anda memiliki versi streaming terbaru sebelum menghapusnya, Anda dapat menentukan versi streaming. Kinesis Video Streams menetapkan versi untuk setiap aliran. Saat Anda memperbarui streaming, Kinesis Video Streams menetapkan nomor versi baru. Untuk mendapatkan versi streaming terbaru, gunakan DescribeStream API.

Operasi ini memerlukan izin untuk tindakan `KinesisVideo:DeleteStream`.

### Minta Sintaks

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "StreamARN": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### CurrentVersion

Opsional: Versi aliran yang ingin Anda hapus.

Tentukan versi sebagai perlindungan untuk memastikan bahwa Anda menghapus aliran yang benar. Untuk mendapatkan versi streaming, gunakan DescribeStream API.

Jika tidak ditentukan, hanya yang `CreationTime` dicentang sebelum menghapus aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: [a-zA-Z0-9]+

Wajib: Tidak

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran yang ingin Anda hapus.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

Diperlukan: Ya

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

VersionMismatchException

Versi streaming yang Anda tentukan bukan versi terbaru. Untuk mendapatkan versi terbaru, gunakan [DescribeStreamAPI](#).

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeEdgeConfiguration

Layanan: Amazon Kinesis Video Streams

Menjelaskan konfigurasi tepi aliran yang disetel menggunakan `StartEdgeConfigurationUpdate` API dan status terbaru dari pekerjaan perekam dan pengunggah agen edge. Gunakan API ini untuk mendapatkan status konfigurasi untuk menentukan apakah konfigurasi sinkron dengan Edge Agent. Gunakan API ini untuk mengevaluasi kesehatan Agen Edge.

### Note

API ini tidak tersedia di wilayah AWS Afrika (Cape Town), af-selatan-1.

### Minta Sintaks

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran. Tentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang konfigurasi tepinya ingin Anda perbarui. Tentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

### Sintaksis Respons

```
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"
}

```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### CreationTime

Stempel waktu di mana konfigurasi tepi aliran pertama kali dibuat.

Tipe: Timestamp

### EdgeAgentStatus

Objek yang berisi detail status terbaru untuk perekam dan pekerjaan pengunggah agen edge. Gunakan informasi ini untuk menentukan kesehatan agen edge saat ini.

Tipe: Objek EdgeAgentStatus



## [EdgeConfig](#)

Deskripsi konfigurasi tepi aliran yang akan digunakan untuk menyinkronkan dengan komponen Edge Agent IoT Greengrass. Komponen Edge Agent akan berjalan pada pengaturan Perangkat IoT Hub di premis Anda.

Tipe: Objek [EdgeConfig](#)

## [FailedStatusDetails](#)

Deskripsi status kegagalan yang dihasilkan.

Jenis: String

## [LastUpdatedTime](#)

Stempel waktu di mana konfigurasi tepi aliran terakhir diperbarui.

Tipe: Timestamp

## [StreamARN](#)

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

## [StreamName](#)

Nama aliran dari mana konfigurasi tepi diperbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

## [SyncStatus](#)

Status terbaru dari pembaruan konfigurasi tepi.

Jenis: String

Nilai yang Valid: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

### StreamEdgeConfigurationNotFoundExcpion

Pengecualian yang dirender saat Amazon Kinesis Video Stream tidak dapat menemukan konfigurasi tepi aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeImageGenerationConfiguration

Layanan: Amazon Kinesis Video Streams

Mendapat ImageGenerationConfiguration aliran video Kinesis tertentu.

Minta Sintaks

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran video Kinesis untuk mengambil konfigurasi pembuatan gambar. Anda harus menentukan salah satu StreamName atauStreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran dari mana untuk mengambil konfigurasi pembuatan gambar. Anda harus menentukan salah satu StreamName atauStreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

## Sintaksis Respons

```
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
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### [ImageGenerationConfiguration](#)

Struktur yang berisi informasi yang diperlukan untuk pengiriman gambar Kinesis video stream (KVS). Jika struktur ini nol, konfigurasi akan dihapus dari aliran.

Tipe: Objek [ImageGenerationConfiguration](#)

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)

- [AWSSDK for Ruby V3](#)

## DescribeMappedResourceConfiguration

Layanan: Amazon Kinesis Video Streams

Mengembalikan informasi terbaru tentang aliran. `streamName` atau `streamARN` harus disediakan dalam input.

### Minta Sintaks

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [MaxResults](#)

Jumlah maksimum hasil untuk kembali dalam respons.

Tipe: Bilangan Bulat

Rentang Valid: Nilai tetap 1.

Wajib: Tidak

#### [NextToken](#)

Token yang akan diberikan dalam permintaan Anda berikutnya, untuk mendapatkan kumpulan hasil lain.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.



Pola: `[a-zA-Z0-9+/=]*`

Wajib: Tidak

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "MappedResourceConfigurationList": [
    {
      "ARN": "string",
      "Type": "string"
    }
  ],
  "NextToken": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### [MappedResourceConfigurationList](#)

Struktur yang merangkum, atau berisi, properti konfigurasi penyimpanan media.

Tipe: Array objek [MappedResourceConfigurationListItem](#)

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

### [NextToken](#)

Token yang digunakan dalam NextToken permintaan untuk mengambil set hasil berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeMediaStorageConfiguration

Layanan: Amazon Kinesis Video Streams

Mengembalikan informasi terbaru tentang saluran. Tentukan ChannelName atau ChannelARN dalam input.

### Minta Sintaks

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### ChannelName

Nama salurannya.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### MediaStorageConfiguration

Struktur yang merangkum, atau berisi, properti konfigurasi penyimpanan media.

Tipe: Objek [MediaStorageConfiguration](#)

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeNotificationConfiguration

Layanan: Amazon Kinesis Video Streams

Mendapat NotificationConfiguration aliran video Kinesis tertentu.

### Minta Sintaks

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [StreamARN](#)

Nama Sumber Daya Amazon (ARN) dari aliran video Kinesis dari tempat Anda ingin mengambil konfigurasi notifikasi. Anda harus menentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

Wajib: Tidak

#### [StreamName](#)

Nama aliran untuk mengambil konfigurasi notifikasi. Anda harus menentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### NotificationConfiguration

Struktur yang berisi informasi yang diperlukan untuk pemberitahuan. Jika strukturnya nol, konfigurasi akan dihapus dari aliran.

Tipe: Objek [NotificationConfiguration](#)

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401



## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

## InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeSignalingChannel

Layanan: Amazon Kinesis Video Streams

Mengembalikan informasi terbaru tentang saluran pensinyalan. Anda harus menentukan nama atau Nama Sumber Daya Amazon (ARN) saluran yang ingin Anda jelaskan.

### Minta Sintaks

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

ARN dari saluran pensinyalan yang ingin Anda gambarkan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### ChannelName

Nama saluran pensinyalan yang ingin Anda gambarkan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "ChannelInfo": {
    "ChannelARN": "string",
    "ChannelName": "string",
    "ChannelStatus": "string",
    "ChannelType": "string",
    "CreationTime": number,
    "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
    },
    "Version": "string"
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### ChannelInfo

Struktur yang merangkum metadata dan properti saluran pensinyalan yang ditentukan.

Tipe: Objek ChannelInfo

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## DescribeStream

Layanan: Amazon Kinesis Video Streams

Mengembalikan informasi terkini tentang aliran tertentu. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

### Minta Sintaks

```
POST /describeStream HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### StreamName

Nama sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.- ]+

Diperlukan: Tidak

## Sintaksis Respons

```
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"
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### StreamInfo

Objek yang menggambarkan aliran.

Tipe: Objek [StreamInfo](#)

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

## InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## GetDataEndpoint

Layanan: Amazon Kinesis Video Streams

Mendapat titik akhir untuk aliran tertentu untuk membaca atau menulis. Gunakan titik akhir ini dalam aplikasi Anda untuk membaca dari aliran yang ditentukan (menggunakan `GetMedia` atau `GetMediaForFragmentList` operasi) atau menulis ke sana (menggunakan `PutMedia` operasi).

### Note

Titik akhir yang dikembalikan tidak memiliki nama API yang ditambahkan. Klien perlu menambahkan nama API ke titik akhir yang dikembalikan.

Dalam permintaan, tentukan aliran baik oleh `StreamName` atau `StreamARN`.

### Minta Sintaks

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
  "APIName": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### APIName

Nama tindakan API untuk mendapatkan titik akhir.

Jenis: String



Nilai yang Valid: 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

Wajib: Ya

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran yang ingin Anda dapatkan titik akhir. Anda harus menentukan parameter ini atau StreamName dalam permintaan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

Wajib: Tidak

### StreamName

Nama aliran yang ingin Anda dapatkan titik akhir. Anda harus menentukan parameter ini atau StreamARN dalam permintaan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

### Sintaksis Respons

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

{
  "DataEndpoint": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### DataEndpoint

Nilai titik akhir. Untuk membaca data dari aliran atau menulis data ke sana, tentukan titik akhir ini di aplikasi Anda.

Jenis: String

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## GetSignalingChannelEndpoint

Layanan: Amazon Kinesis Video Streams

Menyediakan titik akhir untuk saluran pensinyalan yang ditentukan untuk mengirim dan menerima pesan. API ini menggunakan parameter `SingleMasterChannelEndpointConfiguration` input, yang terdiri dari `Protocols` dan `Role` properti.

`Protocols` digunakan untuk menentukan mekanisme komunikasi. Misalnya, jika Anda menetapkan WSS sebagai protokol, API ini menghasilkan titik akhir websocket yang aman. Jika Anda menentukan HTTPS sebagai protokol, API ini menghasilkan titik akhir HTTPS.

`Role` menentukan izin pengiriman pesan. `MASTER` Peran menghasilkan API ini menghasilkan titik akhir yang dapat digunakan klien untuk berkomunikasi dengan pemirsa mana pun di saluran. `VIEWER` Peran menghasilkan API ini menghasilkan titik akhir yang dapat digunakan klien untuk berkomunikasi hanya dengan `MASTER` file.

### Minta Sintaks

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "SingleMasterChannelEndpointConfiguration": {
    "Protocols": [ "string" ],
    "Role": "string"
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan yang ingin Anda dapatkan titik akhir.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

### [SingleMasterChannelEndpointConfiguration](#)

Struktur yang berisi konfigurasi titik akhir untuk jenis SINGLE\_MASTER saluran.

Tipe: Objek [SingleMasterChannelEndpointConfiguration](#)

Wajib: Tidak

### Sintaksis Respons

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

{
  "ResourceEndpointList": [
    {
      "Protocol": "string",
      "ResourceEndpoint": "string"
    }
  ]
}
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### [ResourceEndpointList](#)

Daftar titik akhir untuk saluran pensinyalan yang ditentukan.

Tipe: Array objek [ResourceEndpointListItem](#)

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## ListEdgeAgentConfigurations

Layanan: Amazon Kinesis Video Streams

Mengembalikan array konfigurasi tepi terkait dengan Edge Agent tertentu.

Dalam permintaan, Anda harus menentukan Agen EdgeHubDeviceArn.

### Note

API ini tidak tersedia di wilayah AWS Afrika (Cape Town), af-selatan-1.

### Minta Sintaks

```
POST /listEdgeAgentConfigurations HTTP/1.1
```

```
Content-type: application/json
```

```
{  
  "HubDeviceArn": "string",  
  "MaxResults": number,  
  "NextToken": "string"  
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### HubDeviceArn

“Internet of Things (IoT) Thing” Arn dari agen tepi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Diperlukan: Ya



## MaxResults

Jumlah maksimum konfigurasi tepi untuk kembali dalam respons. Default-nya adalah 5.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 10.

Wajib: Tidak

## NextToken

Jika Anda menentukan parameter ini, ketika hasil `ListEdgeAgentConfigurations` operasi terpotong, panggilan mengembalikan respon. `NextToken` Untuk mendapatkan kumpulan konfigurasi edge lainnya, berikan token ini di permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

Diperlukan: Tidak

## Sintaksis Respons

```
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"
}

```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### EdgeConfigs

Deskripsi konfigurasi tepi aliran tunggal.

Tipe: Array objek [ListEdgeAgentConfigurationsEdgeConfig](#)

### NextToken

Jika respon terpotong, panggilan mengembalikan elemen ini dengan token yang diberikan. Untuk mendapatkan kumpulan konfigurasi edge berikutnya, gunakan token ini dalam permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)

- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## ListSignalingChannels

Layanan: Amazon Kinesis Video Streams

Mengembalikan array `ChannelInfo` objek. Setiap objek menggambarkan saluran pensinyalan. Untuk mengambil hanya saluran yang memenuhi kondisi tertentu, Anda dapat menentukan

`ChannelNameCondition`

Minta Sintaks

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
  "ChannelNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### [ChannelNameCondition](#)

Opsional: Mengembalikan hanya saluran yang memenuhi kondisi tertentu.

Tipe: Objek [ChannelNameCondition](#)

Wajib: Tidak

### [MaxResults](#)

Jumlah maksimum saluran untuk kembali dalam respons. Defaultnya adalah 500.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 10000.

Wajib: Tidak

## NextToken

Jika Anda menentukan parameter ini, ketika hasil `ListSignalingChannels` operasi terpotong, panggilan mengembalikan respon. `NextToken` Untuk mendapatkan kumpulan saluran lain, berikan token ini di permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

Diperlukan: Tidak

## Sintaksis Respons

```
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"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

## [ChannelInfoList](#)

Array objek ChannelInfo

Tipe: Array objek [ChannelInfo](#)

## [NextToken](#)

Jika respon terpotong, panggilan mengembalikan elemen ini dengan token. Untuk mendapatkan kumpulan aliran berikutnya, gunakan token ini dalam permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: [a-zA-Z0-9+/=]\*

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)



## ListStreams

Layanan: Amazon Kinesis Video Streams

Mengembalikan array `StreamInfo` objek. Setiap objek menggambarkan aliran. Untuk mengambil hanya aliran yang memenuhi kondisi tertentu, Anda dapat menentukan file. `StreamNameCondition`

### Minta Sintaks

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string",
  "StreamNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### MaxResults

Jumlah maksimum aliran untuk kembali dalam respons. Default adalah 10.000.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 10000.

Wajib: Tidak

#### NextToken

Jika Anda menentukan parameter ini, ketika hasil `ListStreams` operasi terpotong, panggilan mengembalikan respon. `NextToken` Untuk mendapatkan kumpulan aliran lain, berikan token ini di permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

Wajib: Tidak

### StreamNameCondition

Opsional: Mengembalikan hanya aliran yang memenuhi kondisi tertentu. Saat ini, Anda hanya dapat menentukan awalan nama aliran sebagai kondisi.

Tipe: Objek [StreamNameCondition](#)

Wajib: Tidak

### Sintaksis Respons

```
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"
    }
  ]
}
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### NextToken

Jika respon terpotong, panggilan mengembalikan elemen ini dengan token. Untuk mendapatkan kumpulan aliran berikutnya, gunakan token ini dalam permintaan Anda berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

### StreamInfoList

Array objek `StreamInfo`

Tipe: Array objek [StreamInfo](#)

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

#### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

#### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## ListTagsForResource

Layanan: Amazon Kinesis Video Streams

Mengembalikan daftar tag yang terkait dengan saluran pensinyalan tertentu.

### Minta Sintaks

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "ResourceARN": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [NextToken](#)

Jika Anda menentukan parameter ini dan hasil `ListTagsForResource` panggilan terpotong, respons menyertakan token yang dapat Anda gunakan dalam permintaan berikutnya untuk mengambil kumpulan tag berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

Wajib: Tidak

#### [ResourceARN](#)

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan yang ingin Anda cantumkan tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

## Sintaksis Respons

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### [NextToken](#)

Jika Anda menentukan parameter ini dan hasil `ListTagsForResource` panggilan terpotong, respons menyertakan token yang dapat Anda gunakan dalam permintaan berikutnya untuk mengambil kumpulan tag berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

### [Tags](#)

Peta kunci tag dan nilai yang terkait dengan saluran pensinyalan yang ditentukan.

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 50 item.

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^([\p{L}\p{Z}\p{N}_ . : / = + \ - @] *)$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai: `[\p{L}\p{Z}\p{N}_ . : / = + \ - @]*`

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)



## ListTagsForStream

Layanan: Amazon Kinesis Video Streams

Mengembalikan daftar tag yang terkait dengan aliran tertentu.

Dalam permintaan, Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

### Minta Sintaks

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [NextToken](#)

Jika Anda menentukan parameter ini dan hasil `ListTagsForStream` panggilan terpotong, respons menyertakan token yang dapat Anda gunakan dalam permintaan berikutnya untuk mengambil kumpulan tag berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

Wajib: Tidak

#### [StreamARN](#)

Nama Sumber Daya Amazon (ARN) dari aliran yang ingin Anda cantumkan tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang ingin Anda cantumkan tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### NextToken

Jika Anda menentukan parameter ini dan hasil `ListTags` panggilan terpotong, respons menyertakan token yang dapat Anda gunakan dalam permintaan berikutnya untuk mengambil kumpulan tag berikutnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 512.

Pola: `[a-zA-Z0-9+/=]*`

## Tags

Peta kunci tag dan nilai yang terkait dengan aliran yang ditentukan.

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 50 item.

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^( [\p{L}\p{Z}\p{N}_ . : / = + \ - @ ] * ) $`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai: `[ \p{L}\p{Z}\p{N}_ . : / = + \ - @ ] *`

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### InvalidResourceFormatException

Format tidak `StreamARN` valid.

Kode Status HTTP: 400

## NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## StartEdgeConfigurationUpdate

Layanan: Amazon Kinesis Video Streams

API asinkron yang memperbarui konfigurasi tepi aliran yang ada. Kinesis Video Stream akan menyinkronkan konfigurasi tepi aliran dengan komponen Edge Agent IoT IoT yang berjalan pada Perangkat Hub IoT, disiapkan di premis Anda. Waktu untuk menyinkronkan dapat bervariasi dan tergantung pada konektivitas Perangkat Hub. Ini SyncStatus akan diperbarui saat konfigurasi tepi diakui, dan disinkronkan dengan Agen Edge.

Jika API ini dipanggil untuk pertama kalinya, konfigurasi edge baru akan dibuat untuk streaming, dan status sinkronisasi akan disetel keSYNCING. Anda harus menunggu status sinkronisasi mencapai status terminal seperti:IN\_SYNC, atauSYNC\_FAILED, sebelum menggunakan API ini lagi. Jika Anda menjalankan API ini selama proses sinkronisasi, a ResourceInUseException akan dilemparkan. Konektivitas konfigurasi tepi aliran dan Agen Edge akan dicoba lagi selama 15 menit. Setelah 15 menit, status akan beralih ke SYNC\_FAILED negara bagian.

Untuk memindahkan konfigurasi tepi dari satu perangkat ke perangkat lainnya, gunakan [DeleteEdgeConfiguration](#) untuk menghapus konfigurasi tepi saat ini. Anda kemudian dapat memanggil StartEdgeConfigurationUpdate dengan ARN Perangkat Hub yang diperbarui.

### Note

API ini tidak tersedia di wilayah AWS Afrika (Cape Town), af-selatan-1.

### Minta Sintaks

```
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"
}

```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### EdgeConfig

Detail konfigurasi tepi diperlukan untuk menjalankan proses pembaruan.

Tipe: Objek [EdgeConfig](#)

Wajib: Ya

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran. Tentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang konfigurasi tepinya ingin Anda perbarui. Tentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

```
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"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### CreationTime

Stempel waktu di mana konfigurasi tepi aliran pertama kali dibuat.

Tipe: Timestamp

### EdgeConfig

Deskripsi konfigurasi tepi aliran yang akan digunakan untuk menyinkronkan dengan komponen Edge Agent IoT Greengrass. Komponen Edge Agent akan berjalan pada pengaturan Perangkat IoT Hub di premis Anda.

Tipe: Objek [EdgeConfig](#)

### FailedStatusDetails

Deskripsi status kegagalan yang dihasilkan.

Jenis: String

### LastUpdatedTime

Stempel waktu di mana konfigurasi tepi aliran terakhir diperbarui.



Tipe: Timestamp

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

### StreamName

Nama aliran dari mana konfigurasi tepi diperbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

### SyncStatus

Status sinkronisasi saat ini dari konfigurasi tepi aliran. Saat Anda menjalankan API ini, status sinkronisasi akan disetel ke SYNCING status. Gunakan DescribeEdgeConfiguration API untuk mendapatkan status terbaru dari konfigurasi edge.

Jenis: String

Nilai yang Valid: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

## InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## NoDataRetentionException

Retensi data Stream dalam jam sama dengan nol.

Kode Status HTTP: 400

## ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## TagResource

Layanan: Amazon Kinesis Video Streams

Menambahkan satu atau beberapa tag ke saluran pensinyalan. Tag adalah pasangan kunci-nilai (nilainya opsional) yang dapat Anda tentukan dan tetapkan ke sumber daya. AWS Jika Anda menentukan tanda yang sudah ada, nilai tanda diganti dengan nilai yang Anda tentukan dalam permintaan. Untuk informasi selengkapnya, lihat [Menggunakan Tag Alokasi Biaya](#) di AWS Billing and Cost Management dan Panduan Pengguna Manajemen Biaya.

### Minta Sintaks

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### [ResourceARN](#)

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan yang ingin Anda tambahkan tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

## Tags

Daftar tag untuk dikaitkan dengan saluran pensinyalan yang ditentukan. Setiap tanda adalah pasangan nilai kunci.

Tipe: Array objek [Tag](#)

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Wajib: Ya

## Sintaksis Respons

```
HTTP/1.1 200
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## TagsPerResourceExceededLimitException

Anda telah melampaui batas tag yang dapat Anda kaitkan dengan sumber daya. Aliran video Kinesis dapat mendukung hingga 50 tag.

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## TagStream

Layanan: Amazon Kinesis Video Streams

Menambahkan satu atau lebih tag ke aliran. Tag adalah pasangan kunci-nilai (nilainya opsional) yang dapat Anda tentukan dan tetapkan ke sumber daya. AWS Jika Anda menentukan tanda yang sudah ada, nilai tanda diganti dengan nilai yang Anda tentukan dalam permintaan. Untuk informasi selengkapnya, lihat [Menggunakan Tag Alokasi Biaya](#) di AWS Billing and Cost Management dan Panduan Pengguna Manajemen Biaya.

Anda harus memberikan salah satu `StreamName` atau `StreamARN`.

Operasi ini memerlukan izin untuk tindakan `KinesisVideo:TagStream`.

Aliran video Kinesis dapat mendukung hingga 50 tag.

### Minta Sintaks

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### StreamARN

Nama Sumber Daya Amazon (ARN) dari sumber daya yang ingin Anda tambahkan tag atau tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang ingin Anda tambahkan tag atau tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

### Tags

Daftar tag untuk dikaitkan dengan aliran yang ditentukan. Setiap tag adalah pasangan kunci-nilai (nilainya opsional).

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 50 item.

Batasan Panjang Kunci: Panjang minimum 1. Panjang maksimum 128.

Pola Kunci: `^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*$`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai: `[\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*`

Wajib: Ya

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.



## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### InvalidResourceFormatException

Format tidak `StreamARN` valid.

Kode Status HTTP: 400

### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

### TagsPerResourceExceededLimitException

Anda telah melampaui batas tag yang dapat Anda kaitkan dengan sumber daya. Aliran video Kinesis dapat mendukung hingga 50 tag.

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UntagResource

Layanan: Amazon Kinesis Video Streams

Menghapus satu atau beberapa tag dari saluran pensinyalan. Dalam permintaan, tentukan hanya kunci tag atau kunci; jangan tentukan nilainya. Jika Anda menentukan kunci tag yang tidak ada, itu diabaikan.

### Minta Sintaks

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "TagKeyList": [ "string" ]
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### [ResourceARN](#)

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan tempat Anda ingin menghapus tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

#### [TagKeyList](#)

Daftar kunci tag yang ingin Anda hapus.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^([\backslash\{L\}\backslash\{Z\}\backslash\{N\}_\cdot:/=+\backslash\text{-@}]^*)\$$

Diperlukan: Ya

## Sintaksis Respons

```
HTTP/1.1 200
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UntagStream

Layanan: Amazon Kinesis Video Streams

Menghapus satu atau beberapa tag dari aliran. Dalam permintaan, tentukan hanya kunci tag atau kunci; jangan tentukan nilainya. Jika Anda menentukan kunci tag yang tidak ada, itu diabaikan.

Dalam permintaan tersebut, Anda harus memberikan `StreamName` atau `StreamARN`.

### Minta Sintaks

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "TagKeyList": [ "string" ]
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran yang ingin Anda hapus tag.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### StreamName

Nama aliran yang ingin Anda hapus tag dari.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

### [TagKeyList](#)

Daftar kunci tag yang ingin Anda hapus.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$`

Diperlukan: Ya

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

InvalidResourceFormatException

Format tidak StreamARN valid.

Kode Status HTTP: 400

NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)



## UpdateDataRetention

Layanan: Amazon Kinesis Video Streams

Meningkatkan atau mengurangi periode retensi data stream dengan nilai yang Anda tentukan. Untuk menunjukkan apakah Anda ingin menambah atau mengurangi periode retensi data, tentukan `Operation` parameter di badan permintaan. Dalam permintaan, Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Operasi ini memerlukan izin untuk tindakan `KinesisVideo:UpdateDataRetention`.

Mengubah periode retensi data mempengaruhi data dalam aliran sebagai berikut:

- Jika periode retensi data meningkat, data yang ada disimpan untuk periode retensi baru. Misalnya, jika periode retensi data ditingkatkan dari satu jam menjadi tujuh jam, semua data yang ada disimpan selama tujuh jam.
- Jika periode retensi data menurun, data yang ada disimpan untuk periode retensi baru. Misalnya, jika periode retensi data berkurang dari tujuh jam menjadi satu jam, semua data yang ada disimpan selama satu jam, dan data yang lebih lama dari satu jam segera dihapus.

### Minta Sintaks

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DataRetentionChangeInHours": number,
  "Operation": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

## CurrentVersion

Versi aliran yang periode retensinya ingin Anda ubah. Untuk mendapatkan versinya, hubungi API `DescribeStream` atau `ListStreams` API.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `[a-zA-Z0-9]+`

Diperlukan: Ya

## DataRetentionChangeInHours

Jumlah jam untuk menyesuaikan retensi saat ini oleh. Nilai yang Anda tentukan ditambahkan atau dikurangi dari nilai saat ini, tergantung pada nilai `operation`

Nilai minimum untuk retensi data adalah 0 dan nilai maksimum adalah 87600 (sepuluh tahun).

Tipe: Integer

Rentang yang Valid: Nilai minimum 1.

Wajib: Ya

## Operation

Menunjukkan apakah Anda ingin menambah atau mengurangi periode retensi.

Jenis: String

Nilai yang Valid: `INCREASE_DATA_RETENTION` | `DECREASE_DATA_RETENTION`

Wajib: Ya

## StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran yang periode retensinya ingin Anda ubah.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang periode retensinya ingin Anda ubah.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

## ResourceInUseException

Ketika input StreamARN atau ChannelARN in sudah CLOUD\_STORAGE\_MODE dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika StreamARN input ChannelARN yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. [DescribeMediaStorageConfigurationAPI](#) untuk menentukan saluran aliran yang diberikan dipetakan.
2. [DescribeMappedResourceConfigurationAPI](#) untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. [DescribeSignalingChannelAPI](#) [DescribeStream](#) atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## VersionMismatchException

Versi streaming yang Anda tentukan bukan versi terbaru. Untuk mendapatkan versi terbaru, gunakan [DescribeStreamAPI](#).

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)

- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UpdateImageGenerationConfiguration

Layanan: Amazon Kinesis Video Streams

Memperbarui StreamInfo dan ImageProcessingConfiguration bidang.

### Minta Sintaks

```
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"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ImageGenerationConfiguration

Struktur yang berisi informasi yang diperlukan untuk pengiriman gambar KVS. Jika strukturnya nol, konfigurasi akan dihapus dari aliran.

Tipe: Objek [ImageGenerationConfiguration](#)

Wajib: Tidak

### [StreamARN](#)

Nama Sumber Daya Amazon (ARN) dari aliran video Kinesis dari mana Anda ingin memperbarui konfigurasi pembuatan gambar. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### [StreamName](#)

Nama aliran untuk memperbarui konfigurasi pembuatan gambar. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

## AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

## InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## NoDataRetentionException

Retensi data Stream dalam jam sama dengan nol.

Kode Status HTTP: 400

## ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404



## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UpdateMediaStorageConfiguration

Layanan: Amazon Kinesis Video Streams

Mengaitkan a `SignalingChannel` ke aliran untuk menyimpan media. Ada dua mode pensinyalan yang dapat Anda tentukan:

- Jika `StorageStatus` diaktifkan, data akan disimpan dalam yang `StreamARN` disediakan. Agar `WebRTC Ingestion` berfungsi, aliran harus mengaktifkan retensi data.
- Jika `StorageStatus` dinonaktifkan, tidak ada data yang akan disimpan, dan `StreamARN` parameter tidak akan diperlukan.

### Important

Jika `StorageStatus` diaktifkan, koneksi langsung peer-to-peer (master-viewer) tidak lagi terjadi. Peer terhubung langsung ke sesi penyimpanan. Anda harus memanggil `JoinStorageSession` API untuk memicu pengiriman penawaran SDP dan membuat koneksi antara peer dan sesi penyimpanan.

### Minta Sintaks

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

## ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

## MediaStorageConfiguration

Struktur yang merangkum, atau berisi, properti konfigurasi penyimpanan media.

Tipe: Objek [MediaStorageConfiguration](#)

Wajib: Ya

## Sintaksis Respons

```
HTTP/1.1 200
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

## AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

NoDataRetentionException

Retensi data Stream dalam jam sama dengan nol.

Kode Status HTTP: 400

ResourceInUseException

Ketika input StreamARN atau ChannelARN in sudah CLOUD\_STORAGE\_MODE dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika StreamARN input ChannelARN yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. DescribeMediaStorageConfigurationAPI untuk menentukan saluran aliran yang diberikan dipetakan.
2. DescribeMappedResourceConfigurationAPI untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. DescribeSignalingChannelAPI DescribeStream atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)

- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UpdateNotificationConfiguration

Layanan: Amazon Kinesis Video Streams

Memperbarui informasi notifikasi untuk aliran.

Minta Sintaks

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### NotificationConfiguration

Struktur yang berisi informasi yang diperlukan untuk pemberitahuan. Jika strukturnya nol, konfigurasi akan dihapus dari aliran.

Tipe: Objek [NotificationConfiguration](#)

Wajib: Tidak

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran video Kinesis dari tempat Anda ingin memperbarui konfigurasi notifikasi. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran untuk memperbarui konfigurasi notifikasi. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

## InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

## NoDataRetentionException

Retensi data Stream dalam jam sama dengan nol.

Kode Status HTTP: 400

## ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)



- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UpdateSignalingChannel

Layanan: Amazon Kinesis Video Streams

Memperbarui saluran pensinyalan yang ada. Ini adalah operasi asinkron dan membutuhkan waktu untuk menyelesaikannya.

Jika `MessageTtlSeconds` nilai diperbarui (baik ditingkatkan atau dikurangi), itu hanya berlaku untuk pesan baru yang dikirim melalui saluran ini setelah diperbarui. Pesan yang ada masih kedaluwarsa sesuai `MessageTtlSeconds` nilai sebelumnya.

### Minta Sintaks

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  }
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan yang ingin Anda perbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

## CurrentVersion

Versi saluran pensinyalan saat ini yang ingin Anda perbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: [a-zA-Z0-9]+

Diperlukan: Ya

## SingleMasterConfiguration

Struktur yang berisi konfigurasi untuk SINGLE\_MASTER jenis saluran pensinyalan yang ingin Anda perbarui.

Tipe: Objek [SingleMasterConfiguration](#)

Wajib: Tidak

## Sintaksis Respons

```
HTTP/1.1 200
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

## AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 401

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

### ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

### VersionMismatchException

Versi streaming yang Anda tentukan bukan versi terbaru. Untuk mendapatkan versi terbaru, gunakan [DescribeStreamAPI](#).

Kode Status HTTP: 400

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## UpdateStream

Layanan: Amazon Kinesis Video Streams

Memperbarui metadata aliran, seperti nama perangkat dan jenis media.

Anda harus memberikan nama streaming atau Amazon Resource Name (ARN) dari stream.

Untuk memastikan bahwa Anda memiliki versi streaming terbaru sebelum memperbaruinya, Anda dapat menentukan versi streaming. Kinesis Video Streams menetapkan versi untuk setiap aliran. Saat Anda memperbarui streaming, Kinesis Video Streams menetapkan nomor versi baru. Untuk mendapatkan versi streaming terbaru, gunakan DescribeStream API.

UpdateStream adalah operasi asinkron, dan membutuhkan waktu untuk menyelesaikannya.

### Minta Sintaks

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DeviceName": "string",
  "MediaType": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### CurrentVersion

Versi aliran yang metadatanya ingin Anda perbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `[a-zA-Z0-9]+`

Diperlukan: Ya

### DeviceName

Nama perangkat yang menulis ke aliran.

#### Note

Dalam implementasi saat ini, Kinesis Video Streams tidak menggunakan nama ini.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

### MediaType

Jenis media streaming. Gunakan `MediaType` untuk menentukan jenis konten yang dikandung aliran ke konsumen aliran. Untuk informasi selengkapnya tentang jenis media, lihat [Jenis Media](#). Jika Anda memilih untuk menentukan `MediaType`, lihat [Persyaratan Penamaan](#).

Untuk memutar video di konsol, Anda harus menentukan jenis video yang benar. Misalnya, jika video dalam aliran adalah H.264, tentukan `video/h264` sebagai `MediaType`

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*`

Wajib: Tidak

### StreamARN

ARN dari aliran yang metadatanya ingin Anda perbarui.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

### StreamName

Nama aliran yang metadatanya ingin Anda perbarui.

Nama aliran adalah pengenal untuk aliran, dan harus unik untuk setiap akun dan wilayah.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400



## NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

## ResourceInUseException

Ketika input `StreamARN` atau `ChannelARN` in sudah `CLOUD_STORAGE_MODE` dipetakan ke sumber Kinesis Video Stream yang berbeda, atau jika `StreamARN` input `ChannelARN` yang disediakan atau tidak dalam status Aktif, coba salah satu dari berikut ini:

1. `DescribeMediaStorageConfigurationAPI` untuk menentukan saluran aliran yang diberikan dipetakan.
2. `DescribeMappedResourceConfigurationAPI` untuk menentukan saluran tempat aliran yang diberikan dipetakan.
3. `DescribeSignalingChannelAPI` `DescribeStream` atau untuk menentukan status sumber daya.

Kode Status HTTP: 400

## ResourceNotFoundException

Amazon Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

Kode Status HTTP: 404

## VersionMismatchException

Versi streaming yang Anda tentukan bukan versi terbaru. Untuk mendapatkan versi terbaru, gunakan [DescribeStreamAPI](#).

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)

- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## Arus Amazon Kinesis Video Streams

Tindakan berikut didukung oleh Amazon Kinesis Video Streams Media:

- [GetMedia](#)
- [PutMedia](#)

## GetMedia

Layanan: Amazon Kinesis Video Streams Media

Gunakan API ini untuk mengambil konten media dari aliran video Kinesis. Dalam permintaan, Anda mengidentifikasi nama aliran atau streaming Amazon Resource Name (ARN), dan potongan awal. Kinesis Video Streams kemudian mengembalikan aliran potongan secara berurutan berdasarkan nomor fragmen.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetMedia` permintaan ke titik akhir ini menggunakan parameter `--endpoint-url`.

Saat Anda meletakkan data media (fragmen) di aliran, Kinesis Video Streams menyimpan setiap fragmen yang masuk dan metadata terkait dalam apa yang disebut “potongan”. Untuk informasi lebih lanjut, lihat [PutMedia](#). `GetMedia` API mengembalikan aliran potongan ini mulai dari potongan yang Anda tentukan dalam permintaan.

Batasan berikut berlaku saat menggunakan `GetMedia` API:

- Klien dapat menelepon `GetMedia` hingga lima kali per detik per aliran.
- Kinesis Video Streams mengirimkan data media dengan kecepatan hingga 25 megabyte per detik (atau 200 megabit per detik) selama sesi. `GetMedia`

### Note

Kode status respons `GetMedia` HTTP akan segera dikembalikan, tetapi pembacaan payload respons HTTP akan habis waktu setelah 3 detik jika tidak ada fragmen yang tertelan yang tersedia untuk pemutaran.

### Note

Jika kesalahan muncul setelah menjalankan API media Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah AWS, tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
  "StartSelector": {
    "AfterFragmentNumber": "string",
    "ContinuationToken": "string",
    "StartSelectorType": "string",
    "StartTimestamp": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### [StartSelector](#)

Mengidentifikasi potongan awal untuk mendapatkan dari aliran yang ditentukan.

Tipe: Objek [StartSelector](#)

Wajib: Ya

### [StreamARN](#)

ARN aliran dari mana Anda ingin mendapatkan konten media. Jika Anda tidak menentukan `streamARN`, Anda harus menentukan `streamName`.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

### [StreamName](#)

Nama aliran video Kinesis dari tempat Anda ingin mendapatkan konten media. Jika Anda tidak menentukan `streamName`, Anda harus menentukan `streamARN`.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

```
HTTP/1.1 200
Content-Type: ContentType
```

```
Payload
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Respons mengembalikan header HTTP berikut.

## ContentType

Jenis konten media yang diminta.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^[a-zA-Z0-9_\.\-]+$`

Respons mengembalikan yang berikut sebagai isi HTTP.

## Payload

Pengembalian Kinesis Video Streams payload adalah urutan potongan dari aliran yang ditentukan. Untuk informasi lebih lanjut tentang potongan, lihat [PutMedia](#). Potongan yang ditampilkan Kinesis Video Streams dalam panggilan juga menyertakan tag Matroska GetMedia (MKV) tambahan berikut:

- `AWS_KINESISVIDEO_CONTINUATION_TOKEN` (string UTF-8) - Jika GetMedia panggilan Anda berakhir, Anda dapat menggunakan token kelanjutan ini dalam permintaan berikutnya untuk mendapatkan potongan berikutnya di mana permintaan terakhir dihentikan.
- `AWS_KINESISVIDEO_MILLIS_BEHIND_NOW` (string UTF-8) - Aplikasi klien dapat menggunakan nilai tag ini untuk menentukan seberapa jauh di belakang potongan yang dikembalikan dalam respons dari potongan terbaru di aliran.
- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - Nomor fragmen dikembalikan dalam potongan.
- `AWS_KINESISVIDEO_SERVER_TIMESTAMP` - Stempel waktu server dari fragmen.
- `AWS_KINESISVIDEO_PRODUCER_TIMESTAMP` - Stempel waktu produsen fragmen.

Tag berikut akan hadir jika terjadi kesalahan:

- `AWS_KINESISVIDEO_ERROR_CODE` - Deskripsi string dari kesalahan yang menyebabkan berhenti. GetMedia
- `AWS_KINESISVIDEO_ERROR_ID`: Kode bilangan bulat dari kesalahan.

Kode kesalahan adalah sebagai berikut:

- 3002 - Kesalahan menulis ke aliran
- 4000 - Fragmen yang diminta tidak ditemukan
- 4500 - Akses ditolak untuk kunci KMS stream
- 4501 - Kunci KMS Stream dinonaktifkan
- 4502 - Kesalahan validasi pada kunci KMS aliran

- 4503 - Kunci KMS yang ditentukan dalam aliran tidak tersedia
- 4504 - Penggunaan kunci KMS tidak valid yang ditentukan dalam aliran
- 4505 - Status kunci KMS tidak valid yang ditentukan dalam aliran
- 4506 - Tidak dapat menemukan kunci KMS yang ditentukan dalam aliran
- 5000 - Kesalahan internal

## Galat

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### ConnectionLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas koneksi klien yang diizinkan.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### InvalidEndpointException

Penelepon menggunakan titik akhir yang salah untuk menulis data ke aliran. Saat menerima pengecualian seperti itu, pengguna harus memanggil `GetDataEndpoint` dengan `APIName` set ke `PUT_MEDIA` dan menggunakan titik akhir dari respons untuk memanggil panggilan berikutnya `PutMedia`.

Kode Status HTTP: 400

### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

ResourceNotFoundException

Kode Status: 404, Aliran dengan nama yang diberikan tidak ada.

Kode Status HTTP: 404

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go.](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## PutMedia

Layanan: Amazon Kinesis Video Streams Media

Gunakan API ini untuk mengirim data media ke aliran video Kinesis.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `PutMedia` permintaan ke titik akhir ini menggunakan parameter `--endpoint-url`.

Dalam permintaan, Anda menggunakan header HTTP untuk memberikan informasi parameter, misalnya, nama aliran, stempel waktu, dan apakah nilai stempel waktu mutlak atau relatif terhadap saat produser mulai merekam. Anda menggunakan badan permintaan untuk mengirim data media. Kinesis Video Streams hanya mendukung format kontainer Matroska (MKV) untuk mengirim data media menggunakan API ini.

Anda memiliki opsi berikut untuk mengirim data menggunakan API ini:

- Kirim data media secara real time: Misalnya, kamera keamanan dapat mengirim bingkai secara real time saat menghasilkannya. Pendekatan ini meminimalkan latensi antara perekaman video dan data yang dikirim pada kabel. Ini disebut sebagai produsen berkelanjutan. Dalam hal ini, aplikasi konsumen dapat membaca aliran secara real time atau bila diperlukan.
- Kirim data media secara offline (dalam batch): Misalnya, kamera tubuh mungkin merekam video selama berjam-jam dan menyimpannya di perangkat. Kemudian, ketika Anda menghubungkan kamera ke port docking, kamera dapat memulai `PutMedia` sesi untuk mengirim data ke aliran video Kinesis. Dalam skenario ini, latensi tidak menjadi masalah.

Saat menggunakan API ini, perhatikan pertimbangan berikut:

- Anda dapat menentukan `streamName` atau `streamARN`, tetapi bukan keduanya.
- Untuk dapat memutar media di konsol atau melalui HLS, trek 1 dari setiap fragmen harus berisi video yang disandikan h.264, CodeCID dalam metadata fragmen harus "V\_MPEG/ISO/AVC", dan metadata fragmen harus menyertakan data pribadi codec h.264 yang diformat AVCC. Secara opsional, trek 2 dari setiap fragmen harus berisi audio yang disandikan AAC, CodeCID dalam metadata fragmen harus "A\_AAC", dan metadata fragmen harus menyertakan data pribadi codec AAC.

- Anda mungkin merasa lebih mudah untuk menggunakan satu PutMedia sesi yang berjalan lama dan mengirim sejumlah besar fragmen data media dalam muatan. Untuk setiap fragmen yang diterima, Kinesis Video Streams mengirimkan satu atau lebih ucapan terima kasih. Pertimbangan jaringan potensial dapat menyebabkan Anda tidak mendapatkan semua pengakuan ini saat dihasilkan.
- Anda dapat memilih beberapa PutMedia sesi berturut-turut, masing-masing dengan fragmen yang lebih sedikit untuk memastikan bahwa Anda mendapatkan semua pengakuan dari layanan secara real time.

#### Note

Jika Anda mengirim data ke aliran yang sama pada beberapa PutMedia sesi simultan, fragmen media akan disisipkan di aliran. Anda harus memastikan bahwa ini OK dalam skenario aplikasi Anda.

Batasan berikut berlaku saat menggunakan PutMedia API:

- Klien dapat menelepon PutMedia hingga lima kali per detik per aliran.
- Klien dapat mengirim hingga lima fragmen per detik per aliran.
- Kinesis Video Streams membaca data media dengan kecepatan hingga 12,5 MB/detik, atau 100 Mbps selama sesi. PutMedia

Perhatikan kendala berikut. Dalam kasus ini, Kinesis Video Streams mengirimkan pengakuan Kesalahan dalam respons.

- Fragmen yang memiliki kode waktu yang mencakup lebih lama dari batas maksimum yang diizinkan dan yang berisi lebih dari 50 MB data tidak diperbolehkan.
- Fragmen yang berisi lebih dari tiga trek tidak diperbolehkan. Setiap frame di setiap fragmen harus memiliki nomor trek yang sama dengan salah satu trek yang ditentukan dalam header fragmen. Selain itu, setiap fragmen harus berisi setidaknya satu frame untuk setiap trek yang ditentukan dalam header fragmen.
- Setiap fragmen harus berisi setidaknya satu frame untuk setiap trek yang ditentukan dalam metadata fragmen.

- Stempel waktu bingkai paling awal dalam sebuah fragmen harus setelah stempel waktu bingkai terbaru di fragmen sebelumnya.
- Aliran MKV yang berisi lebih dari satu segmen MKV atau berisi elemen MKV yang tidak diizinkan (seperti track\*) juga menghasilkan pengakuan Kesalahan.

Kinesis Video Streams menyimpan setiap fragmen yang masuk dan metadata terkait dalam apa yang disebut “potongan.” Metadata fragmen meliputi yang berikut:

- Header MKV disediakan di awal permintaan PutMedia
- Metadata khusus Kinesis Video Streams berikut untuk fragmen:
  - `server_timestamp`- Timestamp saat Kinesis Video Streams mulai menerima fragmen.
  - `producer_timestamp`- Timestamp, ketika produser mulai merekam fragmen. Kinesis Video Streams menggunakan tiga informasi yang diterima dalam permintaan untuk menghitung nilai ini.
    - Nilai kode waktu fragmen yang diterima di badan permintaan bersama dengan fragmen.
    - Dua header permintaan: `producerStartTimeStamp` (saat produser mulai merekam) dan `fragmentTimeCodeType` (apakah kode waktu fragmen dalam muatan adalah absolut atau relatif).

Kinesis Video Streams kemudian `producer_timestamp` menghitung untuk fragmen sebagai berikut:

Jika `fragmentTimeCodeType` relatif, maka

`producer_timestamp` `producerStartTimeStamp` += kode waktu fragmen

Jika `fragmentTimeCodeType` mutlak, maka

`producer_timestamp` = kode waktu fragmen (dikonversi ke milidetik)

- Nomor fragmen unik yang ditetapkan oleh Kinesis Video Streams.

#### Note

Saat Anda membuat GetMedia permintaan, Kinesis Video Streams mengembalikan aliran potongan ini. Klien dapat memproses metadata sesuai kebutuhan.

**Note**

Operasi ini hanya tersedia untuk AWS SDK for Java. Ini tidak didukung dalam AWS SDK untuk bahasa lain.

**Note**

Kinesis Video Streams tidak mengurai dan memvalidasi data pribadi codec selama konsumsi dan pengarsipan melalui API. PutMedia KVS mengekstrak dan memvalidasi informasi yang diperlukan dari data pribadi codec untuk kemasan fragmen MPEG-TS dan MP4 saat mengkonsumsi aliran melalui API HLS.

**Note**

Jika kesalahan muncul setelah menjalankan API media Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah AWS, tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
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*

## Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

### [FragmentTimecodeType](#)

Anda meneruskan nilai ini sebagai header `x-amzn-fragment-timecode-type` HTTP.

Menunjukkan apakah kode waktu dalam fragmen (payload, isi permintaan HTTP) adalah absolut atau relatif terhadap `producerStartTimestamp` Kinesis Video Streams menggunakan informasi ini untuk `producer_timestamp` menghitung fragmen yang diterima dalam permintaan, seperti yang dijelaskan dalam ikhtisar API.

Nilai yang Valid: ABSOLUTE | RELATIVE

Diperlukan: Ya

### [ProducerStartTimestamp](#)

Anda meneruskan nilai ini sebagai header `x-amzn-producer-start-timestamp` HTTP.

Ini adalah stempel waktu produsen di mana produsen mulai merekam media (bukan stempel waktu dari fragmen tertentu dalam permintaan).

### [StreamARN](#)

Anda meneruskan nilai ini sebagai header `x-amzn-stream-arn` HTTP.

Nama Sumber Daya Amazon (ARN) dari aliran video Kinesis tempat Anda ingin menulis konten media. Jika Anda tidak menentukan `streamARN`, Anda harus menentukan `streamName`.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

### [StreamName](#)

Anda meneruskan nilai ini sebagai header `x-amzn-stream-name` HTTP.

Nama aliran video Kinesis tempat Anda ingin menulis konten media. Jika Anda tidak menentukan `streamName`, Anda harus menentukan `streamARN`.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

## Isi Permintaan

Permintaan menerima data biner berikut.

### Payload

Konten media untuk menulis ke aliran video Kinesis. Dalam implementasi saat ini, Kinesis Video Streams hanya mendukung format wadah Matroska (MKV) dengan segmen MKV tunggal. Segmen dapat berisi satu atau lebih cluster.

#### Note

Setiap cluster MKV memetakan ke fragmen aliran video Kinesis. Durasi cluster apa pun yang Anda pilih menjadi durasi fragmen.

## Sintaksis Respons

```
HTTP/1.1 200
```

*Payload*

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.


Respons mengembalikan yang berikut sebagai isi HTTP.

### Payload

Setelah Kinesis Video Streams berhasil `PutMedia` menerima permintaan, layanan memvalidasi header permintaan. Layanan kemudian mulai membaca payload dan pertama mengirimkan respons HTTP 200.

Layanan kemudian mengembalikan aliran yang berisi serangkaian objek JSON (`Acknowledgementobjek`) yang dipisahkan oleh baris baru. Ucapan terima kasih diterima pada koneksi yang sama di mana data media dikirim. Mungkin ada banyak ucapan terima kasih atas permintaan. `PutMedia` Masing-masing `Acknowledgement` terdiri dari pasangan kunci-nilai berikut:

- `AckEventType`- Jenis acara pengakuan mewakili.
  - `Buffering`: Kinesis Video Streams telah mulai menerima fragmen. Kinesis Video Streams mengirimkan pengakuan `Buffering` pertama saat byte pertama data fragmen diterima.
  - `Diterima`: Kinesis Video Streams menerima seluruh fragmen. Jika Anda tidak mengonfigurasi aliran untuk mempertahankan data, produsen dapat menghentikan buffering fragmen setelah menerima pengakuan ini.
  - `Bertahan`: Kinesis Video Streams telah mempertahankan fragmen (misalnya, ke Amazon S3). Anda mendapatkan pengakuan ini jika Anda mengonfigurasi aliran untuk mempertahankan data. Setelah Anda menerima pengakuan ini, produsen dapat menghentikan buffering fragmen.
  - `Kesalahan`: Kinesis Video Streams mengalami kesalahan saat memproses fragmen. Anda dapat meninjau kode kesalahan dan menentukan tindakan selanjutnya.
  - `Idle`: `PutMedia` Sesi sedang berlangsung. Namun, Kinesis Video Streams saat ini tidak menerima data. Kinesis Video Streams mengirimkan pengakuan ini secara berkala hingga 30 detik setelah data terakhir diterima. Jika tidak ada data yang diterima dalam 30 detik, Kinesis Video Streams menutup permintaan.

 Note

Pengakuan ini dapat membantu produsen menentukan apakah `PutMedia` koneksi masih hidup, bahkan jika itu tidak mengirim data apa pun.

- `FragmentTimecode`- Kode waktu fragmen yang pengakuannya dikirim.


Elemen dapat hilang jika `AckEventType` `Idle`.

- `FragmentNumber`- Kinesis Video Streams-nomor fragmen yang dihasilkan untuk mana pengakuan dikirim.
- `ErrorId` dan `ErrorCode` - Jika ya `ERROR`, bidang ini menyediakan kode kesalahan yang sesuai. `AckEventType` Berikut ini adalah daftar ID kesalahan dan kode kesalahan serta pesan kesalahan yang sesuai:

- 4000 - STREAM\_READ\_ERROR - Kesalahan membaca aliran data.
- 4001 - MAX\_FRAGMENT\_SIZE\_REACHED - Ukuran fragmen lebih besar dari batas maksimum, 50 MB, diizinkan.
- 4002 - MAX\_FRAGMENT\_DURATION\_REACHED - Durasi fragmen lebih besar dari batas maksimum yang diizinkan.
- 4003 - MAX\_CONNECTION\_DURATION\_REACHED - Durasi koneksi lebih besar dari ambang batas maksimum yang diizinkan.
- 4004 - FRAGMENT\_TIMECODE\_LESSER\_THAN\_PREVIOUS - Kode waktu fragmen kurang dari kode waktu waktu sebelumnya (dalam panggilan, Anda tidak dapat mengirim fragmen yang rusak). PutMedia
- 4005 - MORE\_THAN\_ALLOWED\_TRACKS\_FOUND - Lebih dari satu trek ditemukan di MKV. (usang)
- 4006 - INVALID\_MKV\_DATA - Gagal mengurai aliran masukan sebagai format MKV yang valid.
- 4007 - INVALID\_PRODUCER\_TIMESTAMP - Stempel waktu produsen tidak valid.
- 4008 - STREAM\_NOT\_ACTIVE - Stream tidak ada lagi (dihapus).
- 4009 - FRAGMENT\_METADATA\_LIMIT\_REACHED - Batas metadata fragmen tercapai. Lihat bagian [Batas](#) pada panduan pengembang.
- 4010 - TRACK\_NUMBER\_MISMATCH - Nomor trek dalam bingkai MKV tidak cocok dengan trek di header MKV.
- 4011 - FRAMES\_MISSING\_FOR\_TRACK - Fragmen tidak berisi bingkai apa pun untuk setidaknya satu trek di header MKV.
- 4012 - INVALID\_FRAGMENT\_METADATA - Nama metadata fragmen tidak dapat dimulai dengan string. AWS\_
- 4500 - KMS\_KEY\_ACCESS\_DENIED - Akses ke kunci KMS yang ditentukan aliran ditolak.
- 4501 - KMS\_KEY\_DISABLED - Kunci KMS yang ditentukan aliran dinonaktifkan.
- 4502 - KMS\_KEY\_VALIDATION\_ERROR - Kunci KMS yang ditentukan aliran gagal validasi.
- 4503 - KMS\_KEY\_UNAVAILABLE - Kunci KMS yang ditentukan aliran tidak tersedia.
- 4504 - KMS\_KEY\_INVALID\_USAGE - Penggunaan tidak valid dari kunci KMS yang ditentukan aliran.
- 4505 - KMS\_KEY\_INVALID\_STATE - Kunci KMS yang ditentukan aliran berada dalam keadaan tidak valid.



- 4506 - KMS\_KEY\_NOT\_FOUND - Kunci KMS yang ditentukan aliran tidak ditemukan.
- 5000 - INTERNAL\_ERROR - Kesalahan layanan internal.
- 5001 - ARCHIVAL\_ERROR - Kinesis Video Streams gagal mempertahankan fragmen ke penyimpanan data.

 Note

Produsen, saat mengirim muatan untuk PutMedia permintaan yang berjalan lama, harus membaca tanggapan untuk ucapan terima kasih. Produser mungkin menerima potongan ucapan terima kasih pada saat yang sama, karena buffering pada server proxy perantara. Produser yang ingin menerima ucapan terima kasih tepat waktu dapat mengirim lebih sedikit fragmen di setiap PutMedia permintaan.

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

### ConnectionLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas koneksi klien yang diizinkan.

Kode Status HTTP: 400

### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

### InvalidEndpointException

Penelepon menggunakan titik akhir yang salah untuk menulis data ke aliran. Saat menerima pengecualian seperti itu, pengguna harus memanggil GetDataEndpoint dengan APIName

set ke `PUT_MEDIA` dan menggunakan titik akhir dari respons untuk memanggil panggilan berikutnya `PutMedia`.

Kode Status HTTP: 400

`NotAuthorizedException`

Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

`ResourceNotFoundException`

Kode Status: 404, Aliran dengan nama yang diberikan tidak ada.

Kode Status HTTP: 404

Contoh-contoh

Format Pengakuan

Format pengakuan adalah sebagai berikut:

```
{
  Acknowledgement : {
    "EventType": enum,
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
    "ErrorId" : String
  }
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

## Arus Amazon Kinesis Video Streams

Tindakan berikut didukung oleh Amazon Kinesis Video Streams Archived Media:

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

## GetClip

Layanan: Amazon Kinesis Video Streams Archived Media

Mengunduh file MP4 (klip) yang berisi media yang diarsipkan dan sesuai permintaan dari aliran video yang ditentukan selama rentang waktu yang ditentukan.

Parameter `StreamName` dan parameter bersifat opsional, tetapi Anda harus menentukan salah satu `StreamName` atau `StreamArn` saat menjalankan operasi API ini.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetClip` permintaan ke titik akhir ini menggunakan parameter `--endpoint-url`.

Aliran video Amazon Kinesis memiliki persyaratan berikut untuk menyediakan data melalui MP4:

- [Persyaratan trek pemutaran video](#).
- Retensi data harus lebih besar dari 0.
- Trek video dari setiap fragmen harus berisi data pribadi codec dalam Advanced Video Coding (AVC) untuk format H.264 dan HEVC untuk format H.265. Untuk informasi lebih lanjut, lihat [spesifikasi MPEG-4 ISO/IEC 14496-15](#). Untuk informasi tentang mengadaptasi data aliran ke format tertentu, lihat [Bendera Adaptasi NAL](#).
- [Track audio \(jika ada\) dari setiap fragmen harus berisi data pribadi codec dalam format AAC \(spesifikasi AAC ISO/IEC 13818-7\) atau format MS Wave](#).

Anda dapat memantau jumlah data keluar dengan memantau CloudWatch metrik `GetClip.OutgoingBytes` Amazon. Untuk informasi tentang penggunaan CloudWatch untuk memantau Kinesis Video Streams, [lihat Memantau Kinesis Video Streams](#). [Untuk informasi harga, lihat Harga dan Harga Amazon Kinesis Video Streams AWS](#). Biaya untuk AWS data keluar berlaku.

### Minta Sintaks

```
POST /getClip HTTP/1.1
Content-type: application/json

{
```

```
"ClipFragmentSelector": {
  "FragmentSelectorType": "string",
  "TimestampRange": {
    "EndTimeStamp": number,
    "StartTimeStamp": number
  }
},
"StreamARN": "string",
"StreamName": "string"
}
```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### ClipFragmentSelector

Rentang waktu klip yang diminta dan sumber stempel waktu.

Tipe: Objek [ClipFragmentSelector](#)

Wajib: Ya

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil klip media.

Anda harus menentukan salah satu StreamName atau StreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

### StreamName

Nama aliran untuk mengambil klip media.

Anda harus menentukan salah satu StreamName atau StreamArn.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

## Sintaksis Respons

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Respons mengembalikan header HTTP berikut.

### ContentType

Jenis konten media dalam klip yang diminta.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: ^[a-zA-Z0-9\_\.\\-]+\$

Respons mengembalikan yang berikut sebagai isi HTTP.

### Payload

File MP4 tradisional yang berisi klip media dari aliran video yang ditentukan. Output akan berisi 100 MB pertama atau 200 fragmen pertama dari stempel waktu awal yang ditentukan. Untuk informasi selengkapnya, lihat Batas [Kinesis Video Streams](#).

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

## ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

## InvalidArgumentException

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

## InvalidCodecPrivateDataException

Data pribadi codec di setidaknya salah satu trek aliran video tidak valid untuk operasi ini.

Kode Status HTTP: 400

## InvalidMediaFrameException

Satu atau beberapa frame dalam klip yang diminta tidak dapat diuraikan berdasarkan codec yang ditentukan.

Kode Status HTTP: 400

## MissingCodecPrivateDataException

Tidak ada data pribadi codec yang ditemukan di setidaknya satu trek aliran video.

Kode Status HTTP: 400

## NoDataRetentionException

GetImagesdiminta untuk aliran yang tidak menyimpan data (yaitu, memiliki `DataRetentionInHours` 0).

Kode Status HTTP: 400

## NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

## ResourceNotFoundException

GetImagesakan menimbulkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

GetHLSStreamingSessionURL dan GetDASHStreamingSessionURL membuang kesalahan ini jika sesi dengan PlaybackMode dari ON\_DEMAND atau LIVE\_REPLAY diminta untuk aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan PlaybackMode of LIVE diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404

## UnsupportedStreamMediaTypeException

Jenis media (misalnya, video h.264 atau h.265 atau audio AAC atau G.711) tidak dapat ditentukan dari ID codec trek di fragmen pertama untuk sesi pemutaran. ID codec untuk trek 1 harus V\_MPEG/ISO/AVC dan, secara opsional, ID codec untuk trek 2 seharusnya. A\_AAC

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)



## GetDASHStreamingSessionURL

Layanan: Amazon Kinesis Video Streams Archived Media

Mengambil URL MPEG Dynamic Adaptive Streaming melalui HTTP (DASH) untuk streaming. Anda kemudian dapat membuka URL di pemutar media untuk melihat konten aliran.

Parameter `StreamName` dan `StreamARN` parameter bersifat opsional, tetapi Anda harus menentukan salah satu `StreamName` atau `StreamARN` saat menjalankan operasi API ini.

Aliran video Amazon Kinesis memiliki persyaratan berikut untuk menyediakan data melalui MPEG-DASH:

- [Persyaratan trek pemutaran video](#).
- Retensi data harus lebih besar dari 0.
- Trek video dari setiap fragmen harus berisi data pribadi codec dalam Advanced Video Coding (AVC) untuk format H.264 dan HEVC untuk format H.265. Untuk informasi lebih lanjut, lihat [spesifikasi MPEG-4 ISO/IEC 14496-15](#). Untuk informasi tentang mengadaptasi data aliran ke format tertentu, lihat [Bendera Adaptasi NAL](#).
- [Track audio \(jika ada\) dari setiap fragmen harus berisi data pribadi codec dalam format AAC \(spesifikasi AAC ISO/IEC 13818-7\) atau format MS Wave](#).

Prosedur berikut menunjukkan cara menggunakan MPEG-DASH dengan Kinesis Video Streams:

1. Panggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetDASHStreamingSessionURL` permintaan ke titik akhir ini menggunakan parameter `--endpoint-url`.
2. Ambil URL MPEG-DASH menggunakan `GetDASHStreamingSessionURL`. Kinesis Video Streams membuat sesi streaming MPEG-DASH yang akan digunakan untuk mengakses konten dalam aliran menggunakan protokol MPEG-DASH. `GetDASHStreamingSessionURL` mengembalikan URL yang diautentikasi (yang menyertakan token sesi terenkripsi) untuk manifes MPEG-DASH sesi (sumber daya root yang diperlukan untuk streaming dengan MPEG-DASH).

**Note**

Jangan berbagi atau menyimpan token ini di mana entitas yang tidak berwenang dapat mengaksesnya. Token menyediakan akses ke konten aliran. Lindungi token dengan tindakan yang sama yang Anda gunakan dengan kredensial Anda. AWS

Media yang tersedia melalui manifes hanya terdiri dari aliran yang diminta, rentang waktu, dan format. Tidak ada data media lain (seperti bingkai di luar jendela yang diminta atau bitrate alternatif) yang tersedia.

3. Berikan URL (berisi token sesi terenkripsi) untuk manifes MPEG-DASH ke pemutar media yang mendukung protokol MPEG-DASH. Kinesis Video Streams membuat fragmen inisialisasi dan fragmen media tersedia melalui URL manifes. Fragmen inisialisasi berisi data pribadi codec untuk aliran, dan data lain yang diperlukan untuk mengatur dekoder dan penyaji video atau audio. Fragmen media berisi bingkai video yang dikodekan atau sampel audio yang dikodekan.
4. Pemutar media menerima URL yang diautentikasi dan meminta metadata aliran dan data media secara normal. Ketika pemutar media meminta data, ia memanggil tindakan berikut:
  - `GetDashManifest`: Mengambil manifes MPEG DASH, yang berisi metadata untuk media yang ingin Anda putar.
  - `GetMP4InitFragment`: Mengambil fragmen inisialisasi MP4. Pemutar media biasanya memuat fragmen inisialisasi sebelum memuat fragmen media apa pun. Fragmen ini berisi atom MP4 `moov` dan atom anak yang diperlukan untuk menginisialisasi dekoder pemutar media.

Fragmen inisialisasi tidak sesuai dengan fragmen dalam aliran video Kinesis. Ini hanya berisi data pribadi codec untuk aliran dan trek masing-masing, yang diperlukan pemutar media untuk memecahkan kode bingkai media.

- `GetMP4MediaFragment`: Mengambil fragmen media MP4. Fragmen-fragmen ini mengandung atom MP4 `mdat` dan atom anak mereka, yang berisi bingkai media fragmen yang dikodekan dan stempel waktunya.

**Note**

Setelah fragmen media pertama tersedia dalam sesi streaming, setiap fragmen yang tidak berisi data pribadi codec yang sama menyebabkan kesalahan dikembalikan ketika

fragmen media yang berbeda dimuat. Oleh karena itu, data pribadi codec tidak boleh berubah antar fragmen dalam suatu sesi. Ini juga berarti bahwa sesi gagal jika fragmen dalam aliran berubah dari hanya memiliki video menjadi audio dan video.

Data yang diambil dengan tindakan ini dapat ditagih. Lihat [Harga](#) untuk detailnya.

#### Note

Untuk batasan yang berlaku untuk sesi MPEG-DASH, lihat Batas Kinesis Video [Streams](#).

Anda dapat memantau jumlah data yang dikonsumsi pemutar media dengan memantau CloudWatch metrik `GetMP4MediaFragment.OutgoingBytes` Amazon. Untuk informasi tentang penggunaan CloudWatch untuk memantau Kinesis Video Streams, [lihat Memantau Kinesis Video Streams](#). [Untuk informasi harga, lihat Harga dan Harga Amazon Kinesis Video StreamsAWS](#) . Biaya untuk sesi HLS dan AWS data keluar berlaku.

Untuk informasi selengkapnya tentang HLS, lihat [Streaming Langsung HTTP](#) di [situs Pengembang Apple](#).

#### Important

Jika kesalahan muncul setelah menjalankan API media yang diarsipkan Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah ke AWS tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
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"
}
```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### [DASHFragmentSelector](#)

Rentang waktu fragmen yang diminta dan sumber stempel waktu.

Parameter ini diperlukan jika PlaybackMode ada ON\_DEMAND atau LIVE\_REPLAY. Parameter ini opsional jika PlaybackMode ada LIVE. Jika PlaybackMode yaLIVE, FragmentSelectorType dapat diatur, tetapi tidak TimestampRange harus diatur. Jika PlaybackMode ada ON\_DEMAND atau LIVE\_REPLAY, keduanya FragmentSelectorType dan TimestampRange harus diatur.

Tipe: Objek [DASHFragmentSelector](#)

Diperlukan: Tidak

### [DisplayFragmentNumber](#)

Fragmen diidentifikasi dalam file manifes berdasarkan nomor urutannya dalam sesi. Jika `DisplayFragmentNumber` diatur ke `ALWAYS`, nomor fragmen Kinesis Video Streams ditambahkan ke setiap elemen S dalam file manifes dengan nama atribut "kvs:fn". Nomor fragmen ini dapat digunakan untuk logging atau untuk digunakan dengan API lain (misalnya `GetMedia` dan `GetMediaForFragmentList`). Pemutar media MPEG-DASH khusus diperlukan untuk memanfaatkan atribut kustom ini.

Nilai default-nya adalah `NEVER`.

Jenis: String

Nilai yang Valid: `ALWAYS` | `NEVER`

Diperlukan: Tidak

### [DisplayFragmentTimestamp](#)

Sesuai spesifikasi MPEG-DASH, waktu jam dinding fragmen dalam file manifes dapat diturunkan menggunakan atribut dalam manifes itu sendiri. Namun, biasanya, pemutar media yang kompatibel dengan MPEG-DASH tidak menangani celah dengan benar di timeline media. Kinesis Video Streams menyesuaikan timeline media dalam file manifes untuk mengaktifkan pemutaran media dengan diskontinuitas. Oleh karena itu, waktu jam dinding yang berasal dari file manifes mungkin tidak akurat. Jika `DisplayFragmentTimestamp` diatur ke `ALWAYS`, stempel waktu fragmen akurat ditambahkan ke setiap elemen S dalam file manifes dengan nama atribut "kvs:ts". Pemutar media MPEG-DASH khusus diperlukan untuk memanfaatkan atribut khusus ini.

Nilai default-nya adalah `NEVER`. [DASHFragmentSelector](#)`KapanSERVER_TIMESTAMP`, stempel waktu akan menjadi stempel waktu mulai server. Demikian pula, kapan [DASHFragmentSelector](#)`PRODUCER_TIMESTAMP`, stempel waktu akan menjadi stempel waktu mulai produsen.

Jenis: String

Nilai yang Valid: `ALWAYS` | `NEVER`

Diperlukan: Tidak

## Expires

Waktu dalam hitungan detik hingga sesi yang diminta berakhir. Nilai ini bisa antara 300 (5 menit) dan 43200 (12 jam).

Ketika sesi berakhir, tidak ada panggilan baru ke `GetDashManifest`, `GetMP4InitFragment`, atau `GetMP4MediaFragment` dapat dilakukan untuk sesi itu.

Defaultnya adalah 300 (5 menit).

Jenis: Integer

Rentang Valid: Nilai minimum 300. Nilai maksimum 43200.

Diperlukan: Tidak

## MaxManifestFragmentResults

Jumlah maksimum fragmen yang dikembalikan dalam manifes MPEG-DASH.

Ketika `PlaybackMode` is `LIVE`, fragmen terbaru dikembalikan ke nilai ini. Ketika `PlaybackMode` adalah `ON_DEMAND`, fragmen tertua dikembalikan, hingga jumlah maksimum ini.

Ketika ada jumlah fragmen yang lebih tinggi yang tersedia dalam manifes MPEG-DASH langsung, pemutar video sering menyangga konten sebelum memulai pemutaran. Meningkatkan ukuran buffer meningkatkan latensi pemutaran, tetapi mengurangi kemungkinan rebuffering akan terjadi selama pemutaran. Kami merekomendasikan bahwa manifes MPEG-DASH langsung memiliki minimal 3 fragmen dan maksimum 10 fragmen.

Defaultnya adalah 5 fragmen jika `PlaybackMode` ada `LIVE` atau `LIVE_REPLAY`, dan 1.000 if `PlaybackMode` is `ON_DEMAND`.

Nilai maksimum 1.000 fragmen sesuai dengan lebih dari 16 menit video pada streaming dengan fragmen 1 detik, dan lebih dari 2 1/2 jam video pada aliran dengan fragmen 10 detik.

Tipe: Long

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 5000.


Diperlukan: Tidak

## PlaybackMode

Baik untuk mengambil data langsung, tayangan ulang langsung, atau diarsipkan, berdasarkan permintaan.

Fitur dari tiga jenis sesi meliputi:

- **LIVE**: Untuk sesi jenis ini, manifes MPEG-DASH terus diperbarui dengan fragmen terbaru saat tersedia. Kami menyarankan agar pemutar media mengambil manifes baru pada interval satu detik. Ketika jenis sesi ini diputar di pemutar media, antarmuka pengguna biasanya menampilkan pemberitahuan “langsung”, tanpa kontrol scrubber untuk memilih posisi di jendela pemutaran untuk ditampilkan.

 Note

Dalam LIVE mode, fragmen terbaru yang tersedia disertakan dalam manifes MPEG-DASH, bahkan jika ada celah antara fragmen (yaitu, jika fragmen hilang). Kesenjangan seperti ini dapat menyebabkan pemutar media berhenti atau menyebabkan lompatan dalam pemutaran. Dalam mode ini, fragmen tidak ditambahkan ke manifes MPEG-DASH jika lebih tua dari fragmen terbaru dalam daftar putar. Jika fragmen yang hilang tersedia setelah fragmen berikutnya ditambahkan ke manifes, fragmen yang lebih tua tidak ditambahkan, dan celah tidak terisi.

- **LIVE\_REPLAY**: Untuk sesi jenis ini, manifes MPEG-DASH diperbarui mirip dengan cara diperbarui untuk LIVE mode kecuali dimulai dengan memasukkan fragmen dari waktu mulai tertentu. Alih-alih fragmen ditambahkan saat dicerna, fragmen ditambahkan saat durasi fragmen berikutnya berlalu. Misalnya, jika fragmen dalam sesi berdurasi dua detik, maka fragmen baru ditambahkan ke manifes setiap dua detik. Mode ini berguna untuk dapat memulai pemutaran dari saat suatu peristiwa terdeteksi dan melanjutkan media streaming langsung yang belum tertelan pada saat pembuatan sesi. Mode ini juga berguna untuk streaming media yang diarsipkan sebelumnya tanpa dibatasi oleh batas fragmen 1.000 dalam mode. ON\_DEMAND
- **ON\_DEMAND**: Untuk sesi jenis ini, manifes MPEG-DASH berisi semua fragmen untuk sesi, hingga nomor yang ditentukan dalam `MaxManifestFragmentResults` Manifes harus diambil hanya sekali untuk setiap sesi. Ketika jenis sesi ini diputar di pemutar media, antarmuka pengguna biasanya menampilkan kontrol scrubber untuk memilih posisi di jendela pemutaran untuk ditampilkan.

Dalam semua mode pemutaran, jika `FragmentSelectorType` ada `PRODUCER_TIMESTAMP`, dan jika ada beberapa fragmen dengan stempel waktu awal yang sama, fragmen yang memiliki nomor fragmen yang lebih besar (yaitu, fragmen yang lebih baru) disertakan dalam manifes MPEG-DASH. Fragmen lainnya tidak termasuk. Fragmen yang memiliki stempel waktu yang berbeda tetapi memiliki durasi yang tumpang tindih masih termasuk dalam manifes MPEG-DASH. Ini dapat menyebabkan perilaku tak terduga di pemutar media.

Default-nya adalah LIVE.

Jenis: String

Nilai yang Valid: LIVE | LIVE\_REPLAY | ON\_DEMAND

Diperlukan: Tidak

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil URL manifes MPEG-DASH.

Anda harus menentukan salah satu StreamName atauStreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

### StreamName

Nama aliran untuk mengambil URL manifes MPEG-DASH.

Anda harus menentukan salah satu StreamName atauStreamARN.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

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

{
```



```
"DASHStreamingSessionURL": "string"  
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### DASHStreamingSessionURL

URL (berisi token sesi) yang dapat digunakan pemutar media untuk mengambil manifes MPEG-DASH.

Jenis: String

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

### InvalidArgumentException

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

### InvalidCodecPrivateDataException

Data pribadi codec di setidaknya salah satu trek aliran video tidak valid untuk operasi ini.

Kode Status HTTP: 400

### MissingCodecPrivateDataException

Tidak ada data pribadi codec yang ditemukan di setidaknya satu trek aliran video.

Kode Status HTTP: 400

## NoDataRetentionException

GetImagesdiminta untuk aliran yang tidak menyimpan data (yaitu, memiliki `DataRetentionInHours 0`).

Kode Status HTTP: 400

## NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

## ResourceNotFoundException

GetImagesakan memunculkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

GetHLSStreamingSessionURLdan GetDASHStreamingSessionURL membuang kesalahan ini jika sesi dengan `PlaybackMode` dari `ON_DEMAND` atau `LIVE_REPLAY` diminta untuk aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan `PlaybackMode` of `LIVE` diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404

## UnsupportedStreamMediaTypeException

Jenis media (misalnya, video h.264 atau h.265 atau audio AAC atau G.711) tidak dapat ditentukan dari ID codec trek di fragmen pertama untuk sesi pemutaran. ID codec untuk trek 1 harus `V_MPEG/ISO/AVC` dan, secara opsional, ID codec untuk trek 2 seharusnya `A_AAC`

Kode Status HTTP: 400

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

## GetHLSStreamingSessionURL

Layanan: Amazon Kinesis Video Streams Archived Media

Mengambil URL HTTP Live Streaming (HLS) untuk streaming. Anda kemudian dapat membuka URL di browser atau pemutar media untuk melihat konten streaming.

Parameter `StreamName` dan `StreamARN` parameter bersifat opsional, tetapi Anda harus menentukan salah satu `StreamName` atau `StreamARN` saat menjalankan operasi API ini.

Aliran video Amazon Kinesis memiliki persyaratan berikut untuk menyediakan data melalui HLS:

- [Persyaratan trek pemutaran video](#).
- Retensi data harus lebih besar dari 0.
- [Trek video dari setiap fragmen harus berisi data pribadi codec dalam Advanced Video Coding \(AVC\) untuk format H.264 atau HEVC untuk format H.265 \(spesifikasi MPEG-4 ISO/IEC 14496-15\)](#). Untuk informasi tentang mengadaptasi data aliran ke format tertentu, lihat [Bendera Adaptasi NAL](#).
- Track audio (jika ada) dari setiap fragmen harus berisi data pribadi codec dalam format AAC ([spesifikasi AAC ISO/IEC 13818-7](#)).

Kinesis Video Streams Sesi HLS berisi fragmen dalam bentuk MPEG-4 terfragmentasi (juga disebut fMP4 atau CMAF) atau bentuk MPEG-2 (juga disebut potongan TS, yang juga didukung oleh spesifikasi HLS). Untuk informasi selengkapnya tentang jenis fragmen HLS, lihat spesifikasi [HLS](#).

Prosedur berikut menunjukkan cara menggunakan HLS dengan Kinesis Video Streams:

1. Panggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetHLSStreamingSessionURL` permintaan ke titik akhir ini menggunakan parameter [--endpoint-url](#).
2. Ambil URL HLS menggunakan `GetHLSStreamingSessionURL` Kinesis Video Streams membuat sesi streaming HLS yang akan digunakan untuk mengakses konten dalam aliran menggunakan protokol HLS. `GetHLSStreamingSessionURL` mengembalikan URL yang diautentikasi (yang menyertakan token sesi terenkripsi) untuk daftar putar master HLS sesi (sumber daya root yang diperlukan untuk streaming dengan HLS).

**Note**

Jangan berbagi atau menyimpan token ini di mana entitas yang tidak berwenang dapat mengaksesnya. Token menyediakan akses ke konten aliran. Lindungi token dengan tindakan yang sama yang akan Anda gunakan dengan kredensial Anda. AWS

Media yang tersedia melalui daftar putar hanya terdiri dari aliran, rentang waktu, dan format yang diminta. Tidak ada data media lain (seperti bingkai di luar jendela yang diminta atau bitrate alternatif) yang tersedia.

3. Berikan URL (berisi token sesi terenkripsi) untuk daftar putar master HLS ke pemutar media yang mendukung protokol HLS. Kinesis Video Streams membuat daftar putar media HLS, fragmen inisialisasi, dan fragmen media tersedia melalui URL daftar putar utama. Fragmen inisialisasi berisi data pribadi codec untuk aliran, dan data lain yang diperlukan untuk mengatur dekoder dan penyaji video atau audio. Fragmen media berisi bingkai video yang dikodekan H.264 atau sampel audio yang disandikan AAC.
4. Pemutar media menerima URL yang diautentikasi dan meminta metadata aliran dan data media secara normal. Ketika pemutar media meminta data, ia memanggil tindakan berikut:
  - `GetHLSMasterPlaylist`: Mengambil daftar putar master HLS, yang berisi URL untuk `GetHLSMediaPlaylist` tindakan untuk setiap trek, dan metadata tambahan untuk pemutar media, termasuk perkiraan bitrate dan resolusi.
  - `GetHLSMediaPlaylist`: Mengambil daftar putar media HLS, yang berisi URL untuk mengakses fragmen inisialisasi MP4 dengan `GetMP4InitFragment` tindakan, dan URL untuk mengakses fragmen media MP4 dengan tindakan `GetMP4MediaFragment`. Daftar putar media HLS juga berisi metadata tentang aliran yang dibutuhkan pemain untuk memainkannya, seperti apakah itu atau `PlaybackMode LIVE ON_DEMAND`. Daftar putar media HLS biasanya statis untuk sesi dengan `aPlaybackType.ON_DEMAND`. Daftar putar media HLS terus diperbarui dengan fragmen baru untuk sesi dengan file `PlaybackType LIVE`. Ada daftar putar media HLS yang berbeda untuk trek video dan trek audio (jika ada) yang berisi URL media MP4 untuk trek tertentu.
  - `GetMP4InitFragment`: Mengambil fragmen inisialisasi MP4. Pemutar media biasanya memuat fragmen inisialisasi sebelum memuat fragmen media apa pun. Fragmen ini berisi atom MP4 "" dan moov "", dan atom anak yang diperlukan untuk menginisialisasi dekoder pemutar media.  
fytp

Fragmen inisialisasi tidak sesuai dengan fragmen dalam aliran video Kinesis. Ini hanya berisi data pribadi codec untuk aliran dan trek masing-masing, yang diperlukan pemutar media untuk memecahkan kode bingkai media.

- `GetMP4MediaFragment`: Mengambil fragmen media MP4. Fragmen-fragmen ini mengandung atom MP4 "mdat" dan "" dan atom anak mereka, yang berisi bingkai media fragmen yang dikodekan dan stempel waktunya. moof

#### Note

Untuk sesi streaming HLS, perubahan data pribadi codec (CPD) in-track didukung. Setelah fragmen media pertama tersedia dalam sesi streaming, fragmen dapat berisi perubahan CPD untuk setiap trek. Oleh karena itu, fragmen dalam sesi dapat memiliki resolusi, bit rate, atau informasi lain yang berbeda dalam CPD tanpa mengganggu pemutaran. Namun, setiap perubahan yang dibuat dalam nomor trek atau format codec trek dapat mengembalikan kesalahan ketika fragmen media yang berbeda dimuat. Misalnya, streaming akan gagal jika fragmen dalam aliran berubah dari hanya memiliki video menjadi audio dan video, atau jika trek audio AAC diubah menjadi trek audio ALAW. Untuk setiap sesi streaming, hanya 500 perubahan CPD yang diizinkan.

Data yang diambil dengan tindakan ini dapat ditagih. Untuk selengkapnya, lihat [Harga](#).

- `GettsFragment`: Mengambil fragmen MPEG TS yang berisi inisialisasi dan data media untuk semua trek dalam aliran.

#### Note

Jika `yaMPEG_TS`, API ini digunakan sebagai pengganti `GetMP4InitFragment` dan `GetMP4MediaFragment` untuk mengambil media aliran. `ContainerFormat`

Data yang diambil dengan tindakan ini dapat ditagih. Untuk informasi selengkapnya, lihat [harga Kinesis Video Streams](#).

URL sesi streaming tidak boleh dibagikan antar pemain. Layanan mungkin membatasi sesi jika beberapa pemutar media membagikannya. Untuk batas koneksi, lihat Batas [Kinesis Video Streams](#).

Anda dapat memantau jumlah data yang dikonsumsi pemutar media dengan memantau CloudWatch metrik `GetMP4MediaFragment.OutgoingBytes` Amazon. Untuk informasi tentang penggunaan CloudWatch untuk memantau Kinesis Video Streams, [lihat Memantau Kinesis Video Streams](#). [Untuk informasi harga, lihat Harga dan Harga Amazon Kinesis Video StreamsAWS](#) . Biaya untuk sesi HLS dan AWS data keluar berlaku.

Untuk informasi selengkapnya tentang HLS, lihat [Streaming Langsung HTTP](#) di [situs Pengembang Apple](#).

### Important

Jika kesalahan muncul setelah menjalankan API media yang diarsipkan Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah AWS, tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
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"
}
```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### ContainerFormat

Menentukan format mana yang harus digunakan untuk kemasan media. Menentukan format FRAGMENTED\_MP4 wadah mengemas media menjadi fragmen MP4 (fMP4 atau CMAF). Ini adalah kemasan yang direkomendasikan karena ada overhead kemasan minimal. Opsi format kontainer lainnya adalah MPEG\_TS. HLS telah mendukung potongan MPEG TS sejak dirilis dan terkadang merupakan satu-satunya kemasan yang didukung pada pemain HLS yang lebih tua. MPEG TS biasanya memiliki overhead kemasan 5-25 persen. Ini berarti MPEG TS biasanya membutuhkan bandwidth dan biaya 5-25 persen lebih banyak daripada fMP4.

Default-nya adalah FRAGMENTED\_MP4.

Jenis: String

Nilai yang Valid: FRAGMENTED\_MP4 | MPEG\_TS

Diperlukan: Tidak

### DiscontinuityMode

Menentukan kapan bendera menandai diskontinuitas antara fragmen ditambahkan ke daftar putar media.



Pemutar media biasanya membangun garis waktu konten media untuk diputar, berdasarkan stempel waktu setiap fragmen. Ini berarti bahwa jika ada tumpang tindih atau kesenjangan antara fragmen (seperti biasanya jika [HLSFragmentSelector](#) diatur ke `SERVER_TIMESTAMP`), timeline media player juga akan memiliki celah kecil antara fragmen di beberapa tempat, dan akan menimpa frame di tempat lain. Kesenjangan di timeline pemutar media dapat menyebabkan pemutaran macet dan tumpang tindih dapat menyebabkan pemutaran menjadi gelisah. Ketika ada bendera diskontinuitas antara fragmen, pemutar media diharapkan untuk mengatur ulang timeline, sehingga fragmen berikutnya diputar segera setelah fragmen sebelumnya.

Mode berikut didukung:

- **ALWAYS**: penanda diskontinuitas ditempatkan di antara setiap fragmen dalam daftar putar media HLS. Disarankan untuk menggunakan nilai **ALWAYS** jika stempel waktu fragmen tidak akurat.
- **NEVER**: tidak ada penanda diskontinuitas yang ditempatkan di mana saja. Disarankan untuk menggunakan nilai **NEVER** untuk memastikan garis waktu pemutar media paling akurat memetakan ke stempel waktu produsen.
- **ON\_DISCONTINUITY**: penanda diskontinuitas ditempatkan di antara fragmen yang memiliki celah atau tumpang tindih lebih dari 50 milidetik. Untuk sebagian besar skenario pemutaran, disarankan untuk menggunakan nilai **ON\_DISCONTINUITY** sehingga timeline pemutar media hanya diatur ulang ketika ada masalah signifikan dengan timeline media (misalnya fragmen yang hilang).

[HLSFragmentSelector](#)Defaultnya adalah **ALWAYS** kapan diatur ke `SERVER_TIMESTAMP`, dan **NEVER** ketika diatur ke `PRODUCER_TIMESTAMP`.

Jenis: String

Nilai yang Valid: **ALWAYS** | **NEVER** | **ON\_DISCONTINUITY**

Diperlukan: Tidak

### [DisplayFragmentTimestamp](#)

Menentukan kapan stempel waktu mulai fragmen harus disertakan dalam daftar putar media HLS. Biasanya, pemutar media melaporkan posisi playhead sebagai waktu relatif terhadap awal fragmen pertama dalam sesi pemutaran. Namun, ketika stempel waktu awal disertakan dalam daftar putar media HLS, beberapa pemutar media mungkin melaporkan playhead saat ini sebagai waktu absolut berdasarkan stempel waktu fragmen. Ini dapat berguna untuk menciptakan pengalaman pemutaran yang menunjukkan kepada pemirsa waktu jam dinding media.

Default-nya adalah NEVER. [HLSFragmentSelector](#)KapanSERVER\_TIMESTAMP, stempel waktu akan menjadi stempel waktu mulai server. Demikian pula, kapan [HLSFragmentSelector](#)PRODUCER\_TIMESTAMP, stempel waktu akan menjadi stempel waktu mulai produsen.

Jenis: String

Nilai yang Valid: ALWAYS | NEVER

Diperlukan: Tidak

### Expires

Waktu dalam hitungan detik hingga sesi yang diminta berakhir. Nilai ini bisa antara 300 (5 menit) dan 43200 (12 jam).

Ketika sesi berakhir, tidak ada panggilan baru

keGetHLSMasterPlaylist,GetHLSMediaPlaylist,GetMP4InitFragment,GetMP4MediaFragmen atau GetTSFragment dapat dilakukan untuk sesi itu.

Defaultnya adalah 300 (5 menit).

Jenis: Integer

Rentang Valid: Nilai minimum 300. Nilai maksimum 43200.

Diperlukan: Tidak

### HLSFragmentSelector

Rentang waktu fragmen yang diminta dan sumber stempel waktu.

Parameter ini diperlukan jika PlaybackMode ada ON\_DEMAND atauLIVE\_REPLAY. Parameter ini opsional jika PlaybackMode ada LIVE. Jika PlaybackMode yaLIVE, FragmentSelectorType dapat diatur, tetapi tidak TimestampRange harus diatur. Jika PlaybackMode ada ON\_DEMAND atauLIVE\_REPLAY, keduanya FragmentSelectorType dan TimestampRange harus diatur.

Tipe: Objek [HLSFragmentSelector](#)

Diperlukan: Tidak

### MaxMediaPlaylistFragmentResults

Jumlah maksimum fragmen yang dikembalikan dalam daftar putar media HLS.

Ketika `PlaybackMode` is `LIVE`, fragmen terbaru dikembalikan ke nilai ini. Ketika `PlaybackMode` adalah `ON_DEMAND`, fragmen tertua dikembalikan, hingga jumlah maksimum ini.

Ketika ada jumlah fragmen yang lebih tinggi yang tersedia di daftar putar media HLS langsung, pemutar video sering menyangga konten sebelum memulai pemutaran. Meningkatkan ukuran buffer meningkatkan latensi pemutaran, tetapi mengurangi kemungkinan rebuffering akan terjadi selama pemutaran. Kami merekomendasikan bahwa daftar putar media HLS langsung memiliki minimal 3 fragmen dan maksimal 10 fragmen.

Defaultnya adalah 5 fragmen jika `PlaybackMode` ada `LIVE` atau `LIVE_REPLAY`, dan 1.000 if `PlaybackMode` is `ON_DEMAND`.

Nilai maksimum 5.000 fragmen sesuai dengan lebih dari 80 menit video pada streaming dengan fragmen 1 detik, dan lebih dari 13 jam video pada aliran dengan fragmen 10 detik.

Tipe: Long

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 5000.

Diperlukan: Tidak

## PlaybackMode

Baik untuk mengambil data langsung, tayangan ulang langsung, atau diarsipkan, berdasarkan permintaan.

Fitur dari tiga jenis sesi meliputi:

- **LIVE**: Untuk sesi jenis ini, daftar putar media HLS terus diperbarui dengan fragmen terbaru saat tersedia. Kami menyarankan agar pemutar media mengambil daftar putar baru pada interval satu detik. Ketika jenis sesi ini diputar di pemutar media, antarmuka pengguna biasanya menampilkan pemberitahuan “langsung”, tanpa kontrol scrubber untuk memilih posisi di jendela pemutaran untuk ditampilkan.

### Note

Dalam `LIVE` mode, fragmen terbaru yang tersedia disertakan dalam daftar putar media HLS, bahkan jika ada celah di antara fragmen (yaitu, jika ada fragmen yang hilang). Kesenjangan seperti ini dapat menyebabkan pemutar media berhenti atau menyebabkan lompatan dalam pemutaran. Dalam mode ini, fragmen tidak ditambahkan ke daftar putar media HLS jika lebih tua dari fragmen terbaru di daftar putar. Jika

fragmen yang hilang tersedia setelah fragmen berikutnya ditambahkan ke daftar putar, fragmen yang lebih tua tidak ditambahkan, dan celah tidak terisi.

- **LIVE\_REPLAY:** Untuk sesi jenis ini, daftar putar media HLS diperbarui mirip dengan cara diperbarui untuk LIVE mode kecuali dimulai dengan memasukkan fragmen dari waktu mulai tertentu. Alih-alih fragmen ditambahkan saat dicerna, fragmen ditambahkan saat durasi fragmen berikutnya berlalu. Misalnya, jika fragmen dalam sesi berdurasi dua detik, maka fragmen baru ditambahkan ke daftar putar media setiap dua detik. Mode ini berguna untuk dapat memulai pemutaran dari saat suatu peristiwa terdeteksi dan melanjutkan media streaming langsung yang belum tertelan pada saat pembuatan sesi. Mode ini juga berguna untuk streaming media yang diarsipkan sebelumnya tanpa dibatasi oleh batas fragmen 1.000 dalam mode. ON\_DEMAND
- **ON\_DEMAND:** Untuk sesi jenis ini, daftar putar media HLS berisi semua fragmen untuk sesi, hingga nomor yang ditentukan dalam. `MaxMediaPlaylistFragmentResults` Daftar putar harus diambil hanya sekali untuk setiap sesi. Ketika jenis sesi ini diputar di pemutar media, antarmuka pengguna biasanya menampilkan kontrol scrubber untuk memilih posisi di jendela pemutaran untuk ditampilkan.

Di semua mode pemutaran, jika `FragmentSelectorType` ada `PRODUCER_TIMESTAMP`, dan jika ada beberapa fragmen dengan stempel waktu awal yang sama, fragmen yang memiliki nomor fragmen terbesar (yaitu, fragmen terbaru) termasuk dalam daftar putar media HLS. Fragmen lainnya tidak termasuk. Fragmen yang memiliki stempel waktu berbeda tetapi memiliki durasi yang tumpang tindih masih termasuk dalam daftar putar media HLS. Hal ini dapat menyebabkan perilaku tak terduga di pemutar media.

Default-nya adalah LIVE.

Jenis: String

Nilai yang Valid: LIVE | LIVE\_REPLAY | ON\_DEMAND

Diperlukan: Tidak

## StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil URL daftar putar master HLS.

Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

### StreamName

Nama aliran untuk mengambil URL daftar putar master HLS.

Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

### Sintaksis Respons

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

{
  "HLSStreamingSessionURL": "string"
}
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### HLSStreamingSessionURL

URL (berisi token sesi) yang dapat digunakan pemutar media untuk mengambil daftar putar master HLS.

Jenis: String

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

### InvalidArgumentException

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

### InvalidCodecPrivateDataException

Data pribadi codec di setidaknya salah satu trek aliran video tidak valid untuk operasi ini.

Kode Status HTTP: 400

### MissingCodecPrivateDataException

Tidak ada data pribadi codec yang ditemukan di setidaknya satu trek aliran video.

Kode Status HTTP: 400

### NoDataRetentionException

GetImagesdiminta untuk aliran yang tidak menyimpan data (yaitu, memiliki `DataRetentionInHours` 0).

Kode Status HTTP: 400

### NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

### ResourceNotFoundException

GetImagesakan memunculkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

GetHLSStreamingSessionURL dan GetDASHStreamingSessionURL membuang kesalahan ini jika sesi dengan PlaybackMode dari ON\_DEMAND atau LIVE\_REPLAY diminta untuk aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan PlaybackMode of LIVE diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404

#### UnsupportedStreamMediaTypeException

Jenis media (misalnya, video h.264 atau h.265 atau audio AAC atau G.711) tidak dapat ditentukan dari ID codec trek di fragmen pertama untuk sesi pemutaran. ID codec untuk trek 1 harus V\_MPEG/ISO/AVC dan, secara opsional, ID codec untuk trek 2 seharusnya A\_AAC

Kode Status HTTP: 400

#### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

## GetImages

Layanan: Amazon Kinesis Video Streams Archived Media

Mengambil daftar gambar yang sesuai dengan setiap stempel waktu untuk rentang waktu tertentu, interval pengambilan sampel, dan konfigurasi format gambar.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetImages` permintaan ke titik akhir ini menggunakan parameter `--endpoint-url`.

### Persyaratan trek pemutaran video.

#### Minta Sintaks

```
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
}
```

#### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.



## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### EndTimeStamp

Stempel waktu akhir untuk rentang gambar yang akan dihasilkan. Jika rentang waktu antara `StartTimeStamp` dan `EndTimeStamp` lebih dari 300 detik di atas `StartTimeStamp`, Anda akan menerima `IllegalArgumentException`.

Tipe: Timestamp

Diperlukan: Ya

### Format

Format yang akan digunakan untuk menyandikan gambar.

Jenis: String

Nilai yang Valid: JPEG | PNG

Diperlukan: Ya

### FormatConfig

Daftar struktur pasangan kunci-nilai yang berisi parameter tambahan yang dapat diterapkan saat gambar dihasilkan. `FormatConfig` kuncinya adalah `JPEGQuality`, yang menunjukkan kunci kualitas JPEG yang akan digunakan untuk menghasilkan gambar. `FormatConfig` Nilai menerima int dari 1 hingga 100. Jika nilainya 1, gambar akan dihasilkan dengan kualitas kurang dan kompresi terbaik. Jika nilainya 100, gambar akan dihasilkan dengan kualitas terbaik dan kompresi yang lebih sedikit. Jika tidak ada nilai yang diberikan, nilai default `JPEGQuality` kunci akan diatur ke 80.

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 1 item.

Kunci yang valid: `JPEGQuality`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai: `^[a-zA-Z_0-9]+`

Diperlukan: Tidak

### HeightPixels

Ketinggian gambar output yang digunakan bersamaan dengan WidthPixels parameter. Ketika keduanya HeightPixels dan WidthPixels parameter disediakan, gambar akan diregangkan agar sesuai dengan rasio aspek yang ditentukan. Jika hanya HeightPixels parameter yang disediakan, rasio aspek aslinya akan digunakan untuk menghitung WidthPixels rasio. Jika tidak ada parameter yang disediakan, ukuran gambar asli akan dikembalikan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 2160.

Diperlukan: Tidak

### ImageSelectorType

Asal usul stempel waktu Server atau Produser yang digunakan untuk menghasilkan gambar.

Jenis: String

Nilai yang Valid: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

Diperlukan: Ya

### MaxResults

Jumlah maksimum gambar yang akan dikembalikan oleh API.

#### Note

Batas defaultnya adalah 25 gambar per respons API. Memberikan nilai yang MaxResults lebih besar dari nilai ini akan menghasilkan ukuran halaman 25. Setiap hasil tambahan akan diberi paginasi.

Tipe: Long

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 100.

Diperlukan: Tidak

## NextToken

Token yang menentukan di mana harus memulai paginasi set Gambar berikutnya. Ini adalah `GetImages:NextToken` dari respons yang sebelumnya terpotong.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 4096.

Pola: `[a-zA-Z0-9+/\]{0,2}`

Diperlukan: Tidak

## SamplingInterval

Interval waktu dalam milidetik (ms) di mana gambar perlu dihasilkan dari aliran. Nilai minimum yang dapat diberikan adalah 200 ms (5 gambar per detik). Jika rentang stempel waktu kurang dari interval pengambilan sampel, gambar dari `startTimestamp` akan dikembalikan jika tersedia.

Jenis: Integer

Diperlukan: Ya

## StartTimestamp

Titik awal dari mana gambar harus dihasilkan. Ini `StartTimestamp` harus berada dalam rentang waktu yang inklusif agar gambar dikembalikan.

Tipe: Timestamp

Diperlukan: Ya

## StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil gambar. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

## StreamName

Nama aliran dari mana untuk mengambil gambar. Anda harus menentukan salah satu `StreamName` atau `StreamARN`.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## WidthPixels

Lebar gambar output yang digunakan bersamaan dengan `HeightPixels` parameter. Ketika keduanya `WidthPixels` dan `HeightPixels` parameter disediakan, gambar akan diregangkan agar sesuai dengan rasio aspek yang ditentukan. Jika hanya `WidthPixels` parameter yang disediakan atau jika hanya `HeightPixels` disediakan, a `ValidationException` akan dilemparkan. Jika tidak ada parameter yang disediakan, ukuran gambar asli dari aliran akan dikembalikan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 3840.

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "Images": [
    {
      "Error": "string",
      "ImageContent": "string",
      "TimeStamp": number
    }
  ],
  "NextToken": "string"
```

```
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### Images

Daftar gambar yang dihasilkan dari aliran video. Jika tidak ada media yang tersedia untuk stempel waktu yang diberikan, NO\_MEDIA kesalahan akan tercantum dalam output. Jika terjadi kesalahan saat gambar sedang dihasilkan, MEDIA\_ERROR akan tercantum dalam output sebagai penyebab gambar yang hilang.

Tipe: Array objek [Image](#)

### NextToken

Token terenkripsi yang digunakan dalam permintaan untuk mendapatkan lebih banyak gambar.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 4096.

Pola: `[a-zA-Z0-9+/\]{0,2}`

## Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

### InvalidArgumentException

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

## NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

## ResourceNotFoundException

GetImagesakan memunculkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

GetHLSStreamingSessionURL dan GetDASHStreamingSessionURL membuang kesalahan ini jika sesi dengan PlaybackMode dari ON\_DEMAND atau LIVE\_REPLAY diminta untuk aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan PlaybackMode of LIVE diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

## GetMediaForFragmentList

Layanan: Amazon Kinesis Video Streams Archived Media

Mendapat media untuk daftar fragmen (ditentukan berdasarkan nomor fragmen) dari data yang diarsipkan dalam aliran video Amazon Kinesis.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `GetMediaForFragmentList` permintaan ke titik akhir ini menggunakan parameter [--endpoint-url](#).

Untuk batasan, lihat Batas [Kinesis Video Streams](#).

### Important

Jika kesalahan muncul setelah menjalankan API media yang diarsipkan Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah AWS, tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
POST /getMediaForFragmentList HTTP/1.1
Content-type: application/json
```

```
{
  "Fragments": [ "string" ],
  "StreamARN": "string",
  "StreamName": "string"
}
```

## Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

## Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### Fragments

Daftar jumlah fragmen untuk mengambil media. Anda mengambil nilai-nilai ini dengan [ListFragments](#).

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 1000 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^{\wedge}[0-9]^{\$}$

Diperlukan: Ya

### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil media fragmen. Tentukan parameter ini atau `StreamName` parameter.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola:  $arn:[a-z\d-]^{\+}:kinesisvideo:[a-z0-9-]^{\+}:[0-9]^{\+}:[a-z]^{\+}/[a-zA-Z0-9_.-]^{\+}/[0-9]^{\+}$

Diperlukan: Tidak



## StreamName

Nama aliran untuk mengambil media fragmen. Tentukan parameter ini atau `StreamARN` parameter.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Respons mengembalikan header HTTP berikut.

## ContentType

Jenis konten media yang diminta.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `^[a-zA-Z0-9_\.\\-]+$`

Respons mengembalikan yang berikut sebagai isi HTTP.

## Payload

Payload yang Kinesis Video Streams kembalikan adalah urutan potongan dari aliran yang ditentukan. Untuk informasi tentang potongan, lihat. [PutMedia](#) Potongan yang ditampilkan Kinesis Video Streams dalam panggilan juga menyertakan tag Matroska `GetMediaForFragmentList` (MKV) tambahan berikut:

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - Nomor fragmen dikembalikan dalam potongan.
- `AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP` - Stempel waktu sisi server dari fragmen.
- `AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP` - Stempel waktu sisi produsen dari fragmen.

Tag berikut akan disertakan jika pengecualian terjadi:

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - Jumlah fragmen yang melemparkan pengecualian.
- `AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE` - Kode integer kesalahan.
- `AWS_KINESISVIDEO_EXCEPTION_MESSAGE` - Deskripsi teks pengecualian.

## Galat

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

### InvalidArgumentException

Parameter tertentu melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

### NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

### ResourceNotFoundException

`GetImages` akan menimbulkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

`GetHLSStreamingSessionURL` dan `GetDASHStreamingSessionURL` membuang kesalahan ini jika sesi dengan `PlaybackMode` dari `ON_DEMAND` atau `LIVE_REPLAY` diminta untuk

aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan PlaybackMode of LIVE diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go.](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListFragments

Layanan: Amazon Kinesis Video Streams Archived Media

Mengembalikan daftar [Fragment](#) objek dari rentang aliran dan timestamp tertentu dalam data yang diarsipkan.

Daftar fragmen pada akhirnya konsisten. Ini berarti bahwa bahkan jika produsen menerima pengakuan bahwa fragmen dipertahankan, hasilnya mungkin tidak segera dikembalikan dari permintaan ke `ListFragments`. Namun, hasil biasanya tersedia dalam waktu kurang dari satu detik.

### Note

Anda harus terlebih dahulu memanggil `GetDataEndpoint` API untuk mendapatkan titik akhir. Kemudian kirim `ListFragments` permintaan ke titik akhir ini menggunakan parameter [--endpoint-url](#).

### Important

Jika kesalahan muncul setelah menjalankan API media yang diarsipkan Kinesis Video Streams, selain kode status HTTP dan badan respons, itu mencakup potongan-potongan informasi berikut:

- `x-amz-ErrorTypeHeader` HTTP - berisi jenis kesalahan yang lebih spesifik selain apa yang disediakan kode status HTTP.
- `x-amz-RequestIdHeader` HTTP - jika Anda ingin melaporkan masalah AWS, tim dukungan dapat mendiagnosis masalah dengan lebih baik jika diberi ID Permintaan.

Baik kode status HTTP dan `ErrorType` header dapat digunakan untuk membuat keputusan terprogram tentang apakah kesalahan dapat dicoba ulang dan dalam kondisi apa, serta memberikan informasi tentang tindakan apa yang mungkin perlu dilakukan oleh programmer klien agar berhasil mencoba lagi.

Untuk informasi selengkapnya, lihat bagian Kesalahan di bagian bawah topik ini, serta [Kesalahan Umum](#).

## Minta Sintaks

```
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"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

### FragmentSelector

Menjelaskan rentang stempel waktu dan asal stempel waktu untuk rentang fragmen yang akan dikembalikan.

#### Note

Ini hanya diperlukan ketika NextToken tidak diteruskan di API.

Tipe: Objek FragmentSelector

Diperlukan: Tidak

## MaxResults

Jumlah total fragmen yang akan dikembalikan. Jika jumlah total fragmen yang tersedia lebih dari nilai yang ditentukan dalam `max-results`, maka [ListFragments: NextToken](#) disediakan dalam output yang dapat Anda gunakan untuk melanjutkan pagination.

Nilai default-nya adalah 100.

Tipe: Long

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

## NextToken

Token untuk menentukan di mana harus memulai paginating. Ini adalah [ListFragments: NextToken](#) dari respons yang sebelumnya terpotong.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 4096.

Pola: `[a-zA-Z0-9+/\]{0,2}`

Diperlukan: Tidak

## StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran untuk mengambil daftar fragmen. Tentukan parameter ini atau `StreamName` parameter lainnya.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

## StreamName

Nama aliran dari mana untuk mengambil daftar fragmen. Tentukan parameter ini atau `StreamARN` parameter lainnya.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "Fragments": [
    {
      "FragmentLengthInMilliseconds": number,
      "FragmentNumber": "string",
      "FragmentSizeInBytes": number,
      "ProducerTimestamp": number,
      "ServerTimestamp": number
    }
  ],
  "NextToken": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

### [Fragments](#)

Daftar [Fragment](#) objek yang diarsipkan dari aliran yang memenuhi kriteria pemilih. Hasil tidak dalam urutan tertentu, bahkan di seluruh halaman.

Jika tidak ada fragmen dalam aliran yang memenuhi kriteria pemilih, daftar kosong dikembalikan.

Tipe: Array objek [Fragment](#)

### [NextToken](#)

Jika daftar yang dikembalikan terpotong, operasi mengembalikan token ini untuk digunakan untuk mengambil halaman hasil berikutnya. Nilai ini adalah null ketika tidak ada lagi hasil untuk dikembalikan.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 4096.

Pola: `[a-zA-Z0-9+/\]{0,2}`

## Galat

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas. Coba buat panggilan nanti. Untuk informasi tentang batas, lihat Batas [Kinesis Video Streams](#).

Kode Status HTTP: 400

### InvalidArgumentException

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan.

Kode Status HTTP: 400

### NotAuthorizedException

Kode Status: 403, Penelepon tidak berwenang untuk melakukan operasi pada aliran yang diberikan, atau token telah kedaluwarsa.

Kode Status HTTP: 401

### ResourceNotFoundException

GetImagesakan memunculkan kesalahan ini ketika Kinesis Video Streams tidak dapat menemukan aliran yang Anda tentukan.

GetHLSStreamingSessionURL dan GetDASHStreamingSessionURL membuang kesalahan ini jika sesi dengan PlaybackMode dari ON\_DEMAND atau LIVE\_REPLAY diminta untuk aliran yang tidak memiliki fragmen dalam rentang waktu yang diminta, atau jika sesi dengan PlaybackMode of LIVE diminta untuk aliran yang tidak memiliki fragmen dalam 30 detik terakhir.

Kode Status HTTP: 404



## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go.](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## Saluran Pensinyalan Video Amazon Kinesis

Tindakan berikut didukung oleh Saluran Pensinyalan Video Amazon Amazon Amazon:

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

## GetIceServerConfig

Layanan: Amazon Kinesis Video Signaling Channels

Catatan: Sebelum menggunakan API ini, Anda harus memanggil `GetSignalingChannelEndpoint` API untuk meminta titik akhir HTTPS. Anda kemudian menentukan titik akhir dan wilayah dalam permintaan `GetIceServerConfig` API Anda.

Mendapatkan informasi konfigurasi server Interactive Connectivity Establishment (ICE), termasuk URI, nama pengguna, dan kata sandi yang dapat digunakan untuk mengonfigurasi koneksi WebRTC. Komponen ICE menggunakan informasi konfigurasi ini untuk mengatur koneksi WebRTC, termasuk mengautentikasi dengan Traversal Using Relays around NAT (TURN) relay server.

TURN adalah protokol yang digunakan untuk meningkatkan konektivitas peer-to-peer aplikasi. Dengan menyediakan layanan relai berbasis cloud, TURN memastikan bahwa koneksi dapat dibuat bahkan ketika satu atau lebih rekan tidak mampu melakukan koneksi langsung. peer-to-peer Untuk informasi selengkapnya, lihat [REST API Untuk Akses Untuk Mengubah Layanan](#).

Anda dapat menjalankan API ini untuk membuat mekanisme fallback jika salah satu rekan tidak dapat membuat peer-to-peer koneksi langsung melalui saluran pensinyalan. Anda harus menentukan Nama Sumber Daya Amazon (ARN) saluran pensinyalan Anda untuk menjalankan API ini.

### Minta Sintaks

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ClientId": "string",
  "Service": "string",
  "Username": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

## ChannelARN

ARN dari saluran pensinyalan yang akan digunakan untuk peer-to-peer koneksi antara rekan yang dikonfigurasi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

## ClientId

Pengidentifikasi unik untuk penampil. Harus unik di dalam saluran pensinyalan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

## Service

Menentukan layanan yang diinginkan. Saat ini, TURN adalah satu-satunya nilai yang valid.

Jenis: String

Nilai yang Valid: TURN

Wajib: Tidak

## Username

ID pengguna opsional untuk dikaitkan dengan kredensialnya.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

## Sintaksis Respons

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

{
  "IceServerList": [
    {
      "Password": "string",
      "Ttl": number,
      "Uris": [ "string" ],
      "Username": "string"
    }
  ]
}
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

#### [IceServerList](#)

Daftar objek informasi server ICE.

Tipe: Array objek [IceServer](#)

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

#### ClientLimitExceededException

Permintaan Anda dibatasi karena Anda telah melampaui batas panggilan klien yang diizinkan.  
Coba buat panggilan nanti.

Kode Status HTTP: 400

#### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

InvalidClientException

Klien yang ditentukan tidak valid.

Kode Status HTTP: 400

NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

ResourceNotFoundException

Sumber daya yang ditentukan tidak ditemukan.

Kode Status HTTP: 404

SessionExpiredException

Jika sesi klien kedaluwarsa. Setelah klien terhubung, sesi berlaku selama 45 menit. Klien harus menyambung kembali ke saluran untuk melanjutkan pengiriman/menerima pesan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)



## SendAlexaOfferToMaster

Layanan: Amazon Kinesis Video Signaling Channels

### Note

Sebelum menggunakan API ini, Anda harus memanggil `GetSignalingChannelEndpoint` API untuk mendapatkan titik akhir. Anda kemudian menentukan titik akhir dan wilayah dalam permintaan `SendAlexaOfferToMaster` API Anda.

API ini memungkinkan Anda untuk menghubungkan perangkat berkemampuan WebRTC dengan perangkat tampilan Alexa. Ketika dipanggil, ia mengirimkan penawaran Alexa Session Description Protocol (SDP) ke master peer. Penawaran dikirimkan segera setelah master terhubung ke saluran pensinyalan yang ditentukan. API ini mengembalikan jawaban SDP dari master yang terhubung. Jika master tidak terhubung ke saluran pensinyalan, permintaan pengiriman ulang dilakukan hingga pesan kedaluwarsa.

### Minta Sintaks

```
POST /v1/send-alexa-offer-to-master HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "MessagePayload": "string",
  "SenderClientId": "string"
}
```

### Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

### Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

#### ChannelARN

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan tempat Alexa dan rekan master berkomunikasi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Ya

### MessagePayload

SDP yang dikodekan base64 menawarkan konten.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 10000.

Pola: `[a-zA-Z0-9+/=]+`

Diperlukan: Ya

### SenderClientId

Pengidentifikasi unik untuk klien pengirim.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Ya

## Sintaksis Respons

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

{
  "Answer": "string"
}
```

## Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.



Layanan mengembalikan data berikut dalam format JSON.

### Answer

Konten jawaban SDP yang dikodekan base64.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 10000.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

#### ClientLimitExceededException

Permintaan Anda dibatasi karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

#### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

#### NotAuthorizedException

Penelepon tidak berwenang untuk melakukan operasi ini.

Kode Status HTTP: 401

#### ResourceNotFoundException

Sumber daya yang ditentukan tidak ditemukan.

Kode Status HTTP: 404

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSAntarmuka Baris Perintah](#)
- [AWSSDK for .NET](#)
- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK untuk V3 JavaScript](#)
- [AWSSDK for PHP V3](#)
- [AWSSDK untuk Python](#)
- [AWSSDK for Ruby V3](#)

## Penyimpanan WebRTC Amazon

Tindakan berikut didukung oleh Amazon Kinesis Penyimpanan WebRTC Amazon:

- [JoinStorageSession](#)

## JoinStorageSession

Layanan: Amazon Kinesis Video WebRTC Storage

### Note

Sebelum menggunakan API ini, Anda harus memanggil `GetSignalingChannelEndpoint` API untuk meminta endpoint WEBRTC. Anda kemudian menentukan titik akhir dan wilayah dalam permintaan `JoinStorageSession` API Anda.

Bergabunglah dengan sesi WebRTC audio satu arah dan/atau multi-arah yang sedang berlangsung sebagai perangkat penghasil video untuk saluran input. Jika tidak ada sesi yang ada untuk saluran tersebut, sesi streaming baru perlu dibuat, dan Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan harus disediakan.

Saat ini untuk `SINGLE_MASTER` jenisnya, perangkat penghasil video mampu menyerap media audio dan video ke dalam aliran. Hanya perangkat penghasil video yang dapat bergabung dengan sesi dan merekam media.

### Important

Baik trek audio dan video saat ini diperlukan untuk konsumsi WebRTC.

Setelah peserta master menegosiasikan koneksi melalui WebRTC, sesi media yang dicerna akan disimpan dalam aliran video Kinesis. Beberapa pemirsa kemudian dapat memutar kembali media real-time melalui API Pemutaran kami.

Pelanggan juga dapat menggunakan HLS fitur Kinesis Video Streams yang ada DASH seperti atau pemutaran, Pembuatan gambar, dan lainnya dengan media WebRTC yang tertelan.

### Note

Pengiriman gambar S3 saat ini tidak didukung.

**Note**

Asumsikan bahwa hanya satu klien perangkat penghasil video yang dapat dikaitkan dengan sesi untuk saluran tersebut. Jika lebih dari satu klien bergabung dengan sesi saluran tertentu sebagai perangkat penghasil video, permintaan klien terbaru diutamakan.

**Informasi tambahan**

- Idempoten - API ini tidak idempoten.
- Perilaku coba lagi - Ini dihitung sebagai panggilan API baru.
- Panggilan bersamaan - Panggilan bersamaan diizinkan. Penawaran dikirim satu kali per setiap panggilan.

**Minta Sintaks**

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json

{
  "channelArn": "string"
}
```

**Parameter Permintaan URI**

Permintaan tidak menggunakan parameter URI apa pun.

**Isi Permintaan**

Permintaan menerima data berikut dalam format JSON.

**[channelArn](#)**

Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan.

Jenis: String

Pola: `^arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+$`

Diperlukan: Ya

## Sintaksis Respons

```
HTTP/1.1 200
```

### Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

### Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat [Kesalahan Umum](#).

#### AccessDeniedException

Anda tidak memiliki izin yang diperlukan untuk melakukan operasi ini.

Kode Status HTTP: 403

#### ClientLimitExceededException

Kinesis Video Streams telah membatasi permintaan karena Anda telah melampaui batas panggilan klien yang diizinkan. Coba buat panggilan nanti.

Kode Status HTTP: 400

#### InvalidArgumentException

Nilai untuk parameter input ini tidak valid.

Kode Status HTTP: 400

#### ResourceNotFoundException

Sumber daya yang ditentukan tidak ditemukan.

Kode Status HTTP: 404

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS Antarmuka Baris Perintah](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

## Tipe Data

Jenis data berikut didukung oleh 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](#)

Jenis data berikut didukung oleh Amazon Kinesis Video Streams Media:

- [StartSelector](#)

Jenis data berikut didukung oleh Amazon Kinesis Video Streams Media yang Diarsipkan:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Jenis data berikut didukung oleh Amazon Kinesis Video Signaling Channels:

- [IceServer](#)

Jenis data berikut didukung oleh Amazon Kinesis Video WebRTC Storage:

## Amazon Kinesis Video Streams

Jenis data berikut didukung oleh 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

Layanan: Amazon Kinesis Video Streams

Struktur yang merangkum metadata dan properti saluran sinyal.

### Daftar Isi

#### ChannelARN

Amazon Resource Name (ARN) dari kanal pensinyalan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### ChannelName

Nama saluran pensinyalan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

#### ChannelStatus

Status terkini saluran pensinyalan.

Jenis: String

Nilai yang Valid: `CREATING | ACTIVE | UPDATING | DELETING`

Wajib: Tidak

#### ChannelType

Tipe saluran pensinyalan.

Jenis: String

Nilai yang Valid: SINGLE\_MASTER | FULL\_MESH

Wajib: Tidak

CreationTime

Waktu saat saluran pensinyalan dibuat.

Tipe: Timestamp

Wajib: Tidak

SingleMasterConfiguration

Struktur yang berisi konfigurasi tipe SINGLE\_MASTER saluran.

Tipe: Objek [SingleMasterConfiguration](#)

Wajib: Tidak

Version

Versi saluran pensinyalan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: [a-zA-Z0-9]+

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)



## ChannelNameCondition

Layanan: Amazon Kinesis Video Streams

Parameter input opsional untuk `ListSignalingChannels` API. Ketika parameter ini ditentukan saat memohon `ListSignalingChannels`, API hanya mengembalikan saluran yang memenuhi kondisi yang ditentukan dalam `ChannelNameCondition`.

Daftar Isi

### ComparisonOperator

Sebuah operator perbandingan Saat ini, Anda hanya dapat menentukan `BEGINS_WITH` operator, yang menemukan saluran sinyal yang namanya dimulai dengan awalan yang diberikan.

Jenis: String

Nilai yang Valid: `BEGINS_WITH`

Wajib: Tidak

### ComparisonValue

Nilai untuk membandingkan

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.- ]+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)



## DeletionConfig

Layanan: Amazon Kinesis Video Streams

Detail konfigurasi yang diperlukan untuk menghapus koneksi aliran dari Agen Edge.

Daftar Isi

### DeleteAfterUpload

`boolean` Nilai yang digunakan untuk menunjukkan apakah Anda ingin menandai media untuk dihapus atau tidak, setelah diunggah ke cloud Kinesis Video Stream. File media dapat dihapus jika salah satu nilai konfigurasi penghapusan diatur ke `true`, seperti ketika batas untuk `EdgeRetentionInHours`, atau `MaxLocalMediaSizeInMB`, telah tercapai.

Karena nilai default diatur ke `true`, konfigurasi jadwal pengunggah sehingga file media tidak dihapus sebelum awalnya diunggah ke AWS cloud.

Tipe: Boolean

Wajib: Tidak

### EdgeRetentionInHours

Jumlah jam yang Anda ingin menyimpan data dalam aliran pada Agen Edge. Nilai default waktu retensi adalah 720 jam, yang diterjemahkan menjadi 30 hari.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 720.

Wajib: Tidak

### LocalSizeConfig

Nilai ukuran lokal yang diperlukan untuk menghapus konfigurasi tepi.

Tipe: Objek [LocalSizeConfig](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## EdgeAgentStatus

Layanan: Amazon Kinesis Video Streams

Objek yang berisi detail status terbaru untuk perekam agen edge dan pekerjaan pengunggah. Gunakan informasi ini untuk menentukan kesehatan agen tepi saat ini.

Daftar Isi

### LastRecorderStatus

Status terbaru dari pekerjaan perekaman tepi aliran.

Tipe: Objek [LastRecorderStatus](#)

Wajib: Tidak

### LastUploaderStatus

Status terbaru pekerjaan pengunggah edge to cloud stream.

Tipe: Objek [LastUploaderStatus](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK untuk C++](#)
- [AWSSDK untuk Go](#)
- [AWSSDK untuk Java V2](#)
- [AWSSDK para Ruby V3](#)



## EdgeConfig

Layanan: Amazon Kinesis Video Streams

Deskripsi konfigurasi tepi aliran yang akan digunakan untuk menyinkronkan dengan komponen Edge Agent IoT Greengrass. Komponen Edge Agent akan berjalan pada pengaturan Perangkat IoT Hub di premis Anda.

### Daftar Isi

#### HubDeviceArn

The "Internet of Things (IoT) Hal" Arn sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Diperlukan: Ya

#### RecorderConfig

Konfigurasi perekam terdiri dari `MediaSourceConfig` detail lokal, yang digunakan sebagai kredensial untuk mengakses file media lokal yang dialirkan pada kamera.

Tipe: Objek [RecorderConfig](#)

Wajib: Ya

#### DeletionConfig

Konfigurasi penghapusan terdiri dari detail waktu retensi (`EdgeRetentionInHours`) dan konfigurasi ukuran lokal (`LocalSizeConfig`) yang digunakan untuk melakukan penghapusan.

Tipe: Objek [DeletionConfig](#)

Wajib: Tidak

#### UploaderConfig

Konfigurasi pengunggah berisi `ScheduleExpression` detail yang digunakan untuk menjadwalkan pekerjaan unggahan untuk file media yang direkam dari Agen Edge ke Aliran Video Kinesis.

Tipe: Objek [UploaderConfig](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK for bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## ImageGenerationConfiguration

Layanan: Amazon Kinesis Video Streams

Struktur yang berisi informasi yang diperlukan untuk pengiriman gambar KVS. Jika null, konfigurasi akan dihapus dari sungai.

### Daftar Isi

#### DestinationConfig

Struktur yang berisi informasi yang diperlukan untuk mengirimkan gambar ke pelanggan.

Tipe: Objek [ImageGenerationDestinationConfig](#)

Wajib: Ya

#### Format

Format gambar yang diterima.

Jenis: String

Nilai yang Valid: JPEG | PNG

Wajib: Ya

#### ImageSelectorType

Asal stempel waktu Server atau Produser yang digunakan untuk menghasilkan gambar.

Jenis: String

Nilai yang Valid: SERVER\_TIMESTAMP | PRODUCER\_TIMESTAMP

Wajib: Ya

#### SamplingInterval

Interval waktu dalam milidetik (ms) di mana gambar perlu dihasilkan dari sungai. Nilai minimum yang dapat diberikan adalah 200 ms. Jika rentang stempel waktu kurang dari interval pengambilan sampel, Gambar dari `StartTimestamp` akan dikembalikan jika tersedia.

Tipe: Bilangan Bulat

Valid Range: Nilai minimum 3000. Nilai maksimum 20000.

Wajib: Ya

## Status

Menunjukkan apakah `ContinuousImageGenerationConfigurationsAPI` diaktifkan atau dinonaktifkan.

Jenis: String

Nilai yang Valid: ENABLED | DISABLED

Wajib: Ya

## FormatConfig

Daftar struktur pasangan kunci-nilai yang berisi parameter tambahan yang dapat diterapkan ketika gambar yang dihasilkan. Yang `FormatConfig` kuncinya adalah `JPEGQuality`, yang menunjukkan kunci kualitas JPEG yang akan digunakan untuk menghasilkan gambar. Yang `FormatConfig` nilai menerima ints dari 1 sampai 100. Jika nilainya 1, gambar akan dihasilkan dengan kualitas kurang dan kompresi terbaik. Jika nilainya 100, gambar akan dihasilkan dengan kualitas terbaik dan kompresi yang lebih sedikit. Jika tidak ada nilai yang disediakan, nilai default dari `JPEGQuality` kunci akan diatur ke 80.

Tipe: Peta string ke string

Entri Peta: Jumlah maksimum 1 item.

Kunci yang Valid: `JPEGQuality`

Batasan Panjang Nilai: Panjang minimum 0. Panjang maksimum 256.

Pola nilai: `^[a-zA-Z_0-9]+`

Wajib: Tidak

## HeightPixels

Ketinggian gambar output yang digunakan dalam hubungannya dengan `WidthPixels` parameter. Ketika keduanya `HeightPixels` dan `WidthPixels` parameter disediakan, gambar akan diregangkan agar sesuai dengan rasio aspek yang ditentukan. Kalau saja `HeightPixels` parameter disediakan, rasio aspek aslinya akan digunakan untuk menghitung `WidthPixels` rasio. Jika tidak ada parameter yang disediakan, ukuran gambar asli akan dikembalikan.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 2160.

Wajib: Tidak

#### WidthPixels

Lebar gambar output yang digunakan dalam hubungannya dengan `HeightPixels` parameter. Ketika keduanya `WidthPixels` dan `HeightPixels` parameter disediakan, gambar akan diregangkan agar sesuai dengan rasio aspek yang ditentukan. Kalau saja `WidthPixels` parameter disediakan, rasio aspek aslinya akan digunakan untuk menghitung `HeightPixels` rasio. Jika tidak ada parameter yang disediakan, ukuran gambar asli akan dikembalikan.

Tipe: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 3840.

Wajib: Tidak

#### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWSSDK, lihat yang berikut ini:

- [AWSSDK untuk C++](#)
- [AWSSDK untuk Go](#)
- [AWSSDK untuk Java V2](#)
- [AWSSDK para Ruby V3](#)

## ImageGenerationDestinationConfig

Layanan: Amazon Kinesis Video Streams

Struktur yang berisi informasi yang diperlukan untuk mengirimkan gambar ke pelanggan.

### Daftar Isi

#### DestinationRegion

AWS Wilayah dari bucket S3 tempat gambar dikirimkan. Ini `DestinationRegion` harus sesuai dengan Wilayah tempat aliran berada.

Jenis: String

Batasan Panjang: Panjang minimum 9. Panjang maksimum 14.

Pola: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Diperlukan: Ya

#### Uri

Uniform Resource Identifier (URI) yang mengidentifikasi di mana gambar akan dikirimkan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `^[a-zA-Z_0-9]+:(//)?(^[^/]+)/?([^\s]*)$`

Diperlukan: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)



## LastRecorderStatus

Layanan: Amazon Kinesis Video Streams

Status terbaru dari pekerjaan perekaman tepi aliran.

### Daftar Isi

#### JobStatusDetails

Deskripsi status terbaru pekerjaan perekam.

Tipe: String

Wajib: Tidak

#### LastCollectedTime

Stempel waktu di mana pekerjaan perekam terakhir dieksekusi dan media disimpan ke disk lokal.

Tipe: Timestamp

Wajib: Tidak

#### LastUpdatedTime

Stempel waktu di mana status perekam terakhir diperbarui.

Tipe: Timestamp

Wajib: Tidak

#### RecorderStatus

Status pekerjaan perekam terbaru.

Jenis: String

Nilai yang Valid: SUCCESS | USER\_ERROR | SYSTEM\_ERROR

Wajib: Tidak

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:



- [AWSSDK untuk C++](#)
- [AWSSDK untuk Go](#)
- [AWSSDK untuk Java V2](#)
- [AWSSDK para Ruby V3](#)

## LastUploaderStatus

Layanan: Amazon Kinesis Video Streams

Status terbaru pekerjaan pengunggah edge to cloud stream.

### Daftar Isi

#### JobStatusDetails

Deskripsi status terbaru pekerjaan pengunggah.

Tipe: String

Wajib: Tidak

#### LastCollectedTime

Stempel waktu di mana pekerjaan pengunggah terakhir dieksekusi dan media dikumpulkan ke cloud.

Tipe: Timestamp

Wajib: Tidak

#### LastUpdatedTime

Stempel waktu di mana status pengunggah terakhir diperbarui.

Tipe: Timestamp

Wajib: Tidak

#### UploaderStatus

Status pekerjaan pengunggah terbaru.

Jenis: String

Nilai yang Valid: SUCCESS | USER\_ERROR | SYSTEM\_ERROR

Wajib: Tidak

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK untuk C++](#)
- [AWSSDK untuk Go](#)
- [AWSSDK untuk Java V2](#)
- [AWSSDK para Ruby V3](#)

## ListEdgeAgentConfigurationsEdgeConfig

Layanan: Amazon Kinesis Video Streams

Deskripsi konfigurasi tepi aliran tunggal.

### Daftar Isi

#### CreationTime

Stempel waktu saat streaming pertama kali membuat konfigurasi tepi.

Tipe: Timestamp

Wajib: Tidak

#### EdgeConfig

Deskripsi konfigurasi tepi aliran yang akan digunakan untuk menyinkronkan dengan komponen Edge Agent IoT Greengrass. Komponen Edge Agent akan berjalan pada pengaturan Perangkat IoT Hub di premis Anda.

Tipe: Objek [EdgeConfig](#)

Wajib: Tidak

#### FailedStatusDetails

Deskripsi status kegagalan yang dihasilkan.

Tipe: String

Wajib: Tidak

#### LastUpdatedTime

Stempel waktu saat streaming terakhir memperbarui konfigurasi edge.

Tipe: Timestamp

Wajib: Tidak

#### StreamARN

Amazon Resource Name (ARN) aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

#### StreamName

Nama sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Wajib: Tidak

#### SyncStatus

Status sinkronisasi saat ini dari konfigurasi tepi aliran.

Jenis: String

Nilai yang Valid: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

Wajib: Tidak

#### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK untuk C++](#)
- [AWSSDK untuk Go](#)
- [AWSSDK untuk Java V2](#)
- [AWSSDK para Ruby V3](#)

## LocalSizeConfig

Layanan: Amazon Kinesis Video Streams

Detail konfigurasi yang menyertakan ukuran maksimum media (`MaxLocalMediaSizeInMB`) yang ingin Anda simpan untuk stream di Edge Agent, serta strategi yang harus digunakan (`StrategyOnFullSize`) ketika ukuran maksimum stream telah tercapai.

Daftar Isi

### MaxLocalMediaSizeInMB

Ukuran maksimum keseluruhan media yang ingin Anda simpan untuk streaming di Agen Edge.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 64. Nilai maksimum 2000000.

Wajib: Tidak

### StrategyOnFullSize

Strategi untuk melakukan ketika `MaxLocalMediaSizeInMB` batas aliran tercapai.

Jenis: String

Nilai yang Valid: `DELETE_OLDEST_MEDIA` | `DENY_NEW_MEDIA`

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## MappedResourceConfigurationListItem

Layanan: Amazon Kinesis Video Streams

Struktur yang merangkum, atau berisi, properti konfigurasi penyimpanan media.

Daftar Isi

ARN

Amazon Resource Name (ARN) sumber daya Kinesis Video Stream, yang terkait dengan aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

Type

Jenis sumber daya terkait untuk aliran video kinesis.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## MediaSourceConfig

Layanan: Amazon Kinesis Video Streams

Detail konfigurasi yang terdiri dari kredensi yang diperlukan (MediaUriSecretArn dan MediaUriType) untuk mengakses file media yang dialirkan ke kamera.

Daftar Isi

### MediaUriSecretArn

AWS Secrets Manager ARN untuk nama pengguna dan kata sandi kamera, atau lokasi file media lokal.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

Diperlukan: Ya

### MediaUriType

Uniform Resource Identifier (URI). FILE\_URI Nilai dapat digunakan untuk melakukan streaming file media lokal.

#### Note

Pratinjau hanya mendukung format URI sumber RTSP\_URI media.

Jenis: String

Nilai yang Valid: RTSP\_URI | FILE\_URI

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:



- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## MediaStorageConfiguration

Layanan: Amazon Kinesis Video Streams

Struktur yang merangkum, atau berisi, properti konfigurasi penyimpanan media.

- Jika `StorageStatus` diaktifkan, data akan disimpan dalam yang `StreamARN` disediakan. Agar WebRTC Ingestion berfungsi, aliran harus mengaktifkan retensi data.
- Jika `StorageStatus` dinonaktifkan, tidak ada data yang akan disimpan, dan `StreamARN` parameter tidak akan diperlukan.

### Daftar Isi

#### Status

Status konfigurasi penyimpanan media.

Jenis: String

Nilai yang Valid: ENABLED | DISABLED

Wajib: Ya

#### StreamARN

Nama Sumber Daya Amazon (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Diperlukan: Tidak

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSSDK for C++](#)

- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## NotificationConfiguration

Layanan: Amazon Kinesis Video Streams

Struktur yang berisi informasi notifikasi untuk pengiriman gambar KVS. Jika parameter ini adalah null, konfigurasi akan dihapus dari aliran.

Daftar Isi

### DestinationConfig

Informasi tujuan yang diperlukan untuk mengirimkan pemberitahuan kepada pelanggan.

Tipe: Objek [NotificationDestinationConfig](#)

Wajib: Ya

Status

Menunjukkan apakah konfigurasi pemberitahuan diaktifkan atau dinonaktifkan.

Jenis: String

Nilai yang Valid: ENABLED | DISABLED

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## NotificationDestinationConfig

Layanan: Amazon Kinesis Video Streams

Struktur yang berisi informasi yang diperlukan untuk menyampaikan pemberitahuan kepada pelanggan.

### Daftar Isi

#### Uri

Uniform Resource Identifier (URI) yang mengidentifikasi di mana gambar akan dikirimkan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 255.

Pola: `^[a-zA-Z_0-9]+:(//)?( [/]+ )/?([ ^* ]*)$`

Diperlukan: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## RecorderConfig

Layanan: Amazon Kinesis Video Streams

Konfigurasi perekam terdiri dari `MediaSourceConfig` detail lokal yang digunakan sebagai kredensi untuk mengakses file media lokal yang dialirkan pada kamera.

Daftar Isi

### MediaSourceConfig

Detail konfigurasi yang terdiri dari kredensi yang diperlukan (`MediaUriSecretArn` dan `MediaUriType`) untuk mengakses file media yang dialirkan ke kamera.

Tipe: Objek [MediaSourceConfig](#)

Wajib: Ya

### ScheduleConfig

Konfigurasi yang terdiri dari `ScheduleExpression` dan `DurationInMinutes` rincian yang menentukan penjadwalan untuk merekam dari kamera, atau file media lokal, ke Agen Edge. Jika `ScheduleExpression` atribut tidak disediakan, maka Edge Agent akan selalu diatur ke mode perekaman.

Tipe: Objek [ScheduleConfig](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## ResourceEndpointListItem

Layanan: Amazon Kinesis Video Streams

Objek yang menggambarkan titik akhir saluran pensinyalan yang dikembalikan oleh API.

`GetSignalingChannelEndpoint`

Endpoint server media akan sesuai dengan WEBRTC Protokol.

Daftar Isi

Protocol

Protokol saluran pensinyalan dikembalikan oleh `GetSignalingChannelEndpoint` API.

Jenis: String

Nilai yang Valid: WSS | HTTPS | WEBRTC

Wajib: Tidak

ResourceEndpoint

Titik akhir saluran pensinyalan yang dikembalikan oleh API. `GetSignalingChannelEndpoint`

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## ScheduleConfig

Layanan: Amazon Kinesis Video Streams

API ini memungkinkan Anda untuk menentukan durasi yang kamera, atau file media lokal, harus merekam ke Edge Agent. ScheduleConfig terdiri dari ScheduleExpression dan DurationInMinutes atribut.

Jika tidak ScheduleConfig disediakan di RecorderConfig, maka Edge Agent akan selalu diatur ke mode perekaman.

Jika tidak ScheduleConfig disediakan di UploaderConfig, maka Agen Edge akan mengunggah secara berkala (setiap 1 jam).

### Daftar Isi

#### DurationInSeconds

Total durasi untuk merekam media. Jika ScheduleExpression atribut disediakan, maka DurationInSeconds atribut juga harus ditentukan.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 60. Nilai maksimum 3600.

Wajib: Ya

#### ScheduleExpression

Ekspresi cron Quartz yang menangani penjadwalan pekerjaan untuk merekam dari kamera, atau file media lokal, ke Edge Agent. Jika tidak ScheduleExpression disediakan untuk RecorderConfig, maka Edge Agent akan selalu diatur ke mode perekaman.

Untuk informasi lebih lanjut tentang Quartz, lihat [halaman Cron Trigger Tutorial](#) untuk memahami ekspresi yang valid dan penggunaannya.

Jenis: String

Panjang minimum 11. Panjang maksimum 100.

Pola: `[^\n]{11,100}`

Diperlukan: Ya



## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## SingleMasterChannelEndpointConfiguration

Layanan: Amazon Kinesis Video Streams

Objek yang berisi konfigurasi endpoint untuk jenis SINGLE\_MASTER saluran.

### Daftar Isi

### Protocols

Properti ini digunakan untuk menentukan sifat komunikasi melalui saluran SINGLE\_MASTER pensinyalan ini. Jika WSS ditentukan, API ini mengembalikan endpoint websocket. Jika HTTPS ditentukan, API ini mengembalikan HTTPS titik akhir.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 5 item.

Nilai yang Valid: WSS | HTTPS | WEBRTC

Wajib: Tidak

### Role

Properti ini digunakan untuk menentukan izin pesan di saluran SINGLE\_MASTER pensinyalan ini. Jika MASTER ditentukan, API ini mengembalikan titik akhir yang dapat digunakan klien untuk menerima penawaran dan mengirim jawaban ke salah satu pemirsa di saluran pensinyalan ini. Jika VIEWER ditentukan, API ini mengembalikan titik akhir yang hanya dapat digunakan klien untuk mengirim penawaran ke MASTER klien lain di saluran pensinyalan ini.

Jenis: String

Nilai yang Valid: MASTER | VIEWER

Wajib: Tidak

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)

- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## SingleMasterConfiguration

Layanan: Amazon Kinesis Video Streams

Struktur yang berisi konfigurasi yang berisi SINGLE\_MASTER konfigurasi perangkat.

Daftar Isi

### MessageTtlSeconds

Jangka waktu saluran pensinyalan menyimpan pesan yang tidak terkirim sebelum dibuang.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 5. Nilai maksimum 120.

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## StreamInfo

Layanan: Amazon Kinesis Video Streams

Sebuah objek yang menggambarkan aliran video Kinesis.

### Daftar Isi

#### CreationTime

Cap waktu yang menunjukkan kapan aliran dibuat.

Tipe: Timestamp

Wajib: Tidak

#### DataRetentionInHours

Berapa lama aliran menyimpan data, dalam beberapa jam.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 0.

Wajib: Tidak

#### DeviceName

Nama perangkat yang terkait dengan aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: [a-zA-Z0-9\_.-]+

Wajib: Tidak

#### KmsKeyId

ID kunci AWS Key Management Service (AWS KMS) yang digunakan Kinesis Video Streams untuk mengenkripsi data di stream.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2048.

Pola: .+

Wajib: Tidak

## MediaType

MediaTypeDari sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Wajib: Tidak

## Status

Status aliran.

Jenis: String

Nilai yang Valid: CREATING | ACTIVE | UPDATING | DELETING

Wajib: Tidak

## StreamARN

Amazon Resource Name (ARN) dari aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Wajib: Tidak

## StreamName

Nama aliran.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Wajib: Tidak

Version

Versi sungai.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: [a-zA-Z0-9]+

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## StreamNameCondition

Layanan: Amazon Kinesis Video Streams

Menentukan kondisi bahwa aliran harus memenuhi untuk dikembalikan ketika Anda daftar aliran (lihat API). `ListStreams` Kondisi memiliki operasi perbandingan dan nilai. Saat ini, Anda hanya dapat menentukan `BEGINS_WITH` operator, yang menemukan aliran yang namanya dimulai dengan awalan yang diberikan.

Daftar Isi

### ComparisonOperator

Sebuah operator perbandingan. Saat ini, Anda hanya dapat menentukan `BEGINS_WITH` operator, yang menemukan aliran yang namanya dimulai dengan awalan yang diberikan.

Jenis: String

Nilai yang Valid: `BEGINS_WITH`

Wajib: Tidak

### ComparisonValue

Nilai untuk membandingkan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `[a-zA-Z0-9_.-]+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)



- [AWSSDK for Ruby V3](#)

## Tag

Layanan: Amazon Kinesis Video Streams

Pasangan nilai kunci yang terkait dengan saluran.

### Daftar Isi

#### Key

Kunci tag yang terkait dengan saluran pensinyalan yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^([\backslash p\{L}\backslash p\{Z}\backslash p\{N}\_ . : / = + \backslash - @ ] ^ * ) \$$

Diperlukan: Ya

#### Value

Nilai tag yang terkait dengan saluran pensinyalan yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola:  $[\backslash p\{L}\backslash p\{Z}\backslash p\{N}\_ . : / = + \backslash - @ ] ^ *$

Diperlukan: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## UploaderConfig

Layanan: Amazon Kinesis Video Streams

Konfigurasi yang terdiri dari `ScheduleExpression` dan `DurationInMinutes` rincian yang menentukan penjadwalan untuk merekam dari kamera, atau file media lokal, ke Agen Edge. Jika tidak `ScheduleConfig` disediakan di `UploaderConfig`, maka Agen Edge akan mengunggah secara berkala (setiap 1 jam).

Daftar Isi

### ScheduleConfig

Konfigurasi yang terdiri dari `ScheduleExpression` dan `DurationInMinutes` rincian yang menentukan penjadwalan untuk merekam dari kamera, atau file media lokal, ke Agen Edge. Jika tidak `ScheduleConfig` disediakan dalam hal ini `UploaderConfig`, maka Edge Agent akan mengunggah secara berkala (setiap 1 jam).

Tipe: Objek [ScheduleConfig](#)

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK for bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Arus Amazon Kinesis Video Streams

Jenis data berikut didukung oleh Amazon Kinesis Video Streams Media:

- [StartSelector](#)

## StartSelector

Layanan: Amazon Kinesis Video Streams Media

Mengidentifikasi potongan pada aliran video Kinesis di mana Anda ingin GetMedia API mulai mengembalikan data media. Anda memiliki opsi berikut untuk mengidentifikasi bagian awal:

- Pilih potongan terbaru (atau tertua).
- Identifikasi potongan tertentu. Anda dapat mengidentifikasi potongan tertentu baik dengan memberikan nomor fragmen atau stempel waktu (server atau produser).
- Setiap metadata potongan menyertakan token kelanjutan sebagai tag Matroska (MKV) (). `AWS_KINESISVIDEO_CONTINUATION_TOKEN` Jika GetMedia permintaan sebelumnya dihentikan, Anda dapat menggunakan nilai tag ini di GetMedia permintaan berikutnya. API kemudian mulai mengembalikan potongan mulai di mana API terakhir berakhir.

### Daftar Isi

#### StartSelectorType

Mengidentifikasi fragmen pada aliran video Kinesis tempat Anda ingin mulai mendapatkan datanya.

- `SEKARANG` - Mulailah dengan potongan terbaru di sungai.
- `EARLIEST` - Mulailah dengan potongan paling awal yang tersedia di sungai.
- `FRAGMENT_NUMBER` - Mulai dengan potongan setelah fragmen tertentu. Anda juga harus menentukan parameter `AfterFragmentNumber`.
- `PRODUCER_TIMESTAMP` atau `SERVER_TIMESTAMP` - Mulailah dengan potongan yang berisi fragmen dengan stempel waktu produser atau server yang ditentukan. Anda menentukan stempel waktu dengan menambahkan `StartTimeStamp`
- `CONTINUATION_TOKEN` - Baca menggunakan token kelanjutan yang ditentukan.

#### Note

Jika Anda memilih `NOW`, `EARLIEST`, atau `CONTINUATION_TOKEN` sebagai `startSelectorType`, Anda tidak memberikan informasi tambahan apa pun di `startSelector`

Jenis: String

Nilai yang Valid: FRAGMENT\_NUMBER | SERVER\_TIMESTAMP | PRODUCER\_TIMESTAMP | NOW | EARLIEST | CONTINUATION\_TOKEN

Wajib: Ya

#### AfterFragmentNumber

Menentukan nomor fragmen dari mana Anda ingin GetMedia API untuk mulai mengembalikan fragmen.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^{\wedge}[0-9]^{\wedge}\$$

Wajib: Tidak

#### ContinuationToken

Token kelanjutan yang dikembalikan oleh Kinesis Video Streams dalam respons sebelumnya. GetMedia GetMediaAPI kemudian dimulai dengan potongan yang diidentifikasi oleh token kelanjutan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^{\wedge}[a-zA-Z0-9\_\\.\\-]^{\wedge}\$$

Wajib: Tidak

#### StartTimestamp

Sebuah nilai timestamp. Nilai ini diperlukan jika Anda memilih PRODUCER\_TIMESTAMP atau SERVER\_TIMESTAMP sebagai startSelectorType GetMediaAPI kemudian dimulai dengan potongan yang berisi fragmen yang memiliki stempel waktu yang ditentukan.

Tipe: Timestamp

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Arus Amazon Kinesis Video Streams

Jenis data berikut didukung oleh Amazon Kinesis Video Streams Archived Media:

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

## ClipFragmentSelector

Layanan: Amazon Kinesis Video Streams Archived Media

Menjelaskan rentang timestamp dan asal stempel waktu dari berbagai fragmen.

Fragmen yang memiliki stempel waktu produsen duplikat dideduplikasi. Ini berarti bahwa jika produsen memproduksi aliran fragmen dengan stempel waktu produsen yang kira-kira sama dengan waktu clock yang sebenarnya, klip akan berisi semua fragmen dalam rentang stempel waktu yang diminta. Jika beberapa fragmen tertelan dalam rentang waktu yang sama dan titik waktu yang sangat berbeda, hanya koleksi fragmen tertelan tertua yang dikembalikan.

### Daftar Isi

#### FragmentSelectorType

Asal stempel waktu untuk digunakan (Server atau Produser).

Jenis: String

Nilai yang Valid: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

Wajib: Ya

#### TimestampRange

Kisaran cap waktu untuk kembali.

Tipe: Objek [ClipTimestampRange](#)

Wajib: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## ClipTimestampRange

Layanan: Amazon Kinesis Video Streams Archived Media

Kisaran cap waktu untuk mengembalikan fragmen.

### Daftar Isi

#### EndTimeStamp

Akhir rentang timestamp untuk media yang diminta.

Nilai ini harus dalam waktu 24 jam dari yang ditentukan `StartTimeStamp`, dan harus lebih lambat dari `StartTimeStamp` nilainya. Jika `FragmentSelectorType` untuk permintaan ini `SERVER_TIMESTAMP`, nilai ini harus di masa lalu.

Nilai ini inklusif. Dibandingkan dengan (mulai) timestamp fragmen. `EndTimeStamp` Fragmen yang dimulai sebelum `EndTimeStamp` nilai dan terus melewati itu termasuk dalam sesi.

Tipe: Timestamp

Wajib: Ya

#### StartTimeStamp

Stempel waktu awal dalam kisaran stempel waktu untuk mengembalikan fragmen.

Hanya fragmen yang dimulai tepat pada atau setelah `StartTimeStamp` dimasukkan dalam sesi. Fragmen yang dimulai sebelum `StartTimeStamp` dan berlanjut melewati itu tidak termasuk dalam sesi. Jika `FragmentSelectorType` ya `SERVER_TIMESTAMP`, `StartTimeStamp` harus lebih lambat dari kepala sungai.

Tipe: Timestamp

Wajib: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu SDK di salah satu AWS SDK di salah satu SDK ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)



- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## DASHFragmentSelector

Layanan: Amazon Kinesis Video Streams Archived Media

Berisi rentang cap waktu untuk media yang diminta, dan sumber stempel waktu.

Daftar Isi

### FragmentSelectorType

Sumber cap waktu untuk media yang diminta.

Kapan `FragmentSelectorType` diatur ke `PRODUCER_TIMESTAMP` dan [StreamingSessionURL getDash: PlaybackMode](#) adalah `ON_DEMAND` atau `LIVE_REPLAY`, fragmen pertama yang dicerna dengan stempel waktu produser dalam yang `TimestampRange` ditentukan [FragmentSelector:](#) disertakan dalam daftar putar media. Selain itu, fragmen dengan stempel waktu produser dalam `TimestampRange` tertelan segera setelah fragmen pertama (hingga nilai [StreamingSessionURL getDash: MaxManifestFragmentResults](#)) disertakan.

Fragmen yang memiliki stempel waktu produser duplikat dideduplikasi. Ini berarti bahwa jika produser memproduksi aliran fragmen dengan stempel waktu produser yang kira-kira sama dengan waktu clock sebenarnya, manifes MPEG-DASH akan berisi semua fragmen dalam rentang stempel waktu yang diminta. Jika beberapa fragmen tertelan dalam rentang waktu yang sama dan titik waktu yang sangat berbeda, hanya koleksi fragmen tertelan tertua yang dikembalikan.

Ketika `FragmentSelectorType` diatur ke `PRODUCER_TIMESTAMP` dan [getDash StreamingSession URL: PlaybackMode](#) adalah `LIVE`, cap waktu produser digunakan dalam fragmen MP4 dan untuk deduplikasi. Tetapi fragmen yang paling baru dicerna berdasarkan stempel waktu server disertakan dalam manifes MPEG-DASH. Ini berarti bahwa bahkan jika fragmen yang tertelan di masa lalu memiliki stempel waktu produser dengan nilai sekarang, mereka tidak termasuk dalam daftar putar media HLS.

Default-nya adalah `SERVER_TIMESTAMP`.

Jenis: String

Nilai yang Valid: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Wajib: Tidak

### TimestampRange

Mulai dan akhir rentang timestamp untuk media yang diminta.

Nilai ini tidak harus hadir PlaybackType jika LIVE.

Tipe: Objek [DASHTimestampRange](#)

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## DASHTimestampRange

Layanan: Amazon Kinesis Video Streams Archived Media

Mulai dan akhir rentang timestamp untuk media yang diminta.

Nilai ini tidak harus hadir `PlaybackType` jika `LIVE`.

Nilai-nilai di `DASHTimestampRange` bersifat inklusif. Fragmen yang dimulai tepat pada atau setelah waktu mulai disertakan dalam sesi. Fragmen yang dimulai sebelum waktu mulai dan terus melewatinya tidak termasuk dalam sesi.

### Daftar Isi

#### EndTimeStamp

Akhir rentang timestamp untuk media yang diminta. Nilai ini harus dalam waktu 24 jam dari yang ditentukan `StartTimeStamp`, dan harus lebih lambat dari `StartTimeStamp` nilainya.

Jika `FragmentSelectorType` untuk permintaan ini `SERVER_TIMESTAMP`, nilai ini harus di masa lalu.

`EndTimeStamp` Nilai diperlukan untuk `ON_DEMAND` mode, tetapi opsional untuk `LIVE_REPLAY` mode. Jika tidak `EndTimeStamp` diatur untuk `LIVE_REPLAY` mode maka sesi akan terus menyertakan fragmen baru tertelan sampai sesi berakhir.

#### Note

Nilai ini inklusif. Dibandingkan dengan (mulai) timestamp fragmen. `EndTimeStamp` Fragmen yang dimulai sebelum `EndTimeStamp` nilai dan terus melewati itu termasuk dalam sesi.

Tipe: Timestamp

Wajib: Tidak

#### StartTimeStamp

Awal rentang stempel waktu untuk media yang diminta.

Jika `DASHTimestampRange` nilainya ditentukan, `StartTimeStamp` nilainya diperlukan.

Hanya fragmen yang dimulai tepat pada atau setelah `StartTimestamp` dimasukkan dalam sesi. Fragmen yang dimulai sebelum `StartTimestamp` dan berlanjut melewati itu tidak termasuk dalam sesi. Jika `FragmentSelectorType` ya `SERVER_TIMESTAMP`, `StartTimestamp` harus lebih lambat dari kepala sungai.

Tipe: Timestamp

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK for bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Fragment

Layanan: Amazon Kinesis Video Streams Archived Media

Merupakan segmen video atau data lain yang dibatasi waktu.

Daftar Isi

### FragmentLengthInMilliseconds

Durasi pemutaran atau nilai waktu lain yang terkait dengan fragmen.

Tipe: Panjang

Diperlukan: Tidak

### FragmentNumber

Pengidentifikasi unik dari fragmen. Nilai ini meningkat secara monoton berdasarkan urutan konsumsi.

Tipe: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola:  $^{[0-9]}+\$$

Diperlukan: Tidak

### FragmentSizeInBytes

Ukuran fragmen total, termasuk informasi tentang fragmen dan data media yang terkandung.

Tipe: Panjang

Diperlukan: Tidak

### ProducerTimestamp

Stempel waktu dari produsen sesuai dengan fragmen, dalam milidetik.

Tipe: Timestamp

Diperlukan: Tidak

### ServerTimestamp

Stempel waktu dari AWS server yang sesuai dengan fragmen, dalam milidetik.

Tipe: Timestamp

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#).
- [AWSSDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

## FragmentSelector

Layanan: Amazon Kinesis Video Streams Archived Media

Menjelaskan rentang timestamp dan asal stempel waktu dari berbagai fragmen.

Hanya fragmen dengan timestamp awal dari atau sama dengan waktu mulai yang diberikan dan kurang dari atau sama dengan waktu akhir yang dikembalikan. Misalnya, jika aliran berisi fragmen dengan stempel waktu awal berikut:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

Rentang pemilih fragmen dengan waktu mulai 00:00:01 dan waktu akhir 00:00:04 akan mengembalikan fragmen dengan waktu mulai 00:00:02 dan 00:00:04.

### Daftar Isi

#### FragmentSelectorType

Asal stempel waktu untuk digunakan (Server atau Produser).

Jenis: String

Nilai yang Valid: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

Wajib: Ya

#### TimestampRange

Kisaran cap waktu untuk kembali.

Tipe: Objek [TimestampRange](#)

Wajib: Ya

### Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:



- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK. V2 V2 V2 V2 V2](#)
- [AWSSDK for Ruby V3 V3 V3 V3 V3 V3 V3 V](#)

## HLSFragmentSelector

Layanan: Amazon Kinesis Video Streams Archived Media

Berisi rentang cap waktu untuk media yang diminta, dan sumber stempel waktu.

Daftar Isi

### FragmentSelectorType

Sumber stempel waktu untuk media yang diminta.

Kapan `FragmentSelectorType` disetel ke `PRODUCER_TIMESTAMP` dan [StreamingSessionURL getHLS: PlaybackMode](#) adalah `ON_DEMAND` atau `LIVE_REPLAY`, fragmen pertama yang dicerna dengan stempel waktu produser dalam yang `TimestampRange` ditentukan [FragmentSelector:](#) disertakan dalam daftar putar media. Selain itu, fragmen dengan stempel waktu produser dalam `TimestampRange` tertelan segera setelah fragmen pertama (hingga [StreamingSessionURL getTLS: MaxMediaPlaylistFragmentResults](#) value) disertakan.

Fragmen yang memiliki stempel waktu produser duplikat diduplikasi. Ini berarti bahwa jika produser memproduksi aliran fragmen dengan stempel waktu produser yang kira-kira sama dengan waktu clock yang sebenarnya, playlist media HLS akan berisi semua fragmen dalam rentang stempel waktu yang diminta. Jika beberapa fragmen tertelan dalam rentang waktu yang sama dan titik waktu yang sangat berbeda, hanya koleksi fragmen tertelan tertua yang dikembalikan.

Ketika `FragmentSelectorType` diatur ke `PRODUCER_TIMESTAMP` dan [getHLS StreamingSession URL: PlaybackMode](#) adalah `LIVE`, stempel waktu produser digunakan dalam fragmen MP4 dan untuk deduplikasi. Tetapi fragmen yang paling baru dicerna berdasarkan stempel waktu server disertakan dalam daftar putar media HLS. Ini berarti bahwa bahkan jika fragmen yang tertelan di masa lalu memiliki stempel waktu produser dengan nilai sekarang, mereka tidak termasuk dalam daftar putar media HLS.

Default-nya adalah `SERVER_TIMESTAMP`.

Jenis: String

Nilai yang Valid: `PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

Wajib: Tidak

### TimestampRange

Mulai dan akhir rentang timestamp untuk media yang diminta.

Nilai ini tidak harus hadir PlaybackType jika LIVE.

Tipe: Objek [HLSTimestampRange](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## HLSTimestampRange

Layanan: Amazon Kinesis Video Streams Archived Media

Mulai dan akhir rentang timestamp untuk media yang diminta.

Nilai ini tidak harus hadir `PlaybackType` jika `LIVE`.

### Daftar Isi

#### EndTimeStamp

Akhir rentang timestamp untuk media yang diminta. Nilai ini harus dalam waktu 24 jam dari yang ditentukan `StartTimeStamp`, dan harus lebih lambat dari `StartTimeStamp` nilainya.

Jika `FragmentSelectorType` untuk permintaan ini `SERVER_TIMESTAMP`, nilai ini harus di masa lalu.

`EndTimeStamp` Nilai diperlukan untuk `ON_DEMAND` mode, tetapi opsional untuk `LIVE_REPLAY` mode. Jika tidak `EndTimeStamp` diatur untuk `LIVE_REPLAY` mode maka sesi akan terus menyertakan fragmen baru tertelan sampai sesi berakhir.

#### Note

Nilai ini bersifat inklusif. Dibandingkan dengan (mulai) timestamp fragmen. `EndTimeStamp` Fragmen yang dimulai sebelum `EndTimeStamp` nilai dan terus melewati itu termasuk dalam sesi.

Tipe: Timestamp

Wajib: Tidak

#### StartTimeStamp

Awal rentang stempel waktu untuk media yang diminta.

Jika `HLSTimestampRange` nilainya ditentukan, `StartTimeStamp` nilainya diperlukan.

Hanya fragmen yang dimulai tepat pada atau setelah `StartTimeStamp` dimasukkan dalam sesi. Fragmen yang dimulai sebelum `StartTimeStamp` dan berlanjut melewati itu tidak termasuk dalam sesi. Jika `FragmentSelectorType` ya `SERVER_TIMESTAMP`, `StartTimeStamp` harus lebih lambat dari kepala sungai.

Tipe: Timestamp

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Image

Layanan: Amazon Kinesis Video Streams Archived Media

Sebuah struktur yang berisi `Timestamp`, `Error`, dan `ImageContent`.

### Daftar Isi

#### Error

Pesan kesalahan yang ditampilkan saat gambar untuk stempel waktu yang disediakan tidak diekstraksi karena kesalahan yang tidak dapat dicoba. Kesalahan akan dikembalikan jika:

- Tidak ada media yang ada untuk yang ditentukan `Timestamp`.
- Media untuk waktu yang ditentukan tidak memungkinkan gambar untuk diekstraksi. Dalam hal ini media hanya audio, atau media yang salah telah dicerna.

Jenis: String

Nilai yang Valid: `NO_MEDIA` | `MEDIA_ERROR`

Wajib: Tidak

#### ImageContent

Atribut dari Image objek yang Base64 dikodekan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 6291456.

Wajib: Tidak

#### TimeStamp

Atribut dari Image objek yang digunakan untuk mengekstrak gambar dari aliran video. Bidang ini digunakan untuk mengelola kesenjangan pada gambar atau untuk lebih memahami jendela pagination.

Tipe: Timestamp

Wajib: Tidak

## Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## TimestampRange

Layanan: Amazon Kinesis Video Streams Archived Media

Kisaran cap waktu untuk mengembalikan fragmen.

Daftar Isi

EndTimeStamp

Stempel waktu akhir dalam kisaran cap waktu untuk mengembalikan fragmen.

Tipe: Timestamp

Wajib: Ya

StartTimeStamp

Stempel waktu awal dalam kisaran cap waktu untuk mengembalikan fragmen.

Tipe: Timestamp

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API di salah satu AWS SDK khusus bahasa, lihat yang berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Amazon Kinesis Video

tipe data berikut didukung Amazon Kinesis Channaling Video

- [IceServer](#)



## IceServer

Layanan: Amazon Kinesis Video Signaling Channels

Struktur untuk data koneksi server ICE.

Daftar Isi

Password

Kata sandi untuk masuk ke server ICE.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.-]+

Wajib: Tidak

Ttl

Periode waktu, dalam detik, di mana nama pengguna dan kata sandi valid.

Tipe: Bilangan Bulat

Rentang yang Valid: Nilai minimum 30. Nilai maksimum 86400.

Wajib: Tidak

Uris

Array URI, dalam bentuk yang ditentukan dalam [I-D. petithuguenin-behave-turn-urisspesifikasi](#). URI ini menyediakan alamat dan/atau protokol yang berbeda yang dapat digunakan untuk mencapai server TURN.

Tipe: Array string

Panjang Batasan: Panjang minimum 1. Panjang maksimum 256.

Wajib: Tidak

Username

Nama pengguna untuk masuk ke server ICE.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: [a-zA-Z0-9\_.- ]+

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- [AWSSDK for C++](#)
- [AWSSDK for Go](#)
- [AWSSDK for Java V2](#)
- [AWSSDK for Ruby V3](#)

## Penyimpanan Video WebRTC Amazon

tipe data berikut didukung oleh Amazon Kinesis Video WebRTC Amazon:

## Kesalahan Umum

Bagian ini berisi daftar kesalahan yang umum terjadi pada tindakan API dari semua layanan AWS. Untuk kesalahan khusus pada tindakan API untuk layanan ini, lihat topik untuk tindakan API tersebut.

### AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

### IncompleteSignature

Tanda tangan permintaan tidak sesuai dengan standar AWS.

Kode Status HTTP: 400

### InternalFailure

Pemrosesan permintaan telah gagal karena kesalahan yang tidak diketahui, pengecualian atau kegagalan.

Kode Status HTTP: 500

InvalidAction

Tindakan atau operasi yang diminta tidak valid. Verifikasi bahwa tindakan diketik dengan benar.

Kode Status HTTP: 400

InvalidClientId

Sertifikat X.509 atau access key ID AWS yang diberikan tidak ada dalam catatan kami.

Kode Status HTTP: 403

NotAuthorized

Anda tidak memiliki izin untuk melakukan tindakan ini.

Kode Status HTTP: 400

OptInRequired

Access key ID AWS membutuhkan berlangganan untuk layanan.

Kode Status HTTP: 403

RequestExpired

Permintaan menjangkau layanan lebih dari 15 menit setelah stempel tanggal pada permintaan atau lebih dari 15 menit setelah tanggal kedaluwarsa permintaan (seperti untuk URL pre-signed), atau stempel tanggal pada permintaan lebih dari 15 menit di masa mendatang.

Kode Status HTTP: 400

ServiceUnavailable

Permintaan telah gagal karena kegagalan sementara server.

Kode Status HTTP: 503

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

ValidationError

Input gagal untuk memenuhi batasan yang ditentukan oleh layanan AWS.

Kode Status HTTP: 400

## Parameter Umum

Daftar berikut berisi parameter yang digunakan semua tindakan untuk menandatangani permintaan Tanda Tangan Versi 4 dengan string kueri. Setiap parameter khusus tindakan tercantum dalam topik untuk tindakan tersebut. Untuk informasi selengkapnya tentang Signature Versi 4, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

### Action

Tindakan yang harus dilakukan.

Tipe: string

Wajib: Ya

### Version

Versi API yang ditulis dalam permintaan, dinyatakan dalam format HH-BB-TTTT.

Tipe: string

Wajib: Ya

### X-Amz-Algorithm

Algoritme hash yang Anda gunakan untuk membuat tanda tangan permintaan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Nilai yang Valid: AWS4-HMAC-SHA256

Diperlukan: Kondisional

### X-Amz-Credential

Nilai lingkup kredensial, yang merupakan string yang menyertakan access key Anda, tanggal, wilayah yang Anda targetkan, layanan yang Anda minta, dan string penghentian ("aws4\_request"). Nilai dinyatakan dalam format berikut: access\_key/HHBBTTTT/wilayah/layanan/aws4\_request.

Untuk informasi selengkapnya, lihat [Membuat permintaan AWS API yang ditandatangani](#) di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

#### X-Amz-Date

Tanggal yang digunakan untuk membuat tanda tangan. Format harus berupa format dasar ISO 8601 (YYYYMMDD'T'HMMSS'Z'). Misalnya, waktu tanggal berikut adalah nilai X-Amz-Date yang valid: 20120325T120000Z.

Syarat: X-Amz-Date bersifat opsional untuk semua permintaan; nilai ini dapat digunakan untuk mengganti tanggal yang digunakan untuk menandatangani permintaan. Jika header Tanggal ditentukan dalam format dasar ISO 8601, X-Amz-Date tidak diperlukan. Ketika X-Amz-Date digunakan, ia selalu mengganti nilai header Tanggal. Untuk informasi selengkapnya, lihat [Elemen tanda tangan permintaan AWS API](#) di Panduan Pengguna IAM.

Tipe: string

Diperlukan: Kondisional

#### X-Amz-Security-Token

Token keamanan sementara yang diperoleh melalui panggilan ke AWS Security Token Service (AWS STS). Untuk daftar layanan yang mendukung kredensial keamanan sementara dari AWS STS, lihat [Layanan AWS bahwa bekerja dengan IAM](#) dalam Panduan Pengguna IAM.

Syarat: Jika Anda menggunakan kredensial keamanan sementara dari AWS STS, Anda harus menyertakan token keamanan.

Tipe: string

Diperlukan: Kondisional

#### X-Amz-Signature

Menentukan tanda tangan yang dikodekan oleh hex yang dihitung dari string to sign dan kunci penandatanganan turunan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

#### X-Amz-SignedHeaders

Menentukan semua header HTTP yang disertakan sebagai bagian dari permintaan kanonik.

Untuk informasi selengkapnya tentang menentukan header yang ditandatangani, lihat [Membuat permintaan AWS API yang ditandatangani](#) di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.