aws

Amazon Kinesis Video Streams WebRTC 開發人員指南

Kinesis Video Streams

Kinesis Video Streams: Amazon Kinesis Video Streams WebRTC 開發人員指南

Table of Contents

什麼是 Amazon Kinesis Video Streams 與 WebRTC

WebRTC 是一種開放技術規格，可讓瀏覽器和行動應用程式之間透過簡單 API 進行即時通訊 (RTC)。它使用對等技術在連接的對等之間進行實時數據交換，並提供交 human-to-human 互所需的低延遲媒體流。WebRTC 規範包括一組 IETF 通訊協定，包括互動式連線建立、周遊 NAT 周遊轉送 (TURN)，以及用於建立連線的工作階段穿越公用程式 (STUN)，以及用於建立 peer-to-peer 連線能力的通訊協定規格，以及可靠且安全的即時媒體和資料串流的通訊協定規格。

Amazon Kinesis Video Streams 以全受管功能提供符合標準的 WebRTC 實作。您可以使用 Amazon Kinesis Video Streams with WebRTC 安全地即時串流媒體，或在任何攝影機 IoT 裝置與符合 WebRTC 標準的行動或 Web 播放器之間，執行雙向音訊或視訊互動。因為是全受管功能，您不需要建置、執行或擴展任何與 WebRTC 相關的雲端基礎設施，例如訊號或媒體轉送伺服器，即可在應用程式和裝置之間安全地串流媒體。

將 Kinesis Video Streams 與 WebRTC 搭配使用，您可以輕鬆建立即時 peer-to-peer 媒體串流的應用程式，或在攝影機 IoT 裝置、網頁瀏覽器和行動裝置之間的即時音訊或視訊互動，以適用於各種使用案例。這些應用程式可以幫助家長監控嬰兒的房間、讓屋主使用視訊門鈴檢查誰在門口、讓攝影功能掃地機器人的擁有者在手機上觀看即時攝影機串流，以從遠端控制機器人，諸如此類。

如果您是第一次使用 WebRTC 技術的 Kinesis Video Streams 使用者，我們建議您閱讀下列章節：

- Kinesis Video Streams Video StreWebRTC: 運作方式
- 適用於嵌入式裝置的 C WebRTC 開發套件
- 適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript
- 適用於 Android 的 WebRTC 開發套件
- 適用於 iOS 的 WebRTC 開發套件
- 控制平台 API
- 資料平面 REST API
- 資料平面 Websocket API

區域可用性

使用 WebRTC 技術的 Amazon Kinesis Video Streams 可在下列區域使用：

區域名稱	AWS區域代碼
美國東部 (俄亥俄)	us-east-2
美國東部 (維吉尼亞北部)	us-east-1
美國西部 (奧勒岡)	us-west-2
非洲 (開普敦)	af-south-1
亞太區域 (香港)	ap-east-1
亞太區域 (孟買)	ap-south-1
亞太區域 (首爾)	ap-northeast-2
亞太區域 (新加坡)	ap-southeast-1
亞太區域 (雪梨)	ap-southeast-2
亞太區域 (東京)	ap-northeast-1
加拿大 (中部)	ca-central-1
歐洲 (法蘭克福)	eu-central-1
歐洲 (愛爾蘭)	eu-west-1
歐洲 (倫敦)	eu-west-2
歐洲 (巴黎)	eu-west-3
南美洲 (聖保羅)	sa-east-1

Kinesis Video Streams 與 WebRTC 技術定價

如需使用 WebRTC 定價的 Kinesis Video Streams 的相關資訊，請參閱 Amazon Kinesis Video Streams 定價。

使用 WebRTC 技術存取 Kinesis Video Streams

您可以使用下列任何一種方式搭配 WebRTC 技術使用 Kinesis Video Streams：

AWS Management Console

AWS Management Console 入門

控制台是一個基於瀏覽器的界面，用於訪問和使用AWS服務，包括 Kinesis Video Streams 與 WebRTC。

AWS SDK

AWS 提供軟體開發套件 (SDK)，包含適用於各種程式設計語言和平台 (例如，Java、Python、Ruby、.NET、iOS、Android 等等) 的程式庫及範本程式碼。開發套件提供了一種方便的方式，可透過 WebRTC 技術建立 Kinesis Video Streams 的程式設計存取。如需 AWS 開發套件的其他資訊 (包括如何下載並安裝開發套件)，請參閱 Amazon Web Services 工具。

使用 WebRTC 技術 HTTPS API 的 Kinesis Video Streams

您可以使用 WebRTC 存取 Kinesis Video Streams，也可以透過AWS程式設計方式使用 Kinesis Video Streams 搭配 WebRTC 技術 API 來存取 Kinesis 視訊串流，讓您可以直接向服務發出 API 要求。如需詳細資訊，請參閱 Amazon Kinesis Video Streams API 參考。

Kinesis Video Streams Video StreWebRTC: 運作方式

主題

Amazon Kinesis Video Streams

以下是使用 WebRTC 技術的 Amazon Kinesis Video Streams 的特定關鍵術語和概念。

訊號頻道

可讓應用程式探索、設定、控制及終止 peer-to-peer 通過交換信令消息進行連接。信令消息是兩個應用程序相互交換以建立的元數據 peer-to-peer 連線能力。此中繼資料包括本機媒體資訊，例如媒體轉碼器和轉碼參數，以及兩個應用程式相互連線以進行即時串流可能使用的網路候選路徑。

串流應用程式可以維持與訊號頻道之間的持續連線，並等待其他應用程式來連接它們。或者，只在需要即時串流媒體時才連線到訊號頻道。信號通道可讓應用程式在 one-to-few 模型，使用一個概念主連接到多個觀眾。啟動連線的應用程式負起主節點的責任，使用 `ConnectAsMaster` API 並等待檢視器。然後，最多 10 個應用程式可以負起檢視器的責任，叫用 `ConnectAsViewer` API 來連線到該訊號頻道。將它們連接到信令通道後，主應用程序和查看器應用程序可以互相發送信令消息以建立 peer-to-peer 即時媒體串流的連線能力。

對等節點

任何裝置或應用程式 (例如，行動或網路應用程式、網路攝影機、居家安全攝影機、嬰兒監視器等)，設定為透過 Kinesis Video Streams with WebRTC 進行即時雙向串流。

主

啟動連線並連線至訊號頻道的對等節點，能夠探索任何連接訊號頻道的檢視器並與之交換媒體。

> **Important**
>
> 目前，一個訊號頻道只能有一個主節點。

檢視者

連線至訊號頻道的對等節點，只能夠探索訊號頻道的主節點並與之交換媒體。檢視器無法透過給定的訊號頻道來探索其他檢視器或與之互動。一個訊號頻道最多可以有 10 個連線的檢視器。

WebRTC 技術概念

當您開始使用 WebRTC 技術的 Kinesis Video Streams 時，您也可以從學習 WebRTC 技術所組成的數個相互關聯的通訊協定和 API 中受益。

Session Traversal Utilities for NAT (STUN)

一種通訊協定，用來探索您的公有地址，並判斷路由器中會阻止直接連接對等節點的任何限制。

Traversal Using Relays around NAT (TURN)

一種伺服器，可對 TURN 伺服器建立連線，並透過該伺服器轉送資訊，以避開對稱 NAT 限制。

Session Description Protocol (SDP)

一種標準，描述連線的多媒體內容，例如解析度、格式，轉碼器、加密等，讓對等節點在資料開始傳輸時可以了解彼此。

SDP 提議

由代理程式傳送的 SDP 訊息，以產生工作階段描述來建立或修改工作階段。描述所需媒體通訊的各方面。

SDP 回答

由回答者傳送的 SDP 訊息，以回應來自提議者的提議。回答指出接受哪些方面。例如，是否接受提議中的所有音訊和視訊串流。

互動式連線建立 (ICE)

允許 Web 瀏覽器與對等節點連接的框架。

ICE 候選項

可供傳送端對等節點用來通訊的方法。

STUN、TURN 和 ICE 如何一起運作

假設有 A 和 B 兩個對等節點，兩者都使用 WebRTC 點對點雙向媒體串流 (例如，視頻聊天應用程式)。當 A 想打電話給 B 時會發生什麼情況？

為了連線至 B 的應用程式，A 的應用程式必須產生 SDP 提議。SDP 提議包含 A 的應用程式想建立的工作階段的相關資訊，包括使用什麼轉碼器、這是音訊還是視訊工作階段等。還包含 ICE 候選項清單，這是可供 B 的應用程式嘗試用來連線至 A 的 IP 和連接埠配對。

為了建置 ICE 候選項清單，A 的應用程式會向 STUN 伺服器提出一連串請求。伺服器傳回發出請求的公有 IP 地址和連接埠配對。A 的應用程式將每一對新增至 ICE 候選項清單，換句話說，收集 ICE 候選項。A 的應用程式一旦完成收集 ICE 候選項，就可以傳回 SDP。

接下來，A 的應用程式必須透過這些應用程式通訊的訊號頻道，將 SDP 傳送到 B 的應用程式。WebRTC 標準並未規定此交換的傳輸通訊協定。它可以通過 HTTPS 進行，安全 WebSocket或任何其他通訊協定。

現在，B 的應用程式必須產生 SDP 回答。B 的應用程式遵循 A 在上一步使用的相同步驟：收集 ICE 候選項等。然後，B 的應用程式需要將此 SDP 回答傳回至 A 的應用程式。

A 和 B 交換 SDP 之後，他們會執行一系列的連接檢查。在每個應用程式中，ICE 演算法採用在對方 SDP 中收到的清單中的候選 IP/連接埠配對，並將 STUN 請求傳送至此配對。如果另一個應用程式有所回應，則起源應用程序會認為檢查成功，並將該 IP/連接埠配對標示為有效的 ICE 候選項。

在所有 IP/連接埠配對上完成連線檢查後，應用程式會交涉並決定使用其中一個剩餘的有效配對。選取配對後，媒體就會開始在應用程式之間流動。

如果任一應用程式找不到通過連線檢查的 IP/連接埠配對，則會向 TURN 伺服器提出 STUN 請求，以取得媒體轉送地址。轉送位址是公有 IP 地址和連接埠，可轉送往返於應用程式收到的封包，以設定轉送地址。然後，此轉送地址會新增至候選項清單，並透過訊號頻道交換。

Kinesis Video Streams (WebRTC)

Kinesis Video Streams Video Streams WebRTC Streams 包含以下元件：

- 控制平台

  控制平面元件負責建立和維護具有 WebRTC 技術訊號通道的 Kinesis 視訊串流。如需詳細資訊，請參閱 Amazon Kinesis Video Streams API 參考。
- 信令

  信令組件管理 WebRTC 信令端點，允許應用程序相互安全地連接 peer-to-peer即時媒體串流。訊號元件包含 Amazon Kinesis Video Signaling REST API 和一組 Websocket API。
- STUN

此元件管理 STUN 端點，當應用程式位於 NAT 或防火牆後方時，這些端點可讓應用程式探索其公有 IP 地址。

- 轉

  該組件管理 TURN 端點，當應用程序無法流媒體時，通過雲啟用媒體中繼 peer-to-peer。

- Kinesis Video Streams WebRTC 開發套件

  這些軟件庫可以下載，安裝和配置在設備和應用程序客戶端上，以使具有 WebRTC 功能的攝像機 IoT 設備能夠實現低延遲 peer-to-peer 媒體串流。這些開發套件還能讓 Android、iOS 和網路應用程式用戶端將 Kinesis Video Streams 與 WebRTC 技術的訊號、轉向和 STUN 功能整合至任何符合 WebRTC 標準的行動或網路播放器。

  - 適用於嵌入式裝置的 C WebRTC 開發套件
  - 適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript
  - 適用於 Android 的 WebRTC 開發套件
  - 適用於 iOS 的 WebRTC 開發套件

WebRTC Websocket API

主題

- ConnectAsViewer
- ConnectAsMaster
- SendSdpOffer
- SendSdpAnswer
- SendIceCandidate
- Disconnect
- 非同步訊息接收

ConnectAsViewer

以檢視器身分連線至端點所指定的訊號頻道。任何 WebSocket符合標準的程式庫都可用來連線至從 `GetSignalingEndpoint` API 呼叫取得的端點。必須提供訊號頻道的 Amazon Resource Name (ARN) 和用戶端 ID 作為查詢字串參數。有個別端點可供以主節點和檢視器身分來連線。如果存在與請求中指定的`ClientId`相同的現有連接，則新連接優先。新的資訊會覆寫連線中繼資料。

請求

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

- X-Amz-ChannelARN - 訊號頻道的 ARN。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 1024。
  - 模式：`arn:aws:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`
  - 必要：是
- X-AMZ-ClientId-客戶端的唯一標識符。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`^(?!(?i)AWS_.*)[a-zA-Z0-9_.-]`

    > **Note**
    >
    > X-Amz-ClientId不能開始AWS_。

  - 必要：是

回應

200 OK HTTP 狀態碼和空白內文。

錯誤

- InvalidArgumentException

  指定的參數超過其限制、不支援或無法使用。如需詳細資訊，請參閱傳回的訊息。

  HTTP 狀態碼：400
- AccessDeniedException

  發起人未獲授權存取指定的通道或符記已過期。

HTTP 狀態碼：403

- ResourceNotFoundException

  頻道不存在。

  HTTP 狀態碼：404

- ClientLimitExceededException

  以太高的速率叫用 API 時，或連線至頻道的檢視器數目超出支援的上限時。如需詳細資訊，請參閱使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams中的〈錯誤重試和指數輪詢〉和〈〉。 AWS

  HTTP 狀態碼：400

限制/節流

如果以太高的速率叫用 API，或連線至頻道的檢視器數目超出支援的上限時，則會在帳戶層級節流此 API。節流時會以 `ClientLimitExceededException` 傳回錯誤。

等冪

如果指定`ClientId`和頻道的連線已經存在，則連線中繼資料會以新資訊更新。

重試行為

這視為新的 API 呼叫。

並行呼叫

允許並行呼叫，每次呼叫都會更新連線中繼資料。

ConnectAsMaster

以主節點身分連線至端點指定的訊號頻道。任何 WebSocket投訴庫都可用於連接到從 `GetSignalingChannelEndpoint` API 調用獲取的端點。必須提供訊號頻道的 Amazon Resource Name (ARN) 作為查詢字串參數。有個別端點可供以主節點和檢視器身分來連線。如果有一個以上的用戶端做為主要用戶端連線到特定通道，則最新的要求優先順序。新的連線中繼資料會覆寫現有的連線中繼資料。

請求

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

- X-Amz-ChannelARN - 訊號頻道的 ARN。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 1024。
  - 模式：`arn:aws:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`
  - 必要：是

回應

200 OK HTTP 狀態碼和空白內文。

錯誤

- InvalidArgumentException

  指定的參數超過其限制、不支援或無法使用。如需詳細資訊，請參閱傳回的訊息。

  HTTP 狀態碼：400
- AccessDeniedException

  發起人未獲授權存取指定的通道或符記已過期。

  HTTP 狀態碼：403
- ResourceNotFoundException

  頻道不存在。

  HTTP 狀態碼：404
- ClientLimitExceededException

  以太高的速率叫用 API 時。如需詳細資訊，請參閱使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams中的〈錯誤重試和指數輪詢〉和〈〉。AWS

  HTTP 狀態碼：400

限制/節流

如果以太高的速率叫用 API，則會在帳戶層級節流此 API。節流時會以 `ClientLimitExceededException` 傳回錯誤。

等冪

如果指定的 clientId 和頻道已存在連線，則會以新資訊更新連線中繼資料。

重試行為

這視為新的 API 呼叫。

並行呼叫

允許並行呼叫，每次呼叫都會更新連線中繼資料。

SendSdpOffer

將提議傳送給目標收件者。先決條件是客戶端必須已經連接到從 `GetSignalingChannelEndpoint` API 獲取的 WebSocket 端點。

如果傳送者類型是檢視器，則將提議傳送給主節點。此外，不需要指定 `RecipientClientId`，系統會忽略 `RecipientClientId` 的任何指定值。如果傳送者類型是主節點，則提議會傳送給 `RecipientClientId` 指定的目標檢視器。在這種情況下，`RecipientClientId` 是必要輸入。

允許主節點用戶端應用程式將提議傳送給任何檢視器，但只允許檢視器用戶端應用程式將提議傳送給主節點用戶端應用程式。如果檢視器用戶端應用程式嘗試將提議傳送給另一個檢視器用戶端應用程式，則「不會」執行該請求。如果同一個用戶端有尚未傳遞的未完成提議，則會以新的提議覆寫此提議。

請求

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

- action - 傳送的訊息類型。

  - 類型：ENUM
  - 有效值：SDP_OFFER、SDP_ANSWER、ICE_CANDIDATE
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：是
- recipientClientId-收件者的唯一識別碼。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：是
- messagePayload - base-64 編碼的訊息內容。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 10K。
  - 必要：是
- correlationId - 訊息的唯一識別符。這是選擇性的參數。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：否

回應

如果訊號後端成功接收訊息，則不會傳回任何回應。如果服務遇到錯誤，且請求中指定 `correlationId`，則會以 `STATUS_RESPONSE` 訊息形式傳回錯誤詳細資訊。如需詳細資訊，請參閱非同步訊息接收。

錯誤

- InvalidArgumentException

  指定的參數超過其限制、不支援或無法使用。如需詳細資訊，請參閱傳回的訊息。

  HTTP 狀態碼：400
- ClientLimitExceededException

以太高的速率叫用 API 時。如需詳細資訊，請參閱使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams中的〈錯誤重試和指數輪詢〉和〈〉。AWS

HTTP 狀態碼：400

限制/節流

如果以太高的速率叫用 API，則會在帳戶層級節流此 API。節流時會以 `ClientLimitExceededException` 傳回錯誤。

等冪

此 API 不是等冪。

重試行為

這視為新的 API 呼叫。

並行呼叫

允許並行呼叫。每次呼叫會傳送一次提議。

SendSdpAnswer

將回答傳送給目標收件者。先決條件是客戶端必須已經連接到從 `GetSignalingChannelEndpoint` API 獲取的 WebSocket 端點。

如果傳送者類型是檢視器，則將回答傳送給主節點。此外，不需要指定 `RecipientClientId`，系統會忽略 `RecipientClientId` 的任何指定值。如果傳送者類型是主節點，則回答會傳送給 `RecipientClientId` 指定的目標檢視器。在這種情況下，`RecipientClientId` 是必要輸入。

允許主節點用戶端應用程式將回答傳送給任何檢視器，但只允許檢視器用戶端應用程式將回答傳送給主節點用戶端應用程式。如果檢視器用戶端應用程式嘗試將回答傳送給另一個檢視器用戶端應用程式，則「不會」執行該請求。如果同一個用戶端有尚未傳遞的未完成回答，則會以新的回答覆寫此回答。

請求

```
{
    "action": "SDP_ANSWER",
    "recipientClientId": "string",
```

```
    "messagePayload": "string",
    "correlationId": "string"
}
```

- action - 傳送的訊息類型。
  - 類型：ENUM
  - 有效值：SDP_OFFER、SDP_ANSWER、ICE_CANDIDATE
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：是
- recipientClientId-收件者的唯一識別碼。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：是
- messagePayload - base-64 編碼的訊息內容。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 10K。
  - 必要：是
- correlationId - 訊息的唯一識別符。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：否

回應

如果訊號後端成功接收訊息，則不會傳回任何回應。如果服務遇到錯誤，且請求中指定 correlationId，則會以 STATUS_RESPONSE 訊息形式傳回錯誤詳細資訊。如需詳細資訊，請參閱非同步訊息接收。

錯誤

- InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。如需詳細資訊，請參閱傳回的訊息。

HTTP 狀態碼：400

- ClientLimitExceededException

以太高的速率叫用 API 時傳回。如需詳細資訊，請參閱使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams中的〈錯誤重試和指數輪詢〉和〈〉。AWS

HTTP 狀態碼：400

限制/節流

如果以太高的速率叫用 API，則會在帳戶層級節流此 API。節流時會以 `ClientLimitExceededException` 傳回錯誤。

等冪

此 API 不是等冪。

重試行為

這視為新的 API 呼叫。

並行呼叫

允許並行呼叫。每次呼叫會傳送一次提議。

SendIceCandidate

將 ICE 候選項傳送給目標收件者。先決條件是客戶端必須已經連接到從 `GetSignalingChannelEndpoint` API 獲取的 WebSocket 端點。

如果傳送者類型是檢視器，則會將 ICE 候選項傳送給主節點。此外，不需要指定 `RecipientClientId`，系統會忽略 `RecipientClientId` 的任何指定值。如果傳送者類型為主要，則 ICE 候選項會傳送至由指定的目標`RecipientClientId`。 `RecipientClientId`在這種情況下是必需的輸入。

允許主節點用戶端應用程式將 ICE 候選項傳送給任何檢視器，但只允許檢視器用戶端應用程式將 ICE 候選項傳送給主節點用戶端應用程式。如果檢視器用戶端應用程式嘗試將 ICE 候選項傳送給另一個檢視器用戶端應用程式，則「不會」執行該請求。

請求

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

- action - 傳送的訊息類型。
  - 類型：ENUM
  - 有效值：SDP_OFFER、SDP_ANSWER、ICE_CANDIDATE
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：是
- recipientClientId-收件者的唯一識別碼。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：否
- messagePayload - base64 編碼的訊息內容。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 10K。
  - 必要：是
- correlationId - 訊息的唯一識別符。
  - 類型：字串
  - 長度限制：最小長度為 1。長度上限為 256。
  - 模式：[a-zA-Z0-9_.-]+
  - 必要：否

回應

如果訊號後端成功接收訊息，則不會傳回任何回應。如果服務遇到錯誤，且請求中指定 `correlationId`，則會以 `STATUS_RESPONSE` 訊息形式傳回錯誤詳細資訊。如需詳細資訊，請參閱非同步訊息接收。

錯誤

- InvalidArgumentException

  指定的參數超過其限制、不支援或無法使用。如需詳細資訊，請參閱傳回的訊息。

  HTTP 狀態碼：400
- ClientLimitExceededException

  以太高的速率叫用 API 時。如需詳細資訊，請參閱使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams中的〈錯誤重試和指數輪詢〉和〈〉。AWS

  HTTP 狀態碼：400

限制/節流

如果以太高的速率叫用 API，則會在帳戶層級節流此 API。節流時會以 `ClientLimitExceededException` 傳回錯誤。

等冪

此 API 不是等冪。

重試行為

這視為新的 API 呼叫。

並行呼叫

允許並行呼叫。每次呼叫會傳送一次提議。

Disconnect

用戶端可以隨時關閉連線。WebSocket與程式庫支援關閉功能。當連線關閉時，服務會將用戶端標示為在特定訊號頻道中離線，而且不會嘗試傳遞任何訊息。在閒置連線的情況下也採取相同的行為超時。

服務也會傳送中斷連線指示給用户端，例如在部署或伺服器維護期間。以下是定義的指示訊息：

- 走：此訊息用來初始化連線關閉。可讓用戶端正常處理先前的消息、中斷連線，並在必要時重新連線至訊號頻道。
- 重新連接服務器：此訊息用於初始化轉送連線關閉，還可讓用户端正常中斷連線、取得新的 ICE 伺服器組態，並在必要時重新連線至轉送伺服器。

非同步訊息接收

所有回應訊息都視為事件非同步傳遞至收件者 (例如，SDP 提議或 SDP 回答傳遞)。以下是事件訊息結構。

事件

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

- senderClientId-寄件者用戶端的唯一識別碼。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`[a-zA-Z0-9_.-]+`
  - 必要：否
- messageType - 事件的類型。
  - 類型：ENUM
  - 有效類型：SDP_OFFERSDP_ANSWER、ICE_CANDIDATE、、GO_AWAY、RECONNECT_ICE_SERVER、STATUS
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`[a-zA-Z0-9_.-]+`

  - 必要：是
- messagePayload - base64 編碼的訊息內容。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 10K。
  - 必要：否
- correlationId - 狀態所指的訊息的唯一識別符。這是用戶端訊息 (例如，SDP 提議、SDP 回答或 ICE 候選項) 中提供的相同 correlationId。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`[a-zA-Z0-9_.-]+`
  - 必要：是
- errorType - 錯誤的唯一識別名稱。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`[a-zA-Z0-9_.-]+`
  - 必要：否
- statusCode - 對應於回應性質的 HTTP 狀態碼。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 256。
  - 模式：`[a-zA-Z0-9_.-]+`
  - 必要：否
- description - 解釋狀態的字串描述。
  - 類型：字串
  - 長度限制：長度下限為 1。長度上限為 1K。
  - 必要：否

使用 WebRTC 技術 Service Quotas 的 Amazon Kinesis Video Streams

使用 WebRTC 技術的 Kinesis Video Streams 具有下列服務配額：

> **Important**
>
> 下列服務配額為 soft [s]，可透過提交支援票證來升級，或是 hard [h] (無法增加)。你會在下表中看到個別服務配額旁邊的 [s] 及 [h]。

> **Note**
>
> TPS 代表每秒交易。

主題

控制平面 API 服務配額

下列各節描述控制平面 API 的服務配額。

API	帳戶服務配額：請求	帳戶服務配額：通道	通道層級服務配額	相關例外和備註
CreateSignalingChannel	50 TPS [s]	us-east-1 和 us-west-2 -每個區域每個帳戶 10,000 個		

API	帳戶服務配額：請求	帳戶服務配額：通道	通道層級服務配額	相關例外和備註
		頻道；所有其他支援的區域-每個區域每個帳戶 5,000 個頻道		
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		
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]	

API	帳戶服務配額：請求	帳戶服務配額：通道	通道層級服務配額	相關例外和備註
UpdateMediaStorageConfiguration	10 TPS [h]		5 TPS [h]	
UpdateSignalingChannel	50 TPS [h]	N/A	5 TPS [h]	

訊號 API 服務配額

下節說明使用 WebRTC 技術的 Kinesis 視訊串流中訊號元件的服務配額。如需詳細資訊，請參閱 Kinesis Video Streams Video StreWebRTC: 運作方式。

- ConnectAsMaster
  - API-每個通道 3 TPS（小時）
  - 每個信令通道的主連接數上限-1 (h)
  - 連線持續時間限制-1 小時 (h)
  - 閒置連線逾時-10 分鐘 (h)
  - 當用戶端收到來自伺服器的GO_AWAY訊息時，連線會在 1 分鐘 (h) 的寬限期後終止
- ConnectAsViewer
  - API-每個通道 3 TPS（小時）
  - 每個通道的檢視器連線數上限-10 (s)
  - 連線持續時間限制-1 小時 (h)
  - 閒置連線逾時-10 分鐘 (h)
  - 一旦用戶端收到來自伺服器的GO_AWAY訊息，連線就會在 1 分鐘 (h) 的寬限期後終止
- 中斷連線
  - N/A
- GetIceServerConfig
  - API-每個信令通道 5 TPS（小時）
- SendAlexaOffertoMaster

  - API-每個信令通道 5 TPS（小時）
- SendICECandidate
  - API-每個 WebSocket 連線 20 TPS (小時)
  - 訊息承載大小限制-10k (h)
- SendSDPAnswer
  - API-每個 WebSocket 連接 5 個 TPS（小時）
  - 訊息承載大小限制-10k (h)
- SendSDPOffer
  - API-每個 WebSocket 連接 5 個 TPS（小時）
  - 訊息承載大小限制-10k (h)

TURN 服務配額

下節說明使用 WebRTC 技術的 Kinesis Video Streams 中使用轉送周圍的 NAT (TURN) 元件周遊的服務配額。如需詳細資訊，請參閱 Kinesis Video Streams Video StreWebRTC: 運作方式。

- 位元速率-5 兆比特 (小時)
- 認證生命週期-5 分鐘 (h)
- 分配數量-每個信令通道 50（h）

WebRTC 擷取 Service Quotas

下節說明 Amazon Kinesis Video Streams WebRTC 中媒體錄製元件的服務配額。如需詳細資訊，請參閱 Amazon Kinesis Video Streams WebRTC 技術攝入。

JoinStorageSession

- API：
  - 每個帳戶-50 TPS（小時）
  - 每個通道-2 TPS（小時）
- 串流工作階段配額：
  - 位元速率-1 兆比特
  - 會話持續時間-1 小時（h）

- **閒置逾時-3 分鐘** (h)

開始使用

本節說明如何使用 WebRTC 技術在 Amazon Kinesis Video Streams 中執行下列任務：

- 設置 AWS 帳戶 並創建管理員。
- 使用 WebRTC 技術信號通道建立室壁運動視訊串流。
- 使用 Kinesis Video Streams 搭配 WebRTC SDK 來設定主要和檢視器，以透過訊號通道執行 peer-to-peer視訊和音訊串流。

如果您是使用 WebRTC 技術的 Kinesis Video Streams 的新手，我們建議您先閱讀。Kinesis Video Streams Video StreWebRTC: 運作方式

主題

設置 AWS 帳戶並創建管理員

第一次搭配 WebRTC 使用 Kinesis Video Streams 之前，請先完成下列工作：

主題

註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，會建立AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 root 使用者來執行需要 root 使用者存取權的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理權限的使用者

註冊後，請保護 AWS 帳戶 AWS 帳戶根使用者、啟用和建立系統管理使用者 AWS IAM Identity Center，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。AWS Management Console在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。
2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需指示，請參閱《IAM 使用者指南》中的為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 (主控台)。

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱 AWS IAM Identity Center 使用者指南中的啟用 AWS IAM Identity Center。
2. 在 IAM 身分中心中，將管理存取權授予使用者。

   若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。

以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者登入的說明，請參閱使用AWS 登入 者指南中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「建立權限集」。
2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

   如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「新增群組」。

建立 AWS帳號金鑰

您需要一個 AWS 帳戶金鑰，才能透過 WebRTC 以程式設計方式存取 Kinesis Video Streams。

若要建立 AWS 帳戶金鑰，請執行下列動作：

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 https://console.aws.amazon.com/iam/。
2. 在導覽列中選擇 Users (使用者)，然後選擇 Administrator (管理員) 使用者。
3. 開啟 Security credentials (安全性登入資料) 標籤，然後選擇 Create access key (建立存取金鑰)。
4. 記錄 Access key ID (存取金鑰 ID)。選擇 [秘密存取金鑰] 下的 [顯示]，然後記錄秘密存取金鑰。

建立訊號頻道

您可以使用 Kinesis Video Streams 主控台、 AWS API (CreateSignalingChannel) 或建 AWS CLI 立訊號通道。

使用主控台建立訊號頻道

1. 登入 AWS Management Console 並開啟 Amazon Kinesis Video Streams 主控台。
2. 在 Signaling channels (訊號頻道) 頁面上，選擇 Create signaling channel (建立訊號頻道)。
3. 在 Create a new signaling channel (建立新的訊號頻道) 頁面上，輸入訊號頻道的名稱。保持 60 秒的預設 T ime-to-live (TTL) 值不變。
4. 選擇 Create signaling channel (建立訊號頻道)。

5. 建立訊號頻道之後，在頻道詳細資訊頁面上檢閱詳細資訊。

串流即時媒體

Kinesis Video Streams with WebRTC 包含以下開發套件：

- 適用於嵌入式裝置的 C WebRTC 開發套件
- 適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript
- 適用於 Android 的 WebRTC 開發套件
- 適用於 iOS 的 WebRTC 開發套件

每個 SDK 都包含對應的範例和 step-by-step 指示，可協助您建置和執行這些應用程式。您可以使用這些範例進行低延遲、即時、雙向的音訊和視訊串流，以及在 Web/Android/iOS 應用程式或嵌入式裝置的任何組合之間進行資料交換。換句話說，您可以將嵌入式攝影機裝置的即時音訊和視訊串流至 Android 或 Web 應用程式，或在兩個 Android 應用程式之間串流。

適用於嵌入式裝置的 C WebRTC 開發套件

下列 step-by-step 指示說明如何針對嵌入式裝置及其對應範例使用 WebRTC 技術 SDK 下載、建置和執行 Kinesis Video Streams。

在 C 語言中使用 WebRTC 技術開發套件下載 Kinesis Video Streams

若要針對嵌入式裝置下載含 WebRTC 技術 SDK 的 Kinesis Video Streams，請執行下列命令：

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

使用 C 語言中的 WebRTC 技術開發套件建置 Kinesis Video Streams

> **Important**
>
> 在 macOS 上完成這些步驟之前，您必須運行以使用命令行工具和標題`xcode-select --install`來下載軟件包，具體取決於您所擁有的 macOS 版本。然後打開`/Library/Developer/CommandLineTools/Packages/macOS_SDK_headers_for_macOS_10.14.pkg`並按照安裝程序安裝命令行工具和標題。您只需要在調用`cmake`之前執行此操作一次。如果您已經安裝了命令行工具和標頭，則無需再次執行此命令。

請完成下列步驟：

1. 安裝 CMake：
   - 在 macOS 上，執行 `brew install cmake pkg-config srtp`
   - 在 Ubuntu 上，執行 `sudo apt-get install pkg-config cmake libcap2 libcap-dev`
2. 取得您要用於此示範 AWS 帳戶 的存取金鑰和秘密金鑰。
3. 執行以下命令，在您下載的 WebRTC C 開發套件中建立 `build` 目錄，然後從中執行 `cmake`：

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

4. 現在，你在你剛剛創建與上述步驟的`build`目錄中，運行`make`以構建 WebRTC C SDK 及其提供的示例。

> **Note**
>
> 如果系統沒有`gstreamer`安裝，`kvsWebrtcClientMasterGstSample`將不會建立。要確保它是內置的（在 macOS 上），您必須運行：`brew install gstreamer gst-plugins-base gst-plugins-good`

執行 WebRTC C 開發套件的範例

完成上述程序之後，最後在 `build` 目錄中會出現下列範例應用程式：

- `kvsWebrtcClientMaster`-此應用程序通過信令通道發送樣本 H264/Opus 幀（路徑：/樣本/h264 和/採樣/SampleFrames ）。opusSampleFrames也會接受傳入的音訊 (如果在瀏覽器中啟用)。在瀏覽器中檢查時，將會在終端機列印收到的音訊封包的中繼資料。
- `kvsWebrtcClientViewer` - 此應用程式接受並印列範例 H264/Opus 影格。
- `kvsWebrtcClientMasterGstSample` - 此應用程式從 GStreamer 管道傳送範例 H264/Opus 影格。

若要執行這些範例中的任何一個，請完成以下步驟：

1. 使用您的 AWS 帳戶 憑據設置您的環境：

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

如果您使用臨時 AWS 憑據，請同時導出會話令牌：

```
export AWS_SESSION_TOKEN=YourSessionToken
```

如果您要設定自訂 CA 憑證路徑，您可以使用下列方式進行設定：

```
export AWS_KVS_CACERT_PATH=../certs/cert.pem
```

> **Note**
>
> 根據預設，SSL CA 憑證會設定為.. /證書/cert.pem，它指向中此存儲庫中的文件。GitHub

2. 將您要給訊號頻道的名稱傳給其中一個範例應用程式，以執行應用程式。應用程式會使用您提供的名稱建立訊號頻道。例如，若要建立名為 myChannel 的訊號頻道，並開始透過此頻道傳送範例 H264/Opus 影格，請執行下列命令：

```
./kvsWebrtcClientMaster myChannel
```

當命令列應用程式列印 Connection established 時，您可以繼續進行下一個步驟。

3. 現在，訊號頻道已建立，且連線的主節點正在將媒體串流至訊號頻道，您可以檢視此串流。例如，您可以在 Web 應用程式中檢視此即時串流。若要這麼做，請使用中的步驟開啟 WebRTC SDK 測試頁面，使用 WebRTC 測試頁面的 Kinesis Video Streams並使用您為上述主機指定的相同 AWS 認證和相同的信號通道來設定下列值：

   - 存取金鑰 ID
   - 私密存取金鑰
   - 訊號頻道名稱
   - 用戶端 ID (選擇性)

   選擇 Start viewer (啟動檢視器)，開始範例 H264/Opus 影格的即時視訊串流。

視頻教程

本影片示範如何連接您的攝影機，並開始使用適用於 WebRTC 的 Amazon Kinesis Video Streams。

適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript

您可以在中找到 JavaScript 適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術 SDK 及其對應範例。GitHub

主題

- 使用 WebRTC 技術開發套件安裝 Kinesis Video Streams JavaScript
- Kinesis Video Streams 與 WebRTC JavaScript 技術開發套件說明文件
- 使用 WebRTC 測試頁面的 Kinesis Video Streams
- 編輯使用 WebRTC 測試頁面的 Kinesis Video Streams

使用 WebRTC 技術開發套件安裝 Kinesis Video Streams JavaScript

您是否以及如何使用 WebRTC SDK 安裝 Kinesis Video Streams， JavaScript 取決於程式碼是在`Node.js`模組還是瀏覽器指令碼中執行。

NodeJS module

若要在 Node.js 中使用 WebRTC 技術開發套件安裝 Kinesis Video Streams 的慣 JavaScript 用方式是使用 npm，Node.js 套件管理員。

該軟件包託管在 https://www.npmjs.com/package/ 網amazon-kinesis-video-streams站。

要在`Node.js`項目中安裝此 SDK，請使用終端導航到與項目相同的目錄`package.json`：

輸入下列內容：

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

您可以像典型的 Node.js 模塊一樣導入 SDK 類：

```
// JavaScript
```

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

Browser

您無需另行安裝，也能在瀏覽器指令碼中使用軟體開發套件。您可以 AWS 使用 HTML 頁面中的指令碼直接載入託管 SDK 套件。

若要在瀏覽器中使用 SDK，請將下列指令碼元素新增至您的 HTML 網頁：

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

當您將軟體開發套件載入至頁面後，便可從全域變數 KVSWebRTC (或 window.KVSWebRTC) 取得該軟體開發套件。

例如 window.KVSWebRTC.SignalingClient。

Kinesis Video Streams 與 WebRTC JavaScript 技術開發套件說明文件

SDK 方法的文檔位於 GitHub 自述文檔下。

在 [使用情況] 區段中，還有其他資訊可用於將此 SDK 與 AWS SDK 整合， JavaScript 以建置 Web 型檢視器應用程式。

請參閱目examples錄以取得完整應用程式的範例，包括主要角色和檢視者角色。

使用 WebRTC 測試頁面的 Kinesis Video Streams

使用 WebRTC 的 Kinesis Video Streams 也會託管一個測試頁面，您可以用來建立新的訊號通道或連線至現有頻道，並將其當作主頻道或檢視器使用。

網路 Kinesis Video Streams 測試頁面位於 https://awslabs.github.io/ amazon-kinesis-video-streams-/examples/index.html。webrtc-sdk-js

測試頁面的程式碼位於目examples錄中。

主題

- 從測試頁面流式傳輸到 AWS Management Console

- 從測試頁面流式傳輸到測試頁面

從測試頁面流式傳輸到 AWS Management Console

1. 開啟「使用 WebRTC 技術測試的 Kinesis Video Streams」頁面並完成以下操作：
   - AWS 區域。 例如，us-west-2.
   - IAM 使用者或角色的 AWS 存取金鑰和秘密金鑰。如果您使用長期 AWS 認證，請將工作階段權杖保留空白。
   - 您要連線的訊號頻道的名稱。

     如果要連接到新的信令通道，請選擇「建立通道」以使用方塊中提供的值建立信號通道。

     > **Note**
     >
     > 對於目前帳戶和地區，您的信號頻道名稱必須是唯一的。您可以使用字母、數字、底線 (_) 和連字號 (-)，但不能使用空格。

   - 是否要傳送音訊、視訊或兩者。
   - ICE 候選人一代。離開STUN/TURN選擇並保持Trickle ICE啟用狀態。
2. 選擇「啟動主訊號」以連線至訊號通道。

   允許訪問您的攝像頭和/或麥克風，如果需要。
3. 在中開啟 Kinesis Video Streams 主控台。 AWS Management Console

   確定已選取正確的區域。
4. 在左側導覽列中，選取訊號頻道。

   選取上方信號通道的名稱。如有需要，請使用搜尋列。
5. 展開媒體播放檢視器區段。
6. 選擇視頻播放器上的播放按鈕。這加入 WebRTC 技術會話作為一個. viewer

   在示範頁面上傳送的媒體應該會顯示在中 AWS Management Console。

從測試頁面流式傳輸到測試頁面

1. 開啟包含 WebRTC 技術的 Kinesis Video Streams 訊串流測試頁面，並完成下列資訊：

- AWS 區域。 例如，us-west-2.
- IAM 使用者或角色的 AWS 存取金鑰和秘密金鑰。如果您使用長期 AWS 認證，請將工作階段權杖保留空白。
- 您要連線的訊號頻道的名稱。

  如果要連接到新的信令通道，請選擇「建立通道」以使用方塊中提供的值建立信號通道。

  > **Note**
  >
  > 對於目前帳戶和地區，您的信號頻道名稱必須是唯一的。您可以使用字母、數字、底線 (_) 和連字號 (-)，但不能使用空格。

- 是否要傳送音訊、視訊或兩者。
- ICE 候選人一代。離開STUN/TURN選擇並保持Trickle ICE啟用狀態。

2. 選擇啟動主機以連接到信令通道作為master角色。

   允許訪問您的攝像頭和/或麥克風，如果需要。

3. 開啟另一個瀏覽器索引標籤，然後開啟具有 WebRTC 技術的 Kinesis Video Streams 測試頁面。所有來自先前運行的信息應該加載。
4. 向下捲動並選擇 [啟動檢視器]，以連線至作為viewer角色的信號通道。

   您應該看到與之間交換的媒master體viewer。

編輯使用 WebRTC 測試頁面的 Kinesis Video Streams

要編輯 SDK 和測試頁面以進行開發，請按照以下說明進行操作。

必要條件

NodeJS 版本 16 +

> **Note**
>
> 我們建議您從 https://nodejs.org/en/download 下載最新的長期支持（LTS）版本。

編輯測試頁面

1. 使用 WebRTC 技術開發套件下載 Kinesis Video Streams。 JavaScript

   在終端中鍵入以下內容：

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

2. 瀏覽至包含包裝 .json 檔案的目錄。檔案位於儲存庫的根目錄中。

   在終端中鍵入以下內容：

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

3. 安裝依存項目。

   在終端機中鍵入以下 npm CLI 命令：

   ```
   npm install
   ```

4. 啟動 Web 服務器以開始提供網頁。

   在終端機中鍵入以下 npm CLI 命令：

   ```
   npm run develop
   ```

5. 在您的瀏覽器中，訪問 http://localhost:3001/。

   您可以編輯目錄中的檔案來編輯網`examples`頁。

適用於 Android 的 WebRTC 開發套件

下列 step-by-step 指示說明如何使用適用於 Android 的 WebRTC 技術 SDK 及其對應範例下載、建置和執行 Kinesis Video Streams。

> **Note**
>
> Kinesis Video Streams 不支援安卓系統上的 IPv6 位址。有關在您的安卓設備上禁用 IPv6 的更多信息和步驟，請參閱 https://support.surfshark.com/hc/en-us/articles/360011828279-H ow-to-disable- 安卓版本。

下載適用於 Android 的 WebRTC 開發套件

若要在 Android 中下載 WebRTC 開發套件，請執行下列命令：

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

在 Android 中建置 WebRTC 開發套件

若要在 Android 中建置 WebRTC 開發套件，請完成以下步驟：

1. 導入安卓 WebRTC 技術 SDK 到安卓工作室集成開發環境 (IDE) 通過打開作為項目打**amazon-kinesis-video-streams-webrtc-sdk-android/build.gradle**開.
2. 如果是第一次開啟專案，專案會自動同步。如果不是，請啟動同步。當您看到組建錯誤時，請選擇 Install missing SDK package(s) (安裝缺少的 SDK 套件)，然後選擇 Accept (接受) 並完成安裝，以安裝任何必要的 SDK 套件。
3. 進行 Amazon Cognito (使用者集區和身分集區) 設定。如需詳細步驟，請參閱為 Android WebRTC 開發套件設定 Amazon Cognito。這將產生建置 Android WebRTC 開發套件所需的身分驗證和授權設定。
4. 在您的 Android IDE 中，開啟 `awsconfiguration.json` (從 `src/main/res/raw/`)。檔案看起來如下：

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

以執行 為 Android WebRTC 開發套件設定 Amazon Cognito中的步驟所產生的值，更新 `awsconfiguration.json`。

5. 確保您的 Android 裝置已連線至執行 Android IDE 的電腦。在 Android IDE 中，選取連接的裝置，然後建置並執行 WebRTC Android 開發套件。

   此步驟會在您的 Android 裝置上安裝名為 `AWSKinesisVideoWebRTCDemoApp` 的應用程式。您可以使用此應用程式，驗證行動、Web 和 IoT 裝置用戶端之間的即時 WebRTC 音訊/視訊串流。

執行 Android 範例應用程式

請完成下列步驟：

1. 在您的 Android 裝置上，使用新的 (先建立) 或現有的 Amazon Cognito 帳戶開啟AWSKinesisVideoWebRTCDemoApp並登入。
2. 在中 AWSKinesisVideoWebRTCDemoApp，瀏覽至「通道組態」頁面，然後建立新的信號通道或選擇現有的信號通道。

   > **Note**
   >
   > 目前，使用此 SDK 中的範例應用程式，您只能在中執行一個信號通道。AWSKinesisVideoWebRTCDemoApp

3. 選擇性：如果您要以檢視器身分連線到此頻道，請選擇唯一的 Client Id (用戶端 ID)。只有在多個檢視器連線至頻道時，才需要用戶端 ID。這有助於頻道的主節點識別各自的檢視器。
4. 選擇以AWS 區域及是否要傳送音訊或視訊資料，或兩者都傳送。
5. 若要驗證 peer-to-peer 串流，請執行下列任一項作業：

   > **Note**
   >
   > 請務必在此示範中使用的所有用戶端上，指定相同的訊號通道名稱、AWS地區、檢視者 ID 和AWS帳戶 ID。

- 在兩個 Android 設備之間進行eer-to-peer 流傳輸：主設備和查看器
  - 在兩個 Android 裝置上，使用上述程序下載、建置和執行 Android WebRTC 開發套件。
  - 在主模式下的一台 Android 設備AWSKinesisVideoWebRTCDemoApp上打開（選擇 START MASTER）以啟動新的會話（信令通道）。

    > **Note**
    >
    > 目前，任何給定的訊號頻道只能有一個主節點。

  - 以查看器模式在第二個 Android 設備上打開 AWSKinesisVideoWebRTCDemoApp，以連接到上述步驟中啟動的信令通道（會話）（選擇 START VIEWER）。

    確認檢視器可以看到主節點的音訊/視訊資料。
- 在嵌入式 SDK 主機和安卓設備查看器之間進行eer-to-peer 流式傳輸
  - 在攝影機裝置上，以主節點模式下載、建置和執行 適用於嵌入式裝置的 C WebRTC 開發套件。
  - 在 Android 裝置上，使用上述程序下載、建置和執行 Android WebRTC 開發套件。在此 Android 設備AWSKinesisVideoWebRTCDemoApp上以查看器模式打開，並驗證查看器是否可以看到嵌入式 SDK 主機的音頻/視頻數據。
- 在 Android 設備之間進行eer-to-peer 流式傳輸，作為主瀏覽器和網絡瀏覽器
  - 在 Android 裝置上，使用上述程序下載、建置和執行 Android WebRTC 開發套件。在此 Android 設備AWSKinesisVideoWebRTCDemoApp上以主模式打開（選擇 START MASTER）以啟動新的會話（信令通道）。
  - 以檢視器身分下載、建置和執行 適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript ，並確認檢視器可以看到 Android 主節點的音訊/視訊。

為 Android WebRTC 開發套件設定 Amazon Cognito

必要條件

- 建議使用 Android Studio 檢查、編輯和執行應用程式的程式碼。我們建議使用最新的穩定版本。
- 在範例程式碼中，您提供 Amazon Cognito 登入資料。

請依照下列程序設定 Amazon Cognito 使用者集區和身分集區。

設定使用者集區

設定使用者集區

1. 登入 Amazon Cognito 主控台並確認該區域是否正確。
2. 在左側導覽中選擇 [使用者集區]。
3. 在 [使用者集區] 區段中選擇 [建立使用者集區]。
4. 完成以下各節：

   a. 步驟 1：設定登入體驗-在 Cognito 使用者集區登入選項區段中，選取適當的選項。

      選取下一步。

   b. 步驟 2：設定安全性需求-選取適當的選項。

      選取下一步。

   c. 步驟 3：配置註冊體驗-選擇適當的選項。

      選取下一步。

   d. 步驟 4：設定郵件傳遞-選取適當的選項。

      在 IAM 角色選取欄位中，選取現有角色或建立新角色。

      選取下一步。

   e. 步驟 5：集成您的應用程序-選擇適當的選項。

      在初始應用程式用戶端欄位中，選擇機密用戶端。

      選取下一步。

   f. 步驟 6：檢閱並建立-檢閱先前章節中的選取項目，然後選擇 [建立使用者集區]。

5. 在 [使用者集區] 頁面上，選取您剛建立的集區。

   複製使用者集區 ID 並記下此項，以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.PoolId`。

6. 選取 [應用程式整合] 索引標籤，然後前往頁面底部。
7. 在 [應用程式用戶端清單] 區段中，選擇您剛建立的應用程式用戶端名稱。

   複製用戶端 ID 並記下以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.AppClientId`。

8. 顯示用戶端密碼，並記下以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.AppClientSecret`。

設定身分集區

設定身分集區

1. 登入 Amazon Cognito 主控台並確認該區域是否正確。
2. 在左側導覽中，選擇 [識別集區]。
3. 選擇 建立身分池。
4. 設定身分識別集區。
   a. 步驟 1：設定識別集區信任-完成下列各節：
      - 使用者存取-選取已驗證存取
      - 已驗證身分來源-選取 Amazon Cognito 使用者集區

      選取下一步。
   b. 步驟 2：設定權限-在 [已驗證的角色] 區段中，完成下列欄位：
      - IAM 角色-選取 [建立新的 IAM 角色]
      - IAM 角色名稱-輸入名稱並記下它以供稍後步驟使用。

      選取下一步。
   c. 步驟 3：Connect 身分識別提供者-在 [使用者集區詳細資料] 區段中，完成下列欄位：
      - 使用者集區 ID-選取您之前建立的使用者集區。
      - 應用程式用戶端 ID-選取您之前建立的應用程式用戶端 ID。

      選取下一步。
   d. 步驟 4：配置屬性-在 [身分識別集區名稱] 欄位中輸入名稱。

      選取下一步。
   e. 步驟 5：檢閱並建立-檢閱您在每個區段中的選擇，然後選取 [建立身分集區]。
5. 在 [身分識別集區] 頁面上，選取新的身分識別集區。

複製身分集區 ID 並記下此項目以供稍後使用。在檔`awsconfiguration.json`案中，這是`CredentialsProvider.CognitoIdentity.Default.PoolId`。

6. 更新 IAM 角色的許可。

   a. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。

   b. 在左側導覽列中，選擇 [角色]。

   c. 尋找並選取您在上面建立的角色。

   > **Note**
   >
   > 如有需要，請使用搜尋列。

   d. 選取附加的權限原則。

      選擇 Edit (編輯)。

   e. 選取 JSON 索引標籤，並以下列項目取代原則：

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

      選取下一步。

   f. 如果尚未選取，請選取 [將此新版本設定為預設版本] 旁邊的核取方塊。

      選取儲存變更。

適用於 iOS 的 WebRTC 開發套件

下列 step-by-step 指示說明如何在 iOS 中下載、建置和執行 Kinesis Video Streams WebRTC 技術 SDK 及其對應範例。

在 iOS 中下載 WebRTC 開發套件

若要在 iOS 中下載 WebRTC 開發套件，請執行下列命令：

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

在 iOS 中建置 WebRTC 開發套件

請完成下列步驟：

1. 導入 iOS 的 WebRTC 技術 SDK 到 XCode 集成開發環境（IDE）在 iOS 計算機上打開`KinesisVideoWebRTCDemoApp.xcworkspace`（路徑： amazon-kinesis-video-streams-webrtc-sdk-ios /Swift/ AWSKinesisVideoWebRTCDemoApp .xc 工作區）。
2. 如果是第一次開啟專案，專案會自動建置。如果不是，請啟動建置。

   您可能會看到下列錯誤：

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

   如果您看到此錯誤，請執行下列動作：

   a. 從目前的工作目錄切換至 `amazon-kinesis-video-streams-webrtc-sdk-ios/Swift`，並在命令列中執行以下命令：

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

   b. 從目前的工作目錄切換至 `amazon-kinesis-video-streams-webrtc-sdk-ios`，並在命令列中執行以下命令：

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

   c. Build。

3. 進行 Amazon Cognito (使用者集區和身分集區) 設定。如需詳細步驟，請參閱為 iOS WebRTC 開發套件設定 Amazon Cognito。這將產生建置 iOS WebRTC 開發套件所需的身分驗證和授權設定。

4. 在 IDE 中，開啟 awsconfiguration.json 檔案 (從 /Swift/KVSiOSApp)。檔案看起來如下：

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

以執行 為 Android WebRTC 開發套件設定 Amazon Cognito中的步驟所產生的值，更新 awsconfiguration.json。

5. 在 IDE 中，開啟 Constants.swift 檔案 (從 /Swift/KVSiOSApp)。檔案看起來如下：

```
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"
```

以執行 為 Android WebRTC 開發套件設定 Amazon Cognito中的步驟所產生的值，更新 `Constants.swift`。

6. 確保您的 iOS 裝置已連線至執行 XCode 的 Mac 電腦。在 XCode 中，選取連接的裝置，然後建置並執行 WebRTC iOS 開發套件。

   此步驟會在您的 iOS 裝置上安裝名為 `AWSKinesisVideoWebRTCDemoApp` 的應用程式。您可以使用此應用程式，驗證行動、Web 和 IoT 裝置用戶端之間的即時 WebRTC 音訊/視訊串流。

執行 iOS 範例應用程式

請完成下列步驟：

1. 在您的 iOS 裝置上，使用新的 (先建立) 或現有的 Amazon Cognito 帳戶開啟AWSKinesisVideoWebRTCDemoApp並登入。
2. 在中 AWSKinesisVideoWebRTCDemoApp，瀏覽至「通道組態」頁面，然後建立新的信號通道或選擇現有的信號通道。

> **Note**
>
> 目前，使用此 SDK 中的範例應用程式，您只能在中執行一個信號通道。AWSKinesisVideoWebRTCDemoApp

3. (選擇性) 如果您要以檢視器身分連線到此頻道，請選擇唯一的 Client Id (用戶端 ID)。只有在多個檢視器連線至頻道時，才需要用戶端 ID。這有助於頻道的主節點識別各自的檢視器。
4. 選擇以AWS 區域及是否要傳送音訊或視訊資料，或兩者都傳送。
5. 若要驗證 peer-to-peer 串流，請執行下列任一項作業：

> **Note**
>
> 請務必在此示範中使用的所有用戶端上，指定相同的訊號通道名稱、AWS地區、檢視者 ID 和AWS帳戶 ID。

- P 兩個 iOS 設備之間的eer-to-peer 流式傳輸：主設備和查看器
  - 在兩個 iOS 裝置上，使用上述程序下載、建置和執行 iOS WebRTC 開發套件。
  - 以主要模式AWSKinesisVideoWebRTCDemoApp在一個 iOS 裝置上開啟 (選擇 START MASTER) 以啟動新的工作階段 (訊號通道)。

    > **Note**
    >
    > 目前，任何給定的訊號頻道只能有一個主節點。

  - 在第二個 iOS 設備AWSKinesisVideoWebRTCDemoApp上以查看器模式打開，以連接到上述步驟中啟動的信令通道（會話）（選擇 START VIEWER）。

    確認檢視器可以看到主節點的音訊/視訊資料。
- 在嵌入式 SDK 主機和 iOS 設備查看器之間進行eer-to-peer 流式傳輸
  - 在攝影機裝置上，以主節點模式下載、建置和執行 適用於嵌入式裝置的 C WebRTC 開發套件。
  - 在 iOS 裝置上，使用上述程序下載、建置和執行 iOS WebRTC 開發套件。以檢視器模式AWSKinesisVideoWebRTCDemoApp在此 iOS 裝置上開啟，並確認 iOS 檢視器可以看到嵌入式 SDK 主機的音訊/視訊資料。
- 將 iOS 設備作為主設備和 Web 瀏覽器作為查看器之間的eer-to-peer 流媒體
  - 在 iOS 裝置上，使用上述程序下載、建置和執行 iOS WebRTC 開發套件。在此 iOS 設備AWSKinesisVideoWebRTCDemoApp上以主模式打開（選擇 START MASTER）以啟動新的會話（信令通道）。

- 以查看器的適用於 Web 應用程式的 Kinesis Video Streams 與 WebRTC 技術開發套件 JavaScript 身份下載，構建和運行，並驗證查看 JavaScript 器是否可以看到 Android 主機的音頻/視頻。

為 iOS WebRTC 開發套件設定 Amazon Cognito

必要條件

- 我們建議 XCode 用於檢查，編輯和運行應用程序代碼。我們建議使用最新版本。
- 在範例程式碼中，您提供 Amazon Cognito 登入資料。

請依照下列程序設定 Amazon Cognito 使用者集區和身分集區。

設定使用者集區

設定使用者集區

1. 登入 Amazon Cognito 主控台並確認區域是否正確。
2. 在左側導覽中選擇 [使用者集區]。
3. 在 [使用者集區] 區段中選擇 [建立使用者集區]。
4. 完成以下各節：

   a. 步驟 1：設定登入體驗-在 Cognito 使用者集區登入選項區段中，選取適當的選項。

      選取下一步。

   b. 步驟 2：設定安全性需求-選取適當的選項。

      選取下一步。

   c. 步驟 3：配置註冊體驗-選擇適當的選項。

      選取下一步。

   d. 步驟 4：設定郵件傳遞-選取適當的選項。

      在 IAM 角色選取欄位中，選取現有角色或建立新角色。

      選取下一步。

   e. 步驟 5：集成您的應用程序-選擇適當的選項。

在初始應用程式用戶端欄位中，選擇機密用戶端。

選取下一步。

f. 步驟 6：檢閱並建立-檢閱先前章節中的選取項目，然後選擇 [建立使用者集區]。

5. 在 [使用者集區] 頁面上，選取您剛建立的集區。

   複製使用者集區 ID 並記下此項，以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.PoolId`。

6. 選取 [應用程式整合] 索引標籤，然後前往頁面底部。

7. 在 [應用程式用戶端清單] 區段中，選擇您剛建立的應用程式用戶端名稱。

   複製用戶端 ID 並記下以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.AppClientId`。

8. 顯示用戶端密碼，並記下以供稍後使用。在檔`awsconfiguration.json`案中，這是`CognitoUserPool.Default.AppClientSecret`。

設定身分集區

設定身分集區

1. 登入 Amazon Cognito 主控台並確認區域是否正確。
2. 在左側導覽中，選擇 [識別集區]。
3. 選擇 建立身分池。
4. 設定身分識別集區。

   a. 步驟 1：設定識別集區信任-完成下列各節：

   - 使用者存取-選取已驗證存取
   - 已驗證身分來源-選取 Amazon Cognito 使用者集區

   選取下一步。

   b. 步驟 2：設定權限-在 [已驗證的角色] 區段中，完成下列欄位：

   - IAM 角色-選取 [建立新的 IAM 角色]
   - IAM 角色名稱-輸入名稱並記下它以供稍後步驟使用。

選取下一步。

c. 步驟 3：Connect 身分識別提供者-在 [使用者集區詳細資料] 區段中，完成下列欄位：

- 使用者集區 ID-選取您之前建立的使用者集區。
- 應用程式用戶端 ID-選取您之前建立的應用程式用戶端 ID。

選取下一步。

d. 步驟 4：配置屬性-在 [身分識別集區名稱] 欄位中輸入名稱。

選取下一步。

e. 步驟 5：檢閱並建立-檢閱您在每個區段中的選擇，然後選取 [建立身分集區]。

5. 在 [身分識別集區] 頁面上，選取新的身分集區。

複製身分集區 ID 並記下此項目以供稍後使用。在檔`awsconfiguration.json`案中，這是`CredentialsProvider.CognitoIdentity.Default.PoolId`。

6. 更新 IAM 角色的許可。

a. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。

b. 在左側導覽列中，選擇 [角色]。

c. 尋找並選取您在上面建立的角色。

> **Note**
>
> 如有需要，請使用搜尋列。

d. 選取附加的權限原則。

選擇 Edit (編輯)。

e. 選取 JSON 索引標籤，並以下列項目取代原則：

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

```
            "Action": [
                "cognito-identity:*",
                "kinesisvideo:*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

選取下一步。

f. 如果尚未選取，請選取 [將此新版本設定為預設版本] 旁邊的核取方塊。

選取儲存變更。

WebRTC C SDK 的用戶端度量

使用具有 WebRTC 的 Amazon Kinesis Video Streams 建置的應用程式由各種移動零件組成，包括聯網、信號、候選人交換、對等連線和資料交換。使用 C 語言 WebRTC 的 Kinesis Video Streams 支援各種用戶端指標，可讓您監控並追蹤這些元件在應用程式中的效能和使用情況。支援的指標分為兩大類：專為 Kinesis Video Streams 實作信號和網路而定義的自訂量度，以及衍生自 W3C 標準的媒體和資料相關通訊協定特定度量。請注意，目前只有 W3C 標準量度的子集支援含 WebRTC 的 Kinesis Video Streams (C)。

主題

- 訊號指標
- 支援的 WebRTC 3C 標準度量

訊號指標

信號指標可用於了解應用程序運行時信令客戶端的行為。您可以使用 `STATUS signalingClientGetMetrics (SIGNALING_CLIENT_HANDLE, PSignalingClientMetrics)` API 取得這些訊號指標。這是一個示例使用模式：

```
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));
```

的定義`signalingClientStats`可以在 Stats. h 中找到。

目前支援下列訊號指標：

指標	描述	
cpApiCall延遲	計算控制平面 API 呼叫的延遲。使用指數移動平均線 (EMA) 進行計算。相關的呼叫包括：描述頻道、CreaTech 通道和代理通道。 getChannelEndpoint	
dpApiCall延遲	計算資料平面 API 呼叫的延遲。使用指數移動平均線 (EMA) 進行計算。相關聯的呼叫包括： getIceConfig。	
signalingClientUptime	這表示客戶端對象存在的時間。每次調用此指標時，都會發出最新的正常運行時間值。	
連線持續時間	如果已建立連線，則會發出連線存在的持續時間。否則，0 的值被發射。這與信令客戶端正常運行時間不同，因為連接來來去去，但 signaling ClientUptime 表示客戶端對象本身。	
numberOfMessages已傳送	當對等傳送選件、答案或 ICE 候選項時，會更新此值。	
numberOfMessages已收到	與 Send 不同，此量度numberOfMessages會針對任	

指標	描述
	何類型的信號訊息進行更新。中SIGNALING_MESSAGE_TYPE 提供信令訊息的類型。
iceRefreshCount	這在調用時 getIceConfig 遞增。呼叫此速率是以接收之 ICE 組態的一部分 TTL 為基礎。每次接收到一組新的 ICE 設定時，計時器就會設定為下次重新整理，考慮到組態中認證的有效性減去一些寬限期。
numberOfErrors	計數器用於跟踪信令客戶端中生成的錯誤的數量。追蹤取得 ICE 組態、取得訊號狀態、追蹤訊號指標、傳送訊號訊息，以及將信號用戶端連接至 Web 通訊端以傳送/接收訊息時產生的錯誤。
numberOfRuntime錯誤	此測量結果包括信令從屬端核心執行時所發生的錯誤。這裡會追蹤重新連線失敗、訊息接收失敗和 ICE 組態重新整理錯誤等案例。
numberOfReconnects	每次重新連線時，量度都會遞增。這是了解設置中網絡連接穩定性的有用指標。

支援的 WebRTC 3C 標準度量

目前，使用 WebRTC C SDK 構建的應用程序支持 W3C 標準度量的子集。這些分為以下幾類：

- 網路：

  - Ice Queate: 這些量度提供有關所選本機和遠端候選人的資訊，以便在對等之間進行資料交換。這包括候選人的伺服器來源、IP 位址、為通訊選取的候選人類型，以及候選優先順序。這些量度可做為快照報表。
  - Ice Server：這些指標用於收集有關支持的不同 ICE 服務器的操作信息。這在嘗試了解主要用於通訊和連線檢查的伺服器時非常有用。在某些情況下，如果候選人的收集失敗，檢查這些指標也很有用。
  - 冰候選對：這些指標用於了解對等之間正在交換的字節/數據包的數量以及與時間有關的測量。
- 媒體和數據：
  - 遠端輸入 RTP：這些指標代表寄件者所傳送之資料串流的端點透視。
  - 輸出 RTP：這些測量結果提供外送 RTP 串流的相關資訊。在分析波濤洶湧的流或流式傳輸停止時，它們也非常有用。
  - 輸入 RTP：這些測量結果提供有關傳入媒體的資訊。
  - 資料通道指標：這些指標可協助您分析透過資料通道傳送和接收的訊息和位元組數目。可以使用通道 ID 提取指標。

您可以使用 STATUS rtcPeerConnectionGetMetrics (PRtcPeerConnection, PRtcRtpTransceiver, PRtcStats) API 收集與 ICE、RTP 和資料通道相關的指標。這是一個用法示例：

```
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);
```

以下是另一個示例，顯示了獲取收發器相關統計信息的使用模式：

```
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);
```

在上面的例子中，如果第二個參數 to rtcPeerConnection GetMetrics () 是 NULL，則返回列表中第一個收發器的數據。

的定義 rtcStatsObject 可以在 Stats.h 中找到，而的定義 RtcStats 可以在包含 .h 中找到。

您可以在 WebRTC C SDK 存放庫的範例目錄和 Kinesis 影片串流示範存放庫中找到 API 的範例使用情況和不同指標的範例使用方式。

目前，使用 WebRTC C SDK 構建的應用程序支持以下 W3C 標準指標。

主題

- 聯網
- 媒體
- 資料通道

聯網

ICE 伺服器度量：

指標	描述
URL	正在追蹤的 STUN/Turn 伺服器的網址
連接埠	用戶端使用的連接埠號碼
通訊協定	從 ICE 伺服器 URI 擷取的傳輸通訊協定。如果該值是 UDP，ICE 嘗試翻轉 UDP，否則 ICE 嘗試翻轉 TCP/TLS。如果 URI 不包含傳輸，ICE 會嘗試翻轉 UDP 和 TCP/TLS。在 STUN 伺服器的情況下，此欄位是空的。
傳送的要求總數	該值被更新為每個 srflx 候選請求，同時從回合候選發送綁定請求。
收到的回覆總數	每次收到 STUN 繫結回應時，都會更新該值。

指標	描述	
總往返時間	每次收到請求的等效回應時，都會更新該值。請求數據包在哈希映射中跟踪，並將校驗和作為密鑰。	

ICE 候選人統計：僅包括有關所選候選人（本地和遠程）的信息。

指標	描述	
address	這表示本地和遠程候選人的 IP 地址。	
port	候選人的連接埠號碼	
protocol	用於獲取候選人的協議。有效值是 UDP/TCP。	
候選人類型	選擇的候選人類型-主機，srflx 或繼電器。	
priority	所選本機和遠端候選人的優先順序。	
網址	所選當地候選人的來源。這表示選擇的候選人是否從 STUN 服務器或 TURN 服務器接收到。	
中繼協議	如果使用 TURN 伺服器取得選取的本機候選項目，此欄位會指出使用何種通訊協定來取得它。有效值為 TCP/UDP。	

ICE 候選對統計數據：僅包括有關所選候選對的信息。

指標	描述	
localCandidateId	配對中所選本地候選人的 ID。	
remoteCandidateId	配對中所選遠端候選項的 ID。	
state	正在檢查的候選人對的狀態。	
提名	設置為 TRUE，因為統計數據被提取選定的候選對。	
數據包已發送	傳送的封包數目。這是在呼叫中的.呼writeFrame 叫中計算的。此信息也可以從傳出的 RTP 統計數據中提取，但是由於 Ice 候選對包含時間 lastPacketSent 戳，因此計算兩個時間點之間發送的數據包數可能很有用。	
已接收封包	每次調用 incomingDataHandler 時都會更新此選項。	
字節發送	這是在writeFrame() 呼叫iceAgentSendPacket() 中計算的。這在計算比特率時非常有用。目前，這還包括標題和填充，因為 ICE 層對 RTP 數據包格式沒有影響。	
字節接收	每次調用 incomingDataHandler 時都會更新此選項。目前，這還包括標題和填充，因為 ICE 層對 RTP 數據包格式沒有影響。	
lastPacketSent時間戳	每次傳送封包時都會更新此項目。這可以與 PackketsSend 和應用程序中記錄的開始時間	

指標	描述	
	結合使用到當前數據包的傳輸速率。	
lastPacketReceived時間戳	這會在中接收資料時更新`incomingDataHandler()` 。這可以與 PackketsReceiver 一起使用，以推導出當前的數據包接收速率。開始時間必須記錄在`transceiverOnFrame()` 回調中的應用程序層。	
firstRequestTimestamp	在`iceAgentSendStunPacket()` 成功傳送第一個 STUN 繫結要求時記錄。這可以與 lastRequestTimestamp 和請求一起使用，以查找 STUN 綁定請求之間的平均時間。	
lastRequestTimestamp	每次成功傳送 STUN 繫結要求時都會記錄。`iceAgentSendStunPacket()`	
lastResponseTimestamp	每次收到 STUN 綁定響應時都會記錄。	
totalRoundTrip時間	收到請求的綁定響應時更新。請求和響應被映射到基於校驗和哈希表。	
currentRoundTrip時間	最近的往返時間在收到候選人對的請求的綁定響應時更新。	
已收到的要求	針對每個收到的 STUN 繫結要求進行更新的計數器。	

指標	描述	
要求已傳送	每個傳送的 STUN 繫結要求都會更新的計數器。`iceAgentSendStunPacket()`	
回應	針對中的繫結要求而傳送的每個 STUN 繫結回應上更新的計數器。`handleStunPacket()`	
收到的回應	在`handleStunPacket()` 中收到的每個 STUN 綁定響應上更新的計數器。	
packetsDiscardedOn發送	封包傳送失敗時更新。換句話說，這會在`iceUtilsSendData()` 失敗時更新。這可以確定在特定持續時間內丟棄的數據包百分比。	
bytesDiscardedOn發送	封包傳送失敗時更新。換句話說，這會在`iceUtilsSendData()` 失敗時更新。這在判斷特定持續時間內丟棄的封包百分比時非常有用。請注意，計數器也包含封包的標頭。	

媒體

出埠 RTP 統計資料

指標	描述	
voiceActivityFlag	這是目前在包含 .h 中`RtcEncoderStats` 定義的一部分。如果最後一個音訊	

指標	描述	
	封包含語音，則旗標會設定為 TRUE。範例中目前未設定旗標。	
數據包已發送	這表示為所選 SSRC 送出的 RTP 封包總數。這是 https://www.w3.org/TR/webrtc-stats/#sentrtpstats-dict 的一部分*，並作為出站統計信息的一部分包含在內。每次調用 WriteFrame () 時，這都會遞增。	
字節發送	傳送的位元組總數 (不包括 RTP 標頭和填充)。這在每個寫幀調用上都會更新。	
編碼器實作	這是由應用層作為 RtcEncoderStats 對象的一部分更新。	
packetsDiscardedOn發送	如果 ICE 代理程式因為`iceAgentSendPacket`呼叫中的任何原因而無法傳送加密的 RTP 封包，則會更新此欄位。	
bytesDiscardedOn發送	如果 ICE 代理程式因為`iceAgentSendPacket`呼叫中的任何原因無法傳送加密的 RTP 封包，也會更新此欄位。	
框架	只有當媒體流粘性類型為媒體流追踪 _ 視頻時，這才會遞增。	

指標	描述	
hugeFramesSent	此計數器會針對影格平均大小 2.5 倍的影格進行更新。幀的大小是通過計算 fps（基於最後一個已知的幀計數時間和以時間間隔編碼的幀數）並使用應用程序 RtcEncoderStats 設置的 targetBtBitt 獲得。	
框架編碼	此計數器僅在幀的編碼成功後更新視頻軌道。它在每個寫幀調用更新。	
keyFramesEncoded	成功編碼關鍵影格後，此計數器僅針對視訊軌道進行更新。它在每個寫幀調用更新。	
framesDiscardedOn發送	當幀發送由於`iceAgentSendPacket` 調用失敗而失敗時，這將更新。一個框架由一組數據包組成，當前，framesDiscardedOn如果發送因為錯誤而丟棄任何數據包，則發送失敗。	
框架寬度	理想情況下，這代表了最後一個編碼幀的幀寬度。目前，這是由應用程序設置為一個值作為 RtcEncoderStats * 的一部分，並沒有太大的意義。	
框架高度	這理想地代表了最後一個編碼幀的幀的高度。目前，這是由應用程序設置為一個值的一部分， RtcEncoderStats 並沒有太大的意義。	

指標	描述	
frameBitDepth	這代表了最後一個編碼幀的每像素寬度的位深度。目前，這是由應用程序設置的一部分，RtcEncoderStats 並轉換為出站統計信息。	
納克伯爵	每次在 RTP 封包上收到 NACK 並重新嘗試傳送封包時，都會更新此值。堆棧支持在接收 NACK 時重新傳輸數據包。	
第一次計數	接收 FIR 封包onRtcpPacket時會更新該值 (-> 安裝封包)。它指示流落後的頻率，並且必須跳過幀才能 catch。FIR 封包目前尚未解碼來擷取欄位，因此，即使已設定計數，也不會採取任何動作。	
普利康	接收 PLI 封包時會更新該值 (onRtcpPacket-> 安裝封包)。它表示一個或多個幀丟失了一定數量的編碼視頻數據。	
切片計算機	此值會在接收 SLI 封包時更新 (onRtcpPacket-> 連結封包)。它指出封包遺失影響單一幀的頻率。	
qualityLimitationResolution變更	目前，堆棧支持此指標，但是，不會監視每個編碼幀的幀寬度和高度。	
lastPacketSent時間戳	最後一個封包傳送時的時間戳記。它在每個寫幀調用更新。	

指標	描述	
headerBytesSent	針對此 SSRC 傳送的 RTP 標頭和填補位元組總數 (不包括實際的 RTP 裝載)。	
bytesDiscardedOn發送	當訊框傳送因 iceAgentSend 封包呼叫失敗而失敗時，會更新此選項。一個幀由一組數據包組成，這反過來又由字節組成，當前，如果任何數據包被丟棄，而發 bytesDiscardedOn 送失敗，而發送，因為錯誤。	
retransmittedPacketsSent	接收 PLI/SLI/NACK 時重新傳輸的封包數目。目前，堆疊僅計算 NACK 重新傳送的封包，因為不支援以 PLI 和 SLI 為基礎的重新傳輸。	
retransmittedBytesSent	接收 PLI/SLI/NACK 時重新傳輸的位元組數。目前，堆棧僅計算 NACK 重新發送的字節，因為不支持基於 PLI 和 SLI 的重傳。	
目標位元速率	這是在應用程序級別中設置的。	
totalEncodedBytes目標	每次編碼幀時，這會增加目標幀大小（以字節為單位）。這是使用框架結構中的大小參數進行更新。	
framesPerSecond	這是根據上次已知編碼影格所記錄的時間，以及在一秒內傳送的影格數目來計算。	

指標	描述	
totalEncodeTime	這會在應用程式中設定為任意值，並在內部轉換為輸出統計資料。	
totalPacketSend延遲	由於數據包立即發送 iceAgentSend數據包，因此當前設置為 0。	

遠端入埠 RTP 統計資料：

指標	描述	
roundTripTime	該值是在接收 RTCP 數據包類型 201（接收器報告）時從 RTCP 接收器報告中提取的。該報告包括上次發件人報告和自上次發送人報告以來計算往返時間的延遲。寄件者報告大約每 200 毫秒會產生一次，其中包含資訊，例如傳送的封包數目和從輸出統計資料擷取的傳送位元組。	
totalRoundTrip時間	計算的往返時間總和	
分數丟失	代表自上一個傳送者/接收器 Report Lost 傳送以來，SSRC 遺失的 RTP 封包比例。	
報告收到	每次收到接收者報告類型封包時更新。	
roundTripTime測量	指出 SSRC 收到的報告總數，其中包含有效的往返時間。但是，目前，無論如何，此值	

指標	描述	
	都會增加，因此它的含義與 ReportsReceived 相同。	

入站 RTP 統計資料：

指標	描述	
已接收封包	當收到特定 SSRC 的封包時，計數器會更新。	
抖動	此測量結果表示針對特定 SSRC 測量的封包抖動 (以秒為單位)。	
jitterBufferDelay	此測量結果代表抖動緩衝區中每個封包使用的時間總和。	
jitterBufferEmitted伯爵	從抖動緩衝區中傳出的音訊樣本或視訊影格總數。	
丟棄的包	當抖動緩衝區已滿且封包無法推入其中時，計數器會更新。這可用來計算固定持續時間內捨棄的封包百分比。	
框架掉落	呼叫時會更新此值。`onFrameDroppedFunc()`	
lastPacketReceived時間戳	代表接收此 SSRC 最後一個封包的時間戳記。	
headerBytesReceived	計數器會在接收 RTP 封包時更新。	

指標	描述	
字節接收	接收的位元組數。這不包括標頭字節。此指標可用於計算傳入的位元速率。	
packetsFailedDecryption	當 SRTP 封包解密失敗時，這會遞增。	

資料通道

資料通道量度

指標	描述	
標籤	標籤是被檢查的數據通道的名稱。	
protocol	由於我們的堆棧使用 SCTP，該協議被設置為常量 SCTP。	
dataChannelIdentifier	用於唯一識別資料通道的偶數或奇數識別碼。如果 SDK 是提供者，則此值會更新為奇數值，如果 SDK 是回答者，則會更新為偶數值。	
state	查詢統計資料時的資料通道狀態。目前，支持的兩種狀態是 RTC _ 數據 _ 通道 _ 狀態 _ 連接（當創建通道時）和 RTC_DATA_ 通道 _ 狀態 _ 打開（在 onOpen（ ）事件中設置）。	
訊息發送	SDK 透過資料通道傳送訊息時，計數器會更新。	

指標	描述	
字節發送	計數器會更新為傳送出的訊息中的位元組。這可以用來了解由於失敗而未發送多少字節，即了解發送的字節的百分比。	
訊息已收到	測量結果會在onMessage() 回呼中遞增。	
字節接收	測量結果會在onMessage() 回呼中產生。	

Amazon Kinesis Video Streams WebRTC 技術攝入

Amazon Kinesis Video Streams 提供透過 WebRTC 即時串流視訊和音訊到雲端的功能，以進行儲存、播放和分析處理。客戶可以使用我們增強的 WebRTC 技術 SDK 和雲端 API 來實現即時串流，以及將媒體擷取到雲端。

若要開始使用，您可以在任何具有視訊感應器的安全攝影機或裝 AWS IoT 置上安裝具有 WebRTC SDK 的 Amazon Kinesis Video Streams，並使用我們的 API 啟用延遲不到 1 秒的媒體串流，以及在雲端擷取和儲存。一旦擷取，您就可以透過我們的 easy-to-use API 存取您的資料。Amazon Kinesis Video Streams 可讓您播放視訊以進行即時和隨選檢視，以及透過與 Amazon Rekognition 視訊和整合，快速建置充分利用電腦視覺和視訊分析的應用程式。 SageMaker

主題

API 操作

使用下列 API 操作來設定 Amazon Kinesis Video Streams WebRTC 擷取：

- DescribeMappedResourceConfiguration
- DescribeMediaStorageConfiguration
- JoinStorageSession
- UpdateMediaStorageConfiguration

開始使用 WebRTC 技術擷取和儲存

Amazon Kinesis Video Streams 提供透過 WebRTC 即時串流視訊和音訊到雲端的功能，以進行儲存、播放和分析處理。本主題將提供設定和使用我們的 WebRTC 技術 SDK 和雲端 API 的 step-by-step 說

明，以啟用即時串流和媒體擷取到雲端。這些指示包括使用 AWS Command Line Interface 和 Kinesis Video Streams 主控台的指引。

在您第一次使用 Amazon Kinesis Video Streams 與 WebRTC 技術，請參閱。the section called “設置一個 AWS 帳戶”

使用 WebRTC 技術信號通道建立室壁運動視訊串流

有兩種方法可以創建一個信令通道， AWS Management Console 或. AWS CLI

使用建立信令通道 AWS Management Console

1. 在中 AWS Management Console，開啟 Kinesis Video Streams 主控台。
2. 在左側導覽中，選取 [信號通道]。

   選擇 Create signaling channel (建立訊號頻道)。
3. 在 [建立新的信令通道] 頁面上，輸入信號通道的名稱。

   保留預設 T ime-to-live (TTL) 值為 60 秒。
4. 選擇 Create signaling channel (建立訊號頻道)。
5. 建立信號通道後，請檢閱頻道詳細資料頁面上的詳細資料。

   記下通道 ARN 的。

使用建立信令通道 AWS CLI

在中 AWS CLI，鍵入：

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

建立 串流

有兩種方法可以創建流， AWS Management Console 或 AWS CLI.

> **Important**
>
> WebRTC 擷取需要啟用資料保留功能的 Amazon Kinesis 視訊串流。

使用建立串流 AWS Management Console

1. 在中 AWS Management Console，開啟 Kinesis Video Streams 主控台。
2. 在左側導覽中，選取 [儀表板]。

   選擇 Create video stream (建立影片串流。
3. 在 [建立新視訊串流] 頁面上，輸入此測試串流的名稱。

   使用預設組態。
4. 選擇 Create video stream (建立影片串流。
5. 建立串流後，請檢閱「視訊串流」頁面上的詳細資料。

   記下資料流 ARN。

使用建立串流 AWS CLI

在中 AWS CLI，鍵入：

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

設定媒體擷取和儲存

在中 AWS CLI，設定媒體儲存設定。

如果您使用 AWS Management Console 來設定串流和/或頻道，請從這些步驟複製並貼上資源 ARN。

如果您使用 AWS CLI 來設定串流和/或頻道，請執行下列動作以擷取資源 ARNS。

- 若要擷取通道 ARN，請輸入：

  ```
  $ aws kinesisvideo describe-signaling-channel --channel-name your-channel-name --
  region us-west-2
  ```
- 若要擷取串流 ARN，請輸入：

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

當您有 ARN 時，請設定媒體儲存設定。類型：

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

> **Important**
>
> 如果啟StorageStatus用，則不會再發生直接 peer-to-peer (主要檢視器) 連線。對等直接連線至儲存區工作階段。您必須呼叫 JoinStorageSession API 來觸發 SDP 選件傳送，並建立對等與儲存工作階段之間的連線。

擷取媒體

有以下限制：

- 會話持續時間：一小時，最長
- 信號通道：每個啟用儲存設定的帳戶最多 100 個

從瀏覽器擷取媒體

> **Important**
>
> Chrome 瀏覽器是唯一受支持的瀏覽器。

1. 在範例頁面中使用 WebRTC 技術開發套件開啟 Amazon Kinesis Video Streams。 JavaScript
2. 請填妥下列資訊：

   - KVS 端點。在「地區」欄位中，選取您的地區。

     例如 us-west-2。
   - AWS 憑證

     完成下列欄位：

    - Access Key ID (存取金鑰 ID)
    - Secret Access Key (私密存取金鑰)
    - 工作階段權杖。範例應用程式同時支援臨時和長期登入資料。如果您使用長期 IAM 登入資料，請將此欄位保留空白。如需詳細資訊，請參閱 IAM 中的臨時安全登入資料
  - 信令通道。在「通道名稱」欄位中，輸入您先前設定的訊號通道名稱。如需詳細資訊，請參閱 the section called “設定媒體擷取和儲存”。
  - 軌道。選擇發送視頻和發送音頻。
  - WebRTC 技術擷取和儲存。選取 [自動擷取和儲存對等聯結]。
3. 選取「啟動主版」。

   如果信令通道設定為使用 DescribeMediaStorageConfigurationAPI 擷取，範例應用程式將自動叫用 JoinStorageSessionAPI 以啟動 WebRTC 擷取工作流程。

從 WebRTC C 開發套件擷取媒體

依照程the section called “適用於嵌入式裝置的 C WebRTC 開發套件”序建置範例應用程式。

1. 使用您的 AWS 帳戶 憑據設置您的環境：

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

   如果您使用臨時 AWS 憑據，請同時導出會話令牌：

   ```
   export AWS_SESSION_TOKEN=YourSessionToken
   ```

2. 運行示例：

   主樣本

   導覽至資build料夾並使用「1」作為第二個引數。類型：

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

   流媒體主樣本

導覽至資build料夾並使用 "audio-video-storage" 作為第二個引數。類型：

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

這開始 WebRTC 技術攝入。

> **Note**
>
> 您所提供的訊號通道必須設定為儲存裝置。使用 DescribeMediaStorageConfigurationAPI 進行確認。

檢視 Kinesis Video Streams 中的媒體

1. 開啟 Amazon Kinesis Video Streams 媒體檢視器。
2. 完成下列欄位：
   - 區域。選擇 us-west-2。
   - AWS 存取金鑰
   - AWS 秘密金鑰
   - 串流名稱
   - 播放模式。選取 [直播]。
3. 選取 [開始播放]。

安全

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您將受益於資料中心和網路架構，專為滿足最敏感安全性組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 — AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎架構。 AWS 還為您提供可以安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 AWS 合規計劃的一部分。如要了解適用於 Kinesis Video Streams 的合規計劃，請參閱 合規計劃的AWS 服務範圍。
- 雲端中的安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

本文件可協助您了解在搭配 WebRTC 使用 Amazon Kinesis Video Streams 時，如何套用共同的責任模型。下列主題說明如何使用 WebRTC 設定 Amazon Kinesis Video Streams，以符合您的安全性和合規目標。您也將學習如何使用其他可協助您透過 WebRTC 資源監控和保護 Amazon Kinesis Video Streams 的 AWS 服務。

主題

- 使用 Kinesis Video StreamsAWS Identity and Access Management
- 使用 WebRTC 技術進行 Amazon Kinesis Video Streams 的合規驗證
- Kinesis Video Streams with WebRTC 的彈性
- 使用 WebRTC 技術在室運動視訊串流中的基礎架構安全
- 使用 WebRTC
- WebRTC 技術加密

使用 Kinesis Video StreamsAWS Identity and Access Management

AWS Identity and Access Management搭配 Amazon Kinesis Video Streams Kinesis Video StreAWS ams

如需 IAM 的詳細資訊，請參閱下列各項：

- AWS Identity and Access Management (IAM)
- 入門

- IAM 使用者指南

內容

政策語法

IAM 政策為包含一或多個陳述式的 JSON 文件。每個陳述式的結構如下所示：

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

陳述式由各種元素組成：

- Effect (效果)：效果 可以是 Allow 或 Deny。根據預設，IAM 使用者沒有使用資源和 API 動作的許可，因此所有請求均會遭到拒絕。明確允許覆寫預設值。明確拒絕覆寫任何允許.
- Action (動作)：動作 是您授予或拒絕許可的特定 API 動作。
- Resource (資源)：受動作影響的資源。若要在陳述式中指定資源，您必須使用其 Amazon Resource Name (ARN)。
- Condition (條件)：條件為選擇性。您可以使用它們來控制何時政策開始生效。

建立和管理 IAM 政策時，您可能需要使用 IAM 政策產生器和 IAM 政策模擬器。

使用 Kinesis Video Streams

在 IAM 政策陳述式中，您可以從任何支援 IAM 的服務指定任何 API 動作。Kinesis Video Streams 對使用下列字首：`kinesisvideo:`。例如：`kinesisvideo:CreateSignalingChannel`、`kinesisvideo:ListSignalingChannels` 和 `kinesisvideo:DescribeSignalingChannel`。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

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

您也可以使用萬用字元指定多個動作。例如，您可以指定名稱開頭有「Get」文字的所有動作，如下所示：

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

若要指定所有 Kinesis Video Streams 作業，請使用星號 (*) 萬用字元，如下所示：

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

如需 Kinesis Video Streams API 動作的完整清單，請參閱 Kinesis Video Streams API 參考資料。

Kinesis Video Streams 的亞馬遜資源名稱 (ARN)

每個 IAM 政策陳述式都會套用到您使用其 ARN 指定的資源。

針對 Kinesis Video Streams 使用下列 ARN 資源格式：

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

例如：

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

您可以使用獲取通道的 ARN DescribeSignalingChannel。

授與其他 IAM 帳戶存取 Kinesis 影片串流的存取權

您可能需要授予其他 IAM 帳戶的權限，才能使用 WebRTC 技術訊號通道在 Kinesis Video Streams 上執行作業。服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改

和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務服務。

使用 Kinesis Video Streams

下列範例原則示範如何透過 WebRTC 頻道控制使用者對 Kinesis Video Streams 的存取。

Example 1：允許使用者從任何訊號頻道取得資料

此政策可讓使用者或群組在任何訊號頻道上執行 `DescribeSignalingChannel`、`GetSignalingChannelEndpoint`、`ListSignalingChannels` 和 `ListTagsForResource` 操作。

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

Example 2：允許使用者建立訊號頻道

此政策可讓使用者或群組執行 `CreateSignalingChannel` 操作。

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

```
    ]
}
```

Example 3: 允許使用者透過 WebRTC 資源完全存取所有 Kinesis 影片串流和室運動影片串流

此原則可讓使用者或群組在任何資源上執行任何 Kinesis Video Streams 作業。此政策適用於管理員。

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

Example 4：允許使用者從特定訊號頻道取得資料

此政策允許使用者或群組從特定訊號頻道取得資料。

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

使用 WebRTC 技術進行 Amazon Kinesis Video Streams 的合規驗證

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- Amazon GuardDuty — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。 GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

Kinesis Video Streams with WebRTC 的彈性

AWS 全球基礎架構是以 AWS 區域 與可用區域為中心建置的。AWS 區域 提供多個分開且隔離的實際可用區域，並以具備低延遲、高輸送量和高度備援特性的聯網相互連結。您可以使用區域來設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 與可用區域的詳細資訊，請參閱AWS全球基礎架構。

使用 WebRTC 技術在室運動視訊串流中的基礎架構安全

作為受管服務，Kinesis Video Streams (包括其 WebRTC 功能) 受到 Amazon 網路服務：安AWS全程序概觀白皮書中所述的全球網路安全程序的保護。

您可以使用AWS已發佈的 API 呼叫透過網路存取 Kinesis 視訊串流。用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。建議使用 TLS 1.3 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取索引鍵來簽署。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

使用 WebRTC

在您開發和實作自己的安全政策時，可考慮使用 Amazon Kinesis Video Streams (包括其 WebRTC 功能) 提供的多種安全功能。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

如需您遠端裝置的安全最佳實務，請參閱裝置代理程式的安全最佳實務。

實作最低權限存取

當您授予許可時，需決定哪些使用者會取得得得得得得得得得到 Kinesis Video Streams 的許可。您還需針對這些資源啟用允許執行的動作，因此您只應授予執行任務所需的許可。對降低錯誤或惡意意圖所引起的安全風險和影響而言，實作最低權限存取是相當重要的一環。

例如，將資料傳送至 Kinesis Video Streams 的製作人只需要`PutMedia`、`GetStreamingEndpoint`和`DescribeStream`。請勿授予生產者應用程式所有動作 (*) 或其他動作 (例如 `GetMedia`) 的許可。

如需詳細資訊，請參閱什麼是最低權限以及為什麼需要它？

使用 IAM 角色

製作者和用戶端應用程式必須具有有效的認證才能存取 Kinesis 視訊串流。您不應將AWS登入資料直接在用戶端應用程式中，或在 Amazon S3 儲存體中。這些是不會自動輪換的長期憑證，如果遭到盜用，可能會對業務造成嚴重的影響。

反之，您應使用 IAM 角色來為您的生產者和用戶端應用程式管理暫時性登入資料，以便存取 Kinesis Video Streams。使用角色時，您不必使用長期憑證來存取得其他資源。

如需詳細資訊，請參閱《IAM 使用者指南》中的以下主題：

- IAM 角色
- 常見的角色方案：使用者、應用程式和服務

用 CloudTrail 於監控 API 呼叫

Kinesis Video Streams by WebRTC 與整合，這是一種服務AWS CloudTrail，提供由使用者、角色或在 Kinesis Video Streams in WebRTC 中所採取得之動作的記錄，以及在 Kinesis Video Streams 中的記錄。AWS

您可以利用所 CloudTrail收集的資訊來判斷向 Kinesis Video Streams 發 WebRTC 的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

如需詳細資訊，請參閱 the section called “使用 WebRTC 技術 API 呼叫記錄 Kinesis Video Streams AWS CloudTrail”。

WebRTC 技術加密

端對端加密是 Amazon Kinesis Video Streams 搭配 WebRTC 的強制性功能，Kinesis 影片串流會在所有元件上強制執行此功能，包括訊號和媒體或資料串流。無論是 peer-to-peer 通過 Kinesis Video Streams TURN 端點進行通信還是中繼，所有 WebRTC 技術通信都通過標準化加密協議進行安全加密。

信令消息使用安全的 WebSockets（WSS）進行交換，數據流使用數據報傳輸層安全性（DTLS）進行加密，媒體流使用安全實時傳輸協議（SRTP）進行加密。

監控

為了維護 Amazon Kinesis Video Streams with WebRTC 和AWS解決方案。您應該收集 AWS 解決方案全面的監控資料，以便在出現多點故障時更輕鬆的進行偵錯。不過，在您開始監控 Kinesis Video Streams with WebRTC 之前，應先建立監控計畫，為下列問題提供解答：

- 監控目標是什麼？
- 要監控哪些資源？
- 監控這些資源的頻率為何？
- 要使用哪些監控工具？
- 誰將執行監控任務？
- 發生問題時應該通知誰？

在您定義監控目標並建立監控計劃之後，下一步是為環境中的 Kinesis Video Streams with WebRTC 正常效能建立基準。您應該在不同的時間和不同的負載條件下測量 Kinesis Video Streams with WebRTC 效能。當您監控 Kinesis Video Streams with WebRTC 時，您已收集的監控資料應該保存一份歷史記錄。您可以比較目前的 Kinesis Video Streams with WebRTC 效能資料與歷史資料，協助您辨別正常效能模式和效能異常狀況，並設法解決可能發生的問題。

主題

使用 WebRTC 指標監控 Kinesis Video Streams with WebRTC 指標CloudWatch

您可以使用 Amazon 監控 Kinesis Video Streams with WebRTCCloudWatch，收集來自 Kinesis Video Streams 的原始資料，並處理為便於讀取且幾近即時的指標。這些統計資料會記錄 15 個月的時間，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。

Kinesis Video Streams 提供下列指標：

主題

- TURN 指標

訊號指標

指標名稱	維度	單位	Description (描述)
Failure (失敗)	運算,SignalingChannel名稱	計數	如果維度中提及的運算返回 200 狀態碼響應，則會發出 '0'。否則，則會發出 '0'。
Latency (延遲)	運算,SignalingChannel名稱	毫秒	測量從服務接收請求直到服務傳回回應的時間。
MessagesTransferred.計數	SignalingChannel名稱	計數	給定頻道傳輸 (傳送和接收) 的訊息總數。

`Operation` 維度可套用至下列任何 API：

- ConnectAs主
- ConnectAs檢視者
- SendSdp優惠
- SendSdp答案
- SendCandidate
- SendAlexaOfferTo主
- GetIceServerConfig
- Disconnect

TURN 指標

指標名稱	維度	單位	Description (描述)	
轉 ConnectedMinutes	Signaling Channel名稱	計數	對於一分鐘內用來串流資料的每個 TURN 配置，發出 '1'。	

使用 WebRTC 技術 API 呼叫記錄 Kinesis Video Streams AWS CloudTrail

Amazon Kinesis Video Streams 與 WebRTC 技術整合 AWS CloudTrail，該服務提供使用者、角色或服務中使用 WebRTC 技術的使用者、角色或 AWS 服務所採取的動作記錄。 CloudTrail 使用 WebRTC 技術擷取 Amazon Kinesis Video Streams 的所有 API 呼叫做為事件。擷取的呼叫包括來自 Amazon Kinesis Video Streams 主控台的呼叫，以及程式碼對 Amazon Kinesis Video Streams with WebRTC API 操作的呼叫。如果您建立追蹤，您可以啟用持續向 Amazon S3 儲存貯體傳遞 CloudTrail 事件，包括使用 WebRTC 技術的 Amazon Kinesis Video Streams 的事件。如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷使用 WebRTC 對 Amazon Kinesis Video Streams 提出的請求、提出請求的 IP 位址、提出請求的人員、提出請求的時間以及其他詳細資訊。

若要進一步了解 CloudTrail，包括如何設定和啟用它，請參閱AWS CloudTrail 使用者指南。

Amazon Kinesis Video Streams 與 WebRTC 技術和 CloudTrail

CloudTrail 在您創建 AWS 帳戶時，您的帳戶已啟用。當使用 WebRTC 技術的 Amazon Kinesis Video Streams 中發生受支援的事件活動時，該活動會與事件歷史記錄中的其他 AWS 服務 CloudTrail 事件一起記錄在事件中。您可以在帳戶中查看，搜索和下載最近的事 AWS 件。如需詳細資訊，請參閱檢視具有事 CloudTrail 件記錄的事件。

如需 AWS 帳戶中持續記錄事件 (包括使用 WebRTC 技術的 Amazon Kinesis Video Streams 的事件)，請建立追蹤。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。追蹤記錄來自 AWS 分區中所有區域的事件，並將日誌檔傳送到您指定的 Amazon S3 儲存貯體。此外，您還可以設定其他 AWS 服務，以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀

- CloudTrail 支援的服務與整合
- 設定 Amazon SNS 通知 CloudTrail
- 從多個區域接收 CloudTrail 記錄檔並從多個帳戶接收 CloudTrail 記錄檔

使用 WebRTC 技術的 Amazon Kinesis Video Streams 支援將下列動作記錄為記錄檔中 CloudTrail的事件：

- CreateSignalingChannel
- DeleteSignalingChannel
- DescribeSignalingChannel
- GetSignalingChannelEndpoint
- ListSignalingChannels
- ListTagsForResource
- TagResource
- UntagResource
- UpdateSignalingChannel

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 要求是使用根使用者登入資料還是 AWS Identity and Access Management (IAM) 使用者登入資料提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱CloudTrail 使 userIdentity 元素。

範例：Amazon Kinesis Video Streams with WebRTC 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔的形式傳遞到您指定的 Amazon S3 儲存貯體。 CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，包括有關請求的操作，動作的日期和時間，請求參數等信息。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

下列範例顯示示範CreateSignalingChannel動作的 CloudTrail 記錄項目。

```
{
   "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"
}
```

使用 WebRTC 進行 Amazon Kinesis Video Streams 疑難排解

使用下列資訊對使用 WebRTC 技術的 Amazon Kinesis Video Streams 可能遇到的常見問題進行疑難排解。

主題

Service Quotas

您只能將一個主要和一個或多個檢視器連線到單一訊號通道。

自 2023 年 2 月起，無法將多個主機連接到單個信令通道。如需服務配額的其他資訊，請參閱配額。

網路要求

使用 WebRTC 技術的 Kinesis 視訊串流的訊號通道服務端點的一般網路需求如下：

- 對於託管於的端點進行 HTTPS 呼叫 `https://*.kinesisvideo.{region}.amazonaws.com`
- WebSocket 與端點整合 `wss://*.kinesisvideo.{region}.amazonaws.com`
- STUN伺服器位於 `stun:stun.kinesisvideo.{aws-region}.amazonaws.com:443`
- TURN伺服器位於`turn:_._.kinesisvideo.{aws-region}.amazonaws.com:443`和 `turns:_._.kinesisvideo.{aws-region}.amazonaws.com:443`

作為 RTC 一部分的對等之間使用的協議PeerConnection 可以是基於 TCP 或 UDP。

大多數應用程式會嘗試透過決定每個對等端點的 IP 位址，以及要作為 ICE 候選項交換的 peer-to-peer 連接埠和通訊協定來建立直接連線。這些候選人用於嘗試使用這些候選人彼此連接。他們將嘗試每一對，直到可以建立連接。

網路環境

如果來自檢視器的訊息已傳送至主要伺服器，並且記錄了記錄檔，則找不到有效的連線路由。No valid ICE candidate如果有防火牆阻止直接連接，或者無法連接網絡，則可能會發生這種情況。

請執行下列動作以疑難排解連線問題：

- 如果您沒有TURN在主端使用，請務必啟用TURN。

  TURN默認情況下，在 C SDK 中啟用。在 JavaScript SDK 中，選取STUN/TURN或TURN only在 NAT 周遊中。
- 對於受限制的網路，例如企業網路，請嘗試其他可用的網路 (有線或無線)。

  如果您要連線到 VPN，請中斷連線。

> **Note**
>
> 您可以忽略 Kinesis Video Streams TURN 伺服器傳回的 403 個Forbidden IP錯誤。伺服器會拒絕包含 localhost IP 的 ICE 候選配對，例如192.168.*或10.0.0.*。
> 這可能會導致某些（但不是全部）ICE 候選對失敗。

建立工作階段的問題

WebRTC 技術可以幫助緩解由於以下原因而發生的問題：

- 网络地址转换
- 防火牆
- 對等之間的代理

WebRTC 技術提供了一個框架，以幫助協商和維護連接，只要對等方連接。它還提供了一種機制，用於在無法協商 peer-to-peer 連接的情況下通過TURN服務器轉送媒體。

鑑於建立連接所需的所有組件，因此值得了解一些可用於幫助解決與建立會話相關的問題的工具。

主題

- 會話描述協議（SDP）提供和答案

- 評估 ICE 候選人產生
- 確定哪些候選人被用來建立連接
- 與冰相關的超時

會話描述協議（SDP）提供和答案

會話描述協議（SDP）提供和答案初始化對等之間的 RTC 會話。

要了解有關 SDP 協議的更多信息，請參閱規範。

- 選件是由「觀眾」所產生，這些「觀眾」想要連線至訊號通道的對等，而這些對等會以 WebRTC 的 Kinesis Video Streams 中的「主」身分連線。
- 答案由報價的接收者產生。

提供和答案都是在客戶端生成的，儘管它們可能包含已收集到此時的 ICE 候選項。

適用於 C 的 Kinesis Video Streams WebRTC SDK 包含一個簡單的環境變數，您可以將其設定為記錄 SDP。這對於了解收到的選件和正在生成的答案非常有用。

若要`stdout`從 SDK 記錄 SDP，請設定下列環境變數：`export DEBUG_LOG_SDP=TRUE`。您還可以記錄 SDP 提供和使用`sdpOffer`事件 JavaScript基於客戶端的答案。若要查看此示範，請參閱GitHub。

如需其他資訊，請參閱 the section called “使用 WebRTC 指標監控 Kinesis Video Streams with WebRTC 指標CloudWatch”。

如果未傳回 SDP 答案，則對等可能無法接受 SDP 選件，因為選件不包含任何相容的媒體轉碼器。您可能會看到類似下列內容的記錄檔：

```
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'..
```

當您檢閱 SDP 選件內容時，請尋找`a=rtpmap`以開頭的行，以查看要求的媒體轉碼器。

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

評估 ICE 候選人產生

ICE 候選項由每個對STUN服務器進行調用的客戶端生成。對於使用 WebRTC 技術的 Kinesis Video Streams，伺服器STUN為。`stun:stun.kinesisvideo.{aws-region}.amazonaws.com:443`

除了調用STUN服務器以獲取候選人之外，客戶端還通常調用服務TURN器。他們進行此調用，以便在無法建立直接 peer-to-peer 連接的情況下，轉送服務器可以用作備用。

您可以使用下列工具來產生 ICE 候選項：

- 涓流冰 WebRTC 樣本，它使用涓涓細流冰收集候選人
- IceTest. 資訊。

使用這兩種工具。您可以輸入STUN和TURN服務器信息以收集候選人。

若要使用 WebRTC 取得 Kinesis 視訊串流的TURN伺服器資訊和必要的認證，您可以呼叫 API 作業。GetIceServerConfig

下列 AWS CLI 呼叫示範如何取得此資訊，以便在這兩個工具中使用。

```
export CHANNEL_ARN="YOUR_CHANNEL_ARN"

aws kinesisvideo get-signaling-channel-endpoint \
```

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

get-signaling-channel-endpoint命令的輸出會傳回如下所示的回應：

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

使用 HTTPS ResourceEndpoint 值來取得TURN伺服器清單，如下所示：

```
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
```

該響應包含TURN服務器詳細信息，包括 TCP 和 UDP 的端點以及訪問它們所需的憑據。

> **Note**
>
> 回應中的 TTL 值會決定這些認證的有效期間 (以秒為單位)。在涓流 ICE WebRTC 樣本或 IceTest.Info 中使用這些值，以使用 Kinesis 視訊串流受管理服務端點產生 ICE 候選項目。

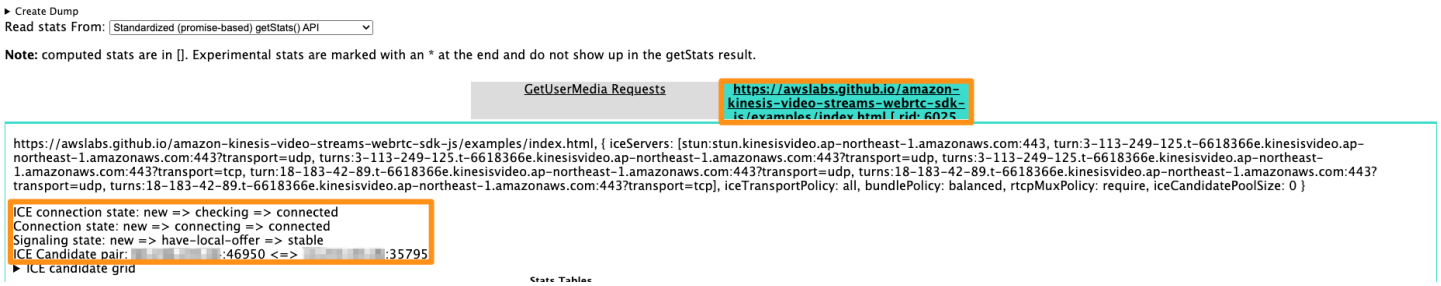
確定哪些候選人被用來建立連接

瞭解哪些候選人被用來成功建立工作階段會很有幫助。如果您有運行已建立會話的基於瀏覽器的客戶端，則可以使用內置的 webrtc 內部實用程序在 Google Chrome 中確定此信息。

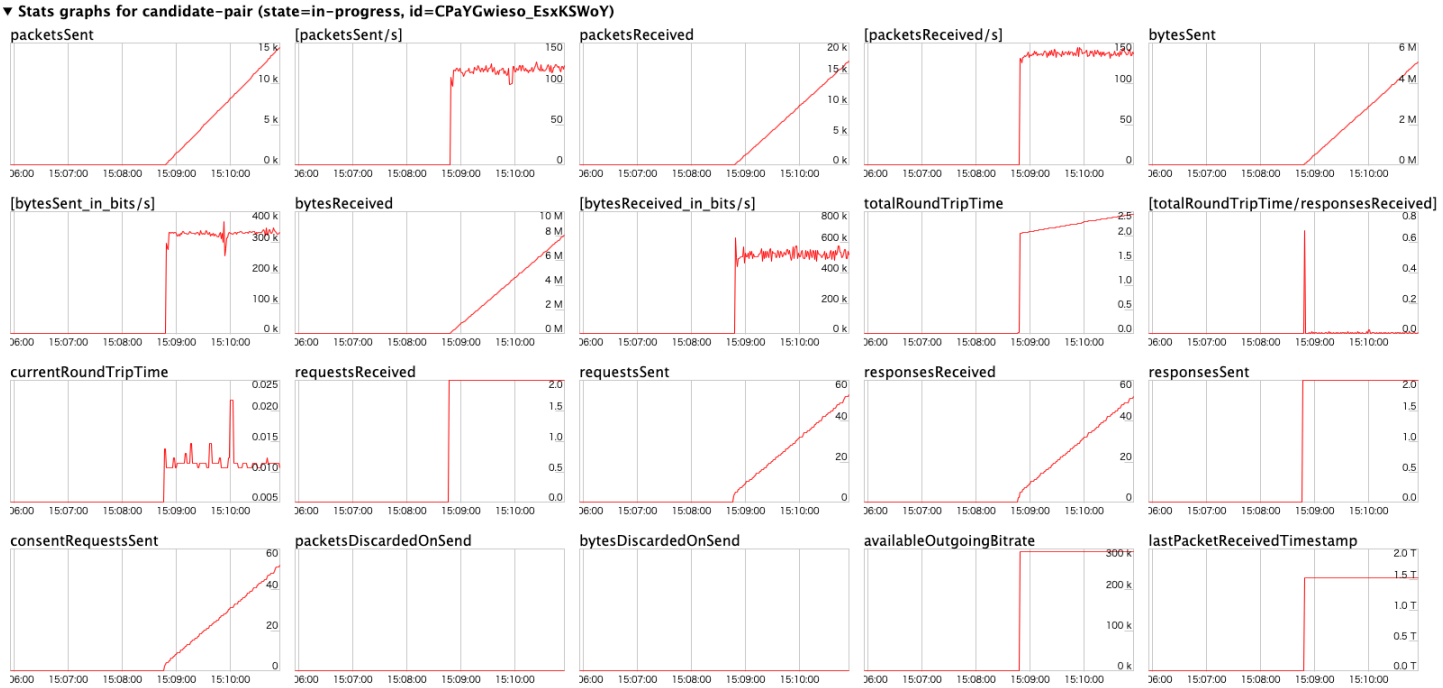
在一個瀏覽器選項卡中打開 WebRTC 技術會話。

在另一個選項卡中，打開chrome://webrtc-internals/。您可以在此選項卡中查看有關正在進行的會話的所有信息。

您將看到有關已建立連接的信息。例如：



您還可以確認已建立連接的指標，如下所示。



與冰相關的超時

預設逾時值會針對中的 ICE 設定KvsRtcConfiguration。對於大多數使用者而言，預設值應該已足夠，但是您可能需要對其進行調整，以提高透過不良網路建立連線的機會。您可以在應用程式中設定這些預設值。

檢閱預設設定的記錄檔：

```
2024-01-08 19:43:44.433 INFO    iceAgentValidateKvsRtcConfig():
```

```
iceLocalCandidateGatheringTimeout: 10000 ms
iceConnectionCheckTimeout: 12000 ms
iceCandidateNominationTimeout: 12000 ms
iceConnectionCheckPollingInterval: 50 ms
```

如果您的網路品質不佳且想要改善連線的機會，請嘗試調整下列值：

- `iceLocalCandidateGatheringTimeout`-增加此逾時限制，以收集其他可能的候選人以嘗試連線。目標是嘗試所有可能的候選對，因此，如果您的網絡不佳，請增加此限制以提供更多時間聚集。

  例如，如果主機候選人不起作用，並且需要嘗試伺服器反射 (srflx) 或轉送候選項，則可能需要增加此逾時時間。由於網絡不佳，候選人聚集緩慢，應用程序不希望在此步驟上花費超過 20 秒。增加逾時可提供更多時間來收集潛在的候選人嘗試連線。

  > **Note**
  >
  > 我們建議這個值小於`iceCandidateNominationTimeout`, 因為提名步驟需要有時間與新的候選人工作.

- `iceConnectionCheckTimeout`-在不穩定或緩慢的網路中增加此逾時時間，其中封包交換和繫結要求/回應需要一段時間。增加此超時時間允許至少一個候選對被嘗試由另一個同行提名。
- `iceCandidateNominationTimeout`-增加此逾時時間，以確保嘗試與本機轉送候選人的候選配對。

  例如，如果收集第一個本機轉送候選項需要大約 15 秒的時間，請將逾時設定為超過 15 秒的值，以確保嘗試與本機轉送候選者的候選配對成功。如果該值設置為小於 15 秒，SDK 將失去嘗試潛在的候選對，導致連接建立失敗。

  > **Note**
  >
  > 我們建議此值大於`iceLocalCandidateGatheringTimeout`，以使其產生效果。

- `iceConnectionCheckPollingInterval`-此值預設為每個規格 50 毫秒。變更此值會變更連線檢查的頻率，以及 ICE 狀態機器轉換的頻率。

  在具有良好系統資源的可靠高性能網絡設置中，您可以降低價值以幫助更快地建立連接。增加值可能有助於減少網路負載，但建立連線可能會變慢。

> **Important**
>
> 我們不建議變更此預設值。

偵錯進行中的連

有跡象表明，可能會導致問題與已建立的，持續的 WebRTC 技術連接，但網絡是最常見的。

您可以透過設定`export AWS_KVS_LOG_LEVEL=1`為環境變數，從 SDK 確認詳細資訊層級記錄。

> **Note**
>
> 如果在指定的逾時內找不到候選配對，ICE 代理程式會傳回錯誤狀態`0x5a00000d`。

如需記錄層級的其他資訊，請參閱GitHub。

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

您將看到類似以下的日誌。從這些記錄檔中，您可以確認互動式連線建立 (ICE) 候選項目 (IP 位址和連接埠) 和選取的候選對。

```
2023-02-13 05:57:16 DEBUG   iceAgentReadyStateSetup(): Selected pair w1UdV9fE+_/
CuBel1pl, 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
```

Amazon Kinesis Video Streams with WebRTC 開發人員指南的文件歷史記錄

變更	描述	日期
初始版本	Amazon Kinesis Video Streams with WebRTC 開發人員指南初版。進一步了解	2019 年 11 月 4 日

AWS 詞彙表

如需最新的 AWS 術語，請參閱《AWS 詞彙表 參考》中的 AWS 詞彙表。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。