



Panduan Pengembang WebRTC Amazon Kinesis Video Streams

Kinesis Video Streams



Kinesis Video Streams: Panduan Pengembang WebRTC Amazon Kinesis Video Streams

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 Amazon Kinesis Video Streams dengan WebRTC	1
Ketersediaan Wilayah	2
Kinesis Video Streams dengan Harga WebRTC	3
Mengakses Kinesis Video Streams dengan WebRTC	3
Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya	4
Arus Amazon Kinesis Video Streams Amazon dengan Konsep WebRTC Amazon Kinesis	4
Konsep Teknologi WebRTC	5
Bagaimana STUN, TURN dan ICE Bekerja Bersama	6
Kinesis Video Streams dengan Komponen WebRTC	7
WebRTC WebSocket API	8
ConnectAsViewer	9
ConnectAsMaster	11
SendSdpOffer	13
SendSdpAnswer	15
SendIceCandidate	17
Memutuskan	20
Penerimaan Pesan Asinkron	20
Kuota	23
Control Plane API Service Quotas	23
Signaling API Service Quotas	25
TURN Service Quotas	26
WebRTC Konsumsi Service Quotas	27
Memulai	28
Mengatur sebuah Akun AWS	28
Mendaftar untuk Akun AWS	28
Membuat pengguna administratif	29
Buat Kunci AWS Akun	30
Buat Saluran Pensinyalan	30
Membuat Saluran Pensinyalan Menggunakan Konsol	30
Streaming Media Langsung	31
WebRTC SDK di C untuk Perangkat Tertanam	31
WebRTC SDK di JavaScript	34
WebRTC SDK for Android	39
WebRTC SDK untuk iOS	46

Metrik Klien untuk WebRTC C SDK	54
WebRTC menelan	77
Operasi API	77
Memulai dengan konsumsi	78
Buat saluran pensinyalan	78
Buat saluran pensinyalan menggunakan AWS Management Console	78
Buat saluran pensinyalan menggunakan AWS CLI	78
Buat aliran	79
Buat aliran menggunakan AWS Management Console	79
Buat aliran menggunakan AWS CLI	79
Konfigurasi konsumsi dan penyimpanan media	79
Menelan media	80
Menelan media dari browser	80
Menelan media dari WebRTC C SDK	81
Lihat media	82
Keamanan	83
Mengontrol Akses ke Kinesis Video Streams dengan Menggunakan Sumber Daya WebRTCAWS Identity and Access Management	84
Sintaks Kebijakan	84
Tindakan untuk Kinesis Video Streams dengan WebRTC	85
Nama Sumber Daya Amazon (ARN) untuk Kinesis Video Streams	86
Memberikan Akses Akun IAM Lainnya ke Aliran Video Kinesis	86
Contoh kebijakan	86
Validasi kepatuhan	88
Ketahanan	89
Keamanan Infrastruktur dalam Aliran Video Kinesis dengan WebRTC	90
Praktik Terbaik Keamanan untuk Kinesis Video Streams dengan WebRTC	90
Terapkan akses hak istimewa yang paling rendah	90
Gunakan IAM role	91
Gunakan CloudTrail untuk Memantau Panggilan API	91
Memantau	93
Pemantauan Kinesis Video Streams dengan CloudWatch	93
Metrik Sinyal	94
Metrik TURN	95
Mencatat Kinesis Video Streams dengan Panggilan API WebRTC dengan AWS CloudTrail	95
Amazon Kinesis Video Streams dengan WebRTC dan CloudTrail	95

Contoh: Amazon Kinesis Video Streams dengan Entri File Log WebRTC	97
Pemecahan Masalah	99
Kuota layanan	99
Persyaratan jaringan	99
Lingkungan jaringan	100
Masalah membangun sesi	100
Penawaran dan jawaban Protokol Deskripsi Sesi (SDP)	101
Evaluasi generasi kandidat ICE	102
Tentukan kandidat mana yang digunakan untuk menjalin hubungan	104
Debugging koneksi yang sedang berlangsung	105
Riwayat Dokumen	107
AWSGlosarium	108
.....	cix

Apa itu Amazon Kinesis Video Streams dengan WebRTC

WebRTC adalah spesifikasi teknologi terbuka untuk memungkinkan komunikasi real-time (RTC) di seluruh browser dan aplikasi seluler melalui API sederhana. Ini menggunakan teknik peering untuk pertukaran data real-time antara rekan-rekan yang terhubung dan menyediakan streaming media latensi rendah yang diperlukan untuk human-to-human interaksi. Spesifikasi WebRTC mencakup seperangkat protokol IETF [termasuk Interactive Connectivity Establishment](#), [Traversal Using Relay around NAT \(TURN\)](#), dan [Session Traversal Utilities for NAT \(STUN\)](#) untuk peer-to-peer membangun konektivitas, selain spesifikasi protokol untuk media real-time dan streaming data yang andal dan aman.

[Amazon Kinesis Video](#) Streams menyediakan implementasi WebRTC yang sesuai standar sebagai kemampuan yang dikelola sepenuhnya. Anda dapat menggunakan Amazon Kinesis Video Streams dengan WebRTC untuk media streaming langsung dengan aman atau melakukan interaksi audio atau video dua arah antara perangkat IoT kamera apa pun dan pemutar seluler atau web yang sesuai dengan WebRTC. Sebagai kemampuan yang dikelola sepenuhnya, Anda tidak perlu membangun, mengoperasikan, atau menskalakan infrastruktur cloud terkait WebRTC apa pun, seperti server pensinyalan atau relai media untuk mengalirkan media secara aman di seluruh aplikasi dan perangkat.

Menggunakan Kinesis Video Streams dengan WebRTC, Anda dapat dengan mudah membangun peer-to-peer aplikasi untuk streaming media langsung, atau interaktivitas audio atau video real-time antara perangkat IoT kamera, browser web, dan perangkat seluler untuk berbagai kasus penggunaan. Aplikasi semacam itu dapat membantu orang tua mengawasi kamar bayi mereka, memungkinkan pemilik rumah menggunakan bel pintu video untuk memeriksa siapa yang ada di pintu, memungkinkan pemilik penyedot debu robot yang diaktifkan kamera untuk mengontrol robot dari jarak jauh dengan melihat aliran kamera langsung di ponsel, dan sebagainya.

Jika Anda pengguna pertama kali Kinesis Video Streams dengan WebRTC, kami sarankan Anda membaca bagian berikut:

- [Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya](#)
- [WebRTC SDK di C untuk Perangkat Tertanam](#)
- [Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web](#)
- [WebRTC SDK for Android](#)
- [WebRTC SDK untuk iOS](#)

- [Kontrol API pesawat](#)
- [API REST pesawat data](#)
- [API WebSocket bidang data](#)

Ketersediaan Wilayah

Amazon Kinesis Video Streams dengan WebRTC 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
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3

Nama Wilayah	AWS Kode Wilayah
Amerika Selatan (Sao Paulo)	sa-east-1

Kinesis Video Streams dengan Harga WebRTC

Untuk informasi tentang Kinesis Video Streams dengan harga WebRTC, [lihat](#) Harga Amazon Kinesis Video Streams.

Mengakses Kinesis Video Streams dengan WebRTC

Anda dapat bekerja dengan Kinesis Video Streams dengan WebRTC dengan salah satu cara berikut:

AWS Management Console

[Memulai dengan AWS Management Console](#)

Konsol adalah antarmuka berbasis browser untuk mengakses dan menggunakan AWS layanan, termasuk Kinesis Video Streams dengan WebRTC.

AWS SDK

AWS menyediakan perangkat pengembangan perangkat lunak (SDK) yang terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman (misalnya, Java, Python, Ruby, .NET, iOS, Android, dan lainnya). SDK menyediakan cara mudah untuk membuat akses terprogram ke Kinesis Video Streams dengan WebRTC. Untuk informasi tentang AWS SDK, termasuk cara mengunduh dan menginstalnya, lihat [Alat untuk Amazon Web Services](#).

Kinesis Video Streams dengan WebRTC HTTPS API

Anda dapat mengakses Kinesis Video Streams dengan AWS WebRTC dan secara terprogram menggunakan Kinesis Video Streams dengan WebRTC API, yang memungkinkan Anda mengeluarkan permintaan API langsung ke layanan. Untuk informasi selengkapnya, lihat [Referensi API Amazon Kinesis Video Streams](#).

Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya

Topik

- [Arus Amazon Kinesis Video Streams Amazon dengan Konsep WebRTC Amazon Kinesis](#)
- [Konsep Teknologi WebRTC](#)
- [Bagaimana STUN, TURN dan ICE Bekerja Bersama](#)
- [Kinesis Video Streams dengan Komponen WebRTC](#)
- [WebRTC WebSocket API](#)

Arus Amazon Kinesis Video Streams Amazon dengan Konsep WebRTC Amazon Kinesis

Berikut ini adalah istilah dan konsep utama yang spesifik untuk Amazon Kinesis Video Streams dengan WebRTC.

Saluran pensinyalan

Sumber daya yang memungkinkan aplikasi untuk menemukan, mengatur, mengontrol, dan mengakhiri a peer-to-peer koneksi dengan bertukar pesan sinyal. Pesan pensinyalan adalah metadata yang ditukar dua aplikasi satu sama lain untuk dibuat peer-to-peer konektivitas. Metadata ini mencakup informasi media lokal, seperti codec media dan parameter codec, dan kemungkinan jalur kandidat jaringan untuk kedua aplikasi untuk terhubung satu sama lain untuk streaming langsung.

Aplikasi streaming dapat mempertahankan konektivitas persisten dengan saluran pensinyalan dan menunggu aplikasi lain terhubung dengannya. Atau, mereka dapat terhubung ke saluran pensinyalan hanya ketika mereka membutuhkan media streaming langsung. Saluran pensinyalan memungkinkan aplikasi untuk terhubung satu sama lain dalam a one-to-few model, menggunakan konsep satumaster menghubungkan ke beberapa pemirsa. Aplikasi yang memulai koneksi mengasumsikan tanggung jawab master menggunakan `ConnectAsMasterAPI` dan menunggu pemirsa. Hingga 10 aplikasi kemudian dapat terhubung ke saluran pensinyalan itu dengan mengasumsikan tanggung jawab pemirsa dengan menerapkan `ConnectAsViewerAPI`. Setelah terhubung ke saluran pensinyalan, aplikasi master dan penampil dapat saling mengirim pesan pensinyalan untuk dibuat peer-to-peer konektivitas untuk streaming media langsung.

Peer

Perangkat atau aplikasi apa pun (misalnya, aplikasi seluler atau web, webcam, kamera keamanan rumah, monitor bayi, dll.) Yang dikonfigurasi untuk streaming dua arah secara real-time melalui Kinesis Video Streams dengan WebRTC.

Tuan

Peer yang memulai koneksi dan terhubung ke saluran pensinyalan dengan kemampuan untuk menemukan dan bertukar media dengan salah satu pemirsa yang terhubung dengan saluran sinyal.

Important

Saat ini, saluran pensinyalan hanya dapat memiliki satu master.

Orang yang melihat

Rekan yang terhubung ke saluran pensinyalan dengan kemampuan untuk menemukan dan bertukar media hanya dengan master saluran pensinyalan. Penonton tidak dapat menemukan atau berinteraksi dengan penonton lain melalui saluran pensinyalan tertentu. Saluran pensinyalan dapat memiliki maksimal 10 pemirsa yang terhubung.

Konsep Teknologi WebRTC

Saat Anda memulai dengan Kinesis Video Streams dengan WebRTC, Anda juga bisa mendapatkan keuntungan dari mempelajari beberapa protokol dan API yang saling terkait yang terdiri dari teknologi WebRTC.

Utilitas Traversal Sesi untuk NAT (STUN)

Protokol yang digunakan untuk menemukan alamat publik Anda dan menentukan batasan apa pun di router Anda yang akan mencegah koneksi langsung dengan rekan.

Traversal Menggunakan Relay di sekitar NAT (TURN)

Server yang digunakan untuk melewati pembatasan NAT Symmetric dengan membuka koneksi dengan server TURN dan menyampaikan semua informasi melalui server itu.

Protokol Deskripsi Sesi (SDP)

Standar untuk menggambarkan konten multimedia koneksi seperti resolusi, format, codec, enkripsi, dll. Sehingga kedua rekan dapat memahami satu sama lain setelah data ditransfer.

Penawaran SDP

Pesan SDP yang dikirim oleh agen yang menghasilkan deskripsi sesi untuk membuat atau memodifikasi sesi. Ini menggambarkan aspek komunikasi media yang diinginkan.

Jawaban SDP

Pesan SDP yang dikirim oleh penjawab sebagai tanggapan atas penawaran yang diterima dari penawaran. Jawabannya menunjukkan aspek yang diterima. Misalnya, jika semua aliran audio dan video dalam penawaran diterima.

Pembentukan Konektivitas Interaktif (ICE)

Kerangka kerja yang memungkinkan browser web Anda terhubung dengan teman sebaya.

Kandidat ICE

Sebuah metode yang mengirim peer dapat digunakan untuk berkomunikasi.

Bagaimana STUN, TURN dan ICE Bekerja Bersama

Mari kita mengambil skenario dua rekan-rekan, A dan B, yang keduanya menggunakan WebRTC peer untuk peer dua arah media streaming (misalnya, aplikasi video chat). Apa yang terjadi ketika A ingin menelepon B?

Untuk terhubung ke aplikasi B, aplikasi A harus menghasilkan penawaran SDP. Penawaran SDP berisi informasi tentang sesi yang ingin dibuat oleh aplikasi A, termasuk codec apa yang akan digunakan, apakah ini sesi audio atau video, dll. Ini juga berisi daftar kandidat ICE, yang merupakan pasangan IP dan port yang dapat digunakan aplikasi B untuk terhubung ke A.

Untuk membangun daftar kandidat ICE, aplikasi A membuat serangkaian permintaan ke server STUN. Server mengembalikan alamat IP publik dan pasangan port yang berasal permintaan. Aplikasi A menambahkan setiap pasangan ke daftar kandidat ICE, dengan kata lain, ia mengumpulkan kandidat ICE. Setelah aplikasi A selesai mengumpulkan kandidat ICE, ia dapat mengembalikan SDP.

Selanjutnya, aplikasi A harus meneruskan SDP ke aplikasi B melalui saluran pensinyalan tempat aplikasi ini berkomunikasi. Protokol transport untuk pertukaran ini tidak ditentukan dalam standar

WebRTC. Hal ini dapat dilakukan melalui HTTPS, aman WebSocket, atau protokol komunikasi lainnya.

Sekarang, aplikasi B harus menghasilkan jawaban SDP. Aplikasi B mengikuti langkah yang sama A yang digunakan pada langkah sebelumnya: mengumpulkan kandidat ICE, dll. Aplikasi B kemudian perlu mengembalikan jawaban SDP ini ke aplikasi A.

Setelah A dan B bertukar SDP, mereka kemudian melakukan serangkaian pemeriksaan konektivitas. Algoritma ICE di setiap aplikasi mengambil pasangan IP/port kandidat dari daftar yang diterima di SDP pihak lain, dan mengirimkannya permintaan STUN. Jika respons datang kembali dari aplikasi lain, aplikasi asal menganggap cek berhasil dan menandai bahwa pasangan IP/port sebagai kandidat ICE valid.

Setelah pemeriksaan konektivitas selesai pada semua pasangan IP/port, aplikasi bernegosiasi dan memutuskan untuk menggunakan salah satu pasangan yang tersisa dan valid. Ketika pasangan dipilih, media mulai mengalir di antara aplikasi.

Jika salah satu aplikasi tidak dapat menemukan pasangan IP/port yang melewati pemeriksaan konektivitas, mereka akan membuat permintaan STUN ke server TURN untuk mendapatkan alamat relay media. Alamat relay adalah alamat IP publik dan port yang meneruskan paket yang diterima ke dan dari aplikasi untuk mengatur alamat relay. Alamat relai ini kemudian ditambahkan ke daftar kandidat dan ditukar melalui saluran pensinyalan.

Kinesis Video Streams dengan Komponen WebRTC

Kinesis Video Streams dengan WebRTC mencakup komponen-komponen berikut:

- Bidang Kontrol

Komponen control plane bertanggung jawab untuk membuat dan memelihara Kinesis Video Streams dengan saluran sinyal WebRTC. Untuk informasi selengkapnya, lihat [Referensi API Amazon Kinesis Video Streams Amazon Kinesis](#).

- Sinyal

Komponen pensinyalan mengelola titik akhir pensinyalan WebRTC yang memungkinkan aplikasi terhubung dengan aman satu sama lain untuk peer-to-peer streaming media langsung. Komponen pensinyalan mencakup [API SISA Sinyal Amazon Kinesis Video Kinesis Amazon Kinesis Amazon](#) dan satu set [API WebSocket](#).

- SETRUM

Komponen ini mengelola titik akhir STUN yang memungkinkan aplikasi untuk menemukan alamat IP publik mereka ketika mereka berada di belakang NAT atau firewall.

- BELOK

Komponen ini mengelola titik akhir TURN yang mengaktifkan relai media melalui cloud saat aplikasi tidak dapat melakukan streaming media peer-to-peer.

- SDK WebRTC Kinesis

Ini adalah pustaka perangkat lunak yang dapat Anda unduh, instal, dan konfigurasi pada perangkat dan klien aplikasi Anda untuk mengaktifkan perangkat IoT kamera Anda dengan kemampuan WebRTC untuk terlibat dalam latensi rendah peer-to-peer Arus media. SDK ini juga memungkinkan klien Android, iOS, dan aplikasi web untuk mengintegrasikan Kinesis Video Streams dengan kemampuan pensinyalan, TURN, dan STUN WebRTC dengan pemutar seluler atau web yang sesuai dengan WebRTC.

- [WebRTC SDK di C untuk Perangkat Tertanam](#)
- [Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web](#)
- [WebRTC SDK for Android](#)
- [WebRTC SDK untuk iOS](#)

WebRTC Websocket API

Topik

- [ConnectAsViewer](#)
- [ConnectAsMaster](#)
- [SendSdpOffer](#)
- [SendSdpAnswer](#)
- [SendIceCandidate](#)
- [Memutuskan](#)
- [Penerimaan Pesan Asinkron](#)

ConnectAsViewer

Terhubung sebagai penampil ke saluran pensinyalan yang ditentukan oleh titik akhir. WebSocketPustaka apa pun yang sesuai dapat digunakan untuk terhubung ke titik akhir yang diperoleh dari panggilan API. `GetSignalingEndpoint` Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan dan ID klien harus disediakan sebagai parameter string kueri. Ada titik akhir terpisah untuk menghubungkan sebagai master dan sebagai penampil. Jika ada koneksi yang ada dengan yang sama `ClientId` seperti yang ditentukan dalam permintaan, koneksi baru diutamakan. Metadata koneksi ditimpa dengan informasi baru.

Permintaan

```
"X-Amz-ChannelARN": "string",  
"X-Amz-ClientId": "string"
```

- X-AMZ-channelarn - ARN dari saluran pensinyalan.
 - Tipe: string
 - Panjang batasan: Panjang minimum 1. Panjang maksimal 1024
 - Pola: `arn:aws:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`
 - Diperlukan: Ya
- X-Amz-ClientId - Pengidentifikasi unik untuk klien.
 - Tipe: string
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: `^(?!(?i)AWS_.*)[a-zA-Z0-9_.-]`

Note

X-Amz-ClientIdTidak bisa mulai denganAWS_.

- Diperlukan: Ya

Respons

200 OK kode status HTTP dengan badan kosong.

Kesalahan

- `InvalidArgumentException`

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan. Untuk informasi selengkapnya, lihat pesan yang dikembalikan.

Kode Status HTTP: 400

- `AccessDeniedException`

Penelepon tidak berwenang untuk mengakses saluran yang diberikan atau token telah kedaluwarsa.

Kode Status HTTP: 403

- `ResourceNotFoundException`

Channelnya tidak ada.

Kode Status HTTP: 404

- `ClientLimitExceededException`

Ketika API dipanggil pada tingkat yang terlalu tinggi atau ketika ada lebih dari jumlah maksimum pemirsa yang didukung yang terhubung ke saluran. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams dengan WebRTC Service Quotas](#) dan [Error Retries dan Exponential Backoff](#) di AWS

Kode Status HTTP: 400

Batas/Pelambatan

API ini dibatasi pada tingkat akun jika API dipanggil pada tingkat yang terlalu tinggi atau ketika jumlah maksimum penonton yang didukung terhubung ke saluran lebih dari jumlah maksimum yang didukung. Kesalahan kembali saat dibatasi dengan `ClientLimitExceededException`

Idempoten

Jika koneksi sudah ada untuk yang ditentukan `ClientId` dan saluran, metadata koneksi diperbarui dengan informasi baru.

Coba lagi perilaku

Ini dihitung sebagai panggilan API baru.

Panggilan bersamaan

Panggilan bersamaan diizinkan, metadata koneksi diperbarui untuk setiap panggilan.

ConnectAsMaster

Terhubung sebagai master ke saluran pensinyalan yang ditentukan oleh titik akhir. Pustaka WebSocket-complaint apa pun dapat digunakan untuk terhubung ke titik akhir yang diperoleh dari panggilan `GetSignalingChannelEndpoint` API. Nama Sumber Daya Amazon (ARN) dari saluran pensinyalan harus disediakan sebagai parameter string kueri. Ada titik akhir terpisah untuk menghubungkan sebagai master dan sebagai penampil. Jika lebih dari satu klien terhubung sebagai master ke saluran tertentu, maka permintaan terbaru diutamakan. Metadata koneksi yang ada ditimpa oleh yang baru.

Permintaan

```
"X-Amz-ChannelARN": "string"
```

- X-AMZ-channelarn - ARN dari saluran pensinyalan.
 - Tipe: string
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 1024.
 - Pola: `arn:aws:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`
 - Diperlukan: Ya

Response

200 OK kode status HTTP dengan badan kosong.

Kesalahan

- `InvalidArgumentException`

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan. Untuk informasi selengkapnya, lihat pesan yang dikembalikan.

Kode Status HTTP: 400

- `AccessDeniedException`

Penelepon tidak berwenang untuk mengakses saluran yang diberikan atau token telah kedaluwarsa.

Kode Status HTTP: 403

- `ResourceNotFoundException`

Channelnya tidak ada.

Kode Status HTTP: 404

- `ClientLimitExceededException`

Ketika API dipanggil pada tingkat yang terlalu tinggi. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams dengan WebRTC Service Quotas](#) dan [Error Retries dan Exponential Backoff](#) di. AWS

Kode Status HTTP: 400

Batas/Pelambatan

API ini dibatasi pada tingkat akun jika API dipanggil pada tingkat yang terlalu tinggi. Kesalahan kembali saat dibatasi dengan. `ClientLimitExceededException`

Idempoten

Jika koneksi sudah ada untuk `clientID` dan saluran yang ditentukan, metadata koneksi diperbarui dengan informasi baru.

Coba lagi perilaku

Ini dihitung sebagai panggilan API baru.

Panggilan bersamaan

Panggilan bersamaan diizinkan, metadata koneksi diperbarui untuk setiap panggilan.

SendSdpOffer

Mengirim penawaran ke penerima target. Prasyaratnya adalah klien harus sudah terhubung ke WebSocket titik akhir yang diperoleh dari API. `GetSignalingChannelEndpoint`

Jika jenis pengirim adalah penampil, maka ia mengirimkan penawaran ke master. Juga, tidak perlu menentukan `RecipientClientId` dan nilai tertentu untuk `RecipientClientId` diabaikan. Jika jenis pengirim adalah master, penawaran dikirim ke penampil target yang ditentukan oleh `RecipientClientId`. `RecipientClientId` adalah masukan yang diperlukan dalam kasus ini.

Aplikasi klien master diizinkan mengirim penawaran ke penampil mana pun, sedangkan aplikasi klien penampil hanya diizinkan mengirim penawaran ke aplikasi klien utama. Jika aplikasi klien penampil mencoba mengirim penawaran ke aplikasi klien penampil lain, permintaan tersebut TIDAK akan dihormati. Jika ada penawaran luar biasa untuk klien yang sama yang belum dikirimkan, itu ditimpa dengan penawaran baru.

Permintaan

```
{
  "action": "SDP_OFFER",
  "recipientClientId": "string",
  "messagePayload": "string",
  "correlationId": "string"
}
```

- `action` - Jenis pesan yang sedang dikirim.
 - Jenis: ENUM
 - Nilai valid: `SDP_OFFER`, `SDP_ANSWER`, `ICE_CANDIDATE`
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: `[a-zA-Z0-9_.-]+`
 - Diperlukan: Ya
- `recipientClientId` - Pengidentifikasi unik untuk penerima.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: `[a-zA-Z0-9_.-]+`
 - Diperlukan: Ya

- **MessagePayload** - Konten pesan yang disandikan basis-64.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 10K.
 - Wajib: Ya
- **CorrelationID** - Sebuah identifier unik untuk pesan. Ini adalah parameter opsional.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Wajib: Tidak

Response

Jika pesan berhasil diterima oleh backend pensinyalan, tidak ada respons yang dikembalikan. Jika layanan mengalami kesalahan dan jika `correlationId` ditentukan dalam permintaan, rincian kesalahan dikembalikan sebagai `STATUS_RESPONSE` pesan. Untuk informasi selengkapnya, lihat [Penerimaan Pesan Asinkron](#).

Kesalahan

- `InvalidArgumentException`

Parameter yang ditentukan melebihi batasannya, tidak didukung, atau tidak dapat digunakan. Untuk informasi selengkapnya, lihat pesan yang dikembalikan.

Kode Status HTTP: 400

- `ClientLimitExceededException`

Ketika API dipanggil pada tingkat yang terlalu tinggi. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams dengan WebRTC Service Quotas](#) dan [Error Retries dan Exponential Backoff](#) di AWS

Kode Status HTTP: 400

Batas/Pelambatan

API ini dibatasi pada tingkat akun jika API dipanggil pada tingkat yang terlalu tinggi. Kesalahan kembali saat dibatasi dengan `ClientLimitExceededException`

Idempoten

API ini tidak idempoten.

Coba lagi perilaku

Ini dihitung sebagai panggilan API baru.

Panggilan bersamaan

Panggilan bersamaan diizinkan. Penawaran dikirim satu kali per setiap panggilan.

SendSdpAnswer

Mengirim jawaban ke penerima target. Prasyaratnya adalah klien harus sudah terhubung ke WebSocket titik akhir yang diperoleh dari API. `GetSignalingChannelEndpoint`

Jika jenis pengirim adalah penampil, maka ia mengirimkan jawaban ke master. Juga, tidak perlu menentukan `RecipientClientId` dan nilai tertentu untuk `RecipientClientId` diabaikan. Jika jenis pengirim adalah master, jawabannya dikirim ke penampil target yang ditentukan oleh `RecipientClientId`. `RecipientClientId` adalah masukan yang diperlukan dalam kasus ini.

Aplikasi klien master diizinkan mengirim jawaban ke penampil mana pun, sedangkan aplikasi klien penampil hanya diizinkan mengirim jawaban ke aplikasi klien utama. Jika aplikasi klien penampil mencoba mengirim jawaban ke aplikasi klien penampil lain, permintaan tersebut TIDAK akan dihormati. Jika ada jawaban yang luar biasa untuk klien yang sama yang belum disampaikan, itu ditimpa dengan jawaban baru.

Permintaan

```
{
  "action": "SDP_ANSWER",
  "recipientClientId": "string",
  "messagePayload": "string",
  "correlationId": "string"
}
```

- `action` - Jenis pesan yang sedang dikirim.
 - Jenis: ENUM

- Nilai valid: SDP_OFFER, SDP_ANSWER, ICE_CANDIDATE
- Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
- Pola: [a-zA-Z0-9_.-]+
- Diperlukan: Ya
- recipientClientId- Pengidentifikasi unik untuk penerima.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Diperlukan: Ya
- MessagePayload - Konten pesan yang disandikan basis-64.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 10K.
 - Wajib: Ya
- CorrelationID - Sebuah identifier unik untuk pesan.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Wajib: Tidak

Response

Tidak ada respons yang dikembalikan jika pesan berhasil diterima oleh backend pensinyalan. Jika layanan mengalami kesalahan dan jika `correlationId` ditentukan dalam permintaan, rincian kesalahan dikembalikan sebagai `STATUS_RESPONSE` pesan. Untuk informasi selengkapnya, lihat [Penerimaan Pesan Asinkron](#).

Kesalahan

- `InvalidArgumentException`

Parameter tertentu melebihi batasannya, tidak didukung, atau tidak dapat digunakan. Untuk informasi selengkapnya, lihat pesan yang dikembalikan.

Kode Status HTTP: 400

- `ClientLimitExceededException`

Dikembalikan ketika API dipanggil pada tingkat yang terlalu tinggi. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams dengan WebRTC Service Quotas](#) dan [Error Retries dan Exponential Backoff](#) di AWS

Kode Status HTTP: 400

Batas/Pelambatan

API ini dibatasi pada tingkat akun jika API dipanggil pada tingkat yang terlalu tinggi. Kesalahan dikembalikan saat dibatasi dengan `ClientLimitExceededException`

Idempoten

API ini tidak idempoten.

Coba lagi perilaku

Ini dihitung sebagai panggilan API baru.

Panggilan bersamaan

Panggilan bersamaan diizinkan. Penawaran dikirim satu kali per setiap panggilan.

SendIceCandidate

Mengirim kandidat ICE ke penerima target. Prasyaratnya adalah klien harus sudah terhubung ke WebSocket titik akhir yang diperoleh dari API `GetSignalingChannelEndpoint`

Jika jenis pengirim adalah penampil, maka ia mengirim kandidat ICE ke master. Juga, tidak perlu menentukan `RecipientClientId` dan nilai tertentu untuk `RecipientClientId` diabaikan. Jika jenis pengirim adalah master, kandidat ICE dikirim ke target yang ditentukan oleh `RecipientClientId`. `RecipientClientId` adalah masukan yang diperlukan dalam kasus ini.

Aplikasi klien master diizinkan mengirim kandidat ICE ke penampil mana pun, sedangkan aplikasi klien penampil hanya diizinkan mengirim kandidat ICE ke aplikasi klien utama. Jika aplikasi klien penampil mencoba mengirim kandidat ICE ke aplikasi klien penampil lain, permintaan tersebut TIDAK akan dihormati.

Permintaan

```
{
  "action": "ICE_CANDIDATE",
  "recipientClientId": "string",
  "messagePayload": "string",
  "correlationId": "string"
}
```

- **action** - Jenis pesan yang sedang dikirim.
 - Jenis: ENUM
 - Nilai valid: SDP_OFFER, SDP_ANSWER, ICE_CANDIDATE
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Diperlukan: Ya
- **recipientClientId**- Pengidentifikasi unik untuk penerima.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Wajib: Tidak
- **MessagePayload** - Konten pesan yang dikodekan base64.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 10K.
 - Wajib: Ya
- **CorrelationID** - Sebuah identifier unik untuk pesan.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Wajib: Tidak

Response

Tidak ada respons yang dikembalikan jika pesan berhasil diterima oleh backend pensinyalan. Jika layanan mengalami kesalahan dan jika `correlationId` ditentukan dalam permintaan, rincian kesalahan dikembalikan sebagai `STATUS_RESPONSE` pesan. Untuk informasi selengkapnya, lihat [Penerimaan Pesan Asinkron](#).

Kesalahan

- `InvalidArgumentException`

Parameter tertentu melebihi batasannya, tidak didukung, atau tidak dapat digunakan. Untuk informasi selengkapnya, lihat pesan yang dikembalikan.

Kode Status HTTP: 400

- `ClientLimitExceededException`

Ketika API dipanggil pada tingkat yang terlalu tinggi. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams dengan WebRTC Service Quotas](#) dan [Error Retries dan Exponential Backoff](#) di AWS

Kode Status HTTP: 400

Batas/Pelambatan

API ini dibatasi pada tingkat akun jika API dipanggil pada tingkat yang terlalu tinggi. Kesalahan kembali saat dibatasi dengan `ClientLimitExceededException`

Idempoten

API ini tidak idempoten.

Coba lagi perilaku

Ini dihitung sebagai panggilan API baru.

Panggilan bersamaan

Panggilan bersamaan diizinkan. Penawaran dikirim satu kali per setiap panggilan.

Memutuskan

Klien dapat menutup koneksi kapan saja. WebSocket pustaka -compliant mendukung fungsionalitas dekat. Ketika koneksi ditutup, layanan menandai klien sebagai offline untuk saluran sinyal tertentu dan tidak mencoba untuk mengirimkan pesan apa pun. Perilaku yang sama juga berlaku jika terjadi koneksi idle batas waktu.

Layanan ini juga mengirimkan indikasi pemutusan ke klien, misalnya, selama penyebaran atau pemeliharaan server. Berikut ini adalah pesan indikasi didefinisikan:

- **GO_AWAY**: Pesan ini digunakan untuk memulai shutdown koneksi. Hal ini memungkinkan klien untuk anggun memproses pesan sebelumnya, memutuskan, dan menyambung kembali ke saluran sinyal jika diperlukan.
- **RECONNECT_ICE_SERVER**: Pesan ini digunakan untuk memulai koneksi relay shutdown dan memungkinkan klien untuk anggun memutuskan, mendapatkan konfigurasi server ICE baru, dan menyambung kembali ke server relay jika diperlukan.

Penerimaan Pesan Asinkron

Semua pesan respons dikirim secara asinkron ke penerima sebagai peristiwa (misalnya, penawaran SDP atau pengiriman jawaban SDP). Berikut ini adalah struktur pesan acara.

Peristiwa

```
{
  "senderClientId": "string",
  "messageType": "string",
  "messagePayload": "string",
  "statusResponse": {
    "correlationId": "string",
    "errorType": "string",
    "statusCode": "string",
    "description": "string"
  }
}
```

- **senderClientId**- Pengidentifikasi unik untuk klien pengirim.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.

- Pola: [a-zA-Z0-9_.-]+
- Diperlukan: Tidak
- MessageType - Jenis acara.
 - Jenis: ENUM
 - Jenis yang Valid:
SDP_OFFERSDP_ANSWER,,ICE_CANDIDATE,GO_AWAY,,RECONNECT_ICE_SERVER,
STATUS_RESPONSE
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Diperlukan: Ya
- MessagePayLoad - Konten pesan yang dikodekan base64.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 10K.
 - Diperlukan: Tidak
- CorrelationID - Pengidentifikasi unik dari pesan yang statusnya dimaksudkan. Ini adalah CorrelationID yang sama yang disediakan dalam pesan klien (misalnya, penawaran SDP, jawaban SDP, atau kandidat ICE).
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Diperlukan: Ya
- ErrorType - Nama untuk mengidentifikasi kesalahan secara unik.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+
 - Diperlukan: Tidak
- StatusCode - kode status HTTP sesuai dengan sifat respon.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 256.
 - Pola: [a-zA-Z0-9_.-]+

- **description** - Deskripsi string yang menjelaskan status.
 - Jenis: String
 - Panjang batasan: Panjang minimum 1. Panjang maksimum 1K.
 - Diperlukan: Tidak

Amazon Kinesis Video Streams dengan WebRTC Service Quotas

Kinesis Video Streams dengan WebRTC 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.

Note

TPS adalah singkatan dari transaksi per detik.

Topik

- [Control Plane API Service Quotas](#)
- [Signaling API Service Quotas](#)
- [TURN Service Quotas](#)
- [WebRTC Konsumsi Service Quotas](#)

Control Plane API Service Quotas

Bagian berikut menjelaskan kuota layanan untuk API bidang kontrol.

API	Kuota layanan akun: Permintaan	Kuota layanan akun: Saluran	Kuota layanan tingkat saluran	Pengecualian dan Catatan yang Relevan
CreateSignalingChannel	50 TPS [s]	us-east-1 dan us-west-2		

API	Kuota layanan akun: Permintaan	Kuota layanan akun: Saluran	Kuota layanan tingkat saluran	Pengecualian dan Catatan yang Relevan
		- 10.000 saluran per akun per wilayah; semua Wilayah lain yang didukung - 5.000 saluran per akun per wilayah		
DeleteSignalingChannel	50 TPS [h]	N/A	5 TPS [h]	
DescribeMediaStorageConfiguration	50 TPS [h]		5 TPS [h]	
DescribeSignalingChannel	300 TPS [h]	N/A	5 TPS [h]	
GetSignalingChannelEndpoint	300 TPS [h]	N/A		
ListSignalingChannels	50 TPS [h]	N/A		

API	Kuota layanan akun: Permintaan	Kuota layanan akun: Saluran	Kuota layanan tingkat saluran	Pengecualian dan Catatan yang Relevan
ListTagsForResource	50 TPS [h]	N/A	5 TPS [h]	
TagResource	50 TPS [h]	N/A	5 TPS [h]	
UntagResource	50 TPS [h]	N/A	5 TPS [h]	
UpdateMediaStorageConfiguration	10 TPS [h]		5 TPS [h]	
UpdateSignalingChannel	50 TPS [h]	N/A	5 TPS [h]	

Signaling API Service Quotas

Bagian berikut menjelaskan kuota layanan untuk komponen pensinyalan di Kinesis Video Streams dengan WebRTC. Untuk informasi selengkapnya, lihat [Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya](#).

- ConnectAsMaster
 - API - 3 TPS per saluran (h)
 - Jumlah maksimum koneksi master per saluran pensinyalan - 1 (jam)
 - Batas durasi koneksi - 1 jam (h)
 - Batas waktu koneksi idle - 10 menit (h)
 - Ketika klien menerima GO_AWAY pesan dari server, koneksi dihentikan setelah masa tenggang 1 menit (h)
- ConnectAsViewer
 - API - 3 TPS per saluran (h)

- Jumlah maksimum koneksi penampil per saluran - 10 (s)
- Batas durasi koneksi - 1 jam (h)
- Batas waktu koneksi idle - 10 menit (h)
- Setelah klien menerima GO_AWAY pesan dari server, koneksi dihentikan setelah masa tenggang 1 menit (h)
- Putuskan sambungan
 - N/A
- GetIceServerConfig
 - API - 5 TPS per saluran pensinyalan (h)
- SendAlexaOfferToMaster
 - API - 5 TPS per saluran pensinyalan (h)
- SendICECandidate
 - API - 20 TPS per WebSocket koneksi (h)
 - Batas ukuran muatan pesan - 10k (h)
- SendSDPAnswer
 - API - 5 TPS per WebSocket koneksi (h)
 - Batas ukuran muatan pesan - 10k (h)
- SendSDPOffer
 - API - 5 TPS per WebSocket koneksi (h)
 - Batas ukuran muatan pesan - 10k (h)

TURN Service Quotas

Bagian berikut menjelaskan kuota layanan untuk Traversal Using Relay around NAT (TURN) komponen di Kinesis Video Streams dengan WebRTC. Untuk informasi selengkapnya, lihat [Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya](#).

- Kecepatan Bit - 5Mbps (h)
- Siklus Hidup Kredensial - 5 menit (h)
- Jumlah alokasi - 50 per saluran pensinyalan (h)

WebRTC Konsumsi Service Quotas

Bagian berikut menjelaskan kuota layanan untuk komponen perekaman media di Amazon Kinesis Video Streams WebRTC. Untuk informasi selengkapnya, lihat [Amazon Kinesis Video Streams konsumsi WebRTC](#).

JoinStorageSession

- API:
 - Per akun - 50 TPS (h)
 - Per saluran - 2 TPS (h)
- Kuota sesi streaming:
 - Bitrate - 1 Mbps
 - Durasi sesi - 1 jam (h)
 - Batas waktu idle - 3 menit (h)

Memulai

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

- Siapkan Akun AWS dan buat administrator.
- Buat Kinesis Video Streams dengan saluran pensinyalan WebRTC.
- Gunakan Kinesis Video Streams dengan SDK WebRTC untuk mengonfigurasi master dan peer-to-peer penampil untuk melakukan streaming video dan audio melalui saluran pensinyalan.

Jika Anda baru mengenal Kinesis Video Streams dengan WebRTC, kami sarankan Anda membaca terlebih dahulu. [Arus Video Kinesis Kinesis dengan WebRTC: Cara Kerjanya](#)

Topik

- [Mengatur AWS Akun dan Membuat Administrator](#)
- [Buat Saluran Pensinyalan](#)
- [Streaming Media Langsung](#)

Mengatur AWS Akun dan Membuat Administrator

Sebelum Anda menggunakan Kinesis Video Streams dengan WebRTC untuk pertama kalinya, selesaikan tugas-tugas berikut:

Topik

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

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 mengirim 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 masukkan 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 Kunci AWS Akun

Anda memerlukan Kunci AWS Akun untuk mengakses Kinesis Video Streams dengan WebRTC secara terprogram.

Untuk membuat AWS Account Key, 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, lalu rekam kunci akses Rahasia.

Buat Saluran Pensinyalan

Anda dapat menggunakan konsol Kinesis Video Streams, [CreateSignalingChannel](#) API () AWS , AWS CLI atau untuk membuat saluran pensinyalan.

Membuat Saluran Pensinyalan Menggunakan Konsol

1. Masuk ke AWS Management Console dan buka konsol [Amazon Kinesis Video Streams](#).
2. Pada halaman Saluran pensinyalan, pilih Buat saluran pensinyalan.

3. Pada halaman Buat saluran pensinyalan baru, ketik nama untuk saluran pensinyalan. Biarkan nilai Time-to-live (Ttl) default 60 detik tidak berubah.
4. Pilih Buat saluran pensinyalan.
5. Setelah saluran pensinyalan dibuat, tinjau detailnya di halaman detail saluran.

Streaming Media Langsung

Kinesis Video Streams dengan WebRTC mencakup SDK berikut:

- [WebRTC SDK di C untuk Perangkat Tertanam](#)
- [Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web](#)
- [WebRTC SDK for Android](#)
- [WebRTC SDK untuk iOS](#)

Setiap SDK menyertakan sampel dan step-by-step instruksi terkait yang dapat membantu Anda membangun dan menjalankan aplikasi tersebut. Anda dapat menggunakan sampel ini untuk latensi rendah, live, streaming audio dan video dua arah dan pertukaran data antara kombinasi aplikasi Web/Android/iOS atau perangkat yang disematkan. Dengan kata lain, Anda dapat melakukan streaming audio dan video langsung dari perangkat kamera yang disematkan ke aplikasi Android atau web atau di antara dua aplikasi Android.

WebRTC SDK di C untuk Perangkat Tertanam

Berikut ini step-by-step petunjuk menjelaskan cara mengunduh, membangun, dan menjalankan Kinesis Video Streams dengan WebRTC SDK di C untuk perangkat yang disematkan dan sampel yang sesuai.

Unduh Aliran Video Kinesis dengan WebRTC SDK di C

Untuk mengunduh Kinesis Video Streams dengan WebRTC SDK di C untuk perangkat yang disematkan, jalankan perintah berikut:

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

Buat Aliran Video Kinesis dengan WebRTC SDK di C

Important

Sebelum Anda menyelesaikan langkah-langkah ini di macOS dan bergantung pada versi macOS yang Anda miliki, Anda harus menjalankannya `xcode-select --install` untuk mengunduh paket dengan alat baris perintah dan header. Kemudian buka `/Library/Developer/CommandLineTools/Packages/macOS_SDK_headers_for_macOS_10.14.pkg` dan ikuti installer untuk menginstal alat baris perintah dan header. Anda hanya perlu melakukan ini sekali dan sebelum memohon `cmake`. Jika Anda sudah menginstal alat baris perintah dan header, Anda tidak perlu menjalankan perintah ini lagi.

Selesaikan langkah-langkah berikut:

1. Instal `cmake`:

- Di macOS, jalankan `brew install cmake pkg-config srt`
- di Ubuntu, jalankan `sudo apt-get install pkg-config cmake libcap2 libcap-dev`

2. Dapatkan kunci akses dan kunci rahasia Akun AWS yang ingin Anda gunakan untuk demo ini.

3. Jalankan perintah berikut untuk membuat `build` direktori di WebRTC C SDK yang Anda unduh, dan jalankan `cmake` dari itu:

```
$ mkdir -p amazon-kinesis-video-streams-webrtc-sdk-c/build; cd amazon-kinesis-video-streams-webrtc-sdk-c/build; cmake ..
```

4. Sekarang bahwa Anda berada di `build` direktori yang baru saja Anda buat dengan langkah di atas, jalankan `make` untuk membangun WebRTC C SDK dan sampel yang disediakan.

Note

Yang `kvswebRTCClientMasterGstSample` TIDAK akan dibangun jika sistem tidak memiliki `gst` terpasang. Untuk memastikannya dibangun (di macOS), Anda harus

```
menjalankan:brew install gstreamer gst-plugins-base gst-plugins-good
```

Jalankan Sampel untuk SDK WebRTC di C

Setelah Anda menyelesaikan prosedur di atas, Anda berakhir dengan contoh aplikasi berikut di builddirektori:

- `kvsWebrtcClientMaster`- Aplikasi ini mengirimkan sampel H264/Opus frame (path: `/samples/h264SampleFrames` dan `/sampel/opusSampleFrames`) melalui saluran pensinyalan. Ini juga menerima audio yang masuk, jika diaktifkan di browser. Saat diperiksa di browser, ia mencetak metadata paket audio yang diterima di terminal Anda.
- `kvsWebrtcClientViewer`- Aplikasi ini menerima sampel H264/Opus frame dan mencetaknya.
- `kvsWebrtcClientMasterGstSample`- Aplikasi ini mengirimkan sampel H264/Opus frame dari pipa GStreamer.

Untuk menjalankan salah satu sampel ini, selesaikan langkah-langkah berikut:

1. Siapkan lingkungan Anda dengan Akun AWS mandat:

```
export AWS_ACCESS_KEY_ID= Your Akun AWS access key
export AWS_SECRET_ACCESS_KEY= Akun AWS secret key
export AWS_KVS_CACERT_PATH= Full path of your cert.pem file. It is typically
available in the certs directory inside
Kinesis-video-webrtc-native-build/certs/cert.pm
```

2. Jalankan salah satu contoh aplikasi dengan meneruskan nama yang ingin Anda berikan ke saluran pensinyalan Anda. Aplikasi membuat saluran pensinyalan menggunakan nama yang Anda berikan. Misalnya, untuk membuat saluran pensinyalan yang disebut `myChannel` dan untuk mulai mengirim sampel H264/Opus frame melalui saluran ini, jalankan perintah berikut:

```
./kvsWebrtcClientMaster myChannel
```

Ketika aplikasi baris perintah mencetak `Connection established`, Anda dapat melanjutkan ke langkah berikutnya.

3. Sekarang setelah saluran pensinyalan Anda dibuat dan master yang terhubung adalah streaming media ke sana, Anda dapat melihat aliran ini. Misalnya, Anda dapat melihat live

stream ini dalam aplikasi web. Untuk melakukannya, buka Halaman Uji SDK WebRTC menggunakan langkah-langkah di [Gunakan Kinesis Video Streams dengan halaman uji WebRTC](#) dan atur nilai berikut menggunakan yang sama AWS kredensi dan saluran pensinyalan yang sama yang Anda tentukan untuk master di atas:

- Access key ID
- Secret access key
- Mensinyalkan nama saluran
- ID Klien (opsional)

Pilih Mulai penampil untuk memulai streaming video langsung dari sampel H264/Opus frame.

Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web

Anda dapat menemukan Kinesis Video Streams dengan JavaScript WebRTC SDK untuk aplikasi web dan sampel yang sesuai di [GitHub](#)

Topik

- [Instal Kinesis Video Streams dengan WebRTC SDK di JavaScript](#)
- [Kinesis Video Streams JavaScript dengan dokumentasi WebRTC SDK](#)
- [Gunakan Kinesis Video Streams dengan halaman uji WebRTC](#)
- [Edit Kinesis Video Streams dengan halaman uji WebRTC](#)

Instal Kinesis Video Streams dengan WebRTC SDK di JavaScript

Apakah dan bagaimana Anda menginstal Kinesis Video Streams dengan JavaScript WebRTC SDK tergantung pada apakah kode dijalankan dalam modul atau skrip browser. Node.js

NodeJS module

[Cara yang lebih disukai untuk menginstal Kinesis Video Streams dengan JavaScript WebRTC SDK untuk Node.js](#) adalah dengan menggunakan npm, pengelola paket Node.js.

Paket ini di-host di <https://www.npmjs.com/package/amazon-kinesis-video-streams-webrtc>.

Untuk menginstal SDK ini di Node . js project Anda, gunakan terminal untuk menavigasi ke direktori yang sama dengan project Anda: package . json

Ketik berikut ini:

```
npm install amazon-kinesis-video-streams-webrtc
```

Anda dapat mengimpor kelas SDK seperti modul Node.js biasa:

```
// JavaScript
const SignalingClient = require('amazon-kinesis-video-streams-
webrtc').SignalingClient;
// TypeScript
import { SignalingClient } from 'amazon-kinesis-video-streams-webrtc';
```

Browser

Anda tidak perlu menginstal SDK untuk menggunakannya dalam skrip browser. Anda dapat memuat paket SDK yang dihosting langsung AWS dengan skrip di halaman HTML Anda.

Untuk menggunakan SDK di browser, tambahkan elemen skrip berikut ke halaman HTML Anda:

```
<script src="https://unpkg.com/amazon-kinesis-video-streams-webrtc/dist/kvs-
webrtc.min.js"></script>
```

Setelah SDK dimuat di halaman Anda, SDK tersedia dari variabel global KVSWebRTC (atau window.KVSWebRTC).

Misalnya, window.KVSWebRTC.SignalingClient.

Kinesis Video Streams JavaScript dengan dokumentasi WebRTC SDK

[Dokumentasi untuk metode SDK ada di GitHub readme, di bawah Dokumentasi.](#)

Di bagian [Penggunaan](#), ada informasi tambahan untuk mengintegrasikan SDK ini bersama dengan AWS SDK JavaScript untuk membangun aplikasi penampil berbasis web.

Lihat examples direktori untuk contoh aplikasi lengkap, termasuk peran master dan penampil.

Gunakan Kinesis Video Streams dengan halaman uji WebRTC

Kinesis Video Streams dengan WebRTC juga menghosting halaman uji yang dapat Anda gunakan untuk membuat saluran pensinyalan baru atau terhubung ke saluran yang ada dan menggunakannya sebagai master atau penampil.

[Kinesis Video Streams dengan halaman uji WebRTC](https://awslabs.github.io/-/examples/index.html) terletak di [https://awslabs.github.io/ - /examples/index.html](https://awslabs.github.io/-/examples/index.html). [amazon-kinesis-video-streams webrtc-sdk-js](#)

Kode untuk halaman uji ada di `examples` direktori.

Topik

- [Streaming dari halaman pengujian ke AWS Management Console](#)
- [Streaming dari halaman pengujian ke halaman pengujian](#)

Streaming dari halaman pengujian ke AWS Management Console

1. Buka [Kinesis Video Streams dengan halaman uji WebRTC](#) dan lengkapi yang berikut ini:

- Wilayah AWS. Sebagai contoh, `us-west-2`.
- Kunci AWS akses dan kunci rahasia untuk pengguna atau peran IAM Anda. Biarkan token sesi kosong jika Anda menggunakan AWS kredensi jangka panjang.
- Nama saluran pensinyalan yang ingin Anda sambungkan.

Jika Anda ingin terhubung ke saluran pensinyalan baru, pilih **Buat Saluran** untuk membuat saluran pensinyalan dengan nilai yang disediakan di kotak.

Note

Nama saluran pensinyalan Anda harus unik untuk akun dan wilayah saat ini. Anda dapat menggunakan huruf, angka, garis bawah (`_`), dan tanda hubung (`-`), tetapi bukan spasi.

- Apakah Anda ingin mengirim audio, video, atau keduanya.
 - Generasi kandidat ICE. Biarkan `STUN/TURN` dipilih dan biarkan `Trickle ICE` diaktifkan.
2. Pilih **Mulai Master** untuk terhubung ke saluran pensinyalan.

Izinkan akses ke kamera dan/atau mikrofon Anda, jika diperlukan.

3. Buka konsol [Kinesis Video Streams](#) di AWS Management Console

Pastikan wilayah yang benar dipilih.

4. Di navigasi kiri, pilih [saluran pensinyalan](#).

Pilih nama saluran pensinyalan di atas. Gunakan bilah pencarian, jika perlu.

5. Perluas bagian Penampil pemutaran media.

6. Pilih tombol putar pada pemutar video. Ini bergabung dengan sesi WebRTC sebagai `viewer`

Media yang sedang dikirim pada halaman demo harus ditampilkan di AWS Management Console.

Streaming dari halaman pengujian ke halaman pengujian

1. Buka [Kinesis Video Streams dengan halaman uji WebRTC](#) dan lengkapi informasi berikut:

- Wilayah AWS. Sebagai contoh, `us-west-2`.
- Kunci AWS akses dan kunci rahasia untuk pengguna atau peran IAM Anda. Biarkan token sesi kosong jika Anda menggunakan AWS kredensi jangka panjang.
- Nama saluran pensinyalan yang ingin Anda sambungkan.

Jika Anda ingin terhubung ke saluran pensinyalan baru, pilih `Buat Saluran` untuk membuat saluran pensinyalan dengan nilai yang disediakan di kotak.

Note

Nama saluran pensinyalan Anda harus unik untuk akun dan wilayah saat ini. Anda dapat menggunakan huruf, angka, garis bawah (`_`), dan tanda hubung (`-`), tetapi bukan spasi.

- Apakah Anda ingin mengirim audio, video, atau keduanya.
 - Generasi kandidat ICE. Biarkan `STUN/TURN` dipilih dan biarkan `Trickle ICE` diaktifkan.
- ### 2. Pilih `Mulai Master` untuk terhubung ke saluran pensinyalan sebagai master peran.

Izinkan akses ke kamera dan/atau mikrofon Anda, jika diperlukan.

3. Buka tab browser lain dan buka [Kinesis Video Streams dengan halaman uji WebRTC](#). Semua informasi dari proses sebelumnya harus dimuat.

4. Gulir ke bawah dan pilih Mulai Penampil untuk terhubung ke saluran pensinyalan sebagai viewer peran.

Anda harus melihat media dipertukarkan antara master dan viewer.

Edit Kinesis Video Streams dengan halaman uji WebRTC

Untuk mengedit SDK dan halaman pengujian untuk tujuan pengembangan, ikuti petunjuk di bawah ini.

Prasyarat

NodeJS versi 16+

Note

Kami merekomendasikan mengunduh versi dukungan jangka panjang (LTS) terbaru dari <https://nodejs.org/en/download>.

Edit halaman pengujian

1. Unduh Kinesis Video Streams dengan JavaScript WebRTC SDK di.

Ketik yang berikut ini di terminal:

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

2. Arahkan ke direktori dengan file package.json. File ini terletak di direktori root repositori.

Ketik yang berikut ini di terminal:

```
cd amazon-kinesis-video-streams-webrtc-sdk-js
```

3. Instal dependensi.

Ketik perintah [CLI npm](#) berikut di terminal:

```
npm install
```

4. Mulai server web untuk mulai melayani halaman web.

Ketik perintah [CLI npm](#) berikut di terminal:

```
npm run develop
```

5. Di browser Anda, kunjungi <http://localhost:3001/>.

Anda dapat melakukan pengeditan ke halaman web dengan mengedit file di `examples` direktori.

WebRTC SDK for Android

step-by-step Petunjuk berikut menjelaskan cara mengunduh, membuat, dan menjalankan Kinesis Video Streams dengan WebRTC SDK for Android dan sampelnya yang sesuai.

Note

Kinesis Video Streams tidak mendukung alamat IPv6 di Android. [Untuk informasi selengkapnya dan langkah-langkah tentang menonaktifkan IPv6 di perangkat Android Anda, lihat https://support.surfshark.com/hc/en-us/articles/360011828279-H-IPv6-on-Android-how-to-disable](https://support.surfshark.com/hc/en-us/articles/360011828279-H-IPv6-on-Android-how-to-disable)

Unduh SDK WebRTC untuk Android

Untuk mengunduh SDK WebRTC di Android, jalankan perintah berikut:

```
$ git clone https://github.com/aws-labs/amazon-kinesis-video-streams-webrtc-sdk-android.git
```

Membangun WebRTC SDK di Android

Untuk membangun SDK WebRTC di Android, selesaikan langkah-langkah berikut:

1. Impor Android WebRTC SDK ke lingkungan pengembangan terintegrasi (IDE) Android Studio dengan membuka dengan Open as Project. **amazon-kinesis-video-streams-webrtc-sdk-android/build.gradle**
2. Jika Anda membuka proyek untuk pertama kalinya, secara otomatis akan disinkronkan. Jika tidak - memulai sinkronisasi. Ketika Anda melihat kesalahan build, pilih untuk menginstal SDK

- yang diperlukan dengan memilih Instal paket SDK yang hilang, lalu pilih Terima dan selesaikan penginstalan.
3. Konfigurasi setelan Amazon Cognito (kumpulan pengguna dan kumpulan identitas). Untuk langkah-langkah detail, lihat [Konfigurasi Amazon Cognito untuk Android WebRTC SDK](#). Ini menghasilkan pengaturan autentikasi dan otorisasi yang diperlukan untuk membangun Android WebRTC SDK.
 4. Di IDE Android Anda, buka `awsconfiguration.json` (di `src/main/res/raw/`). File terlihat seperti berikut:

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "REPLACE_ME",
        "Region": "REPLACE_ME"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "CognitoUserPool": {
    "Default": {
      "AppClientSecret": "REPLACE_ME",
      "AppClientId": "REPLACE_ME",
      "PoolId": "REPLACE_ME",
      "Region": "REPLACE_ME"
    }
  }
}
```

Perbarui `awsconfiguration.json` dengan nilai yang dihasilkan dengan menjalankan langkah-langkah di [Konfigurasi Amazon Cognito untuk Android WebRTC SDK](#).

5. Pastikan perangkat Android Anda terhubung ke komputer tempat Anda menjalankan IDE Android. Di Android IDE, pilih perangkat yang terhubung lalu buat dan jalankan WebRTC Android SDK.

Langkah ini menginstal aplikasi yang dipanggil `AWSKinesisVideoWebRTCDemoApp` di perangkat Android Anda. Dengan menggunakan aplikasi ini, Anda dapat memverifikasi streaming audio/video WebRTC langsung antara klien perangkat seluler, web, dan IoT.

Jalankan Aplikasi Sampel Android

Selesaikan langkah-langkah berikut:

1. Di perangkat Android Anda, buka `AWSKinesisVideoWebRTCDemoApp` dan masuk menggunakan akun baru (dengan membuatnya terlebih dahulu) atau akun Amazon Cognito yang sudah ada.
2. Masuk `AWSKinesisVideoWebRTCDemoApp`, navigasikan ke halaman Konfigurasi Saluran dan buat saluran pensinyalan baru atau pilih yang sudah ada.

Note

Saat ini, dengan menggunakan contoh aplikasi di SDK ini, Anda hanya dapat menjalankan satu saluran pensinyalan. `AWSKinesisVideoWebRTCDemoApp`


3. Opsional: pilih ID Klien unik jika Anda ingin terhubung ke saluran ini sebagai penampil. ID klien hanya diperlukan jika beberapa pemirsa terhubung ke saluran. Ini membantu master saluran mengidentifikasi masing-masing pemirsa.
4. Pilih Wilayah AWS dan apakah Anda ingin mengirim data audio atau video, atau keduanya.
5. Untuk memverifikasi peer-to-peer streaming, lakukan salah satu hal berikut:

Note

Pastikan Anda menentukan nama saluran pensinyalan, AWS wilayah, ID penampil, dan ID AWS akun yang sama pada semua klien yang Anda gunakan dalam demo ini.

- Peer-to-peer streaming antara dua perangkat Android: master dan viewer
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan Android WebRTC SDK di dua perangkat Android.

- Buka `AWSKinesisVideoWebRTCDemoApp` di satu perangkat Android dalam mode master (pilih MULAI MASTER) untuk memulai sesi baru (saluran pensinyalan).

 Note

Saat ini, hanya ada satu master untuk saluran pensinyalan tertentu.

- Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat Android kedua Anda dalam mode penampil untuk terhubung ke saluran pensinyalan (sesi) yang dimulai pada langkah di atas (pilih MULAI PENAMPIL).

Verifikasi bahwa pemirsa dapat melihat data audio/video master.

- Peer-to-peer Streaming P antara master SDK yang disematkan dan penampil perangkat Android
 - Unduh, buat, dan jalankan mode master [WebRTC SDK di C untuk Perangkat Tertanam](#) di perangkat kamera.
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan Android WebRTC SDK di perangkat Android. Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat Android ini dalam mode penampil dan verifikasi bahwa penampil dapat melihat data audio/video master SDK yang disematkan.
- Peer-to-peer streaming antara perangkat Android sebagai master dan browser web sebagai penampil
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan Android WebRTC SDK di perangkat Android. Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat Android ini dalam mode master (pilih MULAI MASTER) untuk memulai sesi baru (saluran pensinyalan).
 - Unduh, buat, dan jalankan penampil [Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web](#) as dan verifikasi bahwa pemirsa dapat melihat audio/video master Android.

Konfigurasi Amazon Cognito untuk Android WebRTC SDK

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

Untuk mengatur kumpulan pengguna

1. Masuk ke [konsol Amazon Cognito](#) dan verifikasi wilayah tersebut 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. Tunjukkan 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 wilayah tersebut 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 yang Diautentikasi
 - 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.

WebRTC SDK untuk iOS

step-by-step Petunjuk berikut menjelaskan cara mengunduh, membuat, dan menjalankan SDK WebRTC Kinesis Video Streams di iOS dan sampelnya yang sesuai.

Unduh SDK WebRTC di iOS

Untuk mengunduh SDK WebRTC di iOS, jalankan perintah berikut:

```
$ git clone https://github.com/aws-labs/amazon-kinesis-video-streams-webrtc-sdk-ios.git
```

Membangun SDK WebRTC di iOS

Selesaikan langkah-langkah berikut:

1. Impor SDK WebRTC iOS ke lingkungan pengembangan terintegrasi (IDE) XCode di komputer iOS dengan `KinesisVideoWebRTCDemoApp.xcworkspace` membuka (path: - /Swift/ .xcworkspace). `amazon-kinesis-video-streams-webrtc-sdk-ios`
`AWSKinesisVideoWebRTCDemoApp`
2. Jika Anda membuka proyek untuk pertama kalinya, proyek akan dibangun secara otomatis. Jika tidak, mulailah membangun.

Anda mungkin melihat kesalahan berikut:

```
error: The sandbox is not in sync with the Podfile.lock. Run 'pod install' or update your CocoaPods installation.
```

Jika Anda melihat ini, lakukan hal berikut:

- a. Ubah direktori kerja Anda saat ini menjadi `amazon-kinesis-video-streams-webrtc-sdk-ios/Swift` dan jalankan yang berikut di baris perintah:

```
pod cache clean --all
pod install
```

- b. Ubah direktori kerja Anda saat ini menjadi `amazon-kinesis-video-streams-webrtc-sdk-ios` dan jalankan yang berikut di baris perintah:

```
$ git checkout Swift/Pods/AWSCore/AWSCore/Service/AWSService.m
```

- c. `Build` lagi.
3. Konfigurasi setelan Amazon Cognito (kumpulan pengguna dan kumpulan identitas). Untuk langkah-langkah detail, lihat [Konfigurasi Amazon Cognito untuk SDK WebRTC iOS](#). Ini menghasilkan pengaturan otentikasi dan otorisasi yang diperlukan untuk membangun SDK WebRTC iOS.
4. Di IDE Anda, buka `awsconfiguration.json` file (dari `Swift/KVSiOSApp`). File tersebut terlihat seperti berikut:

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "REPLACEME",
        "Region": "REPLACEME"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "CognitoUserPool": {
    "Default": {
      "AppClientSecret": "REPLACEME",
      "AppClientId": "REPLACEME",
      "PoolId": "REPLACEME",
      "Region": "REPLACEME"
    }
  }
}
```

Perbarui `awsconfiguration.json` dengan nilai yang dihasilkan dengan menjalankan langkah-langkah di [Konfigurasi Amazon Cognito untuk Android WebRTC SDK](#).

5. Di IDE Anda, buka `Constants.swift` file (dari `Swift/KVSiOSApp`). File tersebut terlihat seperti berikut:

```
import Foundation
import AWSCognitoIdentityProvider

let CognitoIdentityUserPoolRegion = AWSRegionType.USWest2
let CognitoIdentityUserPoolId = "REPLACEME"
let CognitoIdentityUserPoolAppClientId = "REPLACEME"
let CognitoIdentityUserPoolAppClientSecret = "REPLACEME"

let AWSCognitoUserPoolsSignInProviderKey = "UserPool"
let CognitoIdentityPoolID = "REPLACEME"

let AWSKinesisVideoEndpoint = "https://kinesisvideo.us-west-2.amazonaws.com"
let AWSKinesisVideoKey = "kinesisvideo"

let VideoProtocols = ["WSS", "HTTPS"]

let ConnectAsMaster = "connect-as-master"
let ConnectAsViewer = "connect-as-viewer"

let MasterRole = "MASTER"
let ViewerRole = "VIEWER"

let ClientID = "ConsumerViewer"
```

Perbarui `Constants.swift` dengan nilai yang dihasilkan dengan menjalankan langkah-langkah di [Konfigurasi Amazon Cognito untuk Android WebRTC SDK](#).

6. Pastikan perangkat iOS Anda terhubung ke komputer Mac tempat Anda menjalankan XCode. Di XCode, pilih perangkat yang terhubung lalu buat dan jalankan SDK iOS WebRTC.

Langkah ini menginstal aplikasi yang dipanggil `AWSKinesisVideoWebRTCDemoApp` di perangkat iOS Anda. Dengan menggunakan aplikasi ini, Anda dapat memverifikasi streaming audio/video WebRTC langsung antara klien perangkat seluler, web, dan IoT.

Jalankan Aplikasi Sampel iOS

Selesaikan langkah-langkah berikut:

1. Di perangkat iOS Anda, buka `AWSKinesisVideoWebRTCDemoApp` dan masuk menggunakan akun baru (dengan membuatnya terlebih dahulu) atau akun Amazon Cognito yang sudah ada.
2. Masuk `AWSKinesisVideoWebRTCDemoApp`, navigasikan ke halaman Konfigurasi Saluran dan buat saluran pensinyalan baru atau pilih yang sudah ada.

Note

Saat ini, dengan menggunakan contoh aplikasi di SDK ini, Anda hanya dapat menjalankan satu saluran pensinyalan. `AWSKinesisVideoWebRTCDemoApp`

3. (Opsional) Pilih ID Klien unik jika Anda ingin terhubung ke saluran ini sebagai penampil. ID Klien hanya diperlukan jika beberapa pemirsa terhubung ke saluran. Ini membantu master saluran mengidentifikasi masing-masing pemirsa.
4. Pilih Wilayah AWS dan apakah Anda ingin mengirim data audio atau video, atau keduanya.
5. Untuk memverifikasi peer-to-peer streaming, lakukan salah satu hal berikut:

Note

Pastikan Anda menentukan nama saluran pensinyalan, AWS wilayah, ID penampil, dan ID AWS akun yang sama pada semua klien yang Anda gunakan dalam demo ini.

- Peer-to-peer streaming antara dua perangkat iOS: master dan viewer
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan SDK WebRTC iOS di dua perangkat iOS.
 - Buka `AWSKinesisVideoWebRTCDemoApp` di satu perangkat iOS dalam mode master (pilih MULAI MASTER) untuk memulai sesi baru (saluran pensinyalan).

Note

Saat ini, hanya ada satu master untuk saluran pensinyalan tertentu.

- Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat iOS kedua Anda dalam mode penampil untuk terhubung ke saluran pensinyalan (sesi) yang dimulai pada langkah di atas (pilih MULAI PENAMPIL).

Verifikasi bahwa pemirsa dapat melihat data audio/video master.

- Peer-to-peer Streaming P antara master SDK yang disematkan dan penampil perangkat iOS
 - Unduh, buat, dan jalankan mode master [WebRTC SDK di C untuk Perangkat Tertanam](#) di perangkat kamera.
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan SDK WebRTC iOS di perangkat iOS. Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat iOS ini dalam mode penampil dan verifikasi bahwa penampil iOS dapat melihat data audio/video master SDK yang disematkan.
- Peer-to-peer streaming antara perangkat iOS sebagai master dan browser web sebagai penampil
 - Menggunakan prosedur di atas, unduh, buat, dan jalankan SDK WebRTC iOS di perangkat iOS. Buka `AWSKinesisVideoWebRTCDemoApp` di perangkat iOS ini dalam mode master (pilih MULAI MASTER) untuk memulai sesi baru (saluran pensinyalan).
 - Unduh, buat, dan jalankan penampil [Kinesis Video Streams dengan WebRTC JavaScript SDK untuk aplikasi web](#) dan verifikasi bahwa JavaScript pemirsa dapat melihat audio/video master Android.

Konfigurasi Amazon Cognito untuk SDK WebRTC iOS

Prasyarat

- Kami merekomendasikan XCode untuk memeriksa, mengedit, dan menjalankan kode aplikasi. Kami merekomendasikan versi terbaru.
- Dalam kode contoh, Anda memberikan kredensi Amazon Cognito.

Ikuti prosedur ini untuk menyiapkan kumpulan pengguna Amazon Cognito dan kumpulan 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. Tunjukkan 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 yang Diautentikasi
 - Sumber identitas yang diautentikasi - Pilih kumpulan pengguna Amazon Cognito

Pilih 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.

Metrik Klien untuk WebRTC C SDK

Aplikasi yang dibangun dengan Amazon Kinesis Video Streams dengan WebRTC terdiri dari berbagai bagian yang bergerak, termasuk jaringan, pensinyalan, pertukaran kandidat, koneksi rekan, dan pertukaran data. Kinesis Video Streams dengan WebRTC di C mendukung berbagai metrik sisi klien yang memungkinkan Anda memantau dan melacak kinerja dan penggunaan komponen ini dalam aplikasi Anda. [Metrik yang didukung terbagi dalam dua kategori utama: metrik khusus yang didefinisikan secara khusus untuk implementasi sinyal dan jaringan Kinesis Video Streams, dan metrik spesifik protokol terkait media dan data yang berasal dari standar W3C.](#) Perhatikan bahwa hanya sebagian dari metrik standar W3C yang saat ini didukung untuk Kinesis Video Streams dengan WebRTC di C.

Topik

- [Metrik Pensinyalan](#)
- [Metrik Standar W3C Didukung untuk WebRTC C SDK](#)

Metrik Pensinyalan

Metrik pensinyalan dapat digunakan untuk memahami bagaimana klien pensinyalan berperilaku saat aplikasi Anda berjalan. Anda dapat menggunakan STATUS `signalingClientGetMetrics` (`SIGNALING_CLIENT_HANDLE`, `PSignalingClientMetrics`) API untuk mendapatkan metrik pensinyalan ini. Berikut adalah contoh pola penggunaan:

```
SIGNALING_CLIENT_HANDLE signalingClientHandle;
SignalingClientMetrics signalingClientMetrics;
STATUS retStatus = signalingClientGetMetrics(signalingClientHandle,
&signalingClientMetrics);
printf("Signaling client connection duration: %" PRIu64 " ms",
(signalingClientMetrics.signalingClientStats.connectionDuration /
HUNDREDS_OF_NANOS_IN_A_MILLISECOND));
```

Definisi `signalingClientStats` dapat ditemukan di [Stats.h](#).

Metrik pensinyalan berikut saat ini didukung:

Metrik	Deskripsi	
cpApiCallLatensi	Hitung latensi untuk panggilan API bidang kontrol. Perhitungan dilakukan dengan menggunakan Exponential Moving Average (EMA). Panggilan terkait meliputi: DescribeChannel, createChannel, dan deleteChannel. getChannelEndpoint	
dpApiCallLatensi	Hitung latensi untuk panggilan API bidang data. Perhitungan dilakukan dengan menggunakan Exponential Moving Average (EMA). Panggilan terkait meliputi: getIceConfig.	
signalingClientUptime	Ini menunjukkan waktu keberadaan objek klien. Setiap kali metrik ini dipanggil, nilai uptime terbaru dipancarkan.	
KoneksiDurasi	Jika koneksi dibuat, ini memancarkan durasi koneksi hidup. Lain, nilai 0 dipancarkan. Ini berbeda dengan memberi sinyal uptime klien karena, koneksi datang dan pergi, tetapi signalingClientUptime merupakan indikasi dari objek klien itu sendiri.	
numberOfMessagesDikirim	Nilai ini diperbarui ketika rekan mengirimkan penawaran, jawaban, atau kandidat ICE.	

Metrik	Deskripsi	
numberOfMessagesDiterima	Tidak seperti numberOfMessages Terkirim, metrik ini diperbarui untuk semua jenis pesan pensinyalan. Jenis pesan pensinyalan tersedia di SIGNALING_MESSAGE_TYPE .	
iceRefreshCount	Ini bertambah saat getIceConfig dipanggil. Tingkat di mana ini dipanggil didasarkan pada TTL sebagai bagian dari konfigurasi ICE yang diterima. Setiap kali satu set konfigurasi ICE baru diterima, timer diatur untuk menyegarkan waktu berikutnya, mengingat validitas kredensial dalam konfigurasi dikurangi beberapa masa tenggang.	
numberOfErrors	Penghitung digunakan untuk melacak jumlah kesalahan yang dihasilkan dalam klien pensinyalan. Kesalahan yang dihasilkan saat mendapatkan konfigurasi ICE, mendapatkan status pensinyalan, melacak metrik pensinyalan, mengirim pesan pensinyalan, dan menghubungkan klien pensinyalan ke soket web untuk mengirim/menerima pesan dilacak.	

Metrik	Deskripsi
<code>numberOfRuntimeKesalahan</code>	Metrik mencakup kesalahan yang terjadi saat inti klien pensinyalan sedang berjalan. Skenario seperti kegagalan menyambung kembali, kegagalan penerimaan pesan, dan kesalahan penyegaran konfigurasi ICE dilacak di sini.
<code>numberOfReconnects</code>	Metrik bertambah pada setiap penyambungan kembali. Ini adalah metrik yang berguna untuk memahami stabilitas koneksi jaringan dalam pengaturan.

Metrik Standar W3C Didukung untuk WebRTC C SDK

Subset dari metrik standar [W3C](#) saat ini didukung untuk aplikasi yang dibangun dengan WebRTC C SDK. Ini termasuk dalam kategori berikut:

- Jaringan:
 - [Kandidat Es](#): metrik ini memberikan informasi tentang kandidat lokal dan jarak jauh yang dipilih untuk pertukaran data antara rekan-rekan. Ini termasuk sumber server kandidat, alamat IP, jenis kandidat yang dipilih untuk komunikasi, dan prioritas kandidat. Metrik ini berguna sebagai laporan snapshot.
 - [Ice Server](#): metrik ini untuk mengumpulkan informasi operasional tentang berbagai server ICE yang didukung. Ini berguna ketika mencoba memahami server yang terutama digunakan untuk pemeriksaan komunikasi dan konektivitas. Dalam beberapa kasus, juga berguna untuk memeriksa metrik ini jika pengumpulan kandidat gagal.
 - [Pasangan Kandidat Es](#): metrik ini untuk memahami jumlah byte/paket yang dipertukarkan antara rekan-rekan dan juga pengukuran terkait waktu.
- Media dan data:

- [RTP Inbound Jarak Jauh](#): metrik ini mewakili perspektif titik akhir dari aliran data yang dikirim oleh pengirim.
- [Outbound RTP](#): metrik ini memberikan informasi tentang aliran RTP keluar. Mereka juga bisa sangat berguna saat menganalisis streaming berombak atau streaming berhenti.
- [RTP masuk](#): metrik ini memberikan informasi tentang media yang masuk.
- [Metrik saluran data](#): metrik ini dapat membantu Anda menganalisis jumlah pesan dan byte yang dikirim dan diterima melalui saluran data. Metrik dapat ditarik dengan menggunakan ID saluran.

Anda dapat menggunakan STATUS `rtcPeerConnectionGetMetrics` (`PRtcPeerConnection`, `PRtcRtpTransceiver`, `PRtcStats`) API untuk mengumpulkan metrik yang terkait dengan ICE, RTP, dan saluran data. Berikut adalah contoh penggunaan:

```
RtcStats rtcStats;
rtcStats.requestedTypeOfStats = RTC_STATS_TYPE_LOCAL_CANDIDATE;
STATUS retStatus = rtcPeerConnectionGetMetrics (pRtcPeerConnection, NULL, &rtcStats);
printf("Local Candidate address: %s\n",
    rtcStats.rtcStatsObject.localIceCandidateStats.address);
```

Berikut contoh lain yang menunjukkan pola penggunaan untuk mendapatkan statistik terkait transceiver:

```
RtcStats rtcStats;
PRtcRtpTransceiver pVideoRtcRtpTransceiver;
rtcStats.requestedTypeOfStats = RTC_STATS_TYPE_OUTBOUND_RTP;
STATUS retStatus = rtcPeerConnectionGetMetrics (pRtcPeerConnection,
    pVideoRtcRtpTransceiver, &rtcStats);
printf("Number of packets discarded on send: %s\n",
    rtcStats.rtcStatsObject.outboundRtpStreamStats.packetsDiscardedOnSend);
```

Dalam contoh di atas, jika argumen kedua untuk `rtcPeerConnection GetMetrics ()` adalah `NULL`, data untuk transceiver pertama dalam daftar dikembalikan.

[Definisi untuk `rtcStatsObject` dapat ditemukan di `Stats.h`. dan definisi untuk `RtcStats` dapat ditemukan di `Include.h`.](#)

[Contoh penggunaan API dan metrik yang berbeda dapat ditemukan di direktori sampel di repositori `WebRTC C SDK` dan di repositori `demo Kinesis Video Stream`.](#)

Metrik standar [W3C](#) berikut saat ini didukung untuk aplikasi yang dibangun dengan `WebRTC C SDK`.

Topik

- [Jaringan](#)
- [Media](#)
- [Saluran Data](#)

Jaringan

Metrik Server ICE:

Metrik	Deskripsi	
URL	URL server STUN/TURN sedang dilacak	
Pelabuhan	Nomor port yang digunakan oleh klien	
Protokol	Protokol transportasi diekstrak dari URI ICE Server. Jika nilainya UDP, ICE mencoba TURN over UDP, jika tidak ICE mencoba TURN over TCP/TLS. Jika URI tidak mengandung transport, ICE mencoba TURN over UDP dan TCP/TLS. Dalam kasus STUN server, bidang ini kosong.	
Total Permintaan Terkirim	Nilai diperbarui untuk setiap permintaan kandidat srflx dan saat mengirim permintaan yang mengikat dari kandidat giliran.	

Metrik	Deskripsi
Total Tanggapan Diterima	Nilai diperbarui setiap kali respons pengikatan STUN diterima.
Total Waktu Perjalanan Pulang Pergi	Nilai diperbarui setiap kali respons yang setara diterima untuk permintaan. Paket permintaan dilacak dalam peta hash dengan checksum sebagai kuncinya.

Statistik Kandidat ICE: Hanya informasi tentang kandidat yang dipilih (lokal dan jarak jauh) yang disertakan.

Metrik	Deskripsi
alamat	Ini menunjukkan alamat IP kandidat lokal dan jarak jauh.
pelabuhan	Nomor port kandidat
protokol	Protokol digunakan untuk mendapatkan kandidat. Nilai yang valid adalah UDP/TCP.
Jenis CandidateType	Jenis kandidat yang dipilih - host, srflx atau relay.
prioritas	Prioritas kandidat lokal dan jarak jauh yang dipilih.
url	Sumber kandidat lokal yang dipilih. Ini memberikan indikasi apakah kandidat yang dipilih

Metrik	Deskripsi
	diterima dari server STUN atau server TURN.
RelayProtocol	Jika server TURN digunakan untuk mendapatkan kandidat lokal yang dipilih, bidang ini menunjukkan protokol apa yang digunakan untuk mendapatkannya. Nilai yang valid adalah TCP/UDP.

Statistik Pasangan Calon ICE: Hanya informasi tentang pasangan calon yang dipilih yang disertakan.

Metrik	Deskripsi
localCandidateId	ID dari kandidat lokal yang dipilih dalam pasangan.
remoteCandidateId	ID kandidat jarak jauh yang dipilih dalam pasangan.
negara	Keadaan pasangan calon yang sedang diperiksa.
dinominasikan	Setel ke TRUE karena statistik ditarik untuk pasangan kandidat yang dipilih.
PaketsEnt	Jumlah paket yang dikirim Ini dihitung dalam . panggilan dalam writeFrame panggilan. Informasi ini juga dapat diekstraksi dari Statistik RTP keluar, tetapi karena pasangan kandidat Ice menyertakan lastPacke

Metrik	Deskripsi
	<code>tSent</code> stempel waktu, mungkin berguna untuk menghitung jumlah paket yang dikirim antara dua titik waktu.
<code>PaketDiterima</code>	Ini diperbarui setiap kali <code>incomingDataHandler</code> dipanggil.
<code>olehTessent</code>	Ini dihitung <code>iceAgentS</code> <code>endPacket()</code> dalam <code>writeFrame()</code> panggilan. Ini berguna saat menghitung bit rate. Saat ini, ini juga termasuk header dan padding karena lapisan ICE tidak menyadari format paket RTP.
<code>BytesDiterima</code>	Ini diperbarui setiap kali <code>incomingDataHandler</code> dipanggil. Saat ini, ini juga termasuk header dan padding karena lapisan ICE tidak menyadari format paket RTP.
<code>lastPacketSentStempel waktu</code>	Ini diperbarui setiap kali paket dikirim. Ini dapat digunakan bersama dengan <code>PacketsSent</code> dan waktu mulai yang direkam dalam aplikasi ke laju transfer paket saat ini.

Metrik	Deskripsi	
<code>lastPacketReceivedStempel</code> waktu	Ini diperbarui saat menerima data di <code>incomingDataHandler()</code> . Ini dapat digunakan bersama dengan <code>PacketsReceived</code> untuk menyimpulkan tingkat penerimaan paket saat ini. Waktu mulai harus direkam di lapisan aplikasi di <code>transceiverOnFrame()</code> callback.	
<code>firstRequestTimestamp</code>	Terekam ketika permintaan pengikatan STUN pertama berhasil dikirim masuk. <code>iceAgentSendStunPacket()</code> Ini dapat digunakan bersama dengan <code>lastRequestTimestamp</code> dan <code>requestsSent</code> untuk menemukan waktu rata-rata antara permintaan pengikatan STUN.	
<code>lastRequestTimestamp</code>	Direkam setiap kali permintaan pengikatan STUN berhasil dikirim masuk. <code>iceAgentSendStunPacket()</code>	
<code>lastResponseTimestamp</code>	Direkam setiap kali respons pengikatan STUN diterima.	
<code>totalRoundTripWaktu</code>	Diperbarui saat respons yang mengikat diterima untuk permintaan. Permintaan dan respons dipetakan dalam tabel hash berdasarkan checksum.	

Metrik	Deskripsi	
currentRoundTripWaktu	Waktu perjalanan pulang pergi terbaru diperbarui ketika tanggapan yang mengikat diterima untuk permintaan pada pasangan kandidat.	
PermintaanDiterima	Penghitung yang diperbarui pada setiap permintaan pengikatan STUN yang diterima.	
PermintaanSent	Penghitung yang diperbarui pada setiap permintaan pengikatan STUN yang dikirim melalui <code>iceAgent.sendStunPacket()</code> .	
TanggapanTer kirim	Penghitung yang diperbarui pada setiap respons pengikatan STUN yang dikirim sebagai tanggapan atas permintaan yang mengikat di <code>handleStunPacket()</code> .	
TanggapanDiterima	Penghitung yang diperbarui pada setiap respons pengikatan STUN yang diterima di <code>handleStunPacket()</code> .	

Metrik	Deskripsi	
<code>packetsDiscardedOnKirim</code>	Diperbarui saat pengiriman paket gagal. Dengan kata lain, ini diperbarui ketika <code>iceUtilsSendData()</code> gagal. Ini berguna untuk menentukan persentase paket yang dijatuhkan dalam durasi tertentu.	
<code>bytesDiscardedOnKirim</code>	Diperbarui saat pengiriman paket gagal. Dengan kata lain, ini diperbarui ketika <code>iceUtilsSendData()</code> gagal. Ini berguna saat menentukan persentase paket yang dijatuhkan dalam durasi tertentu. Perhatikan bahwa penghitung juga menyertakan header paket.	

Media

Statistik RTP Keluar

Metrik	Deskripsi	
<code>voiceActivityFlag</code>	Ini saat ini merupakan bagian dari <code>RtcEncoderStats</code> didefinisikan dalam <code>Include.h</code> . Bendera diatur ke <code>TRUE</code> jika paket audio terakhir berisi suara. Bendera saat ini tidak diatur dalam sampel.	

Metrik	Deskripsi
PaketsEnt	Ini menunjukkan jumlah total paket RTP yang dikirim untuk SSRC yang dipilih. Ini adalah bagian dari https://www.w3.org/TR/webrtc-stats/#sentrtstats-dict * dan disertakan sebagai bagian dari statistik keluar. Ini bertambah setiap kali <code>writeFrame ()</code> dipanggil.
olehTessent	Jumlah total byte tidak termasuk header RTP dan padding yang dikirim. Ini diperbarui pada setiap panggilan <code>WriteFrame</code> .
EncoderImplementasi	Ini diperbarui oleh lapisan aplikasi sebagai bagian dari <code>RtcEncoderStats</code> objek.
packetsDiscardedOnKirim	Bidang ini diperbarui jika agen ICE gagal mengirim paket RTP terenkripsi karena alasan apa pun dalam panggilan <code>iceAgentSendPacket</code>
bytesDiscardedOnKirim	Bidang ini juga diperbarui jika agen ICE gagal mengirim paket RTP terenkripsi karena alasan apa pun dalam panggilan tersebut. <code>iceAgentSendPacket</code>

Metrik	Deskripsi	
FrameSent	Ini bertambah hanya jika jenis taktik aliran media adalah <code>MEDIA_STREAM_TRACK_KIND_VIDEO</code> .	
hugeFramesSent	Penghitung ini diperbarui untuk bingkai yang 2,5 kali ukuran rata-rata bingkai. Ukuran frame diperoleh dengan menghitung fps (berdasarkan waktu hitungan frame terakhir yang diketahui dan jumlah frame yang dikodekan dalam interval waktu) dan menggunakan <code>targetBitRate</code> yang <code>RtcEncoderStats</code> ditetapkan oleh aplikasi.	
FrameDisandikan	Penghitung ini diperbarui hanya untuk trek video setelah pengkodean frame berhasil. Ini diperbarui pada setiap panggilan <code>WriteFrame</code> .	
keyFramesEncoded	Penghitung ini diperbarui hanya untuk trek video setelah berhasil menyandikan bingkai kunci. Ini diperbarui pada setiap panggilan <code>WriteFrame</code> .	

Metrik	Deskripsi	
framesDiscardedOnKirim	<p>Ini diperbarui ketika pengirim n bingkai gagal karena kegagalan iceAgentSendPacket panggilan . Sebuah frame terdiri dari sekelompok paket dan saat ini, framesDiscardedOnSend gagal jika ada paket yang dibuang saat mengirim karena kesalahan.</p>	
FrameWidth	<p>Ini idealnya mewakili lebar bingkai dari bingkai yang dikodekan terakhir. Saat ini, ini diatur ke nilai oleh aplikasi sebagai bagian dari RtcEncoderStats * * dan tidak terlalu penting.</p>	
FrameHeight	<p>Ini idealnya mewakili ketinggian bingkai dari bingkai yang dikodekan terakhir. Saat ini, ini diatur ke nilai oleh aplikasi sebagai bagian dari RtcEncoderStats dan tidak terlalu penting.</p>	
frameBitDepth	<p>Ini mewakili kedalaman bit per lebar piksel dari bingkai yang dikodekan terakhir. Saat ini, ini diatur oleh aplikasi sebagai bagian dari RtcEncoderStats dan diterjemahkan ke dalam statistik keluar.</p>	

Metrik	Deskripsi	
NackCount	Nilai ini diperbarui setiap kali NACK diterima pada paket RTP dan upaya ulang untuk mengirim paket dilakukan . Tumpukan mendukung transmisi ulang paket saat menerima NACK.	
FirCount	Nilai diperbarui saat menerima paket FIR (onRtcpPacket->onrtcpfirpacket). Ini menunjukkan seberapa sering aliran tertinggal dan harus melewati bingkai untuk mengejar ketinggalan. Paket FIR saat ini tidak diterjemahkan untuk mengekstrak bidang, jadi, meskipun hitungan diatur, tidak ada tindakan yang diambil.	
PliCount	Nilai diperbarui saat menerima paket PLI (-> onrtcppli packet)onRtcpPacket. Ini menunjukkan bahwa sejumlah data video yang dikodekan telah hilang untuk satu atau lebih frame.	
SliCount	Nilai diperbarui saat menerima paket SLI (-> onRTCPsli packet)onRtcpPacket. Ini menunjukkan seberapa sering kehilangan paket mempengaruhi satu frame.	

Metrik	Deskripsi	
qualityLimitationResolution Perubahan	Saat ini, tumpukan mendukung metrik ini, namun, lebar dan tinggi bingkai tidak dipantau untuk setiap bingkai yang dikodekan.	
lastPacketSentStempel waktu	Stempel waktu di mana paket terakhir dikirim. Ini diperbarui pada setiap panggilan WriteFrame.	
headerBytesSent	Jumlah total header RTP dan byte padding yang dikirim untuk SSRC ini tidak termasuk muatan RTP yang sebenarnya.	
bytesDiscardedOnKirim	Ini diperbarui ketika pengirim n bingkai gagal karena kegagalan panggilan iceAgentSend paket. Sebuah frame terdiri dari sekelompok paket, yang pada gilirannya terdiri dari byte dan saat ini, bytesDiscardedOn Send gagal jika ada paket yang dibuang saat mengirim karena kesalahan.	

Metrik	Deskripsi	
retransmittedPacketsSent	Jumlah paket yang ditransmisikan kembali pada penerimaan PLI/SLI/NACK. Saat ini, tumpukan hanya menghitung paket resent NACK karena transmisi ulang berbasis PLI dan SLI tidak didukung.	
retransmittedBytesSent	Jumlah byte yang ditransmisikan kembali pada penerimaan PLI/SLI/NACK. Saat ini, tumpukan hanya menghitung byte resent dari NACK karena transmisi ulang berbasis PLI dan SLI tidak didukung.	
TargetBitrate	Ini diatur dalam tingkat aplikasi.	
totalEncodedBytesTarget	Ini ditingkatkan dengan ukuran frame target dalam byte setiap kali frame dikodekan. Ini diperbarui menggunakan parameter ukuran dalam struktur Frame.	
framesPerSecond	Ini dihitung berdasarkan waktu yang direkam untuk bingkai terencode terakhir yang diketahui dan jumlah frame yang dikirim dalam satu detik.	
totalEncodeTime	Ini diatur ke nilai arbitrer dalam aplikasi dan diterjemahkan ke statistik keluar secara internal.	

Metrik	Deskripsi
totalPacketSendKeterlambatan	Ini saat ini diatur ke 0 karena iceAgentSend Packet mengirim paket segera.

Statistik RTP masuk jarak jauh:

Metrik	Deskripsi
roundTripTime	Nilai diekstraksi dari laporan penerima RTCP saat menerima paket RTCP tipe 201 (laporan penerima). Laporan tersebut terdiri dari rincian seperti laporan pengirim terakhir dan penundaan sejak laporan pengirim terakhir untuk menghitung waktu pulang pergi. Laporan pengirim dihasilkan kira-kira setiap 200 milidetik yang terdiri dari informasi seperti jumlah paket yang dikirim dan byte yang dikirim yang diekstraksi dari statistik keluar.
totalRoundTripWaktu	Jumlah waktu perjalanan pulang pergi dihitung
FraksiHilang	Merupakan fraksi paket RTP yang hilang untuk SSRC sejak pengirim/penerima ReportFractionLost sebelumnya dikirim.

Metrik	Deskripsi	
LaporanDiterima	Diperbarui setiap kali paket jenis laporan penerima diterima.	
roundTripTimePengukuran	Menunjukkan jumlah total laporan yang diterima untuk SSRC yang berisi waktu pulang pergi yang valid. Namun, saat ini nilai ini bertambah terlepas sehingga artinya sama dengan ReportsReceived.	

Statistik RTP Masuk:

Metrik	Deskripsi	
PaketDiterima	Penghitung diperbarui ketika paket diterima untuk SSRC tertentu.	
jitter	Metrik ini menunjukkan paket Jitter yang diukur dalam hitungan detik untuk SSRC tertentu.	
jitterBufferDelay	Metrik ini menunjukkan jumlah waktu yang dihabiskan oleh setiap paket dalam buffer jitter.	
jitterBufferEmittedHitung	Jumlah total sampel audio atau bingkai video yang keluar dari buffer jitter.	

Metrik	Deskripsi	
PaketDibuang	Penghitung diperbarui ketika buffer Jitter penuh dan paket tidak dapat didorong ke dalamnya. Ini dapat digunakan untuk menghitung persentase paket yang dibuang dalam durasi tetap.	
BingkaiDijatuhkan	Nilai ini diperbarui saat <code>onFrameDroppedFunc()</code> dipanggil.	
lastPacketReceivedStempel waktu	Merupakan stempel waktu di mana paket terakhir diterima untuk SSRC ini.	
headerBytesReceived	Penghitung diperbarui saat menerima paket RTP.	
BytesDiterima	Jumlah byte yang diterima. Ini tidak termasuk byte header. Metrik ini dapat digunakan untuk menghitung bit rate yang masuk.	
packetsFailedDecryption	Ini bertambah ketika dekripsi paket SRTP gagal.	

Saluran Data

Metrik Saluran Data

Metrik	Deskripsi	
label	Label adalah nama saluran data yang sedang diperiksa.	

Metrik	Deskripsi	
protokol	Karena tumpukan kami menggunakan SCTP, protokol diatur ke SCTP konstan.	
dataChannelIdentifier	Pengidentifikasi genap atau ganjil digunakan untuk mengidentifikasi saluran data secara unik. Ini diperbarui ke nilai ganjil jika SDK adalah penawar dan nilai genap jika SDK adalah penjawab.	
negara	Status saluran data saat statistik ditanyakan. Saat ini, dua status yang didukung adalah RTC_DATA_CHANNEL_STATE_CONNECTING (saat saluran dibuat) dan RTC_DATA_CHANNEL_STATE_OPEN (Set dalam acara onOpen ()).	
PesanSent	Penghitung diperbarui saat SDK mengirim pesan melalui saluran data.	
olehTessent	Penghitung diperbarui dengan byte dalam pesan yang dikirim. Ini dapat digunakan untuk memahami berapa banyak byte yang tidak dikirim karena kegagalan, yaitu, untuk memahami persentase byte yang dikirim.	

Metrik	Deskripsi	
PesanDiterima	Metrik bertambah dalam <code>onMessage()</code> callback.	
BytesDiterima	Metrik dihasilkan dalam <code>onMessage()</code> callback.	

Amazon Kinesis Video Streams konsumsi WebRTC

Amazon Kinesis Video Streams menawarkan kemampuan untuk melakukan streaming video dan audio secara real-time melalui WebRTC ke cloud untuk penyimpanan, pemutaran, dan pemrosesan analitis. Pelanggan dapat menggunakan WebRTC SDK dan cloud API kami yang disempurnakan untuk mengaktifkan streaming real-time, serta konsumsi media ke cloud.

Untuk memulai, Anda dapat menginstal Amazon Kinesis Video [Streams dengan](#) WebRTC SDK pada kamera atau perangkat AWS IoT keamanan apa pun dengan sensor video [dan menggunakan API kami untuk mengaktifkan streaming media dengan latensi sub 1 detik, serta konsumsi dan](#) penyimpanan di cloud. Setelah tertelan, Anda dapat mengakses data Anda melalui easy-to-use API kami. Amazon Kinesis Video Streams memungkinkan Anda memutar video untuk ditonton langsung dan sesuai permintaan, serta dengan cepat membangun aplikasi yang memanfaatkan visi komputer dan analitik video melalui integrasi dengan Amazon Rekognition Video dan SageMaker

Topik

- [Operasi API](#)
- [Memulai dengan konsumsi dan penyimpanan WebRTC](#)
- [Membuat Kinesis Video Streams dengan WebRTC Signaling Channel](#)
- [Buat aliran](#)
- [Konfigurasi konsumsi dan penyimpanan media](#)
- [Menelan media](#)
- [Lihat media dari aliran Kinesis Video Streams](#)

Operasi API

Gunakan operasi API berikut untuk mengonfigurasi konsumsi Amazon Kinesis Video Streams WebRTC:

- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [JoinStorageSession](#)
- [UpdateMediaStorageConfiguration](#)

Memulai dengan konsumsi dan penyimpanan WebRTC

Amazon Kinesis Video Streams menawarkan kemampuan untuk melakukan streaming video dan audio secara real-time melalui WebRTC ke cloud untuk penyimpanan, pemutaran, dan pemrosesan analitis. Topik ini akan memberikan step-by-step instruksi untuk menyiapkan dan menggunakan SDK WebRTC dan API cloud kami untuk mengaktifkan streaming real-time dan konsumsi media ke cloud. Instruksi ini mencakup panduan untuk menggunakan AWS Command Line Interface dan konsol Kinesis Video Streams.

Sebelum Anda menggunakan Amazon Kinesis Video Streams dengan WebRTC untuk pertama kalinya, lihat. [the section called “Mengatur sebuah Akun AWS”](#)

Membuat Kinesis Video Streams dengan WebRTC Signaling Channel

Ada dua cara untuk membuat saluran pensinyalan, AWS Management Console atau. AWS CLI

Buat saluran pensinyalan menggunakan AWS Management Console

1. DiAWS Management Console, buka konsol [Kinesis Video Streams](#).
2. Di navigasi kiri, pilih Saluran pensinyalan.
Pilih Buat saluran pensinyalan.
3. Pada halaman Buat saluran pensinyalan baru, ketikkan nama untuk saluran pensinyalan.
Biarkan nilai T ime-to-live (Ttl) default sebagai 60 detik.
4. Pilih Buat saluran pensinyalan.
5. Setelah saluran pensinyalan dibuat, tinjau detail di halaman detail saluran.
Catat saluran ARN.

Buat saluran pensinyalan menggunakan AWS CLI

DalamAWS CLI, ketik:

```
$ aws kinesismvideo create-signaling-channel --channel-name your-channel-name --region us-west-2
```

Buat aliran

Ada dua cara untuk membuat aliran, yaitu AWS Management Console atau AWS CLI.

Important

WebRTC Ingestion memerlukan aliran video Amazon Kinesis dengan retensi data diaktifkan.

Buat aliran menggunakan AWS Management Console

1. Di AWS Management Console, buka konsol [Kinesis Video Streams](#).
2. Di navigasi kiri, pilih Dasbor.

Pilih Buat aliran video.

3. Pada halaman Buat aliran video baru, ketikkan nama untuk aliran pengujian ini.

Gunakan konfigurasi default.

4. Pilih Buat aliran video.
5. Setelah streaming dibuat, tinjau detailnya di halaman Video Streams.

Catat aliran ARN.

Buat aliran menggunakan AWS CLI

Dalam AWS CLI, ketik:

```
$ aws kinesismedia create-stream --stream-name your-stream-name --region us-west-2 --  
data-retention-in-hours 24
```

Konfigurasi konsumsi dan penyimpanan media

Dalam AWS CLI, atur konfigurasi penyimpanan media.

Jika Anda menggunakan file AWS Management Console untuk mengonfigurasi aliran dan/atau saluran, salin dan tempel ARN sumber daya dari langkah-langkah tersebut.

Jika Anda menggunakan AWS CLI untuk mengonfigurasi aliran dan/atau saluran Anda, lakukan hal berikut untuk mengambil ARNS sumber daya.

- Untuk mengambil saluran ARN, ketik:

```
$ aws kinesismedia describe-signaling-channel --channel-name your-channel-name --region us-west-2
```

- Untuk mengambil arus ARN, ketik:

```
$ aws kinesismedia describe-stream --stream-name your-stream-name --region us-west-2
```

Saat Anda memiliki ARN, atur konfigurasi penyimpanan media. Jenis:

```
$ aws kinesismedia update-media-storage-configuration \
  --channel-arn your-arn \
  --media-storage-configuration \
    StreamARN="your-stream-arn",Status="ENABLED" \
  --region us-west-2
```

Menelan media

Batasan berikut sudah ada:

- Durasi sesi: satu jam, maksimal
- Saluran pensinyalan: maksimum 100 per akun dengan konfigurasi penyimpanan diaktifkan

Menelan media dari browser

Important

Chrome adalah satu-satunya browser yang didukung.

1. [Buka Amazon Kinesis Video Streams dengan WebRTC SDK di halaman contoh. JavaScript](#)
2. Lengkapi informasi berikut:
 - Titik Akhir KVS. Di bidang Region, pilih wilayah Anda.

Misalnya, `us-west-2`.

- AWSKredensialnya

Lengkapi bidang-bidang berikut:

- ID Kunci Akses
 - Kunci Akses Rahasia
 - Token Sesi. Aplikasi sampel mendukung kredensial sementara dan jangka panjang. Biarkan bidang ini kosong jika Anda menggunakan kredensial IAM jangka panjang.
 - Saluran Pensinyalan. Di bidang Nama Saluran, ketikkan nama saluran pensinyalan yang Anda konfigurasi sebelumnya. Untuk informasi selengkapnya, lihat [the section called “Konfigurasi konsumsi dan penyimpanan media”](#).
 - Trek. Pilih Kirim Video dan Kirim Audio.
 - WebRTC Tertelan dan Penyimpanan. Pilih Ingest dan rekan penyimpanan bergabung secara otomatis.
3. Pilih Mulai Master.

Jika saluran pensinyalan dikonfigurasi untuk konsumsi menggunakan [DescribeMediaStorageConfiguration](#) API, aplikasi sampel akan secara otomatis memanggil [JoinStorageSession](#) API untuk memulai alur kerja penyerapan WebRTC.

Menelan media dari WebRTC C SDK

Ikuti [the section called “WebRTC SDK di C untuk Perangkat Tertanam”](#) prosedur untuk membangun aplikasi sampel.

Sampel master

Arahkan ke `build` folder dan ketik:


```
./samples/kvsWebrtcClientMaster channel-name 1
```

Sampel master GStreamer

Arahkan ke `build` folder dan ketik:

```
./samples/kvsWebrtcClientMasterGstSample channel-name audio-video-storage testsrc
```

Ini memulai konsumsi WebRTC.

 Note

Saluran pensinyalan yang disediakan harus dikonfigurasi untuk penyimpanan. Gunakan [DescribeMediaStorageConfiguration](#) API untuk mengonfirmasi.

Lihat media dari aliran Kinesis Video Streams

1. Buka [Penampil Media Streams Video Amazon Kinesis](#).
2. Lengkapi bidang-bidang berikut:
 - Wilayah. Pilih us-barat-2.
 - AWSKunci Akses
 - AWSKunci Rahasia
 - Nama aliran
 - Mode Pemutaran. Pilih Langsung.
3. Pilih Mulai Pemutaran.

Keamanan

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

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

- Keamanan cloud – AWS berfungsi melindungi infrastruktur yang menjalankan layanan AWS Cloud AWS. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Keefektifan keamanan kami diuji dan diverifikasi secara berkala oleh auditor pihak ketiga sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari program kepatuhan yang berlaku untuk Kinesis Video Streams, lihat [AWS Layanan dalam Cakupan menurut Program Kepatuhan](#).
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas 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 dengan WebRTC. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Kinesis Video Streams dengan WebRTC untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga akan mempelajari cara menggunakan AWS layanan lain yang dapat membantu Anda memantau dan mengamankan Kinesis Video Streams Anda dengan sumber daya WebRTC Anda.

Topik

- [Mengontrol Akses ke Kinesis Video Streams dengan Menggunakan Sumber Daya WebRTC AWS Identity and Access Management](#)
- [Validasi kepatuhan untuk Amazon Kinesis Video Streams dengan WebRTC](#)
- [Ketahanan dalam Kinesis Video Streams dengan WebRTC](#)
- [Keamanan Infrastruktur dalam Aliran Video Kinesis dengan WebRTC](#)
- [Praktik Terbaik Keamanan untuk Kinesis Video Streams dengan WebRTC](#)

Mengontrol Akses ke Kinesis Video Streams dengan Menggunakan Sumber Daya WebRTCAWS Identity and Access Management

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

Untuk informasi selengkapnya tentang IAM, lihat hal berikut:

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

Daftar Isi

- [Sintaks Kebijakan](#)
- [Tindakan untuk Kinesis Video Streams dengan WebRTC](#)
- [Nama Sumber Daya Amazon \(ARN\) untuk Kinesis Video Streams](#)
- [Memberikan Akses Akun IAM Lainnya ke Aliran Video Kinesis](#)
- [Contoh Kebijakan untuk Kinesis Video Streams dengan WebRTC](#)

Sintaks 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:** Efek bisa berupa Allow atau Deny. Secara default, para pengguna IAM tidak memiliki izin untuk menggunakan sumber daya dan tindakan API, jadi semua permintaan akan ditolak. izin eksplisit akan menggantikan izin default. penolakan eksplisit akan menggantikan izin apa pun.
- **Tindakan:** Tindakan adalah tindakan API tertentu yang Anda izinkan atau tolak.
- **Sumber Daya:** Sumber daya yang dipengaruhi oleh tindakan. Untuk menentukan sumber daya dalam sebuah pernyataan kebijakan IAM, gunakan Amazon Resource Name (ARN).
- **Syarat:** Syarat bersifat opsional. Syarat-syarat ini dapat digunakan untuk mengendalikan kapan kebijakan Anda berlaku.

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

Tindakan untuk Kinesis Video Streams dengan WebRTC

Dalam pernyataan kebijakan IAM, Anda dapat menentukan tindakan API apa pun dari layanan apa pun yang mendukung IAM. Untuk Kinesis Video Streams dengan WebRTC, gunakan prefiks berikut ini dengan nama dari tindakan API: `kinesisvideo:`. Misalnya: `kinesisvideo:CreateSignalingChannel`, `kinesisvideo:ListSignalingChannels`, dan `kinesisvideo:DescribeSignalingChannel`.

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 "Get" sebagai berikut:

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

Untuk menentukan semua operasi Kinesis Video Streams, gunakan tanda bintang (*) sebagai berikut ini:

```
"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:channel/channel-name/code
```

Misalnya:

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

Anda bisa mendapatkan ARN saluran menggunakan [DescribeSignalingChannel](#).

Memberikan Akses Akun IAM Lainnya ke Aliran Video Kinesis

Anda mungkin perlu memberikan izin ke akun IAM lain untuk melakukan operasi di Kinesis Video Streams dengan saluran sinyal WebRTC. Peran layanan adalah [IAM role](#) yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin keLayanan AWS](#) dalam Panduan Pengguna IAM.

Contoh Kebijakan untuk Kinesis Video Streams dengan WebRTC

Contoh kebijakan berikut menunjukkan bagaimana Anda dapat mengontrol akses pengguna ke Kinesis Video Streams Anda dengan saluran WebRTC.

Example 1: Memungkinkan pengguna mendapatkan data dari saluran pensinyalan

Kebijakan ini memungkinkan pengguna atau grup untuk melakukan `DescribeSignalingChannel`, `GetSignalingChannelEndpointListSignalingChannel` dan `ListTagsForResource` operasi pada saluran pensinyalan apa pun.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "kinesisvideo:Describe*",
          "kinesisvideo:Get*",
          "kinesisvideo:List*"
        ],
        "Resource": "*"
      }
    ]
  }

```

Example 2: Memungkinkan pengguna untuk membuat saluran pensinyalan

Kebijakan ini memungkinkan pengguna atau grup untuk melakukan `CreateSignalingChannel` operasi.

```

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

```

Example 3: Memungkinkan pengguna akses penuh ke semua Kinesis Video Streams dan Kinesis Video Streams dengan sumber daya WebRTC

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: Memungkinkan pengguna mendapatkan data dari saluran pensinyalan tertentu

Kebijakan ini memungkinkan pengguna atau grup untuk mendapatkan data dari saluran pensinyalan tertentu.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:DescribeSignalingChannel",
      "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:channel/channel_name/0123456789012"
    }
  ]
}

```

Validasi kepatuhan untuk Amazon Kinesis Video Streams dengan WebRTC


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

Anda bisa 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, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Quick Start Keamanan dan Kepatuhan](#) – Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.

- [Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

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

- [Sumber Daya Kepatuhan AWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Panduan Kepatuhan Pelanggan AWS](#) – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- [AWS Security Hub](#) – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda dan untuk memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#) – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Ketahanan dalam Kinesis Video Streams dengan WebRTC

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Anda dapat menggunakan Zona Ketersediaan untuk merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara Zona Ketersediaan tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, toleran terhadap kegagalan, dan dapat diukur skalanya jika dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

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

Keamanan Infrastruktur dalam Aliran Video Kinesis dengan WebRTC

Sebagai layanan terkelola, Kinesis Video Streams (termasuk kemampuan WebRTC-nya) dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper [Amazon Web Services: Gambaran Umum 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. Kami merekomendasikan TLS 1.3 atau yang lebih baru. Klien juga harus mendukung suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

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

Praktik Terbaik Keamanan untuk Kinesis Video Streams dengan WebRTC

Amazon Kinesis Video Streams (termasuk kemampuan WebRTC-nya) menyediakan sejumlah fitur keamanan untuk dipertimbangkan ketika Anda menyusun 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 cukup untuk lingkungan Anda, anggap praktik terbaik tersebut sebagai pertimbangan yang membantu dan bukan sebagai rekomendasi.

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

Terapkan akses hak istimewa yang paling rendah

Saat memberikan izin, Anda memutuskan siapa yang mendapatkan izin yang menjadi sumber daya Kinesis Video Streams yang mana saja. Anda mengaktifkan tindakan tertentu yang ingin Anda

izinkan pada 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 hanya membutuhkan `PutMedia`, `GetStreamingEndpoint`, dan `DescribeStream`. Jangan memberikan izin aplikasi produser untuk semua tindakan (*), atau untuk tindakan lain seperti `GetMedia`.

Untuk informasi lebih lanjut, lihat [Apa itu Hak Istimewa Paling Sedikit & Mengapa Anda Membutuhkannya?](#)

Gunakan IAM role

Aplikasi produser dan klien harus memiliki kredensi yang valid untuk mengakses aliran video Kinesis. Anda tidak boleh menyimpan kredensial AWS secara langsung di aplikasi klien atau bucket Amazon S3. Ini adalah kredensial jangka panjang yang tidak dirotasi secara otomatis dan dapat menimbulkan dampak bisnis yang signifikan jika dibobol.

Sebaliknya, Anda harus menggunakan peran IAM untuk mengelola kredentik sementara untuk aplikasi produser dan klien Anda untuk mengakses aliran video Kinesis. Saat Anda menggunakan peran, Anda tidak perlu menggunakan kredentik jangka panjang untuk mengakses sumber daya lainnya.

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 dengan WebRTC terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Kinesis Video Streams dengan WebRTC.

Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Kinesis Video Streams dengan WebRTC, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk informasi selengkapnya, lihat [the section called “Mencatat Kinesis Video Streams dengan Panggilan API WebRTC dengan AWS CloudTrail”](#).

Memantau

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan kinerja Amazon Kinesis Video Streams dengan WebRTC dan AndaAWSsolusi. Anda harus mengumpulkan data pemantauan dari semua bagian solusi AWS sehingga Anda dapat melakukan debug kegagalan multititik secara lebih mudah jika terjadi kegagalan. Namun sebelum Anda mulai memantau Kinesis Video Streams, Anda harus membuat rencana pemantauan yang mencakup jawaban atas pertanyaan berikut:

- Apa sasaran 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 menetapkan tujuan pemantauan Anda dan membuat rencana pemantauan, langkah berikutnya adalah menetapkan baseline untuk Streams Kinesis Video Streams normal dengan performa WebRTC di lingkungan Anda. Anda harus mengukur Kinesis Video Streams dengan performa WebRTC pada berbagai waktu dan dalam kondisi beban yang berbeda. Saat memantau Kinesis Video Streams dengan WebRTC, Anda harus menyimpan riwayat data pemantauan yang telah Anda kumpulkan. Anda dapat membandingkan Kinesis Video Streams saat ini dengan performa WebRTC dengan data historis ini untuk membantu Anda mengidentifikasi pola performa normal dan anomali performa, dan merancang metode untuk mengatasi masalah yang mungkin timbul.

Topik

- [Pemantauan Kinesis Video Streams denganCloudWatch](#)
- [Mencatat Kinesis Video Streams dengan Panggilan API WebRTC dengan AWS CloudTrail](#)

Pemantauan Kinesis Video Streams denganCloudWatch

Anda dapat memantau Kinesis Video Streams dengan WebRTC menggunakan AmazonCloudWatch, yang mengumpulkan dan memproses data mentah dari Kinesis Video Streams dengan WebRTC menjadi metrik hampir waktu nyata yang dapat dibaca. Statistik ini dicatat untuk jangka waktu 15

bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang bagaimana kinerja aplikasi atau layanan web Anda.

Kinesis Video Streams menyediakan metrik berikut:

Topik

- [Metrik Sinyal](#)
- [Metrik TURN](#)

Metrik Sinyal

Nama metrik	Dimensi	Unit	Deskripsi
Gagal	Operasi, SignalingChannelName	Count	'0' dipancarkan jika Operasi disebutkan dalam dimensi mengembalikan 200 respon kode status. '1' sebaliknya.
Latensi	Operasi, SignalingChannelName	Milidetik	Waktu yang diukur dari saat layanan menerima permintaan sampai layanan mengembalikan respons.
MessagesTransferredCOUNT	SignalingChannelName	Count	Jumlah total pesan yang ditransfer (dikirim dan diterima) untuk saluran tertentu.

ParameterOperationDimensi dapat berlaku untuk salah satu API berikut:

- ConnectAsTuan
- ConnectAsOrang yang melihat
- SendSdpPenawaran
- SendSdpJawaban
- SendCandidate
- SendAlexaOfferToTuan

- GetIceServerConfig
- MEMUSKAN

Metrik TURN

Nama metrik	Dimensi	Unit	Deskripsi
BELOKConn ectedMinu tes	Signaling ChannelNa ma	Count	'1' dipancarkan untuk setiap alokasi TURN yang digunakan untuk mengalirkan data melalui dalam satu menit.

Mencatat Kinesis Video Streams dengan Panggilan API WebRTC dengan AWS CloudTrail

Amazon Kinesis Video Streams dengan WebRTC terintegrasi dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan AWS di Amazon Kinesis Video Streams dengan WebRTC AWS CloudTrail. CloudTrail menangkap semua panggilan API untuk Amazon Kinesis Video Streams dengan WebRTC sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Amazon Kinesis Video Streams dan panggilan kode ke Amazon Kinesis Video Streams dengan operasi WebRTC API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Amazon Kinesis Video Streams dengan WebRTC. 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 ke Amazon Kinesis Video Streams dengan WebRTC, 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 dengan WebRTC dan CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas peristiwa yang didukung terjadi di Amazon Kinesis Video Streams dengan WebRTC, aktivitas tersebut CloudTrail direkam dalam suatu peristiwa bersama dengan peristiwa layanan lainnya dalam riwayat Acara. AWS

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 dengan WebRTC, 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 AWS layanan lain 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](#)
- [Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Amazon Kinesis Video Streams dengan WebRTC mendukung pencatatan tindakan berikut sebagai peristiwa dalam file log: CloudTrail

- [CreateSignalingChannel](#)
- [DeleteSignalingChannel](#)
- [DescribeSignalingChannel](#)
- [GetSignalingChannelEndpoint](#)
- [ListSignalingChannels](#)
- [ListTagsForResource](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateSignalingChannel](#)

Setiap peristiwa atau entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut:

- Apakah permintaan itu dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).

- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen [CloudTrail UserIdentity](#).

Contoh: Amazon Kinesis Video Streams dengan Entri File Log WebRTC

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 mewakili permintaan tunggal dari sumber mana pun dan mencakup 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 [CreateSignalingChannel](#) tindakan.

```
{
  "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": "2019-11-19T22:49:04Z",
  "eventSource": "kinesisvideo.amazonaws.com",
  "eventName": "CreateSignalingChannel",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
  "requestParameters": {
    "channelName": "YourChannelName"
  },
  "responseElements": {
    "channelARN": "arn:aws:kinesisvideo:us-west-2:123456789012:channel/YourChannelName/1574203743620"
  },
}
```

```
"requestID": "df3c99c4-1d97-49da-8569-7de6c92b4856",  
"eventID": "bb74bac2-964c-49b0-903a-3501c6bde632"  
}
```

Memecahkan Masalah Amazon Kinesis Video Streams dengan WebRTC

Gunakan informasi berikut untuk memecahkan masalah umum yang mungkin Anda alami dengan Amazon Kinesis Video Streams dengan WebRTC.

Topik

- [Kuota layanan](#)
- [Persyaratan jaringan](#)
- [Lingkungan jaringan](#)
- [Masalah membangun sesi](#)
- [Debugging koneksi yang sedang berlangsung](#)

Kuota layanan

Anda hanya dapat terhubung satu master dan satu atau lebih pemirsa ke saluran pensinyalan tunggal.

Pada Februari 2023, tidak mungkin untuk menghubungkan beberapa master ke satu saluran pensinyalan. Untuk informasi tambahan tentang kuota layanan, lihat [Kuota](#).

Persyaratan jaringan

Persyaratan jaringan umum untuk endpoint layanan saluran sinyal untuk Kinesis Video Streams dengan WebRTC adalah:

- Panggilan HTTPS ke titik akhir yang dihosting di `https://*.kinesisvideo.{region}.amazonaws.com`
- WebSocket integrasi dengan titik akhir `wss://*.kinesisvideo.{region}.amazonaws.com`
- STUN server `stun:stun.kinesisvideo.{aws-region}.amazonaws.com:443`
- TURN server `turn:._.kinesisvideo.{aws-region}.amazonaws.com:443` dan `turns:._.kinesisvideo.{aws-region}.amazonaws.com:443`

Protokol yang digunakan antara rekan-rekan sebagai bagian dari `RTCPeerConnection` dapat berupa TCP atau UDP berbasis.

Sebagian besar aplikasi mencoba untuk membangun langsung, peer-to-peer koneksi dengan menentukan alamat IP untuk setiap rekan, serta port dan protokol yang akan ditukar sebagai kandidat ICE. Kandidat ini digunakan untuk mencoba untuk terhubung satu sama lain menggunakan kandidat ini. Mereka akan mencoba setiap pasangan sampai koneksi dapat dibuat.

Lingkungan jaringan

Jika pesan dari pemirsa telah dikirim ke master dan log seperti `No valid ice candidate` dicatat, maka tidak ada rute koneksi yang valid telah ditemukan. Ini bisa terjadi jika ada firewall yang mencegah koneksi langsung, atau jika jaringan tidak dapat dijangkau.

Lakukan hal berikut untuk memecahkan masalah konektivitas:

- Jika Anda tidak menggunakan TURN di sisi master, pastikan untuk mengaktifkan TURN.

TURN diaktifkan secara default di C SDK. Dalam JavaScript SDK, pilih `STUN/TURN` atau `TURN only` di NAT Traversal.

- Untuk jaringan terbatas, seperti jaringan perusahaan, coba jaringan lain yang tersedia (kabel atau nirkabel).

Jika Anda terhubung ke VPN, putus sambungan dari VPN tersebut.

Note

Anda dapat mengabaikan `403 Forbidden` IP kesalahan yang dikembalikan oleh Kinesis Video Streams TURN server. Server menolak pasangan kandidat ICE yang berisi `localhost` IP, seperti `192.168.*` atau `10.0.0.*`. Ini dapat menyebabkan beberapa, tetapi tidak semua, pasangan kandidat ICE gagal.

Masalah membangun sesi

WebRTC dapat membantu meringankan masalah yang terjadi karena:

- Terjemahan alamat jaringan (NAT)

- firewall
- Proxy antar teman sebaya

WebRTC menyediakan kerangka kerja untuk membantu bernegosiasi dan memelihara koneksi selama rekan-rekan terhubung. Ini juga menyediakan mekanisme untuk menyampaikan media melalui aTURNserver dalam kasus peer-to-peer koneksi tidak bisa dinegosiasikan.

Mengingat semua komponen yang diperlukan untuk membangun koneksi, ada baiknya memahami beberapa alat yang tersedia untuk membantu memecahkan masalah yang terkait dengan membangun sesi.

Topik

- [Penawaran dan jawaban Protokol Deskripsi Sesi \(SDP\)](#)
- [Evaluasi generasi kandidat ICE](#)
- [Tentukan kandidat mana yang digunakan untuk menjalin hubungan](#)

Penawaran dan jawaban Protokol Deskripsi Sesi (SDP)

Session Description Protocol (SDP) menawarkan dan jawaban menginisialisasi sesi RTC antara rekan-rekan.

Untuk mempelajari lebih lanjut tentang protokol SDP, lihat [spesifikasi](#).

- Penawaran dihasilkan oleh “pemirsa” yang ingin terhubung ke rekan-rekan yang terhubung ke saluran sinyal sebagai “master” di Kinesis Video Streams dengan WebRTC.
- Jawaban dihasilkan oleh penerima penawaran.

Baik penawaran maupun jawaban dihasilkan di sisi klien, meskipun mereka mungkin berisi kandidat ICE yang telah dikumpulkan hingga saat itu.

Yang [Kinesis Video Streaming WebRTC SDK untuk C](#) termasuk variabel lingkungan sederhana yang dapat Anda atur untuk log SDP. Ini berguna untuk memahami penawaran yang diterima dan jawaban yang dihasilkan.

Untuk log SDP `stdout` dari SDK, atur variabel lingkungan berikut: `export DEBUG_LOG_SDP=TRUE`. Anda juga dapat mencatat penawaran dan jawaban SDP JavaScript berbasis klien menggunakan `sdpOffer` cara. Untuk melihat ini ditunjukkan, lihat [GitHub](#).

Untuk informasi tambahan, lihat [the section called “Pemantauan Kinesis Video Streams dengan CloudWatch”](#).

Jika jawaban SDP tidak dikembalikan, ada kemungkinan bahwa rekan tidak dapat menerima penawaran SDP, karena penawaran tersebut tidak mengandung codec media yang kompatibel. Anda mungkin melihat log yang mirip dengan berikut ini:

```
I/webrtc_video_engine.cc: (line 808): SetSendParameters: {codecs:
  [VideoCodec[126:H264]], conference_mode: no, extensions: [], extmap-allow-mixed:
  false, max_bandwidth_bps: -1, mid: video1}
E/webrtc_video_engine.cc: (line 745): No video codecs supported.
E/peer_connection.cc: (line 6009): Failed to set remote video description send
  parameters for m-section with mid='video1'. (INVALID_PARAMETER)
E/peer_connection.cc: (line 3097): Failed to set remote offer sdp: Failed to set remote
  video description send parameters for m-section with mid='video1'.
E/KinesisVideoSdpObserver: onSetFailure(): Error=Failed to set remote offer sdp: Failed
  to set remote video description send parameters for m-section with mid='video1'.
D/KVSWebRtcActivity: Received SDP offer for client ID: null. Creating answer
E/peer_connection.cc: (line 2373): CreateAnswer: Session error code: ERROR_CONTENT.
  Session error description: Failed to set remote video description send parameters for
  m-section with mid='video1'..
E/KinesisVideoSdpObserver: onCreateFailure(): Error=Session error code: ERROR_CONTENT.
  Session error description: Failed to set remote video description send parameters for
  m-section with mid='video1'..
```

Saat Anda meninjau konten penawaran SDP, cari baris yang dimulai dengan `a=rtpmap:` untuk melihat codec media mana yang diminta.

```
...
a=rtpmap:126 H264/90000
...
a=rtpmap:111 opus/48000/2
...
```

Evaluasi generasi kandidat ICE

Kandidat ICE dihasilkan oleh setiap klien yang melakukan panggilan ke STUN peladen. Untuk Kinesis Video Streams dengan WebRTC, STUN server adalah `stun:stun.kinesisvideo.{aws-region}.amazonaws.com:443`.

Selain memanggil STUN server untuk mendapatkan kandidat, klien sering juga memanggil TURN server. Mereka membuat panggilan ini sehingga server relay dapat digunakan sebagai fallback dalam kasus langsung peer-to-peer koneksi tidak dapat dibuat.

Anda dapat menggunakan alat berikut untuk menghasilkan kandidat ICE:

- SEBUAH [Menetes ICE WebRTC sampel](#), yang menggunakan Trickle ICE untuk mengumpulkan kandidat
- [IceTest.Info](#)

Dengan kedua alat ini, Anda dapat memasukkan STUN dan TURN informasi server untuk mengumpulkan kandidat.

Untuk mendapatkan TURN informasi server dan mandat yang diperlukan untuk Kinesis Video Streams dengan WebRTC, Anda dapat menghubungi [GetIceServerConfig Operasi API](#).

Berikut ini AWS CLI panggilan menunjukkan bagaimana untuk mendapatkan informasi ini untuk digunakan dalam dua alat.

```
export CHANNEL_ARN="YOUR_CHANNEL_ARN"

aws kinesishvideo get-signaling-channel-endpoint \
  --channel-arn $CHANNEL_ARN \
  --single-master-channel-endpoint-configuration Protocols=WSS,HTTPS,Role=MASTER
```

Output dari [get-signaling-channel-endpoint](#) perintah mengembalikan respon yang terlihat seperti ini:

```
{
  "ResourceEndpointList": [
    {
      "Protocol": "HTTPS",
      "ResourceEndpoint": "https://your-endpoint.kinesisvideo.us-east-1.amazonaws.com"
    },
    {
      "Protocol": "WSS",
      "ResourceEndpoint": "wss://your-endpoint.kinesisvideo.us-east-1.amazonaws.com"
    }
  ]
}
```

Gunakan `HTTPSResourceEndpoint` nilai untuk mendapatkan daftar `TURN` server sebagai berikut:

```
export ENDPOINT_URL="https://your-endpoint.kinesisvideo.us-east-1.amazonaws.com"

aws kinesis-video-signaling get-ice-server-config \
  --channel-arn $CHANNEL_ARN \
  --service TURN \
  --client-id my-amazing-client \
  --endpoint-url $ENDPOINT_URL
```

Respons berisi `TURN` rincian server, termasuk titik akhir untuk TCP dan UDP dan kredensial yang diperlukan untuk mengaksesnya.

Note

Nilai TTL dalam respons menentukan durasi, dalam hitungan detik, bahwa kredensial ini valid untuk. Gunakan nilai-nilai ini di [Menetes ICE WebRTC sampel](#) atau di [IceTest.Info](#) untuk menghasilkan kandidat ICE menggunakan titik akhir layanan yang dikelola Kinesis Video Streams.

Tentukan kandidat mana yang digunakan untuk menjalin hubungan

Hal ini dapat membantu untuk memahami kandidat mana yang digunakan untuk berhasil membangun sesi. Jika Anda memiliki klien berbasis browser yang menjalankan sesi yang sudah mapan, Anda dapat menentukan informasi ini di Google Chrome dengan menggunakan utilitas internal `webrtc-internal` bawaan.

Buka sesi WebRTC di satu tab browser.

Di tab lain, buka `chrome://webrtc-internals/`. Anda dapat melihat semua informasi tentang sesi Anda yang sedang berlangsung di tab ini.

Anda akan melihat informasi tentang koneksi yang mapan. Misalnya:

► Create Dump

Read stats From: (Standardized (promise-based) getStats() API)

Note: computed stats are in []. Experimental stats are marked with an * at the end and do not show up in the getStats result.

GetUserMedia Requests

<https://aws-labs.github.io/amazon-kinesis-video-streams-webrtc-sdk-js/examples/index.html#id=6025>

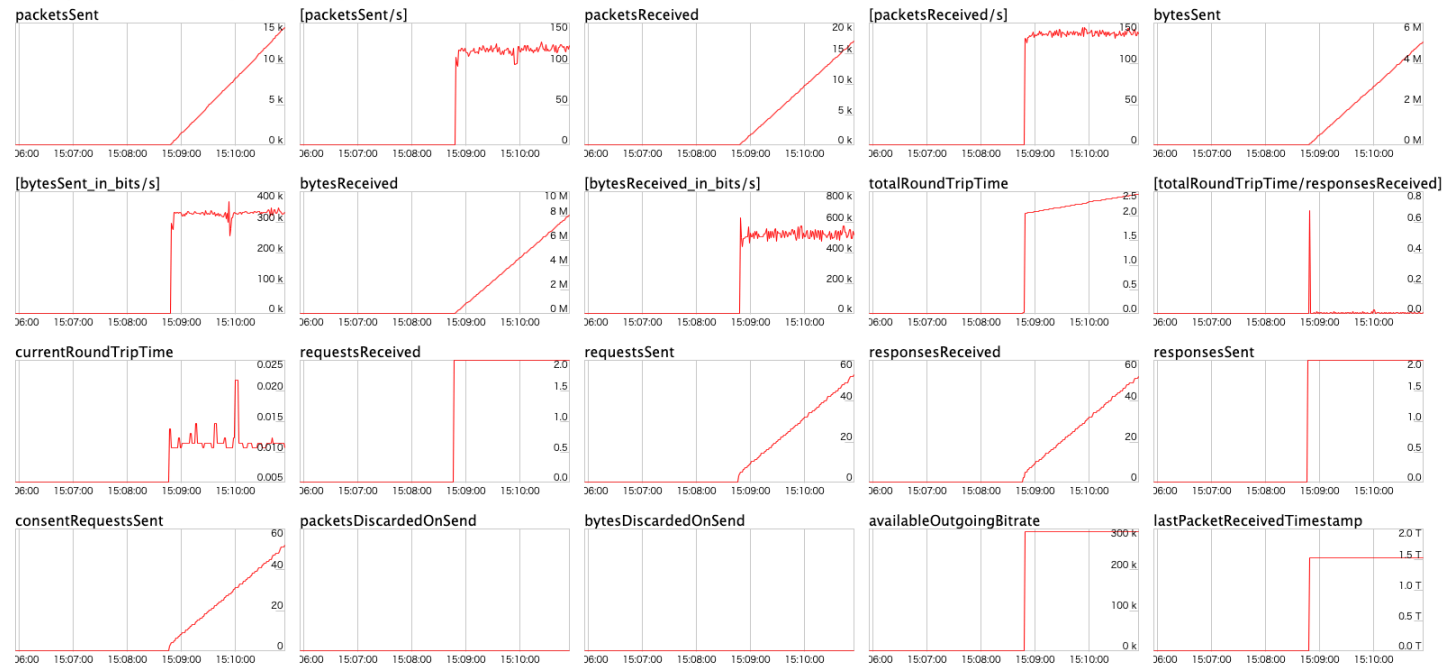
```
https://aws-labs.github.io/amazon-kinesis-video-streams-webrtc-sdk-js/examples/index.html, { iceServers: [stun:stun.kinesisvideo.ap-northeast-1.amazonaws.com:443, turn:3-113-249-125.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=udp, turns:3-113-249-125.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=udp, turns:18-183-42-89.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=udp, turns:18-183-42-89.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=udp, turns:18-183-42-89.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=tcp, turns:18-183-42-89.t-6618366e.kinesisvideo.ap-northeast-1.amazonaws.com:443?transport=tcp], iceTransportPolicy: all, bundlePolicy: balanced, rtcMuxPolicy: require, iceCandidatePoolSize: 0 }
```

```
ICE connection state: new => checking => connected
Connection state: new => connecting => connected
Signaling state: new => have-local-offer => stable
ICE Candidate pair: :46950 <=> :35795
► ICE candidate grid
```

Stats Tables

Anda juga dapat mengonfirmasi metrik seperti berikut ini untuk koneksi yang telah ditetapkan.

▼ Stats graphs for candidate-pair (state=in-progress, id=CPaYGwieso_EsxKSWoY)



Debugging koneksi yang sedang berlangsung

Ada beberapa daerah yang dapat menyebabkan masalah dengan didirikan, koneksi WebRTC yang sedang berlangsung, tetapi jaringan adalah yang paling umum.

Anda dapat mengonfirmasi log tingkat VERBOSE dari SDK dengan menyetel `export AWS_KVS_LOG_LEVEL=1` sebagai variabel lingkungan.

Note

Jika tidak ada pasangan kandidat ditemukan dalam batas waktu yang ditentukan, agen ICE mengembalikan status kesalahan `0x5a00000d`.

Untuk informasi tambahan tentang tingkat log, lihat [GitHub](#).

```
export AWS_KVS_LOG_LEVEL=1
./kvsWebrtcClientMasterGstSample TestChannel
```

Anda akan melihat log seperti berikut ini. Dari log ini, Anda dapat mengonfirmasi kandidat Interactive Connectivity Establishment (ICE) (alamat IP dan port) dan pasangan kandidat yang dipilih.

```
2023-02-13 05:57:16 DEBUG iceAgentReadyStateSetup(): Selected pair w1UdV9fE+_/_
CuBellp1, local candidate type: srflx. Round trip time 7 ms. Local candidate priority:
1694498815, ice candidate pair priority: 7240977859836116990
2023-02-13 05:57:16 INFO onConnectionStateChange(): New connection state 3
2023-02-13 05:57:16 DEBUG rtcPeerConnectionGetMetrics(): ICE local candidate Stats
requested at 16762678365731494
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate IP
Address: XXX.XXX.XXX.XXX
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate
type: srflx
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate
port: 38563
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate
priority: 1694498815
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate
transport protocol: udp
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate
relay protocol: N/A
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Local Candidate Ice
server source: stun.kinesisvideo.ap-northeast-1.amazonaws.com
2023-02-13 05:57:16 DEBUG rtcPeerConnectionGetMetrics(): ICE remote candidate Stats
requested at 16762678365732111
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Remote Candidate IP
Address: XXX.XXX.XXX.XXX
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Remote Candidate
type: srflx
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Remote Candidate
port: 45867
2023-02-13 05:57:16 VERBOSE signalingClientGetCurrentState(): Signaling Client Get
Current State
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Remote Candidate
priority: 1685921535
2023-02-13 05:57:16 DEBUG logSelectedIceCandidatesInformation(): Remote Candidate
transport protocol: udp
```

Riwayat Dokumen untuk Amazon Kinesis Video Streams dengan Panduan Developer WebRTC

Perubahan	Deskripsi	Tanggal
Publikasi awal	Publikasi awal Amazon Kinesis Video Streams dengan Panduan Developer WebRTC. Pelajari selengkapnya	4 November 2019

AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.