



開發人員指南

# Amazon Kinesis Video Streams



# Amazon Kinesis Video Streams: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

什麼是 Kinesis Video Streams ? .....	1
區域可用性 .....	2
您是 Kinesis Video Streams 的首次使用者嗎? .....	3
系統要求 .....	4
相機要求 .....	4
已測試作業系統 .....	5
SDK 儲存空間需求 .....	5
運作方式 .....	6
API 和製作者資料庫 .....	7
Kinesis Video Streams Streams Streams .....	7
製作人程式 .....	9
影片播放 .....	9
播放要求 .....	10
使用 HLS 進行視頻播放 .....	12
使用 MPEG-DASH 播放影片 .....	21
使用串流中繼資料 .....	25
將中繼資料新增至 Kinesis 視訊串流 .....	26
使用內嵌在 Kinesis 視訊串流中的中繼資料 .....	27
串流元數據限制 .....	29
資料模型 .....	29
流頭元素 .....	29
串流軌跡資料 .....	34
框架標題元素 .....	35
MKV 幀數據 .....	35
開始使用 .....	36
設定 帳戶 .....	36
註冊一個 AWS 帳戶 .....	37
建立管理使用者 .....	37
建立 AWS 帳戶 金鑰 .....	38
建立 Kinesis 影片串流 .....	38
使用主控台建立視訊串流 .....	38
使用建立視訊串流 AWS CLI .....	39
將資料傳送至 Amazon Kinesis 影片串流 .....	39
建置 SDK 和範例 .....	40

執行範例以將媒體上傳至 Kinesis Video Streams .....	42
審核確認物件 .....	44
使用媒體資料 .....	44
在主控台中檢視媒體 .....	44
使用 HLS 消耗媒體數據 .....	44
邊緣代理 .....	45
Amazon Kinesis Video Streams 邊緣代理程式 API 作業 .....	46
監控 Amazon Kinesis Video Streams 邊緣代理 .....	46
以非AWS IoT Greengrass模式部署 .....	46
1. 安裝相依項目 .....	47
2. 為您的 IP 攝像機 RTSP 網址創建資源 .....	48
3. 建立 IAM 許可政策 .....	50
4. 建立 IAM 角色 .....	52
5. 建立AWS IoT角色別名 .....	53
6. 建立 AWS IoT 政策 .....	54
7. 創建AWS IoT事物並獲取AWS IoT Core憑據 .....	55
8. 建置和執行 Amazon Kinesis Video Streams 邊緣代理程式 .....	58
9. (選擇性) 安裝 CloudWatch 代理程式 .....	67
10. (選擇性) 執行 Amazon Kinesis Video Streams 邊緣代理程式 .....	70
部署至 AWS IoT Greengrass .....	73
1. 建立執行個體 .....	73
2. 設定AWS IoT Greengrass核心裝置 .....	74
3. 為您的 IP 攝像機 RTSP 網址創建資源 .....	75
4. 將權限新增至 TES 角色 .....	77
5. 安裝密碼管理員元件 .....	80
6. 在裝置上部署 Amazon Kinesis Video Streams 邊緣代理程式 .....	82
7. (選擇性) 安裝記AWS IoT Greengrass錄管理員元件 .....	90
常見問答集 .....	93
Amazon Kinesis Video Streams 邊緣代理程式支援哪些作業系統？ .....	94
Amazon Kinesis Video Streams 邊緣代理程式是否支援 H.265 媒體？ .....	94
Amazon Kinesis Video Streams 邊緣代理程式是否在 AL2 中運作？ .....	94
如何在AWS IoT物件或設備中運行多個流？ .....	94
如何在發送StartEdgeConfigurationUpdate後進行編輯？ .....	94
你有ScheduleConfigs什麼常見的例子嗎？ .....	94
是否有最大串流限制？ .....	95
如何重新啟動錯誤的工作？ .....	95

如何監控 Amazon Kinesis Video Streams 邊緣代理程式的運作狀態？ .....	96
透過 VPC 串流影片 .....	97
其他資訊 .....	97
VPC 端點程序 .....	97
映像 .....	100
GetImages 入門 .....	100
開始使用亞馬遜 S3 交付 .....	100
UpdateImageGenerationConfiguration .....	101
DescribeImageGenerationConfiguration .....	103
製作人 MKV 標籤 .....	104
在製作者 SDK 中使用新增中繼資料標 PutEventMetaData .....	104
限制 .....	105
S3 物件中繼資料 .....	105
S3 物件路徑 (影像) .....	105
亞馬遜 S3 URI 建議，以防止節流 .....	106
通知 .....	107
UpdateNotificationConfiguration .....	107
DescribeNotificationConfiguration .....	107
製作人 MKV 標籤 .....	104
製作人 MKV 標籤的語法 .....	104
MKV 標籤限制 .....	108
.....	108
.....	108
Amazon SNS 主題承載 .....	108
檢視您的 Amazon SNS 訊息 .....	109
安全 .....	111
資料保護 .....	111
什麼是 Kinesis Video Streams 的伺服器端加密？ .....	112
成本、區域和效能考量 .....	112
如何開始使用伺服器端加密？ .....	113
建立和使用使用者產生的 KMS 金鑰 .....	113
使用使用者產生的 KMS 金鑰的權限 .....	114
使用 IAM 控制對 Kinesis Video Streams 資源的存取 .....	116
政策語法 .....	116
室壁運動影片串流的動作 .....	117
Kinesis Video Streams 的 Amazon 資源名稱 (ARN) .....	117

授與其他 IAM 帳戶存取 Kinesis 影片串流的存取權 .....	118
範例政策 .....	118
使用以下方式控制 Kinesis Video Streams 資源的存取 AWS IoT .....	120
AWS IoT ThingName 作為流名稱 .....	121
AWS IoT CertificateId 作為流名稱 .....	126
使用 AWS IoT 認證串流至硬式編碼串流名稱 .....	128
監控 .....	129
合規驗證 .....	129
恢復能力 .....	130
基礎設施安全性 .....	130
安全最佳實務 .....	130
實作最低權限存取 .....	131
使用 IAM 角色 .....	131
用 CloudTrail 於監控 API 呼叫 .....	131
Producer 資料庫 .....	132
室壁運動視頻流生產者客戶端 .....	132
室壁運動視頻流生產者庫 .....	133
相關主題 .....	133
Java Producer Library .....	134
程序：使用 Java Producer 開發套件 .....	134
步驟 1：下載和設定程式碼 .....	135
步驟 2：編寫並檢查代碼 .....	136
步驟 3：運行並驗證代碼 .....	138
Android Producer Library .....	139
程序：使用 Android Producer 開發套件 .....	139
必要條件 .....	139
步驟 1：下載和設定程式碼 .....	143
步驟 2：檢查代碼 .....	144
步驟 3：運行並驗證代碼 .....	146
C++ Producer Library .....	147
物件模型 .....	147
將媒體放入串流 .....	148
回呼介面 .....	148
程序：使用 C++ Producer 開發套件 .....	148
步驟 1：下載和設定程式碼 .....	151
步驟 2：撰寫並檢查程式碼 .....	151

步驟 3：執行程式碼並驗證程式碼 .....	158
使用 C++ 生產者開發套件作為 GStreage 插件使用 .....	158
使用 C++ 生產者 SDK 作為碼頭容器中的 GStreamer 插件 .....	158
使用記錄 .....	158
C Producer Library .....	159
物件模型 .....	159
將媒體放入串流 .....	160
程序：使用 C Producer 開發套件 .....	160
步驟 1：下載程式碼 .....	163
步驟 2：撰寫程式碼 .....	164
步驟 3：執行程式碼並驗證程式碼 .....	166
在樹莓派 C++ 生產者 SDK .....	168
必要條件 .....	168
建立具有寫入 Kinesis Video Streams 之權限的 IAM 使用者 .....	169
加入您的樹莓派到您的 Wi-Fi 網絡 .....	170
遠端 Connect 到您的樹莓派 .....	171
配置樹莓派相機 .....	172
安裝必要軟體 .....	172
下載並建置 Kinesis Video Streams C++ 製作者 SDK .....	173
將影片串流至 Kinesis 視訊串流並檢視即時串流 .....	174
參考資料 .....	175
製作者 SDK 限制 .....	175
錯誤代碼參考 .....	178
NAL 適應性旗標 .....	220
開發套件結構 .....	221
串流結構 .....	223
回呼 .....	236
串流剖析器程式庫 .....	243
程序：使用 Kinesis 視訊串流剖析程式庫 .....	243
先決條件 .....	243
步驟 1：下載並組態程式碼 .....	243
下一步驟 .....	244
步驟 2：撰寫並檢查程式碼 .....	244
StreamingMkvReader .....	244
FragmentMetadataVisitor .....	245
OutputSegmentMerger .....	246

KinesisVideoExample .....	248
下一步驟 .....	251
步驟 3：執行並驗證程式碼 .....	251
範例 .....	252
範例：將資料傳送至 Kinesis Video Streams .....	252
範例：從 Kinesis Video Streams 擷取資料 .....	252
範例：播放視訊資料 .....	252
必要條件 .....	253
GStreamer .....	253
下載，構建和配置元素 .....	254
運行光元素 .....	254
啟動命令 .....	255
在碼頭容器中運行 GStreamer 元素 .....	257
參數參考 .....	260
PutMedia API .....	272
步驟 1：下載並設定程式碼 .....	272
步驟 2：編寫並檢查代碼 .....	274
步驟 3：運行並驗證代碼 .....	275
RTSP 和 Docker .....	276
必要條件 .....	277
建立泊塢視窗映像 .....	277
執行 RTSP 範例應用程式 .....	278
轉譯器 .....	279
必要條件 .....	279
執行轉譯器範例 .....	280
運作方式 .....	281
SageMaker .....	282
必要條件 .....	283
建立應用程式 .....	284
監控應用程式 .....	286
擴充應用程式 .....	287
清理應用程式 .....	288
監控 .....	290
監控指標 CloudWatch .....	290
CloudWatch 指標指引 .....	302
使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch .....	305



CloudWatch Amazon Kinesis Video Streams 邊緣代理程式的指標指引 .....	308
使用 記錄 CloudTrail API 呼叫 .....	310
Amazon Kinesis Video Streams 和 CloudTrail .....	310
範例：Amazon Kinesis Video Streams 日誌檔項目 .....	311
配額 .....	315
控制平面 API 服務配額 .....	315
媒體和封存媒體 API 服務配額 .....	320
片段中繼資料和片段媒體配額 .....	323
片段元數據配額 .....	325
串流標籤 .....	326
故障診斷 .....	327
疑難排解一般問題 .....	327
延遲過高 .....	327
疑難排解 API 問題 .....	328
錯誤：「未知的選項」 .....	328
錯誤：「無法判斷要授權的服務/操作名稱」 .....	328
錯誤：「無法將影格放入串流」 .....	328
錯誤：「在收到最終 AckEvent 服務之前已關閉連接」 .....	329
錯誤：「STATUS_STORE_OUT_OF_MEMORY」 .....	329
疑難排解 HLS 問題 .....	329
Java 問題疑難排解 .....	329
啟用 Java 日誌 .....	330
生產者程式庫問題疑 .....	330
無法編譯製作者開發套件 .....	331
影片串流不會顯示在主控台 .....	331
錯誤：使用 GStreamer 示範應用程式串流資料時「包含在請求中的安全字符無效」 .....	332
錯誤：「無法將影格提交到 Kinesis Video 用戶端」 .....	332
GStreamer 應用程式停止並在 OS X 上出現「串流停止，原因未交涉」訊息 .....	332
在 Raspberry Pi 上的 GStreamer 示範中建立 Kinesis 影片用戶端時出現錯誤：「無法配置堆積」 .....	333
在 Raspberry Pi 上執行 GStreamer 示範時出現錯誤：「非法指示」 .....	333
攝影機無法載入 Raspberry Pi .....	333
在 macOS High Sierra 上找不到攝影機 .....	334
在 macOS High Sierra 上編譯時找不到 jni.h 檔案 .....	334
執行 GStreamer 示範應用程式時 Curl 錯誤 .....	334
在 Raspberry Pi 執行時間的時間戳記/範圍聲明 .....	334

在 Raspberry Pi 上的 <code>gst_value_set_fraction_range_full</code> 聲明 .....	335
在 Android 上發生 <code>STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA</code> (0x3200000d) 錯誤 .....	335
到達片段持續時間上限錯誤 .....	335
使用 IoT 授權時的「傳遞的物件名稱無效」錯誤 .....	336
疑難排解串流剖析程式庫 .....	336
無法從串流存取單一影格 .....	336
片段解碼錯誤 .....	336
文件歷史記錄 .....	338
API 參考 .....	342
動作 .....	342
Amazon Kinesis Video Streams .....	343
Amazon Kinesis Video Streams 媒體 .....	463
Amazon Kinesis Video Streams 歸檔媒體 .....	478
Amazon Kinesis Video Streams .....	520
Amazon Kinesis Video Streams .....	529
資料類型 .....	533
Amazon Kinesis Video Streams .....	534
Amazon Kinesis Video Streams 媒體 .....	574
Amazon Kinesis Video Streams 歸檔媒體 .....	577
Amazon Kinesis Video Streams .....	595
Amazon Kinesis Video Streams .....	597
常見錯誤 .....	597
常見參數 .....	599
.....	dcii

# 什麼是 Kinesis Video Streams ？

您可以使用 Amazon Kinesis Video Streams (全受管AWS 服務) 將即時影片從裝置串流到裝置AWS 雲端，或建置應用程式以進行即時影片處理或批次導向視訊分析。

Kinesis Video Streams 不僅是影片資料的儲存空間。您可以使用它來即時觀賞雲端接收到的影片串流。您可以在中監控即時串流AWS Management Console，也可以開發自己的監控應用程式，使用 Kinesis Video Streams API 程式庫來顯示即時影片。

您可以使用 Kinesis Video Streams 從數百萬個來源擷取大量即時視訊資料，包括智慧型手機、安全攝影機、網路攝影機、內嵌在汽車中的攝影機、無人機和其他來源。您也可以傳送非視訊、時間序列化資料，例如音訊資料、熱影像、深度資料和 RADAR 資料。將這些來源的即時視訊串流到 Kinesis 視訊串流時，您可以建置應用程式以即時存取資料 frame-by-frame，以進行低延遲處理。Kinesis Video Streams 與來源無關。您可以使用媒體[GStreamer](#)櫃從電腦的網路攝影機串流視訊，或使用即時串流通訊協定 (RTSP) 從網路上的攝影機串流視訊。

您也可以將 Kinesis 視訊串流設定為在指定的保留期間內持久儲存媒體資料。Kinesis Video Streams 會自動儲存此資料，並在靜態時加密資料。此外，Kinesis Video Streams 時間索引會根據生產者時間戳記和擷取時間戳記來儲存資料。您可以建置定期批次處理視訊資料的應用程式，也可以建立需要針對不同使用案例一次性存取歷史資料的應用程式。

即時或批次導向的自訂應用程式都可以在 Amazon EC2 執行個體上執行。這些應用程式可能會使用開放原始碼、深度學習演算法處理資料，或使用與 Kinesis Video Streams 整合的第三方應用程式。

使用 Kinesis Video Streams 的好處包括：

- 從數百萬個裝置進行連線和串流 — 您可以使用 Kinesis Video Streams 來 Connect 和串流數百萬種裝置 (包括消費性智慧型手機、無人機和儀表板攝影機) 的視訊、音訊和其他資料。您可以使用 Kinesis Video Streams 製作者程式庫來設定您的裝置，並可靠地即時串流，或作為 after-the-fact 媒體上傳作業。
- 持久存放、加密和索引資料 — 您可以將 Kinesis 視訊串流設定為在自訂保留期間持久儲存媒體資料。Kinesis Video Streams 也會根據生產者產生或服務端時間戳記，針對儲存的資料產生索引。您的應用程式可以使用時間索引擷取串流中的指定資料。
- 專注於管理應用程式而非基礎架構 — Kinesis Video Streams 是無伺服器的，因此無需設定或管理基礎架構。您不必擔心基礎架構的部署、設定或彈性擴展，因為您的資料串流和使用應用程式數量不斷增加和縮減。Kinesis Video Streams 會自動執行管理串流所需的所有管理和維護工作，因此您可以專注於應用程式，而不是基礎結構。

- 在資料串流上建置即時和批次應用程式 — 您可以使用 Kinesis Video Streams 建置可在即時資料串流上運作的自訂即時應用程式，並建立批次或一次性應用程式，以持久保存的資料執行，而不需要嚴格的延遲需求。您可以使用來處理和分析串流來建置、部署和管理自訂應用程式：開放原始碼 (Apache MXNet、OpenCV)、本土開發或協力廠商解決方案。AWS Marketplace 您可以使用 Kinesis Video Streams Get API 建置多個並行應用程式，以即時或批次導向的方式處理資料。
- 更安全地串流資料 — Kinesis Video Streams 會在資料流經服務時以及保留資料時加密所有資料。Kinesis Video Streams 對來自裝置的資料串流強制執行以傳輸層安全性 (TLS) 為基礎的加密，並使用 AWS Key Management Service () 加密所有靜態資料。AWS KMS 此外，您還可以使用 AWS Identity and Access Management (IAM) 管理對資料的存取。
- 隨用隨付-如需詳細資訊，請參閱 [AWS Pricing Calculator](#)。

## 區域可用性

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

區域名稱	AWS區域代碼
中國 (北京)	cn-north-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

## 您是 Kinesis Video Streams 的首次使用者嗎？

如果您是 Kinesis Video Streams 的首次使用者，我們建議您依序閱讀下列各節：

1. [Kinesis Video Streams：它是如何工作的](#)— 瞭解 Kinesis Video Streams 概念。
2. [開始使用 Amazon Kinesis Video Streams](#)— 要設置您的帳戶並測試 Kinesis Video Streams。
3. [室壁運動視頻流生產者庫](#)— 瞭解如何建立 Kinesis Video Streams 製作者應用程式。
4. [Kinesis 視頻流解析器庫](#)— 瞭解如何在 Kinesis 影片串流消費者應用程式中處理傳入資料框架。
5. [Amazon Kinesis Video Streams 範例](#)— 查看更多有關您可以使用 Kinesis Video Streams 做什麼的示例。

# Kinesis Video Streams 系統需求

以下各節包含 Amazon Kinesis Video Streams 的硬體、軟體和儲存需求。

主題

- [相機要求](#)
- [已測試作業系統](#)
- [SDK 儲存空間需求](#)

## 相機要求

用於執行 Kinesis 視訊串流製作者 SDK 和範例的攝影機具有下列記憶體需求：

- 軟體開發套件內容檢視需要 16 MB 的記憶體。
- 範例應用程式預設組態為 128 MiB 的記憶體。這個值對於有良好網路連線的生產者而言是適合的，而且不需額外的緩衝。如果網路連線不良而且需要更多的緩衝，您可以將每秒影格率乘上影格記憶體大小，以計算每秒緩衝的記憶體需求。如需配置記憶體的詳細資訊，請參閱 [StorageInfo](#)。

我們建議使用 USB 或 RTSP (即時串流協定) 並以 H.264 編碼資料的攝影機，因為這可以免除 CPU 的編碼工作負載。

目前，示範應用程式不支援 RTSP 串流的使用者資料包通訊協定 (UDP)。未來將新增此功能。

製作者開發套件支援以下類型的攝影機：

- 網路攝影機。
- USB 攝影機。
- 以 H.264 編碼的攝影機 (較佳)。
- 無 H.264 編碼的攝影機。
- Raspberry Pi 攝影機模組。這是偏好的 Raspberry Pi 裝置，因為它連接到 GPU 以傳輸影片資料，因此沒有 CPU 處理成本。
- RTSP (網路) 攝影機。這些攝影機是較佳的選擇，因為視訊串流已經以 H.264 進行編碼。

## 已測試作業系統

我們已使用下列裝置和作業系統測試 web 攝影機和 RTSP 攝影機：

- Mac mini
  - High Sierra
- MacBook 專業筆記本
  - Sierra (10.12)
  - El Capitan (10.11)
- 執行 Ubuntu 16.04 的 HP 筆記型電腦
- Ubuntu 17.10 (Docker 容器)
- Raspberry Pi 3

## SDK 儲存空間需求

安裝 [室壁運動視頻流生產者庫](#) 有 170 MB 的最低儲存要求，以及 512 MB 的建議儲存要求。

# Kinesis Video Streams：它是如何工作的

## 主題

- [Kinesis Video Streams Streams API 與生產者程式庫支援](#)
- [Kinesis Video Streams 播放](#)
- [搭配 Kinesis 視訊串流使用串流中繼資料](#)
- [Kinesis Video Streams 資料模型](#)

您可以使用完全受管的 Amazon Kinesis Video Streams AWS 服務，將即時影片從裝置串流到 AWS 雲端並持久存放。然後，您可以建置自己的應用程式以供即時影片處理使用，或執行批次導向影片分析。

下圖提供 Kinesis Video Streams Video Streams 的概觀 Kinesis Video Streams 的概觀

下圖示範以下元件間的互動：

- 製作人 — 將資料放入 Kinesis 視訊串流的任何來源。生產者可以是任何產生影片的裝置 (例如保全攝影機、穿戴式攝影機、智慧型手機攝影機或儀表板攝影機)。生產者也可以傳送非影片資料，例如音訊摘要、影像或 RADAR 資料等。

單一生產者可以產生一或多個影片串流。例如，攝影機可以將視訊資料推送到一個 Kinesis 視訊串流，而將音訊資料推送到另一個。

- Kinesis Video Streams 製作者程式庫 — 您可以在裝置上安裝和設定的一組軟體和程式庫。您可以使用這些庫以不同的方式安全地連接和可靠地流式傳輸視頻，包括實時，緩衝幾秒鐘後或作為 after-the-fact 媒體上傳。
- Kinesis 視訊串流 — 一種資源，可用來傳輸即時影片資料、選擇性地儲存，以及即時、批次或一次性使用資料供使用。在典型組態中，Kinesis 視訊串流只有一個製作者將資料發佈到其中。

串流可以攜帶音訊、影片和類似的時間編碼資料串流 (例如深度感應摘要、RADAR 摘要等等)。您可以使用 AWS Management Console 或以程式設計方式使用 AWS SDK 建立 Kinesis 視訊串流。

多個獨立的應用程式可以 parallel 使用 Kinesis 視訊串流。

- 消費者 — 從 Kinesis 視訊串流取得片段和畫面等資料，以便檢視、處理或分析。一般而言，這些消費者稱為 Kinesis Video Streams 應用程式。您可以在 Kinesis Video Streams 中即時撰寫使用和處



理資料的應用程式，或者在不需要低延遲處理的情況下，在資料儲存並建立時間編製索引之後。您可以建立這些消費者應用程式，以在 Amazon EC2 執行個體上執行。

- [Kinesis 視頻流解析器庫](#)— 讓 Kinesis Video Streams 應用程式能夠以低延遲的方式可靠地從 Kinesis 視訊串流取得媒體。此外，它會剖析媒體中的影格邊界，讓應用程式可以專注於處理和分析影格本身。

## Kinesis Video Streams Streams API 與生產者程式庫支援

Kinesis Video Streams 提供 API，讓您建立和管理串流，以及從串流讀取或寫入媒體資料。Kinesis Video Streams 主控台除了管理功能外，也支援即時和video-on-demand播放。Kinesis Video Streams 也提供一組製作者程式庫，您可以在應用程式程式碼中使用這些程式庫，從媒體來源擷取資料並上傳到 Kinesis 影片串流。

### 主題

- [Kinesis Video Streams Streams Streams](#)
- [製作人程式](#)

## Kinesis Video Streams Streams Streams

Kinesis Video Streams 提供用於建立和管理 Kinesis Video Streams 的 API。它還提供 API，可讓您將媒體資料讀取和寫入到串流，如下所示：

- 製作者 API — Kinesis Video Streams 提供 PutMedia API，可將媒體資料寫入 Kinesis 視訊串流。在 PutMedia 要求中，生產者會傳送媒體片段的串流。片段是可獨自運作的一系列影格。屬於一片段的影格應與其他片段中的任何影格不存在相依性。如需詳細資訊，請參閱[PutMedia](#)。

片段到達時，Kinesis Video Streams 會依遞增順序指派唯一的片段編號。它也會將每個片段的製作者端和伺服器端時間戳記儲存為 Kinesis Video Stream 特定的中繼資料。

- 消費者 API — 消費者可以使用下列 API 從串流取得資料：
  - GetMedia- 當您使用此 API 時，消費者必須識別開始片段。API 會以片段新增至串流的順序將片段傳回 (依片段號碼，以遞增的順序)。會將片段中的媒體資料打包為結構化格式 (例如 [Matroska \(MKV\)](#))。如需詳細資訊，請參閱[GetMedia](#)。

**Note**

GetMedia 知道片段所在 (封存在資料存放區或可即時提供)。例如，如果 GetMedia 判定開始片段已封存，它會開始從資料存放區傳回片段。當它必須傳回尚未封存的較新片段時，請 GetMedia 切換至從記憶體內串流緩衝區讀取片段。

這是一種持續的消費者範例，此會以串流導入片段的順序來處理片段。

GetMedia 可讓影片處理應用程式失敗或落後，然後追上進度，而無需執行其他操作。使用 GetMedia，應用程式可以處理已封存在資料存放區的資料，並在應用程式追趕時，GetMedia 會持續在應用程式抵達時即時饋送媒體資料。

- GetMediaFromFragmentList(和 ListFragments) - 批次處理應用程式可視為離線消費者。結合 ListFragments 和 GetMediaFromFragmentList API，離線消費者可選擇明確擷取特定媒體片段或範圍的視訊。ListFragments 及 GetMediaFromFragmentList 可讓應用程式識別特定時間範圍內或片段範圍內的影片片段，然後依序或以平行處理方式擷取這些片段。這種方法是適用於 MapReduce 應用程式套件，也就是在必須快速地平行處理大量資料時。

例如，假設一個消費者想要處理一天量的影片片段。消費者可執行以下操作：

1. 呼叫 ListFragments API 並指定時間範圍來取得片段的清單，以選取所需的片段集合。

API 會傳回在特定時間範圍內的所有片段的中繼資料。中繼資料提供片段編號、生產者端和伺服器端時間戳記等資訊。

2. 使用片段中繼資料清單並以任何順序擷取片段。例如，若要處理當天的所有片段，消費者可能會選擇將清單拆分為子清單，並讓 Worker (例如，多個 Amazon EC2 執行個體) 使用 parallel 擷取片段 GetMediaFromFragmentList，並 parallel 處理。

下圖顯示在這些 API 呼叫期間片段和區塊的資料流。

當生產者傳送 PutMedia 要求時，就會在承載中傳送媒體中繼資料，然後傳送一系列的媒體資料片段。接收到資料後，Kinesis Video Streams 會將傳入的媒體資料儲存為 Kinesis Video Streams 區塊。每個區塊都包含下列項目：

- 媒體中繼資料的複本
- 片段

- Kinesis 影片串流特定的中繼資料；例如片段編號以及伺服器端和生產者端時間戳記

當消費者要求媒體中繼資料時，Kinesis Video Streams 會傳回區塊串流，從您在要求中指定的片段編號開始。

如果您啟用串流的資料持續性，則在接收到串流上的片段後，Kinesis Video Streams 也會將片段的副本儲存至資料存放區。

## 製作人程式

建立 Kinesis 視訊串流後，您就可以開始將資料傳送至串流。在應用程式程式碼中，您可以使用這些程式庫從媒體來源擷取資料，並上傳到 Kinesis 影片串流。如需這些可用製作者資料庫的詳細資訊，請參閱 [室壁運動視頻流生產者庫](#)。

## Kinesis Video Streams 播放

您可以使用下列方法檢視 Kinesis 視訊串流：

- GetMedia— 您可以使用 GetMedia API 建置自己的應用程式，以處理 Kinesis Video Streams。GetMedia 是具有低延遲的實時 API。要創建一個使用的播放器 GetMedia，你必須自己構建它。如需如何開發使用顯示 Kinesis 視訊串流之應用程式的相關資訊 GetMedia，請參閱 [串流剖析器程式庫](#)。
- HLS — [HTTP 即時串流 \(HLS\)](#) 是業界標準的 HTTP 式媒體串流通訊協定。您可以使用 HLS 檢視 Kinesis 視訊串流，以進行即時播放或檢視封存的視訊。

您可以使用 HLS 進行即時播放。延遲通常介於 3-5 秒之間，但視使用案例、播放程式和網路條件而定，延遲時間可能在 1—10 秒之間。您可使用第三方播放器 (例如 [Video.js](#) 或 [Google Shaka Player](#)) 以透過提供 HLS 串流工作階段 URL、編寫程式或手動來顯示影片串流。您也可以通過在 [蘋果 Safari](#) 或 [Microsoft 邊緣](#) 瀏覽器的位置欄中輸入 HLS 流會話網址來播放視頻。

- MPEG-DASH — 通過 [HTTP 進行動態自適應流 \(DASH\)](#)，也稱為 MPEG-DASH，是一種自適應比特率流協議，可通過從傳統的 HTTP Web 服務器提供的互聯網上實現高質量的媒體內容流式傳輸。

您可以使用 MPEG-DASH 進行即時播放。延遲通常介於 3-5 秒之間，但視使用案例、播放程式和網路條件而定，延遲時間可能在 1—10 秒之間。您可以使用第三方播放器 (例如 [dash.js](#) 或 [谷歌沙卡播放器](#)) 通過以編程方式或手動提供 MPEG-DASH 流會話 URL 來顯示視頻流。

- GetClip— 您可以使用 GetClip API 在指定時間範圍內從指定視訊串流下載包含已封存的隨選媒體的剪輯片段 (在 MP4 檔案中)。如需詳細資訊，請參閱 [GetClip API 參考](#)。

## 主題

- [視訊播放曲目需求](#)
- [使用 HLS 進行視頻播放](#)
- [使用 MPEG-DASH 播放影片](#)

## 視訊播放曲目需求

Amazon Kinesis Video Streams 支援以多種格式編碼的媒體。如果 Kinesis 視訊串流使用下列四個 API 之一不支援的格式，請使用[GetMedia](#)或 [GetMediaForFragmentList](#)，因為它們沒有軌跡類型限制。

## 主題

- [GetClip 要求](#)
- [獲取破折號StreamingSession網址要求](#)
- [網址要求 StreamingSession](#)
- [GetImages 要求](#)

## GetClip 要求

如需這種 API 的詳細資訊，請參閱 [GetClip](#)。

音軌 1 說明	音軌 1 轉碼器 ID	路線 2 說明	音軌 2 編解碼器 ID
影片	V_MPEG/ISO /AVC	N/A	N/A
影片	V_MPEG/ISO /AVC	AAC 音訊	A_AAC
影片	V_MPEG/ISO /AVC	G.711 音訊 (僅適用於 A-羅)	A_毫升/ACM
H.265 影片	V_MPEGH/ISO / HEVC	N/A	N/A
H.265 影片	V_MPEGH/ISO / HEVC	AAC 音訊	A_AAC

## 獲取破折號StreamingSession網址要求

如需這種 API 的詳細資訊，請參閱 [GetDASHStreamingSessionURL](#)。

音軌 1 說明	音軌 1 轉碼器 ID	路線 2 說明	音軌 2 編解碼器 ID
影片	V_MPEG/ISO /AVC	N/A	N/A
影片	V_MPEG/ISO /AVC	AAC 音訊	A_AAC
影片	V_MPEG/ISO /AVC	G.711 音訊 (僅適用於 A-羅)	A_毫升/ACM
影片	V_MPEG/ISO /AVC	G.711 音訊 (僅限 U-法律)	A_毫升/ACM
AAC 音訊	A_AAC	N/A	N/A
H.265 影片	V_MPEGH/ISO / HEVC	N/A	N/A
H.265 影片	V_MPEGH/ISO / HEVC	AAC 音訊	A_AAC

## 網址要求 StreamingSession

如需這種 API 的詳細資訊，請參閱 [GetHLSStreamingSessionURL](#)。

### HLS MP4

音軌 1 說明	音軌 1 轉碼器 ID	路線 2 說明	音軌 2 編解碼器 ID
影片	V_MPEG/ISO /AVC	N/A	N/A
影片	V_MPEG/ISO /AVC	AAC 音訊	A_AAC
AAC 音訊	A_AAC	N/A	N/A
H.265 影片	V_MPEGH/ISO / HEVC	N/A	N/A

音軌 1 說明	音軌 1 轉碼器 ID	路線 2 說明	音軌 2 編解碼器 ID
H.265 影片	V_MPEGH/ISO / HEVC	AAC 音訊	A_AAC

## 海爾斯 TS

音軌 1 說明	音軌 1 轉碼器 ID	路線 2 說明	音軌 2 編解碼器 ID
影片	V_MPEG/ISO /AVC	N/A	N/A
影片	V_MPEG/ISO /AVC	AAC 音訊	A_AAC
AAC 音訊	A_AAC	N/A	N/A

## GetImages 要求

如需這種 API 的詳細資訊，請參閱 [GetImages](#)。

### Note

GetImages 媒體應包含音軌 1 中的視訊軌道。

## 使用 HLS 進行視頻播放

[HTTP 即時串流 \(HLS\)](#) 是業界標準的 HTTP 式媒體串流通訊協定。您可以使用 HLS 檢視 Kinesis 視訊串流，以進行即時播放或檢視封存的視訊。

您可以使用 HLS 進行即時播放。延遲通常介於 3-5 秒之間，但視使用案例、播放程式和網路條件而定，延遲時間可能在 1—10 秒之間。您可使用第三方播放器 (例如 [Video.js](#) 或 [Google Shaka Player](#)) 以透過提供 HLS 串流工作階段 URL、編寫程式或手動來顯示影片串流。您也可以通過在 [蘋果 Safari](#) 或 [Microsoft 邊緣](#) 瀏覽器的位置欄中輸入 HLS 流會話網址來播放視頻。

若要使用 HLS 檢視 Kinesis 視訊串流，請先使用 [Get StreamingSession](#) Hls URL 建立串流工作階段。這個動作會傳回一個 URL (包含工作階段字符) 以存取 HLS 工作階段。然後您可以使用媒體播放器的 URL 或獨立應用程式來顯示串流。

**⚠ Important**

並非所有傳送至 Kinesis Video Streams 的媒體都能透過 HLS 播放。如需特[the section called "GetHLSStreamingSessionURL"](#)定上傳需求，請參閱。

## 主題

- [使用擷 AWS CLI 取 HLS 串流工作階段 URL](#)
- [範例：在 HTML 和中使用 HLS JavaScript](#)
- [疑難排解 HLS 問題](#)

## 使用擷 AWS CLI 取 HLS 串流工作階段 URL

請遵循以下程序來為 Kinesis 視訊串流產生 HLS 串流工作階段 URL。AWS CLI

如需安裝指示，請參閱使[AWS Command Line Interface 用者指南](#)。安裝後，[配置使 AWS CLI 用憑據和區域](#)。

或者，打開已 AWS CLI 安裝和配置的 AWS CloudShell 終端。如需詳細資訊，請參閱[AWS CloudShell 使用者指南](#)。

為您的 Kinesis 影片串流擷取 HLS URL 端點。

1. 在終端機中鍵入以下內容：

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

您會收到如下所示的回應：

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

2. 向該傳回的端點發出 HLS 串流工作階段 URL 要求。

## Live

對於實時播放，HLS 媒體播放列表在可用時不斷更新為最新媒體。當您在媒體播放程式中播放這種類型的工作階段時，使用者介面通常會顯示「即時」通知，而且沒有用於選擇播放視窗中要顯示的位置的定位器控制項。

執行此命令時，請確定您正在將媒體上傳至此串流。

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

## Live replay

對於即時重播，播放會從指定的開始時間開始。HLS 媒體播放列表也會在可用時不斷更新最新媒體。工作階段將繼續包含新擷取的媒體，直到工作階段過期為止，或直到指定的結束時間(以先到者為準)。此模式對於能夠在偵測到事件時開始播放，並繼續在工作階段建立時尚未擷取的即時串流媒體非常有用。

決定開始時間戳記。

在這個例子中，我們使用 Unix 紀元時間以秒為單位的格式。如需[時間戳記格式](#)的詳細資訊，請參閱 AWS Command Line Interface 使用指南中的「時間戳記」一節。

如需轉換工具，請參閱 [UnixTime.org](https://unixtime.org)。

- 1708471800 等於二零二四年二月二十日下午 3:30 格林尼治標準時間

在此範例中，我們不指定結束時間戳記，這表示工作階段將繼續包含新擷取的媒體，直到工作階段過期為止。

在LIVE\_REPLAY播放模式和指定 [HLS 片段選擇器](#)的情況下調用 GetHLSStreamingSessionURL API。

```
aws kinesis-video-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode LIVE_REPLAY \  
  --hls-fragment-selector \  
  --start-time 1708471800
```



```
"FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800}"
```

## On-demand

對於隨選播放，HLS 媒體播放清單包含 HLS 片段選擇器指定的媒體。在媒體播放程式中播放此類工作階段時，使用者介面通常會顯示定位器控制項，以便在播放視窗中選擇要顯示的位置。

若要為串流的特定區段建立 URL，請先決定開始和結束時間戳記。

在這個例子中，我們使用 Unix 紀元時間以秒為單位的格式。如需[時間戳記格式](#)的詳細資訊，請參閱 AWS Command Line Interface 使用指南中的「時間戳記」一節。

如需轉換工具，請參閱 [UnixTime.org](https://unixtime.org)。

- 1708471800 等於二零二四年二月二十日下午 3:30 格林尼治標準時間
- 1708471860 等於二零二四年二月二十日下午 3:31 下午格林尼治標準時間

在ON\_DEMAND播放模式和指定 [HLS 片段選擇器](#)的情況下調用 GetHLSStreamingSessionURL API。

```
aws kinesis-video-archived-media get-hls-streaming-session-url \  
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \  
  --stream-name YourStreamName \  
  --playback-mode ON_DEMAND \  
  --hls-fragment-selector \  
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800, EndTimestamp=1708471860}"
```

### Note

如文檔中所述，時間戳記必須在彼此[the section called "HLSTimestampRange"](#)之間的 24 小時內。

您會收到如下所示的回應：

```
{
```

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

### Important

不要在未經授權的實體可以訪問它的地方共享或存儲此令牌。令牌提供對流內容的訪問。使用與 AWS 憑據一起使用的相同措施來保護令牌。

您可以使用此 URL 和任何 HLS 播放器來查看 HLS 流。

例如，使用 VLC 媒體播放器。

您也可以通過在蘋果 Safari 或 Microsoft 邊緣瀏覽器的位置欄中輸入 HLS 流會話 URL 來播放 HLS 流。

## 範例：在 HTML 和中使用 HLS JavaScript

下列範例顯示如何使用適用於 JavaScript v2 的 AWS SDK 擷取 Kinesis 視訊串流的 HLS 串流工作階段，並在網頁中播放。本範例說明如何在下列播放器播放影片：

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

在中檢視[完整的範例程式碼](#)和[託管網頁](#) GitHub。

程式碼逐步解說主題：

- [匯入適用於瀏覽器 JavaScript 的 AWS SDK](#)
- [設定 Kinesis Video Streams 用戶端](#)
- [擷取 HLS 播放的端點](#)
- [設定 Kinesis Video Streams 封存媒體用戶端](#)
- [擷取 HLS 串流工作階段 URL](#)
- [在網頁上顯示 HLS 流](#)

## 匯入適用於瀏覽器 JavaScript 的 AWS SDK

在網頁中，包括以下腳本標記，以將 JavaScript v2 的 AWS SDK 導入到項目中。

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

如需詳細資訊，請參閱 [AWS SDK 以取得 JavaScript 說明文件](#)。

## 設定 Kinesis Video Streams 用戶端

若要使用 HLS 存取串流視訊，請先建立並設定 Kinesis Video Streams 用戶端。如需其他驗證方法，請參閱在 [Web 瀏覽器中設定認證](#)。

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

該應用程式會在 HTML 頁面的輸入方塊擷取必要的值。

## 擷取 HLS 播放的端點

使用 Kinesis Video Streams 用戶端呼叫 [the section called "GetDataEndpoint"](#) API 以擷取端點。

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

此程式碼會將端點儲存在 `hlsDataEndpoint` 變數中。

## 設定 Kinesis Video Streams 封存媒體用戶端

在 Kinesis Video Streams 封存媒體用戶端的用戶端組態中，指定您在上一個步驟中取得的端點。

```
const archivedMediaClientConfig = {
```

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

## 擷取 HLS 串流工作階段 URL

使用 Kinesis Video Streams 封存媒體用戶端呼叫 [the section called “GetHLSStreamingSessionURL”](#) API 以擷取 HLS 播放 URL。

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

## 在網頁上顯示 HLS 流

當您擁有 HLS 串流工作階段的 URL 時，請將其提供給影片播放器。將 URL 提供給影片播放器的方法，因影片播放器種類而異。

### Video.js

請執行下列動作，將 [Video.js](#) 及其 CSS 類別匯入我們的瀏覽器指令碼：

```
<link rel="stylesheet" href="https://vjs.zencdn.net/6.6.3/video-js.css">
<script src="https://vjs.zencdn.net/6.6.3/video.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/videojs-contrib-hls/5.14.1/
videojs-contrib-hls.js"></script>
```

創建一個 video HTML 元素來顯示視頻：

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

設置 HLS 網址作為 HTML 視頻元素源：

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

## Shaka

請執行以下操作以將 [Google Saka 播放器](#) 導入到我們的瀏覽器腳本中：

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

創建一個 video HTML 元素來顯示視頻：

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

創建一個沙卡播放器指定視頻元素，並調用 load 方法：

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

## hls.js

執行以下操作以將 [hls.js](#) 導入我們的瀏覽器腳本中：

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

創建一個 video HTML 元素來顯示視頻：

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

創建一個 hls.js 播放器，給它 HLS 網址，並告訴它播放：

```
var playerElement = document.getElementById('hlsjs');
var player = new Hls();
player.loadSource(hlsUrl);
player.attachMedia(playerElement);
```

```
player.on(Hls.Events.MANIFEST_PARSED, function() {  
    video.play();  
});
```

## 疑難排解 HLS 問題

本節說明搭配 Kinesis 視訊串流使用 HTTP 即時串流 (HLS) 時可能會遇到的問題。

### 問題

- [擷取 HLS 串流工作階段 URL 成功，但在影片播放器播放時失敗](#)
- [製作者和播放器間的延遲太高](#)

### 擷取 HLS 串流工作階段 URL 成功，但在影片播放器播放時失敗

這種情況發生於當 HLS 串流工作階段 URL 可透過 `GetHLSStreamingSessionURL` 成功擷取，但是影片播放器卻無法透過提供的 URL 成功播放時。

若要排除這種情況，請嘗試：

- 判斷視訊串流是否在 Kinesis Video Streams 主控台中播放。請考慮主控台顯示的任何錯誤。
- 如果片段持續時間少於一秒，請將其增加到一秒。如果片段持續時間太短，服務可能會限制播放器，因為它太頻繁地發出視訊片段的要求。
- 請確認每個 HLS 串流工作階段 URL 僅由一個播放器使用。如果超過一個播放器正在使用單一 HLS 串流工作階段 URL，服務可能會因為收到太多請求而對這些播放器進行調節。
- 確認您的播放器支援您為 HLS 串流工作階段指定的所有選項。為以下參數嘗試不同的組合值：
  - `ContainerFormat`
  - `PlaybackMode`
  - `FragmentSelectorType`
  - `DiscontinuityMode`
  - `MaxMediaPlaylistFragmentResults`

有些媒體播放程式 (如 HTML5 和行動播放程式) 通常只支援具有 fMP4 容器格式的 HLS。其他媒體播放器 (例如 Flash 和自定義播放器) 可能只支持具有 MPEG TS 容器格式的 HLS。我們建議您嘗試 `ContainerFormat` 參數以開始疑難排解。

- 確認每個片段都有一致的軌道數量。確認串流中的片段在同時具有音訊和視訊軌道以及僅具有視訊軌道之間沒有變更。還要驗證編碼器設置 (分辨率和幀速率) 在每個軌道中的片段之間沒有改變。

## 製作者和播放器間的延遲太高

當視訊擷取時間與影片播放器播放的時間延遲太高時，便會發生此狀況。

系統會以每個片段的方式透過 HLS 播放影片。因此，延遲不得少於片段持續時間。延遲也包含緩衝和傳輸資料所需的時間。如果您的解決方案需要少於一秒的延遲，請考慮使用 GetMedia API。

您可以調整以下參數來降低整體延遲，但調整這些參數也可能降低影片品質或增加重新緩衝的時間。

- 片段持續時間 — 片段持續時間是由視訊編碼器產生的關鍵影格頻率所控制的串流中分割之間的視訊量。建議的值為一秒。擁有較短片段持續時間，代表在傳輸影片資料至服務之前所等待的片段完成時間較少。較短的片段也能讓服務處理得更快速。不過，如果片段持續時間過短，播放器因內容播畢而停止並緩衝內容的可能性也會提升。如果片段持續時間少於 500 毫秒，則製作者可能建立了太多請求以致服務開始對其進行調節。
- 位元速率 — 比特率較低的視頻流讀取，寫入和傳輸所需的時間更少。不過，較低位元速率的影片串流通常會有較差的影片品質。
- 媒體播放清單中的片段計數 — 對延遲敏感的播放器應該只載入媒體播放清單中的最新片段。大多數玩家從最早的片段開始。通過減少播放列表中的片段數量，可以減少先前片段和新片段之間的時間間隔。如果播放清單大小較小，如果播放清單新增片段有延遲，或者播放器取得更新的播放清單有延遲，則播放期間可能會略過片段。我們建議使用 3-5 個片段，並使用設定為只載入播放清單中最新片段的播放程式。
- 播放器緩衝區大小 — 大多數視頻播放器具有可配置的最小緩衝持續時間，通常默認值為 10 秒。如需將延遲降到最低，您可以將此值設為 0 秒。但是，這樣做意味著如果有任何延遲產生片段，播放器會重新緩衝，因為播放器將沒有緩衝區來吸收延遲。
- 播放器「catch」 — 如果緩衝區填滿，視頻播放器通常不會自動 catch 到視頻緩衝區前面的播放，例如延遲片段導致積壓的片段播放。自訂播放器可以透過刪除影格或提高播放速度 (例如提高至 1.1 倍速) 趕上緩衝之前來避免這點。此操作會造成播放不穩定，或在播放器追上進度時加快播放速度，而若緩衝大小保持過短時，重新緩衝可能更為頻繁。

## 使用 MPEG-DASH 播放影片

若要使用 MPEG DASH 檢視 Kinesis 視訊串流，請先使用 [GetDash URL 建立串流工作階段](#)。StreamingSession 這個動作會傳回一個 URL (包含工作階段字符) 以存取 MPEG-DASH 工作階段。然後您可以使用媒體播放器的 URL 或獨立應用程式來顯示串流。

Amazon Kinesis 影片串流對透過 MPEG-DASH 提供視訊有以下要求：

- 如需串流視訊播放曲目的需求，請參閱 [the section called “獲取破折號 StreamingSession 網址”](#)。

- 資料保留必須大於 0。
- 每個片段的影片軌道必須針對視訊媒體的 H.264 格式的進階視訊編碼 (AVC)，以及針對 H.265 格式的 HEVC 中包含轉碼器私有資料。如需詳細資訊，請參閱 [MPEG-4 specification ISO/IEC 14496-15](#)。如需將串流資料轉換為特定格式的相關資訊，請參閱 [NAL Adaptation Flags](#)。
- 每個片段的音訊軌道 (如果有) 必須包含 AAC 格式的編碼器私有資料 ([AAC 規格 ISO/IEC 13818-7](#)) 或 [MS Wave 格式](#)。

例如：在 HTML 和中使用 MPEG-破折號 JavaScript

下列範例顯示如何擷取 Kinesis 視訊串流的 MPEG-DASH 串流工作階段，並在網頁中播放。本範例說明如何在下列播放器播放影片：

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

## 主題

- [為 MPEG DASH 播放設定 Kinesis Video Streams 用戶端](#)
- [擷取用於 MPEG DASH 播放的 Kinesis Video Streams 封存內容端點](#)
- [擷取 MPEG-DASH 串流工作階段 URL](#)
- [播放 MPEG-DASH 以顯示串流影片](#)
- [完整的範例](#)

## 為 MPEG DASH 播放設定 Kinesis Video Streams 用戶端

若要使用 MPEG-DASH 存取串流視訊，請先建立並設定 Kinesis Video Streams 用戶端 (以擷取服務端點) 和封存的媒體用戶端 (以擷取 MPEG-DASH 串流工作階段)。該應用程式會在 HTML 頁面的輸入方塊擷取必要的值。

```
var streamName = $('#streamName').val();

// Step 1: Configure SDK Clients
var options = {
  accessKeyId: $('#accessKeyId').val(),
  secretAccessKey: $('#secretAccessKey').val(),
  sessionToken: $('#sessionToken').val() || undefined,
  region: $('#region').val(),
```



```

    endpoint: $('#endpoint').val() || undefined
  }
  var kinesisVideo = new AWS.KinesisVideo(options);
  var kinesisVideoArchivedContent = new AWS.KinesisVideoArchivedMedia(options);

```

## 擷取用於 MPEG DASH 播放的 Kinesis Video Streams 封存內容端點

啟動用戶端之後，擷取 Kinesis Video Streams 封存的內容端點，以便您可以擷取 MPEG-DASH 串流工作階段 URL，如下所示：

```

// Step 2: Get a data endpoint for the stream
console.log('Fetching data endpoint');
kinesisVideo.getDataEndpoint({
  StreamName: streamName,
  APIName: "GET_DASH_STREAMING_SESSION_URL"
}, function(err, response) {
  if (err) { return console.error(err); }
  console.log('Data endpoint: ' + response.DataEndpoint);
  kinesisVideoArchivedContent.endpoint = new AWS.Endpoint(response.DataEndpoint);

```

## 擷取 MPEG-DASH 串流工作階段 URL

當您擁有封存的內容端點時，請呼叫 [GetDash StreamingSession URL](#) API 以擷取 MPEG DASH 串流工作階段 URL，如下所示：

```

// Step 3: Get a Streaming Session URL
var consoleInfo = 'Fetching ' + protocol + ' Streaming Session URL';
console.log(consoleInfo);

if (protocol === 'DASH') {
  kinesisVideoArchivedContent.getDASHStreamingSessionURL({
    StreamName: streamName,
    PlaybackMode: $('#playbackMode').val(),
    DASHFragmentSelector: {
      FragmentSelectorType: $('#fragmentSelectorType').val(),
      TimestampRange: $('#playbackMode').val() === "LIVE" ? undefined : {
        StartTimestamp: new Date($('#startTimestamp').val()),
        EndTimestamp: new Date($('#endTimestamp').val())
      }
    }
  },
  DisplayFragmentTimestamp: $('#displayFragmentTimestamp').val(),

```

```
    DisplayFragmentNumber: $('#displayFragmentNumber').val(),
    MaxManifestFragmentResults: parseInt($('#maxResults').val()),
    Expires: parseInt($('#expires').val())
  }, function(err, response) {
    if (err) { return console.error(err); }
    console.log('DASH Streaming Session URL: ' + response.DASHStreamingSessionURL);
```

## 播放 MPEG-DASH 以顯示串流影片

當您擁有 MPEG-DASH 串流工作階段的 URL 時，請將其提供給影片播放器。將 URL 提供給影片播放器的方法，因影片播放器種類而異。

以下程式碼範例說明如何提供串流工作階段 URL 到 [Google Shaka](#) 播放器：

```
// Step 4: Give the URL to the video player.

//Shaka Player elements
<video id="shaka" class="player" controls autoplay></video>
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js">
</script>
...

var playerName = $('#player').val();

if (playerName === 'Shaka Player') {
  var playerElement = $('#shaka');
  playerElement.show();

  var player = new shaka.Player(playerElement[0]);
  console.log('Created Shaka Player');

  player.load(response.DASHStreamingSessionURL).then(function() {
    console.log('Starting playback');
  });
  console.log('Set player source');
}
```

以下程式碼範例說明如何提供串流工作階段 URL 到 [dash.js](#) 播放器：

```
<!-- dash.js Player elements -->
<video id="dashjs" class="player" controls autoplay=""></video>
<script src="https://cdn.dashjs.org/latest/dash.all.min.js"></script>

...

var playerElement = $('#dashjs');
playerElement.show();

var player = dashjs.MediaPlayer().create();
console.log('Created DASH.js Player');

player.initialize(document.querySelector('#dashjs'), response.DASHStreamingSessionURL,
  true);
console.log('Starting playback');
console.log('Set player source');
}
```

## 完整的範例

您可以在上[下載或檢視已完成的範例程式碼](#) GitHub。

## 搭配 Kinesis 視訊串流使用串流中繼資料

您可以使用 Amazon Kinesis Video Streams 製作者開發套件，在 Kinesis 視訊串流中的個別片段層級內嵌中繼資料。Kinesis Video Streams 中的中繼資料是可變的索引鍵值配對。您可以使用它來描述片段的內容，嵌入必須與實際片段一起傳輸的相關傳感器讀數，或者滿足其他自定義需求。中繼資料提供為 [the section called “GetMedia”](#) 或 [the section called “GetMediaForFragmentList”](#) API 操作的一部分。在流保留期的整個持續時間內，它與片段一起存儲。您的使用應用程式可以讀取、處理和回應，根據中繼資料，使用[Kinesis 視頻流解析器庫](#)。

中繼資料可以用兩種模式內嵌於串流中的片段：

- 非持續性 — 您可以根據已發生的業務特定條件，在串流中的片段上單次或特定基礎上，將中繼資料附加到串流中的片段。例如，智慧型攝影機會偵測動作並將中繼資料新增至包含動作的對應片段，然後再將片段傳送至 Kinesis 視訊串流。您可以將中繼資料套用至下列格式的片段中：`Motion = true`。
- 持續性 — 您可以根據持續需求，將中繼資料附加至串流中連續的連續片段。例如，智慧型攝影機會傳送與傳送至 Kinesis 視訊串流的所有片段相關聯的目前經緯度座標。您可以將中繼資料套用至所有下列格式的片段中：`Lat = 47.608013N`，`Long = -122.335167W`

您可以根據應用程式的需求，同時在同一片段附加兩種模式的中繼資料。內嵌的中繼資料可能包括偵測到的物件、追蹤活動、GPS 座標等任何您欲附加至串流片段的自訂資料。中繼資料是以鍵值字串對編碼。

## 主題

- [將中繼資料新增至 Kinesis 視訊串流](#)
- [使用內嵌在 Kinesis 視訊串流中的中繼資料](#)
- [串流元數據限制](#)

## 將中繼資料新增至 Kinesis 視訊串流

您新增至 Kinesis 視訊串流的中繼資料會建模為 MKV 標籤，並以索引鍵值配對的方式實作。

中繼資料可以是暫時性，例如標記串流內的事件，或持久性，例如識別指定事件發生的片段。持續性中繼資料項目會保留，並套用至每個連續片段，直到取消為止。

### Note

使用 [Producer 資料庫](#) 新增的中繼資料項目不同於以 [the section called “TagStream”](#)、[the section called “UntagStream”](#)、[the section called “ListTagsForStream”](#) 執行的串流層級標記 API。

## 串流中繼資料 API

您可以使用下列製作者開發套件中的操作來執行串流中繼資料。

### PIC

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

### C++ 製作者開發套件

```
/**
```

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

## Java 製作者開發套件

您可以使用 Java 生產者 SDK，將元數據添加到MediaSource使用MediaSourceSink.onCodecPrivateData：

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

## 持續性和非永久性中繼資

對於非持久性中繼資料，您可以新增數個相同名稱的中繼資料項目。在中繼資料項目附加至下一個片段前，製作者開發套件會搜集中繼資料佇列中的所有中繼資料項目。中繼資料項目套用至串流後，中繼資料佇列將清空。若要重複中繼資料，請再次呼叫 putKinesisVideoFragmentMetadata 或 putFragmentMetadata。

對於持久的中繼資料，製作者開發套件將比照非持久中繼資料，收集中繼資料佇列中的所有中繼資料項目。但是，當中繼資料項目附加到下一個片段前面時，不會從佇列中移除。

呼叫 putKinesisVideoFragmentMetadata 或 putFragmentMetadata，persistent 設定為 true，有以下行為：

- 呼叫 API 將中繼資料項目放入佇列。佇列中的中繼資料項目將以 MKV 標籤新增至每個片段。
- 如先前新增的中繼資料項目覆寫另一項目，請以相同名稱不同數值呼叫 API。
- 若以空數值呼叫 API，將移除 (取消) 中繼資料佇列中的中繼資料項目。

## 使用內嵌在 Kinesis 視訊串流中的中繼資料

若要使用 Kinesis 視訊串流中的中繼資料，請使用下列項目的MkvTagProcessor實作：

```
public interface MkvTagProcessor {
    default void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
```

```
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
    default void clear() {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
}
}
```

此界面位於 [Kinesis 視頻流解析器庫](#) 中的 [FragmentMetadataVisitor](#) 類別。

`FragmentMetadataVisitor` 類別包含 `MkvTagProcessor` 的一個實作：

```
public static final class BasicMkvTagProcessor implements
FragmentMetadataVisitor.MkvTagProcessor {
    @Getter
    private List<MkvTag> tags = new ArrayList<>();

    @Override
    public void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        tags.add(mkvTag);
    }

    @Override
    public void clear() {
        tags.clear();
    }
}
```

`KinesisVideoRendererExample` 類別包含如何使用 `BasicMkvTagProcessor` 的範例。以下範例將新增 `BasicMkvTagProcessor` 至應用程式的 `MediaProcessingArguments` 中。

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

片段中繼資料送達時將呼叫 `BasicMkvTagProcessor.process` 方法。您可以用 `GetTags` 擷取累積的中繼資料。若要擷取單一中繼資料項目，`clear`請先呼叫清除收集的中繼資料，然後再次擷取中繼資料項目。

## 串流元數據限制

如[the section called “片段元數據配額”](#)需將串流中繼資料新增至 Kinesis 視訊串流的限制的詳細資訊，請參閱

## Kinesis Video Streams 資料模型

[Producer 資料庫](#) 和 [串流剖析器程式庫](#) 會以可支援嵌入資訊搭配影片資料的格式來傳送和接收影片資料。此格式取決於 Matroska (MKV) 規格。

[MKV 格式](#)是適用於媒體資料的開放規格。Amazon Kinesis Video Streams 開發人員指南中的所有程式庫和程式碼範例都以 MKV 格式傳送或接收資料。

[室壁運動視頻流生產者庫](#)使用StreamDefinition和Frame類型來生成 MKV 流標題，幀標題和幀數據。

如需完整 MKV 規格的詳細資訊，請參閱 [Matroska 規格](#)。

下列各節說明 [C++ Producer Library](#) 產生的 MKV 格式化之資料的元件。

### 主題

- [流頭元素](#)
- [串流軌跡資料](#)
- [框架標題元素](#)
- [MKV 幀數據](#)

## 流頭元素

以下 MKV 標頭元素會由 StreamDefinition 所使用 (在 StreamDefinition.h 中加以定義)。

Element	描述	典型值
stream_name	對應至 Kinesis 視訊串流的名稱。	my-stream
retention_period	Kinesis 影片串流會保留串流資料的持續時間 (以小時為單	24

Element	描述	典型值
	位)。指0定不保留資料的串流。	
tags	使用者資料的金鑰值集合。此資料會顯示在 AWS Management Console 且可透過用戶端應用程式讀取來篩選或取得串流的相關資訊。	
kms_key_id	如果存在，則使用者定義AWS KMS金鑰來加密串流上的資料。如果不存在，則會使用 Kinesis 提供的金鑰 ( ) 加密資料。aws/kinesis-video	01234567-89ab-cdef-0123-456789ab
streaming_type	目前唯一有效串流類型為：STREAMING_TYPE_REALTIME 。	STREAMING_TYPE_REALTIME
content_type	使用者定義的內容類型。對於要在主控台中播放的串流影片資料，內容類型必須是 video/h264 。	影片/h264
max_latency	此值目前未使用，應該設定為0。	0
fragment_duration	片段應持續多久的預估值，此值會用於優化。實際片段持續時間取決於串流資料。	2



Element	描述	典型值
timecode_scale	<p>表示影格時間戳記使用的比例。預設設定為 1 毫秒。指定 0 也會指派 1 毫秒的預設值。這個值可以介於 100 奈秒和 1 秒之間。</p> <p>如需詳細資訊，請參閱 Matroska 文件 <a href="#">TimecodeScale</a> 中的。</p>	
key_frame_fragmentation	若為 true，在收到關鍵影格時，串流即會啟動新叢集。	true
frame_timecodes	如果 true，Kinesis Video Streams 會使用簡報時間戳記 (pts) 和解碼接收影格的時間戳記 (dts) 值。如果 false，Kinesis Video Streams 在收到影格時會使用系統產生的時間值加上戳記。	true
absolute_fragment_time	若是 true，會將叢集時間碼解譯為使用絕對時間 (例如，透過生產者的系統時鐘)。若是 false，會將叢集時間碼解譯為串流的相對開始時間。	true
fragment_acks	如果是 true，則會在 Kinesis Video Streams 收到資料時傳送確認 (ACK)。您可以使用 <code>KinesisVideoStreamFragmentAck</code> 或 <code>KinesisVideoStreamParseFragmentAck</code> 回呼來接收 ACK。	true

Element	描述	典型值
restart_on_error	指出串流應在發生串流錯誤後恢復傳輸。	true
nal_adaptation_flags	指出 NAL (網路抽象層) 適應性或轉碼器私有資料在內容中是否存在。有效旗標包括 NAL_ADAPTATION_ANNEXB_NALS 與 NAL_ADAPTATION_ANNEXB_CPD_NALS。	NAL_ADAPTATION_ANNEXB_NALS
frame_rate	內容影格率的預計值。這個值是用於最佳化；實際影格率取決於傳入資料的速率。指定 0 會指派預設值 24。	24
avg_bandwidth_bps	內容頻寬的估計值，以 Mbps 為單位。這個值是用於最佳化；實際速率取決於傳入資料的頻寬。例如，對於執行 25 FPS 的 720 p 解析度影片串流，您可以預期平均頻寬為 5 Mbps。	5
buffer_duration	內容在生產者上緩衝的持續時間。如果網路延遲很低，則可以減少此值。如果網路延遲很高，則增加此值可防止因配置無法將框架放入較小的緩衝區而導致訊框在傳送之前丟棄。	

Element	描述	典型值
replay_duration	如果連線中斷，視訊資料串流會「倒回」的時間長度。如果不擔心連線中斷而遺失影格，則此值可以為零。如果消耗的應用程式可以移除多餘的框架，則該值可以增加。此值應小於緩衝區持續時間，否則會使用緩衝區持續時間。	
connection_staleness	沒有接收到任何資料時，連線的維護持續時間。	
codec_id	內容使用的轉碼器。如需詳細資訊，請參閱 Matroska 規格中的 <a href="#">CodecID</a> 。	V_MPEG2
track_name	使用者定義的追蹤名稱。	my_track
codecPrivateData	編碼器提供用於解碼影格資料的資料 (例如影格寬度和高度 (以像素為單位))，許多下游消費者會需要此資料。在 <a href="#">C++ Producer Library</a> 中，MkvStatics.cpp 中的 gMkvTrackVideoBits 陣列會包含影格的像素寬度與高度。	
codecPrivateData大小	codecPrivateData 參數中的資料大小。	
track_type	串流的軌道類型。	MKV_TRACK_INFO_TYPE_AUDIO 或 MKV_TRACK_INFO_TYPE_VIDEO

Element	描述	典型值
segment_uuid	使用者定義的區段 uuid (16 個位元組)。	
default_track_id	軌道的唯一非零編號。	1

## 串流軌跡資料

以下 MKV 軌道元素會由 StreamDefinition 所使用 (在 StreamDefinition.h 中加以定義)。

Element	描述	典型值
track_name	使用者定義的軌道名稱。例如，「音訊」代表音軌。	音訊
codec_id	軌道的轉碼器 ID。例如，「A_AAC」代表音軌。	A_AAC
cpd	編碼器用於解碼影格資料所提供的資料。此資料可以包含影格寬度和高度 (以像素為單位)，許多下游消費者都需要此資訊。在 <a href="#">C++ 生產者庫</a> 中，MkvStatics.cpp 中的 gMkvTrackVideoBits 數組包含框架的像素寬度和高度。	
cpd_size	codecPrivateData 參數中資料的大小。	
track_type	軌道的類型。例如，您可以針對音訊使用 MKV_TRACK_INFO_TYPE_AUDIO 的列舉值。	MKV_TRACK_INFO_TYPE_AUDIO

## 框架標題元素

以下 MKV 標頭元素會由 Frame 所使用 (在 mkvgen/Include.h 中的 KinesisVideoPic 套件中加以定義)。

- 影格索引：單調增加的值。
- 旗標：影格的類型。有效值包括以下項目：
  - FRAME\_FLAGS\_NONE
  - FRAME\_FLAG\_KEY\_FRAME：如果 key\_frame\_fragmentation 是在串流上設定，關鍵影格會開始新的片段。
  - FRAME\_FLAG\_DISCARDABLE\_FRAME：告知解碼器，當解碼速度太慢時，其可以放棄此影格。
  - FRAME\_FLAG\_INVISIBLE\_FRAME：此區塊的持續時間為 0。
- 解碼時間戳記：解碼此影格的時間戳記。如果先前的影格依賴此影格進行解碼，則此時間戳記可能會比較早前影格的時間戳記。這個值相對於該片段的開始。
- 簡報時間戳記：顯示此影格的時間戳記。這個值相對於該片段的開始。
- 持續時間：影格播放的持續時間。
- 大小：影格資料大小 (以位元組為單位)

## MKV 幀數據

frame.frameData 中的資料可能僅包含影格的媒體資料，也可能包含進一步巢狀標頭資訊 (根據所使用之編碼的結構描述而定)。若要顯示在中 AWS Management Console，資料必須以 [H.264](#) 轉碼器編碼，但 Kinesis Video Streams 可以接收任何格式的時間序列化資料串流。

# 開始使用 Amazon Kinesis Video Streams

本節說明如何在 Amazon Kinesis Video Streams 中執行下列任務：

- 設置您的 AWS 帳戶 並創建管理員，如果您尚未這樣做。
- 建立 Kinesis 視訊串流。
- 從攝影機將資料傳送至 Kinesis 視訊串流，並在主控台中檢視媒體。

如果您是 Amazon Kinesis Video Streams 的新手，建議您[Kinesis Video Streams：它是如何工作的](#)先閱讀。

## Note

遵循入門示例將不會對您產生任何費用 AWS 帳戶。如需您所在區域的資料成本，請參閱 [Amazon Kinesis Video Streams 定價](#)。

## 主題

- [設定帳戶](#)
- [建立 Kinesis 影片串流](#)
- [將資料傳送至 Amazon Kinesis 影片串流](#)
- [使用媒體資料](#)

## 設定帳戶

第一次使用 Amazon Kinesis Video Streams 之前，請先完成以下任務。

## 主題

- [註冊一個 AWS 帳戶](#)
- [建立管理使用者](#)
- [建立 AWS 帳戶 金鑰](#)

## 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。

當您註冊一個時 AWS 帳戶，將創建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。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

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。

## 建立管理使用者

註冊後，請確保您的安全 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，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 \(主控台\)](#)。

建立管理使用者

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 Identity Center 中，將管理權限授予管理使用者。

[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」。](#)

## 以管理員的身分簽署

- 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](#)。

## 建立 AWS 帳戶 金鑰

您需要一個金 AWS 帳戶 鑰，才能以程式設計方式存取 Amazon 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。記錄 Secret access key (私密存取金鑰)。

## 建立 Kinesis 影片串流

本節說明如何建立 Kinesis 視訊串流。

本節包含下列程序：

- [the section called “使用主控台建立視訊串流”](#)
- [the section called “使用建立視訊串流 AWS CLI”](#)

## 使用主控台建立視訊串流

1. 在開啟主控台 <https://console.aws.amazon.com/kinesisvideo/home>。



2. 在 Video streams (影片串流) 頁面上，選擇 Create video stream (建立影片串流)。
3. 在 [建立新的視訊串流] 頁面上，輸入 *YourStreamName* 串流名稱。保持選取 [預設組態] 按鈕。
4. 選擇 Create video stream (建立影片串流)。
5. Amazon Kinesis 影片串流建立串流後，請檢閱 *YourStreamName* 頁面上的詳細資訊。

## 使用建立視訊串流 AWS CLI

1. 確認您已 AWS CLI 安裝並設定。如需詳細資訊，請參閱 [AWS Command Line Interface](#) 文件。
2. 在 AWS CLI 中執行以下 Create-Stream 命令：

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

回應將會如下所示：

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

## 將資料傳送至 Amazon Kinesis 影片串流

本節說明如何將媒體資料從攝影機傳送到您在上一節中建立的 Kinesis 視訊串流。本區段使用 [C++ Producer Library](#) 做為 [GStreamer](#) 外掛程式。

為了從各種作業系統上的各種裝置傳送媒體，本教學課程使用 Kinesis Video Streams C++ 製作者程式庫和 [GStreamer](#)，這是一種將攝影機和其他媒體來源存取標準化的開放原始碼媒體架構。

### 主題

- [建置 SDK 和範例](#)
- [執行範例以將媒體上傳至 Kinesis Video Streams](#)
- [審核確認物件](#)

## 建置 SDK 和範例

您可以在計算機上或中構建 SDK 和示例 AWS Cloud9。請遵循以下適當的程序。

### Build on your computer

使用 [README 檔案](#) 中的指示來建置製作者程式庫和範例應用程式。

其中包含：

- 安裝相依項目
- 複製存放庫
- 使用 CMake 生成生成文件
- 使用 make 構建二進製文件

### Build in AWS Cloud9

請依照下列程序上傳至中 AWS Cloud9 的 Kinesis Video Streams。您無需將任何內容下載到計算機上。

1. 在中 AWS Management Console，開啟 [AWS Cloud9](#)。

選取 [建立環境]。

2. 在「建立環境」畫面上，完成下列操作：

- 名稱-輸入新環境的名稱。
- 平台-選擇 Ubuntu 服務器 22.04 LTS。

您可以將其他欄位保留為預設選項。

3. 建立環境後，請在 Cloud9 IDE 資料行中選取「開啟」。

在螢幕的中間下方區域，您會看到 `Admin:~/environment $`。這是 AWS Cloud9 (Amazon EC2) 終端。

#### Note

如果不小心關閉終端機，請選取「視窗」，「新增終端機」。

在終端機中執行下列命令，將磁碟區變更為 20 GiB。

- a. 下載 指令碼。

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

- b. 授予指令碼執行權限。

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

- c. 執行指令碼。

```
./resize_volume.sh
```

4. 獲取有關您可以通過高級打包工具 ( APT ) 安裝或更新的所有軟件的最新信息。

這個命令不會更新軟體本身，但請確定您的系統知道最新的可用版本。

```
sudo apt-get update
```

5. 安裝 C ++ 生產者 SDK 依賴關係。

```
sudo apt-get install -y cmake m4 git build-essential pkg-config libssl-dev  
libcurl4-openssl-dev \  
liblog4cplus-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \  
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad gstreamer1.0-plugins-  
good \  
gstreamer1.0-plugins-ugly gstreamer1.0-tools
```

6. 使用 git 克隆 C ++ 生產者 SDK。

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

7. 準備一個構建目錄。

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

## 8. 使用 CMake 生成生成文件。

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

預期的輸出結束如下所示：

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

## 9. 使用 make 編譯 SDK 和示例應用程序，以及構建最終的可執行文件。

```
make
```

預期的輸出結束如下所示：

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

## 10. 確認範例檔案是否已建立。列出當前目錄中的文件：

```
ls
```

確認存在下列檔案：

- 克斯特雷姆\_ 樣本
- 李格斯特克斯坦

## 執行範例以將媒體上傳至 Kinesis Video Streams

範例應用程式不支援 IMDS 認證。在終端機中，為您的 IAM 使用者或角色以及串流所在區域匯出 AWS 登入資料。

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

如果您使用的是臨時 AWS 憑據，請同時導出會話令牌：

```
export AWS_SESSION_TOKEN=YourSessionToken
```

## .mp4 files

下載範例 .mp4 影片以上傳至 Kinesis Video Streams。

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

使用您之前創建的流的名稱運行以下命令。如果您尚未建立串流，請參閱[the section called “建立 Kinesis 影片串流”](#)。

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

## Sample video from GStreamer

使用以下命令可以使用 GStreamer 生成視頻。

告訴在哪裡可以找到 kvssink GStreamer 插件。在您的構建目錄中，指定包含該文件的 libgstkvssink.so 文件夾的路徑。

從您的構建目錄中，運行以下命令：

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

此 GStreamer 管道會產生具有標準測試模式的即時測試視訊串流，該模式以每秒 10 個畫面的速度執行，解析度為 640x480 像素。會新增顯示目前系統時間和日期的覆蓋圖。然後將視訊編碼為 H.264 格式，最多每 10 個影格產生一次關鍵影格，因此片段持續時間（也稱為一組圖片 (GoP) 大小）為 1 秒。kvssink 採用 H.264 編碼的視訊串流，將其封裝成 Matroska (MKV) 容器格式，並將其上傳到您的 Kinesis 視訊串流。

執行以下命令：

```
gst-launch-1.0 -v videotestsrc is-live=true \  
! video/x-raw,framerate=10/1,width=640,height=480 \  
! clockoverlay time-format="%a %B %d, %Y %I:%M:%S %p" \  
! x264enc bframes=0 key-int-max=10 \  
! h264parse \  
! kvssink stream-name="YourStreamName"
```

若要停止 GStreamer 管線，請選取終端機視窗，然後按下 CTRL+C 鍵。

### Note

如需有關使用 GStreamer 外掛程式從攝影機或 USB 攝影機串流 RTSP 串流視訊的詳細資訊，請參閱。[範例：Kinesis Video Streams 產生器 SDK 外掛程式](#)

## 審核確認物件

在上傳期間，Kinesis Video Streams 會將確認物件傳送回執行上傳作業的用戶端。您應該在命令輸出中看到這些內容。範例如下所示：

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

如果確認 EventType 是 PERSISTED，則表示 Kinesis Video Streams 已經持久地儲存和加密這個媒體區塊，以便擷取、分析和長期儲存。

如需有關確認的詳細資訊，請參閱。[the section called “PutMedia”](#)

## 使用媒體資料

您可以在主控台中檢視媒體資料，或建立使用「超文字即時串流」(HLS) 從串流讀取媒體資料的應用程式來取用媒體資料。

### 在主控台中檢視媒體

在其他瀏覽器索引標籤中，開啟 AWS Management Console。在 Kinesis Video Streams 儀表中，選取[視訊串流](#)。

在串流清單中選取串流名稱。如有必要，請使用搜尋列。

展開「媒體播放」區段。如果視頻仍在上傳，則將顯示該視頻。如果上傳完成，請選取雙向左箭頭。

### 使用 HLS 消耗媒體數據

您可以建立使用 HLS 從 Kinesis 視訊串流取用資料的用戶端應用程式。如需關於使用 HLS 建立使用媒體資料的應用程式詳細資訊，請參閱[the section called “影片播放”](#)。

# Amazon Kinesis Video Streams 邊緣代理

Amazon Kinesis Video Streams 提供高效率且符合成本效益的方式，可連接到客戶現場部署的 IP 攝影機。使用 Amazon Kinesis Video Streams Edge 代理程式，您可以在本機錄製並存放攝影機的視訊，並按照客戶定義的排程將影片串流到雲端，以進行長期儲存、播放和分析處理。

## Note

若要存取 Amazon Kinesis Video Streams 邊緣代理程式，請填寫此[簡短表單](#)。

您可以下載 Amazon Kinesis Video Streams 邊緣代理程式，並將其部署到現場部署邊緣運算裝置。您也可以輕鬆地將它們部署到 Amazon EC2 執行個體上執行的 Docker 容器中。部署後，您可以使用 Amazon Kinesis Video Streams API 來更新視訊錄製和雲端上傳設定。該功能適用於任何可以通過 RTSP 協議流式傳輸的 IP 攝像機。它不需要任何額外的固件部署到相機。

我們為 Amazon Kinesis Video Streams 邊緣代理程式提供下列安裝：

- 做為AWS IoT Greengrass V2元件：您可以在任何AWS IoT Greengrass認證的裝置上安裝 Amazon Kinesis Video Streams 邊緣代理程式做為AWS IoT Greengrass元件。若要進一步了解 AWS IoT Greengrass，請參閱 [《AWS IoT Greengrass Version 2 開發人員指南》](#)。
- 開啟AWS Snowball Edge：您可以在 Snowball 邊緣裝置上執行 Amazon Kinesis Video Streams 邊緣代理程式。若要深入了解，請參閱 [AWS SnowballEdge 開發人員指南](#)。
- 在原生AWS IoT部署上：您可以在任何運算執行個體上以原生方式安裝 Amazon Kinesis Video Streams 邊緣代理程式。邊緣 SDK 用[AWS IoT Core](#)於管理邊緣透過[the section called “Amazon Kinesis Video Streams”](#)。

若要開始使用 Amazon Kinesis Video Streams 邊緣代理程式，請繼續執行以下適當的程序。

## 主題

- [Amazon Kinesis Video Streams 邊緣代理程式 API 作業](#)
- [監控 Amazon Kinesis Video Streams 邊緣代理](#)
- [在非模式下AWS IoT Greengrass執行 Amazon Kinesis Video Streams 邊緣代理程式](#)
- [將 Amazon Kinesis Video Streams 邊緣代理程式部署到 AWS IoT Greengrass](#)
- [Amazon Kinesis Video Streams 邊緣代理常見問題](#)

# Amazon Kinesis Video Streams 邊緣代理程式 API 作業

使用下列 API 操作來設定 Amazon Kinesis Video Streams 邊緣代理程式：

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

## 監控 Amazon Kinesis Video Streams 邊緣代理

若要監控您的 Amazon Kinesis Video Streams 邊緣代理程式，請參閱[the section called “使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch”](#)。

## 在非模式下AWS IoT Greengrass執行 Amazon Kinesis Video Streams 邊緣代理程式

請依照下列步驟執行 Amazon Kinesis Video Streams 邊緣代理程式，並將 AWS IoT MQTT 做為獨立部署執行。

### 主題

- [步驟 1：在設備上安裝必要的依賴關係](#)
- [步驟 2：為您的 IP 攝影機 RTSP 網址建立亞馬遜室運動影片串流和AWS Secrets Manager資源](#)
- [步驟 3：建立 IAM 許可政策](#)
- [步驟 4：建立 IAM 角色](#)
- [步驟 5：建立AWS IoT角色別名](#)
- [步驟 6：建立原AWS IoT則](#)
- [步驟 7：創建一個AWS IoT東西並獲得憑據 AWS IoT Core](#)
- [步驟 8：建置並執行 Amazon Kinesis Video Streams 邊緣代理程式](#)
- [步驟 9：\(選擇性\) 在裝置上安裝 CloudWatch 代理程式](#)
- [步驟 10：\(選用\) 以原生程序執行 Amazon Kinesis Video Streams 邊緣代理程式](#)



## 步驟 1：在設備上安裝必要的依賴關係

### Note

如需支援的作業系統清單，請參閱[the section called “Amazon Kinesis Video Streams 邊緣代理程式支援哪些作業系統？”](#)。

### 在設備上安裝依賴關係

- 若要執行 Amazon Kinesis Video Streams 邊緣代理程式，請在您的裝置上安裝下列適當的程式庫：

#### Ubuntu

類型：

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

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

#### Amazon Linux 2

類型：

```
sudo yum update -y && sudo yum upgrade -y && sudo yum clean all -y
sudo yum install -y gcc-c++ openssl-devel libcurl-devel gstreamer1* wget \
java-11-amazon-corretto tar
```

log4cplus-2.1.0 從來源安裝。

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/
log4cplus-2.1.0.tar.gz
tar -xzvf log4cplus-2.1.0.tar.gz
cd log4cplus-2.1.0 && \
```

```
mkdir build && \  
cd build && \  
cmake .. && \  
sudo make && \  
sudo make install
```

apache-maven-3.9.2從來源安裝。

```
wget https://d1cdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-  
bin.tar.gz  
RUN tar -xzvf apache-maven-3.9.2-bin.tar.gz -C /opt
```

### Important

如果您看到一個畫面，告訴您某些服務需要重新啟動，請按 Enter 鍵以選取 [確定]。

如需詳細資訊，請參閱 [Amazon 網路 11 使用者指南](#)。

2. 安裝AWS Command Line Interface。請參閱《使用指南》[中的〈安裝或更新最新版本的AWS CLI程序AWS Command Line Interface〉](#)。

## 步驟 2：為您的 IP 攝影機 RTSP 網址建立亞馬遜室運動影片串流和AWS Secrets Manager資源

請遵循下列程序來建立中所需的串流和密碼AWS Secrets Manager。請先執行此步驟，因為您需要策略中已建立資源的 ARN。

### 創建 Amazon Kinesis Video Streams

使用AWS Management Console、AWS CLI或 API 建立 Amazon Kinesis Video Streams。

在中AWS Management Console，開啟 [Amazon Kinesis Video Streams 主控台](#)。選擇視頻流在左側導航中。

如需詳細資訊，請參閱 [the section called “建立 Kinesis 影片串流”](#)。

### 在 AWS Secrets Manager 中建立秘密

在中AWS Management Console，開啟主[AWS Secrets Manager控制台](#)。選擇左側導覽列中的 [秘密]。


確認已選取適當的「區域」。

1. 選擇存放新的機密。

a. 步驟 1：選擇秘密類型

- 選取其他機密類型。
- 在「鍵/值對」部分中，添加鍵值對。

索引鍵：MediaURI

 Note

金鑰必須是MediaURI。這是區分大小寫的。如果輸入錯誤，應用程式將無法正常工作。

值：*Your MediaURI*。


Example

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

- b. 步驟 2：設定密碼。給這個秘密起個名字。將其命名為任何你想要的。
  - c. 步驟 3：配置旋轉-可選。選擇下一步。
  - d. 步驟 4：檢閱。選擇儲存。
2. 如果您的密碼未立即顯示，請選取重新整理按鈕。

選擇您的秘密名稱。記下秘密 ARN。

3. 對要從中流式傳輸的每個媒體 URI 重複此過程。

 Note

AWS網路封鎖一些公用 RTSP 來源。您無法從 Amazon EC2 執行個體中存取這些執行個體，或者如果您在連線到 VPN 時未受管執行。

**⚠ Important**

您的攝影機 RTSP 網址應該以 h.264 格式串流影片。片段持續時間不得超過中提到的限制 [the section called “製作者 SDK 限制”](#)。

Amazon Kinesis Video Streams 邊緣代理程式僅支援視訊。

運行 `gst-discoverer-1.0 Your RtspUrl` 以確保您的相機可以從設備訪問。

為您建立的所有串流和密碼儲存 ARN。您需要這些來進行下一步。

### 步驟 3：建立 IAM 許可政策

請依照下列程序建立 IAM 政策。此權限原則允許 AWS 資源的選擇性存取控制 (支援作業的子集)。在這種情況下，AWS 資源就是您希望 Amazon Kinesis 影片串流邊緣代理程式串流至的視訊串流。這些資源還包括 Amazon Kinesis Video Streams 邊緣代理程式可擷取的 AWS Secrets Manager 密碼。如需詳細資訊，請參閱 [IAM 政策](#)。

使用 JSON 政策編輯器建立政策

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在左側導覽窗格中選擇 Policies (政策)。

如果這是您第一次選擇 Policies (政策)，將會顯示 Welcome to Managed Policies (歡迎使用受管政策) 頁面。選擇 Get Started (開始使用)。

3. 在頁面頂端，選擇 Create policy (建立政策)。
4. 在政策編輯器中，選擇 JSON 選項。
5. 輸入下列 JSON 政策文件：

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

```

        "kinesisvideo:ListStreams",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
    ]
},
{
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
    ]
}
]
}

```

### Note

arn:aws:kinesisvideo:\*:\*:stream/streamName2/\* 用視訊串流的 ARN 取代 arn:aws:secretsmanager:\*:\*:secret:\* 和 [MediaURI 密碼的 ARN](#) 取代。arn:aws:kinesisvideo:\*:\*:stream/streamName1/\* [the section called “2. 為您的 IP 攝像機 RTSP 網址創建資源”](#) 針對您希望 Amazon Kinesis Video Streams 邊緣代理程式存取的密碼，請使用 ARN。

## 6. 選擇下一步。

### Note

您可以隨時切換視覺化與 JSON 編輯器選項。不過，如果您進行變更或在視覺化編輯器中選擇下一步，IAM 就可能調整您的政策結構，以便針對視覺化編輯器進行最佳化。如需詳細資訊，請參閱 IAM 使用者指南中的[政策重組](#)。

7. 在 [檢閱並建立] 頁面上，輸入您要建立之原則的原則名稱和選擇性說明。檢視此政策中定義的許可，來查看您的政策所授予的許可。
8. 選擇 Create policy (建立政策) 儲存您的新政策。

## 步驟 4：建立 IAM 角色

您在此步驟中建立的角色可假設 AWS IoT 為從 AWS Security Token Service (AWS STS) 取得暫時認證。這是在從 Amazon Kinesis Video Streams 邊緣代理程式執行登入資料授權要求時完成的。

為 Amazon Kinesis Video Streams (IAM 主控台) 建立服務角色

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 選擇 [自訂信任原則] 角色類型並貼上下列原則：

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

4. 選取您在中建立的 IAM 政策旁邊的核取方塊 [the section called “3. 建立 IAM 許可政策”](#)。
5. 選擇下一步。
6. 輸入角色名稱或角色名稱尾碼，以協助您識別此角色的用途。

## Example

範例：KvsEdgeAgentRole

7. (選用) 在 Description (說明) 中，輸入新角色的說明。
8. (選擇性) 將標籤附加為索引鍵/值配對，將中繼資料新增至角色。

如需在 IAM 中使用標籤的詳細資訊，請參閱 [IAM 使用者指南中的標記 IAM 資源](#)。

9. 檢閱角色，然後選擇 Create role (建立角色)。

## 步驟 5：建立AWS IoT角色別名

請遵循下列程序，為AWS IoT您在中建立的 IAM 角色建立角色別名 [the section called “4. 建立 IAM 角色”](#)。角色別名是指向 IAM 角色的替代資料模型。AWS IoT登入資料提供者請求必須包含角色別名，以指出要承擔的 IAM 角色，以便從 AWS Security Token Service (AWS STS) 取得臨時登入資料。如需詳細資訊，請參閱 [如何使用憑證取得安全性權杖](#)。

### 建立AWS IoT角色別名

1. 請登入AWS Management Console並開啟AWS IoT Core主控台，網址為 <https://console.aws.amazon.com/iot/>。
2. 確認已選取適當的「區域」。
3. 在左側導覽中，選取 [安全性]，然後選擇 [角色別名]。
4. 選擇 [建立角色別名]。
5. 輸入角色別名的名稱。

## Example

範例：KvsEdgeAgentRoleAlias

6. 在「角色」下拉式清單中，選取您在中建立的 IAM 角色 [the section called “4. 建立 IAM 角色”](#)。
7. 選擇建立。在下一頁上，您會看到已成功建立角色別名的附註。
8. 搜尋並選取新建立的角色別名。記下角色別名 ARN。您在下一個步驟中對AWS IoT策略需要此功能。

## 步驟 6：建立原AWS IoT則

請遵循下列程序來建立將附加至裝置憑證的AWS IoT原則。這會授予權AWS IoT能的權限，並允許使用憑證假設角色別名。

使用AWS IoT Core原則，您可以控制對資AWS IoT Core料平面的存取。資AWS IoT Core料平面包含可用來執行下列作業的作業：

- Connect 至AWS IoT Core訊息代理程式
- 傳送和接收 MQTT 訊息
- 取得或更新物件的裝置陰影

如需詳細資訊，請參閱 [AWS IoT Core 政策](#)。

使用AWS IoT策略編輯器建立AWS IoT策略

1. 請登入AWS Management Console並開啟AWS IoT Core主控台，網址為 <https://console.aws.amazon.com/iot/>。
2. 在左側導覽列中，選取 [安全性]，然後選擇 [原則]。
3. 選擇建立政策。
4. 輸入政策的名稱。

### Example

策略名稱的一個例子是KvsEdgeAccesslo策略。

5. (選用) 藉由連接標籤作為鍵值組，將中繼資料新增至政策。

如需在 IAM 中使用標籤的詳細資訊，請參閱AWS IoT Core開發人員指南中的[標記AWS IoT資源](#)。

6. 選擇 JSON 標籤。
7. 貼上下列 JSON 政策文件：

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



```
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iot:AssumeRoleWithCertificate"
    ],
    "Resource": "your-role-alias-arn"
}
]
```

#### Note

以您在 [the section called “5. 建立AWS IoT角色別名”](#) 建立之角色別名的 ARN 取 *your-role-alias-arn* 代。

8. 選擇「建立」以儲存作業。

## 步驟 7：創建一個AWS IoT東西並獲得憑據 AWS IoT Core

在這一點上，你已經創建了：

- IAM 許可政策。請參閱 [the section called “3. 建立 IAM 許可政策”](#)。
- IAM 角色，並附加了許可政策。請參閱 [the section called “4. 建立 IAM 角色”](#)。
- IAM AWS IoT 角色的角色別名。請參閱 [the section called “5. 建立AWS IoT角色別名”](#)。
- 目前未附加至任何AWS資源的AWS IoT策略。請參閱 [the section called “6. 建立 AWS IoT 政策”](#)。

若要建立和註冊物AWS IoT件並AWS IoT Core取得存取認證

1. 將裝置註冊為物AWS IoT件，並產生裝置的 X.509 憑證。

- a. 請登入AWS Management Console並開啟AWS IoT Core主控台，網址為 <https://console.aws.amazon.com/iot/>。
- b. 選取適當的「區域」。
- c. 在左側導覽列中，選取 [所有裝置]，然後選擇 [物件]。
- d. 選擇 [建立物件]。
- e. 選取建立單一物件，然後選擇下一步。

#### 1. 步驟 1. 指定物件內容

輸入物件的名稱，然後選擇「下一步」。

#### 2. 步驟 2. 設定裝置憑證

選取「自動產生新憑證 (建議選項)」，然後選擇「下一步」。

#### 3. 步驟 3. 將原則附加至憑證

搜尋您在中建立的權限原則 [the section called “6. 建立 AWS IoT 政策”](#)。

選取原則旁邊的核取方塊，然後選擇 [建立物件]。

- f. 在出現的視窗中，下載下列檔案：

- 裝置憑證。這是 X.509 憑證。
- 公開金鑰檔案
- 私密金鑰檔案
- Amazon 信任服務端點 (RSA 2048 位密鑰：Amazon 根 CA 1)

請記下每個檔案的位置，以便稍後使用。

- g. 選擇完成。在下一頁上，您會看到已成功建立物件的附註。
- h. 將上面下載的文件傳輸到您的AWS IoT東西上，如果尚未存在。

2. 取得您AWS帳戶的認證提供者端點。

## AWS CLI

執行以下命令：

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

## AWS Management Console

在中 [AWS CloudShell](#)，執行下列命令：

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

請記下此資訊，以便稍後進行步驟。

3. 取得您AWS帳戶的裝置資料端點。

## AWS CLI

執行以下命令：

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

## AWS Management Console

請執行下列操作：

1. 請登入AWS Management Console並開啟AWS IoT Core主控台，網址為 <https://console.aws.amazon.com/iot/>。
2. 在左側導覽列中，選取 [設定]。
3. 找到裝置資料端點。

請記下此資訊，以便稍後進行步驟。

4. (選擇性) 確認您的憑證是否已正確產生。

執行下列命令以驗證您的項目是否已正確產生。

```
curl --header "x-amzn-iot-thingname:your-thing-name" \  
  --cert /path/to/certificateID-certificate.pem.crt \  
  --key /path/to/certificateID-private.pem.key \  
  --cacert /path/to/AmazonRootCA1.pem \  
  https://your-credential-provider-endpoint/role-aliases/your-role-alias-name/  
  credentials
```

如需詳細資訊，請參閱[如何使用憑證取得安全性權杖](#)。

## 步驟 8：建置並執行 Amazon Kinesis Video Streams 邊緣代理程式

### 建置和執行 Amazon Kinesis Video Streams 邊緣代理程式

1. 使用tar提供給您的鏈接下載文件。

如果您已填妥 Amazon Kinesis Video Streams 邊緣代理程式興趣表單，請查看您的電子郵件以取得下載連結。如果您尚未填寫表格，請[在此](#)填寫。

2. 驗證總和檢查碼。
3. 提取設備中的二進製文件和 jar。

類型:tar -xvf kvs-edge-agent.tar.gz.

解壓縮後，您的資料夾結構將如下所示：

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

**Note**

發行版本資料夾名稱應該以反映最新二進位版本號碼的方式來設定。例如，1.0.0 版本的資料夾名稱設定為 1.0.0。

## 4. 構建依賴關係 jar。

**Note**

隨附的 jar `kvs-edge-agent.tar.gz` 沒有相依性。請使用下列步驟來建置這些程式庫。

導覽至包含的資料夾 `kvs-edge-agent/pom.xml`。

輸入 `mvn clean package`。

這會產生一個 jar 檔案，其中包含 Amazon Kinesis Video Streams 邊緣代理程式所需的相依性。`kvs-edge-agent/target/libs.jar`

5. `libs.jar` 將放置在包含元件加工品的資料夾中。

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

## 6. 使用先前步驟中的值設定環境變數。下表提供變數的說明。

環境變數名稱	必要	描述
AWS_REGION	是	所使用的區域。  範例：美西 -2
AWS_IOT_CA_CERT	是	用於透過 TLS 與後端服務建立信任之 CA 憑證的檔案路徑。  範例： <code><i>/file/path/to/</i>AmazonRootCA1.pem</code>

環境變數名稱	必要	描述
AWS_IOT_CORE_CERT	是	<p>X.509 憑證的檔案路徑。</p> <p>範 例：<i>/file/path/to/certificateID-certificate</i> .pem.crt</p>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	是	<p>您AWS帳戶的<a href="#">AWS IoT Core 認證端點提供者</a>端點。</p> <p>範例：<i>credential-account-specific-prefix</i> .credentials.iot. <i>aws-region</i> .amazonaws.com</p>
AWS_IOT_CORE_DATA_PLANE_ENDPOINT	是	<p>您AWS帳戶的<a href="#">AWS IoT Core 資料平面端點</a>。</p> <p>範例：<i>data-account-specific-prefix</i> .iot. <i>aws-region</i> .amazonaws.com</p>
AWS_IOT_CORE_PRIVATE_KEY	是	<p>公開/私密 key pair 中使用之 私密金鑰的檔案路徑。如需詳細資訊，請參閱<a href="#">中的金鑰管理 AWS IoT</a>。</p> <p>範 例：<i>/file/path/to/certificateID-private</i> .pem.key</p>

環境變數名稱	必要	描述
AWS_IOT_CORE_ROLE_ALIAS	是	<p>連線時要使用的 AWS IAM 角色指向的角色别名名称AWS IoT Core。</p> <p>範例：kvs-edge-role-alias</p>
AWS_IOT_CORE_THING_NAME	是	<p>執行應用程式之AWS IoT物件的名称。</p> <p>範例：my-edge-device-thing</p>
GST_PLUGIN_PATH	是	<p>指向包含gstkvssink 和IngestorPipelineJNI 平台相關程式庫的資料夾的檔案路徑。讓GStreamer 加載這些插件。如需詳細資訊，請參閱<a href="#">下載、建置和設定 GStreamer</a> 元素。</p> <p>範例：<i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesis.video.KvsEdgeComponent/ EdgeAgent Version /</i></p>

環境變數名稱	必要	描述
LD_LIBRARY_PATH	是	<p>指向包含cproducer和KinesisVideoProducer 平台相關程式庫的目錄的檔案路徑。</p> <p>範例：<i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	否	<p>判斷 Amazon Kinesis Video Streams 邊緣代理程式是否會將任務運作狀態指標張貼到上面Amazon CloudWatch。</p> <p>接受的值：TRUE/FALSE ( 不區分大小寫 )。FALSE如果未提供，則預設為。</p> <p>例如：假</p>



環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_LEVEL	否	<p>記錄 Amazon Kinesis Video Streams 邊緣代理程式輸出的層級。</p> <p>接受的值：</p> <ul style="list-style-type: none"> <li>• OFF</li> <li>• ALL</li> <li>• 致命的</li> <li>• ERROR</li> <li>• WARN</li> <li>• INFO，默認值，如果未提供</li> <li>• DEBUG</li> <li>• TRACE</li> </ul> <p>範例：INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	否	<p>一旦記錄檔達到此大小，就會發生變換。</p> <ul style="list-style-type: none"> <li>• 分鐘:0</li> <li>• 最大值:1 萬</li> <li>• 預設值：20 (如果未提供)</li> <li>• 單位:百萬位元組 (MB)</li> </ul> <p>範例：5</p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	否	<p>指向輸出 Amazon Kinesis Video Streams 邊緣代理程式日誌所在目錄的檔案路徑。 ./log 如果未提供，則預設為。</p> <p>範例：<i>/file/path/</i></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	否	<p>刪除前要保留的轉換記錄檔數目。</p> <ul style="list-style-type: none"> <li>• 分鐘：1</li> <li>• 最大值：1</li> <li>• 預設值：10 (如果未提供)</li> </ul> <p>範例：20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	否	<p>指向記錄媒體的目錄的文件路徑將被寫入。如果未提供，則默認為當前目錄。</p> <p>範例：<i>/file/path/</i></p>
GST_DEBUG	否	<p>指定要輸出的 GStreamer 記錄檔層級。如需詳細資訊，請參閱 <a href="#">GStreamer</a> 文件。</p> <p>範例：0</p>

環境變數名稱	必要	描述
GST_DEBUG_FILE	否	指定 GStreamer 偵錯記錄檔的輸出檔案。如果未設置，調試日誌將輸出到標準錯誤。如需詳細資訊，請參閱 <a href="#">GStreamer</a> 文件。  範例： <code>/tmp/gstreamer-logging.log</code>

## 7. 清除緩存。類型：

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

如需詳細資訊，請參閱 [GStreamer 登錄](#) 文件。

## 8. 準備並執行 java 指令。Amazon Kinesis Video Streams 邊緣代理程式接受下列引數：

爪哇屬性名稱	必要	描述
java.library.path	否	指向包含 gstkvssink 和 IngestorPipelineJNI 從屬資源庫的資料夾的檔案路徑。如果未提供，Amazon Kinesis Video Streams 邊緣代理程式會在目前目錄中搜尋這些代理程式。

### Important

如果 Amazon Kinesis Video Streams 邊緣代理程式找不到這些檔案，將無法正常運作。

爪哇屬性名稱	必要	描述
		範例： <code>/file/path/</code>

要設置這些，添加 `-Djava-property-name=value` 到用於運行 jar 的 java 命令。

例如：

```
java -Djava.library.path=download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion \
--add-opens java.base/jdk.internal.misc=ALL-UNNAMED \
-Dio.netty.tryReflectionSetAccessible=true \
-cp kvs-edge-agent.jar:libs.jar \
com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

#### Important

從相同的目錄中運行上面的 java 命令 `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion`。

9. 使用將組態傳送至應用程式 AWS CLI。
  - a. 建立新檔案、`example-edge-configuration.json`。

將以下程式碼貼到檔案。這是一個範例設定，每天從上午 9:00:00 到 4:59:59 PM 進行記錄 (根據裝置上的系統時間)。AWS IoT 它還每天從晚上 7 點 00 分到晚上 9 點 59 分鐘上傳錄製的媒體。

如需詳細資訊，請參閱 [the section called "StartEdgeConfigurationUpdate"](#)。

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
```

```

    "MediaSourceConfig": {
      "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
      "MediaUriType": "RTSP_URI"
    },
    "ScheduleConfig": {
      "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * *",
      "DurationInSeconds": 3599
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "ScheduleExpression": "0 0 19,20,21 ? * * *",
      "DurationInSeconds": 3599
    }
  },
  "DeletionConfig": {
    "EdgeRetentionInHours": 15,
    "LocalSizeConfig": {
      "MaxLocalMediaSizeInMB": 2800,
      "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
    },
    "DeleteAfterUpload": true
  }
}

```

- b. 若要將檔案傳送至 Amazon Kinesis Video Streams 邊緣代理程式，請在中輸入下列內容：  
AWS CLI

```

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

```

10. 針對 Amazon Kinesis 影片串流邊緣代理程式的每個串流重複上述步驟。

## 步驟 9：(選擇性) 在裝置上安裝 CloudWatch 代理程式

### Note

請注意 [CloudWatch 配額](#)。

依照下列程序安裝並設定 CloudWatch 代理程式，以自動將 Amazon Kinesis Video Streams 邊緣代理程式產生的日誌上傳到 CloudWatch。

如需在裝置上安裝 CloudWatch 代理程式的[程序](#)，請參閱 Amazon CloudWatch 使用者指南。

當系統提示您輸入組態時，請選擇下列其中一個模型組態。

### ⚠ Important

下列組態 `file_path` 中的假設使用預設記錄輸出位置。

使用的檔案路徑假設您正在從以下位置執行 Amazon Kinesis Video Streams 邊緣代理程式：`download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version`。

- 若要將 CloudWatch 代理程式設定為上傳記錄檔並張貼裝置 RAM 和 CPU 度量，請將以下內容貼到組態檔中。

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

```

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

```

- 要僅上傳日誌而不收集設備的 RAM 和 CPU，請使用以下配置：

```

{
    "logs": {
        "logs_collected": {

```

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

## 步驟 10 : (選用) 以原生程序執行 Amazon Kinesis Video Streams 邊緣代理程式

將 Amazon Kinesis Video Streams 邊緣代理程式設定為系統服務。

systemd 是 Linux 設備上的系統和服務管理器。systemd 這是管理程序的建議方式，因為如果應用程式發生錯誤或執行應用程式的裝置中斷電力，它會重新啟動 Amazon Kinesis Video Streams Edge 代理程式。

請執行下列操作：



## 以原生程序的形式執行 Amazon Kinesis Video Streams 邊緣代理程式

1. 在中建立新檔案/etc/systemd/system並命名*aws.kinesisvideo.edge-runtime-agent.service*。

粘貼以下內容：

```
[Unit]
Description=AWS Kinesis Video Streams edge agent
After=network.target
StartLimitBurst=3
StartLimitInterval=30

[Service]
Type=simple
Restart=on-failure
RestartSec=10
WorkingDirectory=/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
Environment="GST_PLUGIN_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion"
Environment="LD_LIBRARY_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib"
...
Environment="AWS_IOT_CORE_DATA_ATS_ENDPOINT=data-account-specific-prefix.iot.aws-
region.amazonaws.com"
ExecStart=/usr/lib/jvm/java-11-amazon-corretto/bin/java --add-opens java.base/
jdk.internal.misc=ALL-UNNAMED -Dio.netty.tryReflectionSetAccessible=true -cp kvs-
edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp

[Install]
WantedBy=multi-user.target
```

如需systemd服務組態檔接受之參數的詳細資訊，請參閱[文件](#)。

### Note

在...位置新增必要的環境變數，如中所指定[the section called “8. 建置和執行 Amazon Kinesis Video Streams 邊緣代理程式”](#)。

2. 重新載入服務檔案以包含新服務。

輸入 `sudo systemctl daemon-reload`。

3. 啟動服務。

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

4. 檢查 Amazon Kinesis Video Streams 邊緣代理程式服務的狀態，以確認服務正在執行。

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

以下是您將看到的輸出範例。

```
aws.kinesisvideo.edge-runtime-agent.service - AWS Kinesis Video Streams edge agent
  Loaded: loaded (/etc/systemd/system/aws.kinesisvideo.edge-runtime-agent.service; disabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-06-08 19:15:02 UTC; 6s ago
  Main PID: 506483 (java)
  Tasks: 23 (limit: 9518)
  Memory: 77.5M
  CPU: 4.214s
  CGroup: /system.slice/aws.kinesisvideo.edge-runtime-agent.service
          ##506483 /usr/lib/jvm/java-11-amazon-corretto/bin/java -cp kvs-edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

5. 檢查日誌是否有任何錯誤。

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

6. 輸入要使 `systemctl --help` 用的管理流程的完整選項清單 `systemctl`。

以下是管理 Amazon Kinesis Video Streams 邊緣代理程式的一些常用命令：

- 若要重新啟動，請鍵入 `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`。
- 若要停止，請鍵入 `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`。
- 要在每次設備重新啟動時自動啟動，請鍵入 `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`。

# 將 Amazon Kinesis Video Streams 邊緣代理程式部署到 AWS IoT Greengrass

請依照下列步驟部署 Amazon Kinesis Video Streams 邊緣代理程式，以 AWS IoT Greengrass 便從 IP 攝影機錄製和上傳媒體。

## 主題

- [第 1 步：創建一個 Ubuntu 的 Amazon EC2 實例](#)
- [步驟 2：在設備上設置 AWS IoT Greengrass V2 核心設備](#)
- [步驟 3：為您的 IP 攝影機 RTSP 網址建立亞馬遜室運動影片串流和 AWS Secrets Manager 資源](#)
- [步驟 4：將權限新增至權杖交換服務 \(TES\) 角色](#)
- [步驟 5：在設備上安裝 AWS IoT Greengrass 密碼管理器組件](#)
- [步驟 6：在裝置上部署 Amazon Kinesis Video Streams 邊緣代理程式 AWS IoT Greengrass 元件](#)
- [步驟 7：\(可選\) 在設備上安裝 AWS IoT Greengrass 日誌管理器組件](#)

## 第 1 步：創建一個 Ubuntu 的 Amazon EC2 實例

請執行以下操作以創建 Ubuntu Amazon EC2 實例。

### 創建一個 Ubuntu 的 Amazon EC2 實例

1. 請登入 AWS Management Console，並在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。

確認已選取適當的「區域」。

2. 選擇 Launch Instance (啟動執行個體)。

完成下列欄位：

- 名稱 — 輸入實例的名稱。
- 應用程序和操作系統映像 (Amazon 機器映像) - 選擇 Ubuntu。
- 執行個體類型 — 選取 t2.large。
- key pair 登入 — 建立您自己的金鑰組。
- 網路設定 — 保留預設值。
- 配置儲存區 — 將磁碟區增加至 256 GiB。

- 進階設定 — 保留預設值。
3. 啟動執行個體並進入 SSH。

請執行下列操作：

1. 在左側導覽中選取「執行個體」，然後選取執行個體 ID。
  2. 選擇右上角的「Connect」。
  3. 選擇 SSH 客戶端，然後按照屏幕上的說明進行操作。
  4. 打開終端並導航到下載的 .pem 文件（可能在中~/Downloads）。
  5. 第一次執行這些程序時，您會收到「無法建立 host (...) 的真實性」訊息。鍵入是。
4. 安裝系統程式庫，以在執行個體上建置 Amazon Kinesis Video Streams 邊緣代理程式。

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

sudo apt-get update

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

#### Important

如果您看到一個畫面，告訴您某些服務需要重新啟動，請按 Enter 鍵以選取 [確定]。

如需詳細資訊，請參閱 [Amazon 網路 11 使用者指南](#)。

## 步驟 2：在設備上設置 AWS IoT Greengrass V2 核心設備

請依照下列程序在 Amazon EC2 執行個體上安裝 AWS IoT Greengrass 核心軟體。

### 設定 AWS IoT Greengrass 核心裝置

1. 請登入以下 AWS Management Console [網址](https://console.aws.amazon.com/iot/)：<https://console.aws.amazon.com/iot/>。

確認已選取適當的「區域」。

2. 在左側導覽列中，選取 Greengrass 裝置、核心裝置。
3. 選擇 [設定一個核心裝置]。
4. 完成畫面上的步驟。

- 步驟 1：註冊一個核心設備。輸入裝置的名稱。
- 步驟 2：新增至物件群組以套用持續部署。選取「無群組」。
- 步驟 3：安裝 Greengrass 核心軟件。選取 Linux。
  - 步驟 3.1：在設備上安裝 Java

Java 是作為的一部分進行安裝[the section called “1. 建立執行個體”](#)。如果您尚未安裝 Java，請返回該步驟。

- 步驟 3.2：將AWS憑據複製到設備上

開啟選bash/zsh項，然後將匯出命令貼到 Amazon EC2 執行個體中。

- 步驟 3.3：執行安裝程式
  1. 在 Ubuntu Amazon EC2 實例中複製並運行下載安裝程序並運行安裝程序命令。

#### Note

[執行安裝程式] 命令會根據您在上一個步驟中選擇的名稱自動更新。

2. 記下所建立的權杖交換服務 (TES) 角色。供稍後使用。

#### Note

默認情況下，創建的角色被稱為格雷格拉斯 V2 TokenExchangeRole。

## 步驟 3：為您的 IP 攝影機 RTSP 網址建立亞馬遜室運動影片串流和AWS Secrets Manager資源

請遵循下列程序來建立中所需的串流和密碼AWS Secrets Manager。請先執行此步驟，因為您需要策略中已建立資源的 ARN。

## 創建 Amazon Kinesis Video Streams

使用AWS Management Console、AWS CLI或 API 建立 Amazon Kinesis Video Streams。

在中AWS Management Console，開啟 [Amazon Kinesis Video Streams 主控台](#)。選擇視頻流在左側導航中。

如需詳細資訊，請參閱 [the section called “建立 Kinesis 影片串流”](#)。

### 在 AWS Secrets Manager 中建立秘密

在中AWS Management Console，開啟主[AWS Secrets Manager](#)控制台。選擇左側導覽列中的 [秘密]。

確認已選取適當的「區域」。

1. 選擇存放新的機密。

a. 步驟 1：選擇秘密類型

- 選取其他機密類型。
- 在「鍵/值對」部分中，添加鍵值對。

索引鍵：MediaURI

#### Note

金鑰必須是MediaURI。這是區分大小寫的。如果輸入錯誤，應用程式將無法正常工作。

值：*Your MediaURI*。

Example

例如：`rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/  
YourCameraMediaURI`。

- b. 步驟 2：設定密碼。給這個秘密起個名字。將其命名為任何你想要的。
- c. 步驟 3：配置旋轉-可選。選擇下一步。
- d. 步驟 4：檢閱。選擇儲存。

2. 如果您的密碼未立即顯示，請選取重新整理按鈕。

3. 為您的 IP 攝像機 RTSP 網址創建資源

選擇您的秘密名稱。記下秘密 ARN。

3. 對要從中流式傳輸的每個媒體 URI 重複此過程。

#### Note

AWS 網路封鎖一些公用 RTSP 來源。您無法從 Amazon EC2 執行個體中存取這些執行個體，或者如果您在連線到 VPN 時未受管執行。

#### Important

您的攝影機 RTSP 網址應該以 h.264 格式串流影片。片段持續時間不得超過中提到的限制 [the section called “製作者 SDK 限制”](#)。

Amazon Kinesis Video Streams 邊緣代理程式僅支援視訊。

運行 `gst-discoverer-1.0 Your RtspUrl` 以確保您的相機可以從設備訪問。

為您建立的所有串流和密碼儲存 ARN。您需要這些來進行下一步。

## 步驟 4：將權限新增至權杖交換服務 (TES) 角色

將權杖交換服務 (TES) 角色授與裝置，該裝置具有查看密碼的權限。這對於 AWS Secrets Manager AWS IoT Greengrass 組件正常工作是必要的。

將權限新增至 TES 角色

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在左側導覽中選擇「角色」，然後搜尋您先前在流 `'b5' '7b` 中建立的 TES 角色。
3. 在 [新增權限] 下拉清單中，選取 [附加原則]
4. 選擇建立政策。
5. 向下捲動並選取 [編輯]。
6. 在政策編輯器中，選擇 JSON 並編輯策略。

以下列項目取代原則：

**Note**

將您在上一個步驟中建立的串流的 ARN 取代 `arn:aws:kinesisvideo:*:*:stream/streamName1/*` 和 `arn:aws:kinesisvideo:*:*:stream/streamName2/*`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:ListStreams"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
      ]
    }
  ]
}
```

7. 在 Add tags (新增標籤) 頁面上，選擇 Next: Review (下一步：檢閱)。
8. 為原則命名，然後選擇 [建立原則]。

策略名稱的範例為 `KvsEdgeAccessPolicy`。

9. 關閉索引標籤，並返回您將原則附加至 TES 角色的索引標籤。

選擇重新整理按鈕，然後搜尋新建立的原則。



選取核取方塊，然後選擇 [附加原則]。

在下一個畫面上，您會看到說明原則已成功附加至角色的附註。

10. 創建並附加另一個策略，這次是為了您的秘密。

以下列項目取代原則：

**Note**

以包含您在中建立之 MediaURI 密碼的 ARN

取 `arn:aws:secretsmanager:*:*:secret:*` 代。 [the section called “3. 為您的 IP 攝像機 RTSP 網址創建資源”](#)

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

11. 建立並附加另一個原則，這次是用於 Amazon CloudWatch 指標。以下列項目取代原則：

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

```
    ]
  }
]
}
```

## 步驟 5：在設備上安裝AWS IoT Greengrass密碼管理器組件

Amazon Kinesis Video Streams 邊緣代理程式需要先在裝置上安裝AWS IoT Greengrass秘密管理員元件。

### 安裝密碼管理員元件

1. 請登入AWS Management Console並開啟AWS IoT Core主控台，[網址為 https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/)。確認已選取適當的「區域」。
2. 在左側導覽中，選擇 Greengrass 裝置、部署。

選擇與我們在中建立的物件具有相同目標的部署[the section called “2. 設定AWS IoT Greengrass核心裝置”](#)。

3. 在右上角的「動作」下拉式清單中，選擇「修改」。

在出現的快顯視窗中，選擇 [修訂部署]。

4. 完成以下各節：
  - 步驟 1：指定目標。選擇下一步。
  - 步驟 2：選擇組件。
    - 確認已選取 AWS.GLEX 元件。請勿解除安裝此元件。
    - 切換僅顯示選取的元件切換並搜尋 aws.greengrass。SecretManager。
    - 選中旁邊的複選框。SecretManager，然後選擇「下一步」。
  - 步驟 3：設定元件。將AWS IoT Greengrass密碼管理員元件設定為從AWS IoT Greengrass環境中下載密碼。

選擇角色。SecretManager元件，然後選擇設定元件。

在出現的畫面中，更新「要合併的組態」方塊中的 AWS Secrets Manager ARN。

**Note**

以您在 [the section called “3. 為您的 IP 攝像機 RTSP 網址創建資源”](#) 建立之密碼的 ARN 取 `arn:aws:secretsmanager:*:*:secret:*` 代。

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

**Note**

`cloudSecrets` 是帶有密鑰的對象列表 `arn`。如需詳細資訊，請參閱 [AWS IoT Greengrass Version 2 發人員指南](#) 中的 [秘密管理員設定](#) 一節。

完成後，請選取 [確認]，然後選擇 [下一步]。

- 步驟 4：設定進階設定。選取下一步。
- 步驟 5：檢閱。選取 Deploy (部署)。

#### 5. 確認 AWS Secrets Manager 元件和權限已正確安裝。

在 Ubuntu 亞馬遜 EC2 執行個體上，輸入 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` 以確認元件是否已收到更新的組態。

#### 6. 檢查 AWS IoT Greengrass 核心日誌。

輸入 `sudo less /greengrass/v2/logs/greengrass.log`。

檢閱部署錯誤。

如果發生錯誤，請修訂部署以移除`aws.greengrass.SecretManager`元件。

鍵入`sudo service greengrass restart`以重新啟動AWS IoT Greengrass核心服務。

如果部署錯誤與遺失權限有關，請檢閱[the section called “4. 將權限新增至 TES 角色”](#)本節以確定TES 角色具有適當的權限。然後，重複此部分。

## 更新密AWS IoT Greengrass碼管理員元件上的密碼

### Important

Sec AWS IoT Greengrass ret Manager 元件只會在部署更新時擷取和快取密碼。

若要更新 Sec AWS IoT Greengrass ret Manager 元件上的密碼，請遵循前面的步驟 1—6，並進行下列變更。

步驟 3：設定元件。將AWS IoT Greengrass密碼管理員元件設定為從AWS IoT Greengrass環境中下載密碼。

選擇角色。SecretManager元件，然後選擇設定元件。

在出現的畫面[""]中，貼上「重設路徑」方塊，然後在「要合併的組態」方塊中更新 AWS Secrets Manager ARN。

如需詳細資訊，請參閱[重設更新](#)。

## 步驟 6：在裝置上部署 Amazon Kinesis Video Streams 邊緣代理程式AWS IoT Greengrass元件

在裝置上部署 Amazon Kinesis Video Streams 邊緣代理程式AWS IoT Greengrass元件

1. 使用提供的鏈接下載tar文件。

如果您已填妥 Amazon Kinesis Video Streams 邊緣代理程式興趣表單，請查看您的電子郵件以取得下載連結。如果您尚未填寫表格，請[在此](#)填寫。

2. 驗證總和檢查碼。
3. 提取設備中的二進製文件和 jar。

類型:tar -xvf kvs-edge-agent.tar.gz.

解壓縮後，您的資料夾結構將如下所示：

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config

kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

#### Note

發行版本資料夾名稱應該以反映最新二進位版本號碼的方式來設定。例如，1.0.0 版本的資料夾名稱設定為 1.0.0。

4. 構建依賴關係 jar。

**Note**

包含在 `kvs-edge-agent.tar.gz` 中的 jar 沒有依賴關係。請使用下列步驟來建置這些程式庫。

導覽至包含的資料夾 `kvs-edge-agent/pom.xml`。

輸入 `mvn clean package`。

這將產生一個 jar 檔案，其中包含 Amazon Kinesis Video Streams 邊緣代理程式所需的相依性。`kvs-edge-agent/target/libs.jar`

- 將 `libs.jar` 放入包含元件加工品的資料夾中。

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

- 「選用」。配置屬性。Amazon Kinesis Video Streams 邊緣代理程式在 AWS IoT Greengrass 模式下接受下列環境變數：

環境變數名稱	必要	描述
AWS_REGION	是	所使用的區域。  範例：美西 -2  AWS IoT Greengrass 核心軟體會自動為您設定這個價值。如需詳細資訊，請參閱開AWS IoT Greengrass Version 2 發人員指南中的 <a href="#">元件環境變數參考</a> 主題。
GST_PLUGIN_PATH	是	指向包含 <code>gstkvssink</code> 和 <code>IngestorPipelineJNI</code> 平台相關程式庫的資料夾的檔案路徑。這可以讓 GStreamer 加載

環境變數名稱	必要	描述
LD_LIBRARY_PATH	是	<p data-bbox="1084 205 1463 338">這些插件。如需詳細資訊，請參閱<a href="#">下載、建置和設定 GStreamer</a> 元素。</p> <p data-bbox="1084 380 1438 758">範例：<code>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgent Version</i> /</code></p> <p data-bbox="1084 800 1500 978">指向包含 <code>cproducer</code> 和 <code>KinesisVideoProducer</code> 平台相關程式庫的目錄的檔案路徑。</p> <p data-bbox="1084 1020 1438 1396">範例：<code>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgent Version</i> /lib/</code></p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	否	<p>判斷 Amazon Kinesis Video Streams 邊緣代理程式是否會將任務運作狀態指標張貼到上面 Amazon CloudWatch。</p> <p>接受的值：TRUE/FALSE ( 不區分大小寫 )。FALSE 如果未提供，則預設為。</p> <p>例如：假</p>
AWS_KVS_EDGE_LOG_LEVEL	否	<p>記錄 Amazon Kinesis Video Streams 邊緣代理程式輸出的層級。</p> <p>接受的值：</p> <ul style="list-style-type: none"> <li>• OFF</li> <li>• ALL</li> <li>• 致命的</li> <li>• ERROR</li> <li>• WARN</li> <li>• INFO，默認值，如果未提供</li> <li>• DEBUG</li> <li>• TRACE</li> </ul> <p>範例：INFO</p>



環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_M AX_FILE_SIZE	否	<p>一旦記錄檔達到此大小，就會發生變換。</p> <ul style="list-style-type: none"> <li>• 分鐘：1</li> <li>• 最大值：1</li> <li>• 預設值：20 (如果未提供)</li> <li>• 單位:百萬位元組 (MB)</li> </ul> <p>範例：5</p>
AWS_KVS_EDGE_LOG_O UTPUT_DIRECTORY	否	<p>指向輸出 Amazon Kinesis Video Streams 邊緣代理程式日誌所在目錄的檔案路徑。 ./log如果未提供，則預設為。</p> <p>範例：<i>/file/path/</i></p>
AWS_KVS_EDGE_LOG_R OLLOVER_COUNT	否	<p>刪除前要保留的轉換記錄檔數目。</p> <ul style="list-style-type: none"> <li>• 分鐘：1</li> <li>• 最大值：1</li> <li>• 預設值：10 (如果未提供)</li> </ul> <p>範例：20</p>
AWS_KVS_EDGE_RECOR DING_DIRECTORY	否	<p>指向記錄媒體的目錄的文件路徑將被寫入。如果未提供，則默認為當前目錄。</p> <p>範例：<i>/file/path/</i></p>

環境變數名稱	必要	描述
GREENGRASS_ROOT_DIRECTORY	否	<p>AWS IoT Greengrass根目錄的檔案路徑。</p> <p>如果未提供，/greengrass/v2/ 則預設為。</p> <p>範例：<i>/file/path/</i></p>
GST_DEBUG	否	<p>指定要輸出的 GStreamer 記錄檔層級。如需詳細資訊，請參閱 <a href="#">GStreamer</a> 文件。</p> <p>範例：0</p>
GST_DEBUG_FILE	否	<p>指定 GStreamer 偵錯記錄檔的輸出檔案。如果未設置，調試日誌將輸出到標準錯誤。如需詳細資訊，請參閱 <a href="#">GStreamer</a> 文件。</p> <p>範例：<i>/tmp/gstreamer-logging.log</i></p>

開啟kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml並修改執行指令碼，以新增任何先前的環境變數。

### Important

確保修改後的運行腳本不包含任何製表符。AWS IoT Greengrass核心軟件將無法讀取配方。

## 7. 部署 Amazon Kinesis Video Streams 邊緣代理程式AWS IoT Greengrass元件。

類型：

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \
  --recipeDir <download location>/kvs-edge-agent/KvsEdgeComponent/recipes/ \
```

```
--artifactDir <download location>/kvs-edge-agent/KvsEdgeComponent/artifacts/ \
--merge "aws.kinesisvideo.KvsEdgeComponent=EdgeAgentVersion"
```

如需其他資訊，請參閱開AWS IoT Greengrass Version 2發人員指南中的下列章節：

- [AWS IoT GreengrassCLI 命令](#)
- [將AWS IoT Greengrass元件部署到裝置](#)

8. 使用將組態傳送至應用程式AWS CLI。
  - a. 建立新檔案、*example-edge-configuration.json*。

將以下程式碼貼到檔案。這是一個範例設定，每天從上午 9:00:00 到 4:59:59 PM 進行記錄 (根據裝置上的系統時間)。AWS IoT它還每天從晚上 7 點 00 分到晚上 9 點 59 分鐘上傳錄製的媒體。

如需詳細資訊，請參閱 [the section called "StartEdgeConfigurationUpdate"](#)。

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
      },
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * *",
        "DurationInSeconds": 3599
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * *",
        "DurationInSeconds": 3599
      }
    },
    "DeletionConfig": {
      "EdgeRetentionInHours": 15,

```

```
        "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": 2800,
            "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
        },
        "DeleteAfterUpload": true
    }
}
```

- b. 在中輸入以下內容，將檔案傳送AWS CLI至 Amazon Kinesis Video Streams 邊緣代理程式：

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

9. 針對 Amazon Kinesis 影片串流邊緣代理程式的每個串流重複上述步驟。

## 步驟 7：（可選）在設備上安裝AWS IoT Greengrass日誌管理器組件

### Note

請注意 [CloudWatch配額](#)。

請遵循下列程序，將 Amazon Kinesis Video Streams 邊緣代理程式日誌設定為 CloudWatch 使用日 AWS IoT Greengrass誌管理員元件自動上傳到。


### 安裝AWS IoT Greengrass日誌管理器組件

1. 確認AWS IoT Greengrass裝置角色具有 [適當的權限](#)。
  - a. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
  - b. 按一下左側導覽列中的「角色」。
  - c. 選擇在中建立的 TES 角色名稱 [the section called "2. 設定AWS IoT Greengrass核心裝置"](#)。如有必要，請使用搜尋列。
  - d. 選取GreengrassV2TokenExchangeRoleAccess策略。
  - e. 選取 JSON 索引標籤，並確認原則如下所示：

```
{
```

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

- f. 如果GreengrassV2TokenExchangeRoleAccess政策不存在，或者缺少某些必要許可，請使用這些許可建立新的 IAM 政策，並將其附加到中建立的 TES 角色[the section called “2. 設定AWS IoT Greengrass核心裝置”](#)。
2. 請登入AWS Management Console並開啟AWS IoT Core主控台，網址為 <https://console.aws.amazon.com/iot/>。確認已選取適當的「區域」。
3. 在左側導覽中，選擇 Greengrass 裝置、部署。  
  
選擇與您在中建立物件具有相同目標的部署[the section called “2. 設定AWS IoT Greengrass核心裝置”](#)。
4. 選取右上角的「動作」，然後選擇「修改」。  
  
在出現的快顯視窗中，選擇 [修訂部署]。
5. 完成以下各節：
  - a. 步驟 1：指定目標。選擇下一步。
  - b. 步驟 2：選擇組件。
    - i. 請驗證是否為 AWS.GLEX 元件和安全性。SecretManager元件仍處於選取狀態。

 Important

請勿解除安裝這些元件。

- ii. 切換僅顯示選取的元件切換並搜尋 aws.greengrass。LogManager。

- iii. 選取「要點選」旁邊的方塊。LogManager，然後選擇「下一步」。
- c. 步驟 3：設定元件。設定日AWS IoT Greengrass誌管理員元件以上傳 Amazon Kinesis Video Streams 邊緣代理程式產生的日誌。

選擇角色。LogManager元件，然後選擇設定元件。

在出現的畫面中，將下列記錄管理員組態貼到 [要合併的組態] 方塊中。

```
{
  "logsUploaderConfiguration": {
    "componentLogsConfigurationMap": {
      "aws.kinesisvideo.KvsEdgeComponent/java_kvs.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "java_kvs.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_edge.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_edge.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvssink.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvssink.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_streams.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_streams.log\\w*"
      }
    }
  },
  "periodicUploadIntervalSec": "1"
}
```

```
}
```

**⚠ Important**

上述組態logFileDirectoryPath中的假設使用預設記錄輸出位置。

**📘 Note**

如需有關記錄檔管理員設定之每個參數的詳細資訊，請參閱AWS IoT Greengrass Version 2開發人員指南的 < [記錄檔管理員](#) > 一節。

完成後，請選取 [確認]，然後選擇 [下一步]。

- d. 步驟 4：設定進階設定。選取下一步。
  - e. 步驟 5：檢閱。選取 Deploy (部署)。
6. 確認AWS記錄檔管理員元件和權限已正確安裝。
  7. 在 Ubuntu Amazon EC2 實例上，鍵入 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` 以驗證該組件接收到更新的配置。
  8. 檢查AWS IoT Greengrass核心日誌。

輸入 `sudo less /greengrass/v2/logs/greengrass.log`。

檢閱部署錯誤。

如果發生錯誤，請修訂部署以移除 `aws.greengrass.LogManager` 元件。

鍵入 `sudo service greengrass restart` 以重新啟動AWS IoT Greengrass核心服務。

如果部署錯誤與遺失權限有關，請檢閱 [the section called “4. 將權限新增至 TES 角色”](#) 以確定 TES 角色具有適當的權限。然後，重複此部分。

## Amazon Kinesis Video Streams 邊緣代理常見問題

以下是 Amazon Kinesis Video Streams 邊緣代理程式服務的一些常見問題。

## Amazon Kinesis Video Streams 邊緣代理程式支援哪些作業系統？

Amazon Kinesis Video Streams 邊緣代理程式目前支援下列作業系統：

Ubuntu

- 22.x
  - AMD64
- 18.x
  - ARM

AL2

- 安茲尼 2
  - AMD64 亞馬遜亞馬遜營養劑:Snowball

## Amazon Kinesis Video Streams 邊緣代理程式是否支援 H.265 媒體？

Amazon Kinesis Video Streams 邊緣代理程式僅支援 H.264 基本串流。

## Amazon Kinesis Video Streams 邊緣代理程式是否在 AL2 中運作？

是。

## 如何在AWS IoT物件或設備中運行多個流？

[the section called “StartEdgeConfigurationUpdate”](#)將另一個發送到相同HubDeviceArn但不同的亞馬遜 Kinesis 視頻流/AWS Secrets ManagerARN。

## 如何在發送**StartEdgeConfigurationUpdate**後進行編輯？

HubDeviceArn使用相同的 Amazon Kinesis Video Streams ARN 發送更新[the section called “StartEdgeConfigurationUpdate”](#)到相同的。當應用程式收到來自 Amazon Kinesis Video Streams 的訊息時，會覆寫該串流先前的組態。然後將發生更改。

## 你有**ScheduleConfigs**什麼常見的例子嗎？

Amazon Kinesis Video Streams 邊緣代理程式會使用其執行所在裝置的系統時間。



描述	ScheduleExpression	DurationInSeconds
全天候錄製，每小時上傳	(## ScheduleConfig)	
每日上午 9 時至下午 4 時 59 分	0 0 9-16 *? *	3599
9:00:00 AM - 4:59:59 PM weekdays	0 9-16# * ###	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599
週末上午 9 時至下午 4 時 59 分	0 0 9-16 ? * SAT,SUN *	3599
週一至晚上十時至晚上十時五十九分	0 0 22,23 ? * MON-FRI *	3599
每日上午 9 時至凌晨 10 時	0 0 9 * * ? *	3600
每日下午 4 時至 5 時 59 分	0 0 16-17 * * ? *	3599

如需更多範例，請參閱 [Quartz 文件](#)。

## 是否有最大串流限制？

Amazon Kinesis Video Streams 邊緣代理程式目前每個裝置的硬性限制為 16 個串流。使用 [the section called "DeleteEdgeConfiguration"](#) API 從裝置刪除串流。使用更新相同串流的設定 [the section called "StartEdgeConfigurationUpdate"](#) 並不會增加裝置的串流計數。

## 如何重新啟動錯誤的工作？

如果發生錯誤，Amazon Kinesis Video Streams 邊緣代理程式將嘗試重新啟動任務。但是，如果發生某些錯誤 (例如組態錯誤)，您必須手動重新啟動工作。

若要判斷哪些工作需要手動重新啟動，請參閱中的FatalError指標[the section called “使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch”](#)。

重新傳送[the section called “StartEdgeConfigurationUpdate”](#)以重新啟動串流的工作。

## 如何監控 Amazon Kinesis Video Streams 邊緣代理程式的運作狀態？

如需更多詳細資訊，請參閱 [the section called “使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch”](#)。

# 透過 VPC 串流影片

此測試版在歐洲（巴黎）地區（歐洲-西部 -3）提供預覽版。要訪問這些組件和我們的入門指南，請[發送電子郵件給我們](#)。

Amazon Kinesis Video Streams VPC 端點服務可讓您透過 Amazon 網路串流和取用影片，而不需要透過公用網際網路傳輸任何資料。

要請求訪問，請[給我們發送](#)以下信息：

- 帳戶 ID
- 串流 ARN
- VPC ID

## Note

我們可能需要長達一周的時間才能將您添加到服務中。

如果您過去未使用過 VPC 端點，請檢閱下列資訊以熟悉此概念：

- [AWS PrivateLink 背景](#)
- [VPC 入門指南](#)

## 其他資訊

一旦您加入測試版，我們將透過電子郵件傳送連結給您，指向您提供有關此功能的其他資訊。

## VPC 端點程序

### 配額

主要配額差異如下：

- 降低所有頻寬 API 的配額 (2 mbps)：

- PutMedia
- GetMedia
- GetMediaForFragmentList
- 每位客戶允許 10 個串流

## 建立端點

列出允許後，您將收到 Amazon Kinesis Video Streams 的 VPC 端點服務名稱。它看起來像 `com.amazonaws.region.kinesisvideo`。

使用 Amazon 虛擬私人雲端主控台或 () 為 [Amazon Kinesis Video Streams 建立介面](#) VPC 人雲端端點。AWS Command Line Interface AWS CLI

在中AWS CLI，鍵入下列內容：

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

### Important

VPC 內的流量將使用私有 DNS 透過端點進行路由。如果您未啟用此功能，則需要實作您自己的 DNS 邏輯。如需私有 DNS 的詳細資訊，請參閱[AWS PrivateLink文件](#)。

如需AWS CLI選項的詳細資訊，請參閱[create-vpc-endpoint](#)。

## 控制端點的存取

您可以將端點政策附加到 VPC 端點，以控制對 Amazon Kinesis Video Streams 的存取。此政策會指定下列資訊：

- 可以執行動作的主參與者、
- 可以執行的動作，以及
- 可以執行動作的資源。

如需詳細資訊，請參閱AWS PrivateLink指南中的[使用端點策略控制使用 VPC 端點對服務的存取](#)。

以下是適用於 Amazon Kinesis Video Streams 的端點政策範例。連接至端點時，此策略會拒絕存取所有資源上所有主參與者所列出的PutMedia動作。

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

## 室壁運動影片串流中的影像

您可以使用 Amazon Kinesis 影片串流 API 和開發套件，協助您從影片串流擷取影像。您可以將這些影像用於增強的播放應用程式，例如縮圖或增強型拖曳，或用於機器學習管線。Kinesis Video Streams 透過 API 提供隨選影像擷取，或從擷取影片中的中繼資料標籤自動擷取影像。

如需使用 Kinesis 影片串流受管理映像支援的相關資訊，請參閱：

- [按需產生影像 \(GetImages\)](#)-此 API 可讓客戶從 Kinesis 影片串流中儲存的影片擷取單一或多個影像。
- [自動影像產生 \(S3 交付\)](#) -將 Kinesis Video Streams 設定為根據上傳影片中的標籤即時從視訊資料擷取影像，並將映像傳送至客戶指定的 S3 儲存貯體。

### 主題

- [GetImages 入門](#)
- [開始使用亞馬遜 S3 交付](#)

## GetImages 入門

對映像的受管理支援提供完全受控的方式，可從 Kinesis Video Streams 中串流和儲存的視訊資料取得影像。您可以使用影像執行機器學習 (ML) 工作負載，例如人員、寵物或車輛偵測。影像也可用來新增互動元素以進行播放，例如動作事件的影像預覽和拖曳視訊片段。

有關的更多信息 Get Images 功能，請參閱 [GetImages](#) 在亞馬遜室壁運動影片串流封存媒體 API 參考指南。

## 開始使用亞馬遜 S3 交付

目前，客戶會執行並管理自己的影像轉碼管道，以建立用於各種用途的映像，例如清洗、影像預覽、在映像上執行 ML 模型等。Kinesis 視訊串流提供轉碼和交付影像的功能。Kinesis Video Streams 會根據標籤即時從影片資料中自動擷取影像，並將映像傳送至客戶指定的 S3 儲存貯體。

## UpdateImageGenerationConfiguration

若要設定 Kinesis 影片串流，以便向 Amazon S3 產生影像：

1. 創建一個儲存貯體根據使用新 API 在 SDK 中添加的標籤生成圖像。請注意S3 類型更新串流的映像產生組態時，下一個步驟是必要的。
2. 建立名為的 JSON 檔案update-image-generation-input.json以下內容作為輸入。

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

您可以使用AWS CLI以呼叫[UpdateImageGenerationConfiguration](#)用於新增先前建立的 Amazon S3 ARN 的 API 操作，並將狀態變更為ENABLED。

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

請求：

```
UpdateImageGenerationConfiguration HTTP/1.1
```

```
Method: 'POST'
```

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

回應：

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

#### Note

更新映像產生組態後，至少需要 1 分鐘才能啟動映像產生工作流程。調用之前至少等待 1 分鐘PutMedia在更新調用之後。



## DescribeImageGenerationConfiguration

若要檢視已針對串流設定的映像產生設定，客戶可以建立DescribeImageGenerationConfiguration請求，如下。

要求:

```
DescribeImageGenerationConfiguration HTTP/1.1
```

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

回應 :

```
HTTP/1.1 200
Content-type: application/json
Body: {
  ImageGenerationConfiguration : {
    Status: 'Enum',
    ImageSelectorType: 'Enum', // SERVER_TIMESTAMP | PRODUCER_TIMESTAMP
    DestinationConfig: {
      DestinationRegion: 'String'
      Uri: 'string',
    },
    SamplingInterval: 'Number',
    Format: 'Enum',
    FormatConfig: {
      'String': 'String',
    },
    WidthPixels: 'Number',
    HeightPixels: 'Number'
  }
}
```

若要進一步瞭解DescribeImageGenerationConfiguration功能，請參閱[DescribeImageGenerationConfiguration](#) 在亞馬遜 Kinesis 視頻流開發人員指南。

## 製作人 MKV 標籤

您可以透過在 SDK 中公開 API 作業，使用 Kinesis 視訊串流製作工具開發套件來標記感興趣的特定片段。如需標籤範例，請參閱[這段代碼](#)。調用此 API 時，SDK 將添加一組預定義的 MKV 標籤以及片段數據。Kinesis 視訊串流會辨識這些特殊的 MKV 標籤，並根據該串流的影像處理設定啟動影像產生工作流程。

與 Amazon S3 影像產生標籤一起提供的任何片段中繼資料都會儲存為 Amazon S3 中繼資料。

### 製作人 MKV 標籤的語法

```
|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger image generation for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION

| // OPTIONAL: S3 prefix which will be set as prefix for generated image.
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_PREFIX
| + String: image_prefix_in_s3 // 256 bytes max m

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

## 在製作者 SDK 中使用新增中繼資料標PutEventMetaData

該PutEventMetaData函數附加與事件相關聯的 MKV 文件。PutEventMetaData需要兩個參數。第一個參數是一個事件，其值來自STREAM\_EVENT\_TYPE枚舉。第二個參數，[pStreamEventMetadata](#)，是選用的，可用來包含其他中繼資料做為鍵值配對。可以新增的中繼資料的五個索引鍵值配對有限制。

## 限制

下表列出與中繼資料標籤相關聯的限制。如果可調整中繼資料標籤限制，您可以透過帳戶管理員要求提高。

限制	最大值	可調整
影像字首長度	256	否
可選中繼資料金鑰長	128	否
可選中繼資料值長度	256	否
可選中繼資料的最大數目	10	是

## S3 物件中繼資料

根據預設，Kinesis 影片串流會設定片段編號、製片人，以及伺服器時戳以 Amazon S3 物件中繼資料形式產生的影像。如果 MKV 標籤中指定了任何其他片段資料，這些標籤也會新增至 Amazon S3 物件中繼資料。下列範例顯示 Amazon S3 物件中繼資料的正確語法。

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

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

## S3 物件路徑 (影像)

下列清單顯示物件路徑的正確格式，並說明路徑中的每個元素。

格式:

*ImagePrefix\_## ID\_StreamName\_ImageTimecode\_###.####*

1. ImagePrefix - 的價值 AWS\_KINESISVIDEO\_IMAGE\_PREFIX。
2. AccountID - 用來建立串流的帳戶 ID。
3. StreamName - 產生影像的串流名稱。
4. ImageTimecode - 產生影像之片段中的 Epoch 時間碼。
5. RandomID - 隨機指南。
6. file-extension - 根據要求的圖像格式提供 JPG 或 PNG 格式。

## 亞馬遜 S3 URI 建議，以防止節流

如果您將數千個映像寫入 Amazon S3，則存在節流的風險。如需詳細資訊，請參閱 [S3 前綴放置請求限制](#)。

Amazon S3 前綴以每秒 3,500 個 PUT 請求的 PUT 限制開始，並會隨著時間逐漸增加唯一前綴。避免使用日期和時間作為 Amazon S3 前綴。時間編碼的資料將一次影響一個前綴，並且也會定期更改，使之前的前綴縮放無效。為了實現更快、一致的 Amazon S3 擴展，我們建議在 Amazon S3 目的地 URI 新增一個隨機前綴，例如十六進位代碼或 UUID。例如，十六進位程式碼前置字元會自然地將您的請求隨機分割為 16 個不同的前置字元 (每個唯一十六進位字元的前置字元)，在 Amazon S3 自動調整規模後，每秒允許 56,000 個 PUT 請求。

# Kinesis Video Streams 中的通知

當媒體片段可供使用時，Kinesis Video Streams 會使用 Amazon Simple Notification Service (Amazon SNS) 通知通知客戶。下列主題說明如何開始使用通知。

## UpdateNotificationConfiguration

使用此 API 作業可更新串流的通知資訊。如需有關UpdateNotificationConfiguration此功能的詳細資訊，請參閱 Amazon Kinesis Video Streams 開發人員指南[UpdateNotificationConfiguration](#)中的。

### Note

更新通知組態後，至少需要一分鐘的時間來啟動通知。請至少等待一分鐘，然後再PutMedia後再呼叫更新呼叫。

## DescribeNotificationConfiguration

使用此 API 描述連接到串流的通知設定。如需有關DescribeNotificationConfiguration此功能的詳細資訊，請參閱 Amazon Kinesis Video Streams 開發人員指南[DescribeNotificationConfiguration](#)中的。

## 製作人 MKV 標籤

您可以透過在 SDK 中公開發 API 作業，使用 Kinesis Video Streams 製作工具開發套件來標記感興趣的特定片段。在此代碼[部分中查看其工作原理的示例](#)。調用此 API 時，SDK 將添加一組預定義的 MKV 標籤以及片段數據。Kinesis Video Streams 會辨識這些特殊的 MKV 標籤，並針對已標記的片段啟動通知。

隨著通知 MKV 標籤提供的任何片段中繼資料將作為 Amazon SNS 主題承載的一部分發佈。

## 製作人 MKV 標籤的語法

```
|+ Tags  
| + Tag
```

```

| // MANDATORY: Predefined MKV tag to trigger the notification for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_NOTIFICATION
| + String
| // OPTIONAL: Key value pairs that will be sent as part of the Notification payload
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String:CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes

```

## MKV 標籤限制

下表列出與中繼資料標籤相關聯的限制。如果可調整中繼資料標籤限制，您可以透過帳戶管理員要求提高。

限制	最大值	可調整
可選中繼資料金鑰長	128	否
可選中繼資料值長度	256	否
可選中繼資料的最大數目	10	是

## Amazon SNS 主題承載

透過先前工作流程啟動的任何通知都會傳遞 Amazon SNS 主題承載，如下列範例所示。此範例是 Amazon SNS 訊息，在使用來自 Amazon Simple Queue Service (Amazon SQS) 佇列的通知資料後發生。

```

{
  "Type" : "Notification",
  "MessageId" : Message ID,
  "TopicArn" : SNS ARN,
  "Subject" : "Kinesis Video Streams Notification",
  "Message" : "{\"StreamArn\":Stream Arn,\"FragmentNumber\":Fragment Number,
  \"FragmentStartProducerTimestamp\":FragmentStartProducerTimestamp,
```

```

        \ "FragmentStartServerTimestamp\" : FragmentStartServerTimestamp ,
    \ "NotificationType\" : \ "PERSISTED\" , \ "NotificationPayload\" : { \ CUSTOM_KEY_1 :
    \ CUSTOM_VALUE_1 ,
        \ CUSTOM_KEY_2 : \ CUSTOM_VALUE_2 } } " ,
    "Timestamp" : "2022-04-25T18:36:29.194Z" ,
    "SignatureVersion" : Signature Version ,
    "Signature" : Signature ,
    "SigningCertURL" : Signing Cert URL ,
    "UnsubscribeURL" : Unsubscribe URL
}

```

```

Subject: "Kinesis Video Streams Notification"
Message:
{
    "StreamArn" : Stream Arn ,
    "FragmentNumber" : Fragment Number ,
    "FragmentStartProducerTimestamp" : Fragment Start Producer Timestamp ,
    "FragmentStartServerTimestamp" : Fragment Start Server Timestamp ,
    "NotificationType" : "PERSISTED" ,
    "NotificationPayload" : {
        CUSTOM_KEY_1 : CUSTOM_VALUE_1 ,
        CUSTOM_KEY_2 : CUSTOM_VALUE_2
    }
}

```

## 檢視您的 Amazon SNS 訊息

您無法直接從 Amazon SNS 主題讀取訊息，因為沒有 API 可以這麼做。若要檢視訊息，請訂閱 SQS 佇列以取得 SNS 主題，或選擇任何其他 [Amazon SNS 支援的目的地](#)。不過，檢視訊息最有效的選項是使用 Amazon SQS。

使用 Amazon SQS 查看您的 Amazon SNS 消息

1. 建立 [Amazon SQS 佇列](#)。
2. 從中 AWS Management Console 開啟設定為目的地的 Amazon SNS 主題 NotificationConfiguration。
3. 選擇「建立訂閱」，然後選擇在第一個步驟中建立的 Amazon SQS 佇列。
4. 在啟用通知配置並將通知 MKV 標籤添加到片段的情況下運行 PutMedia 會話。
5. 在 Amazon SQS 主控台中選擇 Amazon SQS 佇列，然後選取 Amazon SQS 佇列的傳送和接收訊息。

6. 輪詢訊息。此命令應該顯示PutMedia會話生成的所有通知。如需輪詢的相關資訊，請參閱 [Amazon SQS 短輪詢和長輪詢](#)。



# Amazon Kinesis Video Streams 中的安全性

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。

本文件可協助您瞭解如何在使用 Kinesis Video Streams 時套用共同的責任模型。下列主題說明如何設定 Kinesis Video Streams 以符合安全性和合規性目標。您也將學習如何使用其他可 AWS 協助您監控和保護 Kinesis Video Streams 資源的服務。

## 主題

- [Kinesis Video Streams 中的資料保護](#)
- [使用 IAM 控制對 Kinesis Video Streams 資源的存取](#)
- [使用以下方式控制 Kinesis Video Streams 資源的存取 AWS IoT](#)
- [監控 Amazon Kinesis Video Streams](#)
- [Amazon Kinesis Video Streams 的合規驗證](#)
- [Amazon Kinesis Video Streams 的彈性](#)
- [Kinesis Video Streams 中的基礎架構安全性](#)
- [Kinesis Video Streams 的安全性最佳做法](#)

## Kinesis Video Streams 中的資料保護

您可以使用 () 金鑰使用伺服器端加密 AWS Key Management Service (SSE AWS KMS)，方法是加密 Amazon Kinesis Video Streams 中的靜態資料，以符合嚴格的資料管理需求。

## 主題

- [什麼是 Kinesis Video Streams 的伺服器端加密？](#)
- [成本、區域和效能考量](#)
- [如何開始使用伺服器端加密？](#)
- [建立和使用使用者產生的 KMS 金鑰](#)
- [使用使用者產生的 KMS 金鑰的權限](#)

## 什麼是 Kinesis Video Streams 的伺服器端加密？

伺服器端加密是 Kinesis Video Streams 中的一項功能，可在資料處於靜態狀態之前，使用您指定的資料自動加密。AWS KMS 在將資料寫入 Kinesis Video Streams 儲存層之前，會先加密資料，並在從儲存體擷取資料後解密資料。因此，Kinesis Video Streams 服務中的靜態資料一律會加密。

透過伺服器端加密，Kinesis 視訊串流製作者和消費者不需要管理 KMS 金鑰或加密作業。如果啟用資料保留，您的資料會在進入和離開 Kinesis Video Streams 時自動加密，因此您的靜態資料會加密。AWS KMS 提供伺服器端加密功能使用的所有金鑰。AWS KMS 簡化 Kinesis Video Streams 的 KMS 金鑰的使用 AWS，而該 Kinesis Video Streams 由匯入服務的使用者指定金 AWS KMS 鑰所管理。  
AWS KMS

## 成本、區域和效能考量

套用伺服器端加密時，您需要支付 AWS KMS API 使用量和金鑰成本。與自訂金 AWS KMS 鑰不同，(Default) `aws/kinesis-videoKMS` 金鑰是免費提供的。不過，您仍然必須支付 Kinesis Video Streams 代表您產生的 API 使用費用。

API 使用費用適用於每個 KMS 金鑰，包括自訂金鑰。AWS KMS 成本會隨著您在資料生產者和取用者上使用的使用者認證數量而調整，因為每個使用者憑證都需要唯一的 API 呼叫 AWS KMS。

以下依據資源說明各項成本：

### 鍵

- 由 AWS (別名 `=aws/kinesis-video`) 管理的 Kinesis Video Streams 的 KMS 金鑰不收取任何費用。
- 使用者產生的 KMS 金鑰需 AWS KMS key 支付費用。如需詳細資訊，請參閱 [AWS Key Management Service 定價](#)。

## AWS KMS API 使用情況

產生新資料加密金鑰或擷取現有加密金鑰的 API 要求會隨流量增加而增加，且需要支付 AWS KMS 使用費用。如需詳細資訊，請參閱[AWS Key Management Service 定價：用量](#)。

即使保留設定為 0 (無保留)，Kinesis Video Streams 也會產生金鑰要求。

## 依區域提供伺服器端加密

Kinesis 影片串流可在所有提供 Kinesis 影片串流的伺服器端加密使用。AWS 區域

## 如何開始使用伺服器端加密？

Kinesis Video Streams 上一律啟用伺服器端加密。如果在建立串流時未指定使用者提供的金鑰，則會使用預設金鑰 (由 Kinesis Video Streams 提供)。

建立 Kinesis 視訊串流時，必須將使用者提供的 KMS 金鑰指派給該金鑰。您之後無法使用 [UpdateStream](#) API 將不同的金鑰指派給串流。

您可以透過兩種方式將使用者提供的 KMS 金鑰指派給 Kinesis 視訊串流：

- 在中建立 Kinesis 視訊串流時 AWS Management Console，請在 [建立新視訊串流] 頁面的 [加密] 索引標籤中指定 KMS 金鑰。
- 使用 [CreateStream](#) API 建立 Kinesis 視訊串流時，請在 `KmsKeyId` 參數中指定金鑰識別碼。

## 建立和使用使用者產生的 KMS 金鑰

本節說明如何建立和使用您自己的 KMS 金鑰，而不是使用 Amazon Kinesis Video Streams 管理的金鑰。

### 建立使用者產生 KMS 金鑰

如需如何建立自己金鑰的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [建立金鑰](#)。為帳戶建立金鑰後，Kinesis Video Streams 服務會在 KMS 主金鑰清單中傳回這些金鑰。

### 使用用戶產生的 KMS 金鑰

將正確的權限套用至您的取用者、生產者和管理員之後，您可以在自己 AWS 帳戶 或其他金鑰中使用自訂 KMS 金鑰 AWS 帳戶。您帳戶中的所有 KMS 金鑰都會顯示在主控台的 KMS 主金鑰清單中。

若要使用位於其他帳戶的自訂 KMS 金鑰，您必須擁有使用這些金鑰的權限。您也必須使用 CreateStream API 建立串流。您無法在主控台中建立的串流中使用來自不同帳戶的 KMS 金鑰。

#### Note

在執行 PutMedia 或 GetMedia 作業之前，不會存取 KMS 金鑰。如此會產生下列結果：

- 如果您指定的索引鍵不存在，CreateStream 作業會成功，但串流上 PutMedia 的作 GetMedia 業會失敗。
- 如果您使用提供的 key (aws/kinesis-video)，則在執行第一個 PutMedia 或 GetMedia 操作之前，密鑰不會出現在您的帳戶中。

## 使用使用者產生的 KMS 金鑰的權限

在使用者產生的 KMS 金鑰上使用伺服器端加密之前，您必須先設定 KMS 金鑰原則，以允許串流加密以及串流記錄的加密和解密。如需有關 AWS KMS 權限的範例和詳細資訊，請參閱 [AWS KMS API 權限：動作和資源參考](#)。

#### Note

使用預設服務金鑰進行加密不需要申請自訂 IAM 許可。

在使用使用者產生的 KMS 金鑰之前，請確認 Kinesis 視訊串流產生者和取用者 (IAM 主體) 是主金鑰政策中的 AWS KMS 使用者。否則串流的寫入及讀取將會失敗，最終可能造成資料遺失、延遲處理或應用程式故障。您可使用 IAM 政策管理 KMS 金鑰的許可。如需詳細資訊，請參閱 [搭配使用 IAM 政策 AWS KMS](#)。

## 示例生產者權限

您的 Kinesis 視訊串流製作者必須擁有下列 kms:GenerateDataKey 權限：

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

```
    "Action": [
      "kms:GenerateDataKey"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis-video:PutMedia",
    ],
    "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
  }
]
```

## 示例消費者權限

您的 Kinesis 視訊串流消費者必須擁有下列kms:Decrypt權限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesis-video:GetMedia",
      ],
      "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
    }
  ]
}
```

# 使用 IAM 控制對 Kinesis Video Streams 資源的存取

您可以將 AWS Identity and Access Management (IAM) 與 Amazon Kinesis Video Streams 搭配使用，控制組織中的使用者是否可以使用特定的 Kinesis 影片串流 API 操作執行任務，以及他們是否可以使用特定 AWS 資源。

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](#)

## 目錄

- [政策語法](#)
- [室壁運動影片串流的動作](#)
- [Kinesis Video Streams 的 Amazon 資源名稱 \(ARN\)](#)
- [授與其他 IAM 帳戶存取 Kinesis 影片串流的存取權](#)
- [Kinesis Video Streams 的範例政策](#)

## 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：

- 效果 — 效果可以是Allow或Deny。根據預設，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。
- 動作 — 動作是您授與或拒絕權限的特定 API 動作。
- 資源 — 受動作影響的資源。若要在陳述式中指定資源，您必須使用它的 Amazon Resource Name (ARN)。
- 條件 — 條件是選擇性的。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，建議您使用 [IAM 政策產生器](#) 和 [IAM 政策模擬器](#)。

## 室壁運動影片串流的動作

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。對於 Kinesis Video Streams，請使用下列前置詞以及 API 動作的名稱：kinesisvideo:例如：kinesisvideo:CreateStream、kinesisvideo:ListStreams 和 kinesisvideo:DescribeStream。

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 Amazon 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:stream/stream-name/code
```

例如：

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

您可以使[DescribeStream](#)用獲取流的 ARN。

## 授與其他 IAM 帳戶存取 Kinesis 影片串流的存取權

您可能需要授與其他 IAM 帳戶的權限，才能在 Kinesis 影片串流上執行操作。以下概述說明將視訊串流存取權跨帳戶授與的一般步驟：

1. 取得授予執行串流操作許可的 12 位數帳戶 ID (例如，111111111111)。
2. 在擁有串流的帳戶上建立受管政策，讓您可以授予想要的存取層級。如需 Kinesis Video Streams 資源的範例原則，請參閱[範例政策](#)下一節。
3. 建立角色，並指定您要授與權限的帳戶。然後附加您在上一個步驟中建立的策略。
4. 建立受管理的原則，允許AssumeRole對您在上一個步驟中建立的角色執行動作。例如，此角色看起來類似如下：

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

有關授予跨帳戶存取權的 step-by-step 指示，請參閱[跨 AWS 帳戶使用 IAM 角色委派存取權](#)。

## Kinesis Video Streams 的範例政策

下列範例政策示範如何控制使用者對 Kinesis Video Streams 的存取權

Example 1: 允許使用者從任何 Kinesis 視訊串流取得資料

此原則可讓使用者或群組對任何 Kinesis 視訊串流執行GetMediaListStreams、  
和ListTagsForStream作業。DescribeStream GetDataEndpoint此政策適用於可以取得任何視訊串流資料的使用者。



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

### Example 2: 允許使用者建立 Kinesis 視訊串流並將資料寫入其中

此政策可讓使用者或群組執行 `CreateStream` 和 `PutMedia` 操作。此政策適用於可以建立視訊串流並傳入資料的安全攝影機。

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

### Example 3: 允許使用者完整存取所有 Kinesis Video Streams 資源

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：允許使用者將資料寫入特定 Kinesis 視訊串流

此政策可讓使用者或群組將資料寫入特定的視訊串流。此政策適用於可將資料傳送到單一串流的裝置。

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

## 使用以下方式控制 Kinesis Video Streams 資源的存取 AWS IoT

本節說明如何讓裝置 (例如攝影機) 僅傳送音訊和視訊資料至一個特定的 Kinesis 視訊串流。您可以使用 AWS IoT 登入資料提供者和 AWS Identity and Access Management (IAM) 角色來執行此操作。

裝置可以使用 X.509 憑證來連線到 AWS IoT 使用 TLS 相互驗證通訊協定。其他 AWS 服務 (例如 Kinesis Video Streams) 不支援 AWS 憑證型驗證，但可以使用「AWS 簽名版本 4」格式的認證來呼叫。簽名版本 4 演算法通常要求呼叫者具有存取金鑰 ID 和秘密存取金鑰。AWS IoT 擁有認證提供者，可讓您使用內建的 X.509 憑證作為唯一裝置身分識別來驗證 AWS 要求 (例如，對 Kinesis Video Streams 的要求)。這樣就不需要在設備上存儲訪問密鑰 ID 和秘密訪問密鑰。

認證提供者會使用 X.509 憑證來驗證用戶端 (在此情況下，即在您要將資料傳送至視訊串流的攝影機上執行的 Kinesis Video Streams SDK)，並發出臨時、有限權限的安全性權杖。您可以使用權杖來簽署和驗證任何 AWS 要求 (在本例中為 Kinesis Video Streams 的呼叫)。如需詳細資訊，請參閱[授權直接呼叫 AWS 服務](#)。

這種驗證攝影機對 Kinesis Video Streams 的請求的方式需要您建立和設定 IAM 角色，並將適當的 IAM 政策附加到該角色，以便 AWS IoT 登入資料提供者可以代表您擔任該角色。

如需有關的詳細資訊 AWS IoT，請參閱[AWS IoT Core 文件](#)。如需 IAM 的詳細資訊，請參閱 [AWS Identity and Access Management \(IAM\)](#)。

## 主題

- [AWS IoT ThingName 作為流名稱](#)
- [AWS IoT CertificateId 作為流名稱](#)
- [使用 AWS IoT 認證串流至硬式編碼串流名稱](#)

## AWS IoT ThingName 作為流名稱

### 主題

- [步驟 1：建立 AWS IoT 物件類型和物 AWS IoT 件](#)
- [步驟 2：建立要承擔的 IAM 角色 AWS IoT](#)
- [步驟 3：建立並設定 X.509 憑證](#)
- [步驟 4：使用 Kinesis 視訊串流測試 AWS IoT 憑證](#)
- [步驟 5：在攝影機的檔案系統上部署 AWS IoT 憑證和認證，並將資料串流至視訊串流](#)

### 步驟 1：建立 AWS IoT 物件類型和物 AWS IoT 件

在中 AWS IoT，物件是特定裝置或邏輯實體的表示。在此情況下，物 AWS IoT 件代表您要設定資源層級存取控制的 Kinesis 視訊串流。若要建立物件，首先，您必須建立 AWS IoT 物件類型。您可以使用 AWS IoT 物類型來儲存與相同物件類型相關聯之所有物件通用的描述和組態資訊。

1. 下列範例命令會建立物件類型 `kvs_example_camera`：

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

2. 此範例命令會建立 `kvs_example_camera_stream` 物件類型的 `kvs_example_camera` 物件：

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

## 步驟 2：建立要承擔的 IAM 角色 AWS IoT

IAM 角色與使用者類似，因為角色是具有許可政策的 AWS 身分識別，可決定身分可以執行和不能在其中執行的作業 AWS。需要角色的任何人都可以擔任該角色。當您擔任角色時，其會為您的角色工作階段提供臨時安全性登入資料。

您在此步驟中建立的角色可假設在從用 AWS IoT 戶端執行認證授權要求時，從安全性 Token 服務 (STS) 取得暫時認證。在這種情況下，用戶端是在您的攝影機上執行的 Kinesis 視訊串流 SDK。

請執行下列步驟，來建立並設定此 IAM 角色：

### 1. 建立 IAM 角色。

以下範例命令會建立名為 KVSCameraCertificateBasedIAMRole 的 IAM 角色：

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

您可以針對 iam-policy-document.json 使用下列信任政策 JSON：

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

2. 接下來，將許可政策附加到您先前建立的 IAM 角色。此權限原則允許 AWS 資源的選擇性存取控制 (支援作業的子集)。在這種情況下，AWS 資源是您希望攝像機發送數據的視頻流。換言之，一旦所有設定步驟完成，此相機將就只能將資料傳送至此視訊串流。

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

您可以針對 `iam-permission-document.json` 使用下列身分與存取權管理政策 JSON：

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

請注意，此原則只會針對預留位置 (`${憑證s-iot:}`) 所指定的視訊串流 (AWS 資源) 授權指定的動作。`ThingName` 當 AWS IoT 認證提供者在要求中傳送視訊串流名稱 `ThingName` 時，此預留位置會採用 AWS IoT thing 屬性的值。

3. 接下來，為您的 IAM 角色建立角色別名。角色別名是指向 IAM 角色的替代資料模型。AWS IoT 登入資料提供者請求必須包含一個角色別名，以指出從 STS 取得臨時登入資料時要假設哪個 IAM 角色。

以下範例命令會建立名為 `KvsCameraIoTRoleAlias` 的角色別名。

```
aws --profile default iot create-role-alias --role-alias KvsCameraIoTRoleAlias --
role-arn $(jq --raw-output '.Role.Arn' iam-role.json) --credential-duration-seconds
3600 > iot-role-alias.json
```

4. 現在，您可以使用角色別名建立原則，讓 AWS IoT 憑證承擔角色 (附加後) 的角色。

下列範例命令會建立已 AWS IoT 呼叫的原則 `KvsCameraIoTPolicy`。

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

您可以使用以下命令來創建 `iot-policy-document` JSON 文檔的 `.json`：

```
cat > iot-policy-document.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "$(jq --raw-output '.roleAliasArn' iot-role-alias.json)"
    }
  ]
}
EOF
```

### 步驟 3：建立並設定 X.509 憑證

裝置 (您的視訊串流) 之間的通訊，並 AWS IoT 透過使用 X.509 憑證受到保護。

1. 建立您必須附加先前建立之原則 AWS IoT 的憑證。

```
aws --profile default iot create-keys-and-certificate --set-as-active --
certificate-pem-outfile certificate.pem --public-key-outfile public.pem.key --
private-key-outfile private.pem.key > certificate
```

2. 將 (先前 KvsCameraIoTPolicy 建立 AWS IoT) 的原則附加至此憑證。

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

3. 將您的 AWS IoT 東西 ( kvs\_example\_camera\_stream ) 附加到剛剛創建的證書上：

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

4. 若要透過 AWS IoT 認證提供者授權要求，您需要 AWS IoT 憑證端點，這對您的 AWS 帳戶 ID 來說是唯一的。您可以使用以下命令來獲取 AWS IoT 憑據端點。

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

5. 除了先前建立的 X.509 憑證之外，您還必須擁有 CA 憑證，才能透過 TLS 與後端服務建立信任。您可以使用下列命令來取得 CA 憑證：

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

## 步驟 4：使用 Kinesis 視訊串流測試 AWS IoT 憑證

現在，您可以測試到目前為止設定的 AWS IoT 認證。

1. 首先，建立您要用來測試此組態的 Kinesis 視訊串流。

### Important

使用與您在上一個步驟 (kvs\_example\_camera\_stream) 中建立的 AWS IoT 物件名稱相同的名稱建立視訊串流。

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

2. 接下來，調用 AWS IoT 憑據提供程序以獲取臨時憑據：

```
curl --silent -H "x-amzn-iot-thingname:kvs_example_camera_stream" --cert
certificate.pem --key private.pem.key https://IOT_GET_CREDENTIAL_ENDPOINT/role-
aliases/KvsCameraIoTRoleAlias/credentials --cacert ./cacert.pem > token.json
```

### Note

您可以使用以下命令來獲取 IOT\_GET\_CREDENTIAL\_ENDPOINT：

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

輸出 JSON 包含 `accessKey`、`secretKey` 和工作 `sessionToken`，您可以使用它來存取 Kinesis Video Streams。

- 對於測試，您可以使用這些認證來為範例視訊串流叫用 Kinesis `kvs_example_camera_stream` 影片串流 `DescribeStream` API。

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

步驟 5：在攝影機的檔案系統上部署 AWS IoT 憑證和認證，並將資料串流至視訊串流

### Note

本節中的步驟說明從使用中的攝影機將媒體傳送至 Kinesis 視訊串流。[the section called “C++ Producer Library”](#)

- 將先前步驟中產生的 X.509 憑證、私密金鑰和 CA 憑證複製到相機的檔案系統。指定儲存這些檔案的路徑、角色別名名稱以及執行 `gst-launch-1.0` 命令或範例應用程式的 AWS IoT 認證端點。
- 下列範例命令使用 AWS IoT 憑證授權將視訊傳送至 Kinesis Video Streams：

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink stream-
name="kvs_example_camera_stream" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem.key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias"
```

## AWS IoT CertificateId 作為流名稱

若要透過物 AWS IoT 件來代表您的裝置 (例如，您的攝影機)，但要授權不同的串流名稱，您可以使用此 AWS IoT `certificateId` 屬性做為串流名稱，並使 AWS IoT 提供串流的 Kinesis Video Streams 權限。完成此操作的步驟與先前概述的步驟類似，只需進行一些更改。

- 將權限政策修改為您的 IAM 角色 (`iam-permission-document.json`)，如下所示：



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

### Note

資源 ARN 會將憑證 ID 做為串流名稱的預留位置使用。當您使用憑證 ID 做為串流名稱時，IAM 權限將起作用。從憑證取得憑證 ID，以便您可以在下列描述串流 API 呼叫中將其用作串流名稱。

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

- 使用 Kinesis Video Streams describe-stream CLI 命令驗證此變更。

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

- 將 certificateId 傳遞給 Kinesis Video Streams C++ SDK 中 [範例應用程式](#) 中的認證提供者：AWS IoT

```
credential_provider =
  make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
    cert_path,
    private_key_path,
    role_alias,
```

```
ca_cert_path,  
certificateId);
```

### Note

請注意，您要將物件名稱傳遞給 AWS IoT 認證提供者。您可以使用 `getenv` 將物件名稱傳遞給示範應用程式，類似於傳遞其他屬性。AWS IoT 當您在執行範例應用程式時，請在命令列參數中將憑證 ID 做為串流名稱使用。

## 使用 AWS IoT 認證串流至硬式編碼串流名稱

若要透過物 AWS IoT 件代表您的裝置 (例如，您的攝影機)，但授權串流至特定 Amazon Kinesis 影片串流，請使用以下方式在串流上提供 Amazon Kinesis 影片串流許可。AWS IoT 該過程與前面的章節類似，只有一些更改。

修改 IAM 角色 (`iam-permission-document.json`) 的許可政策，如下所示：

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

將先前步驟中產生的 X.509 憑證、私密金鑰和 CA 憑證複製到相機的檔案系統。

指定儲存這些檔案的路徑、角色別名名稱、AWS IoT 物件名稱以及執行 `gst-launch-1.0` 命令或範例應用程式的 AWS IoT 認證端點。

下列範例命令使用 AWS IoT 憑證授權將影片傳送到 Amazon Kinesis Video Streams：

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink
stream-name="YourStreamName" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem.key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias,iot-
thing-name=YourThingName"
```

## 監控 Amazon Kinesis Video Streams

Kinesis Video Streams 為您的交付串流提供監控功能。如需詳細資訊，請參閱 [監控](#)。

## 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](#)。

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](#) — 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](#)。
- [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。

## Amazon Kinesis Video Streams 的彈性

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。AWS 區域提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠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。

如需區域和可用區域的相關 AWS 資訊，請參閱[AWS 全域基礎結構](#)。

## Kinesis Video Streams 中的基礎架構安全性

作為受管服務，Amazon Kinesis Video Streams 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 Video Streams。用戶端必須支援 Transport Layer Security (TLS) 1.2 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，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的秘密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

## Kinesis Video Streams 的安全性最佳做法

Amazon Kinesis Video Streams 提供許多安全功能，可在您開發和實作自己的安全政策時考慮。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

授與權限時，您可以決定誰取得哪些 Kinesis Video Streams 資源的權限。您還需針對這些資源啟用允許執行的動作，因此，您只應授與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PutMediaGetStreamingEndpoint、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

生產者和用戶端應用程式必須具有有效的認證才能存取 Kinesis Video Streams。您不應該將 AWS 憑證直接存放在用戶端應用程式或 Amazon S3 儲存貯體中。這些是長期憑證，不會自動輪換，如果遭到破壞，可能會產生重大的業務影響。

相反地，您應該使用 IAM 角色來管理生產者和用戶端應用程式的臨時登入資料，以存取 Kinesis Video Streams。使用角色時，您不需要使用長期認證 (例如使用者名稱和密碼或存取金鑰) 來存取其他資源。

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 可搭配使用者 AWS CloudTrail、角色或 Kinesis Video Streams AWS 服務中所採取的動作記錄的服務。

您可以使用收集的資訊 CloudTrail 來判斷向 Kinesis Video Streams 提出的要求、提出要求的來源 IP 位址、提出要求的人員、提出要求的時間以及其他詳細資訊。

如需詳細資訊，請參閱 [the section called “使用 記錄 CloudTrail API 呼叫”](#)。

## 室壁運動視頻流生產者庫

Amazon Kinesis 影片串流製作者程式庫是 Kinesis 影片串流製作者開發套件中的一組程式庫。用戶端使用程式庫和 SDK 建置裝置上的應用程式，以安全地連線到 Kinesis Video Streams，並使用串流媒體資料以在主控台或用戶端應用程式中即時檢視。

媒體資料可以依下列方式進行串流：

- 實時
- 緩衝幾秒鐘後
- 媒體上傳後

建立 Kinesis 影片串流後，您就可以開始將資料傳送至該串流。您可以使用 SDK 建立從媒體來源擷取視訊資料 (稱為畫面) 的應用程式程式碼，並將其上傳到 Kinesis Video Streams。這些應用程式也稱為生產者應用程式。

製作者資料庫包含以下元件：

- [室壁運動視頻流生產者客戶端](#)
- [室壁運動視頻流生產者庫](#)

## 室壁運動視頻流生產者客戶端

Kinesis 視訊串流製作者用戶端包含單一 `KinesisVideoClient` 類。此類別會在資料從媒體來源流向 Kinesis Video Streams 時管理媒體來源、接收來自來源的資料，以及管理串流生命週期。它還提供了一個 `MediaSource` 用於定義 Kinesis 影片串流與您專屬硬體和軟體之間互動的介面。

媒體來源可以是任何項目。例如，您可以使用攝影機媒體來源或麥克風媒體來源。媒體來源不僅限於音訊和影片來源。例如，資料日誌可能是文字檔案，但它們依然可以做為資料串流來傳送。您也可以電話上擁有多個攝影機，以便同時串接資料。

若要從這些資源取得資料，您可以實作 `MediaSource` 介面。此介面可啟用我們不提供內建支援的其他案例。例如，您可以選擇將下列內容傳送至 Kinesis 影片串流：

- 診斷資料串流 (例如，應用程式日誌和事件)
- 從紅外線攝影機、RADARs 或深度攝影機的資料

Kinesis 視訊串流不為攝影機等媒體產生的裝置提供內建實作。若要從這些裝置擷取資料，您必須實作程式碼，藉此建立您自己的自訂媒體來源實作。然後，您可以使用明確註冊自訂媒體來源 KinesisVideoClient 會將資料上傳至 Kinesis 影片串流。

室壁運動視訊串流產生器用戶端可用於 Java 和安卓應用程式。如需詳細資訊，請參閱 [使用 Java Producer Library](#) 及 [使用 Android Producer Library](#)。

## 室壁運動視頻流生產者庫

Kinesis 視訊串流製作者程式庫包含在 Kinesis 視訊串流製作者用戶端中。該庫也可以直接用於那些想要與 Kinesis 視頻流進行更深入集成的用戶。它讓您可透過內含專屬的作業系統、網路堆疊，有限裝置內資源的裝置進行整合。

Kinesis 視訊串流產生器程式庫實作用於串流至 Kinesis 視訊串流的狀態機器。它提供回呼勾點，此需要您提供自己的傳輸實作和明確處理每則傳入和來自服務的訊息。

基於下列原因，您可能會選擇直接使用 Kinesis 影片串流製作者程式庫：

- 您想要執行應用程式的裝置沒有 Java 虛擬機器。
- 您想以非 Java 語言編寫應用程式程式碼。
- 由於內存和處理能力等限制，您希望減少代碼中的開銷量並將其限制為最低級別的抽象。

目前，室壁運動影片串流產生器程式庫可供安卓、C、C++ 和 Java 應用程式使用。如需詳細資訊，請參閱下列支援的語言相關主題。

## 相關主題

[使用 Java Producer Library](#)

[使用 Android Producer Library](#)

[使用 C++ Producer Library](#)

[使用 C Producer Library](#)

[在 Raspberry Pi 使用 C++ Producer 開發套件](#)



## 使用 Java Producer Library

您可以使用 Amazon Kinesis Video Streams 提供的 Java 製作者程式庫，以最少的組態撰寫應用程式程式碼，將媒體資料從裝置傳送到 Kinesis 影片串流。

請執行下列步驟，將您的程式碼與 Kinesis Video Streams 整合，以便您的應用程式可以開始將資料串流到 Kinesis 視訊串流：

1. 建立 `KinesisVideoClient` 物件的執行個體。
2. 透過提供媒體資源資訊建立 `MediaSource` 物件。例如，當建立攝影機媒體資源時，您提供如識別攝影機和指定攝影機所使用之編碼的資訊。

當您想要開始串流，您必須建立自訂媒體資源。

3. 使用 `KinesisVideoClient` 註冊媒體來源。

使用 `KinesisVideoClient` 註冊媒體來源後，每當媒體來源有可用的資料時，它會使用資料呼叫 `KinesisVideoClient`。

### 程序：使用 Java Producer 開發套件

此程序示範如何在 Java 應用程式中使用 Kinesis Video Streams Java 製作者用戶端，將資料傳送至您的 Kinesis 視訊串流。

這些步驟不會要求您擁有媒體來源 (例如相機或麥克風)。反之，為了進行測試，程式碼會產生由一系列位元組組成的範例影格。當您透過真實的來源 (例如相機和麥克風) 來傳送媒體資料時，您可以使用相同的編碼模式。

此程序包含以下步驟：

- [下載和設定程式碼](#)
- [撰寫和檢查程式碼](#)
- [執行並驗證程式碼](#)

### 必要條件

- 在範例程式碼中，您可以透過指定您在認證設定檔檔案中設定的設定檔來提供 AWS 認證。如果您尚未這麼做，請先設定您的登入資料設定檔。如需詳細資訊，請參閱中的 [設定 AWS 認證和開發區域AWS SDK for Java](#)。



**Note**

Java 範例會使用 `SystemPropertiesCredentialsProvider` 物件取得認證。供應商會從 `aws.accessKeyId` 和 `aws.secretKey` Java 系統屬性擷取這些登入資料。您在 Java 開發環境中設定這些系統屬性。如需有關如何設定 Java 系統屬性的詳細資訊，請參閱適用於您特定整合開發環境 (IDE) 的文件。

- 您 `NativeLibraryPath` 必須包含您的 `KinesisVideoProducerJNI` 文件，可在以下位置找到 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>。此檔案的副檔名取決於您的作業系統：
  - `KinesisVideoProducer` 適用於 Linux 的 `JN.so`
  - `KinesisVideoProducer` 適用於 macOS 的 `Jni.Dylib`
  - `KinesisVideoProducer` 適用於視窗的 `JNI.dll`

**Note**

適用於 macOS，Ubuntu，視窗和覆盆子的預構建庫可在以下 `src/main/resources/lib` 位置找到 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java> .git。若是其他環境，則編譯 [C++ Producer Library](#)。

## 步驟 1：下載並設定 Java 製作者程式庫程式碼

在 Java Producer Library 程序的此區段，您下載 Java 範例程式碼、將專案匯入到您的 Java IDE 並設定程式庫位置。

有關此範例的先決條件和其他詳細資訊，請參閱 [使用 Java Producer Library](#)。

1. 建立目錄，然後從 GitHub 儲存庫複製範例原始程式碼。

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

2. 開啟您使用的 Java 整合式開發環境 (IDE) (例如，[Eclipse](#) 或 [JetBrains IntelliJ IDEA](#))，然後匯入您下載的 Apache Maven 專案：
  - 在 IntelliJ IDEA：選擇 Import (匯入)。巡覽至已下載套件根目錄的 `pom.xml` 檔案。

- 在 Eclipse 中：選擇 File (檔案)、Import (匯入)、Maven (Maven)、Existing Maven Projects (現有 Maven 專案)。接著，前往 `kinesis-video-java-demo` 目錄。

如需更多詳細資訊，請參閱您的 IDE 文件。

3. Java 範例程式碼會使用目前的 AWS 認證。若要使用不同的登入資料設定檔，請在 `DemoAppMain.java` 中尋找以下程式碼：

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

將程式碼變更為下列內容：

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

如需詳細資訊，請參閱 AWS SDK for Java 參考中的 [ProfileCredentialsProvider](#)。

## 後續步驟

[the section called “步驟 2：編寫並檢查代碼”](#)

### 步驟 2：編寫並檢查代碼

在「[Java 製作者程式庫](#)」程序的這個區段中，您會撰寫並檢查您在上一節中下載的 Java 範例程式碼。

Java 測試應用程式 ([DemoAppMain](#)) 顯示以下編碼模式：

- 建立 `KinesisVideoClient` 的執行個體。
- 建立 `MediaSource` 的執行個體。
- 使用用戶端註冊 `MediaSource`。
- 開始串流。啟動 `MediaSource`，它開始將數據發送到客戶端。

下列各節將詳細說明。

## 建立執行個體 KinesisVideoClient

您呼叫 `createKinesisVideoClient` 操作建立 `KinesisVideoClient` 物件。

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

如果要讓 `KinesisVideoClient` 進行網路呼叫，需要可驗證的登入資料。您會傳遞 `SystemPropertiesCredentialsProvider` 執行個體，其會針對登入資料檔案中的預設設定檔讀取 `AWSCredentials`：

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

## 建立執行個體 MediaSource

若要將位元組傳送至 Kinesis 視訊串流，您必須產生資料。Amazon Kinesis Video Streams 提供代表資料來源的 `MediaSource` 介面。

例如，Kinesis Video Streams Java 程式庫提供 `MediaSource` 介面的 `ImageFileMediaSource` 實作。此類別只會從一系列媒體檔案讀取資料，而不是 Kinesis 視訊串流，但您可以使用它來測試程式碼。

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

## 向 MediaSource 用戶端註冊

註冊您使用 `KinesisVideoClient` 所建立的媒體來源，系統就會知道該用戶端 (並可將資料傳送到用戶端)。

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

## 啟動媒體來源

啟動媒體來源，以便它可以開始產生資料並將其傳送至用戶端。

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

## 下一步驟

[the section called “步驟 3：運行並驗證代碼”](#)

## 步驟 3：運行並驗證代碼

要運行 Java [生產者庫的 Java](#) 測試工具，請執行以下操作。

1. 選擇 DemoAppMain。
2. 選擇運行，運行 'DemoAppMain'。
3. 將您的登入資料新增至應用程式的 JVM 引數：
  - 對於非臨時 AWS 身份證明："-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
  - 對於臨時 AWS 登入資料："-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"
4. 登入 AWS Management Console 並開啟 [Kinesis Video Streams 主控台](#)。

在 Manage Streams (管理串流) 頁面中，選擇您的串流。

5. 範例影片會在嵌入式播放器中播放。在出現視訊之前，您可能需要稍微等待畫面聚積 (在一般性頻寬和處理器的條件下最多 10 秒)。

建立串流的程式碼範例。當程式碼中的 MediaSource 開啟時，會開始將範例影格傳送到 KinesisVideoClient。用戶端接著會將資料傳送到 Kinesis 視訊串流。

## 使用 Android Producer Library

您可以使用 Amazon Kinesis Video Streams 提供的 Android 製作者程式庫撰寫應用程式程式碼 (最小組態)，將媒體資料從 Android 裝置傳送到 Kinesis 影片串流。

請執行下列步驟，將您的程式碼與 Kinesis Video Streams 整合，以便您的應用程式可以開始將資料串流到 Kinesis 視訊串流：

1. 建立 KinesisVideoClient 物件的執行個體。
2. 透過提供媒體資源資訊建立 MediaSource 物件。例如，當建立攝影機媒體資源時，您提供如識別攝影機和指定攝影機所使用之編碼的資訊。

當您想要開始串流，您必須建立自訂媒體資源。

### 程序：使用 Android Producer 開發套件

此程序示範如何在 Android 應用程式中使用 Kinesis Video Streams Android 製作者用戶端，將資料傳送至您的 Kinesis 視訊串流。

此程序包含以下步驟：

- [the section called “必要條件”](#)
- [the section called “步驟 1：下載和設定程式碼”](#)
- [the section called “步驟 2：檢查代碼”](#)
- [the section called “步驟 3：運行並驗證代碼”](#)

### 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：集成您的應用程式-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`。

- 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

- 登入 [Amazon Cognito 主控台](#) 並確認區域是否正確。
- 在左側導覽中，選擇 [識別集區]。
- 選擇 建立身分池。
- 設定身分識別集區。
  - 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。
  - 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。
  - 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。
  - 步驟 4：配置屬性-在 [識別集區名稱] 欄位中輸入名稱。  
選取下一步。
  - 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]。
- 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 IAM 主控台，網址為 <https://console.aws.amazon.com/iam/>。
  - b. 在左側導覽中，選擇 [角色]。
  - c. 尋找並選取您在上一步建立的角色。

 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. 選取 [將此新版本設定為預設值] 旁邊的方塊 (如果尚未選取)。

選取儲存變更。



## 步驟 1：下載並配置 Android 生產者庫代碼

在 Android Producer Library 程序的這個區段中，您下載 Android 範例程式碼並在 Android Studio 中開啟專案。

有關此範例的先決條件和其他詳細資訊，請參閱[使用 Android Producer Library](#)。

1. 建立目錄，然後 AWS Mobile SDK for Android 從 GitHub 儲存庫複製該目錄。

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

2. 開啟 [Android Studio](#)。
3. 在開啟畫面中，選擇 Open an existing Android Studio project (開放現有的 Android Studio 專案)。
4. 導覽至 aws-sdk-android-samples/AmazonKinesisVideoDemoApp 目錄，並選擇 OK (確定)。
5. 開啟 AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json 檔案。

在 CredentialsProvider 節點中，從 [必要條件] 段落中的 [設定識別集區] 程序提供識別集區識別碼，並提供您的 AWS 區域 (例如，**us-west-2**)。

在 CognitoUserPool 節點中，從 [必要條件] 區段中的 [要設定使用者集區程序] 提供 [應用程式用戶端密碼]、[應用程式用戶端識別碼] 和 [集區識別碼]，並提供您的 AWS 區域 (例如，**us-west-2**)。

6. 您的 awsconfiguration.json 檔案看起來如下：

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

```
"CognitoUserPool": {
  "Default": {
    "AppClientSecret": "abcdefghijklmnopqrstuvwxyz0123456789abcdefghijklmnopqrstuvwxyz",
    "AppClientId": "0123456789abcdefghijklmnopqrstuvwxyz",
    "PoolId": "us-west-2_qRsTuVwXy",
    "Region": "us-west-2"
  }
}
```

7. AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java使用您的區域更新（在以下示例中，它設置為US\_WEST\_2）：

```
public class KinesisVideoDemoApp extends Application {
    public static final String TAG = KinesisVideoDemoApp.class.getSimpleName();
    public static Regions KINESIS_VIDEO_REGION = Regions.US_WEST_2;
```

如需 AWS 區域常數的相關資訊，請參閱[區域](#)。

## 下一步驟

[the section called “步驟 2：檢查代碼”](#)

## 步驟 2：檢查代碼

在 [Android Producer Library 程序](#) 的這個區段中，您檢查範例程式碼。

Android 測試應用程式 (AmazonKinesisVideoDemoApp) 顯示以下編碼模式：

- 建立 KinesisVideoClient 的執行個體。
- 建立 MediaSource 的執行個體。
- 開始串流。啟動 MediaSource，然後開始將資料傳送至用戶端。

下列各節將詳細說明。

## 建立執行個體 KinesisVideoClient

您呼叫 [createKinesisVideoClient](#) 操作建立 [KinesisVideoClient](#) 物件。

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(
    getActivity(),
    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,
    KinesisVideoDemoApp.getCredentialsProvider());
```

如果要讓 `KinesisVideoClient` 進行網路呼叫，需要可驗證的登入資料。您傳入的執行個體 `AWSCredentialsProvider`，該執行個體會從您在上一節中修改的 `awsconfiguration.json` 檔案讀取 Amazon Cognito 登入資料。

## 建立執行個體 MediaSource

若要將位元組傳送至 Kinesis 視訊串流，您必須產生資料。Amazon Kinesis Video Streams 提供代表資料來源的 [MediaSource](#) 介面。

例如，Kinesis Video Streams Android 程式庫提供 `MediaSource` 介面的 [AndroidCameraMediaSource](#) 實作。此類別讀取其中一個裝置攝影機的資料。

在以下程式碼範例 (來自 [fragment/StreamConfigurationFragment.java](#) 檔案) 中，將建立媒體資源的組態：

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

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

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

            .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
                .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
                .withIsEncoderHardwareAccelerated(
                    mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
                .withFrameRate(FRAMERATE_20)
                .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
                .withEncodingBitRate(BITRATE_384_KBPS)
```

```
        .withCameraOrientation(-
mCamerasDropdown.getSelectedItem().getCameraOrientation())

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

在以下程式碼範例 (來自 [fragment/StreamingFragment.java](#) 檔案) 中，將建立媒體資源：

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

## 啟動媒體來源

開啟媒體來源，以開始產生資料並傳送到用戶端。以下是來自 [fragment/StreamingFragment.java](#) 檔案的程式碼範例：

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

## 下一步驟

[the section called “步驟 3：運行並驗證代碼”](#)

## 步驟 3：運行並驗證代碼

若要執行 [Android Producer Library](#) 的 Android 範例應用程式，請執行下列動作。

1. 連接 Android 裝置。
2. 選擇 Run (執行)、Run... (執行...)，然後選擇 Edit configurations... (編輯組態...)。
3. 選擇加號圖標 (+)，安卓應用。在 Name (名稱) 欄位中，輸入 **AmazonKinesisVideoDemoApp**。在「模組」下拉式清單中，選擇 AmazonKinesisVideoDemoApp。選擇 確定。
4. 選擇 Run (執行)、Run (執行)。
5. 在 Select Deployment Target (選取部署目標) 畫面中，選擇您的連線裝置，然後選擇 OK (確定)。
6. 在裝置上的 AWSKinesisVideoDemoApp 應用程式中，選擇 [建立新帳戶]。
7. 輸入 USERNAME (使用者名稱)、Password (密碼)、Given name (名字)、Email address (電子郵件地址) 和 Phone number (電話號碼) 的值，然後選擇 Sign up (註冊)。

**Note**

這些值有下列限制：

- 密碼：必須包含大小寫字母、數字和特殊字元。您可以在 [Amazon Cognito 主控台](#) 的使用者集區頁面中變更這些限制。
- 電子郵件地址：必須為有效的地址，讓您可以收到確認代碼。
- 電話號碼：格式須如下：**+<Country code><Number>**，例如，**+12065551212**。

8. 輸入您透過電子郵件收到的驗證碼，然後選擇「確認」。選擇確定。
9. 在下一頁上，保留預設值，然後選擇「串流」。
10. 登入 AWS Management Console 並開啟美國西部 (奧勒岡) 區域的 [Kinesis Video Streams 主控台](#)。

在 Manage Streams (管理串流) 頁面中，選擇 demo-stream (示範串流)。

11. 串流視訊在內建播放器中播放。在出現視訊之前，您可能需要稍微等待畫面聚積 (在一般性頻寬和處理器的條件下最多 10 秒)。

**Note**

如果裝置的畫面旋轉 (例如，從縱向到橫向)，應用程式會停止串流視訊。

建立串流的程式碼範例。當程式碼中的 `MediaSource` 開啟時，會開始將攝影機的畫面傳送到 `KinesisVideoClient`。用戶端接著會將資料傳送至名為示範串流的 Kinesis 視訊串流。

## 使用 C++ Producer Library

您可以使用 Amazon Kinesis Video Streams 提供的 C++ 製作者程式庫撰寫應用程式程式碼，將媒體資料從裝置傳送到 Kinesis 視訊串流。

### 物件模型

C++ 程式庫提供下列物件來管理將資料傳送至 Kinesis 視訊串流：

- `KinesisVideoProducer`：包含媒體來源和 AWS 認證的相關資訊，並維護回呼以報告 Kinesis Video Streams 事件。

- `KinesisVideoStream`：代表 Kinesis 視訊串流。包含視訊串流參數的相關資訊，例如名稱、資料保留期和媒體內容類型。

## 將媒體放入串流

您可以使用 C++ 庫提供的方法（例如，`PutFrame`）將數據放入對 `KinesisVideoStream` 象中。然後程式庫會管理資料的內部狀態，包括以下任務：

- 執行身分驗證。
- 查看網路延遲。如果延遲太高，程式庫可能選擇減少影格數。
- 追蹤進行中串流的狀態。

## 回呼介面

此層公開一組回呼介面，使其能與應用程式層溝通。這些回呼介面包含以下項目：

- 服務回調 interface (`CallbackProvider`)：庫在創建流時調用通過此接口獲得的事件，獲取流描述並刪除流。
- 用戶端就緒狀態或低儲存事件界面 (`ClientCallbackProvider`)：當用戶端就緒，或偵測到可能耗盡可用的儲存或記憶體時，程式庫呼叫此介面的事件。
- 串流事件回呼界面 (`StreamCallbackProvider`)：當發生如串流進入就緒狀態、畫面數減少或串流錯誤等串流事件時，程式庫呼叫此介面的事件。

Kinesis Video Streams 提供這些介面的預設實作。您也可以提供自己的自訂實作，例如，如果您需要自訂網路邏輯，或想要將低儲存條件公開給使用者介面。

如需有關製作者程式庫回呼的詳細資訊，請參閱 [生產者 SDK 回呼](#)。

## 程序：使用 C++ Producer 開發套件

此程序示範如何在 C++ 應用程式中使用 Kinesis Video Streams 用戶端和媒體來源，將資料傳送至 Kinesis 視訊串流。

此程序包含以下步驟：

- [步驟 1：下載和設定程式碼](#)
- [步驟 2：撰寫和檢查程式碼](#)

- [步驟 3：執行並驗證程式碼](#)

## 先決條件

- **認證：**在範例程式碼中，您可以透過指定您在認證設定檔檔案中設定的設定檔來提供AWS認證。如果您尚未這麼做，請先設定您的登入資料設定檔。

如需詳細資訊，請參閱[設定AWS認證和開發區域](#)。

- **憑證儲存區整合：**Kinesis Video Streams 製作者程式庫必須與其呼叫的服務建立信任。這是透過驗證公用憑證存放區中的憑證授權單位 (CA) 來完成的。在以 Linux 為基礎的模型中，此存放位於 `/etc/ssl/` 目錄。

將下列位置的憑證下載到您的憑證存放區：

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

- 安裝下列適用於 macOS 的建構相依性：
  - [Autoconf 2.69](#) (授權 GPLv3+/Autoconf：GNU GPL 第 3 版或更新版本)
  - [CMake 3.7 或 3.8](#)
  - [Pkg-Config](#)
  - [Flex 2.5.35 Apple \(flex-31\) 或更新版本](#)
  - [Bison 2.4](#) (GNU 授權)
  - [Automake 1.15.1](#) (GNU 授權)
  - GNU Libtool (Apple Inc. cctools-898 版)
  - xCode (macOS)/clang/gcc (xcode-select 2347 版)
  - Java 開發套件 (JDK) (適用於 Java JNI 編譯)
  - [Lib-Pkg](#)
- 安裝適用於 Ubuntu 的建構相依性 (截去版本命令的回應)：
  - 安裝 Git：`sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- 安裝 [CMake](#)：`sudo apt-get install cmake`

```
cmake version 3.9.1
```

- 安裝 Libtool : `sudo apt-get install libtool`

```
2.4.6-2
```

- 安裝 libtool-bin : `sudo apt-get install libtool-bin`

```
$ libtool --version  
libtool (GNU libtool) 2.4.6  
Written by Gordon Matzigkeit, 1996
```

- 安裝 GNU Automake : `sudo apt-get install automake`

```
$ automake --version  
automake (GNU automake) 1.15
```

- 安裝 GNU Bison : `sudo apt-get install bison`

```
$ bison -V  
bison (GNU Bison) 3.0.4
```

- 安裝 G++ : `sudo apt-get install g++`

```
g++ --version  
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- 安裝 curl : `sudo apt-get install curl`

```
$ curl --version  
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11  
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- 安裝 pkg-config : `sudo apt-get install pkg-config`

```
$ pkg-config --version  
0.29.1
```

- 安裝 Flex : `sudo apt-get install flex`

```
$ flex --version  
flex 2.6.1
```



- 安裝 OpenJDK : `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- 設定 JAVA\_HOME 環境變數 : `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
- 執行建構指令碼 : `./install-script`

## 下一步驟

### [步驟 1：下載和設定 C++ Producer Library 程式碼](#)

## 步驟 1：下載並設定 C++ 生產者程式庫程式碼

如需如何下載和設定 C++ 製作者程式庫的詳細資訊，請參閱 [Amazon Kinesis Video Streams CPP 製作者、GStreamer 外掛程式和 JNI](#)。

如需有關此範例的必要條件和詳細資訊，請參閱[使用 C++ 製作者程式庫](#)。

## 下一步驟

### [步驟 2：編寫並檢查代碼](#)

## 步驟 2：撰寫並檢查程式碼

在此 [C++ Producer Library 程序](#) 的區段中，您在 C++ 測試工具 (tst/ProducerTestFixture.h 和其他檔案) 中檢查程式碼。您在上一節中下載了此程式碼。

平台獨立 C++ 範例顯示下列編碼模式：

- 建立的執行個體 `KinesisVideoProducer` 以存取 Kinesis Video Streams。
- 建立 `KinesisVideoStream` 的執行個體。AWS 帳戶如果不存在相同名稱的串流，這會在您的 Kinesis 視訊串流中建立。
- 當可用時，在 `KinesisVideoStream` 上呼叫每個資料畫面的 `putFrame`，將其傳送到串流。

以下區段提供有關此程式碼模式的更多資訊。

## 建立執行個體 KinesisVideoProducer

您呼叫 `KinesisVideoProducer` 方法建立 `KinesisVideoProducer::createSync` 物件。下列範例在 `KinesisVideoProducer` 檔案中建立 `ProducerTestFixture.h`：

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

`createSync` 方法會使用下列三個參數：

- `DeviceInfoProvider` 物件，其會傳回包含裝置或存放區組態相關資訊的 `DeviceInfo` 物件。

### Note

您使用 `deviceInfo.storageInfo.storageSize` 參數設定內容存放區大小。您的內容串流會共用內容存放區。若要判斷您的儲存大小需求，請將平均畫面大小乘以為所有串流最大持續時間存放的畫面數量，再乘以 1.2 以計算磁碟重組。例如，假設您的應用程式具有以下組態：

- 三個串流
- 最大持續時間 3 分鐘
- 每個串流為每秒 30 個影格 (FPS)
- 每個畫面大小為 10,000 KB

此應用程式的內容存放區需求為  $3 \text{ (串流)} * 3 \text{ (分鐘)} * 60 \text{ (一分鐘內的秒數)} * 10000 \text{ (kb)} * 1.2 \text{ (磁碟重組容量)} = 194.4 \text{ Mb} \sim 200 \text{ Mb}$ 。

- `ClientCallbackProvider` 物件，其會傳回報告用戶端特定事件的函數指標。
- `StreamCallbackProvider` 物件，其會傳回串流特定事件發生後收回的函數指標。
- `CredentialProvider` 物件，可讓您存取 AWS 認證環境變數。
- AWS 區域 (「美國西部 -2」)。服務端點取決於區域。

## 建立執行個體 KinesisVideoStream

您呼叫使用 `StreamDefinition` 參數的 `KinesisVideoProducer::CreateStream` 方法建立 `KinesisVideoStream` 物件。該範例會在 `ProducerTestFixture.h` 檔案中建立 `KinesisVideoStream`，並將軌道類型設為視訊，軌道 ID 設為 1：

```
auto stream_definition = make_unique<StreamDefinition>(stream_name,
                                                       hours(2),
                                                       tags,
                                                       "",
                                                       STREAMING_TYPE_REALTIME,
                                                       "video/h264",
                                                       milliseconds::zero(),
                                                       seconds(2),
                                                       milliseconds(1),
                                                       true,
                                                       true,
                                                       true);
return kinesis_video_producer_->createStream(move(stream_definition));
```

`StreamDefinition` 物件包含以下欄位：

- 串流名稱。
- 資料保留期間。
- 標籤的串流。這些標籤可供消費者應用程式用於尋找正確的串流，或是取得更多關於串流的資訊。也可以在 AWS Management Console 查看這些標籤。
- AWS KMS 串流的加密金鑰。如需詳細資訊，請參閱[使用 Kinesis Video Streams 使用進行伺服器端加密 Kin](#)
- 串流類型。目前唯一有效的值為：`STREAMING_TYPE_REALTIME`。
- 媒體內容類型。
- 媒體延遲。此值目前未使用，應該設定為 0。
- 每個片段的播放持續時間。
- 媒體時間碼擴展。
- 媒體是否使用關鍵影格片段。
- 媒體是否使用時間碼。
- 媒體是否使用絕對片段時間。

## 將音訊軌道加入 Kinesis 視訊串流

您可以使用下列項目的 `addTrack` 方法，將音軌詳細資料新增至視訊軌道串流定義：StreamDefinition

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

`addTrack` 方法需要下列參數：

- 音軌 ID ( 作為音頻的一個 )。這應該是唯一且非零的數值。
- 使用者定義的音軌名稱 (例如，音軌的「音訊」)。
- 此音軌的編解碼器 ID (例如，音軌「A\_AAC」)。
- 軌道類型 ( 例如，對音頻使用 `MKV_TRACK_ 類型 _ 音頻` 的枚舉值 )。

如果您有音軌的轉碼器私有資料，則可以在呼叫 `addTrack` 函數時傳遞。在中呼叫 `start` 方法時，您也可以在建 KinesisVideoStream 物件之後傳送轉碼器專用資料 KinesisVideoStream。

## 將影格放入 Kinesis 視訊串流中

您可以使用將媒體放入 Kinesis 視訊串流 `KinesisVideoStream::putFrame`，並傳入包含標頭和媒體資料的 `Frame` 物件。此範例呼叫 `ProducerApiTest.cpp` 檔案中的 `putFrame`：

```
frame.duration = FRAME_DURATION_IN_MICROS * HUNDREDS_OF_NANOS_IN_A_MICROSECOND;
frame.size = sizeof(frameBuffer_);
frame.frameData = frameBuffer_;
memset(frame.frameData, 0x55, frame.size);

while (!stop_producer_) {
    // Produce frames
    timestamp = std::chrono::duration_cast<std::chrono::nanoseconds>(
        std::chrono::system_clock::now().time_since_epoch()).count() /
    DEFAULT_TIME_UNIT_IN_NANOS;
    frame.index = index++;
    frame.decodingTs = timestamp;
    frame.presentationTs = timestamp;

    // Key frame every 50th
    frame.flags = (frame.index % 50 == 0) ? FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    ...
}
```

```
EXPECT_TRUE(kinesis_video_stream->putFrame(frame));
```

### Note

上述 C++ Producer 範例傳送測試資料的緩衝。在實際的應用程式中，您應該從媒體資源 (例如攝影機) 獲得畫面資料的畫面緩衝和大小。

Frame 物件包含以下欄位：

- 畫面索引。此應為單一遞增的值。
- 與畫面關聯的旗標。例如，如果設定編碼器產生關鍵影格，將會指定 FRAME\_FLAG\_KEY\_FRAME 旗標給此畫面。
- 解碼時間戳記。
- 演示時間戳記。
- 畫面的持續時間 (到 100 ns 單位)。
- 畫面的大小 (位元組)。
- 畫面資料。

如需框架格式的詳細資訊，請參閱 [Kinesis Video Streams 資料模型](#)。

## 把一個KinesisVideoFrame到一個特定的軌道 KinesisVideoStream

您可以使用該PutFrameHelper類將幀數據放入特定軌道中。首先，調用 getFrameData Buffer 以獲取指向其中一個預先分配緩衝區的指針以填充數KinesisVideoFrame據。然後，您可以調用 putFrameMulti Track 與布爾值—KinesisVideoFrame起發送，以指示幀數據的類型。如果是視訊資料，使用 true，或者，如果影格包含音訊資料，則使用 false。T putFrameMulti rack 方法使用排隊機制來確保 MKV 片段保持單調增加的幀時間戳，並且任何兩個片段不會重疊。例如，片段第一幀的 MKV 時間戳應始終大於上一個片段最後一幀的 MKV 時間戳。

其PutFrameHelper欄位如下：

- 佇列中的音訊影格數目上限。
- 佇列中的視訊影格數目上限。
- 分配給單個音頻幀的大小。

- 為單一視訊影格配置的大小。

## 指標和指標記錄

C++ Producer 開發套件包含適用於指標和指標記錄的功能。

您可以使用 `getKinesisVideoMetrics` 和 `getKinesisVideoStreamMetrics` API 操作來擷取 Kinesis 影片串流和作用中串流的相關資訊。

以下是來自 `kinesis-video-pic/src/client/include/com/amazonaws/kinesis/video/client/Include.h` 檔案的程式碼。

```
/**
 * Gets information about the storage availability.
 *
 * @param 1 CLIENT_HANDLE - the client object handle.
 * @param 2 PKinesisVideoMetrics - OUT - Kinesis Video metrics to be filled.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoMetrics(CLIENT_HANDLE, PKinesisVideoMetrics);

/**
 * Gets information about the stream content view.
 *
 * @param 1 STREAM_HANDLE - the stream object handle.
 * @param 2 PStreamMetrics - Stream metrics to fill.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoStreamMetrics(STREAM_HANDLE, PStreamMetrics);
```

由 `getKinesisVideoMetrics` 填滿的 `PClientMetrics` 物件包含下列資訊：

- `contentStoreSize`：內容存放區 (用於儲存串流資料的記憶體) 的整體大小 (以位元組為單位)。
- `contentStoreAvailable` 大小：內容存放區中的可用記憶體，以位元組為單位。
- `contentStoreAllocated` 大小：內容存放區中配置的記憶體。
- `totalContentViews` 大小：內容視圖使用的總內存。內容檢視是內容存放區中的一系列資訊索引。
- `totalFrameRate`：所有作用中串流每秒的影格彙總數。

- `totalTransferRate` : 在所有串流中傳送的每秒總位元數 (bps)。

由 `getKinesisVideoStreamMetrics` 填滿的 `PStreamMetrics` 物件包含下列資訊：

- `currentViewDuration`: 內容檢視的標頭 (當影格編碼時) 與目前位置 (當影格資料傳送至 Kinesis Video Streams 時) 之間的差異 (以 100 ns 為單位)。
- `overallViewDuration`: 內容檢視標頭 (當框架編碼時) 到尾端 (當畫面從記憶體中清除時，可能是因為超出內容檢視的配置空間總計，或是因為從 Kinesis Video Streams 接收 `PersistedAck` 訊息，且已知保留的框架會清除) 之間的差異 (以 100 ns 為單位)。
- `currentViewSize` : 從頭部 (當畫面編碼時) 到目前位置 (當畫面傳送至 Kinesis Video Streams 時) 的內容檢視大小 (以位元組為單位)。
- `overallViewSize` : 內容視圖的總大小 (以字節為單位)。
- `currentFrameRate`: 串流的最後測量速率，以每秒影格數為單位。
- `currentTransferRate`: 串流的最後測量速率，以每秒位元組為單位。

## 分解

如果您想要傳送緩衝區中剩餘的位元組，並等待 ACK，則可以使用 `stopSync`：

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

或者，您可以呼叫 `stop` 來結束串流：

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

停止串流後，您可以透過叫用下列 API 來釋出串流：

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

## 下一步驟

[the section called “步驟 3：執行程式碼並驗證程式碼”](#)

## 步驟 3：執行程式碼並驗證程式碼

若要執行並驗證 [C++ 生產者程式庫程序](#) 的程式碼，請參閱下列作業系統特定指示：

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [Raspberry Pi OS](#)

您可以透過在 Amazon CloudWatch 主控台觀看與串流相關聯的指標來監控串流上的流量，例如 `PutMedia.IncomingBytes`。

### 使用 C++ 生產者開發套件作為 GStreage 插件使用

[GStreamer](#) 是一種流行的媒體框架，由多個攝像機和視頻源通過組合模塊化插件來創建自定義媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化您現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。

如需有關使用 C++ Producer 開發套件做為 GStreamer 外掛程式的詳細資訊，請參閱 [範例：Kinesis Video Streams 產生器 SDK 外掛程式](#)。

### 使用 C++ 生產者 SDK 作為碼頭容器中的 GStreamer 插件

[GStreamer](#) 是一種流行的媒體框架，由多個攝像機和視頻源通過組合模塊化插件來創建自定義媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化您現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。

此外，使用 [Docker](#) 建立 GStreamer 管道可將 Kinesis Video Streams 的作業環境標準化，進而簡化應用程式的建置和執行作業。

如需有關使用 C++ Producer 開發套件做為在 Docker 容器中 GStreamer 外掛程式的詳細資訊，請參閱 [在碼頭容器中運行 GStreamer 元素](#)。

### 使用記錄與 C++ 製作者 SDK

您可在 `kvs_log_configuration` 資料夾中的 `kinesis-video-native-build` 檔案為 C++ Producer 開發套件應用程式設定記錄。



以下範例顯示預設組態檔案的第一行，可用來設定應用程式將 DEBUG 層級日誌項目寫至 AWS Management Console：

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

您可以將記錄層級設為 INFO，以用於較為詳細的記錄。

若要設定應用程式將記錄項目寫入記錄檔，請將檔案中的第一行更新為下列項目：

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

這會把應用程式設為將日誌項目寫入至 `kvs.log` 資料夾的 `kinesis-video-native-build/log` 中。

若要變更日誌檔位置，請以新路徑更新下列行：

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

#### Note

如果 DEBUG - 層級記錄為寫入至檔案，則日誌檔可能快速用掉裝置上的可用儲存空間。

## 使用 C Producer Library

您可以使用 Amazon Kinesis Video Streams 提供的 C 製作者程式庫撰寫應用程式程式碼，將媒體資料從裝置傳送到 Kinesis 視訊串流。

### 物件模型

Kinesis Video Streams C 產生器程式庫是以稱為平台獨立程式碼庫 (PIC) 的通用元件為基礎，網址為 <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/>。GitHub PIC 包含基礎組件的獨立於平台的業務邏輯。Kinesis Video Streams C 製作者程式庫會將 PIC 包裝成額外的 API 層，以允許案例和平台特定的回呼和事件。Kinesis Video Streams C 製作者程式庫具有建置在 PIC 之上的下列元件：

- 裝置資訊提供者 — 公開可直接提供給 PIC API 的 `DeviceInfo` 結構。您可以設定一組提供者，包括應用程式案例最佳化的提供者，這些提供者可根據應用程式處理的串流數量和類型，以及根據可用 RAM 數量設定的必要緩衝數量，最佳化內容存放區。

- 串流資訊提供者 — 公開可直接提供給 PIC API 的 `StreamInfo` 結構。有一組特定於應用程式類型和常見的串流案例類型的提供者。其中包括視頻，音頻以及音頻和視頻多軌等提供商。這些案例中的每一個都有預設值，您可以根據應用程式的需求自訂這些預設值。
- 回呼提供者 — 公開可直接提供給 PIC API 的 `ClientCallbacks` 結構。這包括一組網路回呼提供者 (以 `CURL` 為基礎的 API 回呼)、授權 (AWS 憑證 API)，以及在錯誤回呼時重試串流。回呼提供者 API 需要一些引數來設定，例如 AWS 區域和授權資訊。這是透過使用 IoT 憑證或使用 `AWSAccessKeyIdSecretKey`、或來完成 `SessionToken`。如果您的應用程式需要進一步處理特定回呼，才能實現一些應用程式特定邏輯，您可以使用自訂回呼來增強回呼提供者。
- `FrameOrderCoordinator` — 幫助處理多軌場景的音頻和視頻同步。它具有默認行為，您可以自定義該行為以處理應用程式的特定邏輯。它還簡化了 PIC 幀結構中的幀元數據打包，然後將其提交到下層 PIC API。對於非多軌案例，此元件是 PIC `putFrame` API 的傳遞。

C 程式庫提供以下物件，管理將資料傳送到 Kinesis 影片串流的作業：

- `KinesisVideoClient` — 包含有關您裝置的資訊並維護回呼，以報告 Kinesis Video Streams 事件。
- `KinesisVideoStream` — 代表視訊串流參數的相關資訊，例如名稱、資料保留期和媒體內容類型。

## 將媒體放入串流

您可以使用 C 庫提供的方法 (例如，`PutKinesisVideoFrame`) 將數據放入 `KinesisVideoStream` 對象。然後程式庫會管理資料的內部狀態，包括以下任務：

- 執行身分驗證。
- 查看網路延遲。如果延遲太高，程式庫可能選擇減少影格數。
- 追蹤進行中串流的狀態。

## 程序：使用 C Producer 開發套件

此程序示範如何在 C 應用程式中使用 Kinesis Video Streams 用戶端和媒體來源，將 H.264 編碼的視訊影格傳送到 Kinesis 視訊串流。

此程序包含以下步驟：

- [步驟 1：下載 C Producer Library Library 程式碼](#)
- [步驟 2：撰寫程式碼](#)
- [步驟 3：執行程式碼並驗證程式碼](#)

## 先決條件

- 認證 — 在範例程式碼中，您可以透過指定您在認證設定檔檔案中設定的設定檔來提供AWS認證。如果您尚未這麼做，請先設定您的登入資料設定檔。

如需詳細資訊，請參閱[設定AWS認證和開發區域](#)。

- 憑證存放區整合 — Kinesis Video Streams 製作者程式庫必須與其呼叫的服務建立信任。這是透過驗證公用憑證存放區中的憑證授權單位 (CA) 來完成的。在以 Linux 為基礎的模型中，此存放位於 `/etc/ssl/` 目錄。

將下列位置的憑證下載到您的憑證存放區：

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

- 安裝下列適用於 macOS 的建構相依性：
  - [Autoconf 2.69](#) (授權 GPLv3+/Autoconf : GNU GPL 第 3 版或更新版本)
  - [CMake 3.7 或 3.8](#)
  - [Pkg-Config](#)
  - [Flex 2.5.35 Apple \(flex-31\) 或更新版本](#)
  - [Bison 2.4](#) (GNU 授權)
  - [Automake 1.15.1](#) (GNU 授權)
  - GNU Libtool (Apple Inc. cctools-898 版)
  - xCode (macOS)/clang/gcc (xcode-select 2347 版)
  - Java 開發套件 (JDK) (適用於 Java JNI 編譯)
  - [Lib-Pkg](#)
- 安裝適用於 Ubuntu 的建構相依性 (截去版本命令的回應)：
  - 安裝 Git : `sudo apt-get install git`

```
$ git --version
git version 2.14.1
```

- 安裝 [CMake](#) : `sudo apt-get install cmake`

```
$ cmake --version
cmake version 3.9.1
```

- 程序：使用 C Producer 開發套件
- 安裝 Libtool : `sudo apt-get install libtool`

```
2.4.6-2
```

- 安裝 libtool-bin : `sudo apt-get install libtool-bin`

```
$ libtool --version
libtool (GNU libtool) 2.4.6
Written by Gordon Matzigkeit, 1996
```

- 安裝 GNU Automake : `sudo apt-get install automake`

```
$ automake --version
automake (GNU automake) 1.15
```

- 安裝 GNU Bison : `sudo apt-get install bison`

```
$ bison -V
bison (GNU Bison) 3.0.4
```

- 安裝 G++ : `sudo apt-get install g++`

```
g++ --version
g++ (Ubuntu 7.2.0-8ubuntu3) 7.2.0
```

- 安裝 curl : `sudo apt-get install curl`

```
$ curl --version
curl 7.55.1 (x86_64-pc-linux-gnu) libcurl/7.55.1 OpenSSL/1.0.2g zlib/1.2.11
libidn2/2.0.2 libpsl/0.18.0 (+libidn2/2.0.2) librtmp/2.3
```

- 安裝 pkg-config : `sudo apt-get install pkg-config`

```
$ pkg-config --version
0.29.1
```

- 安裝 Flex : `sudo apt-get install flex`

```
$ flex --version
flex 2.6.1
```

- 安裝 OpenJDK : `sudo apt-get install openjdk-8-jdk`

```
$ java -version
openjdk version "1.8.0_171"
```

- 設定 JAVA\_HOME 環境變數：export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
- 執行建構指令碼：./install-script

## 下一步驟

### [步驟 1：下載 C Producer Library Library 程式碼](#)

## 步驟 1：下載 C Producer Library Library 程式碼

在本節中，您會下載低階程式庫。有關此範例的先決條件和其他詳細資訊，請參閱[使用 C Producer Library](#)。

1. 建立目錄，然後從GitHub儲存庫複製範例原始程式碼。

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

### Note

如果您錯過了搭配 `--recursive` 執行 Git 複製，請在 `amazon-kinesis-video-streams-producer-c/open-source` 目錄中運行 `git submodule update --init`。您還必須安裝 `pkg` 配置，自動製作，`CMake` 和構建環境。有關更多信息，請參閱 `README.md` 中的 [amazon-kinesis-video-streams生產者](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c)。git。 <https://github.com/aws-labs/>

2. 在您選擇的整合開發環境 (IDE) 中開啟程式碼 (例如 [Eclipse](#))。

## 下一步驟

### [步驟 2：撰寫程式碼](#)

## 步驟 2：撰寫程式碼

在本節中，您將檢查 <https://github.com/awslabs/amazon-kinesis-video-streams-生產者-c> 存儲庫的 samples 文件夾 KvsVideoOnlyStreamingSample.c 中示例應用程序的代碼。GitHub 您在上一個步驟中下載了此程式碼。本範例示範如何使用 C 製作者程式庫，將資料夾內的 H.264 編碼視訊影格傳送 samples/h264SampleFrames 至 Kinesis 視訊串流。

此範例應用程式有三個部分：

- 初始化及設定：
  - 初始化及設定平台特定的媒體管道。
  - 初始化和配置 KinesisVideoClient 和管道，設置回調，集成特定 KinesisVideoStream 於案例的身份驗證，提取和提交編解碼器私有數據，以及使流進入 READY 狀態。
- 主要迴圈：
  - 從包含時間戳記及旗標的媒體管道取得影格。
  - 將框架提交到 KinesisVideoStream。
- 分解：
  - 停止 ( 同步 ) KinesisVideoStream，釋放 KinesisVideoStream，釋放 KinesisVideoClient。

此範例應用程式完成下列任務：

- 呼叫 createDefaultDeviceInfo API 以建立 deviceInfo 物件，其中包含裝置或儲存組態的相關資訊。

```
// default storage size is 128MB. Use setDeviceInfoStorageSize after create to change storage size.
CHK_STATUS(createDefaultDeviceInfo(&pDeviceInfo));
// adjust members of pDeviceInfo here if needed
pDeviceInfo->clientInfo.loggerLogLevel = LOG_LEVEL_DEBUG;
```

- 呼叫 createRealtimeVideoStreamInfoProvider API 以建立 StreamInfo 物件。

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

- 呼叫 `createDefaultCallbacksProviderWithAwsCredentials` API，根據靜態認AWS證建立預設回呼提供者。

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

- 呼叫 `createKinesisVideoClient` API 以建立包含裝置儲存體相關資訊的 `KinesisVideoClient` 物件，並維護回呼以報告 Kinesis Video Streams 事件。

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

- 呼叫 `createKinesisVideoStreamSync` API 以建立 `KinesisVideoStream` 物件。

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

- 設定範例影格並呼叫 `PutKinesisVideoFrame` API，將該影格傳送至 `KinesisVideoStream` 物件。

```
// setup sample frame
MEMSET(frameBuffer, 0x00, frameSize);
frame.frameData = frameBuffer;
frame.version = FRAME_CURRENT_VERSION;
frame.trackId = DEFAULT_VIDEO_TRACK_ID;
frame.duration = HUNDREDS_OF_NANOS_IN_A_SECOND / DEFAULT_FPS_VALUE;
frame.decodingTs = defaultGetTime(); // current time
```

```

frame.presentationTs = frame.decodingTs;

while(defaultGetTime() > streamStopTime) {
    frame.index = frameIndex;
    frame.flags = fileIndex % DEFAULT_KEY_FRAME_INTERVAL == 0 ?
FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    frame.size = sizeof(frameBuffer);

    CHK_STATUS(readFrameData(&frame, frameFilePath));

    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));
    defaultThreadSleep(frame.duration);

    frame.decodingTs += frame.duration;
    frame.presentationTs = frame.decodingTs;
    frameIndex++;
    fileIndex++;
    fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
}

```

- 分解：

```

CHK_STATUS(stopKinesisVideoStreamSync(streamHandle));
CHK_STATUS(freeKinesisVideoStream(&streamHandle));
CHK_STATUS(freeKinesisVideoClient(&clientHandle));

```

## 下一步驟

### [步驟 3：執程式碼並驗證程式碼](#)

## 步驟 3：執程式碼並驗證程式碼

若要執行和驗證 [Producer Library 程序](#) 的程式碼，請執行下列動作：

1. 運行以下命令以在[下載的 C SDK](#) 中創建一個build目錄，然後cmake從中啟動：

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



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

您可以將以下選項傳給 `cmake ..`

- `-DBUILD_DEPENDENCIES`-是否從源代碼構建依賴庫。
- `-DBUILD_TEST=TRUE`-構建單元和集成測試。可能有助於確認對您的設備的支持。

```
./tst/webrtc_client_test
```

- `-DCODE_COVERAGE`-啟用覆蓋報告。
- `-DCOMPILER_WARNINGS`-啟用所有編譯器警告。
- `-DADDRESS_SANITIZER`-建立與AddressSanitizer。
- `-DMEMORY_SANITIZER`-建立與MemorySanitizer。
- `-DTHREAD_SANITIZER`-建立與ThreadSanitizer。
- `-DUNDEFINED_BEHAVIOR_SANITIZER`-建立與UndefinedBehaviorSanitizer。
- `-DALIGNED_MEMORY_MODEL` - 僅針對對齊記憶體模型的裝置建立。預設值為 OFF。

2. 導航到您剛剛與上一步創建的build目錄，並運行make以構建 WebRTC C SDK 及其提供的示例。

```
make
```

3. 範例應用程式`kinesis_video_cproducer_video_only_sample`會將資料夾內的 h.264 編碼視訊影格傳送`samples/h264SampleFrames`至 Kinesis Video Streams。下列命令會將視訊影格以十秒的方式傳送至 Kinesis Video Streams：

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

如果您想要從其他資料夾傳送 H.264 編碼的影格 (例如，`MyH264FramesFolder`)，請使用下列引數執行範例：

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

- 若要啟用詳細日誌，請取消 CMakeList.txt 中適當的程式列來定義 HEAP\_DEBUG 和 LOG\_STREAMING C-defines。

在 IDE 中，您可以監控偵錯輸出測試套件的進度。您也可以透過在 Amazon CloudWatch 主控台中觀看與串流相關聯的指標來監控串流上的流量，例如 PutMedia.IncomingBytes。

#### Note

因為測試工具只傳送空位元組的影格，所以主控台不會將資料顯示為視訊串流。

## 在 Raspberry Pi 使用 C++ Producer 開發套件

Raspberry Pi 是一款小型且便宜的電腦，可用來教授和學習基本電腦程式設計技能。本教學說明如何在樹莓派裝置上設定和使用 Amazon Kinesis Video Streams C++ 生產者 SDK。此步驟同時包含如何使用 GStreamer 示範應用程式驗證安裝。

### 主題

- [必要條件](#)
- [建立具有寫入 Kinesis Video Streams 之權限的 IAM 使用者](#)
- [加入您的樹莓派到您的 Wi-Fi 網絡](#)
- [遠端 Connect 到您的樹莓派](#)
- [配置樹莓派相機](#)
- [安裝必要軟體](#)
- [下載並建置 Kinesis Video Streams C++ 製作者 SDK](#)
- [將影片串流至 Kinesis 視訊串流並檢視即時串流](#)

## 必要條件

在 Raspberry Pi 上設定 C++ Producer 開發套件之前，請確認您符合以下先決條件：

- 使用以下組態的 Raspberry Pi 裝置：
  - 機板版本：3 Model B 或更新版本。

- 連接的攝影機模組。
- 容量至少 8 GB 的 SD 卡。
- 已安裝 Raspbian 作業系統 (4.9 核心版或更新版本)。您可以下載最新的樹莓派操作系統 (以前稱為 Raspbian) 圖像從[樹莓派網站](#)。遵循 Raspberry Pi 指示將下載的映像安裝在 SD 卡上。
- AWS 帳戶 與 Kinesis 視頻流。如需更多資訊，請參閱 [Kinesis 視訊串流入門](#)。

#### Note

C++ 生產者 SDK 預設使用美國西部 (奧勒岡 us-west-2) () 區域。若要使用預設值，請在美國西部 (奧勒岡) 區域 AWS 區域 建立 Kinesis 視訊串流。

若要為 Kinesis 影片串流使用不同的區域，請執行下列其中一個動作：

- 將以下環境變數設為您的區域 (例如，*us-east-1*)：

```
export AWS_DEFAULT_REGION=us-east-1
```

## 建立具有寫入 Kinesis Video Streams 之權限的 IAM 使用者

如果您尚未這麼做，請設定具有寫入 Kinesis 影片串流權限的 AWS Identity and Access Management (IAM) 使用者。

這些程序旨在協助您快速開始使用 AWS 存取 key pair。裝置可以使用 X.509 憑證來 AWS IoT 連線。如[the section called “使用以下方式控制 Kinesis Video Streams 資源的存取 AWS IoT”](#)需如何將裝置設定為使用憑證型驗證的詳細資訊，請參閱。

1. 登入 AWS Management Console 並開啟 IAM 主控台，網址為 <https://console.aws.amazon.com/iam/>。
2. 在左側的導覽功能表中，選擇 Users (使用者)。
3. 若要建立新使用者，請選擇 Add user (新增使用者)。
4. 提供使用者的描述性使用者名稱，例如 **kinesis-video-raspberry-pi-producer**。
5. 在 Access type (存取類型) 下，選擇 Programmatic access (程式設計存取)。
6. 選擇 Next: Permissions (下一步：許可)。
7. 在 [設定 kinesis-video-raspberry-pi-producer 的權限] 底下，選擇 [直接附加現有原則]。
8. 選擇建立政策。Create policy (建立政策) 頁面在 web 瀏覽器的新索引標籤中開啟。

- 選擇 JSON 標籤。
- 將下列 JSON 政策複製並貼入文字區域。此原則授予使用者建立和寫入 Kinesis 影片串流資料的權限。

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

- 選擇檢閱政策。
- 為您的政策提供一個名稱，例如 **kinesis-video-stream-write-policy**。
- 選擇建立政策。
- 在瀏覽器中，返回 Add user (新增使用者) 索引標籤，然後選擇 Refresh (重新整理)。
- 在搜尋方塊中，輸入您建立之政策的名稱。
- 在清單中，選取新政策旁邊的核取方塊。
- 選擇 Next:Review (下一步：檢閱)。
- 選擇 Create user (建立使用者)。
- 主控台會顯示新使用者的 Access key ID (存取金鑰 ID)。請選擇 Show (顯示)，以顯示 Secret access key (私密存取金鑰)。記錄這些值；設定應用程式時需要它們。

## 加入您的樹莓派到您的 Wi-Fi 網絡

您可以透過無頭 (headless) 模式使用 Raspberry Pi，也就是無需連接鍵盤、螢幕或網路纜線。如果您使用連接的螢幕和鍵盤，請前往 [配置樹莓派相機](#)。

- 在您的電腦上，建立名為 wpa\_supplicant.conf 的檔案。

## 2. 複製下列文字並貼到 `wpa_supplicant.conf` 檔案中：

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
ssid="Your Wi-Fi SSID"
scan_ssid=1
key_mgmt=WPA-PSK
psk="Your Wi-Fi Password"
}
```

將 `ssid` 和 `psk` 值更換為 Wi-Fi 網路的資訊。

3. 將 `wpa_supplicant.conf` 檔案複製到 SD 卡。必須複製到 boot 磁碟區的根使用者。
4. 將 SD 卡插入 Raspberry Pi 並開機。它會加入您的 Wi-Fi 網路，且 SSH 已啟用。

## 遠端 Connect 到您的樹莓派

您可以透過無頭模式，遠端連接 Raspberry Pi。如果您使用連接螢幕和鍵盤的 Raspberry Pi，請前往 [配置樹莓派相機](#)。

1. 在遠端連接 Raspberry Pi 裝置之前，請執行以下其中一項操作以判斷其 IP 地址：
  - 如果您可以存取網路的 Wi-Fi 路由器，請查看連接的 Wi-Fi 裝置。尋找名為 Raspberry Pi 的裝置，以找到您裝置的 IP 地址。
  - 如果您無法存取網路的 Wi-Fi 路由器，可以使用其他軟體來尋找網路上的裝置。[Fing](#) 是一款常見的應用程式，同時適用於 Android 和 iOS 裝置。您可以使用此應用程式的免費版本，在網路上尋找裝置的 IP 地址。
2. 當您知道 Raspberry Pi 裝置的 IP 地址之後，就可以使用任何終端應用程式進行連接。
  - 在 macOS 或 Linux 上，請使用 `ssh`：

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

- 在 Windows 上，請使用 [PuTTY](#)，其為適用於 Windows 的免費 SSH 用戶端。

新安裝的 Raspbian 中，使用者名稱為 `pi`，密碼為 `raspberry`。建議您 [變更預設密碼](#)。

## 配置樹莓派相機

請依照下列步驟設定 Raspberry Pi 攝影機，將視訊從裝置傳送至 Kinesis 視訊串流。

1. 使用下列命令開啟編輯器來更新 `modules` 檔案：

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

2. 如果檔案結尾沒有下列內容，請將它加入：

```
bcm2835-v4l2
```

3. 儲存檔案，然後結束編輯器 (Ctrl-X)。
4. 重新啟動 Raspberry Pi：

```
$ sudo reboot
```

5. 裝置重新啟動時，如果您是遠端連接，請透過終端應用程式再次連接。
6. Open `raspi-config`:

```
$ sudo raspi-config
```

7. 選擇「介面選項」>「舊式相機」。在較舊版本的 Raspbian 操作系統中，此菜單選項可能位於接口選項，攝像頭下。

如果沒有啟用攝影機則將其啟用，並在出現提示訊息時重新啟動。

8. 輸入下列命令，驗證攝影機是否正常運作：

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

如果您的相機配置正確，此指令會從相機擷取影像，將其儲存到名為的檔案中 `test.jpg`，並顯示資訊訊息。

## 安裝必要軟體

C++ Producer 開發套件需要您在 Raspberry Pi 安裝以下軟體先決條件。

1. 更新套件清單並安裝建置 SDK 所需的程式庫。鍵入以下命令：

```
$ sudo apt update
$ sudo apt install -y \
  automake \
  build-essential \
  cmake \
  git \
  gstreamer1.0-plugins-base-apps \
  gstreamer1.0-plugins-bad \
  gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-ugly \
  gstreamer1.0-tools \
  gstreamer1.0-omx-generic \
  libcurl4-openssl-dev \
  libgstreamer1.0-dev \
  libgstreamer-plugins-base1.0-dev \
  liblog4cplus-dev \
  libssl-dev \
  pkg-config
```

2. 將以下 PEM 檔案複製到 `/etc/ssl/cert.pem` :

```
$ sudo curl https://www.amazontrust.com/repository/AmazonRootCA1.pem -o /etc/ssl/
AmazonRootCA1.pem
$ sudo chmod 644 /etc/ssl/AmazonRootCA1.pem
```

## 下載並建置 Kinesis Video Streams C++ 製作者 SDK

您可以使用下列程序下載並建置 Kinesis Video Streams C++ 製作者 SDK。這種方法所花費的建置時間較長，視網路連線能力和處理器速度而定。

1. 下載軟體開發套件。類型：

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

2. 準備一個構建目錄。類型：

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

3. 建置 SDK 和範例應用程式。根據您要構建的 Raspberry Pi 模型，首次運行可能需要幾個小時：

```
$ cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=FALSE
$ make
```

## 將影片串流至 Kinesis 視訊串流並檢視即時串流

1. 若要執行範本應用程式，您需要以下資訊：

- 在 [必要條件](#) 區段中建立之串流的名稱。
- 在 [建立具有寫入 Kinesis Video Streams 之權限的 IAM 使用者](#) 中建立的帳戶登入資料 (存取金鑰 ID 和私密存取金鑰)。

2. 使用下列命令執行範例應用程式。將預留位置取代為您環境的值。

```
$ export GST_PLUGIN_PATH=Directory Where You Cloned the SDK/amazon-kinesis-video-streams-producer-sdk-cpp/build
$ export AWS_DEFAULT_REGION=AWS Region i.e. us-east-1
$ export AWS_ACCESS_KEY_ID=Access Key ID
$ export AWS_SECRET_ACCESS_KEY=Secret Access Key
$ ./kvs_gstreamer_sample Your Stream Name
```

3. 如果範例應用程式因 library not found 錯誤而結束，請輸入下列命令，以確認專案已正確連結至其開放原始碼相依性：

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

4. 開啟 [Kinesis Video Streams 主控台](#)。
5. 選擇您建立之串流的 Stream name (串流名稱)。

從 Raspberry Pi 傳送的視訊串流，會出現在主控台。

播放串流時，您可以嘗試 Kinesis Video Streams 主控台的下列功能：

- 在 Video preview (視訊預覽) 區段中，請使用導覽控制，以倒轉或向前快轉串流。
- 在 Stream info (串流資訊) 區段中，請注意串流的轉碼器、解析度和位元傳輸率。此教學課程特意在 Raspberry Pi 上調低解析度和位元傳輸率的值，將頻寬使用量降到最低。若要檢視為串流建立的 Amazon CloudWatch 指標，請選擇在中檢視串流指標 CloudWatch。



- 在 Data retention period (資料保留期間) 下，請注意視訊串流的保留期間為 1 天。您可以編輯此值，將其設定為 No data retention (無資料保留)，或將值設定為從一天到數年。

在伺服器端加密下，請注意您的資料正在使用 AWS Key Management Service (AWS KMS) 維護的金鑰進行靜態加密。

## 製作者開發套件參考

本節包含 [室壁運動視頻流生產者庫](#) 的限制、錯誤碼和其他參考資訊。

### 主題

- [製作者 SDK 限制](#)
- [錯誤代碼參考](#)
- [網路抽象層 \(NAL\) 適應性旗標參考](#)
- [製作者 SDK 結構](#)
- [Kinesis 視頻流結構](#)
- [生產者 SDK 回呼](#)

## 製作者 SDK 限制

下表包含 [Producer 資料庫](#) 中目前對於各項數值的限制。

### Note

設定這些值之前，必須驗證您的輸入。開發套件不驗證這些限制，如果超過限制則執行時間會發生錯誤。

Value	限制	備註
最大的串流計數	128	製作者物件可建立的最高串流數量。此為軟限制 (您可以請求提高)。它保證生產者不會意外地遞歸地創建流。
最高裝置名稱長度	128 個字元	

Value	限制	備註
最高標籤計數	每個串流 50 個	
最高串流名稱長度	256 個字元	
最低儲存空間大小	10 MiB = 10 * 1024 * 1024 位元	
最高儲存空間大小	10 GiB = 10 * 1024 * 1024 * 1024 位元	
最高根目錄路徑長度	4,096 個字元	
最高授權資訊長度	10,000 位元組	
最高 URI 字串長度	10,000 個字元	
最高標籤名稱長度	128 個字元	
最高標籤數值長度	1,024 個字元	
最短安全字符期間	30 秒	
安全字符寬限期間	40 分鐘	如果指定的持續時間較長，則僅限於此值。
保留期間	0 或大於 1 小時	0 代表沒有保留。
最小叢集持續時間	1 秒鐘	該值是以 100 奈秒 (ns) 為單位表示，採用軟體開發套件標準。
最大叢集持續時間	30 秒	該值是以 100 奈秒 (ns) 為單位表示，採用軟體開發套件標準。後端 API 可以強制執行較短的叢集持續時間。
最大區段大小	50 MB	如需詳細資訊，請參閱 <a href="#">Kinesis Video Streams 服務配額</a> 。

Value	限制	備註
最長區段持續時間	20 秒	如需詳細資訊，請參閱 <a href="#">Kinesis Video Streams 服務配額</a> 。
最長連線持續時間	45 分鐘	過了此時間後，後端將關閉連線。軟體開發套件在該時間內輪換字符並建立新連接。
最常 ACK 區段長度	1,024 個字元	傳送到 ACK 解析器函數的最高確認區段長度。
最長內容類型字串長度	128 個字元	
最大轉碼器 ID 字串長度	32 個字元	
最大音軌名稱字串長度	32 個字元	
最大轉碼器私有資料長度	1 MiB = 1 * 1024 * 1024 位元	
最小時間碼比例值長度	100 奈秒 (ns)	在產生的 MKV 叢集中表示影格時間戳記的最小時間碼比例值。該值是以 100 奈秒 (ns) 為增加幅度，採用軟體開發套件標準。
最大時間碼比例值長度	1 秒鐘	在產生的 MKV 叢集中表示影格時間戳記的最大時間碼比例值。該值是以 100 奈秒 (ns) 為增加幅度，採用軟體開發套件標準。
最小內容檢視項目計數	10	
最短緩衝持續時間	20 秒	該值是以 100 奈秒 (ns) 為增加幅度，採用軟體開發套件標準。
最大更新版本長度	128 個字元	

Value	限制	備註
最大 ARN 長度	1024 個字元	
最大區段序列長度	128 個字元	
最高保留期間	10 年	

## 錯誤代碼參考

本節包含 [Producer 資料庫](#) 的錯誤碼與狀態碼資訊。

有關常見問題的解決方法，請參閱 [Kinesis Video Streams 疑難排解](#)。

### 主題

- [回 PutFrame 呼傳回的錯誤和狀態碼-平台獨立程式碼 \(PIC\)](#)
- [回 PutFrame 調返回的錯誤和狀態碼-C 生產者庫](#)

## 回 PutFrame 呼傳回的錯誤和狀態碼-平台獨立程式碼 (PIC)

下列各節包含「平台獨立程式碼」(PIC) 內PutFrame作業的回呼所傳回的錯誤與狀態資訊。

### 主題

- [用戶端程式庫傳回的錯誤和狀態碼](#)
- [持續時間庫返回的錯誤和狀態碼](#)
- [通用庫返回的錯誤和狀態碼](#)
- [堆積程式庫傳回的錯誤和狀態碼](#)
- [MKVGen 程式庫傳回的錯誤和狀態碼](#)
- [追蹤程式庫傳回的錯誤和狀態碼](#)
- [Utils 程式庫傳回的錯誤和狀態碼](#)
- [視圖庫返回的錯誤和狀態碼](#)

## 用戶端程式庫傳回的錯誤和狀態碼

下表包含 Kinesis 視訊串流Client程庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息	描述	建議的動作
0x52000001	STATUS_MAX_STREAM_COUNT	已達串流次數上限。	請依照 <a href="#">製作者 SDK 限制</a> 中說明之方式在 DeviceInfo 中指定較大的最高串流計數。
0x52000002	STATUS_MIN_STREAM_COUNT	最低串流計數錯誤。	在中指定大於零的最大串流數 DeviceInfo 。
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	無效裝置名稱長度。	請參閱中指定的最大裝置名稱長度 (以字元為單位) <a href="#">製作者 SDK 限制</a> 。
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	無效的 DeviceInfo 架構版本。	指定正確的目前架構版本。
0x52000005	STATUS_MAX_TAG_COUNT	已達標籤次數上限。	請參閱在中指定的當前最大標籤計數 <a href="#">製作者 SDK 限制</a> 。
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	無效的 Callbacks 架構版本。	指定正確的目前架構版本。
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	無效的 StreamInfo 架構版本。	指定正確的目前架構版本。
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	無效串流名稱長度。	請參閱中指定的最大串流名稱長度 (以字元表示) <a href="#">製作者 SDK 限制</a> 。

代碼	訊息	描述	建議的動作
0x5200000a	STATUS_INVALID_STORAGE_SIZE	指定了無效的儲存空間大小。	以位元表示的儲存空間大小需介於 <a href="#">製作者 SDK 限制</a> 中所指定的上限內。
0x5200000b	STATUS_INVALID_ROOT_DIRECTORY_LENGTH	無效的根目錄字串長度。	請參閱中指定的最大根目錄路徑長度 <a href="#">製作者 SDK 限制</a> 。
0x5200000c	STATUS_INVALID_SPILL_RATIO	無效的溢出率。	以 0-100 之間的百分比表示溢出率。
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	無效的 StorageInfo 架構版本。	指定正確的目前架構版本。
0x5200000e	STATUS_INVALID_STREAM_STATE	串流的狀態不允許目前的操作。	最常見的是，當 SDK 無法達到執行請求操作所需的狀態時，就會發生此錯誤。例如，如果 GetStreamingEndpoint API 呼叫失敗，且用戶端應用程式忽略該失敗並繼續將影格放入串流中，則會出現該錯誤。
0x5200000f	STATUS_SERVICE_CALLBACK_MISSING	對於部分必要函數來說，Callbacks 架構含有遺失的函數入口點。	驗證強制回調是否在客戶端應用程式中實現。此錯誤僅暴露於平台獨立代碼 ( PIC ) 客戶端。C++ 和其他更高級別的包裝函式可滿足這些呼叫的要求。

代碼	訊息	描述	建議的動作
0x52000010	STATUS_SERVICE_CALL_NOT_AUTHORIZED_ERROR	未獲授權。	驗證安全令牌，證書，安全令牌集成和過期。驗證令牌具有正確的相關權限。對於 Kinesis Video Streams 範例應用程式，請確認環境變數設定正確。
0x52000011	STATUS_DESCRIBESTREAM_CALL_FAILED	DescribeStream API 失敗。	此錯誤會在 DescribeStream API 重試失敗後傳回。PIC 客戶端在停止重試後返回此錯誤。
0x52000012	STATUS_INVALID_DESCRIBESTREAM_RESPONSE	無效的 DescribeStreamResponse 架構。	傳遞到 DescribeStreamResultEvent 的架構會是 null 或者包含無效項目，例如 null 的 Amazon Resource Name (ARN)。
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	正在刪除的串流。	刪除串流時導致 API 失敗。確認串流使用中時，沒有其他處理序嘗試刪除串流。
0x52000014	STATUS_SERVICE_CALL_INVALID_ARG_ERROR	服務呼叫指定了無效引數。	當服務呼叫引數無效或 SDK 遇到無法解譯的錯誤時，後端會傳回此錯誤。
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	找不到該裝置。	確認裝置在使用中未遭到刪除。

代碼	訊息	描述	建議的動作
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	該裝置未佈建。	確認已佈建裝置。
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	找不到該服務傳回的一般資源。	在服務找不到資源 (例如, 串流) 時, 將會出現該錯誤。它可能在不同上下文中具有不同的含意, 但可能的原因是在建立串流之前使用 API。使用 SDK 可確認串流是先建立的。
0x52000018	STATUS_INVALID_AUTH_LEN	無效授權資訊長度。	請參考 <a href="#">製作者 SDK 限制</a> 中指定的目前數值。
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	CreateStream API 呼叫失敗。	有關操作失敗原因的詳細資訊, 請參考錯誤字串。
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	GetStreamingToken 呼叫失敗。	有關操作失敗原因的詳細資訊, 請參考錯誤字串。
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	GetStreamingEndpoint API 呼叫失敗。	有關操作失敗原因的詳細資訊, 請參考錯誤字串。
0x5200002c	STATUS_INVALID_URI_LEN	從 GetStreamingEndpoint API 傳回了無效的 URI 字串長度。	請參考 <a href="#">製作者 SDK 限制</a> 中指定的目前最高數值。



代碼	訊息	描述	建議的動作
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	PutMedia API 呼叫失敗。	有關操作失敗原因的詳細資訊，請參考錯誤字串。
0x5200002e	STATUS_STORE_OUT_OF_MEMORY	內容存放區大於記憶體容量。	內容存放區是在串流之間共用的，應具有足夠的容量以儲存所有串流的最大持續時間內的內容 + ~20% (考慮磁碟重組)。請勿超出儲存空間。請選擇與累計儲存空間大小和容許的延遲所對應的每個串流的最大持續時間值。我們建議您在框架從內容視圖窗口中掉下來，而不是僅僅放置 (內容存儲內存壓力)。這是因為刪除幀會啟動流壓力通知回調。然後，應用程式可以調整上游媒體元件 (如編碼器) 以降低位元速率，刪除影格或執行對應的操作。
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	目前的串流沒有其他可用資料。	在媒體管道產生影格的速度比網路執行緒使用傳送到服務的影格的速度慢時，這可能是一個有效的結果。較高級別的客戶端 (例如 C++，Java 或 Android) 不會看到此警告，因為它是在內部處理的。

代碼	訊息	描述	建議的動作
0x52000030	STATUS_INVALID_TAG_VERSION	無效的 Tag 架構版本。	指定正確的目前架構版本。
0x52000031	STATUS_SERVICE_CALL_UNKNOWN_ERROR	從網路堆疊中傳回未知或常規錯誤。	有關更詳細資訊，請參閱記錄。
0x52000032	STATUS_SERVICE_CALL_RESOURCE_IN_USE_ERROR	資源使用中。	從服務傳回。如需詳細資訊，請參閱 Kinesis Video Streams API 參考資料。
0x52000033	STATUS_SERVICE_CALL_CLIENT_LIMIT_ERROR	客戶端限制。	從服務傳回。如需詳細資訊，請參閱 Kinesis Video Streams API 參考資料。
0x52000034	STATUS_SERVICE_CALL_DEVICE_LIMIT_ERROR	裝置限制。	從服務傳回。如需詳細資訊，請參閱 Kinesis Video Streams API 參考資料。
0x52000035	STATUS_SERVICE_CALL_STREAM_LIMIT_ERROR	串流限制。	從服務傳回。如需詳細資訊，請參閱 Kinesis Video Streams API 參考資料。
0x52000036	STATUS_SERVICE_CALL_RESOURCE_DELETED_ERROR	已刪除或正在刪除資源。	從服務傳回。如需詳細資訊，請參閱 Kinesis Video Streams API 參考資料。

代碼	訊息	描述	建議的動作
0x52000037	STATUS_SERVICE_CALL_TIMEOUT_ERROR	服務呼叫逾時。	呼叫特定服務 API 產生逾時。請確認您擁有有效的網路連線。PIC 會自動重試操作。
0x52000038	STATUS_STREAM_READ_CALLBACK_FAILED	串流就緒通知。	該通知從 PIC 發送到客戶端，表示已建立非同步串流。
0x52000039	STATUS_DESERVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	指定了無效的標籤。	標籤計數不是零，但標籤是空的。確認已指定標籤或計數為零。
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	無效的 StreamDescription 架構版本。	指定正確的目前架構版本。
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	無效標籤名稱長度。	請參考 <a href="#">製作者 SDK 限制</a> 中指定的標籤名稱限制。
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	無效標籤值長度。	請參考 <a href="#">製作者 SDK 限制</a> 中指定的標籤值限制。
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	TagResource API 失敗。	TagResource API 呼叫失敗。檢查是否有有效的網路連線。查看關於操作失敗的記錄詳細資訊。

代碼	訊息	描述	建議的動作
0x5200003e	STATUS_INVALID_CUSTOM_DATA	無效的自訂資料呼叫 PIC API。	無效的自訂資料已在對 PIC API 的呼叫中指定。此情況只會在直接使用 PIC 的客戶端發生。
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	無效的 CreateStreamResponse 架構。	架構或其成員欄位無效 (也就是 ARN 為 null 或大於 <a href="#">製作者 SDK 限制</a> 中指定的值)。
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	客戶端授權失敗。	在多次重試後，PIC 無法獲得適當的身份驗證信息 ( AccessKeyId 或 SecretAccessKey )。檢查身分驗證整合。範例應用程式使用環境變數將登入資料資訊傳入到 C++ Producer Library。
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	取得安全字符合呼叫失敗。	此情況只會在直接使用 PIC 的客戶端發生。在重試一定次數後，呼叫失敗並出現該錯誤。
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	佈建錯誤。	佈建未實作。
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	無法建立製作者客戶端。	在客戶端建立失敗時，在重試一定次數後，PIC 會傳回一個一般錯誤。

代碼	訊息	描述	建議的動作
0x52000044	STATUS_CLIENT_READY_CALLBACK_FAILED	無法讓製作者用戶端成為「就緒」(Ready) 狀態。	若 PIC 無法移動為「就緒」(Ready) 狀態，將由 PIC 狀態機器傳回。查看關於根本原因的記錄詳細資訊。
0x52000045	STATUS_TAG_CLIENT_CALL_FAILED	製作者客戶端的 TagResource 失敗。	製作者客戶端的 TagResource API 呼叫失敗。查看關於根本原因的記錄詳細資訊。
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	裝置/製作者建立失敗。	更高級別的 SDK ( 例如 C++ 或 Java ) 尚未實現設備或生產者創建 API。直接使用 PIC 的客戶端可以使用結果通知來顯示操作失敗。
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	在視圖中不顯示收到的 ACK 的時間戳記。	如果與收到的 ACK 對應的影格未位於內容視圖視窗中，則會出現該錯誤。通常，如果 ACK 傳送速率較慢就會發生這種情況。可以將其解釋為警告，並顯示下行線路速度較慢。
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	無效的 FragmentAck 架構版本。	指定正確的目前 FragmentAck 架構版本。

代碼	訊息	描述	建議的動作
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	無效的安全字串過期。	安全令牌到期日的 future 應具有絕對時間戳記，該時間戳記大於當前時間戳，並具有寬限期。對於寬限期的限制，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x5200004a	STATUS_END_OF_STREAM	串流結束 (EOS) 指標。	在 GetStreamData API 呼叫中，表示目前上傳處理工作階段已結束。如果工作階段結束或出現錯誤，或者工作階段字串已過期並且正在輪換工作階段，則會發生這種情況。
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	重複的串流名稱。	多個串流不能具有相同的串流名稱。請為串流選擇專屬名稱。
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	無效保留期間。	StreamInfo 結構中指定了無效的保留期。關於保留期數值的有效範圍資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x5200004d	STATUS_INVALID_ACK_KEY_START	無效的 FragmentAck 。	無法解析區段 ACK 字串。無效的金鑰開始指標。區段 ACK 字串可能已損壞。它可以自行更正，並且可以將該錯誤視為警告。

代碼	訊息	描述	建議的動作
0x5200004e	STATUS_INVALID_ACK_DUPLICATE_KEY_NAME	無效的 FragmentAck 。	無法解析區段 ACK 字串。多個金鑰有相同的名稱。區段 ACK 字串可能已損壞。它可以自行更正，並且可以將該錯誤視為警告。
0x5200004f	STATUS_INVALID_ACK_INVALID_VALUE_START	無效的 FragmentAck 。	因為無效的金鑰值開始指標而無法解析區段 ACK 字串。區段 ACK 字串可能已損壞。它可以自行更正，並且可以將該錯誤視為警告。
0x52000050	STATUS_INVALID_ACK_INVALID_VALUE_END	無效的 FragmentAck 。	因為無效的金鑰值結束指標而無法解析區段 ACK 字串。區段 ACK 字串可能已損壞。它可以自行更正，並且可以將該錯誤視為警告。
0x52000051	STATUS_INVALID_PARSED_ACK_TYPE	無效的 FragmentAck 。	因指定了無效的 ACK 類型，無法解析區段 ACK 字串。
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	串流已停止。	串流已停止，但影格仍放入串流中。
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	無效的 StreamMetrics 架構版本。	指定正確的目前 StreamMetrics 架構版本。

代碼	訊息	描述	建議的動作
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	無效的 ClientMetrics 架構版本。	指定正確的目前 ClientMetrics 架構版本。
0x52000055	STATUS_INVALID_READY_STATE	製作者初始化無法達到「就緒」(Ready) 狀態。	無法在製作者客戶端初始化期間達到「就緒」(Ready) 狀態。查看記錄了解更多資訊。
0x52000056	STATUS_MACHINE_STATE_NOT_FOUND	內部狀態機器錯誤。	不是公開顯示的錯誤。
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	FragmentAck 結構中指定的無效 ACK 類型。	FragmentAck 結構應包含公開標題中定義的 ACK 類型。
0x52000058	STATUS_INVALID_STREAM_READY_STATE	內部狀態機器轉移錯誤。	不是公開顯示的錯誤。
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	在釋放建立者後釋放串流對象。	嘗試在釋放製作者物件後釋放串流物件。只有在直接使用 PIC 的客戶端中可能會發生這種情況。
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	內部儲存空間錯誤。	內部錯誤，指出內容存放區的實際配置大小小於封裝框架和片段的大小。



代碼	訊息	描述	建議的動作
0x5200005b	STATUS_VI EW_ITEM_S IZE_GREAT ER_THAN_A LLOCATION	內部儲存空間錯誤。	分配在內容視圖的儲存大小大於內容存放區中的分配大小。
0x5200005c	STATUS_AC K_ERR_STR EAM_READ_ERROR	串流讀取錯誤 ACK。	ACK 從後端傳回的錯誤，表示串流讀取或剖析錯誤。在後端無法擷取串流時，通常會發生這種情況。自動重新資料串流通常可以修正該錯誤。
0x5200005d	STATUS_AC K_ERR_FRA GMENT_SIZ E_REACHED	已達最高區段大小上限。	以位元表示的最高區段大小定義於 <a href="#">製作者 SDK 限制</a> 中。該錯誤表示具有非常大的影格，或者沒有用於建立可管理大小的區段的關鍵影格。檢查編碼器設置並確認關鍵幀是否正確生產。對於具有較高密度的串流，請將編碼器配置以較小持續時間產生的區段以管理最大大小。
0x5200005e	STATUS_AC K_ERR_FRA GMENT_DUR ATION_REACHED	已達最高區段持續時間上限。	最高區段持續時間定義於 <a href="#">製作者 SDK 限制</a> 中。該錯誤表示每秒具有非常小的影格，或者沒有用於建立可管理持續時間的區段的關鍵影格。檢查編碼器設置並確認是否定期正確生產關鍵幀。

代碼	訊息	描述	建議的動作
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	已達最高連線持續時間上限。	Kinesis Video Streams 會強制執行中所指定的最大連線持續時間。 <a href="#">製作者 SDK 限制</a> 在達到最大值之前，生產者 SDK 會自動旋轉流或令牌。使用 SDK 的客戶端不應收到此錯誤。
0x52000060	STATUS_ACK_ERR_FRAGMENT_TIMECODE_NOT_MONOTONIC	時間碼並非以單調遞增。	生產者 SDK 會強制執行時間戳記，因此使用 SDK 的用戶端不應該收到此錯誤。
0x52000061	STATUS_ACK_ERR_MULTITRACK_MKV	在 MKV 中的多個音軌。	生產者 SDK 會強制執行單軌串流，因此使用 SDK 的用戶端不應該收到此錯誤。
0x52000062	STATUS_ACK_ERR_INVALID_MKV_DATA	無效的 MKV 資料。	後端 MKV 解析器在解析串流時遇到錯誤。如果資料流在轉換過程中損毀，使用 SDK 的用戶端可能會遇到此錯誤。如果緩衝壓力迫使 SDK 丟棄部分傳輸的尾部框架，也可能發生這種情況。在後一種情況下，我們建議您降低 FPS 和解析度、增加壓縮率，或者 (如果有「爆發」網路) 允許較大的內容存放區和緩衝期間，以因應暫時的壓力。

代碼	訊息	描述	建議的動作
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	無效的製作者時間戳記。	如果製作者時鐘大幅漂移到將來的時間，服務將傳回該錯誤 ACK。更高級別的軟體開發套件 (例如，Java 或 C++) 使用某種版本的系統時鐘以符合從 PIC 中回呼目前時間的要求。確認系統時鐘設定正確。直接使用 PIC 的客戶端應驗證其回調函數是否返回正確的時間戳。
0x52000064	STATUS_ACK_ERR_STREAM_NOT_ACTIVE	閒置串流。	在串流未處於「使用中」狀態時，對後端 API 進行呼叫。如果客戶端建立串流並立即繼續將影格推入串流中，則會發生這種情況。軟體開發套件使用狀態機器和復原機制來處理這種情況。
0x52000065	STATUS_ACK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS 存取遭拒錯誤。	當帳戶沒有存取特定金鑰的權限時傳回。
0x52000066	STATUS_ACK_ERR_KMS_KEY_DISABLED	AWS KMS 金鑰已停用。	指定的金鑰已停用。
0x52000067	STATUS_ACK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS 金鑰驗證錯誤。	一般驗證錯誤。如需詳細資訊，請參閱 <a href="#">AWS Key Management Service API 參考</a> 。

代碼	訊息	描述	建議的動作
0x52000068	STATUS_ACK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key不可用。	金鑰無法使用。如需詳細資訊，請參閱 <a href="#">AWS Key Management Service API 參考</a> 。
0x52000069	STATUS_ACK_ERR_KMS_KEY_INVALID_USAGE	無效的 KMS 金鑰使用。	未配置為在此前後關聯中使用。AWS KMS key如需詳細資訊，請參閱 <a href="#">AWS Key Management Service API 參考</a> 。
0x5200006a	STATUS_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS 無效狀態。	如需詳細資訊，請參閱 <a href="#">AWS Key Management Service API 參考</a> 。
0x5200006b	STATUS_ACK_ERR_KMS_KEY_NOT_FOUND	找不到 KMS 金鑰。	找不到金鑰。如需詳細資訊，請參閱 <a href="#">AWS Key Management Service API 參考</a> 。
0x5200006c	STATUS_ACK_ERR_STREAM_DELETED	串流已被刪除或者正在刪除。	串流正由其他應用程式刪除或者透過 AWS Management Console 刪除。
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	內部錯誤。	一般服務內部錯誤。

代碼	訊息	描述	建議的動作
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARC_HIVAL_ERROR	區段存檔錯誤。	在服務無法長期保留區段並為其編制索引時傳回。雖然非常少見，但可能會因為各種原因而發生此情況。預設情況下，軟體開發套件會再次嘗試發送區段。
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	不明錯誤。	此服務傳回了一個不明錯誤。
0x52000070	STATUS_MISSING_ERR_ACK_ID	遺失的 ACK 資訊。	ACK 解析器已完成解析，但是仍缺少 FragmentAck 資訊。
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	無效的 ACK 區段長度。	指定給 ACK 解析器的 ACK 區段字串為無效長度。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x52000074	STATUS_MAX_FRAGMENT_METADATA_COUNT	片段中繼資料項目新增數量已達上限。	Kinesis 視訊串流最多可將 10 個中繼資料項目新增至片段，方法是將非永久性項目新增至片段，或是將永久性項目新增至中繼資料佇列。如需詳細資訊，請參閱 <a href="#">搭配 Kinesis 視訊串流使用串流中繼資料</a> 。

代碼	訊息	描述	建議的動作
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	已達上限 (中繼資料計數上限、中繼資料名稱長度上限、中繼資料數值長度上限)。	製作者開發套件會限制中繼資料項目的名稱與大小。除非生產者 SDK 程式碼中的限制已變更，否則不會發生此錯誤。如需詳細資訊，請參閱 <a href="#">搭配 Kinesis 視訊串流使用串流中繼資料</a> 。
0x52000076	STATUS_BLOCKING_INTERRUPTED_STREAM_TERMINATED	不實作。	
0x52000077	STATUS_INVALID_METADATA_NAME	中繼資料名稱無效。	中繼資料名稱不能以字串 "AWS" 開頭。如果發生此錯誤，則不會將中繼資料項目新增至片段或中繼資料佇列。如需詳細資訊，請參閱 <a href="#">搭配 Kinesis 視訊串流使用串流中繼資料</a> 。
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	片段影格的結尾為無效狀態。	片段的末尾不應該在 non-key-frame 碎片流中發送。
0x52000079	STATUS_TRACK_INFO_MISSING	遺失軌道資訊。	曲目編號必須大於零，並與曲目 ID 相符。
0x5200007a	STATUS_MAX_TRACK_COUNT_EXCEEDED	超過最大軌道數量。	每個串流最多可以有三個音軌。

代碼	訊息	描述	建議的動作
0x5200007b	STATUS_OF FLINE_MOD E_WITH_ZE RO_RETENTION	離線串流模式保留時間設定為零。	離線串流模式的保留時間不應設為零。
0x5200007c	STATUS_AC K_ERR_TRA CK_NUMBER _MISMATCH	錯誤 ACK 的軌道編號不符。	
0x5200007d	STATUS_AC K_ERR_FRA MES_MISSI NG_FOR_TRACK	軌道遺漏影格。	
0x5200007e	STATUS_AC K_ERR_MOR E_THAN_AL LOWED_TRA CKS_FOUND	超過允許的軌道數量上限。	
0x5200007f	STATUS_UP LOAD_HAN DLE_ABORTED	上傳處理中止。	
0x52000080	STATUS_IN VALID_CER T_PATH_LENGTH	無效的憑證路徑長度。	
0x52000081	STATUS_DU PLICATE_T RACK_ID_FOUND	找到重複的軌道 ID。	
0x52000082	STATUS_IN VALID_CLI ENT_INFO_VERSION		

代碼	訊息	描述	建議的動作
0x52000083	STATUS_INVALID_CLIENT_ID_STRING_LENGTH		
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAX_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUMEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		



代碼	訊息	描述	建議的動作
0x52000089	STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS		
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_METADATA_VALUE_LENGTH		

### 持續時間庫返回的錯誤和狀態碼

下表包含Duration程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

## 通用庫返回的錯誤和狀態碼

下表包含Common程式庫中方法所傳回的錯誤和狀態資訊。

### Note

這些錯誤與狀態資訊代碼常見於許多 API。

代碼	訊息	描述
0x00000001	STATUS_NULL_ARG	為必要引數傳遞 NULL。
0x00000002	STATUS_INVALID_ARG	為引數指定了無效的值。
0x00000003	STATUS_INVALID_ARG_LEN	指定了無效的引數長度。
0x00000004	STATUS_NOT_ENOUGH_MEMORY	無法配置足夠的記憶體。
0x00000005	STATUS_BUFFER_TOO_SMALL	指定的緩衝大小太小。
0x00000006	STATUS_UNEXPECTED_EOF	達到未預期的檔案結尾。
0x00000007	STATUS_FORMAT_ERROR	遇到無效的格式。
0x00000008	STATUS_INVALID_HANDLE_ERROR	無效的處理常式值。
0x00000009	STATUS_OPEN_FILE_FAILED	無法開啟檔案。

代碼	訊息	描述
0x0000000a	STATUS_READ_FILE_FAILED	無法從檔案讀取。
0x0000000b	STATUS_WRITE_TO_FILE_FAILED	無法寫入檔案。
0x0000000c	STATUS_INTERNAL_ERROR	通常不會出現的內部錯誤，可能表示軟體開發套件或服務 API 錯誤。
0x0000000d	STATUS_INVALID_OPERATION	具有無效的操作，或者不允許執行該操作。
0x0000000e	STATUS_NOT_IMPLEMENTED	未套用功能。
0x0000000f	STATUS_OPERATION_TIMED_OUT	此操作逾時。
0x00000010	STATUS_NOT_FOUND	找不到需要的資源。
0x00000011	STATUS_CREATE_THREAD_FAILED	無法建立執行緒。
0x00000012	STATUS_THREAD_NOT_ENOUGH_RESOURCES	資源不足而無法建立另一個執行緒，或遇到系統對執行緒數目的限制。
0x00000013	STATUS_THREAD_INVALID_ARG	指定了無效的線程屬性，或者另一個線程已經在等待與此線程加入。
0x00000014	STATUS_THREAD_PERMISSIONS	沒有權限可設定執行緒屬性中指定的排程原則和參數。
0x00000015	STATUS_THREAD_DEADLOCKED	檢測到死鎖或加入線程指定調用線程。

代碼	訊息	描述
0x00000016	STATUS_THREAD_DOES_NOT_EXIST	找不到具有指定執行緒 ID 的執行緒。
0x00000017	STATUS_JOIN_THREAD_FAILED	執行緒連接作業傳回未知或一般錯誤。
0x00000018	STATUS_WAIT_FAILED	超過等待條件變數的最長時間。
0x00000019	STATUS_CANCEL_THREAD_FAILED	執行緒取消作業傳回未知或一般錯誤。
0x0000001A	STATUS_THREAD_IS_NOT_JOINABLE	在不可連接的線程上請求線程連接操作。
0x0000001b	STATUS_DETACH_THREAD_FAILED	執行緒分離作業傳回未知或一般錯誤。
0x0000001c	STATUS_THREAD_ATTR_INIT_FAILED	無法初始化執行緒屬性物件。
0x0000001D	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	無法設定執行緒屬性物件的堆疊大小。
0x0000001e	STATUS_MEMORY_NOT_FREED	僅在測試中使用。表示並非所有要求的記憶體都已釋放。
0x10000015	STATUS_INVALID_ALLLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

## 堆積程式庫傳回的錯誤和狀態碼

下表包含Heap程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息	描述
0x01000001	STATUS_HEAP_FLAGS_ERROR	指定的標籤組合無效。
0x01000002	STATUS_HEAP_NOT_INITIALIZED	堆積初始化前已嘗試操作。
0x01000003	STATUS_HEAP_CORRUPTED	督機已損壞或已覆蓋保護頻寬 (在偵錯模式下)。客戶端代碼中的緩衝區溢位可能導致堆積損壞。
0x01000004	STATUS_HEAP_VRAM_LIB_MISSING	VRAM (視訊 RAM) 使用者或核心模式程式庫無法載入或遺失。檢查基礎平台是否支援 VRAM 分配。
0x01000005	STATUS_HEAP_VRAM_LIB_REOPEN	無法開啟 VRAM 程式庫。
0x01000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	無法載入 INIT 函數匯出。
0x01000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	無法載入 ALLOC 函數匯出。
0x01000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	無法載入 FREE 函數匯出。
0x01000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	無法載入 LOCK 函數匯出。
0x0100000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	無法載入 UNLOCK 函數匯出。

代碼	訊息	描述
0x0100000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	無法載入 UNINIT 函數匯出。
0x0100000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	無法載入 GETMAX 函數匯出。
0x0100000d	STATUS_HEAP_DIRECT_MEM_INIT	無法初始化在混合堆積中的主要堆積集區。
0x0100000e	STATUS_HEAP_VRAM_INIT_FAILED	VRAM 動態初始化失敗。
0x0100000f	STATUS_HEAP_LIBRARY_FREE_FAILED	無法取消分配和釋放 VRAM 程式庫。
0x01000010	STATUS_HEAP_VRAM_ALLOC_FAILED	VRAM 分配失敗。
0x01000011	STATUS_HEAP_VRAM_FREE_FAILED	VRAM 釋放失敗。
0x01000012	STATUS_HEAP_VRAM_MAP_FAILED	VRAM 對應失敗。
0x01000013	STATUS_HEAP_VRAM_UNMAP_FAILED	VRAM 取消對應失敗。
0x01000014	STATUS_HEAP_VRAM_UNINIT_FAILED	VRAM 取消初始化失敗。

## MKVGen 程式庫傳回的錯誤和狀態碼

下表包含MKVGen程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息	說明/建議的動作
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Frame 資料結構的無效成員。請確定持續時間、大小和影格資料都是有效的，且在中指定的限制範圍內 <a href="#">製作者 SDK 限制</a> 。
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	無效的影格時間戳記。計算的 PTS (簡報時間戳記) 和 DTS (解碼時間戳記) 大於或等於片段開始影格的時間戳記。這表示有潛在的媒體管道重新開機或編碼器穩定性問題。如需故障診斷資訊，請參閱 <a href="#">錯誤：「無法將影格提交到 Kinesis Video 用戶端」</a> 。
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	指定了無效的區段持續時間。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	無效的內容類型字串長度。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	嘗試編碼的數位太大，無法使用 EBML (可擴展二進位元語言) 格式表示。不會向軟體開發套件客戶端顯示該錯誤。
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	無效的轉碼器 ID 字串長度。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	無效的音軌名稱字串長度。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。

代碼	訊息	說明/建議的動作
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	無效的轉碼器私有資料長度。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	編解碼器私人資料 (CPD) 為 NULL，而持續專業發展大小則大於零。
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	時間碼比例值無效。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	影格時間碼大於最大值。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	已達最高影格時間碼。MKV 格式使用有符號 16 位元表示影格相對於叢集開頭的時間碼。如果無法表示影格時間碼，則會產生錯誤。該錯誤表示選擇的時間碼比例不正確叢集持續時間太長，因此，表示影格時間碼將導致有符號 16 位元空間溢出。



代碼	訊息	說明/建議的動作
0x3200000d	STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA	遇到無效的 Annex-B 啟動代碼。例如，指定了 Annex-B 調節標記，且代碼遇到包含超過三個零的無效開始序列。有效的 Annex-B 格式應具有「模擬防護」序列，以便對位元組串流中包含三個或更多零的序列進行逸出。如需詳細資訊，請參閱 MPEG 規格。如需 Android 上此錯誤的詳細資訊，請參閱 <a href="#">在 Android 上發生 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) 錯誤</a> 。
0x3200000e	STATUS_MKV_INVALID_AVCC_NALU_IN_FRAME_DATA	指定適應 AVCC 旗標時，無效的 AVCC NALU 封裝。確認位元組串流是否為有效的 AVCC 格式。如需詳細資訊，請參閱 MPEG 規格。
0x3200000f	STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED	指定了適應 AVCC 和 Annex-B 納盧斯。指定其中的一個 NAL，或者不指定任何 NAL。
0x32000010	STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD	在指定了調節 Annex-B 標記時，CPD 的 Annex-B 格式無效。驗證持續專業發展是否為有效的 Annex-B 格式。如果不是，則刪除 CPD Annex-B 適應標誌。

代碼	訊息	說明/建議的動作
0x32000011	STATUS_MKV_PTS_DTS _ARE_NOT_SAME	Kinesis Video Streams 會強制執行 PTS (簡報時間戳記) 和 DTS (解碼時間戳記) 對片段起始畫面相同。這些是區段開頭的關鍵影格。
0x32000012	STATUS_MKV_INVALID _H264_H265_CPD	無法解析 H264/H265 轉碼器私有資料。
0x32000013	STATUS_MKV_INVALID _H264_H265_SPS_WID TH	無法從轉碼器私有資料中取得寬度。
0x32000014	STATUS_MKV_INVALID _H264_H265_SPS_HEI GHT	無法從轉碼器私有資料中取得高度。
0x32000015	STATUS_MKV_INVALID _H264_H265_SPS_NALU	無效的 H264/H265 薩普納魯。
0x32000016	STATUS_MKV_INVALID _BIH_CPD	轉碼器私有資料中的點陣圖資訊標題格式無效。
0x32000017	STATUS_MKV_INVALID _HEVC_NALU_COUNT	無效的高效能視訊編碼 (HEVC) 網路抽象層單元 (NALU) 計數。
0x32000018	STATUS_MKV_INVALID _HEVC_FORMAT	無效的 HEVC 格式。
0x32000019	STATUS_MKV_HEVC_SP S_NALU_MISSING	序列參數集 (SPS) 中缺少 HEVC NALU。
0x3200001a	STATUS_MKV_INVALID _HEVC_SPS_NALU_SIZE	無效的 HEVC SPS NALU 大小。

代碼	訊息	說明/建議的動作
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	無效的 Chroma 格式 IDC。
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	無效的 HEVC 預留 SPS。
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	最小 AnnexBb 編解碼器私有 beta 值大小。對於 H264，此數值必須等於或大於 11。對於 H265，此數值必須等於或大於 15。
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Annex-B NALU 中缺少轉碼器私有資料。
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Annex-B NALU 中的轉碼器私有 beta 無效。
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	無效標籤名稱長度。有效值為大於零並小於 128。
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	無效標籤值長度。有效值大於零且小於 256。
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	無效的產生器狀態標籤。
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	無效的 AAC 轉碼器私有資料取樣頻率索引。
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	無效的 AAC 轉碼器私有資料管道組態。

代碼	訊息	說明/建議的動作
0x32000025	STATUS_MKV_INVALID_AAC_CPD	無效的 AAC 轉碼器私有資料。
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	找不到軌道資訊。
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	無效的區段 UUID。
0x32000028	STATUS_MKV_INVALID_TRACK_UID	無效的軌道 UID。
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	
0x3200002A	STATUS_MKV_INVALID_AMS_ACM_CPD	
0x3200002b	STATUS_MKV_MISSING_SPS_FROM_H264_CPD	
0x3200002C	STATUS_MKV_MISSING_PPS_FROM_H264_CPD	
0x3200002d	STATUS_MKV_INVALID_PARENT_TYPE	

### 追蹤程式庫傳回的錯誤和狀態碼

下表包含 Trace 程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息
0x10100001	STATUS_MIN_PROFILER_BUFFER

## Utils 程式庫傳回的錯誤和狀態碼

下表包含Utils程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息
0x40000001	STATUS_INVALID_BASE64_ENCODE
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED
0x40000011	STATUS_UTIL_MAX_TAG_COUNT

代碼	訊息
0x40000012	STATUS_UTIL_INVALID_TAG_VERSION
0x40000013	STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL
0x40000014	STATUS_UTIL_INVALID_TAG_NAME_LEN
0x40000015	STATUS_UTIL_INVALID_TAG_VALUE_LEN
0x4000002a	STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE
0x4000002b	STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED
0x4000002c	STATUS_THREADPOOL_MAX_COUNT
0x4000002d	STATUS_THREADPOOL_INTERNAL_ERROR
0x40100001	STATUS_HASH_KEY_NOT_PRESENT
0x40100002	STATUS_HASH_KEY_ALREADY_PRESENT
0x40100003	STATUS_HASH_ENTRY_ITERATION_ABORT
0x41000001	STATUS_BIT_READER_OUT_OF_RANGE
0x41000002	STATUS_BIT_READER_INVALID_SIZE
0x41100001	STATUS_TIMER_QUEUE_STOP_SCHEDULING
0x41100002	STATUS_INVALID_TIMER_COUNT_VALUE
0x41100003	STATUS_INVALID_TIMER_PERIOD_VALUE

代碼	訊息
0x41100004	STATUS_MAX_TIMER_COUNT_REACHED
0x41100005	STATUS_TIMER_QUEUE_SHUTDOWN
0x41200001	STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN
0x41200002	STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

## 視圖庫返回的錯誤和狀態碼

下表包含View程式庫中方法所傳回的錯誤和狀態資訊。

代碼	訊息	描述
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	指定的內容視圖項目數無效。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	指定的內容視圖持續時間無效。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	嘗試獲取的位置超過標題。
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	指定的索引無效。
0x30000005	STATUS_CONTENT_VIEW_INVALID_TIMESTAMP	具有無效的時間戳記，或者時間戳記發生重疊。影格解碼時間戳記應大於或等於上一個影格時間戳記，加上上一個

代碼	訊息	描述
		影格持續時間: `DTS(n) >= DTS(n-1) + Duration(n-1)`。此錯誤通常會指出「不穩定」的編碼器。編碼器會產生一組突發編碼影格，並且其時間戳記小於影格內的持續時間。或者，串流配置為使用軟體開發套件時間戳記，並且發送影格的速度比影格持續時間快。為了說明解決編碼器中的一些「抖動」問題，請在 <code>StreamInfo.StreamCaps</code> 架構中指定較小的影格持續時間。例如，如果串流為 25 FPS，則每個影格的持續時間為 40 毫秒。但是，為了處理編碼器「抖動」，我們建議您使用該幀持續時間的一半（20 毫秒）。某些串流要求更精確地控制錯誤檢測時間。
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	指定的內容視圖項目資料長度無效。

## 回 PutFrame 調返回的錯誤和狀態碼-C 生產者庫

下一節包含 C 生產者程式庫中 PutFrame 作業的回呼所傳回的錯誤和狀態資訊。

代碼	訊息	描述	建議的動作
0x15000001	STATUS_STOP_CALLBACK_CHAIN	回呼鏈已停止。	



代碼	訊息	描述	建議的動作
0x15000002	STATUS_MAXIMUM_CALLBACK_CHAIN	已達到回呼鏈上限。	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	無效的 PlatformCallbacks 架構版本。	指定正確的目前架構版本。
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	無效的 ProducerCallbacks 架構版本。	指定正確的目前架構版本。
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	無效的 StreamCallbacks 架構版本。	指定正確的目前架構版本。
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	無效的 AuthCallbacks 架構版本。	指定正確的目前架構版本。
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	無效的 ApiCallbacks 架構版本。	指定正確的目前架構版本。
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	無效的 AwsCredentials 架構版本。	指定正確的目前架構版本。
0x15000009	STATUS_MAXIMUM_REQUEST_HEADER_COUNT	已達到的請求標頭計數上限。	

代碼	訊息	描述	建議的動作
0x1500000a	STATUS_MAX_REQUEST_HEADER_NAME_LEN	已達到請求標頭名稱長度上限。	
0x1500000b	STATUS_MAX_REQUEST_HEADER_VALUE_LEN	已達到請求標頭值長度上限。	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	針對 API 呼叫傳回無效的 JSON。	
0x1500000d	STATUS_CURL_INIT_FAILED	Curl 初始化失敗。	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Curl lib 初始化失敗。	
0x1500000f	STATUS_INVALID_DESCRIPTOR_RETURN_JSON	的傳回 JSON 無效 DescribeStream。	
0x15000010	STATUS_HMAC_GENERATION_ERROR	HMAC 產生錯誤。	
0x15000011	STATUS_IOT_FAILED	IoT 授權失敗。	
0x15000012	STATUS_MAX_ROLE_ALIASES_LEN_EXCEEDED	已達到角色別名長度上限。	指定較短的別名長度。

代碼	訊息	描述	建議的動作
0x15000013	STATUS_MAXIMUM_USER_AGENT_NAME_POSTFIX_LENGTH_EXCEEDED	已達到代理程式名稱後置長度上限。	
0x15000014	STATUS_MAXIMUM_CUSTOM_USER_AGENT_LENGTH_EXCEEDED	已達到客戶使用者代理程式長度上限。	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	無效的使用者代理程式長度。	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	無效的端點快取期間。	指定少於 24 小時的快取期間。
0x15000017	STATUS_IOT_EXPIRATION_OCCURRED_IN_PAST	IoT 到期時間戳記發生在過去。	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	IoT 到期剖析失敗。	
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNCTION		

代碼	訊息	描述	建議的動作
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAXIMUM_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_CONTEXT_FAILED		
0x15000020	STATUS_INVALID_CERT_PATH		

代碼	訊息	描述	建議的動作
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEGINNING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEGINNING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RESET_FAILED		
0x16000001	STATUS_CURL_PERFORMANCE_FAILED		

代碼	訊息	描述	建議的動作
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH		
0x16000003	STATUS_IOT_NULL_AWS_CREDS		
0x16000004	STATUS_IOT_INVALID_URI_LEN		
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT		

## 網路抽象層 (NAL) 適應性旗標參考

本節包含有關 `StreamInfo.NalAdaptationFlags` 列舉的可用旗標的資訊。

應用程式中的[基礎串流](#)可以是 Annex-B 或 AVCC 格式：

- Annex-B 格式使用兩個位元組的零劃分 [NALU \(網路抽象層單位\)](#)，後面跟隨一或三個位元組的零，然後是數位 1 (稱為起始碼，例如 00000001)。
- AVCC 格式也對 NALU 進行了包裝，但每個 NALU 前都有一個表示 NALU 大小 (通常為 4 個位元組) 的值。

許多編碼器產生 Annex-B 位元串流格式。一些較高級別的位元串流處理器 (例如播放引擎或 [的 Media Source Extensions \(MSE\)](#) AWS Management Console 播放器) 則為其影格使用 AVCC 格式。

轉碼器私有資料 (CPD) 為 H.264 轉碼器的 SPS/PPS (序列參數集/圖片參數集)，也可以採用 Annex-B 或 AVCC 格式。但是，對 CPD 而言，格式與前面描述的有所不同。

這些旗標告知軟體開發套件將 NALU 與影格資料和 CPD 的 AVCC 或 Annex-B 調節，如下所示：

旗標	調節
NAL_ADAPTATION_FLAG_NONE	沒有適應。
NAL_ADAPTATION_ANNEXB_NALS	使阿訥克斯-B 納盧斯適應 AVCC 納盧斯。
NAL_ADAPTATION_AVCC_NALS	使 AVCC 納洛斯適應安納洛斯-B 納盧斯。
NAL_ADAPTATION_ANNEXB_CPD_NALS	將編解碼器私有數據調整為 AVCC 格式的 NALUS。
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	將 Annex-B NALUS 的編解碼器和框架私有數據調整為 AVCC 格式的 NALUS。

有關 NALU 類型的更多資訊，請參閱 RFC 3984 中的 [第 1.3 節：網路抽象層單位類型](#)。

## 製作者 SDK 結構

本節包含可用來提供資料給 Kinesis Video Streams 製作者物件之結構的相關資訊。

### 主題

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

### DeviceInfo/DefaultDeviceInfoProvider

DeviceInfo和DefaultDeviceInfoProvider物件控制 Kinesis Video Streams 製作者物件的行為。

### 成員欄位

- version — 整數值，用來確定結構的正確版本與目前版本的程式碼基底搭配使用。目前版本指定使用 DEVICE\_INFO\_CURRENT\_VERSION 巨集。
- name — 裝置的人類可讀名稱。

- 標記數/標籤 — 目前未使用。
- streamCount — 設備可以處理的最大流數量。這將為最初指向串流物件的指標預先分配儲存空間，但實際串流物件會隨後建立。預設值為 16 的串流，但您可於檔案中更改 DefaultDeviceInfoProvider.cpp 此編號。
- storageInfo : 描述主要儲存空間組態的物件。如需詳細資訊，請參閱[StorageInfo](#)。

## StorageInfo

指定 Kinesis Video Streams 的主要儲存裝置組態。

預設實施根據低區段快速堆積實施，此實作針對串流式處理進行最佳化。它使用 MEMALLOC 分配器，可以在指定平台上覆蓋。一些平台具有虛擬記憶體分配，沒有物理頁分配的支援。由於使用了記憶體，虛擬頁由實體頁提供支援。在儲存利用率偏低時，這會導致整體系統上的低記憶體壓力。

預設儲存空間大小根據以下公式計算。DefragmentationFactor 應設定為 1.2 (20%)。

```
Size = NumberOfStreams * AverageFrameSize * FramesPerSecond * BufferDurationInSeconds *  
DefragmentationFactor
```

在以下範例中，裝置具有音訊和視訊串流。音訊串流每秒採樣 512 次，平均樣本 100 位元組。視訊串流每秒 25 影格，平均樣本 10,000 位元組。每個串流有 3 分鐘的緩衝時間。

```
Size = (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) *  
1.2 = 65059200 = ~ 66MB.
```

如果裝置有更多可用記憶體，建議您為儲存裝置新增更多記憶體，以避免嚴重的碎片。

驗證儲存大小是否足以容納所有流的完整緩衝區，編碼複雜性高（當幀大小由於高運動而變大時）或頻寬低時。如果生產者達到內存壓力，它發出儲存溢出壓力回調（）StorageOverflowPressureFunc。但是，當內容存放區中沒有可用的記憶體時，它會捨棄正在推入 Kinesis Video Streams 的影格，並出現錯誤 (STATUS\_STORE\_OUT\_OF\_MEMORY = 0x5200002e)。如需詳細資訊，請參閱[用戶端程式庫傳回的錯誤和狀態碼](#)。這也可能發生，如果應用程式無法使用確認 (ACK) 或 ACK 持續延遲。在這種情況下，緩衝區填充到「緩衝區持續時間」容量之前，先前的幀開始丟棄。

## 成員欄位

- version — 整數值，用來確定結構的正確版本與目前版本的程式碼基底搭配使用。



- `storageType` — 一種 `DEVICE_STORAGE_TYPE` 枚舉，用於指定存儲的基礎支持和實現。目前僅支援的值為 `DEVICE_STORAGE_TYPE_IN_MEM`。未來的實作中將支援 `DEVICE_STORAGE_TYPE_HYBRID_FILE`，表示儲存空間退回到檔案支援的內容儲存空間。
- `storageSize` — 要預先配置的儲存區大小 (以位元組為單位)。最小配置為 10 MB，最大配置為 10 GB。(這會變更檔案支援內容的存放區的未來實作。)
- `spillRatio` — 整數值，代表要從直接記憶體儲存類型 (RAM) 配置的儲存體百分比，而不是次要溢位儲存體 (檔案儲存)。目前未使用。
- `rootDirectory`：檔案支援的內容儲存空間所在目錄的路徑位置。目前未使用。

## Kinesis 視頻流結構

您可以使用下列結構將資料提供給 Kinesis 視訊串流的執行個體。

### 主題

- [StreamDefinition/StreamInfo](#)
- [ClientMetrics](#)
- [StreamMetrics](#)

## StreamDefinition/StreamInfo

`StreamDefinition` 物件在 C++ 層中包裝 `StreamInfo` 物件於獨立於平台的代碼中，並提供建構函數中的某些預設值。

### 成員欄位

欄位	資料類型	描述	預設值
<code>stream_name</code>	<code>string</code>	選用的串流名稱。如需更多關於串流名稱長度的資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。每個串流皆應有唯一的名稱。	如果未指定名稱，則會產生隨機的名稱。

欄位	資料類型	描述	預設值
retention_period	duration<uint64_t, ratio<3600>>	串流的保留期間以秒為單位。指定 0 代表沒有保留。	3600 (一小時)
tags	const map<string, string>*	包含使用者資訊之金鑰值對的對應。如果串流已有一組標籤，則新標籤會附加到現有的一組標籤之後。	無標籤
kms_key_id	string	使用於加密串流的 AWS KMS 金鑰 ID。如需詳細資訊，請參閱 <a href="#">Kinesis Video Streams 中的資料保護</a> 。	預設 KMS 金鑰 (aws/kinesis-video )。
streaming_type	STREAMING_TYPE 列舉	唯一支援的值為 STREAMING_TYPE_REALTIME 。	
content_type	string	串流的內容格式。Kinesis Video Streams 主控台可以播放 video/h264 格式的內容。	video/h264
max_latency	duration<uint64_t, milli>	串流的最大延遲，以毫秒為單位。當緩衝持續時間超過此時間量時，會呼叫串流延遲壓力回呼 (如果指定)。指定 0 表示將不呼叫串流延遲壓力回呼。	milliseconds::zero()

欄位	資料類型	描述	預設值
fragment_duration	duration<uint64_t>	您要的片段持續時間，以秒為單位。此值與 key_frame_fragmentation 值結合使用。如果此值為 false，則 Kinesis Video Streams 會在此持續時間過後在關鍵影格上產生片段。例如，進階音訊編碼 (AAC) 音訊串流將每個影格做為關鍵影格。指定 key_frame_fragmentation = false 導致在經過此持續時間之後在關鍵影格上片段化，產生 2 秒的片段。	2

欄位	資料類型	描述	預設值
timecode_scale	duration< uint64_t, milli>	MKV 時間碼比例以毫秒為單位，這用於指定 MKV 叢集中影格的時間碼精細度。MKV 影格時間碼一律相對於叢集的開始。MKV 使用簽章的 16 位元值 (0-32767) 來代表時間碼在叢集 (片段) 中。確認影格時間碼可以使用指定的時間碼刻度來表示。時間碼規模的預設值 1 毫秒可確保最大的影格，則可以代表 32767 毫秒 ≈ 32 秒。這是超過最大持續時間指定片段 <a href="#">Kinesis Video Streams 服務配額</a> 為 10 秒。	1
key_frame_fragmentation	bool	是否在關鍵影格上產生片段。如果為 true，則開發套件在每次關鍵影格出現時會產生片段的開始。如果 false，Kinesis Video Streams 至少會等待，fragment_duration 並在其後面的關鍵影格上產生一個新片段。	true

欄位	資料類型	描述	預設值
frame_timecodes	bool	是否使用影格時間碼或者使用目前時間回呼產生時間戳記。許多編碼器不會產生時間戳記的影格。因此，false為此參數指定可確保在將影格放入 Kinesis Video Streams 時的時間戳記。	true
absolute_fragment_times	bool	Kinesis Video Streams 使用 MKV 作為其基礎封裝機制。有關影格的時間碼 MKV 規格嚴格的相對於叢集 (片段) 的開始。不過，在叢集的時間碼可以絕對或相對於串流的開始時間。如果時間戳記是相對的，PutMedia 服務 API 呼叫會使用可選的串流開始時間戳記並調節叢集時間戳記。該服務隨時儲存區段及其絕對時間戳記。	true
fragment_acks	bool	是否接收應用程序級片段 ACK ( 確認 ) 。	true，表示軟體開發套件接收 ACK 並依此操作。

欄位	資料類型	描述	預設值
restart_on_error	bool	是否在出現特定錯誤時重新啟動。	true，表示在出現任何錯誤時開發套件將嘗試重新啟動串流。
recalculate_metrics	bool	是否重新計算指標。每個擷取指標的呼叫可以重新計算這些值，以獲取最新的「執行中」值，這可能會產生較小的 CPU 影響。您可能需要設定為 false 在極低功耗/容量的裝置上以避免佔用 CPU 週期。否則，我們不建議使 false 用此值。	true

欄位	資料類型	描述	預設值
nal_adaptation_flags	uint32_t	<p>指定網路抽象層單位 (NALU) 調節旗標。如果位元串流是 H.264 編碼，則在 NALU 中可以處理做為原始資料或封裝。這些可以為 Annex-B 或 AVCC 格式。大多數基本流生產者和消費者 (讀取編碼器和解碼器) 使用 Annex-B 格式，因為它具有諸如錯誤恢復等優點。較高層級的系統使用 AVCC 格式的預設格式，適用於 MPEG、HLS、DASH 等。主控台播放使用瀏覽器的 MSE (Media Source Extensions) 來解碼和播放使用 AVCC 格式的串流。對於 H264 (以及對於 M-JPEG 和 H265)，開發套件提供自我調節功能。</p> <p>許多基本串流採用以下格式。在此範例中，Ab 是 Annex-B 開始代碼 (001 或 0001)。</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B- frame) Ab(P/B-fr ame)... Ab(Sps)Ab</pre> </div>	預設情況下，影格資料和編解碼器私有資料均為從 Annex-B 格式調節為 AVCC 格式。

欄位	資料類型	描述	預設值
		<p>(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)</p> <p>在 H.264 的情況下，編解碼器私有數據 (CPD) 位於 SPS (序列參數集) 和 PPS (圖片參數集) 參數中，並且可以適應 AVCC 格式。除非媒體管道讓 CPD 分開，應用程式可以從影格擷取 CPD。它可以通過查找第一個 IDR 框架 (應包含 SPS 和 PPS) 來實現此目的，提取兩個 NALU (即 Ab(Sps)Ab(Pps))，並將其設置在 CPD 中。StreamDefinition</p> <p>如需詳細資訊，請參閱 <a href="#">NAL 適應性旗標</a>。</p>	
frame_rate	uint32_t	預期的影格率。此值用於更完善計算緩衝需求。	25
avg_bandwidth_bps	uint32_t	串流的預期平均頻寬。此值用於更完善計算緩衝需求。	4 * 1024 * 1024



欄位	資料類型	描述	預設值
buffer_duration	duration<uint64_t>	串流緩衝持續時間，以秒為單位。SDK 會將內容存放區中的框架保留為最多buffer_duration，之後會在視窗向前移動時捨棄先前的框架。如果正在刪除的幀尚未發送到後端，則會調用丟棄的幀回調。如果目前緩衝區持續時間大於 max_latency，則將呼叫延遲壓力回呼。收到片段持續 ACK 時，緩衝區會裁剪到下一個片段啟動。這表示其內容的已長期存放在雲端，所以不再需要在執行個體的本機裝置的內容儲存。	120

欄位	資料類型	描述	預設值
replay_duration	duration<uint64_t>	如果啟用重新啟動，則在發生錯誤期間向後捲動目前讀取器以重播的持續時間 (以秒為單位)。轉返操作將在緩衝開始時停止 (在剛啟動串流時或已出現持續的 ACK 時)。轉返嘗試停留在顯示片段開始的關鍵影格上。如果 '導致重新啟動的錯誤不是指示無效的主機 (主機仍然存在，並且在其內部緩衝區中包含框架資料)，則回滾會停止在上次收到的 ACK 框架。它會將轉發到下一個金鑰影格，因為整個片段已存放在主機記憶體。	40
connection_staleness	duration<uint64_t>	如果 SDK 未收到緩衝 ACK，則會呼叫串流失效回呼的時間 (以秒為單位)。它表示幀正在從設備發送，但後端未承認它們。此條件表示在中間躍點或負載平衡器的連線已中斷。	30
codec_id	string	MKV 音軌的轉碼器 ID。	V_MPEG4/ISO/AVC

欄位	資料類型	描述	預設值
track_name	string	MKV 音軌名稱。	kinesis_video
codecPrivateData	unsigned char*	編解碼器私有資料 (CPD) 緩衝區。如果媒體管道在串流啟動之前具有 CPD 的相關資訊，則可以在 StreamDefinition.codecPrivateData 中發送。此時將複製位元，在建立串流的呼叫之後可以重新使用緩衝區或將其釋放。但是，如果資料在建立串流時無法使用，則可以在KinesisVideoStream.start(cpd) 函數的其中一個重載中設定資料。	null
codecPrivateData大小	uint32_t	編解碼器私有資料緩衝區大小。	0

## ClientMetrics

該ClientMetrics對象通過調用填充getKinesisVideoMetrics。

## 成員欄位

欄位	資料類型	描述
version	UINT32	結構的版本，在 CLIENT_METRICS_CURRENT_VERSION 巨集中定義。
contentStoreSize	UINT64	整體內容存放區大小，以位元組為單位。這是 DeviceInfo.StorageInfo.storageSize 中指定的值。
contentStoreAvailable大小	UINT64	目前可用的儲存大小 (位元組)。
contentStoreAllocated大小	UINT64	目前配置的大小。由於內部記帳和內容存放區的實施，配置加上可用的大小應略小於總儲存空間的大小。
totalContentViews大小	UINT64	所有串流的所有內容視圖的已分配記憶體大小。這不會計入儲存空間大小。此記憶體使用 MEMALLOC 巨集分配，可以覆蓋該值以提供自訂分配器。
totalFrameRate	UINT64	在所有串流上觀察到的總影格率。
totalTransferRate	UINT64	在所有串流上觀察到的總串流速率，以每秒位元組數為單位。

## StreamMetrics

該StreamMetrics對象通過調用填充getKinesisVideoMetrics。

## 成員欄位

欄位	資料類型	描述
version	UINT32	結構的版本，在 STREAM_METRICS_CURRENT_VERSION 巨集中定義。
currentViewDuration	UINT64	累積影格的時間長度。在快速聯網的情況下，此持續時間可能是零或幀持續時間（因為幀正在傳輸）。如果持續時間長於中max_latency 指定的StreamDefinition，則會呼叫串流延遲回呼（如果已指定）。以 100 奈秒 (ns) 為單位指定持續時間，這是 PIC 層的預設時間單位。
overallViewDuration	UINT64	整體檢視持續時間。如果串流設定沒有 ACK 或持續性，則此值會隨著影格放入 Kinesis 視訊串流中而變得等於buffer_duration 中的StreamDefinition 而增加。啟用 ACK 並收到持續的 ACK 時，緩衝區會裁剪到下一個關鍵影格。這是因為 ACK 時間戳表示整個片段的開始。以 100 奈秒 (ns) 為單位指定持續時間，這是 PIC 層的預設時間單位。
currentViewSize	UINT64	目前緩衝區的大小，以位元組為單位。

欄位	資料類型	描述
overallViewSize	UINT64	整體檢視大小，以位元組為單位。
currentFrameRate	UINT64	目前串流觀察到的影格率。
currentTransferRate	UINT64	目前串流觀察到的傳輸速率，以每秒位元組數為單位。

## 生產者 SDK 回呼

Amazon Kinesis Video Streams 產生器開發套件中的類別和方法不會維護自己的程序。反之，他們會使用傳入的功能呼叫和事件來排程回呼與應用程式通訊。

有兩種呼模式應用程式可以使用與開發套件互動：

- [CallbackProvider](#)— 此物件會將平台無關程式碼 (PIC) 元件的每個回呼公開至應用程式。此模式允許完整的功能，但也表示實作必須處理所有公有 API 方法和在 C++ 層的簽章。
- [StreamCallbackProvider](#)和 [ClientCallbackProvider](#)-這些對象公開了特定於流和客戶端特定的回調，並且 SDK 的 C++ 層公開了其餘的回調。這是與製作者開發套件互動偏好的回呼模式。

下圖說明回呼物件的物件模型：

在先前的圖表，DefaultCallbackProvider 從衍生的 CallbackProvider (公開在 PIC 的所有回呼) 和包含 StreamCallbackProvider 和 ClientCallbackProvider。

本主題包含下列章節：

- [ClientCallbackProvider](#)
- [StreamCallbackProvider](#)
- [ClientCallbacks 結構](#)
- [重試流式傳輸的回調實現](#)

## ClientCallbackProvider

ClientCallbackProvider 用戶端層級回呼函數公開的物件。函數的詳細資訊說明在 [ClientCallbacks](#) 節。

回呼方法：

- getClientReadyCallback— 報告用戶端的就緒狀態。
- getStorageOverflowPressureCallback— 報告儲存空間溢位或壓力。此回呼稱為當儲存使用率低於 STORAGE\_PRESSURE\_NOTIFICATION\_THRESHOLD 值，這是 5% 的整體儲存大小。如需詳細資訊，請參閱 [StorageInfo](#)。

## StreamCallbackProvider

StreamCallbackProvider 串流層級回呼函數公開的物件。

回呼方法：

- getDroppedFragmentReportCallback：報告捨棄的片段。
- getDroppedFrameReportCallback— 報告丟棄的框架。
- getFragmentAckReceivedCallback— 報告為流收到片段 ACK。
- getStreamClosedCallback— 報告流關閉條件。
- getStreamConnectionStaleCallback— 報告過時的連線狀況。在這種情況下，生產者正在向服務發送數據，但沒有收到確認。
- getStreamDataAvailableCallback— 報告資料流中可用的資料。
- getStreamErrorReportCallback— 報告串流錯誤情況。
- getStreamLatencyPressureCallback— 報告串流延遲情況，也就是累積緩衝區大小大於 max\_latency 值時。如需詳細資訊，請參閱 [StreamDefinition/StreamInfo](#)。
- getStreamReadyCallback: — 報告資料流就緒條件。
- getStreamUnderflowReportCallback— 報告資料流下溢狀況。此功能目前未使用，並保留供 future 使用。

如需的原始程式碼 StreamCallbackProvider，請參閱 [StreamCallbackProvider.h](#)。

## ClientCallbacks 結構

此 ClientCallbacks 架構包含回呼函數進入點，當特定事件發生時的 PIC 呼叫。結構也包含版本資訊 CALLBACKS\_CURRENT\_VERSION 的欄位，以及 customData 欄位使用者定義資料回傳的個別回呼功能。

用戶端應用程式可以使用 this 指標 custom\_data 欄位到對應成員函數的靜態 ClientCallback 函數在執行時間，如下列程式碼範例所示：

```
STATUS TestStreamCallbackProvider::streamClosedHandler(UINT64 custom_data,
    STREAM_HANDLE stream_handle, UINT64 stream_upload_handle) {
    LOG_INFO("Reporting stream stopped.");

    TestStreamCallbackProvider* streamCallbackProvider =
        reinterpret_cast<TestStreamCallbackProvider*> (custom_data);
    streamCallbackProvider->streamClosedHandler(...);
```

### 事件

函數	Description (描述)	Type
CreateDeviceFunc	目前並未在後端實作。從 Java 或 C++ 呼叫時發生故障。其他用戶端執行平台特定的初始化。	後端 API
CreateStreamFunc	建立串流時呼叫。	後端 API
DescribeStreamFunc	呼叫 DescribeStream 時已呼叫。	後端 API
GetStreamingEndpointFunc	呼叫 GetStreamingEndpoint 時已呼叫。	後端 API
GetStreamingTokenFunc	呼叫 GetStreamingToken 時已呼叫。	後端 API
PutStreamFunc	呼叫 PutStream 時已呼叫。	後端 API
TagResourceFunc	呼叫 TagResource 時已呼叫。	後端 API



函數	Description (描述)	Type
CreateMutexFunc	建立同步互斥。	同步
FreeMutexFunc	釋放互斥。	同步
LockMutexFunc	鎖定同步互斥。	同步
TryLockMutexFunc	嘗試鎖定互斥。目前並未實作。	同步
UnlockMutexFunc	解除鎖定互斥。	同步
ClientReadyFunc	當用戶端進入就緒狀態時呼叫。	通知
DroppedFrameReportFunc	當影格已刪除時報告。	通知
DroppedFragmentReportFunc	當片段已刪除時報告。此功能目前未使用，並保留供 future 使用。	通知
FragmentAckReceivedFunc	當片段 ACK (緩衝、收到、保留和錯誤) 收到呼叫。	通知
StorageOverflowPressureFunc	當儲存使用率低於 STORAGE_PRESSURE_NOTIFICATION_THRESHOLD 值時呼叫，這是定義為 5% 的整體儲存大小。	通知
StreamClosedFunc	當呼叫最後位元串流的剩餘影格。	通知

函數	Description (描述)	Type
StreamConnectionStaleFunc	當串流輸入過時連線狀態時呼叫。在這個條件，是製作者傳送資料到服務但未收到確認。	通知
StreamDataAvailableFunc	串流資料可用時呼叫。	通知
StreamErrorReportFunc	出現串流錯誤時呼叫。在這個條件下的 PIC 自動關閉串流。	通知
StreamLatencyPressureFunc	串流進入延遲條件時呼叫，這是當累計緩衝大小大於 max_latency 值。如需詳細資訊，請參閱 <a href="#">StreamDefinition/StreamInfo</a> 。	通知
StreamReadyFunc	當串流進入就緒狀態時呼叫。	通知
StreamUnderflowReportFunc	此功能目前未使用，並保留供 future 使用。	通知
DeviceCertToTokenFunc	傳回連線憑證為字符。	平台整合
GetCurrentTimeFunc	傳回目前的時間。	平台整合
GetDeviceCertificateFunc	傳回裝置憑證。此功能目前未使用，並保留供 future 使用。	平台整合
GetDeviceFingerprintFunc	傳回裝置指紋。此功能目前未使用，並保留供 future 使用。	平台整合
GetRandomNumberFunc	傳回一隨機編號介於 0 和 RAND_MAX 之間。	平台整合

函數	Description (描述)	Type
GetSecurityTokenFunc	傳回傳遞給與後端 API 通訊之函數的安全性權杖。此實作可指定序列化 AccessKey Id , SecretKeyId 以及工作階段字符。	平台整合
LogPrintFunc	文字行日誌與標籤和日誌層級。如需詳細資訊，請參閱PlatformUtils.h 。	平台整合

對於先前表格的平台整合功能，最後參數是一種 ServiceCallContext 架構，其中包含下列欄位：

- version：此結構的版本。
- callAfter：絕對時間之後的呼叫函數。
- timeout：逾時在 100 奈米秒單位的操作。
- customData：使用者定義的值傳回用戶端。
- pAuthInfo：呼叫的登入資料。如需詳細資訊，請參閱下列 (\_\_AuthInfo) 結構。

授權資訊提供使用的 \_\_AuthInfo 結構，這可以是序列化登入資料或提供者特定身分驗證字符。此架構包含下列欄位：

- version：此版本的 \_\_AuthInfo 結構。
- type：AUTH\_INFO\_TYPE 數值定義登入資料類型 (憑證或安全字符)。
- data：位元組陣列包含身分驗證資訊。
- size：參數 data 的大小。
- expiration：登入資料在 100 奈米秒單位過期。

## 重試流式傳輸的回調實現

Kinesis Video Producer SDK 提供透過回呼函數串流的狀態。建議您實作下列回呼機制，以便從串流期間遇到的任何暫時網路問題中復原。

- 串流延遲壓力回呼-當 SDK 遇到串流延遲條件時，就會啟動此回呼機制。累積的緩衝大小大於 MAX\_LATENCY 值時，就會發生這種情況。建立串流時，串流應用程式會將 MAX\_LATENCY 設為 60 秒的預設值。此回呼的典型實作是重設連線。您可以根據需要使用示例實現 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/src/source/kinesis-video-c-producer.c>。StreamLatencyStateMachine 請注意，沒有選項可以將由於網絡中斷而未傳遞的幀存儲到輔助存儲中以進行回填。
- 串流失效回呼-當生產者可以將資料傳送至 Amazon Kinesis Data Streams 服務 (上行) 但無法及時取得確認 (緩衝 ACK) 時，便會啟動此回呼 (預設值為 60 秒)。視網路設定而定，串流延遲壓力回呼或串流失效回呼，或兩者皆可啟動。與串流延遲壓力回呼重試實作類似，典型實作是重設連線並開始新的串流連線。您可以根據需要在 [https://github.com/aws-labs/amazon-kinesis-video-streams-生產者-C/B\\_ConnectionStateStateMachine](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/src/source/C/B_ConnectionStateStateMachine.c) blob/master/src/source/.c 中使用示例實現。
- 串流錯誤回呼-當 SDK 在呼叫 KVS API 服務呼叫期間遇到網路連線逾時或其他錯誤時，就會啟動此回呼。
- 丟棄幀回調-當存儲大小已滿無論是因為網路速度慢或流錯誤，這個回調被啟動。如果網路速度導致影格下降，您可以增加儲存大小、縮小視訊影格大小或影格速率以符合網路速度。

# Kinesis 視頻流解析器庫

Kinesis 視訊串流剖析器程式庫是一組工具，您可以在 Java 應用程式中使用 Kinesis 視訊串流中使用 MKV 資料。

程式庫包含下列工具：

- [StreamingMkvReader](#)：此類別可由影片串流讀取指定的 MKV 元素。
- [FragmentMetadataVisitor](#)：此類別可擷取片段 (媒體元素) 及軌道 (包含媒體資訊的個別資料流，例如音訊或字幕) 的中繼資料。
- [OutputSegmentMerger](#)：此類別可合併影片串流之中的連續片段或區塊。
- [KinesisVideoExample](#)：這是示範如何使用 Kinesis 視訊串流剖析器程式庫的範例應用程式。

程式庫也包括測試，顯示工具的使用方式。

## 程序：使用 Kinesis 視訊串流剖析程式庫

本程序包含下列步驟：

- [the section called “步驟 1：下載並組態程式碼”](#)。
- [the section called “步驟 2：撰寫並檢查程式碼”](#)。
- [the section called “步驟 3：執行並驗證程式碼”](#)。

## 先決條件

您必須具備下列項目，才能檢查並使用 Kinesis 視訊串流剖析器程式庫：

- 一個 Amazon Web Services (AWS) 帳戶。如果您還沒有 AWS 帳戶，請參閱[the section called “註冊一個 AWS 帳戶”](#)。
- 一個 Java 的集成開發環境 (IDE)，如[日食 Java 霓虹燈](#)或[JetBrains IntelliJ 理念](#)。

## 步驟 1：下載並組態程式碼

在本節中，您將下載 Java 程式庫及測試程式碼，並將專案匯入至 Java IDE。

如需先決條件及此程序的其他詳細資料，請參閱 [串流剖析器程式庫](#)。

1. 創建一個目錄並從存儲庫克隆GitHub庫源代碼 ( [https://github.com/aws/amazon-kinesis-video-streams-解析器庫](https://github.com/aws/amazon-kinesis-video-streams-parser-library) )。

```
$ git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. 打開您正在使用的 Java IDE ( 例如 , [日食](#) 或 [IntelliJ 理想](#) ) , 然後導入您下載的阿帕奇 Maven 項目 :
  - 在 Eclipse 中 : 請選擇 File (檔案)、Import (匯入)、Maven、Existing Maven Projects (現有 Maven 專案) , 並巡覽至 kinesis-video-streams-parser-lib 資料夾。
  - 在 IntelliJ Idea : 選擇 Import (匯入)。巡覽至已下載套件根目錄的 pom.xml 檔案。

如需更多詳細資訊 , 請參閱 IDE 文件。

## 下一步驟

[the section called “步驟 2 : 撰寫並檢查程式碼”](#).

## 步驟 2 : 撰寫並檢查程式碼

本節您將檢驗 Java 程式庫及測試程式碼 , 並了解如何在自己的程式碼之中 , 使用程式庫的各項工具。

Kinesis 視訊串流剖析器程式庫包含下列工具 :

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

## StreamingMkvReader

此類別能以不封鎖的方式 , 由串流讀取指定的 MKV 元素。

下列程式碼範例 (來自 `FragmentMetadataVisitorTest`) 顯示如何建立及使用 `StreamingMkvReader` , 由名為 `inputStream` 的輸入串流擷取 `MkvElement` 物件。

```
StreamingMkvReader mkvStreamReader =
```

```

        StreamingMkvReader.createDefault(new
InputStreamParserByteSource(inputStream));
    while (mkvStreamReader.mightHaveNext()) {
        Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
        if (mkvElement.isPresent()) {
            mkvElement.get().accept(fragmentVisitor);
            ...
        }
    }
}

```

## FragmentMetadataVisitor

此類別會擷取片段 (媒體元素) 的中繼資料，並追蹤包含媒體資訊的個別資料串流，例如編解碼器私人資料、像素寬度或像素高度。

下列程式碼範例 (來自 `FragmentMetadataVisitorTest` 檔案) 顯示如何使用 `FragmentMetadataVisitor` 由 `MkvElement` 物件擷取資料：

```

FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));
int segmentCount = 0;
while(mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaInfo().getTypeInfo()))
{
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame =
((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
            MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
            assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
        }
        if
(MkvTypeInfoos.SEGMENT.equals(mkvElement.get().getElementMetaInfo().getTypeInfo())) {
            if (mkvElement.get() instanceof MkvEndMasterElement) {
                if (segmentCount < continuationTokens.size()) {
                    Optional<String> continuationToken =
fragmentVisitor.getContinuationToken();

```





```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

OutputSegmentMerger outputSegmentMerger =
    OutputSegmentMerger.createDefault(outputStream);

CompositeMkvElementVisitor compositeVisitor =
    new TestCompositeVisitor(fragmentVisitor, outputSegmentMerger);

final InputStream in = TestResourceUtil.getTestInputStream("output_get_media.mkv");

StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));

while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(compositeVisitor);
        if
(MkvTypeInfoos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
        {
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
            Assert.assertTrue(frame.getFrameData().limit() > 0);
            MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
            assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
        }
    }
}
```

前述範例顯示下列編碼模式：

- 建立 [FragmentMetadataVisitor](#) 由串流擷取中繼資料。
- 建立輸出串流以接收合併的中繼資料。
- 建立 OutputSegmentMerger，在 ByteArrayOutputStream 之中傳送。
- 建立包含兩個訪客的 CompositeMkvElementVisitor。
- 建立 InputStream 指向指定檔案。
- 將輸入資料的各個元素合併為輸出串流。

## KinesisVideoExample

這是示範如何使用 Kinesis 視訊串流剖析器程式庫的範例應用程式。

此類別執行下列操作：

- 建立 Kinesis 視訊串流。如果特定名稱串流已經存在，串流將遭到刪除並重新建立。
- 呼叫 [PutMedia](#) 將視訊片段串流至 Kinesis 視訊串流。
- 呼叫 [GetMedia](#) 串流視訊片段，從 Kinesis 視訊串流。
- 使用 [StreamingMkvReader](#) 剖析在串流傳回的片段，並使用 [FragmentMetadataVisitor](#) 記錄片段。

### 刪除及重新建立串流

下列程式碼範例 (從 `StreamOps.java` 檔案) 刪除指定的 Kinesis 視訊串流：

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

下列程式碼範例 (從 `StreamOps.java` 檔案) 建立具有指定名稱的 Kinesis 視訊串流：

```
amazonKinesisVideo.createStream(new CreateStreamRequest().withStreamName(streamName)
    .withDataRetentionInHours(DATA_RETENTION_IN_HOURS)
    .withMediaType("video/h264"));
```

### 呼叫 PutMedia

下列程式碼範例 (來自 `PutMediaWorker.java` 檔案) [PutMedia](#) 會呼叫串流：

```
putMedia.putMedia(new PutMediaRequest().withStreamName(streamName)
    .withFragmentTimecodeType(FragmentTimecodeType.RELATIVE)
    .withProducerStartTimestamp(new Date())
    .withPayload(inputStream), new PutMediaAckResponseHandler() {
    ...
});
```

### 呼叫 GetMedia

下列程式碼範例 (來自 `GetMediaWorker.java` 檔案) [GetMedia](#) 會呼叫串流：

```
GetMediaResult result = videoMedia.getMedia(new  
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

## 解析結果GetMedia果

本節說明如何使用 [StreamingMkvReader](#)、[FragmentMetadataVisitor](#) 及 [CompositeMkvElementVisitor](#)，以剖析、儲存至檔案，以及記錄 GetMedia 傳回的資料。

### 讀取GetMedia與的輸出 StreamingMkvReader

下面的代碼示例（從 `GetMediaWorker.java` 文件）創建一個 [StreamingMkvReader](#) 並使用它來解析 [GetMedia](#) 操作的結果：

```
StreamingMkvReader mkvStreamReader = StreamingMkvReader.createDefault(new  
    InputStreamParserByteSource(result.getPayload()));  
log.info("StreamingMkvReader created for stream {}", streamName);  
try {  
    mkvStreamReader.apply(this.elementVisitor);  
} catch (MkvElementVisitException e) {  
    log.error("Exception while accepting visitor {}", e);  
}
```

在前述程式碼範例中，[StreamingMkvReader](#) 由 GetMedia 結果的承載擷取 MKVElement 物件。在下列節之中，元素將傳送至 [FragmentMetadataVisitor](#)。

### 使用檢索片段 FragmentMetadataVisitor

下列程式碼範例（來自 `KinesisVideoExample.java` 及 `StreamingMkvReader.java` 檔案）建立 [FragmentMetadataVisitor](#)。MkvElement 物件（由 [StreamingMkvReader](#) 重複執行）將以 `accept` 方法傳送至訪客。

來自 `KinesisVideoExample.java`：

```
FragmentMetadataVisitor fragmentMetadataVisitor = FragmentMetadataVisitor.create();
```

來自 `StreamingMkvReader.java`：

```
if (mkvElementOptional.isPresent()) {  
    //Apply the MkvElement to the visitor
```

```
mkvElementOptional.get().accept(elementVisitor);
}
```

## 記錄元素並將其寫入檔案

下列程式碼範例 (來自 `KinesisVideoExample.java` 檔案) 建立下列物件，並將其以 `GetMediaProcessingArguments` 函數傳回值的一部分傳回：

- `LogVisitor` (`MkvElementVisitor` 延伸) 寫入系統記錄。
- `OutputStream` 將傳入資料寫入 MKV 檔案。
- `BufferedOutputStream` 針對 `OutputStream` 緩衝資料邊界。
- [the section called “OutputSegmentMerger”](#) 以相同軌道及 EBML 資料合併 `GetMedia` 結果之中的連續元素。
- 組 `CompositeMkvElementVisitor` 成 [FragmentMetadataVisitor](#)、[the section called “OutputSegmentMerger”](#) 和 `LogVisitor` 到單一元素訪客的 A。

```
//A visitor used to log as the GetMedia stream is processed.
LogVisitor logVisitor = new LogVisitor(fragmentMetadataVisitor);

//An OutputSegmentMerger to combine multiple segments that share track and ebml
metadata into one
//mkv segment.
OutputStream fileOutputStream =
Files.newOutputStream(Paths.get("kinesis_video_example_merged_output2.mkv"),
StandardOpenOption.WRITE, StandardOpenOption.CREATE);
BufferedOutputStream outputStream = new BufferedOutputStream(fileOutputStream);
OutputSegmentMerger outputSegmentMerger =
OutputSegmentMerger.createDefault(outputStream);

//A composite visitor to encapsulate the three visitors.
CompositeMkvElementVisitor mkvElementVisitor =
new CompositeMkvElementVisitor(fragmentMetadataVisitor,
outputSegmentMerger, logVisitor);

return new GetMediaProcessingArguments(outputStream, logVisitor,
mkvElementVisitor);
```

然後將媒體處理引數傳遞到 `GetMediaWorker`，該引數又傳遞給 `ExecutorService`，該引數會在單獨的執行緒上執行 `Worker`：

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getMkvElementVisitor());
executorService.submit(getMediaWorker);
```

## 下一步驟

[the section called “步驟 3：執行並驗證程式碼”](#)

## 步驟 3：執行並驗證程式碼

Kinesis 視訊串流剖析器程式庫包含的工具可供您在自己的專案中使用。專案包含工具的單元測試，可執行用於驗證安裝。

程式庫包含下列單元測試：

- mkv
  - ElementSizeAndOffsetVisitorTest
  - MkvValueTest
  - StreamingMkvReaderTest
- 公用程式
  - FragmentMetadataVisitorTest
  - OutputSegmentMergerTest

# Amazon Kinesis Video Streams 範例

下列程式碼範例示範如何使用 Kinesis Video Streams API：

## 範例：將資料傳送至 Kinesis Video Streams

- [範例：Kinesis Video Streams 產生器 SDK 外掛程式](#)：示範如何建置 Kinesis Video Streams 製作工具開發套件，以用作 GStreamer 目的地。
- [在碼頭容器中運行 GStreamer 元素](#)：示範如何使用預先建立的 Docker 映像檔，將 RTSP 視訊從 IP 攝影機傳送至 Kinesis Video Streams。
- [範例：從 RTSP 來源進行串流](#)：示範如何建立您自己的 Docker 映像檔，並將 RTSP 視訊從 IP 攝影機傳送至 Kinesis Video Streams。
- [範例：使用 API 將資料傳送至 Kinesis Video Streams PutMedia](#)：示範如何使用 [PutMedia API](#) 將資料傳送使用 [Java Producer Library](#) 至 Kinesis Video Streams 已經採用容器格式 (MKV) 的 Kinesis 影片串流。

## 範例：從 Kinesis Video Streams 擷取資料

- [KinesisVideoExample](#)：示範如何使用 Kinesis 視訊串流剖析器程式庫剖析和記錄視訊片段。
- [範例：剖析和呈現 Kinesis Video Streams 片段](#)：示範如何使用 [JCodec](#) 和 [J Frame](#) 來剖析和呈現 Kinesis 視訊串流片段。
- [the section called “SageMaker”](#)：顯示使用 Amazon SageMaker 判斷某些物件何時出現在視訊串流中的解決方案。

## 範例：播放視訊資料

- [範例：在 HTML 和中使用 HLS JavaScript](#)：顯示如何擷取 Kinesis 視訊串流的 HLS 串流工作階段，並在網頁中播放。

## 必要條件

- 在範例程式碼中，您可以指定您 AWS 在認證設定檔檔案中設定的設定檔，或在整合式開發環境 (IDE) 的 Java 系統屬性中提供認證來提供認證。如果您尚未這樣做，請先設定您的憑證。如需詳細資訊，請參閱[設定 AWS 認證和開發區域](#)。
- 我們建議您使用 Java IDE 檢視及執行程式碼，例如：
  - [Eclipse Java Neon](#)
  - [JetBrains IntelliJ 理念](#)

## 範例：Kinesis Video Streams 產生器 SDK 外掛程式

本主題說明如何建置 Amazon Kinesis Video Streams 製作工具開發套件，以做為 GStreamer 外掛程式使用。

### 主題

- [下載，構建和配置元素](#)
- [運行光元素](#)
- [啟動命令示例](#)
- [在碼頭容器中運行 GStreamer 元素](#)
- [元素參數參數](#)

[GStreamer](#) 是一種流行的媒體框架，由多個攝像機和視頻源通過組合模塊化插件來創建自定義媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化您現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。整合 GStreamer 後，您可以將網路攝影機或即時串流通訊協定 (RTSP) 攝影機影片串流至 Kinesis Video Streams，以便進行即時或稍後的播放、儲存和進一步分析。

GStreamer 外掛程式會自動管理您的視訊串流傳輸到 Kinesis Video Streams，方法是將 Kinesis Video Streams 訊串流製作工具開發套件所提供的功能封裝在 GStreamer 接收器元素中。kvssinkGStreamer 架構提供標準受管的環境以建構媒體流量裝置，例如相機或其他影片來源以進行進一步的處理、轉譯或儲存。

GStreamer 管道通常包含來源 (攝影機) 和目的元素 (其中為呈現影片的播放器，或離線擷取的儲存) 之間的連結。在這個範例中，您使用製作者開發套件元素做為目的或媒體目的地，對於視訊來源 (網路攝影機或 IP 相機)。封裝 SDK 的外掛程式元素，然後將視訊串流傳送至 Kinesis Video Streams。

本主題說明如何建構 GStreamer 媒體管線，該管線能夠從視訊來源串流視訊，例如網路攝影機或 RTSP 串流，通常透過中繼編碼階段 (使用 H.264 編碼) 連接至 Kinesis Video Streams。當您的視訊串流以 Kinesis 視訊串流的形式提供時，您可以使用 Kinesis 視訊串流剖析器程式庫來進一步處理、播放、儲存或分析您的視訊串流。

## 下載，構建和配置元素

GStreamer 外掛程式範例包含在 Kinesis Video Streams C++ 製作者 SDK 中。如需開發套件先決條件和下載的相關資訊，請參閱 [步驟 1：下載並設定 C++ 生產者程式庫程式碼](#)。

您可以建立 Producer SDK GStreamer 目的地作為 macOS、Ubuntu、Raspberry Pi，或 Windows 的動態程式庫。GStreamer 外掛程式位於您的 build 目錄中。要加載此插件，它必須在您的 GST\_PLUGIN\_PATH。執行以下命令：

```
export GST_PLUGIN_PATH=`pwd`/build
```

### Note

在 macOS 環境中，您只能在 Docker 容器內從網路攝影機執行 GStreamer 串流視訊。在 macOS 環境中不支援 Docker 容器內 USB 相機的串流視訊。

## 運行光元素

若要以 Kinesis Video Streams 製作者 SDK 元素做為接收器執行 GStreamer，請使用命令。gst-launch-1.0 使用適合 GStreamer 外掛程式使用的設定。例如，[v4l2src](#) 適用於在 Linux 系統上的 v4l2 裝置，或 [rtspsrc](#) 適用於 RTSP 裝置。指定 kvssink 做為目的 (最終目的地管道) 以傳送影片到製作者開發套件。

kvssink 元素具有下列必要參數：

- **stream-name**— 目的地 Kinesis 視訊串流的名稱。
- **storage-size**— 設備的存儲大小 (以千字節為單位)。如需設定裝置儲存的詳細資訊，請參閱 [StorageInfo](#)。
- **access-key**— 用來 AWS 存取 Kinesis 影片串流的存取金鑰。您必須提供此參數或 credential-path。



- **secret-key**— 用來存取 Kinesis Video Streams 的 AWS 秘密金鑰。您必須提供此參數或 `credential-path`。
- **credential-path**— 包含用於存取 Kinesis Video Streams 之認證的檔案路徑。如需輪換登入資料的詳細資訊，請參閱[管理 IAM 使用者的存取金鑰](#)。您必須提供此參數或 `access-key` 和 `secret-key`。

如需選用參數 `kvssink` 的詳細資訊，請參閱[元素參數參數](#)。

有關 GStreamer 插件和參數的最新信息，請參閱 [GStreamer 插件](#)，或使用以下命令列出選項：

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

如果構建失敗或 `GST_PLUGIN_PATH` 沒有正確設置，您的輸出將如下所示：

```
No such element or plugin 'kvssink'
```

## 啟動命令示例

以下示例演示瞭如何使用 GStreamer 插件從不同類型的設備流式傳輸視頻。

### 示例 1：在 Ubuntu 上從 RTSP 攝像機流式傳輸視頻

下列命令是以 Ubuntu 環境從網路 RTSP 相機串流建立的 GStreamer 管道，使用 [rtspsrc](#) GStreamer 外掛程式：

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

### 示例 2：在 Ubuntu 上從 USB 攝像機編碼和流式傳輸視頻

下列命令會在 Ubuntu 上建立 GStreamer 管線，以 H.264 格式將來自 USB 攝影機的串流進行編碼，並將其串流至 Kinesis Video Streams。這個範例使用了 [v4l2](#) src 外掛程式。

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-
raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc bframes=0 key-int-max=45
bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink
stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

### 範例 3：從 Ubuntu 上的 USB 攝影機串流預先編碼的視訊

下列命令會在 Ubuntu 上建立 GStreamer 管線，該管道會將攝影機已經以 H.264 格式編碼的視訊串流到 Kinesis Video Streams。這個範例使用了 [v4l2](#) src 外掛程式。

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

### 示例 4：在 macOS 上從網絡攝像機流式傳輸視頻

以下指令會在 macOS 上建立 GStreamer 管道，該管道可從網路攝影機將視訊串流至 Kinesis 視訊串流。此範例使用 [rtspsrc](#) GStreamer 外掛程式。

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

### 示例 5：在 Windows 上從網絡攝像機流式傳輸視頻

下列命令會在 Windows 上建立 GStreamer 管道，該管道可從網路攝影機將視訊串流至 Kinesis 視訊串流。此範例使用 [rtspsrc](#) GStreamer 外掛程式。

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

### 示例 6：從樹莓派上的攝像機流式傳輸視頻

下列命令會在樹莓派上建立 GStreamer 管道，將視訊串流到 Kinesis Video Streams。這個範例使用了 [v4l2](#) src 外掛程式。

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-raw,format=I420,width=640,height=480,framerate=30/1 ! omxh264enc control-rate=1 target-bitrate=5120000 periodicity-idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink
```

```
stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"  
aws-region="YourAWSRegion"
```

## 示例 7：從樹莓派上的攝像機流式傳輸視頻並指定區域

下列命令會在樹莓派上建立 GStreamer 管道，將視訊串流至美國東部 (維吉尼亞北部) 區域的 Kinesis Video Streams。這個範例使用了 [v4l2](#) src 外掛程式。

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert !  
video/x-raw,format=I420,width=640,height=480,framerate=30/1 !  
omxh264enc control-rate=1 target-bitrate=5120000 periodicity-  
idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-  
format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink  
stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"  
aws-region="YourAWSRegion"
```

## 示例 8：在樹莓派和 Ubuntu 中流式傳輸音頻和視頻

請參閱如何執行 [gst-launch-1.0 命令](#)，開始在 [Raspberry-PI](#) 和 [Ubuntu](#) 中串流音訊和視訊。

## 示例 9：在 macOS 中同時流式傳輸音頻和視頻

請參閱如何執行 [gst-launch-1.0 命令](#)，開始在 [MacOS](#) 中串流音訊和視訊。

## 範例 10：上傳包含音訊和視訊的 MKV 檔案

了解如何執行 [gst-launch-1.0 命令](#) 以上傳包含音訊和視訊的 MKV 檔案。

## 在碼頭容器中運行 GStreamer 元素

Docker 是一種開發、部署和執行應用程式的使用容器平台。使用 Docker 建立 GStreamer 管道可將 Kinesis Video Streams 的作業環境標準化，進而簡化應用程式的建置和使用。

要安裝與設定 Docker，請參閱以下內容：

- [Docker 下載說明](#)
- [Docker 入門](#)

安裝 Docker 之後，您可以使用命令從 Amazon 彈性容器登錄下載 Kinesis Video Streams C++ 生產者開發套件 (和 GStreamer 外掛程式)。`docker pull`

若要使用 Kinesis Video Streams 製作者 SDK 元素來執行 GStreamer 做為 Docker 容器中的接收器，請執行下列動作：

## 主題

- [驗證您的碼頭客戶端](#)
- [下載 Ubuntu、macOS、Windows 或 Raspberry Pi 的 Docker 影像](#)
- [運行碼頭映像](#)

## 驗證您的碼頭客戶端

向打算提取映像的 Amazon ECR 登錄檔驗證您的 Docker 用戶端。您必須為使用的每個註冊表獲取身份驗證令牌。令牌有效期為 12 小時。如需詳細資訊，請參閱《Amazon Elastic Container Registry 使用者指南》中的[登錄檔身分驗證](#)。

Example：使用 Amazon ECR 進行身分驗證

```
aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

如果成功，輸出會顯示 Login Succeeded。

## 下載 Ubuntu、macOS、Windows 或 Raspberry Pi 的 Docker 影像

視您的作業系統而定，使用下列命令下載 Docker 影像到您的 Docker 環境：

### 下載 Docker 影像到 Ubuntu 環境

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

### 下載 Docker 影像到 macOS

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

### 下載 Docker 影像到 Windows 環境

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

## 下載 Docker 影像到 Raspberry Pi

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

若要驗證已成功新增的影像，請執行下列命令：

```
docker images
```

## 運行碼頭映像

視您的作業系統而定，使用下列其中一個命令執行 Docker 影像：

### 在 Ubuntu 上運行碼頭映像

```
sudo docker run -it --network="host" --device=/dev/video0 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

### 在 macOS 上運行碼頭映像

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

### 在視窗上運行碼頭映像

```
docker run -it 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-windows <AWS_ACCESS_KEY_ID> <AWS_SECRET_ACCESS_KEY> <RTSP_URL> <STREAM_NAME>
```

### 在樹莓派上運行碼頭圖像

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi /bin/bash
```

Docker 會啟動容器，並顯示命令提示字元，讓您在容器中使用命令。

在容器中請使用下列命令來設定環境變數：

```
export LD_LIBRARY_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$LD_LIBRARY_PATH
```

```
export PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/bin:$PATH
export GST_PLUGIN_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$GST_PLUGIN_PATH
```

使用適合您裝置的 `gst-launch-1.0` 指令從相機開始串流。

對於使用 `gst-launch-1.0` 命令來連接到本地 Web 相機或網路 RTSP 相機的範例，請參閱 [啟動命令](#)。

## 元素參數參數

若要將視訊傳送至 Amazon Kinesis Video Streams 製作者開發套件，請指定 `kvssink` 為接收器或管道的最終目的地。此參考提供有關必要的 `kvssink` 和選用參數的資訊。如需詳細資訊，請參閱 [the section called “GStreamer”](#)。

### 主題

- [the section called “提供認證 kvssink”](#)
- [the section called “提供一個區域 kvssink”](#)
- [the section called “kvssink 必要參數”](#)
- [the section called “kvssink 可選參數”](#)

## 提供認證 kvssink

若要允許 `kvssink` GStreamer 元素向其提出請求 AWS，請提供 AWS 登入資料供其呼叫 Amazon Kinesis Video Streams 服務時使用。認證提供者鏈結會依下列順序尋找認證：

### 1. AWS IoT 認證

若要設定 AWS IoT 認證，請參閱 [the section called “使用以下方式控制 Kinesis Video Streams 資源的存取 AWS IoT”](#)。

`iot-credentials` 參數值必須以下列 `### = 值` 配對的逗號分隔清單開頭，然後再加上逗號分隔的清 `#`。 `iot-certificate`,

金鑰	必要	描述
<code>ca-path</code>	是	用於透過 TLS 與後端服務建立信任之 CA 憑證的檔案路徑。

金鑰	必要	描述
		<p>Example</p> <p>範例：<code>/file/path/to/certificate.pem</code></p>
cert-path	是	<p>X.509 憑證的檔案路徑。</p> <p>Example</p> <p>範例：<code>/file/path/to/certificateID -certificate.pem.crt</code></p>
endpoint	是	<p>您 AWS 帳戶的 AWS IoT Core 認證端點提供者端點。請參閱 <a href="#">開AWS IoT 發人員指南</a>。</p> <p>Example</p> <p>範例：<code>credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com</code></p>
key-path	是	<p>公開/私密 key pair 中使用之私密金鑰的檔案路徑。</p> <p>Example</p> <p>範例：<code>/file/path/to/certificateID -private.pem.key</code></p>

金鑰	必要	描述
role-aliases	是	連線時要使用的 AWS IAM 角色指向的角色别名名称 AWS IoT Core。  Example  範例： <i>KvsCameraIoTRoleAlias</i>
iot-thing-name	否	iot-thing-name 是選擇性的。如果iot-thing-name 未提供，則會使用stream-name 參數值。  Example  範例： <i>kvs_example_camera</i>

## Example

範例：

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  iot-certificate="iot-certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com,cert-path=certificateID-certificate.pem.crt,key-path=certificateID-private.pem.key,ca-path=certificate.pem,role-aliases=YourRoleAlias,iot-thing-name=YourThingName"
```

## 2. 環境變數

若要kvssink使用環境中的認證，請設定下列環境變數：



環境變數名稱	必要	描述
AWS_ACCESS_KEY_ID	是	用來 AWS 存取 Amazon Kinesis Video Streams 的存取金鑰。
AWS_SECRET_ACCESS_KEY	是	與存取金鑰相關聯的 AWS 秘密金鑰。
AWS_SESSION_TOKEN	否	如果您直接從 AWS STS 作業使用臨時安全認證，則指定所需的工作階段 Token 值。

設定環境變數會變更使用的數值，直到 Shell 工作階段結束或直到您將該變數設為其他數值。要使變量在 future 的會話中持續存在，請在 shell 的啟動腳本中設置它們。

### 3. **access-key** , **secret-key** 參數

若要直接將認證指定為 kvssink 參數，請設定下列參數：

kvssink 參數名稱	必要	描述
access-key	是	用來 AWS 存取 Amazon Kinesis Video Streams 的存取金鑰。
secret-key	是	與存取金鑰相關聯的 AWS 秘密金鑰。
session-token	否	如果您直接從 AWS STS 作業使用臨時安全認證，則指定所需的工作階段 Token 值。

#### Example

使用靜態認證：

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

## Example

使用臨時登入資料：

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

## 4. 認證檔案

### ⚠ Important

如果您已選取先前的其中一個方法，則無法使用該 `credential-filekvssink` 參數。

kvssink參數名稱	必要	描述
credential-file	是	包含特定格式認證的文字檔案路徑。

文字檔案必須包含下列其中一種格式的認證：

- 認證 `YourAccessKeyYourSecretKey`
- 認證 `YourAccessKey## YourSecretKeySessionToken`

## Example

範例：您的 `credentials.txt` 檔案位於 `/home/ubuntu` 並包含下列內容：

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEEXAMPLE STEXAMPLE
```

要在中使用它 kvssink，請鍵入：

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  credential-file="/home/ubuntu/credentials.txt"
```

**Note**

future 的到期時間應該至少為  $5 + 30 + 3 = 38$  秒。寬限期定義為中的 IOT\_CREDENTIAL\_FETCH\_GRACE\_PERIOD 變數 [IotCredentialProvider.h](#)。如果認證在啟動時太接近到期時 kvssink，您會收到錯誤碼 0x52000049 - STATUS\_INVALID\_TOKEN\_EXPIRATION。

**Important**

kvssink 不會修改憑證檔案。如果您使用臨時認證，則在到期時間減去寬限期之前，必須由外部來源更新認證檔案。

## 提供一個區域 kvssink

以下是區域查尋順序：

1. AWS\_DEFAULT\_REGION 環境變量首先審查。如果已設定，則會使用該區域來設定用戶端。
2. aws-region 接下來會檢閱參數。如果已設定，則會使用該區域來設定用戶端。
3. 如果沒有使用先前的方法，則 kvssink 預設為 us-west-2。

## kvssink 必要參數

除了提供認證和區域之外，kvssink 元素還具有下列必要參數：



stream-name-目的地 Amazon Kinesis 視訊串流的名稱。

## kvssink 可選參數

kvssink 元素具有下列選用參數：如需這些參數的相關資訊，請參閱 [Kinesis 視頻流結構](#)。

參數	描述	單位/類型	預設
absolute-fragment-times	是否使用絕對片段時間。	Boolean	true

參數	描述	單位/類型	預設
access-key	<p>用來 AWS 存取 Kinesis 影片串流的存取金鑰。</p> <p>您必須設定 AWS 認證或提供此參數。若要提供此資訊，請輸入下列內容：</p> <pre>\$ export   AWS_ACCES   S_KEY_ID=</pre>		
avg-bandwidth-bps	串流的預期平均頻寬。	每秒的位元組	4194304

參數	描述	單位/類型	預設
aws-region	<p>要 AWS 區域 使用的。</p> <div data-bbox="472 352 792 1003" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>您也可以為區域提供 <code>AWS_DEFAULT_REGION</code> 環境變數。如果同時設定了環境變數和 <code>kvssink</code> 參數，則環境變數的優先順序。</p> </div> <div data-bbox="472 1066 792 1434" style="border: 1px solid #ff9999; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Important</b></p> <p><code>us-west-2</code> 如果沒有其他指定，則該區域將默認為。</p> </div>	字串	us-west-2
buffer-duration	串流緩衝持續時間。	秒鐘	180
codec-id	串流的編解碼器 ID。	字串	"V_MPEG4/ISO/AVC"
connection-staleness	之後的時間，該時間流不穩定回調被調用。	秒鐘	60

參數	描述	單位/類型	預設
content-type	串流的内容類型。	字串	"video/h264"
fragment-acks	發生錯誤時欲採取的動作。	Boolean	true
fragment-duration	您需要的片段持續時間。	毫秒	2000
framerate	預期的影格率。	每秒影格	25
frame-timecodes	是否使用影格時間碼或者使用目前時間回呼產生時間戳記。	Boolean	true
key-frame-fragmentation	是否在關鍵影格上產生片段。	Boolean	true
log-config	日誌設定路徑。	字串	"./kvs_log_configuration"
max-latency	串流的最大延遲。	秒鐘	60
recalculate-metrics	是否重新計算指標。	Boolean	true
replay-duration	若發生錯誤啟用重新啟動時，則向後捲動目前閱讀器以重放的持續時間。	秒鐘	40
restart-on-error	是否發生錯誤時重新啟動。	Boolean	true
retention-period	串流是保留所需的時間。	小時	2

參數	描述	單位/類型	預設
rotation-period	金鑰輪換期間。如需更多的詳細資訊，請參閱 <a href="#">輪換客戶主金鑰</a> 。	秒鐘	2400
secret-key	<p>用來存取 Kinesis Video Streams 的 AWS 秘密金鑰。</p> <p>您必須設定 AWS 認證或提供此參數。</p> <pre>\$ export   AWS_SECRE   T_ACCESS_   KEY_ID=</pre>		
session-token	如果您直接從 AWS STS 作業使用臨時安全認證，則指定所需的工作階段 Token 值。		
storage-size	裝置儲存空間大小 MegaBytes (MB)。如需設定裝置儲存的詳細資訊，請參閱 <a href="#">StorageInfo</a> 。	MegaBytes	128
streaming-type	<p>串流類型。有效值包含：</p> <ul style="list-style-type: none"> <li>• 0：即時</li> <li>• 1：近乎即時 (目前不支援)</li> <li>• 2：離線</li> </ul>	列舉 GstKvsSinkStreamingType	0：即時

參數	描述	單位/類型	預設
timecode-scale	MKV 時間碼比例。	毫秒	1
track-name	MKV 音軌名稱。	字串	"kinesis_video"



參數	描述	單位/類型	預設
iot-certificate	<p>AWS IoT 要在 kvssink 元素中使用的憑證。</p> <p>iot-certificate 接受以下鍵和值：</p> <div data-bbox="472 575 792 1083" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>iot-thing-name 是選擇性的。如果 iot-thing-name 未提供，則會使用 stream-name 參數值。</p> </div> <ul style="list-style-type: none"> <li>• endpoint=iotcredentialsproviderendpoint</li> <li>• cert-path=/localdirectorypath/to/certificate</li> <li>• key-path=/localdirectorypath/to/private/key</li> </ul>	字串	無

參數	描述	單位/類型	預設
	<ul style="list-style-type: none"> <li>ca-path=/localdirectorypath/to/ca-cert</li> <li>role-aliases =role-aliases</li> <li>iot-thing-name=YourIotThingName</li> </ul>		

## 範例：使用 API 將資料傳送至 Kinesis Video Streams PutMedia

這個例子演示瞭如何使用 [PutMedia](#) API。它顯示瞭如何發送已經在容器格式 ( MKV ) 的數據。如果在傳送之前必須將資料組合成容器格式 (例如，如果要將攝影機視訊資料組合到影格中)，請參閱 [室壁運動視頻流生產者庫](#)。

### Note

此 PutMedia 作業只能在 C++ 和 Java 開發套件中使用。這是由於對連線、資料流程和確認進行全雙工管理所致。它不支持其他語言。

本範例包含下列步驟：

- [步驟 1：下載並設定程式碼](#)
- [步驟 2：編寫並檢查代碼](#)
- [步驟 3：運行並驗證代碼](#)

## 步驟 1：下載並設定程式碼

請依照下列步驟下載 Java 範例程式碼、將專案匯入 Java IDE、設定程式庫位置，以及設定程式碼以使用您的 AWS 認證。

1. 創建一個目錄並從 GitHub 存儲庫克隆示例源代碼。PutMedia 範例為 [Java Producer Library](#) 的一部分。

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

2. 打開您正在使用的 Java IDE (例如, [日食](#) 或 [IntelliJ IDEA](#)) , 然後導入您下載的阿帕奇 Maven 項目 :

- 在 Eclipse 中 : 請選擇 File (檔案)、Import (匯入)、Maven、Existing Maven Projects (現有 Maven 專案) , 並瀏覽至下載套件的根目錄。選取 pom.xml 檔案。
- 在 IntelliJ Idea : 選擇 Import (匯入)。瀏覽至已下載套件根目錄的 pom.xml 檔案。

如需更多詳細資訊, 請參閱 IDE 文件。

3. 請更新專案, 讓 IDE 能夠找到您匯入的程式庫。

- 對於 IntelliJ IDEA 請執行下列步驟 :
  - a. 開啟專案 lib 目錄的內容 (按一下右鍵) 功能表, 然後選擇 Add as library (新增為程式庫)。
  - b. 選擇檔案, 然後選擇「專案結構」。
  - c. 請在 Project Settings (專案詳細資訊) 下選擇 Modules (模組)。
  - d. 請在 Sources (來源) 標籤將 Language Level (語言層級) 設定為 7 或以上。
- 對於 Eclipse 請執行下列步驟 :
  - a. 開啟專案的內容 (按一下右鍵) 功能表, 然後選擇 Properties (屬性)、Java Build Path (Java 建置路徑)、Source (來源)。然後執行下列動作 :
    1. 請在 Source (來源) 標籤按兩下 Native library location (原生程式庫位置)。
    2. 請在 Native Library Folder Configuration (原生程式庫資料夾設定) 精靈之中選擇 Workspace (工作空間)。
    3. 請在 Native Library Folder (原生程式庫資料夾) 選項之中選擇專案的 lib 目錄。
  - b. 開啟專案的內容 (按一下右鍵) 功能表, 然後選擇 Properties (屬性)。然後執行下列動作 :
    1. 請在 Libraries (程式庫) 標籤選擇 Add Jars (新增 Jar)。
    2. 請在 JAR selection (JAR 選擇) 精靈之中選擇專案 lib 目錄之中的所有 .jar。

## 步驟 2：編寫並檢查代碼

PutMedia API 範例 (PutMediaDemo) 顯示下列編碼模式：

主題

- [創建 PutMediaClient](#)
- [串流媒體並暫停執行緒](#)

本節程式碼範例來自 PutMediaDemo 類別。

### 創建 PutMediaClient

建立 PutMediaClient 物件需要下列參數：

- PutMedia 端點的 URI。
- InputStream 指向 MKV 檔案進行串流。
- 串流名稱。本範例使用 [使用 Java Producer Library](#) (my-stream) 建立的串流。如欲使用不同串流，請變更下列參數：

```
private static final String STREAM_NAME="my-stream";
```

#### Note

PutMediaAPI 範例不會建立串流。您必須使用 Kinesis Video Streams 主控台的測試應用程式或 [使用 Java Producer Library AWS CLI](#)

- 目前的時間戳記。
- 時間編碼類型。範例使用 RELATIVE，顯示時間戳記與容器開始有關。
- AWSKinesisVideoV4Signer 物件驗證收到的封包是由獲得授權的寄件者傳送。
- 以 Kbps 為單位的最大上游頻寬。
- AckConsumer 物件接收封包收到的確認。

下列程式碼建立 PutMediaClient 物件：

```
/* actually URI to send PutMedia request */  
final URI uri = URI.create(KINESIS_VIDEO_DATA_ENDPOINT + PUT_MEDIA_API);
```

```
/* input stream for sample MKV file */
final InputStream inputStream = new FileInputStream(MKV_FILE_PATH);

/* use a latch for main thread to wait for response to complete */
final CountDownLatch latch = new CountDownLatch(1);

/* a consumer for PutMedia ACK events */
final AckConsumer ackConsumer = new AckConsumer(latch);

/* client configuration used for AWS SigV4 signer */
final ClientConfiguration configuration = getClientConfiguration(uri);

/* PutMedia client */
final PutMediaClient client = PutMediaClient.builder()
    .putMediaDestinationUri(uri)
    .mkvStream(inputStream)
    .streamName(STREAM_NAME)
    .timestamp(System.currentTimeMillis())
    .fragmentTimeCodeType("RELATIVE")
    .signWith(getKinesisVideoSigner(configuration))
    .upstreamKbps(MAX_BANDWIDTH_KBPS)
    .receiveAcks(ackConsumer)
    .build();
```

## 串流媒體並暫停執行緒

建立用戶端之後，範例開始與 `putMediaInBackground` 進行非同步串流。主要執行緒將由 `latch.await` 暫停直到 `AckConsumer` 傳回為止，屆時用戶端將會關閉。

```
/* start streaming video in a background thread */
    client.putMediaInBackground();

    /* wait for request/response to complete */
    latch.await();

    /* close the client */
    client.close();
```

## 步驟 3：運行並驗證代碼

如欲執行 PutMedia API 範例，請執行下列步驟：

1. 建立my-stream在 Kinesis 影片串流主控台中命名的串流，或使用 AWS CLI。
2. 將工作目錄變更為 Java 製作者開發套件目錄：

```
cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-producer-sdk-java/
```

3. 編譯 Java 開發套件和示範應用程式：

```
mvn package
```

4. 在目錄中建立一個 /tmp 暫時的檔案名稱：

```
jar_files=$(mktemp)
```

5. 建立從本機儲存庫 classpath 有相依性字串的檔案：

```
mvn -Dmdep.outputFile=$jar_files dependency:build-classpath
```

6. 設定 LD\_LIBRARY\_PATH 環境變數的值如下：

```
export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$LD_LIBRARY_PATH
$ classpath_values=$(cat $jar_files)
```

7. 從命令列執行示範，如下所示，提供您的 AWS 認證：

```
java -classpath target/kinesisvideo-java-demo-1.0-SNAPSHOT.jar:$classpath_values -Daws.accessKeyId=${ACCESS_KEY} -Daws.secretKey=${SECRET_KEY} -Djava.library.path=/opt/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build com.amazonaws.kinesisvideo.demoapp.DemoAppMain
```

8. 開啟 [Kinesis Video Streams 主控台](#)，然後在「管理串流」頁面上選擇您的串流。影片於 Video Preview (影片預覽) 窗格播放。

## 範例：從 RTSP 來源進行串流

[C++ Producer Library](#) 包含定義 [Docker](#) 容器連接到 RTSP (即時串流通訊協定) 的網路攝影機。使用 Docker 可將 Kinesis Video Streams 的作業環境標準化，進而簡化應用程式的建置和使用。

下列程序示範如何設定及使用 RTSP 示範應用程式。

## 主題

- [必要條件](#)
- [建立泊塢視窗映像](#)
- [執行 RTSP 範例應用程式](#)

## 必要條件

若要執行 Kinesis Video Streams RTSP 範例應用程式，您必須具備下列項目：

- Docker：如需有關安裝和使用 Docker 的詳細資訊，請參閱以下連結：
  - [Docker 下載說明](#)
  - [Docker 入門](#)
- RTSP 網路攝影機原始碼：有關攝影機建議的詳細資訊，請參閱[系統要求](#)。

## 建立泊塢視窗映像

首先，構建演示應用程式將在其中運行的 Docker 映像。

1. 複製 Amazon Kinesis Video Streams 示範儲存庫。

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. 切換到包含碼頭文件的目錄。在這種情況下，它是[碼頭-r tsp](#)目錄。

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. 使用以下命令來構建 Docker 映像。此指令會建立影像，並將其標記為 `rtspdockertest`。

```
docker build -t rtspdockertest .
```

4. 執行 `docker images` 並搜尋標記為的影像 ID `rtspdockertest`。

例如，在下面的示例輸出中，IMAGE ID 是 `54f0d65f69b2`。

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
	BLOB SIZE				

```
rtspdockertest    latest    54f0d65f69b2    10 minutes ago    linux/arm64    653.1
MiB    292.4 MiB
```

在稍後的步驟中，您將需要此功能。

## 執行 RTSP 範例應用程式

您可以從 Docker 容器內部或外部執行 RTSP 範例應用程式。請遵循以下適當的指示。

### 主題

- [在碼頭容器內](#)
- [碼頭容器外](#)

### 在碼頭容器內

#### 執行 RTSP 範例應用程式

1. 使用下列命令啟動 Amazon Kinesis Video Streams 泊塢視訊串流碼頭容器：

```
docker run -it YourImageId /bin/bash
```

2. 若要啟動範例應用程式，請提供您的 AWS 登入資料、Amazon Kinesis 視訊串流的名稱，以及 RTSP 網路攝影機的 URL。

#### Important

如果您正在使用臨時憑據，則還需要提供您的 `AWS_SESSION_TOKEN`。請參閱下面的第二個例子。

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

臨時登入資料：

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
```



```
export AWS_SECRET_ACCESS_KEY_ID=YourSecretKeyId
export AWS_SESSION_TOKEN=YourSessionToken
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

3. 登入 AWS Management Console 並開啟 [Kinesis Video Streams 主控台](#)。

檢視串流。

4. 若要結束 Docker 容器，請關閉終端機視窗或鍵入 `exit`。

## 碼頭容器外

從 Docker 容器外部，使用以下命令：

```
docker run -it YourImageId /bin/bash -c "export AWS_ACCESS_KEY_ID=YourAccessKeyId;
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId; export
AWS_SESSION_TOKEN=YourSessionToken; export AWS_DEFAULT_REGION=YourAWSRegion; ./
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

## 範例：剖析和呈現 Kinesis Video Streams 片段

[串流剖析器程式庫](#) 包含名為的示範應用程式，`KinesisVideoRendererExample` 該應用程式示範剖析和轉譯 Amazon Kinesis 視訊串流片段。範例使用 [JCodec](#) 解碼 H.264 編碼影格 (以 [範例：Kinesis Video Streams 產生器 SDK 外掛程式](#) 應用程式擷取)。使用 JCodec 解碼影格後，將使用 [JFrame](#) 轉譯可見影像。

此範例說明如何執行下列步驟：

- 使用 `GetMedia` API 從 Kinesis 視訊串流擷取畫面，並呈現串流以供檢視。
- 在自訂應用程式中檢視串流的視訊內容，而不是使用 Kinesis Video Streams 主控台。

您也可以使用此範例中的類別來檢視未編碼為 H.264 的 Kinesis 視訊串流內容，例如在顯示前不需要解碼的 JPEG 檔案串流。

下列程序示範如何設定及使用 `Renderer` 示範應用程式。

## 必要條件

為了檢驗及使用 `Renderer` 範例程式庫，您必須具有下列項目：

- Amazon Web Services ( AWS ) 帳戶。如果您還沒有 AWS 帳戶，請參閱 [Kinesis Video Streams 入門](#)。
- 一個 Java 的集成開發環境 ( IDE )，如 [日食 Java 霓虹燈](#) 或 [JetBrains IntelliJ 理念](#)。

## 執行轉譯器範例

1. 建立目錄，然後從 GitHub 儲存庫複製範例原始程式碼。

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. 開啟您正在使用的 Java IDE (例如 [Eclipse](#) 或 [IntelliJ IDEA](#))，然後匯入您之前下載的 Apache Maven 專案：

- 在 Eclipse 中：選擇 File (檔案)、Import (匯入)、Maven (Maven)、Existing Maven Projects (現有 Maven 專案)。請前往 `kinesis-video-streams-parser-lib` 目錄。
- 在 IntelliJ Idea：選擇 Import (匯入)。巡覽至已下載套件根目錄的 `pom.xml` 檔案。

### Note

如果 IntelliJ 找不到相依性，您可能需要執行下列：

- Build clean: (清除項目) Choose File (檔案)、Settings (設定)、Build, Execution, Deployment (建立、執行、部署)、Compiler (編譯器)。確認已選取 [重建時清除輸出目錄]，然後選擇 [建置] > [建置專案]。
- Reimport the project (重新匯入專案)：開啟 (用滑鼠右鍵按一下) 專案的內容功能表，然後選擇 Maven、Reimport (重新匯入)。

如需更多詳細資訊，請參閱 IDE 文件。

3. 請由 Java IDE 開啟 `src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRenderExampleTest`。
4. 由檔案移除 `@Ignore` 指令。
5. 使用 Kinesis 視訊串流的名稱更新 `.stream` 參數。
6. 執行 `KinesisVideoRenderExample` 測試。

## 運作方式

範例應用程式可示範下列項目：

- [傳送 MKV 資料](#)
- [將 MKV 片段解析為幀](#)
- [解碼和顯示框架](#)

### 傳送 MKV 資料

該示例從rendering\_example\_video.mkv文件發送示例 MKV 數據，用PutMedia於將視頻數據發送到名為的流render-example-stream。

應用程式建立 PutMediaWorker：

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

如需更多詳細資訊了解 PutMediaWorker 類別，請參閱 [呼叫 PutMedia](#) (位於 [串流剖析器程式庫](#) 文件)。

### 將 MKV 片段解析為幀

範例之後使用 GetMediaWorker 擷取及剖析串流之中的 MKV 片段：

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getFrameVisitor());
executorService.submit(getMediaWorker);
```

如需更多詳細資訊了解 GetMediaWorker 類別，請參閱 [呼叫 GetMedia](#) (位於 [串流剖析器程式庫](#) 文件)。

## 解碼和顯示框架

範例之後使用 [JFrame](#) 解碼及顯示影像。

下列程式碼範例來自 `KinesisVideoFrameViewer` 類別，延伸 `JFrame`：

```
public void setImage(BufferedImage bufferedImage) {  
    image = bufferedImage;  
    repaint();  
}
```

此影像會顯示為 [java.awt.image](#) 的執行個體。 [BufferedImage](#)。如需如何使用 `BufferedImage` 的範例，請參閱[讀取/載入影像](#)。

## 範例：使用識別視訊串流中的物件 SageMaker

本範例示範如何建立解決方案，[SageMaker](#)以識別某些物件何時出現在 Amazon Kinesis 影片串流中。[SageMaker](#) 是開發人員和資料科學家的受管理平台，可讓您快速輕鬆地建置、訓練和部署機器學習模型。

此範例包含包含應用程式功能的 [Docker](#) 容器，以及可自動部署應用程式資源的範本 [AWS CloudFormation](#)。AWS

AWS CloudFormation 範本會建立下列資源：

- [Amazon Elastic Container Service \(Amazon ECS\)](#) 叢集，使用執行程式庫軟體的 [AWS Fargate](#) 運算引擎。
- [Amazon DynamoDB](#) 表格，可在 Fargate 任務上執行的工作者間維護檢查點和相關狀態。
- [Kinesis 資料串流](#)，可擷取從中產生的推論輸出。 [SageMaker](#)
- 解析來源 [SageMaker](#)輸出的 [AWS Lambda 函數](#)。
- AWS Identity and Access Management 提供跨服務存取權的 [\(IAM\)](#) 資源。
- 用於監控應用程式的 [Amazon CloudWatch](#) 資源。

該應用程式與處理數據的任何 [SageMaker](#) 端點兼容。此範例包含建立使用範例物件識別演算法範本之 [SageMaker](#) 端點的指示。您可以根據應用程式的使用案例和要求來修改或取代演算法。

### 主題

- [必要條件](#)

- [建立應用程式](#)
- [監控應用程式](#)
- [擴充應用程式](#)
- [清理應用程式](#)

## 必要條件

範例應用程式有以下先決條件：

- [SageMaker](#)
- [Kinesis 視訊串流](#)
- [服務連結角色](#)

## SageMaker

此範例需要 SageMaker 筆記本。如需有關建立筆記本的資訊，請參閱 Amazon SageMaker 開發人員指南中的[建立筆記本執行個體](#)。建立您的筆記型電腦時請注意以下事項：

- 將範Amazon\_JumpStart\_Object\_Detection.ipynb例 (從 Jupyter 主控台的「SageMaker 範例」索引標籤中的「Amazon 演算法簡介」區段) 新增至筆記本。
- 建立 Amazon Simple Storage Service (Amazon S3) 儲存貯體，並在新增範例時在先決條件步驟中提供其名稱。
- 建立筆記本之後，請在 SageMaker 主控台上選擇 [端點組態]，並記下端點名稱。

## Kinesis 視訊串流

此範例需要一或多個具有即時視訊資料的 Kinesis 視訊串流。如需有關建立 Kinesis 視訊串流並從攝影機傳送資料至該串流的資訊，請參閱[GStreamer](#)。記下您的 Kinesis 視訊串流名稱。

## 服務連結角色

此示例要求您的帳戶具有服務鏈接角色才能執行 Fargate 操作。新 AWS 帳戶 的預設會啟用此角色。如果建立應用程式時看到以下錯誤，您必須啟用服務連結的角色：

```
Unable to assume the service linked role. Please verify that the ECS service linked role exists
```

若要啟用服務連結的角色，請執行下列命令：

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

## 建立應用程式

若要建立範例應用程式，請使用 AWS CloudFormation 和所提供的範本。

用來建 AWS CloudFormation 立應用程式

1. 登入 AWS Management Console 並使用您的下列其中一個連結開啟 AWS CloudFormation 主控台 AWS 區域。連結會啟動您區域的正確堆疊：
  - [於亞太區域 \(雪梨\) 地區 \(AP-東南 -2\) 推出](#)
  - [於亞太區域 \(東京\) 地區 \(東北 1\) 推出](#)
  - [在歐洲 \(法蘭克福\) 區域 \(歐盟-中部 -1\) 推出](#)
  - [在歐洲 \(愛爾蘭\) 區域 \(歐洲-西部 -1\) 推出](#)
  - [在美國東部 \(維吉尼亞北部\) 區域啟動 \(us-east-1\)](#)
  - [於美國西部 \(奧勒岡\) 區域 \(美國西部 -2\) 啟動](#)
2. 在 Create Stack (建立堆疊) 頁面上，提供下列值：
  - 為堆疊指定唯一名稱 (例如，####稱-KVS-SageMaker)。
  - 提供您在 SageMaker 上一節中建立的端點名稱 (不是端點 ARN)。
  - 提供 Kinesis 視訊串流的名稱。如果您有多個 Kinesis 視訊串流，請以引號提供串流名稱，並以逗號分隔。
  - 維持其他設定。

選擇下一步。

3. 在 Options (選項) 頁面上，維持原設定。
4. 選取 [我確認 AWS CloudFormation 可能會建立 IAM 資源] 核取方塊。選擇下一步。

AWS CloudFormation 建立應用程式。

下表列出當您使用此 AWS CloudFormation 範本建立堆疊時，Docker 容器所使用的數個參數。這些參數的值會預先定義在此範本的 SSM 資源中，但您可以視需要自訂這些值。

資源名稱	預設值	描述
inferenceInterval	6	The sampling ratio for video frames that are sent to the SageMaker endpoint. The default value of 6 means that 6 frames are sent to the SageMaker endpoint.
sageMakerTaskQueueSize	5000	The size of the queue that maintains the pending requests to the SageMaker endpoint. The size of the queue is affected by 'inferenceInterval' and 'sageMakerTaskThreadPoolSize'. If SageMaker inference takes longer, requests are buffered in this queue.
sageMakerTaskThreadPoolSize	20	Number of threads that's used to concurrently carry out SageMaker inference requests.
sageMakerTaskTimeoutInMilli	20000	The maximum duration accepted for a single request (or a retry request) to the SageMaker endpoint.
sageMakerTaskThreadPoolName	SageMakerThreadPool-%d	The name of the threadpool that's sending requests to the SageMaker endpoint.

若要自訂這些參數的值，請在「AWS CloudFormation 建立堆疊」頁面中選擇樣板 URL 來下載樣板，然後在樣板Params段落中找出這些參數，如下所示：

```
Params:
  Type: AWS::SSM::Parameter
  Properties:
    Name:
      Ref: AppName
    Description: "Configuration for SageMaker app"
    Type: String
    Value:
      Fn::Sub: |
        {"streamNames":[${StreamNames}], "tagFilters":
[${TagFilters}], "sageMakerEndpoint": "${SageMakerEndpoint}",
        "endPointAcceptContentType": "${EndPointAcceptContentType}",

        "kdsStreamName": "${Kds}", "inferenceInterval": 6, "sageMakerTaskQueueSize": 5000,
        "sageMakerTaskThreadPoolSize": 20, "sageMakerTaskTimeoutInMilli": 20000,
        "sageMakerTaskThreadPoolName": "SageMakerThreadPool-%d"}
```

## 監控應用程式

範本建立的應用程式包括 Amazon CloudWatch 儀表板 AWS CloudFormation 板和用於監控應用程式指標和事件的 CloudWatch 日誌串流。

### 應用儀表板

該應用程序包括用於監視應用程序指標的 CloudWatch 儀表板 若要檢視應用程式儀表板，請在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台，然後選擇左側導覽列中的 [儀表板]。

選擇 KVS-SageMaker 驅動程式 KvsSageMakerIntegration--**aw**s-region 儀表板。儀表板會顯示下列資訊：

- 影格指標：用於處理視訊串流、將影格傳送至 SageMaker 端點，以及寫入 Kinesis 資料串流的度量，該資料串流會連接 SageMaker 筆記本與處理 SageMaker 推論輸出結果的 AWS Lambda 函數。
- IngestToProcessLatency: 將視訊影格擷取到 Kinesis Video Streams 服務與應用程式接收影格之間的時間差異。
- 目前租用總額：應用程式被授與使用租用從 Kinesis 視訊串流讀取的權限。此指標顯示有效租約的數量。應用程式在每個 Kinesis 視訊串流使用一個租用，而一個租用則用於串流之間的同步處理。
- 租約同步指標：許可租約同步的頻率和持續時間。
- LeaseCount 每個工作者：工作者執行緒之間的 SageMaker 租賃分配。
- 工作者數目：處理串流的 SageMaker 工作者數目。Amazon ECS 叢集中的每個任務都有一個工作者正在執行。一個工作者可以處理多個串流。
- ECS 服務使用率：Amazon ECS 叢集的使用量指標。
- KinesisDataStream：Kinesis 資料串流的使用量度。
- SageMaker：SageMaker 筆記型電腦執行的作業。
- Lambda：處理 SageMaker 筆記本輸出的 Lambda 函數的數量和持續時間。

如果這些圖表中的任何資訊指出營運問題 (例如某值穩定成長而非穩定不變)，請參閱以下區段，以了解如何讀取應用程式日誌來判斷問題。

### CloudWatch 日誌

該應用程序包括兩個 CloudWatch 日誌：



## 主題

- [應用程式記錄](#)
- [Lambda 函數日誌](#)

### 應用程式記錄

您可以使用應用程式日誌來監控應用程式事件和錯誤情況。如果您需要聯絡產品支援以解決問題，也可以使用此記錄檔。

#### 讀取應用程式日誌

1. 開啟 Amazon ECS 主控台，網址為 <https://console.aws.amazon.com/ecs>。
2. 選擇 KVS-Sagemaker-Driver 叢集。
3. 在「服務」選項 #####-SageMakerDriverService 服務。
4. 選擇 Logs (日誌) 索引標籤。

應用程式日誌會顯示事件，例如初始化、組態和租用活動。

### Lambda 函數日誌

您可以使用 Lambda 函數記錄來追蹤成功的物件識別。

#### 若要讀取 Lambda 記錄

1. [請在以下位置開啟 AWS Lambda 主控台。](https://console.aws.amazon.com/lambda) <https://console.aws.amazon.com/lambda>
2. 為您的應用程式選擇 Lambda 函數。Lambda 函數名稱的格式如下：

```
stack-name-LambdaFunction-A1B2C3D4E5F6G
```

3. 選擇 Monitoring (監控) 面板。
4. 選擇 [檢視登入] CloudWatch。

應用程式的 CloudWatch 記錄會顯示 Kinesis 視訊串流和其他應用程式事件中物件的成功識別。

## 擴充應用程式

您可以透過修改 AWS CloudFormation 範本視窗中提供的值，將自訂功能新增至應用程式，如下所示：

- `EndPointAcceptContentType`：如果您的 SageMaker 端點不接受 JPG 格式的框架，則可以變更此值。下列是支援的格式：
  - `image/jpeg`
  - `image/png`
  - `image/bmp`
  - `image/gif`
  - `application/x-image`
- `LambdaFunctionBucket`，`LambdaFunctionKey`：提供的設置使用處理 SageMaker 輸出並將其寫入 CloudWatch 日誌的 AWS Lambda 函數。如果您想將 SageMaker 輸出發送到其他地方，則可以提供自己的 Lambda 函數。
- 標籤篩選：如果您有使用 [the section called "TagStream"](#) 動作加上標籤的串流，您可以指定要處理的串流標籤。例如，如果您有兩個串流，其 `Location` 金鑰具有 `Front` 和 `Parking` 值，您可以使用下列項目篩選只使用這些串流：

```
{"key":"Location","values":["Front","Parking"]}
```

## 清理應用程式

在您完成為此教學課程建立的應用程式後，建議您刪除所有不想保留的資源，以避免產生任何持續費用。

1. SageMaker 端點：如果您為本教學課程建立了 SageMaker 端點，而不是使用現有端點，請刪除端點。在 [SageMaker 控制台] 中，選擇 [端點設定]。選擇您建立的端點，然後選擇 [動作] > [刪除]。確認刪除。
2. SageMaker 筆記本：在 SageMaker 控制台上，選擇筆記本執行個體。選擇您建立的筆記型電腦，然後選擇 Actions (動作)、Stop (停止)。當筆記型電腦顯示其 Status (狀態) 為 Stopped (已停止)，選擇 Actions (動作)、Delete (刪除)。確認刪除。

### Note

如需有關清理 [SageMaker](#) 資源的詳細資訊，請參閱 [SageMaker 開發人員指南](#) 中的清理。

3. SageMaker 執行政策：在 IAM 主控台的導覽窗格中，選擇 [政策]。選擇您為此教學課程建立的政策。策略的名稱類似於以下內容：`AmazonSageMaker-ExecutionPolicy-timestamp`。

選擇 政策動作，然後 刪除。確認刪除。

4. SageMaker 執行角色：在 IAM 主控台的導覽窗格中，選擇 [角色]。選擇您為此教學課程建立的角色。角色的名稱類似於以下內容：AmazonSageMaker-ExecutionRole-*timestamp*。

選擇 Delete role (刪除角色)。確認刪除。

5. AWS CloudFormation stack: 在 AWS CloudFormation 控制台上，選擇您為本教程創建的堆棧。選擇 Actions (動作)、Delete Stack (刪除堆疊)。確認刪除。
6. Amazon S3 儲存貯體：在 Amazon S3 主控台上，選擇您建立用來存放 SageMaker 資產的儲存貯體。選擇刪除。輸入儲存貯體的名稱，然後選擇 Confirm (確認) 以確認刪除。
7. Kinesis 視訊串流：在 Kinesis Video Streams 主控台上，選擇您為應用程式建立的視訊串流。選擇 Delete (刪除)。確認刪除。

# 監控 Amazon Kinesis Video Streams

監控是維持 Amazon Kinesis Video Streams 和 AWS 解決方案的可靠性、可用性和效能的重要組成部分。我們建議您從 AWS 解決方案的所有部分收集監控資料，以協助您在發生多點故障時進行除錯。在開始監控 Amazon Kinesis Video Streams 之前，我們建議您建立一個監控計劃，其中包含下列問題的答案：

- 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？

定義監控目標並建立監控計劃後，下一步就是為環境中的正常 Amazon Kinesis Video Streams 效能建立基準。您應該在不同時間和不同負載條件下測量 Amazon Kinesis Video Streams 效能。監控 Amazon Kinesis Video Streams 時，請存放您收集的監控資料的歷史記錄。您可以將目前的 Amazon Kinesis Video Streams 效能與這些歷史資料進行比較，以協助您識別正常的效能模式和效能異常，並設計解決可能出現的問題的方法。

## 主題

- [使用監控 Amazon Kinesis Video Streams 指標 CloudWatch](#)
- [使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch](#)
- [使用記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail](#)

## 使用監控 Amazon Kinesis Video Streams 指標 CloudWatch

您可以使用 Amazon 監控 Kinesis 影片串流 CloudWatch，Amazon 會從 Amazon Kinesis Video Streams 收集原始資料並將其處理成可讀的近即時指標。這些統計數據會記錄 15 個月，以便您可以訪問歷史信息，並更好地了解 Web 應用程式或服務的性能。

在 [Amazon Kinesis Video Streams 主控台](#) 中，您可以透過兩種方式檢視 Amazon Kinesis 影片串流的 CloudWatch 指標：

- 在 [儀表板] 頁面中，選擇 [目前區域的帳戶層級指標] 區段中的 [視訊串流] 索引標籤。

- 在影片串流詳細資訊頁面中，選擇 Monitoring (監控) 索引標籤。

Amazon Kinesis Video Streams 提供下列指標：


指標	描述
ArchivedFragmentsConsumed.Media	所有 API 使用的片段媒體配額點數目。如需配額點概念的說明，請參閱 <a href="#">the section called “片段中繼資料和片段媒體配額”</a> 。  單位：計數
ArchivedFragmentsConsumed.Metadata	所有 API 使用的片段中繼資料配額點數目。如需配額點概念的說明，請參閱 <a href="#">the section called “片段中繼資料和片段媒體配額”</a> 。  單位：計數
PutMedia.Requests	指定串流的 PutMedia API 要求數目。  單位：計數
PutMedia.IncomingBytes	接收做為串流一部分PutMedia的位元組數。  單位：位元組
PutMedia.IncomingFragments	接收做為串流一部分的PutMedia完整片段數。  單位：計數
PutMedia.IncomingFrames	作為串流一部分接收的PutMedia完整影格數。  單位：計數
PutMedia.ActiveConnections	服務主機的連線總數。  單位：計數
PutMedia.ConnectionErrors	建立串流PutMedia連線時出現錯誤。  單位：計數

指標	描述
PutMedia.FragmentIngestionLatency	Amazon Kinesis Video Streams 接收到片段的第一個和最後一個位元組之間的時間差。  單位：毫秒
PutMedia.FragmentPersistLatency	接收並封存完整片段資料所花費的時間。  單位：計數
PutMedia.Latency	建立連接時的請求和 HTTP 響應之間的 InletService 時間差。  單位：計數
PutMedia.BufferingAckLatency	Amazon Kinesis Video Streams 接收到新片段的第一個位元組與為片段傳送緩衝 ACK 之間的時間差異。  單位：毫秒
PutMedia.ReceivedAckLatency	Amazon Kinesis Video Streams 流接收到新片段的最後一個位元組與針對片段傳送已接收 ACK 之間的時間差異。  單位：毫秒
PutMedia.PersistedAckLatency	Amazon Kinesis Video Streams 接收到新片段的最後一個位元組與為片段傳送持續 ACK 之間的時間差異。  單位：毫秒
PutMedia.ErrorAckCount	執PutMedia行串流時傳送的錯誤 ACK 數目。  單位：計數
PutMedia.Success	1 代表成功寫入的每個片段；0 代表每個失敗的片段。此指標的平均值表示已傳送多少完整的有效片段。  單位：計數


指標	描述
GetMedia.Requests	指定串流的 GetMedia API 要求數目。  單位：計數
GetMedia.OutgoingBytes	指定串流的 GetMedia API 一部分，從服務傳出的位元組總數。  單位：位元組
GetMedia.OutgoingFragments	GetMedia 為串流執行動作時傳送的片段數目。  單位：計數
GetMedia.OutgoingFrames	指定串流期間 GetMedia 傳送的影格數目。  單位：計數
GetMedia.MillisBehindNow	目前伺服器時間戳記與上次傳送片段的伺服器時間戳記之間的時間差異。  單位：毫秒
GetMedia.ConnectionErrors	未成功建立的連線數。  單位：計數
GetMedia.Success	1 代表成功傳送的每個片段；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 1409 1508 1724" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要記錄</a>。</p> </div> 單位：計數

指標	描述
GetMediaForFragmentList.OutgoingBytes	指定串流的 GetMediaForFragmentList API 一部分，從服務傳出的位元組總數。  單位：位元組
GetMediaForFragmentList.OutgoingFragments	作為指定串流之 GetMediaForFragmentList API 一部分，從服務傳送出去的片段總數。  單位：計數
GetMediaForFragmentList.OutgoingFrames	作為指定串流之 GetMediaForFragmentList API 一部分，從服務傳出的畫面總數。  單位：計數
GetMediaForFragmentList.Requests	指定串流的 GetMediaForFragmentList API 要求數目。  單位：計數
GetMediaForFragmentList.Success	1 代表成功傳送的每個片段；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 1188 1507 1499" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> 單位：計數
ListFragments.Latency	ListFragments API 的延遲調用給定的流名稱。  單位：毫秒



指標	描述
ListFragments.Requests	指定串流的 ListFragments API 要求數目。  單位：計數
ListFragments.Success	1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 512 1508 827" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要記錄</a>。</p> </div> 單位：計數
GetHLSStreamingSessionURL.Latency	GetHLSStreamingSessionURL API 的延遲調用給定的流名稱。  單位：毫秒
GetHLSStreamingSessionURL.Requests	指定串流的 GetHLSStreamingSessionURL API 要求數目。  單位：計數

指標	描述
GetHLSStreamingSessionURL.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div data-bbox="751 352 1507 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>
GetHLSMasterPlaylist.Latency	<p>GetHLSMasterPlaylist API 的延遲調用給定的流名稱。</p> <p>單位：毫秒</p>
GetHLSMasterPlaylist.Requests	<p>指定串流的 GetHLSMasterPlaylist API 要求數目。</p> <p>單位：計數</p>
GetHLSMasterPlaylist.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div data-bbox="751 1360 1507 1675" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>

指標	描述
GetHLSMediaPlaylist.Latency	GetHLSMediaPlaylist API 的延遲調用給定的流名稱。  單位：毫秒
GetHLSMediaPlaylist.Requests	指定串流的 GetHLSMediaPlaylist API 要求數目。  單位：計數
GetHLSMediaPlaylist.Success	1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 768 1507 1083" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要記錄</a>。</p> </div> 單位：計數
GetMP4InitFragment.Latency	GetMP4InitFragment API 的延遲調用給定的流名稱。  單位：毫秒
GetMP4InitFragment.Requests	指定串流的 GetMP4InitFragment API 要求數目。  單位：計數

指標	描述
GetMP4InitFragment.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div data-bbox="748 352 1507 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>
GetMP4MediaFragment.Latency	<p>GetMP4MediaFragment API 的延遲調用給定的流名稱。</p> <p>單位：毫秒</p>
GetMP4MediaFragment.Requests	<p>指定串流的 GetMP4MediaFragment API 要求數目。</p> <p>單位：計數</p>
GetMP4MediaFragment.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div data-bbox="748 1360 1507 1675" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>

指標	描述
GetMP4MediaFragment.OutgoingBytes	指定串流的 GetMP4MediaFragment API 一部分，從服務傳出的位元組總數。  單位：位元組
GetTSFragment.Latency	GetTSFragment API 的延遲調用給定的流名稱。  單位：毫秒
GetTSFragment.Requests	指定串流的 GetTSFragment API 要求數目。  單位：計數
GetTSFragment.Success	1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 884 1507 1192" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要記錄</a>。</p> </div> 單位：計數
GetTSFragment.OutgoingBytes	指定串流的 GetTSFragment API 一部分，從服務傳出的位元組總數。  單位：位元組
GetDASHStreamingSessionURL.Latency	GetDASHStreamingSessionURL API 的延遲調用給定的流名稱。  單位：毫秒

指標	描述
GetDASHStreamingSessionURL. Requests	指定串流的 GetDASHStreamingSessionURL API 要求數目。  單位：計數
GetDASHStreamingSessionURL. Success	1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。  <div data-bbox="748 562 1507 873"><p> <b>Note</b></p><p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要記錄</a>。</p></div> 單位：計數
GetDASHManifest.Latency	GetDASHManifest API 的延遲調用給定的流名稱。  單位：毫秒
GetDASHManifest.Requests	指定串流的 GetDASHManifest API 要求數目。  單位：計數

指標	描述
GetDASHManifest.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>
GetClip.Latency	<p>GetClip API 呼叫指定視訊串流名稱的延遲。</p> <p>單位：毫秒</p>
GetClip.Requests	<p>指定視訊串流的 GetClip API 要求數目。</p> <p>單位：計數</p>
GetClip.Success	<p>1 代表每個成功的請求；0 代表每個失敗。平均值表示成功率。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用要求與回應摘要 (包括 AWS 要求 ID) 的詳細資訊，<a href="#">請參閱要求/回應摘要</a>記錄。</p> </div> <p>單位：計數</p>

指標	描述
GetClip.OutgoingBytes	作為指定視訊串流之 GetClip API 一部分，從服務傳出的位元組總數。  單位：位元組

## CloudWatch 指標指引

CloudWatch 指標可協助尋找下列問題的答案：

### 主題

- [資料是否會到達 Amazon Kinesis Video Streams 服務？](#)
- [為什麼 Amazon Kinesis Video Streams 服務無法成功擷取資料？](#)
- [為什麼無法以與從生產者傳送的相同速率從 Amazon Kinesis Video Streams 服務讀取資料？](#)
- [為什麼主控台沒有影片，或為何影片播放會發生延遲？](#)
- [讀取即時資料時延遲的原因為何，以及為何用戶端落後於串流前頭之後？](#)
- [用戶端是否從 Kinesis 視訊串流讀取資料，以及以何種速度讀取資料？](#)
- [為什麼用戶端無法從 Kinesis 視訊串流中讀取資料？](#)

## 資料是否會到達 Amazon Kinesis Video Streams 服務？

相關指標：

- PutMedia.IncomingBytes
- PutMedia.IncomingFragments
- PutMedia.IncomingFrames

動作項目：

- 如果這些指標有所下降，請檢查您的應用程式是否仍在向服務傳送資料。
- 檢查網路頻寬。如果您的網路頻寬不足，可能會降低服務接收資料的速率。



## 為什麼 Amazon Kinesis Video Streams 服務無法成功擷取資料？

相關指標：

- PutMedia.Requests
- PutMedia.ConnectionErrors
- PutMedia.Success
- PutMedia.ErrorAckCount

動作項目：

- 如果增加PutMedia.ConnectionErrors，請查看生產者用戶端收到的 HTTP 回應和錯誤碼，以查看建立連線時發生的錯誤。
- 如果有下降PutMedia.Success或增加PutMedia.ErrorAckCount，請查看服務發送的 ack 響應中的 ack 錯誤代碼，以了解為什麼數據導入失敗。如需詳細資訊，請參閱 [AckErrorCode.Values](#)。

## 為什麼無法以與從生產者傳送的相同速率從 Amazon Kinesis Video Streams 服務讀取資料？

相關指標：

- PutMedia.FragmentIngestionLatency
- PutMedia.IncomingBytes

動作項目：

- 如果這些指標有所下降，請檢查連線的網路頻寬。低頻寬連線可能導致資料以較低的速率傳送至服務。

## 為什麼主控台沒有影片，或為何影片播放會發生延遲？

相關指標：

- PutMedia.FragmentIngestionLatency
- PutMedia.FragmentPersistLatency
- PutMedia.Success

- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

動作項目：

- 如果增加 `PutMedia.FragmentIngestionLatency` 或下降 `PutMedia.IncomingFragments`，請檢查網路頻寬以及資料是否仍在傳送中。
- 如果有下降 `PutMedia.Success`，請檢查 `ack` 錯誤代碼。如需詳細資訊，請參閱 [AckErrorCode.Values](#)。
- 如果 `PutMedia.FragmentPersistLatency` 或增加 `ListFragments.Latency`，您很可能遇到服務問題。如果情況持續存在一段時間，請與您的客戶服務聯絡人聯絡，瞭解您的服務是否有問題。

讀取即時資料時延遲的原因為何，以及為何用戶端落後於串流前頭之後？

相關指標：

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

動作項目：

- 如果增加 `GetMedia.ConnectionErrors`，則由於頻繁嘗試重新連接到流，消費者可能會在讀取流時落後。查看 `GetMedia` 請求傳回的 HTTP 回應/錯誤代碼。
- 如果下降 `GetMedia.Success`，則可能是由於服務無法將數據發送給消費者，這將導致連接中斷，並從消費者重新連接，這將導致消費者滯後於流的頭部。
- 如果有增加 `GetMedia.MillisBehindNow`，請查看您的頻寬限制，看看您是否因為頻寬較低而以較慢的速率接收資料。

用戶端是否從 Kinesis 視訊串流讀取資料，以及以何種速度讀取資料？

相關指標：

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`

- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

動作項目：

- 這些指標會指出讀取即時和封存資料的速率。

為什麼用戶端無法從 Kinesis 視訊串流中讀取資料？

相關指標：

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

動作項目：

- 如果有增加 `GetMedia.ConnectionErrors`，請查看 `GetMedia` 請求返回的 HTTP 響應和錯誤代碼。如需詳細資訊，請參閱 [AckErrorCode.Values](#)。
- 如果您嘗試讀取最新或即時資料，請檢 `PutMedia.IncomingBytes` 查串流中是否有資料可供服務傳送給消費者。
- 如果 `GetMedia.Success` 或下降 `GetMediaForFragmentList.Success`，則可能是由於服務無法將數據發送給消費者。如果情況持續存在一段時間，請與您的客戶服務聯絡人聯絡，瞭解您的服務是否有問題。

## 使用監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch

您可以使用 Amazon 監控 Amazon Kinesis Video Streams 邊緣代理程式 CloudWatch，該代理程式會收集原始資料並將其處理為可讀且接近即時的指標。這些統計數字為期 15 個月。有了這些歷史資

訊，您就可以更好地瞭解 Web 應用程式或 Amazon Kinesis Video Streams 邊緣代理程式服務的執行情況。

若要檢視測量結果，請執行下列動作：

1. 請登入 AWS Management Console 並開啟 CloudWatch 主控台，網址為 <https://console.aws.amazon.com/cloudwatch/>。
2. 在左側導覽中的「量度」下，選取「所有量度」。
3. 選擇瀏覽索引標籤，然後選取EdgeRuntimeAgent自訂命名空間。

Amazon Kinesis Video Streams 邊緣代理程式會在命名空間EdgeRuntimeAgent下發佈下列指標：

維度	州	描述
串流名稱，RecordJob	執行中	<p>在執行時持RecordJob 續發佈。</p> <p>單位：無。只要處於此狀態，RecordJob 就會發佈「1」。</p>
	FatalError	<p>如果發生RecordJob 致命錯誤，則發佈。</p> <p>單位：無。發生此事件時，「1」會發佈一次。</p> <div style="border: 1px solid #007bff; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note 如需其他資訊，請參閱記錄。</p> </div>
	已完成	<p>完成時RecordJob 發佈。</p> <p>單位：無。發生此事件時，「1」會發佈一次。</p>
串流名稱，UploadJob	執行中	<p>在執行時持UploadJob 續發佈。</p> <p>單位：無。只要處於此狀態，UploadJob 就會發佈「1」。</p>
	FatalError	<p>如果發布UploadJob 致命錯誤。</p> <p>單位：無。發生此事件時，「1」會發佈一次。</p>

維度	州	描述
		<p> <b>Note</b></p> <p>如需其他資訊，請參閱記錄。</p>
	已完成	<p>完成時UploadJob 發佈。</p> <p>單位：無。發生此事件時，「1」會發佈一次。</p>
串流名稱	PercentageSpaceUsed	<p>這是在 Amazon Kinesis Video Streams 邊緣代理程式組態中用於錄製媒體的總空間中使用的百分比。如需詳細資訊，請參閱<a href="#">the section called "LocalSizeConfig"</a>。</p> <p>單位：百分比 ( 比例 0-1 ) 。</p>
實物名稱	活著	<p>每分鐘從 Amazon Kinesis Video Streams 邊緣代理程式發佈一次，不論其上執行的任何組態為何。</p> <p>這可用來瞭解 Amazon Kinesis Video Streams 邊緣代理程式是否有效且準備好接受組態。</p> <p>單位：無。「1」每分鐘發佈一次。</p>
	RecordJobs.HealthyJobCount	<p>Amazon Kinesis Video Streams 邊緣代理程式上執行中和排程記錄任務的總數。</p> <p>單位：計數。</p>
	UploadJobs.HealthyJobCount	<p>Amazon Kinesis Video Streams 邊緣代理程式上執行中和排程上傳任務的總計數。</p> <p>單位：計數。</p>
	RecordJobs.UnhealthyJobCount	<p>目前錯誤記錄工作的總數。</p> <p>單位：計數。</p>

維度	州	描述
	UploadJobs.UnhealthyJobCount	目前錯誤的上傳工作總數。 單位：計數。
	RecordJobs.RunningJobCount	主動執行記錄工作的總數。 單位：計數。
	UploadJobs.RunningJobCount	主動執行中上傳工作的總數。 單位：計數。
	RecordJobs.EdgeConfigCount	Amazon Kinesis Video Streams 邊緣代理程式上處理中的記錄組態總數。 單位：計數。
	UploadJobs.EdgeConfigCount	Amazon Kinesis Video Streams 邊緣代理程式上處理中的上傳組態總數。 單位：計數。

## CloudWatch Amazon Kinesis Video Streams 邊緣代理程式的指標指引

CloudWatch 指標對於尋找下列問題的答案很有用：

### 主題

- [Amazon Kinesis Video Streams 邊緣代理程式是否有足夠的空間來記錄？](#)
- [Amazon Kinesis Video Streams 邊緣代理程式是否還活著？](#)
- [有沒有不健康的工作？](#)
- [是否有任何工作需要外部干預？](#)

### Amazon Kinesis Video Streams 邊緣代理程式是否有足夠的空間來記錄？

相關指標：PercentageSpaceUsed

動作：不需要採取任何動作。

## Amazon Kinesis Video Streams 邊緣代理程式是否還活著？

相關指標：Alive

動作：如果您在任何時候停止接收此指標，表示 Amazon Kinesis Video Streams 邊緣代理程式遇到下列一或多個問題：

- 應用程式執行階段問題：記憶體或其他資源限制、錯誤等
- 代理 AWS IoT 程式在關機、當機或終止時執行的裝置
- 裝 AWS IoT 置沒有網路連線

## 有沒有不健康的工作？

相關指標：

- RecordJobs.UnhealthyJobCount
- UploadJobs.UnhealthyJobCount

動作：檢查記錄檔並尋找指FatalError標。

- 如果指FatalError標存在，則發生嚴重錯誤，您需要手動重新啟動工作。在使StartEdgeConfigurationUpdate用手動重新啟動工作之前，請檢查記錄檔並修正問題。
- 如果FatalError指標不存在，就會發生暫時性 (非嚴重) 錯誤，而 Amazon Kinesis Video Streams 邊緣代理程式正在重試工作。

### Note

若要讓代理程式重新嘗試嚴重錯誤的工作，請使用。[the section called “StartEdgeConfigurationUpdate”](#)

## 是否有任何工作需要外部干預？

相關指標：

- `PercentageSpaceUsed`— 如果超過特定值，則記錄工作會暫停，並僅在有空間可用時 (當媒體不保留時) 繼續。您可以傳送較高版本的更新組態，`MaxLocalMediaSizeInMB` 以立即更新工作。
- `RecordJob.FatalError/UploadJob.FatalError`— 調查代理程式的記錄檔，然後再次傳送設定以繼續工作。

動作：使用配置進行 API 調用，以重新啟動遇到此問題的作業。

## 使用記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail

Amazon Kinesis Video Streams 可與服務搭配使用者 AWS CloudTrail、角色或 Amazon Kinesis Video Streams AWS 服務中所採取的動作記錄。CloudTrail 以事件形式擷取 Amazon Kinesis Video Streams 的所有 API 呼叫。擷取的呼叫包括來自 Amazon Kinesis Video Streams 主控台的呼叫，以及對 Amazon Kinesis Video Streams API 操作進行程式碼呼叫的程式碼呼叫。如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon S3 儲存貯體，包括 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，您可以判斷向 Amazon Kinesis Video Streams 提出的請求、提出請求的來源 IP 地址、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 和 CloudTrail

CloudTrail 在您創建 AWS 帳戶時，您的帳戶已啟用。當 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 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](#)
- [設定的 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](#)



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

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

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：

- 該請求是否使用根或 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臨時安全憑證
- 該請求是否由另一項 AWS 服務服務提出。

如需詳細資訊，請參閱[CloudTrail 使用 userIdentity 元素](#)。

## 範例：Amazon Kinesis Video Streams 日誌檔項目

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。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。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[CreateStream](#)動作的 CloudTrail 記錄項目。

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
```

```

        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2018-05-25T00:16:31Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "CreateStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
        "streamName": "VideoStream",
        "dataRetentionInHours": 2,
        "mediaType": "mediaType",
        "kmsKeyId": "arn:aws:kms::us-east-1:123456789012:alias",
        "deviceName": "my-device"
    },
    "responseElements": {
        "streamARN": "arn:aws:kinesisvideo:us-east-1:123456789012:stream/VideoStream/12345"
    },
    "requestID": "db6c59f8-c757-11e3-bc3b-57923b443c1c",
    "eventID": "b7acfc0-6ca9-4ee1-a3d7-c4e8d420d99b"
    },
    {
        "eventVersion": "1.05",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
        "eventTime": "2018-05-25:17:06Z",
        "eventSource": "kinesisvideo.amazonaws.com",
        "eventName": "DeleteStream",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
        "requestParameters": {
            "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/VideoStream/12345",
            "currentVersion": "keqrjeqkj9"
        }
    }
}

```

```

    },
    "responseElements": null,
    "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
    "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:02Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "DescribeStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream"
    },
    "responseElements": null,
    "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:03Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "GetDataEndpoint",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",

```

```

    "requestParameters": {
      "streamName": "VideoStream",
      "apiName": "LIST_FRAGMENTS"
    },
    "responseElements": null,
    "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2018-05-25T00:16:56Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "ListStreams",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "maxResults": 100,
      "streamNameCondition": {"comparisonValue": "MyVideoStream"
comparisonOperator": "BEGINS_WITH"}}
    },
    "responseElements": null,
    "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
    "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
  }
]
}

```

# Kinesis Video Streams 服務配額

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]。


## 控制平面 API 服務配額

下節說明控制平面 API 的服務配額。TPS 代表每秒交易。

達到帳戶層級或資源層級要求限制時，就會擲回 `a. ClientLimitExceededException`

### 控制平面 API 服務配額

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
<a href="#">the section called “CreateStream”</a>	50 TPS [s]	在美國東部 (維吉尼亞北部) 和美國西部 (奧勒岡) 區域中，每個帳戶有 10000 個串流。在所有其他支援的區域中，每個帳戶有 5000 個串流。		裝置、CLI、SDK 導向的存取和主控台，全都可以呼叫此 API。如果串流不存在，只會有一個 API 呼叫成功。

 **Note**  
此限制可以提

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
		<p>高到每個帳戶 100,000 個 (或更多) 串流。登入位 AWS Management Console 於 <a href="https://console.aws.amazon.com/">https://console.aws.amazon.com/</a> 並提交 Kinesis Video Streams <a href="#">提高服務限制案例</a>，以要求提高此限制。</p>		

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
<a href="#">the section called “DeleteEdgeConfiguration”</a>	10 TPS [h]	N/A	1 TPS [h]	
<a href="#">the section called “DeleteStream”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “DescribeEdgeConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “DescribeImageGenerationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “DescribeMappedResourceConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
<a href="#">the section called “DescribeNotificationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “DescribeStream”</a>	300 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “GetDataEndpoint”</a>	300 TPS [h]	N/A	5 TPS [h]	每 45 分鐘進行呼叫，以為大部分的 PutMedia/GetMedia 使用案例重新整理串流字符。若應用程式在故障時重新載入快取資料端點，快取資料端點仍是安全的。
<a href="#">the section called “ListEdgeAgentConfigurations”</a>	50 TPS [h]	N/A	N/A	
<a href="#">the section called “ListStreams”</a>	50 TPS [h]	N/A		
<a href="#">the section called “ListTagsForStream”</a>	50 TPS [h]	N/A	5 TPS [h]	



API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
<a href="#">the section called “StartEdgeConfigurationUpdate”</a>	10 TPS [h]	N/A	1 TPS [h]	
<a href="#">the section called “TagStream”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “UntagStream”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “UpdateDataRetention”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “UpdateImageGenerationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	
<a href="#">the section called “UpdateNotificationConfiguration”</a>	50 TPS [h]	N/A	5 TPS [h]	

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外情況及備註
<a href="#">the section called “UpdateStream”</a>	50 TPS [h]	N/A	5 TPS [h]	

## 媒體和封存媒體 API 服務配額

下節說明媒體和封存媒體 API 的服務配額。

達到帳戶層級或資源層級要求限制時，就會擲回 `ClientLimitExceededException`

當達到連接層級限制時，就會擲出 `ConnectionLimitExceededException`。

當達到片段層級的限制時，就會擲出以下錯誤或 `ack`：

- 傳回的片段 `MIN_FRAGMENT_DURATION_REACHED ack` 低於持續時間下限。
- 傳回的片段 `MAX_FRAGMENT_DURATION_REACHED ack` 高於持續時間上限。
- 傳回的片段 `MAX_FRAGMENT_SIZE ack` 高於資料大小上限。
- 如果 `FragmentLimitExceeded` 操作中達到片段限制，就會擲出 `GetMediaForFragmentList` 例外。

### 資料平面 API 服務配額

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外情況及備註
<a href="#">the section called “PutMedia”</a>	5 TPS [h]	1 (s)	每秒 12.5 MB，或每個串流 100 兆比特	<ul style="list-style-type: none"> <li>• 片段持續時間下限：1 秒 (h)</li> <li>• 最大片段持續時間：20 秒 [h]</li> </ul>	典型的 <code>PutMedia</code> 請求包含數秒鐘的資料，使每個串流產生較低的 TPS。如果有多個並行連線超出配額，則會接受最後一個連線。

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外情況及備註
				<ul style="list-style-type: none"> <li>• 片段大小 上限：50 MB (h)</li> <li>• 軌道數目 上限：3 [s]</li> <li>• 每秒傳送的最大片段數：5 [h]</li> <li>• 片段中繼資料上限 上限：10 個標籤 [h]</li> </ul>	
<a href="#">the section called “GetClip”</a>	N/A	N/A	100 MB 的大小限制 [h]	片段數量上限：200 [h]	
<a href="#">the section called “GetDASHStreamingSessionURL”</a>	25 租售價 [h]	N/A	N/A	N/A	
<a href="#">the section called “GetHLSStreamingSessionURL”</a>	25 租售價 [h]	N/A	N/A	N/A	

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外情況及備註
<a href="#">the section called “GetImages”</a>	N/A	N/A	100 MB [小時]	N/A	<p>每個請求的最大影像數量為 100 [h]。</p> <div data-bbox="1133 401 1507 810" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>的最小值SamplingInterval 為 200 毫秒 (ms)，也就是每秒 5 個影像。</p> </div>
<a href="#">the section called “GetMedia”</a>	5 TPS [h]	3 (s)	25 MB/秒或 200 Mbps (s)	每秒傳送的最大片段數：6 [h]	<p>一個獨特的消費客戶端不應該需要超過兩個或三個 TPS，因為建立連接後，應用程式應該連續讀取。</p> <p>如果一般片段大約為 5 MB，則此限制表示每個 Kinesis 視訊串流約 75 Mbps。此等串流會有串流傳入位元速率 2 倍的傳出位元速率。</p> <div data-bbox="1133 1388 1507 1654" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>GetMedia不會用於 HLS/ 破折號播放。</p> </div>

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外情況及備註
<a href="#">the section called "GetMediaForFragmentList"</a>	N/A	5 (s)	25 MB/秒或 200 Mbps (s)	片段數量上限：1000 [h]	五個以片段為基礎的消費應用程式可以同時叫用。GetMediaForFragmentList 多出的連接都會遭到拒絕。

## 視訊播放通訊協定 API 服務配額

API	工作階段層級限制	片段層級限制
<a href="#">獲取破折號 ManifestPlaylist</a>	5 TPS [h]	每個播放清單的片段數目上限：5000 [h]
<a href="#">GETHLS MasterPlaylist</a>	5 TPS [h]	N/A
<a href="#">GETHLS MediaPlaylist</a>	5 TPS [h]	每個播放清單的片段數目上限：5000 [h]
<a href="#">獲取 MP4 InitFragment</a>	5 TPS [h]	N/A
<a href="#">獲取 MP4 MediaFragment</a>	20 租置標準 [h]	N/A
<a href="#">GetTSFragment</a>	20 租置標準 [h]	N/A

## 片段中繼資料和片段媒體配額

Kinesis Video Streams [API 存取封存媒體](#) 會根據要求的片段數量而非 API 呼叫次數進行限制。API 受片段中繼資料數量和要求的片段媒體數量的速率限制。片段中繼資料和片段媒體配額會針對每個串流套用。換句話說，一個串流中繼資料或媒體的要求不適用於另一個串流的配額。但是，在給定的流中，每個配額都會跨多個 API 共享。這意味著，對於給定的流，跨不同 API 的片段請求消耗相同的配額。當超出串流的片段中繼資料或片段媒體配額時，API 會傳回 `ClientLimitExceededException`。下表顯示 API 如何從兩種類型的配額中使用。對於這些表中的第二列，假設如果流的配額為 N，這意味著 API 有 N 個點可以從該流的配額類型中消耗。GetClipAPI 會出現在兩個表格中。

## 片段元數據配額消耗

API	每個請求使用的配額積分	共用配額 (N)
<a href="#">the section called “ListFragments”</a>	MaxResults 參數值	每秒 10,000 個配額點數，每個串流 [h]
<a href="#">the section called “GetClip”</a>	產生的剪輯片段中的片段數	
<a href="#">GetHLSMediaPlaylist</a>	MaxMediaPlaylistFragmentResults 參數值	
<a href="#">GetDASHManifest</a>	MaxManifestFragmentResults 參數值	
<a href="#">the section called “GetImages”</a>	要求的最大圖像數量 400 +	

## 片段媒體配額消耗

API	每個請求使用的配額積分	共用配額 (N)
<a href="#">the section called “GetMediaForFragmentList”</a>	片段參數中的片段數	每個串流每秒 500 個配額點 [h]
<a href="#">the section called “GetClip”</a>	產生的剪輯片段中的片段數	
<a href="#">GetMP4MediaFragment</a>	1	
<a href="#">GetTSFragment</a>	1	
<a href="#">the section called “GetImages”</a>	要求的最大影像數	

例如，配額為每秒 500 個片段媒體，則支援特定串流的下列呼叫模式：

- 每秒 5 個請求，每個剪輯中GetClip有 100 個片段。
- 每秒 100 個請求，每個剪輯中GetClip有 5 個片段。
- 每秒 2 個請求，每個剪輯中GetClip有 100 個片段，每個剪輯每秒 3 個請求。GetMediaForFragmentList
- 每秒 400 個請求GetMP4MediaFragment，每秒 100 個請求GetTSFragment。

這些配額對於每個串流可支援的 HLS 和 MPEG-DASH 工作階段數目具有重要意義。媒體播放器在給定時間可以使用的 HLS 和 DASH 會話數量沒有限制。因此，重要的是，播放應用程式不允許同時使用太多工作階段。下列兩個範例說明如何判斷可支援的同時播放工作階段數目：

### 範例 1：即時串流

在具有 1 秒持續時間片段、音訊和視訊軌道且 `MaxMediaPlaylistFragmentResults` 設定為 5 的 HLS 的即時串流案例中，媒體播放程式通常會 `GetHLSMediaPlaylist` 每秒撥打兩次呼叫。一個呼叫用於最新的視訊中繼資料，另一個呼叫用於對應的音訊中繼資料。這兩個呼叫會消耗五個片段中繼資料配額點。它還 `GetMP4MediaFragment` 每秒撥打兩次電話：一個呼叫最新視頻，另一個呼叫相應的音頻。每個調用都會消耗一個片段媒體令牌，因此總共消耗了兩個令牌。

在這個案例中，最多可支援 250 個同時播放工作階段。在 250 個工作階段中，此案例每秒耗用 2,500 個片段中繼資料配額點 (遠低於 10,000 個配額) 和每秒 500 個片段媒體配額點。

### 範例 2：隨選播放

在過去事件的隨選播放案例中使用 MPEG-DASH (音訊和視訊軌道且 `MaxManifestFragmentResults` 設為 1,000)，媒體播放程式通常會在工作階段開始時呼叫 `GetDASHManifest` 一次 (消耗 1,000 個片段中繼資料配額點)，並以每秒 5 次的速率呼叫 `GetMP4MediaFragment` (消耗 5 個片段媒體配額點)，直到載入所有片段為止。在這個案例中，每秒最多可啟動 10 個新工作階段 (右邊是每秒 10,000 個片段中繼資料配額)，而最多 100 個工作階段可以以每秒 5 次的速率主動載入片段媒體 (就在每秒 500 個片段媒體配額上)。

您可以分別

使 `ArchivedFragmentsConsumed.Media` 用 `ArchivedFragmentsConsumed.Metadata` 和監控片段中繼資料和片段媒體配額點的使用情況。如需監視的相關資訊，請參閱 [監控](#)。

## 片段元數據配額

下列服務配額適用於將片段中繼資料新增至 Kinesis 視訊串流中的片段：

- 您可以在一個片段之前最多新增 10 個中繼資料項目。
- 片段中繼資料名稱長度上限為 128 位元組。
- 片段中繼資料數值長度上限為 256 位元組。
- 片段中繼資料名稱不能以字串 "AWS" 開頭。若新增上述中繼資料項目，PIC 中的 `putFragmentMetadata` 方法將回報 `STATUS_INVALID_METADATA_NAME` 錯誤 (錯誤代碼 0x52000077)。您的應用程式可以忽略該錯誤 (PIC 將不會新增中繼資料項目) 或回應錯誤。

## 串流標籤

這些中繼資料索引鍵值配對會套用至整體 Kinesis Video Streams 資源，而非 Kinesis 視訊串流中包含的個別片段。

每個 Kinesis 視訊串流最多支援 50 個標籤。

如[the section called “TagStream”](#)需串流標籤的索引鍵和值的限制，請參閱。



# Kinesis Video Streams 疑難排解

使用下列資訊對 Amazon Kinesis Video Streams 遇到的常見問題進行疑難排解。

## 主題

- [疑難排解一般問題](#)
- [疑難排解 API 問題](#)
- [疑難排解 HLS 問題](#)
- [Java 問題疑難排解](#)
- [生產者程式庫問題疑](#)
- [疑難排解串流剖析程式庫](#)

## 疑難排解一般問題

本節說明使用 Kinesis Video Streams 時可能會遇到的一般問題。

### 問題

- [延遲過高](#)

## 延遲過高

延遲可能是因為傳送至 Kinesis 影片串流服務的片段持續時間所造成。減少生產者和服務之間延遲的方法之一是，設定媒體管道以產生較短的片段持續期間。

若要減少每個片段中傳送的影格數，請在中減少下列值kinesis\_video\_gstreamer\_sample\_app.cpp：

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512, NULL);
```

### Note

因為影片轉譯的內部實作，在 Mozilla Firefox 瀏覽器的延遲很嚴重。

## 疑難排解 API 問題

本節說明使用 Kinesis Video Streams 時可能遇到的 API 問題。

### 問題

- [錯誤：「未知的選項」](#)
- [錯誤：「無法判斷要授權的服務/操作名稱」](#)
- [錯誤：「無法將影格放入串流」](#)
- [錯誤：「在收到最終 AckEvent 服務之前已關閉連接」](#)
- [錯誤：「STATUS\\_STORE\\_OUT\\_OF\\_MEMORY」](#)

### 錯誤：「未知的選項」

GetMedia 和 GetMediaForFragmentList 可能會因為下列錯誤而失敗：

```
Unknown options: <filename>.mkv
```

如果您將 AWS CLI output 類型設定為 `json`，就會發生此錯誤。AWS CLI 使用預設輸出類型 `(none)` 重新配置。若要取得有關規劃的資訊 AWS CLI，請參閱 [《AWS CLI 指令參考》](#) 中的規劃。

### 錯誤：「無法判斷要授權的服務/操作名稱」

GetMedia 可能會因為下列錯誤而失敗：

```
Unable to determine service/operation name to be authorized
```

若未正確指定端點，可能會發生此錯誤。取得端點時，請務必在 `GetDataEndpoint` 呼叫中包含下列參數，具體取決於要呼叫的 API：

```
--api-name GET_MEDIA  
--api-name PUT_MEDIA  
--api-name GET_MEDIA_FOR_FRAGMENT_LIST  
--api-name LIST_FRAGMENTS
```

### 錯誤：「無法將影格放入串流」

PutMedia 可能會因為下列錯誤而失敗：

```
Failed to put a frame in the stream
```

若未將連線或權限提供給服務，可能會發生此錯誤。在中執行下列命令 AWS CLI，並確認可擷取串流資訊：

```
aws kinesishvideo describe-stream --stream-name StreamName --endpoint https://  
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

如果呼叫失敗，請參閱[疑難排解 AWS CLI 錯誤](#)以取得詳細資訊。

**錯誤：**「在收到最終 AckEvent 服務之前已關閉連接」

PutMedia 可能會因為下列錯誤而失敗：

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was  
received
```

如果 PushbackInputStream 未正確實作，可能會發生此錯誤。驗證unread()方法是否正確實施。

**錯誤：**「STATUS\_STORE\_OUT\_OF\_MEMORY」

PutMedia 可能會因為下列錯誤而失敗：

```
The content store is out of memory.
```

當內容存放區未分配足夠的大小時，將會發生此錯誤。若要提高內容存放區的大小，請提高StorageInfo.storageSize 的值。如需詳細資訊，請參閱 [StorageInfo](#)。

## 疑難排解 HLS 問題

如果您的視訊串流無法正確播放，請參閱[the section called “疑難排解 HLS 問題”](#)。

## Java 問題疑難排解

本節說明如何疑難排解使用 Kinesis Video Streams 時遇到的常見 Java 問題。

### 問題

- [啟用 Java 日誌](#)

## 啟用 Java 日誌

若要疑難排解 Java 範例和程式庫的問題，啟用和檢查偵錯記錄檔會很有幫助。若要啟用偵錯日誌，請依下列步驟執行：

1. 將 log4j 新增至 pom.xml 檔案，在 dependencies 節點中：

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. 在 target/classes 目錄中，建立名為 log4j.properties 的檔案，內含下列內容：

```
# Root logger option
log4j.rootLogger=DEBUG, stdout

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:
%L - %m%n

log4j.logger.org.apache.http.wire=DEBUG
```

偵錯日誌將會列印到 IDE 主控台。

## 生產者程式庫問題疑

本節說明使用 [Producer 資料庫](#) 時可能遇到的問題。

### 問題

- [無法編譯製作者開發套件](#)
- [影片串流不會顯示在主控台](#)
- [錯誤：使用 GStreamer 示範應用程式串流資料時「包含在請求中的安全字符無效」](#)

- [錯誤：「無法將影格提交到 Kinesis Video 用戶端」](#)
- [GStreamer 應用程式停止並在 OS X 上出現「串流停止，原因未交涉」訊息](#)
- [在 Raspberry Pi 上的 GStreamer 示範中建立 Kinesis 影片用戶端時出現錯誤：「無法配置堆積」](#)
- [在 Raspberry Pi 上執行 GStreamer 示範時出現錯誤：「非法指示」](#)
- [攝影機無法載入 Raspberry Pi](#)
- [在 macOS High Sierra 上找不到攝影機](#)
- [在 macOS High Sierra 上編譯時找不到 jni.h 檔案](#)
- [執行 GStreamer 示範應用程式時 Curl 錯誤](#)
- [在 Raspberry Pi 執行時間的時間戳記/範圍聲明](#)
- [在 Raspberry Pi 上的 gst\\_value\\_set\\_fraction\\_range\\_full 聲明](#)
- [在 Android 上發生 STATUS\\_MKV\\_INVALID\\_ANNEXB\\_NALU\\_IN\\_FRAME\\_DATA \(0x320000d\) 錯誤](#)
- [到達片段持續時間上限錯誤](#)
- [使用 IoT 授權時的「傳遞的物件名稱無效」錯誤](#)

## 無法編譯製作者開發套件

確認您的路徑中有所需的程式庫。若要確認，請使用下列命令：

```
$ env | grep LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/local/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/
kinesis-video-native-build/downloads/local/lib
```

## 影片串流不會顯示在主控台

若要在主控台顯示您的視訊串流，必須使用 AvCC 格式中的 H.264 編碼。如果您的串流不會顯示，請確認以下資訊：

- 如果原始串流為 Annex-B 格式，您的 [NAL 適應性旗標](#) 已設定為 NAL\_ADAPTATION\_ANNEXB\_NALS | NAL\_ADAPTATION\_ANNEXB\_CPD\_NALS。這是 StreamDefinition 建構函數中的預設值。
- 您提供正確的轉碼器私有資料。對於 H.264，這是序列參數集 (SPS) 和圖片參數集 (PPS)。根據您的媒體資料來源，此資料可能會從媒體來源單獨擷取或編碼到框架。

許多基礎串流為以下格式，其中的 Ab 是 Annex-B 起始碼 (001 或 0001)：

```
Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)
```

CPD (編解碼器私人數據)，如果 H.264 作為 SPS 和 PPS 在流中，則可以適應 AVCC 格式。除非媒體管道單獨提供 CPD，否則應用程序可以通過查找第一個 IDR 框架 (應包含 SPS 和 PPS) 從框架中提取 CPD，提取兩個 NALU (將是 Ab ( Sps ) Ab ( Ps ) ) 並將其設置在 CPD 中。StreamDefinition

## 錯誤：使用 GStreamer 示範應用程式串流資料時「包含在請求中的安全字符無效」

如果發生此錯誤，表示您的登入資料有問題。請確認下列內容：

- 如果您使用臨時的登入資料，必須指定工作階段字符。
- 確認您的臨時登入資料並未過期。
- 確認您已設定適當的權利。
- 在 macOS 上，確認您沒有在鑰匙圈中快取登入資料。

## 錯誤：「無法將影格提交到 Kinesis Video 用戶端」

如果發生此錯誤，表示來源串流中的時間戳記未正確設定。請嘗試以下做法：

- 使用最新的軟體開發套件範例，其中可能有更新可修正您的問題。
- 將高質量流設置為更高的比特率，並在相機支持的情況下修復源流中的任何抖動。

## GStreamer 應用程式停止並在 OS X 上出現「串流停止，原因未交涉」訊息

串流可能會在 OS X 上停止並出現以下訊息：

```
Debugging information: gstbasesrc.c(2939): void gst_base_src_loop(GstPad *) (): /
GstPipeline:test-pipeline/GstAutoVideoSrc:source/GstAVFVideoSrc:source-actual-src-
avfvide:
streaming stopped, reason not-negotiated (-4)
```

可能的解決方法是從gst\_caps\_new\_simple呼叫中移除影格速率參數kinesis\_video\_gstreamer\_sample\_app.cpp：

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

在 Raspberry Pi 上的 GStreamer 示範中建立 Kinesis 影片用戶端時出現錯誤：「無法配置堆積」

GStreamer 範例應用程式嘗試配置 512 MB 的 RAM，但您的系統可能無法提供。您可以降低以下 KinesisVideoProducer.cpp 中的值以降低此配置：

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```

在 Raspberry Pi 上執行 GStreamer 示範時出現錯誤：「非法指示」

如果您在執行 GStreamer 示範時遇到下列錯誤，請確認您已針對正確的裝置版本編譯應用程式。(例如，確認你不編譯樹莓派 3 當你在樹莓派上運行 2。)

```
INFO - Initializing curl.
Illegal instruction
```

## 攝影機無法載入 Raspberry Pi

若要檢查是否已載入攝影機，請執行下列動作：

```
$ ls /dev/video*
```

若未發現任何結果，請執行下列動作：

```
$ vccgcmd get_camera
```

輸出格式應類似以下內容：

```
supported=1 detected=1
```

如果驅動程式未偵測到攝影機，請執行下列動作：

1. 檢查實體攝影機設定並確認正確連接。
2. 執行以下動作以升級韌體：

```
$ sudo rpi-update
```

3. 重新啟動裝置。
4. 執行以下動作以載入驅動程式：

```
$ sudo modprobe bcm2835-v4l2
```

5. 確認已偵測到攝影機：

```
$ ls /dev/video*
```

## 在 macOS High Sierra 上找不到攝影機

在 macOS High Sierra 上，示範應用程式找不到攝影機 (如果有多個攝影機)。

## 在 macOS High Sierra 上編譯時找不到 jni.h 檔案

若要解決這個錯誤，請將您安裝的 Xcode 更新到最新版本。

## 執行 GStreamer 示範應用程式時 Curl 錯誤

若要解決當您執行 GStreamer 示範應用程式時 curl 錯誤，請將 [此憑證檔案](#) 複製至 `/etc/ssl/cert.pem`。

## 在 Raspberry Pi 執行時間的時間戳記/範圍聲明

如果在執行時間發生時間戳記聲明，請更新韌體並重新啟動裝置：

```
$ sudo rpi-update  
$ sudo reboot
```



## 在 Raspberry Pi 上的 `gst_value_set_fraction_range_full` 聲明

如果 `uv4l` 服務正在執行，將會顯示以下聲明：

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
denominator_end) < 0' failed
```

如果發生這種情況，請停止 `uv4l` 服務並重新啟動應用程式。

## 在 Android 上發生

### `STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d)` 錯誤

如果媒體串流的 [NAL 適應性旗標](#) 不正確，則會顯示以下錯誤：

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

如果發生此錯誤，請提供媒體的正確 `.withNalAdaptationFlags` 旗標 (例如，`NAL_ADAPTATION_ANNEXB_CPD_NALS`)。在以下 [Android Producer Library](#) 一行中提供此旗標：

<https://github.com/aws-labs/aws-sdk-android-samples/tree/master/src/main/java/com/amazonaws/mediastreamingdemo/AmazonKinesisVideoDemoApp.java#L169> `StreamConfigurationFragment`

## 到達片段持續時間上限錯誤

此錯誤會在串流中的媒體片段超過片段持續時間上限時發生。請參閱本 [the section called “媒體和封存媒體 API 服務配額”](#) 節中的最大片段持續時間限制。

請嘗試下列方法解決此問題：

- 如果您使用的是網路攝影機/USB 攝影機，請執行以下其中一項：
  - 如果您使用的是基於關鍵幀的碎片，請將編碼器設置為在 10 秒內提供關鍵幀。
  - 如果您沒有使用以索引鍵影格為基礎的片段，則在中定義串流時 [步驟 2：撰寫並檢查程式碼](#)，請將片段持續時間上限設定為小於 10 秒的值。
  - 如果您在 GStreamer 管道中使用軟件編碼器 (例如 `x264`)，則可以在 10 秒內將 `key-int-max` 屬性設置為一個值。例如，設定 `key-int-max` 為 60，且 `fps` 設定為 30，則每 2 秒啟用一次關鍵影格。

- 如果您使用的是 RPI 攝影機，請將關鍵畫面間隔屬性設定為小於 10 秒。
- 如果您使用的是 IP ( RTSP ) 相機，請將 GOP 大小設置為 60。

## 使用 IoT 授權時的「傳遞的物件名稱無效」錯誤

若要在使用 IoT 認證進行授權時避免此錯誤 (HTTP Error 403: Response: {"message": "Invalid thing name passed"}), 請確定的值 stream-name (kvssink 元素的必要參數) 與的值相同 iot-thingname。如需詳細資訊，請參閱 [元素參數參數](#)。

## 疑難排解串流剖析程式庫

本節說明使用 [串流剖析器程式庫](#) 時可能遇到的問題。

### 問題

- [無法從串流存取單一影格](#)
- [片段解碼錯誤](#)

## 無法從串流存取單一影格

若要從消費者應用程式中的串流來源存取單一畫面，請確認您的串流包含正確的轉碼器私人資料。如需有關串流資料格式的詳細資訊，請參閱 [資料模型](#)。

要了解如何使用編解碼器私人數據訪問框架，請參閱 GitHub 網站上的以下測試文件：[KinesisVideoRendererExampleTest.java](#)

## 片段解碼錯誤

如果您的片段未以瀏覽器支援的 H.264 格式和層級正確地編碼，當您在主控台播放串流時，可能會看到以下錯誤：

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264 content
```

如果發生這種情況，請檢查下列各項：

- 影格的解析度符合轉碼器私有資料中指定的解析度。

- 編碼影格的 H.264 設定檔和層級符合轉碼器私有資料中指定的設定檔和層級。
- 瀏覽器支援設定檔/層級組合。大多數目前的瀏覽器皆支援所有設定檔和層級組合。
- 時間戳記準確並且順序正確，並且未建立重複的時間戳記。
- 您的應用程式使用 H.264 格式編碼影格資料。

# Amazon Kinesis Video Streams 的文件歷史記錄

下表說明自上次發行 Amazon Kinesis 影片串流以來對文件所做的重要變更。

- 最新的 API 版本：2017-11-29
- 最新文件更新：2023 年 6 月 27 日

變更	描述	日期
亞馬遜 Kinesis 視訊串流邊緣代理程式邊緣到雲端連線	新功能版本。如需詳細資訊，請參閱 <a href="#">邊緣代理</a> 。	2023年6月27日
開始使用：將資料傳送至 Kinesis 視訊串流	將媒體資料從攝影機傳送至 Kinesis 視訊串流的基本教學課程。如需詳細資訊，請參閱 <a href="#">將資料傳送至 Amazon Kinesis 影片串流</a> 。	2019 年 1 月 21 日
程式庫範本用於整合 SageMaker	Kinesis 影片串流的範例應用程式，用SageMaker於識別某些物件何時出現在 Amazon Kinesis 影片串流中。如需詳細資訊，請參閱 <a href="#">SageMaker</a> 。	2018 年 11 月 19 日
串流中繼資料	您可以使用製作者 SDK 將中繼資料嵌入 Kinesis 視訊串流中。如需詳細資訊，請參閱 <a href="#">搭配 Kinesis 視訊串流使用串流中繼資料</a> 。	2018 年 9 月 28 日
C++ 製作者開發套件記錄	您可以為 C++ 製作者開發套件應用程式設定記錄。如需詳細資訊，請參閱 <a href="#">使用記錄與 C++ 製作者 SDK</a> 。	2018 年 7 月 18 日
HLS 視訊串流	您現在可以使用 HTTP 即時串流來檢視 Kinesis 視訊串流。	2018 年 7 月 13 日

變更	描述	日期
	如需詳細資訊，請參閱 <a href="#">Kinesis Video Streams 播放</a> 。	
使用 RTSP 來源進行串流	Kinesis 影片串流的範例應用程式，該應用程式會在 Docker 容器中執行，並從 RTSP 來源串流視訊。如需詳細資訊，請參閱 <a href="#">RTSP 和 Docker</a> 。	2018 年 6 月 20 日
C++ 製作者開發套件 GStreamer 外掛程式	顯示如何建立 <a href="#">C++ Producer Library</a> 做為 GStreamer 目的地。如需詳細資訊，請參閱 <a href="#">GStreamer</a> 。	2018 年 6 月 15 日
製作者開發套件回呼參考文件	用於回呼的參考文件 <a href="#">室壁運動視頻流生產者庫</a> 。如需詳細資訊，請參閱 <a href="#">生產者 SDK 回呼</a> 。	2018 年 6 月 12 日
系統要求	有關生產者裝置和軟體開發套件之記憶體和儲存需求的文件。如需詳細資訊，請參閱 <a href="#">Kinesis Video Streams 系統需求</a> 。	2018 年 5 月 30 日
CloudTrail 支援	用於監控 API CloudTrail 使用情況的文件。如需詳細資訊，請參閱 <a href="#">使用記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail</a> 。	2018 年 5 月 24 日
製作者開發套件結構參考文件	<a href="#">室壁運動視頻流生產者庫</a> 使用之結構的參考文件。如需詳細資訊，請參閱 <a href="#">製作者 SDK 結構</a> 及 <a href="#">Kinesis 視頻流結構</a> 。	2018 年 5 月 7 日

變更	描述	日期
轉譯器範例文件	轉譯器範例應用程式的文件，其中顯示如何解碼和顯示 Kinesis 視訊串流的影格。如需詳細資訊，請參閱 <a href="#">範例：剖析和呈現 Kinesis Video Streams 片段</a> 。	2018 年 3 月 15 日
製作者開發套件限制參考文件	有關在 <a href="#">C++ Producer Library</a> 中操作限制之資訊。如需詳細資訊，請參閱 <a href="#">製作者 SDK 限制</a> 。	2018 年 3 月 13 日
監控	有關使用 Amazon CloudWatch 和 監控 Kinesis 影片串流指標和 API 呼叫的資訊。AWS CloudTrail如需詳細資訊，請參閱 <a href="#">監控 Amazon Kinesis Video Streams</a> 。	2018 年 2 月 5 日
網路抽象層 (NAL) 適應性旗標參考	有關在使用串流視訊時設定 NAL 適應性旗標的資訊。如需詳細資訊，請參閱 <a href="#">NAL 適應性旗標</a> 。	2018 年 1 月 15 日
Android 支援串流視訊	Kinesis 視訊串流現在支援從 Android 裝置串流影片。如需詳細資訊，請參閱 <a href="#">Android Producer Library</a> 。	2018 年 1 月 12 日
室壁運動影片範例文件	Kinesis Video 範例應用程式的文件，其中顯示如何在應用程式 <a href="#">Kinesis 視頻流解析器庫</a> 中使用。如需詳細資訊，請參閱 <a href="#">KinesisVideoExample</a> 。	2018 年 1 月 9 日

變更	描述	日期
室壁運動影片串流文件已發佈	這是 Amazon Kinesis Video Streams 開發人員指南的最初版本。	2017 年 11 月 29 日

# API 參考

此節點下的區段包含 API 參考文件。使用左窗格中的目錄移至不同的 API 參考區段。

## 動作

Amazon Kinesis 影片串流支援下列動作：

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)
- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)



- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)
- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)

Amazon Kinesis 影片串流媒體支援下列動作：

- [GetMedia](#)
- [PutMedia](#)

Amazon Kinesis 影片串流封存媒體支援下列動作：

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

Amazon Kinesis 視訊訊號通道支援下列動作：

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

Amazon 室運動影片 WebRTC 儲存支援下列動作：

- [JoinStorageSession](#)

## Amazon Kinesis Video Streams

Amazon Kinesis 影片串流支援下列動作：

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)
- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)
- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)



## CreateSignalingChannel

服務: Amazon Kinesis Video Streams

創建一個信令通道。

CreateSignalingChannel 是一種非同步操作。

請求語法

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelName": "string",
  "ChannelType": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI 請求參數

請求不會使用任何 URI 參數。

請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelName

您正在建立的訊號通道的名稱。其對於每個 AWS 帳戶 和 AWS 區域 必須是唯一的。

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

### ChannelType

您正在建立的訊號通道的類型。目前唯一支援的通道類型為 SINGLE\_MASTER。

類型：字串

有效值: SINGLE\_MASTER | FULL\_MESH

必要：否

### SingleMasterConfiguration

包含SINGLE\_MASTER通道類型配置的結構。

類型：[SingleMasterConfiguration](#) 物件

必要：否

### Tags

您要與此通道關聯的一組標籤（鍵值對）。

類型：[Tag](#) 物件陣列

陣列成員：項目數下限為 0。項目數上限為 50。

必要：否

### 回應語法

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

{
  "ChannelARN": "string"
}
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

## ChannelARN

Amazon 資源名稱 ( ARN ) 創建的渠道。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

### AccountChannelLimitExceededException

您已達到此區域的有效信號通道AWS 帳戶上限。

HTTP 狀態碼：400

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel 用來判斷資源狀態的 DescribeStream 或 API。

HTTP 狀態碼：400

#### TagsPerResourceExceededLimitException

您已超過可與資源關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## CreateStream

服務: Amazon Kinesis Video Streams

建立新的 Kinesis 視訊串流。

當您建立新串流時，Kinesis Video Streams 會為其指派版本編號。當您變更串流的中繼資料時，Kinesis Video Streams 會更新版本。

CreateStream 是一種非同步操作。

如需有關服務運作方式的資訊，請參閱[運作方式](#)。

您必須具備 KinesisVideo:CreateStream 動作的許可。

### 請求語法

```
POST /createStream HTTP/1.1
Content-type: application/json

{
  "DataRetentionInHours": number,
  "DeviceName": "string",
  "KmsKeyId": "string",
  "MediaType": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### [DataRetentionInHours](#)

您要在串流中保留資料的小時數。Kinesis Video Streams 會將資料保留在與串流相關聯的資料存放區中。



預設值為 0，表示串流不會保留資料。

當DataRetentionInHours值為 0 時，取用者仍然可以使用服務主機緩衝區中保留的片段，其保留時間限制為 5 分鐘，保留記憶體限制為 200 MB。達到任一限制時，片段將從緩衝區中刪除。

類型：整數

有效範圍：最小值為 0。

必要：否

### DeviceName

寫入串流的裝置名稱。

#### Note

在目前的實作中，Kinesis Video Streams 不會使用此名稱。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：[a-zA-Z0-9\_.-]+

必要：否

### KmsKeyId

您希望 Kinesis 影片串流用來加密串流資料的 AWS Key Management Service (AWS KMS) 金鑰識別碼。

如果未指定金鑰識別碼，則會使用預設的 Kinesis 視訊管理金鑰 (AWS/kinesisvideo)。

如需詳細資訊，請參閱[DescribeKey](#)。

類型：字串

長度限制：長度下限為 1。長度上限為 2048。

模式：.+

必要：否

## MediaType

串流的媒體類型。串流的取用者可以在處理串流時使用此資訊。如需媒體類型的詳細資訊，請參閱[媒體類型](#)。如果您選擇指定 MediaType，請參閱[命名需求](#)以取得指導方針。

有效值示例包括「視頻/h264」和「視頻/h264，音頻/AAC」。

此參數是選擇性的；預設值為 null (或 JSON 中為空白)。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：`[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

必要：否

## StreamName

您要建立之串流的名稱。

串流名稱是串流的識別碼，每個帳戶和區域都必須是唯一的。

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

## Tags

要與指定資料流相關聯的標籤清單。每個標籤都是一個鍵值對 ( 值是可選的 )。

類型：字串到字串映射

地圖項目：最多 50 個項目。

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^[{\p{L}}{\p{Z}}{\p{N}}_./=+\-@]*$`

值長度限制：最小長度為 0。長度上限為 256。

值模式：`[{\p{L}}{\p{Z}}{\p{N}}_./=+\-@]*`

必要：否

## 回應語法

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

{
  "StreamARN": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

## StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccountStreamLimitExceededException

為帳戶建立的串流數量太高。

HTTP 狀態碼：400

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

DeviceStreamLimitExceededException

不實作。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

InvalidDeviceException

不實作。

HTTP 狀態碼：400

ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

TagsPerResourceExceededLimitException

您已超過可與資源關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DeleteEdgeConfiguration

服務: Amazon Kinesis Video Streams

一種非同步 API，可刪除串流的現有邊緣設定，以及來自 Edge 代理程式的對應媒體。

當您呼叫此 API 時，同步狀態會設定為 DELETING。刪除程序會啟動，其中作用中邊緣工作會停止，並從 Edge 裝置刪除所有媒體。刪除的時間會根據儲存的媒體總量而有所不同。如果刪除程序失敗，同步狀態會變更為 DELETE\_FAILED。您將需要重新嘗試刪除。

當刪除程序順利完成時，將無法再存取 Edge 組態。

### Note

AWS 非洲 (開普敦) 地區南部 -1 不提供此 API。

### 請求語法

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### StreamARN

串流的 Amazon Resource Name (ARN)。指定 StreamName 或 StreamARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

要從中刪除 Edge 配置的流的名稱。指定StreamName或StreamARN。

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

### 回應語法

```
HTTP/1.1 200
```

### 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

#### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

#### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

#### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

### StreamEdgeConfigurationNotFoundException

當 Amazon Kinesis 視訊串流找不到您指定的串流邊緣組態時，會呈現例外狀況。

HTTP 狀態碼：404

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## DeleteSignalingChannel

服務: Amazon Kinesis Video Streams

刪除指定的信令通道。DeleteSignalingChannel是非同步作業。如果您未指定頻道的目前版本，則會刪除最新版本。

### 請求語法

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelARN

您要刪除的信令通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：是

### CurrentVersion

您要刪除的信令通道的當前版本。您可以透過叫用DescribeSignalingChannel或ListSignalingChannels API 作業來取得目前的版本。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：`[a-zA-Z0-9]+`

必要：否

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。

2. 用於確定給定流映射到的信道的 `DescribeMappedResourceConfiguration` API。
3. `DescribeSignalingChannel` 用來判斷資源狀態的 `DescribeStream` 或 API。

HTTP 狀態碼：400

#### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

#### VersionMismatchException

您指定的串流版本不是最新版本。若要取得最新版本，請使用 [DescribeStream](#) API。

HTTP 狀態碼：400

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DeleteStream

服務: Amazon Kinesis Video Streams

刪除 Kinesis 視訊串流和串流中包含的資料。

此方法標記要刪除的流，並使流中的數據立即無法訪問。

若要確保您在刪除串流之前擁有最新版本的串流，您可以指定串流版本。Kinesis 視訊串流會為每個串流指派一個版本。當您更新串流時，Kinesis Video Streams 會指派新的版本號碼。若要取得最新的串流版本，請使用 DescribeStream API。

此操作需要 KinesisVideo:DeleteStream 動作的權限。

請求語法

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "StreamARN": "string"
}
```

URI 請求參數

請求不會使用任何 URI 參數。

請求主體

請求接受採用 JSON 格式的下列資料。

### CurrentVersion

選用性：您要刪除的串流版本。

將版本指定為保護措施，以確保刪除正確的串流。若要取得串流版本，請使用 DescribeStream API。

如果未指定，則在刪除流之前僅檢查 CreationTime

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：`[a-zA-Z0-9]+`

必要：否

### StreamARN

您要刪除的資料流的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：是

### 回應語法

```
HTTP/1.1 200
```

### 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

## ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## VersionMismatchException

您指定的串流版本不是最新版本。若要取得最新版本，請使用 [DescribeStreamAPI](#)。

HTTP 狀態碼：400

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)

- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeEdgeConfiguration

服務: Amazon Kinesis Video Streams

描述使用 `StartEdgeConfigurationUpdate` API 設定的串流邊緣設定，以及 Edge 代理程式的記錄器和上傳程式工作的最新狀態。使用此 API 取得組態的狀態，以判斷組態是否與 Edge 代理程式同步。使用此 API 來評估 Edge 代理程式的健全狀況。

### Note

AWS 非洲 (開普敦) 地區南部 -1 不提供此 API。

### 請求語法

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### StreamARN

串流的 Amazon Resource Name (ARN)。指定 `StreamName` 或 `StreamARN`。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`



必要：否

## StreamName

要更新其邊緣配置的流的名稱。指定StreamName或StreamARN。

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

## 回應語法

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

{
  "CreationTime": number,
  "EdgeAgentStatus": {
    "LastRecorderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "RecorderStatus": "string"
    },
    "LastUploaderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "UploaderStatus": "string"
    }
  },
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    }
  }
}
```

```
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      },
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    }
  },
  "FailedStatusDetails": "string",
  "LastUpdatedTime": number,
  "StreamARN": "string",
  "StreamName": "string",
  "SyncStatus": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### CreationTime

第一次建立串流邊緣設定的時間戳記。

類型：Timestamp

### EdgeAgentStatus

包含 Edge 代理程式的記錄器和上傳程式工作之最新狀態詳細資料的物件。使用此資訊可判斷 Edge 代理程式目前的健全狀況。

類型：[EdgeAgentStatus](#) 物件

## [EdgeConfig](#)

將用來與邊緣代理程式 IoT Greengrass 元件同步處理之串流邊緣設定的說明。Edge 代理程式元件將在您的內部部署的 IoT 中樞裝置設定上執行。

類型：[EdgeConfig](#) 物件

## [FailedStatusDetails](#)

產生失敗狀態的描述。

類型：字串

## [LastUpdatedTime](#)

上次更新串流邊緣設定的時間戳記。

類型：Timestamp

## [StreamARN](#)

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

## [StreamName](#)

邊緣組態已更新來源的資料流名稱。

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_.-]+`

## [SyncStatus](#)

邊緣組態更新的最新狀態。

類型：字串

有效值: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

### StreamEdgeConfigurationNotFoundExcpetion

當 Amazon Kinesis 視訊串流找不到您指定的串流邊緣組態時，會呈現例外狀況。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeImageGenerationConfiguration

服務: Amazon Kinesis Video Streams

取得指ImageGenerationConfiguration定 Kinesis 視訊串流的。

### 請求語法

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### StreamARN

Kinesis 影片串流的 Amazon 資源名稱 (ARN) , 可從中擷取映像產生組態。您必須指定StreamName或StreamARN。

類型 : 字串

長度限制 : 長度下限為 1。長度上限為 1024。

模式 : arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要 : 否

#### StreamName

要從中擷取映像產生組態的串流名稱。您必須指定StreamName或StreamARN。

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

## 回應語法

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

{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [ImageGenerationConfiguration](#)

包含 Kinesis 視訊串流 (KVS) 影像傳送所需資訊的結構。如果此結構為 null，則配置將從流中刪除。

類型：[ImageGenerationConfiguration](#) 物件

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)



- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeMappedResourceConfiguration

服務: Amazon Kinesis Video Streams

返回有關流的最新信息。streamName或streamARN應在輸入中提供。

### 請求語法

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### MaxResults

要在回應中傳回的結果數目上限。

類型：整數

有效範圍：固定值為 1。

必要：否

#### NextToken

在下一個請求中提供的令牌，以獲得另一批結果。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

必要：否

### StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

串流的名稱。

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

### 回應語法

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

{
  "MappedResourceConfigurationList": [
    {
      "ARN": "string",
      "Type": "string"
    }
  ],
  "NextToken": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [MappedResourceConfigurationList](#)

封裝或包含媒體儲存裝置組態屬性的結構。

類型：[MappedResourceConfigurationListItem](#) 物件陣列

陣列成員：項目數下限為 0。項目數上限為 1。

### [NextToken](#)

NextToken 要求中用來擷取下一組結果的權杖。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeMediaStorageConfiguration

服務: Amazon Kinesis Video Streams

返回有關通道的最新信息。在輸入ChannelARN中指定ChannelName或。

### 請求語法

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelARN

Amazon 資源名稱 ( ARN ) 的通道。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### ChannelName

頻道名稱。

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

## 回應語法

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

{
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### MediaStorageConfiguration

封裝或包含媒體儲存裝置組態屬性的結構。

類型：[MediaStorageConfiguration](#) 物件

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## DescribeNotificationConfiguration

服務: Amazon Kinesis Video Streams

取得指定 NotificationConfiguration 的 Kinesis 視訊串流的。

### 請求語法

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### [StreamARN](#)

Kinesis 影片串流的 Amazon 資源名稱 (ARN)，您要從中擷取通知組態的位置。您必須指定 `StreamName` 或串流網路。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

#### [StreamName](#)

要從中擷取通知組態的資料流名稱。您必須指定 `StreamName` 或 `StreamARN`。

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

## 回應語法

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

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [NotificationConfiguration](#)

包含通知所需資訊的結構。如果結構為 null，配置將從流中刪除。

類型：[NotificationConfiguration](#) 物件

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeSignalingChannel

服務: Amazon Kinesis Video Streams

返回有關信令通道的最新信息。您必須指定要描述之通道的名稱或 Amazon 資源名稱 (ARN)。

### 請求語法

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### ChannelARN

您要描述的訊號通道的 ARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

#### ChannelName

您要描述的信令通道的名稱。

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

## 回應語法

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

{
  "ChannelInfo": {
    "ChannelARN": "string",
    "ChannelName": "string",
    "ChannelStatus": "string",
    "ChannelType": "string",
    "CreationTime": number,
    "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
    },
    "Version": "string"
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### ChannelInfo

封裝指定信令通道的中繼資料和屬性的結構。

類型：[ChannelInfo](#) 物件

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

#### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

#### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

#### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DescribeStream

服務: Amazon Kinesis Video Streams

返回有關指定流的最新信息。您必須指定StreamName或StreamARN。

### 請求語法

```
POST /describeStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

串流的名稱。

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

## 回應語法

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

{
  "StreamInfo": {
    "CreationTime": number,
    "DataRetentionInHours": number,
    "DeviceName": "string",
    "KmsKeyId": "string",
    "MediaType": "string",
    "Status": "string",
    "StreamARN": "string",
    "StreamName": "string",
    "Version": "string"
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### StreamInfo

描述串流的物件。

類型：[StreamInfo](#) 物件

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。



HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## GetDataEndpoint

服務: Amazon Kinesis Video Streams

獲取用於讀取或寫入指定流的端點。在應用程式中使用此端點從指定的串流讀取 (使用或作GetMediaForFragmentList業) 或寫入 (使用PutMedia作業)。GetMedia

### Note

返回的端點沒有附加 API 名稱。客戶端需要將 API 名稱添加到返回的端點。

在要求中，透過StreamName或指定串流StreamARN。

### 請求語法

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
  "APIName": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### APIName

要取得端點之 API 動作的名稱。

類型：字串

有效值: PUT\_MEDIA | GET\_MEDIA | LIST\_FRAGMENTS |  
GET\_MEDIA\_FOR\_FRAGMENT\_LIST | GET\_HLS\_STREAMING\_SESSION\_URL |  
GET\_DASH\_STREAMING\_SESSION\_URL | GET\_CLIP | GET\_IMAGES

必要：是

### StreamARN

您想要取得端點的資料流的 Amazon 資源名稱 (ARN)。您必須在請求StreamName中指定此參數或。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

您想要取得端點的資料流名稱。您必須在請求StreamARN中指定此參數或。

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

### 回應語法

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

{
  "DataEndpoint": "string"
}
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

## [DataEndpoint](#)

端點值。若要從串流讀取資料或將資料寫入資料，請在應用程式中指定此端點。

類型：字串

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## GetSignalingChannelEndpoint

服務: Amazon Kinesis Video Streams

為指定的信令通道提供用於傳送和接收訊息的端點。此 API 使用 `SingleMasterChannelEndpointConfiguration` 輸入參數，該參數由 `Protocols` 和 `Role` 屬性組成。

`Protocols` 用於確定通信機制。例如，如果您指定 WSS 為通訊協定，則此 API 會產生安全的 Websocket 端點。如果您指定 HTTPS 為通訊協定，則此 API 會產生一個 HTTPS 端點。

`Role` 決定訊息傳送權限。MASTER 角色導致此 API 生成一個端點，客戶可以使用該端點與渠道上的任何觀眾進行通信。VIEWER 角色導致此 API 生成一個端點，客戶端可用於僅與 MASTER。

### 請求語法

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "SingleMasterChannelEndpointConfiguration": {
    "Protocols": [ "string" ],
    "Role": "string"
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### ChannelARN

您要取得端點之信號通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：是

### [SingleMasterChannelEndpointConfiguration](#)

包含SINGLE\_MASTER通道類型的端點配置的結構。

類型：[SingleMasterChannelEndpointConfiguration](#) 物件

必要：否

### 回應語法

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

{
  "ResourceEndpointList": [
    {
      "Protocol": "string",
      "ResourceEndpoint": "string"
    }
  ]
}
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [ResourceEndpointList](#)

指定信令通道的端點清單。

類型：[ResourceEndpointListItem](#) 物件陣列

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)



- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ListEdgeAgentConfigurations

服務: Amazon Kinesis Video Streams

傳回與指定 Edge 代理程式相關聯的邊緣組態陣列。

在要求中，您必須指定 Edge 代理程式 HubDeviceArn。

### Note

AWS 非洲 (開普敦) 地區南部 -1 不提供此 API。

### 請求語法

```
POST /listEdgeAgentConfigurations HTTP/1.1
```

```
Content-type: application/json
```

```
{  
  "HubDeviceArn": "string",  
  "MaxResults": number,  
  "NextToken": "string"  
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### HubDeviceArn

邊緣代理程式的「物聯網 (IoT) 物聯網」 Arn。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

必要：是

## MaxResults

要在回應中傳回的邊緣組態數目上限。預設值為 5。

類型：整數

有效範圍：最小值為 1。最大值為 10。

必要：否

## NextToken

如果您指定此參數，當截斷ListEdgeAgentConfigurations作業的結果時，呼叫會在回應NextToken中傳回。要獲取另一批 Edge 配置，請在下一個請求中提供此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：[a-zA-Z0-9+/=]\*

必要：否

## 回應語法

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

{
  "EdgeConfigs": [
    {
      "CreationTime": number,
      "EdgeConfig": {
        "DeletionConfig": {
          "DeleteAfterUpload": boolean,
          "EdgeRetentionInHours": number,
          "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
          }
        },
        "HubDeviceArn": "string",
        "RecorderConfig": {
```

```
    "MediaSourceConfig": {
      "MediaUriSecretArn": "string",
      "MediaUriType": "string"
    },
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "FailedStatusDetails": "string",
  "LastUpdatedTime": number,
  "StreamARN": "string",
  "StreamName": "string",
  "SyncStatus": "string"
}
],
"NextToken": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [EdgeConfigs](#)

單一資料流邊緣設定的描述。

類型：[ListEdgeAgentConfigurationsEdgeConfig](#) 物件陣列

### [NextToken](#)

如果響應被截斷，調用返回帶有給定令牌的這個元素。要獲取下一批 Edge 配置，請在下一個請求中使用此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## ListSignalingChannels

服務: Amazon Kinesis Video Streams

返回對ChannelInfo象的數組。每個對象描述一個信令通道。若只要擷取滿足特定條件的通道，您可以指定ChannelNameCondition。

請求語法

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
  "ChannelNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

URI 請求參數

請求不會使用任何 URI 參數。

請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelNameCondition

選用性：僅傳回滿足特定條件的通道。

類型：[ChannelNameCondition](#) 物件

必要：否

### MaxResults

要在回應中傳回的通道數目上限。預設值為 500。

類型：整數

有效範圍：最小值為 1。最大值為 10000。

必要：否

## [NextToken](#)

如果您指定此參數，當截斷ListSignalingChannels作業的結果時，呼叫會在回應NextToken中傳回。要獲取另一批渠道，請在下一個請求中提供此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

必要：否

## 回應語法

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

{
  "ChannelInfoList": [
    {
      "ChannelARN": "string",
      "ChannelName": "string",
      "ChannelStatus": "string",
      "ChannelType": "string",
      "CreationTime": number,
      "SingleMasterConfiguration": {
        "MessageTtlSeconds": number
      },
      "Version": "string"
    }
  ],
  "NextToken": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。



## ChannelInfoList

ChannelInfo 物件的陣列。

類型：[ChannelInfo](#) 物件陣列

## NextToken

如果響應被截斷，調用返回帶有令牌的這個元素。要獲取下一批流，請在下一個請求中使用此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ListStreams

服務: Amazon Kinesis Video Streams

返回對StreamInfo象的數組。每個對象描述了一個流。若只要擷取滿足特定條件的串流，您可以指定StreamNameCondition。

### 請求語法

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string",
  "StreamNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### MaxResults

要在回應中傳回的串流數目上限。預設值為 10,000。

類型：整數

有效範圍：最小值為 1。最大值為 10000。

必要：否

#### NextToken

如果您指定此參數，當截斷ListStreams作業的結果時，呼叫會在回應NextToken中傳回。要獲取另一批流，請在下一個請求中提供此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

必要：否

### StreamNameCondition

選用性：僅傳回滿足特定條件的串流。目前，您只能指定串流名稱的前置詞做為條件。

類型：[StreamNameCondition](#) 物件

必要：否

### 回應語法

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

{
  "NextToken": "string",
  "StreamInfoList": [
    {
      "CreationTime": number,
      "DataRetentionInHours": number,
      "DeviceName": "string",
      "KmsKeyId": "string",
      "MediaType": "string",
      "Status": "string",
      "StreamARN": "string",
      "StreamName": "string",
      "Version": "string"
    }
  ]
}
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

## [NextToken](#)

如果響應被截斷，調用返回帶有令牌的這個元素。要獲取下一批流，請在下一個請求中使用此令牌。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

## [StreamInfoList](#)

StreamInfo 物件的陣列。

類型：[StreamInfo](#) 物件陣列

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ListTagsForResource

服務: Amazon Kinesis Video Streams

返回與指定信令信道相關聯的標籤列表。

### 請求語法

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
  "NextToken": "string",
  "ResourceARN": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### [NextToken](#)

如果您指定此參數且 ListTagsForResource 呼叫的結果會被截斷，則回應會包含一個 Token，您可以在下一個要求中使用這個 Token 來擷取下一批標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：[a-zA-Z0-9+/=]\*

必要：否

#### [ResourceARN](#)

您要列出標籤的信令通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：是

## 回應語法

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [NextToken](#)

如果您指定此參數且 `ListTagsForResource` 呼叫的結果會被截斷，則回應會包含一個 Token，您可以在下一個要求中使用這個 Token 來擷取下一組標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

### [Tags](#)

與指定信令通道相關聯的標籤鍵和值的映射。

類型：字串到字串映射

地圖項目：最多 50 個項目。

索引鍵長度限制：長度下限為 1。長度上限為 128。



金鑰模式：`^([\p{L}\p{Z}\p{N}_./=+\-@]*)$`

值長度限制：最小長度為 0。長度上限為 256。

值模式：`[\p{L}\p{Z}\p{N}_./=+\-@]*`

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ListTagsForStream

服務: Amazon Kinesis Video Streams

返回與指定流相關聯的標籤列表。

在請求中，您必須指定StreamName或StreamARN。

### 請求語法

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### [NextToken](#)

如果您指定此參數且ListTagsForStream呼叫的結果會被截斷，則回應會包含一個Token，您可以在下一個要求中使用這個Token來擷取下一批標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：[a-zA-Z0-9+/=]\*

必要：否

### [StreamARN](#)

您要列出標籤的流的 Amazon 資源名稱 ( ARN )。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

您要列出標籤的流的名稱。

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

### 回應語法

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

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### NextToken

如果您指定此參數且 `ListTags` 呼叫的結果會被截斷，則回應會包含一個 Token，您可以在下一個要求中使用這個 Token 來擷取下一組標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 512。

模式：`[a-zA-Z0-9+/=]*`

## Tags

與指定流相關聯的標籤鍵和值的映射。

類型：字串到字串映射

地圖項目：最多 50 個項目。

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式：`^( [\p{L}\p{Z}\p{N}_.:/=+\-@]* )$`

值長度限制：最小長度為 0。長度上限為 256。

值模式：`[ \p{L}\p{Z}\p{N}_.:/=+\-@]*`

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### InvalidResourceFormatException

的格式StreamARN無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## StartEdgeConfigurationUpdate

服務: Amazon Kinesis Video Streams

可更新串流現有邊緣設定的非同步 API。Kinesis 視訊串流會將串流的邊緣設定與在 IoT 中樞裝置上執行的邊緣代理程式 IoT Greengrass 元件同步，並在您的內部部署進行設定。同步的時間可能會有所不同，並取決於集線器裝置的連線能力。SyncStatus 將在確認邊緣組態並與 Edge 代理程式同步時更新。

如果第一次叫用此 API，則會為串流建立新的 Edge 設定，並將同步狀態設定為 SYNCING。在再次使用此 API 之前 IN\_SYNC，您必須等待同步狀態達到終端狀態 SYNC\_FAILED，例如：`IN_SYNC`，或者 `SYNC_FAILED`。如果您在同步過程中調用此 API，ResourceInUseException 將拋出。串流邊緣設定和 Edge 代理程式的連線會重試 15 分鐘。15 分鐘後，狀態將轉換為 SYNC\_FAILED 狀態。

若要將邊緣組態從一個裝置移至另一個裝置，請使用 [DeleteEdgeConfiguration](#) 來刪除目前的邊緣組態。然後，您可以使用更新 StartEdgeConfigurationUpdate 的集線器裝置 ARN 呼叫。

### Note

AWS 非洲 (開普敦) 地區南部 -1 不提供此 API。

### 請求語法

```
POST /startEdgeConfigurationUpdate HTTP/1.1
Content-type: application/json

{
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      }
    }
  }
}
```

```
    },
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  }
},
"StreamARN": "string",
"StreamName": "string"
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

## 請求主體

請求接受採用 JSON 格式的下列資料。

## EdgeConfig

呼叫更新程序所需的 Edge 組態詳細資訊。

類型：[EdgeConfig](#) 物件

必要：是

## StreamARN

串流的 Amazon Resource Name (ARN)。指定StreamName或StreamARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否



## StreamName

要更新其邊緣配置的流的名稱。指定StreamName或StreamARN。

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

## 回應語法

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

{
  "CreationTime": number,
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      },
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    }
  }
}
```

```
    }  
  }  
},  
"FailedStatusDetails": "string",  
"LastUpdatedTime": number,  
"StreamARN": "string",  
"StreamName": "string",  
"SyncStatus": "string"  
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [CreationTime](#)

第一次建立串流邊緣設定的時間戳記。

類型：Timestamp

### [EdgeConfig](#)

將用來與邊緣代理程式 IoT Greengrass 元件同步處理之串流邊緣設定的說明。Edge 代理程式元件將在您的內部部署的 IoT 中樞裝置設定上執行。

類型：[EdgeConfig](#) 物件

### [FailedStatusDetails](#)

產生失敗狀態的描述。

類型：字串

### [LastUpdatedTime](#)

上次更新串流邊緣設定的時間戳記。

類型：Timestamp

### [StreamARN](#)

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

### StreamName

邊緣組態已更新來源的資料流名稱。

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_.-]+`

### SyncStatus

串流邊緣設定的目前同步狀態。當您調用此 API 時，同步狀態將設置為狀 SYNCING 態。使用 DescribeEdgeConfiguration API 取得邊緣設定的最新狀態。

類型：字串

有效值: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱 [常見錯誤](#)。

#### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

#### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

#### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NoDataRetentionException

以小時為單位的串流資料保留等於零。

HTTP 狀態碼：400

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者提供的輸入StreamARN或不處ChannelARN於作用中狀態時，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## TagResource

服務: Amazon Kinesis Video Streams

將一個或多個標籤添加到信令通道。標籤是您可以定義並指派給AWS資源的索引鍵值配對 (值為選用)。如果您指定已存在的標籤，則標籤值會取代為您在請求中指定的值。如需詳細資訊，請參閱與[成本管理使用指南中AWS Billing and Cost Management的使用成本配置標籤](#)。

### 請求語法

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### [ResourceARN](#)

您要向其中添加標籤的信令通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：是

## Tags

要與指定信令通道相關聯的標籤清單。每個標籤都是金鑰值對。

類型：[Tag](#) 物件陣列

陣列成員：項目數下限為 1。項目數上限為 50。

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

TagsPerResourceExceededLimitException

您已超過可與資源關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## TagStream

服務: Amazon Kinesis Video Streams

向流添加一個或多個標籤。標籤是您可以定義並指派給AWS資源的索引鍵值配對 (值為選用)。如果您指定已存在的標籤，則標籤值會取代為您在請求中指定的值。如需詳細資訊，請參閱與[成本管理使用指南中AWS Billing and Cost Management的使用成本配置標籤](#)。

您必須提供StreamName或StreamARN。

此操作需要KinesisVideo:TagStream動作的權限。

Kinesis 影片串流最多可支援 50 個標籤。

### 請求語法

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### StreamARN

您要新增標籤或標籤的資源的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+



必要：否

## StreamName

您要新增一或多個標籤的資料流名稱。

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

## Tags

要與指定資料流相關聯的標籤清單。每個標籤都是一個鍵值對（該值是可選的）。

類型：字串到字串映射

地圖項目：最多 50 個項目。

索引鍵長度限制：長度下限為 1。長度上限為 128。

金鑰模式： $^([\p{L}\p{Z}\p{N}_.:/=+\-@]^*)\$$

值長度限制：最小長度為 0。長度上限為 256。

值模式： $[\p{L}\p{Z}\p{N}_.:/=+\-@]^*$

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## InvalidResourceFormatException

的格式StreamARN無效。

HTTP 狀態碼：400

## NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## TagsPerResourceExceededLimitException

您已超過可與資源關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UntagResource

服務: Amazon Kinesis Video Streams

從信號通道移除一個或多個標籤。在要求中，只指定一個或多個標籤鍵；請勿指定值。如果您指定的標籤鍵不存在，它將被忽略。

### 請求語法

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "TagKeyList": [ "string" ]
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### ResourceARN

您要從中刪除標籤的信令通道的 Amazon 資源名稱 ( ARN )。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：是

#### TagKeyList

您要移除之標籤的金鑰清單。

類型：字串陣列

陣列成員：項目數下限為 1。項目數上限為 50。

長度限制：長度下限為 1。長度上限為 128。

模式：`^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UntagStream

服務: Amazon Kinesis Video Streams

從串流中移除一或多個標籤。在要求中，只指定一個或多個標籤鍵；請勿指定值。如果您指定的標籤鍵不存在，它將被忽略。

在要求中，您必須提供StreamName或StreamARN。

### 請求語法

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "TagKeyList": [ "string" ]
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### StreamARN

要從中刪除標籤的流的 Amazon 資源名稱 ( ARN )。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

要從中刪除標籤的流的名稱。

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

## TagKeyList

您要移除之標籤的金鑰清單。

類型：字串陣列

陣列成員：項目數下限為 1。項目數上限為 50。

長度限制：長度下限為 1。長度上限為 128。

模式：`^[\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$`

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。



HTTP 狀態碼：400

InvalidResourceFormatException

的格式StreamARN無效。

HTTP 狀態碼：400

NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UpdateDataRetention

服務: Amazon Kinesis Video Streams

依照您指定的值增加或減少串流的資料保留期間。若要指出您要增加或縮短資料保留期間，請在要求主體中指定Operation參數。在請求中，您必須指定StreamName或StreamARN。

此操作需要KinesisVideo:UpdateDataRetention動作的權限。

變更資料保留期限會影響串流中的資料，如下所示：

- 如果資料保留期間增加，現有資料會保留新的保留期間。例如，如果資料保留期從一小時增加到七小時，則所有現有資料都會保留 7 小時。
- 如果資料保留期間縮短，則會在新的保留期間內保留現有資料。例如，如果資料保留時間從 7 小時縮短為一小時，則所有現有資料都會保留一小時，且會立即刪除超過一小時的任何資料。

### 請求語法

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DataRetentionChangeInHours": number,
  "Operation": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### CurrentVersion

您要變更其保留期間的串流版本。若要取得版本，請呼叫DescribeStream或ListStreams API。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：`[a-zA-Z0-9]+`

必要：是

### DataRetentionChangeInHours

調整目前保留的時數。您指定的值會加入目前值或從目前值中減去，這取決於 `operation`

資料保留的最小值為 0，而最大值為 87600 (十年)。

類型：整數

有效範圍：最小值為 1。

必要：是

### Operation

指出您要增加或減少保留期間。

類型：字串

有效值：`INCREASE_DATA_RETENTION` | `DECREASE_DATA_RETENTION`

必要：是

### StreamARN

您要變更其保留期的資料流的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

您要變更其保留期間的串流名稱。

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

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel 用來判斷資源狀態的 DescribeStream 或 API。

HTTP 狀態碼：400

#### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

#### VersionMismatchException

您指定的串流版本不是最新版本。若要取得最新版本，請使用 [DescribeStream API](#)。

HTTP 狀態碼：400

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UpdateImageGenerationConfiguration

服務: Amazon Kinesis Video Streams

更新StreamInfo和ImageProcessingConfiguration欄位。

### 請求語法

```
POST /updateImageGenerationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ImageGenerationConfiguration

包含 KVS 影像傳送所需資訊的結構。如果結構為 null，則配置將從流中刪除。

類型：[ImageGenerationConfiguration](#) 物件

必要：否

### StreamARN

Kinesis 影片串流的 Amazon 資源名稱 (ARN) ，您要從中更新映像產生組態的位置。您必須指定StreamName或StreamARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

要從中更新映像產生組態的串流名稱。您必須指定StreamName或StreamARN。

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

### 回應語法

```
HTTP/1.1 200
```

### 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

#### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

#### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

#### NoDataRetentionException

以小時為單位的串流資料保留等於零。

HTTP 狀態碼：400

#### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

#### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)



- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UpdateMediaStorageConfiguration

服務: Amazon Kinesis Video Streams

將 a 關聯 `SignalingChannel` 到一個流來存儲媒體。您可以指定兩種信號模式：

- 如果啟 `StorageStatus` 用，資料將儲存在 `StreamARN` 提供的中。為了讓 WebRTC 技術擷取能夠運作，串流必須啟用資料保留。
- 如果 `StorageStatus` 停用，則不會儲存任何資料，也不需要 `StreamARN` 參數。

### 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。

### 請求語法

```
POST /updateMediaStorageConfiguration HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelARN

Amazon 資源名稱 ( ARN ) 的通道。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：是

## MediaStorageConfiguration

封裝或包含媒體儲存裝置組態屬性的結構。

類型：[MediaStorageConfiguration](#) 物件

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NoDataRetentionException

以小時為單位的串流資料保留等於零。

HTTP 狀態碼：400

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UpdateNotificationConfiguration

服務: Amazon Kinesis Video Streams

更新串流的通知資訊。

### 請求語法

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### NotificationConfiguration

包含通知所需資訊的結構。如果結構為 null，則配置將從流中刪除。

類型：[NotificationConfiguration](#) 物件

必要：否

#### StreamARN

Kinesis 影片串流的 Amazon 資源名稱 (ARN)，您要從中更新通知組態的位置。您必須指定 `StreamName` 或 `StreamARN`。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

## StreamName

要從中更新通知組態的資料流名稱。您必須指定StreamName或StreamARN。

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

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：4

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## NoDataRetentionException

以小時為單位的串流資料保留等於零。

HTTP 狀態碼：400

## ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)

- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## UpdateSignalingChannel

服務: Amazon Kinesis Video Streams

更新現有的信令通道。這是非同步作業，需要時間才能完成。

如果該MessageTtlSeconds值已更新（增加或減少），則僅適用於在更新後通過此通道發送的新消息。現有消息仍然按照以前的MessageTtlSeconds值過期。

### 請求語法

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  }
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelARN

您要更新的信令通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：是

## CurrentVersion

您要更新的信令通道的目前版本。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：[a-zA-Z0-9]+

必要：是

## SingleMasterConfiguration

包含您要更新之信號通道SINGLE\_MASTER類型之組態的結構。

類型：[SingleMasterConfiguration](#) 物件

必要：否

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

## AccessDeniedException

您沒有執行此作業所需的權限。

狀態碼：

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

## ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

## VersionMismatchException

您指定的串流版本不是最新版本。若要取得最新版本，請使用 [DescribeStreamAPI](#)。

HTTP 狀態碼：400

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)

- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UpdateStream

服務: Amazon Kinesis Video Streams

更新串流中繼資料，例如裝置名稱和媒體類型。

您必須提供串流名稱或資料流的 Amazon 資源名稱 (ARN)。

若要在更新串流之前確定您擁有最新版本的串流，您可以指定串流版本。Kinesis 視訊串流會為每個串流指派一個版本。當您更新串流時，Kinesis Video Streams 會指派新的版本號碼。若要取得最新的串流版本，請使用 DescribeStream API。

UpdateStream 是非同步作業，需要時間才能完成。

### 請求語法

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DeviceName": "string",
  "MediaType": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### CurrentVersion

您要更新其中繼資料的串流版本。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：[a-zA-Z0-9]+

必要：是

### DeviceName

寫入串流的裝置名稱。

#### Note

在目前的實作中，Kinesis Video Streams 不會使用此名稱。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：[a-zA-Z0-9\_.-]+

必要：否

### MediaType

串流的媒體類型。用MediaType於指定串流包含給串流取用者的內容類型。如需媒體類型的詳細資訊，請參閱[媒體類型](#)。如果您選擇指定MediaType，請參閱[命名需求](#)。

若要在主機上播放視訊，您必須指定正確的視訊類型。例如，如果串流中的視訊是 H.264，請指定video/h264MediaType為。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：[\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)\*

必要：否

### StreamARN

您要更新其中繼資料的串流 ARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

您要更新其中繼資料的串流名稱。

串流名稱是串流的識別碼，每個帳戶和區域都必須是唯一的。

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

### 回應語法

```
HTTP/1.1 200
```

### 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

### 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

### ResourceInUseException

當輸入StreamARN或 ChannelARN in CLOUD\_STORAGE\_MODE 已對應至不同的 Kinesis Video Stream 資源，或者如果提供的輸入StreamARN或不處ChannelARN於作用中狀態，請嘗試下列其中一種方法：

1. 用於確定給定通道的流映射到的 DescribeMediaStorageConfiguration API。
2. 用於確定給定流映射到的信道的 DescribeMappedResourceConfiguration API。
3. DescribeSignalingChannel用來判斷資源狀態的DescribeStream或 API。

HTTP 狀態碼：400

### ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

### VersionMismatchException

您指定的串流版本不是最新版本。若要取得最新版本，請使用 [DescribeStream](#)API。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## Amazon Kinesis Video Streams 媒體

Amazon Kinesis Video Streams Video Streams 支持以下操作：

- [GetMedia](#)
- [PutMedia](#)

## GetMedia

服務：Amazon Kinesis Video Streams Media

使用此 API 可從 Kinesis 視訊串流擷取媒體內容。在請求中，您可以識別串流名稱或串流 Amazon 資源名稱 (ARN)，以及起始區塊。然後 Kinesis Video Streams 會依片段編號的順序傳回區塊串流。

### Note

您必須先呼叫 `GetDataEndpoint` API 以取得端點。然後使用 `--end` 點 url 參數將 `GetMedia` 請求發送到此端點。

當您將媒體資料 (片段) 放在串流上時，Kinesis Video Streams 會將每個傳入片段和相關中繼資料儲存在稱為「區塊」的位置。如需詳細資訊，請參閱 [PutMedia](#)。GetMedia API 會從您在要求中指定的區塊開始傳回這些區塊的串流。

使用 GetMedia API 時適用以下限制：

- 用戶端每個串流每秒 GetMedia 最多可呼叫五次。
- 在工作階段期間，Kinesis Video Streams 以每秒 25 MB (或每秒 200 MB) 的速率傳送媒體資料。GetMedia

### Note

GetMedia HTTP 回應狀態碼會立即傳回，但如果沒有可供播放的擷取片段，HTTP 回應裝載的讀取會在 3 秒後逾時。

### Note

如果在叫用 Kinesis 影片串流媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- `x-amz-ErrorType` HTTP 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- `x-amz-RequestId` HTTP 標頭 — 如果您想向其報告問題 AWS，支持團隊可以在給出請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可用於對錯誤是否可重試以及在什麼情況下進行程式設計決定，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及[常見錯誤](#)。

## 請求語法

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
  "StartSelector": {
    "AfterFragmentNumber": "string",
    "ContinuationToken": "string",
    "StartSelectorType": "string",
    "StartTimestamp": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

## 請求主體

請求接受採用 JSON 格式的下列資料。

### [StartSelector](#)

標識要從指定的流獲取的起始塊。

類型：[StartSelector](#) 物件

必要：是

### [StreamARN](#)

您要從其中取得媒體內容的串流的 ARN。如果未指定 `streamARN`，則必須指定 `streamName`。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

Kinesis 視訊串流名稱，您要從中取得媒體內容的位置。如果未指定 `streamName`，則必須指定 `streamARN`。

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

### 回應語法

```
HTTP/1.1 200  
Content-Type: ContentType
```

```
Payload
```

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

回應會傳回下列 HTTP 標頭。

### ContentType

要求的媒體的內容類型。

長度限制：長度下限為 1。長度上限為 128。

模式：`^[a-zA-Z0-9_\. \-]+$`

回應傳回以下內容作為 HTTP 主體。

## Payload

承載 Kinesis Video Streams 傳回是指定串流中的一系列區塊。如需區塊的詳細資訊，請參閱 [PutMedia](#)。Kinesis Video Streams 在 GetMedia 呼叫中傳回的區塊也包含下列其他 Matroska (MKV) 標記：

- AWS\_KINESISVideo\_CONTION\_TOKEN ( UTF-8 字符串 ) -如果您的 GetMedia 調用終止，您可以在下一個請求中使用此延續令牌來獲取最後一個請求終止的下一個塊。
- AWS\_KINESISVE\_MILLIS\_HEND\_NOW (UTF-8 字串)-用戶端應用程式可以使用此標記值來判斷回應中傳回的區塊後面有多遠離串流上的最新區塊。
- AW\_ 動態片段-區塊中傳回的片段編號。
- AW\_ 激活服務器時間戳-片段的服務器時間戳。
- AW\_ 生產者 \_ 時間戳記-片段的生產者時間戳。

如果發生錯誤，將出現以下標籤：

- AW\_ 錯誤代碼-導致停止的錯誤的字符串描述。 GetMedia
- AW\_ 激活視頻錯誤 ID：錯誤的整數代碼。

錯誤代碼如下：

- 3002-寫入資料流時發生錯誤
- 4000-找不到請求的片段
- 4500-對串流的 KMS 金鑰存取遭拒
- 4501-串流的 KMS 金鑰已停用
- 4502-串流的 KMS 金鑰上的驗證錯誤
- 4503-在串流中指定的 KMS 金鑰無法使用
- 4504-串流中指定的 KMS 金鑰使用方式無效
- 4505-串流中指定的 KMS 金鑰狀態無效
- 4506-找不到串流中指定的 KMS 金鑰
- 內部錯誤

## 錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

## ConnectionLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端連線限制。

HTTP 狀態碼：400

## InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

## InvalidEndpointException

調用者使用錯誤的端點將數據寫入流。在接收這樣的異常時，用戶必須調用 `GetDataEndpoint` 用 `APIName` 設置為 `PUT_MEDIA` 並使用端點從響應調用下一個 `PutMedia` 調用。

HTTP 狀態碼：400

## NotAuthorizedException

調用者未被授權在給定的流上執行操作，或令牌已過期。

狀態碼：

## ResourceNotFoundException

狀態碼：404，具有給定名稱的流不存在。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

## PutMedia

服務：Amazon Kinesis Video Streams Media

使用此 API 可將媒體資料傳送至 Kinesis 視訊串流。

### Note

您必須先呼叫 GetDataEndpoint API 以取得端點。然後使用 `--end` 點 url 參數將 PutMedia 請求發送到此端點。

在要求中，您可以使用 HTTP 標頭來提供參數資訊，例如串流名稱、時間戳記，以及時間戳記值是絕對值還是相對於生產者開始記錄的時間。您可以使用要求主體來傳送媒體資料。Kinesis Video Streams 僅支援 Matroska (MKV) 容器格式，以便使用此 API 傳送媒體資料。

使用此 API 傳送資料時，您可以使用下列選項：

- 實時發送媒體數據：例如，安全攝像機可以在生成幀時實時發送幀。這種方法可以最大限度地減少視頻錄製和在線發送數據之間的延遲。這被稱為連續生產者。在這種情況下，消費者應用程序可以實時或在需要時讀取流。
- 離線傳送媒體資料（批次）：例如，身體攝影機可能會錄製影片數小時，並將其儲存在裝置上。稍後，當您將攝影機連接到連接埠時，攝影機可以啟動 PutMedia 工作階段以將資料傳送至 Kinesis 視訊串流。在這個案例中，延遲不是問題。

使用此 API 時，請注意下列考量事項：

- 您必須指定 streamName 或 streamARN，但不能同時指定兩者。
- 為了能夠在控制台上或通過 HLS 播放媒體，每個片段的軌道 1 應包含 h.264 編碼的視頻，片段元數據中的 CodeId 應該是「V\_MPEG/ISO/AVC」，並且片段元數據應包括 AVCC 格式的 h.264 編碼器私有數據。或者，每個片段的音軌 2 應包含 AAC 編碼音訊，片段中繼資料中的 CodeId 應該是「A\_AAC」，而片段中繼資料應包含 AAC 轉碼器私人資料。
- 您可能會發現使用單個長時間運行的會 PutMedia 話並在有效負載中發送大量媒體數據片段更容易。針對收到的每個片段，Kinesis Video Streams 都會傳送一或多個確認訊息。潛在的網路考量可能會導致您無法在產生這些確認時取得所有這些確認。
- 您可以選擇多個連續的 PutMedia 工作階段，每個工作階段的片段較少，以確保您即時從服務中獲得所有確認。



**Note**

如果您在多個同時PutMedia工作階段上將資料傳送至相同的串流，則媒體片段會在串流上交錯。您應該確保在應用程序場景中這是正常的。

使用 PutMedia API 時適用以下限制：

- 用戶端每個串流每秒PutMedia最多可呼叫五次。
- 用戶端每秒最多可以傳送五個片段的每個串流。
- Kinesis Video Streams 讀取媒體資料的速率最高可達每秒 12.5 MB，或在工作階段期間讀取 100 Mbps。PutMedia

請注意下列限制。在這些情況下，Kinesis Video Streams 會在回應中傳送錯誤確認。

- 不允許具有時間碼超過允許上限且包含超過 50 MB 資料的片段。
- 不允許包含三個以上軌道的片段。每個片段中的每個幀都必須具有與片段標題中定義的軌道之一相同的軌道編號。此外，每個片段必須為片段標頭中定義的每個軌道包含至少一個幀。
- 對於片段元數據中定義的每個軌道，每個片段必須至少包含一個幀。
- 片段中最早的影格時間戳記必須在前一個片段中的最新影格時間戳記之後。
- 包含多個 MKV 片段或包含不允許的 MKV 元素 ( 如track\* ) 的 MKV 流也會導致錯誤確認。

Kinesis Video Streams 會將每個傳入片段和相關中繼資料儲存在稱為「區塊」的項目中。片段元數據包括以下內容：

- PutMedia請求開始時提供的 MKV 標頭
- 下列適用於片段的 Kinesis 影片串流特定中繼資料：
  - server\_timestamp-Kinesis Video Streams 開始接收片段時的時間戳記。
  - producer\_timestamp-時間戳記，生產者開始記錄片段時。Kinesis Video Streams 會使用請求中收到的三條資訊來計算此值。
    - 在要求主體中接收到的片段時間碼值以及片段。
    - 兩個請求標頭：producerStartTimestamp ( 生產者開始記錄時 ) 和fragmentTimeCodeType ( 有效負載中的片段時間碼是絕對還是相對的 ) 。

然後 Kinesis Video Streams 會計算片段，producer\_timestamp如下所示：

如果 `fragmentTimeCodeType` 是相對的，那麼

`producer_timestamp = producerStartTimeStamp + 片段時間碼`

如果 `fragmentTimeCodeType` 是絕對的，那麼

`producer_timestamp = 片段時間碼 (轉換為毫秒)`

- Kinesis Video Streams 指派的唯一片段編號。

#### Note

當您提出 `GetMedia` 要求時，Kinesis Video Streams 會傳回這些區塊的串流。用戶端可以視需要處理中繼資料。

#### Note

此作業僅適用於 AWS SDK for Java。其他語言的 AWS SDK 不支援此功能。

#### Note

Kinesis Video Streams 不會剖析和驗證透過 API 擷取和封存轉碼器的私人資料。PutMedia 當透過 HLS API 取用串流時，KVS 會從 MPEG-TS 和 MP4 片段封裝的轉碼器私人資料中擷取和驗證必要的資訊。

#### Note

如果在叫用 Kinesis 影片串流媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- `x-amz-ErrorTypeHTTP` 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- `x-amz-RequestIdHTTP` 標頭 — 如果您想向其報告問題 AWS，支持團隊可以在給出請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可用於對錯誤是否可重試以及在什麼情況下進程式設計決定，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及[常見錯誤](#)。

## 請求語法

```
POST /putMedia HTTP/1.1
x-amzn-stream-name: StreamName
x-amzn-stream-arn: StreamARN
x-amzn-fragment-timecode-type: FragmentTimecodeType
x-amzn-producer-start-timestamp: ProducerStartTimestamp
```

*Payload*

## URI 請求參數

請求會使用下列 URI 參數。

### [FragmentTimecodeType](#)

您可以將此值當做 `x-amzn-fragment-timecode-type` HTTP 標頭傳遞。

指出片段（有效負載，HTTP 請求主體）中的時間碼是絕對還是相對於 `producerStartTimestamp`。Kinesis Video Streams 使用此資訊來 `producer_timestamp` 計算要求中收到的片段，如 API 概觀所述。

有效值:ABSOLUTE | RELATIVE

必要：是

### [ProducerStartTimestamp](#)

您可以將此值當做 `x-amzn-producer-start-timestamp` HTTP 標頭傳遞。

這是生產者開始記錄媒體時間戳記（而不是請求中特定片段的時間戳）。

### [StreamARN](#)

您可以將此值當做 `x-amzn-stream-arn` HTTP 標頭傳遞。

您要在其中寫入媒體內容的 Kinesis 視訊串流的 Amazon 資源名稱 (ARN)。如果未指定 `streamARN`，則必須指定 `streamName`。

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

### StreamName

您可以將此值當做 `x-amzn-stream-name` HTTP 標頭傳遞。

您要在其中寫入媒體內容的 Kinesis 視訊串流的名稱。如果未指定 `streamName`，則必須指定 `streamARN`。

長度限制：長度下限為 1。長度上限為 256。

模式：`[a-zA-Z0-9_.-]+`

### 請求主體

請求接受下列二進位資料。

### Payload

要寫入 Kinesis 視訊串流的媒體內容。在目前的實作中，Kinesis Video Streams 僅支援具有單一 MKV 區段的 Matroska (MKV) 容器格式。一個區段可以包含一或多個叢集。

#### Note

每個 MKV 群集映射到一個 Kinesis 視頻流片段。無論您選擇什麼叢集持續時間都會成為片段持續

### 回應語法

```
HTTP/1.1 200
```

*Payload*

### 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

回應傳回以下內容作為 HTTP 主體。

## Payload

Kinesis Video Streams 成功收到PutMedia要求之後，服務會驗證要求標頭。然後，服務會開始讀取有效負載，並先傳送 HTTP 200 回應。

然後，服務返回一個包含由換行符分隔的一系列 JSON Acknowledgement 對象 (對象) 的流。在傳送媒體資料的相同連接上接收確認。一PutMedia個請求可以有很多確認。每一個都Acknowledgement包含下列索引鍵值配對：

- AckEventType-確認代表的事件類型。
  - 緩衝：Kinesis Video Streams 已開始接收片段。Kinesis Video Streams 會在收到片段資料的第一個位元組時傳送第一個緩衝確認。
  - 已收到：Kinesis Video Streams 收到整個片段。如果您未將串流設定為保留資料，則生產者可以在收到此確認後停止緩衝片段。
  - 持續存在：Kinesis Video Streams 已保留片段 (例如，傳送至 Amazon S3)。如果您將串流設定為保留資料，則會收到此確認。收到此確認後，生產者可以停止緩衝片段。
  - 錯誤：Kinesis Video Streams 在處理片段時發生錯誤。您可以檢閱錯誤碼並決定下一個動作方案。
  - 閒置：PutMedia工作階段正在進行中。不過，Kinesis Video Streams 目前並未接收資料。Kinesis Video Streams 會在最後一次接收到的資料之後定期傳送此確認，最長可達 30 秒。如果在 30 秒內未收到任何資料，Kinesis Video Streams 會關閉要求。

### Note

此確認可以幫助生產者確定PutMedia連接是否存在，即使它沒有發送任何數據。

- FragmentTimecode-傳送確認的片段時間碼。

如果AckEventType為「閒置」，則元素可能會遺失。
- FragmentNumber-Kinesis 視訊串流產生的片段編號，以傳送確認訊息。
- ErrorId和 ErrorCode-如果AckEventType是Error，則此欄位會提供對應的錯誤代碼。以下是錯誤 ID 及其對應的錯誤代碼和錯誤訊息的清單：
  - 4000-串流讀取錯誤-讀取資料串流時發生錯誤。
  - 4001-最大片段 \_ 大小到達-片段大小大於允許的最大限制為 50 MB。
  - 4002-最大片段 \_ 持續時間達到-片段持續時間大於允許的最大限制。
  - 4003-最大連線 \_ 持續時間達到-連線持續時間大於允許的最大臨界值。
  - 4004-片段時間碼小於上一個時間碼 (在調用中，您不能按順序發送片段)。PutMedia

- 4005-超過 \_ 允許追蹤 \_ 找到-在 MKV 中找到多個軌道。(已取代)
- 4006-無效資料-無法將輸入串流剖析為有效的 MKV 格式。
- 4007-無效生產者 \_ 時間戳記-無效的生產者時間戳記。
- 4008-流不活躍-流不再存在 ( 刪除 )。
- 4009-片段 \_ 中繼資料達到限制-已達片段中繼資料限制。請參閱開發人員指南的「[限制](#)」一節。
- 4010-軌道編號不匹配-MKV 幀中的曲目編號與 MKV 標題中的曲目不匹配。
- 4011-框架錯誤-片段不包含 MKV 標頭中至少一個軌道的任何幀。
- 4012-無效的片段中繼資料-片段中繼資料名稱不能以字串開頭。 AWS\_
- 4500-拒絕存取串流的指定 KMS 金鑰。
- 已停用-已停用串流的指定 KMS 金鑰。
- 4502-驗證錯誤-串流的指定 KMS 金鑰驗證失敗。
- 4503-無法使用-串流的指定 KMS 金鑰無法使用。
- 4504-無效使用-串流指定的 KMS 金鑰用法無效。
- 4505-無效狀態-串流的指定 KMS 金鑰狀態無效。
- 4506-找不到串流的指定 KMS 金鑰。
- 5000-內部錯誤-內部服務錯誤。
- 5001-封存 \_ 錯誤-Kinesis Video Streams 無法將片段保存至資料存放區。

#### Note

生產者在為長時間運行的PutMedia請求發送有效負載時，應該讀取確認的響應。由於中間代理服務器上的緩衝，生產者可能會同時收到確認塊。想要及時收到確認的生產者可以在每個PutMedia請求中傳送較少的片段。

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### ConnectionLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端連線限制。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### InvalidEndpointException

調用者使用錯誤的端點將數據寫入流。在接收這樣的異常時，用戶必須調用 `GetDataEndpoint` 用 `APIName` 設置為 `PUT_MEDIA` 並使用端點從響應調用下一個 `PutMedia` 調用。

HTTP 狀態碼：400

### NotAuthorizedException

調用者未被授權在給定的流上執行操作，或令牌已過期。

狀態碼：

### ResourceNotFoundException

狀態碼：404，具有給定名稱的流不存在。

HTTP 狀態碼：404

## 範例

### 確認格式

確認書的格式如下：

```
{
  Acknowledgement : {
    "EventType": enum,
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
```

```
        "ErrorId" : String
    }
}
```

### 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

## Amazon Kinesis Video Streams 歸檔媒體

Amazon Kinesis Video Streams 存檔媒體支持以下操作：

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)



## GetClip

服務：Amazon Kinesis Video Streams Archived Media

在指定時間範圍內，從指定的視訊串流下載包含已封存的隨選媒體的 MP4 檔案 (剪輯片段)。

StreamName 和 StreamArn 參數都是選擇性的，但您必須在呼叫此 API 作業時指定 StreamName 或 StreamArn。

### Note

您必須先呼叫 GetDataEndpoint API 以取得端點。然後使用 `--end` 點 url 參數將 GetClip 請求發送到此端點。

Amazon Kinesis 影片串流對於透過 MP4 提供資料有以下要求：

- [視頻播放軌道要求](#)。
- 資料保留必須大於 0。
- 每個片段的影片軌道必須針對視訊媒體的 H.264 格式的進階視訊編碼 (AVC)，以及針對 H.265 格式的 HEVC 中包含轉碼器私有資料。如需詳細資訊，請參閱 [MPEG-4 specification ISO/IEC 14496-15](#)。如需將串流資料轉換為特定格式的相關資訊，請參閱 [NAL Adaptation Flags](#)。
- 每個片段的音訊軌道 (如果有) 必須包含 AAC 格式的編碼器私有資料 ([AAC 規格 ISO/IEC 13818-7](#)) 或 [MS Wave 格式](#)。

您可以透過監控 GetClip.OutgoingBytes Amazon CloudWatch 指標來監控傳出資料量。如需使用 CloudWatch 監視 Kinesis Video Streams 的相關資訊，請參閱 [監控 Kinesis Video Streams](#)。如需定價資訊，請參閱 [Amazon Kinesis Video Streams 定價和定AWS價](#)。外寄資 AWS 料需支付費用。

### 請求語法

```
POST /getClip HTTP/1.1
Content-type: application/json

{
  "ClipFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  }
}
```

```
    }  
  },  
  "StreamARN": "string",  
  "StreamName": "string"  
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

## 請求主體

請求接受採用 JSON 格式的下列資料。

### ClipFragmentSelector

要求的剪輯片段的時間範圍和時間戳記的來源。

類型：[ClipFragmentSelector](#) 物件

必要：是

### StreamARN

要擷取媒體剪輯的資料流的 Amazon 資源名稱 (ARN)。

您必須指定 StreamName 或串流網路。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

要擷取媒體片段的資料流名稱。

您必須指定 StreamName 或串流網路。

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

## 回應語法

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

回應會傳回下列 HTTP 標頭。

### ContentType

要求剪輯片段中媒體的內容類型。

長度限制：長度下限為 1。長度上限為 128。

模式：`^[a-zA-Z0-9_\.\\-]+$`

回應傳回以下內容作為 HTTP 主體。

### Payload

傳統的 MP4 文件，其中包含來自指定視頻流的媒體剪輯。輸出將包含指定開始時間戳記的前 100 MB 或前 200 個片段。如需詳細資訊，請參閱 [Kinesis Video Streams 限制](#)。

## 錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱 [Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

#### InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

#### InvalidCodecPrivateDataException

視訊串流中至少一個軌道中的編解碼器私人資料對此作業無效。

HTTP 狀態碼：400

#### InvalidMediaFrameException

根據指定的編解碼器，無法剖析要求剪輯片段中的一個或多個影格。

HTTP 狀態碼：400

#### MissingCodecPrivateDataException

在視訊串流的至少一個軌道中找不到轉碼器私人資料。

HTTP 狀態碼：400

#### NoDataRetentionException

GetImages 要求不保留資料的串流 (也就是說，a DataRetentionInHours 為 0)。

HTTP 狀態碼：400

#### NotAuthorizedException

狀態碼：403，調用者未授權在給定流上執行操作，或令牌已過期。

狀態碼：

#### ResourceNotFoundException

GetImages 當 Kinesis 影片串流找不到您指定的串流時，就會擲回此錯誤。

GetHLSStreamingSessionURL 如果在請求的時間範圍內沒有片段 PlaybackMode 的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話，或者如果在過去 30 秒內沒有片段 PlaybackMode 的 LIVE 流請求具有 of 的會話，則 GetDASHStreamingSessionURL 拋出此錯誤。

HTTP 狀態碼：404

## UnsupportedStreamMediaTypeException

媒體類型 (例如 h.264 或 h.265 視訊或 AAC 或 G.711 音訊) 無法從播放工作階段的第一個片段中曲目的轉碼器識別碼來決定。音軌 1 的轉碼器 ID 應該是 V\_MPEG/ISO/AVC 音軌 2 的編解碼器識別碼 (選擇性) A\_AAC。

HTTP 狀態碼 : 400

### 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

## GetDASHStreamingSessionURL

服務：Amazon Kinesis Video Streams Archived Media

透過 HTTP (破折號) URL 擷取串流的 MPEG 動態自適應串流。然後，您可以在媒體播放器中打開 URL 以查看流內容。

StreamName和StreamARN參數都是可選的，但您必須在調用此 API 操作StreamARN時指定StreamName或。

亞 Amazon Kinesis 影片串流對透過 MPEG-DASH 提供資料有以下要求：

- [視頻播放軌道要求](#)。
- 資料保留必須大於 0。
- 每個片段的影片軌道必須針對視訊媒體的 H.264 格式的進階視訊編碼 (AVC)，以及針對 H.265 格式的 HEVC 中包含轉碼器私有資料。如需詳細資訊，請參閱 [MPEG-4 specification ISO/IEC 14496-15](#)。如需將串流資料轉換為特定格式的相關資訊，請參閱 [NAL Adaptation Flags](#)。
- 每個片段的音訊軌道 (如果有) 必須包含 AAC 格式的編碼器私有資料 ([AAC 規格 ISO/IEC 13818-7](#)) 或 [MS Wave 格式](#)。

下列程序說明如何搭配 Kinesis Video Streams 使用 MPEG-DASH：

1. 呼叫 GetDataEndpoint API 以取得端點。然後使用 `--end` 點 url 參數將GetDASHStreamingSessionURL請求發送到此端點。
2. 使用擷取 MPEG-破折號網址。GetDASHStreamingSessionURLKinesis Video Streams 會建立 MPEG-DASH 串流工作階段，以便使用 MPEG DASH 通訊協定存取串流中的內容。GetDASHStreamingSessionURL傳回工作階段 MPEG-DASH 資訊清單 (使用 MPEG DASH 串流所需的根資源) 的驗證 URL (包含加密的工作階段權杖)。

### Note

不要在未經授權的實體可以訪問它的地方共享或存儲此令牌。令牌提供對流內容的訪問。使用與 AWS 憑據一起使用的相同措施保護令牌。

通過清單提供的媒體僅由請求的流，時間範圍和格式組成。沒有其他媒體資料 (例如要求視窗外的框架或替代位元率) 可用。

3. 將 MPEG-DASH 資訊清單的 URL ( 包含加密的工作階段權杖 ) 提供給支援 MPEG-DASH 通訊協定的媒體播放器。Kinesis Video Streams 可透過資訊清單 URL 使用初始化片段和媒體片段。初始化片段包含流的編解碼器私有數據，以及設置視頻或音頻解碼器和渲染器所需的其他數據。媒體片段包含編碼的視頻幀或編碼的音頻樣本。
4. 媒體播放器接收經過身份驗證的 URL，並正常請求流元數據和媒體數據。當媒體播放器請求數據時，它會調用以下操作：
  - 獲取 DashManifest：檢索 MPEG 破折號清單，其中包含要播放的媒體的元數據。
  - 獲取 MP4InitFragment：檢索 MP4 初始化片段。媒體播放器通常會在載入任何媒體片段之前載入初始化片段。此片段包含「fytpmoov」和「MP4 原子，以及初始化媒體播放器解碼器所需的子原子。

初始化片段不對應於 Kinesis 視訊串流中的片段。它僅包含流和相應軌道的編解碼器私有數據，媒體播放器需要解碼媒體幀。

- 獲取 MP4MediaFragment：檢索 MP4 媒體片段。這些片段包含「moofmdat」和「MP4 原子及其子原子，其中包含編碼片段的媒體框架及其時間戳記。

#### Note

在串流工作階段中提供第一個媒體片段之後，任何不包含相同轉碼器私有資料的片段都會在載入這些不同的媒體片段時傳回錯誤。因此，編解碼器私有資料不應該在工作階段中的片段之間變更。這也意味著如果流中的片段從僅具有視頻變為同時具有音頻和視頻，則會話失敗。

使用此動作擷取的資料可計費。詳情請參閱[定價](#)。

#### Note

如需適用於 MPEG-DASH 工作階段的限制，請參閱 [Kinesis Video Streams 限制](#)。

您可以透過監控 `GetMP4MediaFragment.OutgoingBytes` Amazon CloudWatch 指標來監控媒體播放器使用的資料量。如需使用 CloudWatch 監視 Kinesis Video Streams 的相關資訊，請參閱[監控 Kinesis Video Streams](#)。如需定價資訊，請參閱[Amazon Kinesis Video Streams 定價和定價](#)。HLS 工作階段和傳出 AWS 資料均需支付費用。

如需有關 HLS 的詳細資訊，請參閱 [Apple 開發人員網站](#)上的 [HTTP 即時串流](#)。

### ⚠ Important

如果在叫用 Kinesis 影片串流封存媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- `x-amz-ErrorTypeHTTP` 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- `x-amz-RequestIdHTTP` 標頭-如果您想向支持團隊報告問題，可以更好地診斷問題，如果給出請求 ID。AWS

HTTP 狀態碼和 `ErrorType` 標頭都可用於對錯誤是否可重試以及在什麼情況下進行程式設計決定，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及[常見錯誤](#)。

### 請求語法

```
POST /getDASHStreamingSessionURL HTTP/1.1
Content-type: application/json
```

```
{
  "DASHFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "DisplayFragmentNumber": "string",
  "DisplayFragmentTimeStamp": "string",
  "Expires": number,
  "MaxManifestFragmentResults": number,
  "PlaybackMode": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。



## 請求主體

請求接受採用 JSON 格式的下列資料。

### [DASHFragmentSelector](#)

請求片段的時間範圍和時間戳記的來源。

如果是或，則PlaybackMode需要此參數ON\_DEMAND或LIVE\_REPLAY。如果是，則此參數PlaybackMode是可選的LIVE。如果PlaybackMode是LIVE，則FragmentSelectorType可以設定，但不TimestampRange應設定。如果PlaybackMode是ON\_DEMAND或LIVE\_REPLAY，則TimestampRange必須同時設定FragmentSelectorType和。

類型：[DASHFragmentSelector](#) 物件

必要：否

### [DisplayFragmentNumber](#)

片段會根據工作階段中的序號在資訊清單檔案中識別。如果設定 DisplayFragmentNumber 為ALWAYS，則會將 Kinesis Video Streams 片段編號新增至資訊清單檔案中的每個 S 元素，其屬性名稱為「kv: fn」。這些片段編號可用於記錄或與其他 API ( 例如GetMedia和GetMediaForFragmentList ) 一起使用。利用這些自定義屬性需要自定義 MPEG-DASH 媒體播放器。

預設值為 NEVER。

類型：字串

有效值:ALWAYS | NEVER

必要：否

### [DisplayFragmentTimestamp](#)

根據 MPEG-DASH 規範，清單文件中片段的掛鐘時間可以使用清單本身中的屬性來導出。但是，通常，MPEG-DASH 兼容的媒體播放器無法正確處理媒體時間軸中的間隙。Kinesis Video Streams 會調整資訊清單檔案中的媒體時間表，以啟用不連續性的媒體播放。因此，從資訊清單檔案衍生的掛鐘時間可能不正確。如果設定 DisplayFragmentTimestamp 為ALWAYS，則會將精確的片段時間戳記新增至資訊清單檔案中的每個 S 元素，其屬性名稱為「kv: ts」。利用此自定義屬性需要自定義 MPEG-DASH 媒體播放器。

預設值為 NEVER。如果 [DASHFragmentSelector](#) 是 SERVER\_TIMESTAMP，時間戳記將是伺服器啟動時間戳記。同樣，如果 [DASHFragmentSelector](#) 是 PRODUCER\_TIMESTAMP，時間戳記將是生產者開始時間戳記。

類型：字串

有效值: ALWAYS | NEVER

必要：否

### [Expires](#)

要求的工作階段到期前的時間 (以秒為單位)。此值可以介於 300 (5 分鐘) 到 43200 (12 小時) 之間。

當工作階段到期時，無法對該工作階段 `GetMP4MediaFragment` 進行新呼叫 `GetMP4InitFragment`、或 `GetDashManifest`

預設值為 300 (5 分鐘)。

類型：整數

有效範圍：最小值 300。最大值為 43200。

必要：否

### [MaxManifestFragmentResults](#)

MPEG-DASH 資訊清單中傳回的片段數目上限。

如果 `PlaybackMode` 是 LIVE，則最新的片段會返回到此值。如果 `PlaybackMode` 是 ON\_DEMAND，則返回最舊的片段，直到此最大數量。

當即時 MPEG-DASH 資訊清單中可用的片段數目較多時，視訊播放程式通常會在開始播放之前緩衝內容。增加緩衝區大小會增加播放延遲，但會降低播放期間發生重新緩衝的可能性。我們建議實時 MPEG-DASH 清單至少具有 3 個片段和最多 10 個片段。

如果是 LIVE 或 LIVE\_REPLAY，預設值為 5 個片段，如果 `PlaybackMode` 是，則預設值 `PlaybackMode` 為 1,000 ON\_DEMAND。

1,000 個片段的最大值對應於具有 1 秒片段的流上的視頻超過 16 分鐘，以及具有 10 秒片段的流上的視頻超過 2 1/2 小時。

類型：Long

有效範圍：最小值為 1。最大值為 5000。

必要：否

## PlaybackMode

是否要擷取即時、即時重播或封存的隨選資料。

三種工作階段類型的功能包括下列項目：

- **LIVE**：對於此類型的工作階段，MPEG-DASH 資訊清單會在最新片段可用時持續更新。我們建議媒體播放程式以一秒鐘的時間間隔擷取新的資訊清單。在媒體播放程式中播放此類工作階段時，使用者介面通常會顯示「即時」通知，而且沒有用於選擇播放視窗中要顯示的位置的定位器控制項。

### Note

在LIVE模式下，最新的可用片段包含在 MPEG-DASH 資訊清單中，即使片段之間存在間隙（也就是說，如果片段遺失）。像這樣的間隙可能會導致媒體播放器停止或導致播放跳躍。在此模式下，如果片段比播放清單中的最新片段舊，則不會將片段添加到 MPEG-DASH 清單中。如果在將後續片段新增至資訊清單後，遺失的片段變為可用，則不會新增較舊的片段，且不會填滿間隙。

- **LIVE\_REPLAY**：對於此類型的工作階段，MPEG-DASH 資訊清單的更新方式與LIVE模式的更新方式類似，不同之處在於它會從指定開始時間包含片段開始。而不是在攝入片段時添加片段，而是在下一個片段的持續時間過去時添加片段。例如，如果會話中的片段長度為兩秒鐘，則每兩秒鐘將一個新的片段添加到清單中。此模式對於能夠在偵測到事件時開始播放，並繼續在工作階段建立時尚未擷取的即時串流媒體非常有用。此模式對於串流先前封存的媒體也很有用，而不受ON\_DEMAND模式中 1,000 片段限制的限制。
- **ON\_DEMAND**：對於此類型的工作階段，MPEG-DASH 資訊清單會包含工作階段的所有片段，直到中指定的數目為止。MaxManifestFragmentResults每個工作階段只能擷取一次資訊清單。在媒體播放程式中播放此類工作階段時，使用者介面通常會顯示定位器控制項，以便在播放視窗中選擇要顯示的位置。

在所有播放模式中，如果FragmentSelectorType是PRODUCER\_TIMESTAMP，如果有多個片段具有相同的開始時間戳記，則 MPEG-DASH 清單中包含具有較大片段編號的片段（即較新的片段）。其他片段不包括在內。MPEG-DASH 清單中仍包含具有不同時間戳但持續時間重疊的片段。這可能會導致媒體播放器中出現意外行為。

預設值為 LIVE。

類型：字串

有效值:LIVE | LIVE\_REPLAY | ON\_DEMAND

必要：否

### StreamARN

要擷取 MPEG-DASH 資訊清單 URL 的資料流的 Amazon 資源名稱 (ARN)。

您必須指定StreamName或StreamARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

要擷取 MPEG DASH 資訊清單 URL 之資料流的名稱。

您必須指定StreamName或StreamARN。

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

### 回應語法

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

{
  "DASHStreamingSessionURL": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [DASHStreamingSessionURL](#)

媒體播放程式可用來擷取 MPEG-DASH 資訊清單的 URL (包含工作階段權杖)。

類型：字串

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱 [Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

### InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

### InvalidCodecPrivateDataException

視訊串流中至少一個軌道中的編解碼器私人資料對此作業無效。

HTTP 狀態碼：400

### MissingCodecPrivateDataException

在視訊串流的至少一個軌道中找不到轉碼器私人資料。

HTTP 狀態碼：400

### NoDataRetentionException

GetImages 要求不保留資料的串流 (也就是說，a DataRetentionInHours 為 0)。

HTTP 狀態碼：400

## NotAuthorizedException

狀態碼：403，調用者未授權在給定流上執行操作，或令牌已過期。

狀態碼：

## ResourceNotFoundException

GetImages當 Kinesis 影片串流找不到您指定的串流時，就會擲回此錯誤。

GetHLSStreamingSessionURL如果在請求的時間範圍內沒有片段PlaybackMode的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話，或者如果在過去 30 秒內沒有片段PlaybackMode的LIVE流請求具有 a 的會話，則GetDASHStreamingSessionURL拋出此錯誤。

HTTP 狀態碼：404

## UnsupportedStreamMediaTypeException

媒體類型 (例如 h.264 或 h.265 視訊或 AAC 或 G.711 音訊) 無法從播放工作階段的第一個片段中曲目的編解碼器 ID 來決定。音軌 1 的轉碼器 ID 應該是V\_MPEG/ISO/AVC音軌 2 的編解碼器識別碼 (選擇性) A\_AAC。

HTTP 狀態碼：400

## 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

## GetHLSStreamingSessionURL

服務：Amazon Kinesis Video Streams Archived Media

擷取串流的 HTTP 即時串流 (HLS) 網址。然後，您可以在瀏覽器或媒體播放器中打開 URL 以查看流內容。

StreamName和StreamARN參數都是可選的，但您必須在調用此 API 操作StreamARN時指定StreamName或。

Amazon Kinesis 影片串流對於透過 HLS 提供資料有以下要求：

- [視頻播放軌道要求](#)。
- 資料保留必須大於 0。
- 每個片段的視訊軌道必須包含 H.264 格式的「進階視訊編碼」(AVC) 中的編解碼器私人資料，或是 H.265 格式的 HEVC ([MPEG-4 規格](#))。如需將串流資料轉換為特定格式的相關資訊，請參閱 [NAL Adaptation Flags](#)。
- 每個片段的音軌 (如果存在) 必須包含 AAC 格式的編解碼器私有數據 (A [AC 規格 ISO/IEC 13818-7](#))。

Kinesis Video Streams HLS 工作階段包含片段的 MPEG-4 形式 (也稱為 FMP4 或 CMAF) 或 MPEG-2 形式 (也稱為 TS 區塊，HLS 規格也支援) 的片段。如需有關 HLS 片段類型的詳細資訊，請參閱 [HLS 規格](#)。

下列程序說明如何搭配 Kinesis Video Streams 使用 HLS：

1. 呼叫 GetDataEndpoint API 以取得端點。然後使用 `--end` 點 url 參數將GetHLSStreamingSessionURL請求發送到此端點。
2. 使GetHLSStreamingSessionURL用擷取 HLS 網址。Kinesis Video Streams 會建立 HLS 串流工作階段，以便使用 HLS 通訊協定存取串流中的內容。GetHLSStreamingSessionURL傳回工作階段的 HLS 主播放清單 (使用 HLS 串流所需的根資源) 的驗證 URL (包含加密的工作階段權杖)。

### Note

不要在未經授權的實體可以訪問它的地方共享或存儲此令牌。令牌提供對流內容的訪問。使用與 AWS 憑據一起使用的相同措施來保護令牌。

透過播放清單提供的媒體僅包含要求的串流、時間範圍和格式。沒有其他媒體資料 (例如要求視窗外的框架或替代位元率) 可用。

3. 將 HLS 主播放清單的 URL (包含加密的工作階段權杖) 提供給支援 HLS 通訊協定的媒體播放器。Kinesis Video Streams 可讓 HLS 媒體播放清單、初始化片段和媒體片段透過主播放清單 URL 取得。初始化片段包含流的編解碼器私有數據，以及設置視頻或音頻解碼器和渲染器所需的其他數據。媒體片段包含 H.264 編碼的視訊影格或 AAC 編碼的音訊樣本。
4. 媒體播放器接收經過身份驗證的 URL，並正常請求流元數據和媒體數據。當媒體播放器請求數據時，它會調用以下操作：
  - **GetHLSMasterPlaylist**：擷取 HLS 主播放清單，其中包含每個音軌 **GetHLSMediaPlaylist** 動作的 URL，以及媒體播放器的其他中繼資料，包括估計的位元率和解析度。
  - **GetHLSMediaPlaylist**：擷取 HLS 媒體播放清單，其中包含可透過動作存取 MP4 初始化片段的 URL，以及使用 **GetMP4InitFragment** 動作存取 MP4 媒體片段的 URL。**GetMP4MediaFragmentHLS** 媒體播放列表還包含有關播放器需要播放的流的元數據，例如 **PlaybackMode** 是 **LIVE** 或 **ON\_DEMAND**。對於使用的工作階段，HLS 媒體播放清單通常是靜態 **PlaybackType** 的 **ON\_DEMAND**。HLS 媒體播放清單會持續更新，其中包含新的片段，適用於使用 **PlaybackType** 的 **LIVE** 工作階段。視頻軌道和音軌 (如果適用) 有一個不同的 HLS 媒體播放列表，其中包含特定軌道的 MP4 媒體 URL。
  - **獲取 MP4InitFragment**：檢索 MP4 初始化片段。媒體播放器通常會在載入任何媒體片段之前載入初始化片段。此片段包含「fytpmoov」和「MP4 原子」，以及初始化媒體播放器解碼器所需的子原子。

初始化片段不對應於 Kinesis 視訊串流中的片段。它僅包含流和相應軌道的編解碼器私有數據，媒體播放器需要解碼媒體幀。

- **獲取 MP4MediaFragment**：檢索 MP4 媒體片段。這些片段包含「moofmdat」和「MP4 原子及其子原子」，其中包含編碼片段的媒體框架及其時間戳記。

#### Note

對於 HLS 串流工作階段，支援軌道轉碼器私人資料 (CPD) 變更。在串流工作階段中提供第一個媒體片段之後，片段可以包含每個軌道的 CPD 變更。因此，會話中的片段可以在 CPD 中具有不同的分辨率，比特率或其他信息，而不會中斷播放。不過，在載入這些不同媒體片段時，曲目編號或音軌轉碼器格式所做的任何變更都可能傳回錯誤。例如，如果串流中的片段從只有視訊變更為同時具有音訊和視訊，或是 AAC 音軌變更為 ALAW 音軌，串流就會失敗。對於每個串流工作階段，只允許 500 個 CPD 變更。



使用此動作擷取的資料可計費。如需相關資訊，請參閱 [定價](#)。

- GetTSFragment：檢索包含流中所有軌道的初始化和媒體數據的 MPEG TS 片段。

#### Note

如果ContainerFormat是MPEG\_TS，則使用此 API 代替GetMP4InitFragment和GetMP4MediaFragment檢索流媒體。

使用此動作擷取的資料可計費。如需詳細資訊，請參閱 [Kinesis Video Streams 定價](#)。

串流工作階段 URL 不得在播放器之間共用。如果多個媒體播放器共用工作階段，服務可能會限制工作階段。如需連線限制，請參閱 [Kinesis Video Streams 限制](#)。

您可以透過監控 GetMP4MediaFragment.OutgoingBytes Amazon CloudWatch 指標來監控媒體播放器使用的資料量。如需使用 CloudWatch 監視 Kinesis Video Streams 的相關資訊，請參閱 [監控 Kinesis Video Streams](#)。如需定價資訊，請參閱 [Amazon Kinesis Video Streams 定價和定 AWS 價](#)。HLS 工作階段和傳出 AWS 資料均需支付費用。

如需有關 HLS 的詳細資訊，請參閱 [Apple 開發人員網站](#)上的 [HTTP 即時串流](#)。

#### Important

如果在叫用 Kinesis 影片串流封存媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- x-amz-ErrorTypeHTTP 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- x-amz-RequestIdHTTP 標頭 — 如果您想向其報告問題 AWS，支持團隊可以在給出請求 ID 時更好地診斷問題。

HTTP 狀態碼和 ErrorType 標頭都可用於對錯誤是否可重試以及在什麼情況下進程式設計決定，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及 [常見錯誤](#)。

## 請求語法

```
POST /getHLSStreamingSessionURL HTTP/1.1
Content-type: application/json

{
  "ContainerFormat": "string",
  "DiscontinuityMode": "string",
  "DisplayFragmentTimestamp": "string",
  "Expires": number,
  "HLSFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimestamp": number,
      "StartTimestamp": number
    }
  },
  "MaxMediaPlaylistFragmentResults": number,
  "PlaybackMode": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ContainerFormat

指定應該用於封裝媒體的格式。指定FRAGMENTED\_MP4容器格式會將媒體封裝至 MP4 片段 (fMP4 或 CMAF)。這是建議使用的包裝，因為包裝開銷最小。另一個容器格式選項是MPEG\_TS。HLS 自發布以來一直支持 MPEG TS 塊，有時是舊版 HLS 播放器上唯一支持的包裝。MPEG TS 通常具有 5-25% 的封裝開銷。這意味著 MPEG TS 通常需要比 fMP4 多 5-25% 的頻寬和成本。

預設值為 FRAGMENTED\_MP4。

類型：字串

有效值:FRAGMENTED\_MP4 | MPEG\_TS

必要：否

## DiscontinuityMode

指定何時標記片段之間不連續性的旗標新增至媒體播放清單。

媒體播放程式通常會根據每個片段的時間戳記，建立要播放的媒體內容時間軸。這意味著，如果片段之間存在任何重疊或間隙（通常情況下，如果設置 [HLSFragmentSelector](#) 為 SERVER\_TIMESTAMP），媒體播放器時間軸在某些地方的片段之間也會有很小的間隙，並且會覆蓋其他地方的幀。媒體播放器時間軸中的間隙可能會導致播放停止，而重疊可能會導致播放不安。當片段之間存在不連續標誌時，媒體播放器預期會重置時間軸，從而導致下一個片段在上一個片段之後立即播放。

支援下列模式：

- ALWAYS：在 HLS 媒體播放列表中的每個片段之間放置一個不連續性標記。ALWAYS 如果片段時間戳記不準確，建議使用的值為。
- NEVER：不會將不連續性標記放在任何地方。建議使用的 NEVER 值，以確保媒體播放器時間軸最準確地對應至製作者時間戳記。
- ON\_DISCONTINUITY：不連續性標記放在間隙或重疊超過 50 毫秒的片段之間。對於大多數播放情況，建議使用的值為，ON\_DISCONTINUITY 以便只有在媒體時間軸出現重大問題時才會重設媒體播放器時間軸（例如遺失片段）。

預設值 [HLSFragmentSelector](#) 為 ALWAYS 何時設定為 SERVER\_TIMESTAMP，以及設定為 NEVER 時 PRODUCER\_TIMESTAMP。

類型：字串

有效值: ALWAYS | NEVER | ON\_DISCONTINUITY

必要：否

## DisplayFragmentTimestamp

指定 HLS 媒體播放清單中應包含片段開始時間戳記的時間。媒體播放程式通常會將播放磁頭位置報告為相對於播放工作階段中第一個片段開始的時間。但是，當開始時間戳記包含在 HLS 媒體播放清單中時，某些媒體播放器可能會根據片段時間戳記將目前的播放磁頭報告為絕對時間。這對於建立可向觀眾顯示媒體掛鐘時間的播放體驗非常有用。

預設值為 NEVER。如果 [HLSFragmentSelector](#) 是 SERVER\_TIMESTAMP，時間戳記將是伺服器啟動時間戳記。同樣，如果 [HLSFragmentSelector](#) 是 PRODUCER\_TIMESTAMP，時間戳記將是生產者開始時間戳記。

類型：字串

有效值:ALWAYS | NEVER

必要：否

### Expires

要求的工作階段到期前的時間 (以秒為單位)。此值可以介於 300 (5 分鐘) 到 43200 (12 小時) 之間。

當工作階段到期時，無法對該工作階段GetTSFragment進行GetMP4InitFragmentGetMP4MediaFragment、或的新呼叫。GetHLSMasterPlaylist  
GetHLSMediaPlaylist

預設值為 300 (5 分鐘)。

類型：整數

有效範圍：最小值 300。最大值為 43200。

必要：否

### HLSFragmentSelector

請求片段的時間範圍和時間戳記的來源。

如果是或，則PlaybackMode需要此參ON\_DEMAND數LIVE\_REPLAY。如果是，則此參數  
PlaybackMode 是可選的LIVE。如果PlaybackMode是LIVE，則FragmentSelectorType可  
以設定，但不TimestampRange應設定。如果PlaybackMode是ON\_DEMAND或LIVE\_REPLAY，  
則TimestampRange必須同時FragmentSelectorType設定 AND。

類型：[HLSFragmentSelector](#) 物件

必要：否

### MaxMediaPlaylistFragmentResults

HLS 媒體播放清單中傳回的片段數目上限。

如果PlaybackMode是LIVE，則最新的片段會返回到此值。如果PlaybackMode是ON\_DEMAND，  
則返回最舊的片段，直到此最大數量。

當即時 HLS 媒體播放清單中可用的片段數目較多時，視訊播放程式通常會在開始播放之前緩衝內容。增加緩衝區大小會增加播放延遲，但會降低播放期間發生重新緩衝的可能性。我們建議即時 HLS 媒體播放清單至少有 3 個片段和最多 10 個片段。

如果是LIVE或LIVE\_REPLAY，則預設值為 5 個片段，如果PlaybackMode是，則預設值PlaybackMode為 1,000 ON\_DEMAND。

5,000 個片段的最大值對應於具有 1 秒片段的流上的視頻 80 分鐘以上，以及具有 10 秒片段的流上的視頻超過 13 小時。

類型：Long

有效範圍：最小值為 1。最大值為 5000。

必要：否

## PlaybackMode

是否要擷取即時、即時重播或封存的隨選資料。

三種工作階段類型的功能包括下列項目：

- **LIVE**：對於此類型的工作階段，HLS 媒體播放清單會在可用時不斷更新為最新片段。我們建議媒體播放器以一秒鐘的時間間隔擷取新的播放清單。在媒體播放程式中播放此類工作階段時，使用者介面通常會顯示「即時」通知，而且沒有用於選擇播放視窗中要顯示的位置的定位器控制項。

### Note

在LIVE模式下，最新的可用片段包含在 HLS 媒體播放列表中，即使片段之間存在間隙（也就是說，如果缺少片段）。像這樣的間隙可能會導致媒體播放器停止或導致播放跳躍。在此模式下，如果片段比播放清單中的最新片段舊，則不會將片段新增至 HLS 媒體播放清單。如果在後續片段新增至播放清單後，遺失的片段變為可用，則不會新增較舊的片段，且不會填滿間隙。

- **LIVE\_REPLAY**：對於此類型的工作階段，HLS 媒體播放清單的更新方式與LIVE模式更新的方式類似，不同之處在於它會從指定開始時間包含片段開始。而不是在攝入片段時添加片段，而是在下一個片段的持續時間過去時添加片段。例如，如果工作階段中的片段長度為兩秒鐘，則每兩秒鐘就會將新片段新增至媒體播放清單。此模式對於能夠在偵測到事件時開始播放，並繼續在工作階段建立時尚未擷取的即時串流媒體非常有用。此模式對於串流先前封存的媒體也很有用，而不受ON\_DEMAND模式中 1,000 片段限制的限制。

- **ON\_DEMAND**：對於此類型的工作階段，HLS 媒體播放清單會包含工作階段的所有片段，直到 `MaxMediaPlaylistFragmentResults` 指定的數目為止。每個工作階段只能擷取一次播放清單。在媒體播放程式中播放此類工作階段時，使用者介面通常會顯示定位器控制項，以便在播放視窗中選擇要顯示的位置。

在所有播放模式中，如果 `FragmentSelectorType` 是 `PRODUCER_TIMESTAMP`，且如果有多個片段具有相同的開始時間戳記，則 HLS 媒體播放清單中會包含具有最大片段編號 (即最新片段) 的片段。其他片段不包括在內。具有不同時間戳但持續時間重疊的片段仍包含在 HLS 媒體播放清單中。這可能會導致媒體播放器中出現意外行為。

預設值為 `LIVE`。

類型：字串

有效值：`LIVE` | `LIVE_REPLAY` | `ON_DEMAND`

必要：否

### StreamARN

要擷取 HLS 主播放清單 URL 的資料流的 Amazon 資源名稱 (ARN)。

您必須指定 `StreamName` 或 `StreamARN`。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

要擷取 HLS 主播放清單 URL 的串流名稱。

您必須指定 `StreamName` 或 `StreamARN`。

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

## 回應語法

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

{
  "HLSStreamingSessionURL": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [HLSStreamingSessionURL](#)

媒體播放程式可用來擷取 HLS 主播放清單的 URL (包含工作階段權杖)。

類型：字串

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱[Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

### InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

### InvalidCodecPrivateDataException

視訊串流中至少一個軌道中的編解碼器私人資料對此作業無效。

HTTP 狀態碼：400

### MissingCodecPrivateDataException

在視訊串流的至少一個軌道中找不到轉碼器私人資料。

HTTP 狀態碼：400

### NoDataRetentionException

GetImages要求不保留資料的串流 (也就是說, a DataRetentionInHours 為 0)。

HTTP 狀態碼：400

### NotAuthorizedException

狀態碼：403, 調用者未授權在給定流上執行操作, 或令牌已過期。

狀態碼：

### ResourceNotFoundException

GetImages當 Kinesis 影片串流找不到您指定的串流時, 就會擲回此錯誤。

GetHLSStreamingSessionURL如果在請求的時間範圍內沒有片段PlaybackMode的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話, 或者如果在過去 30 秒內沒有片段PlaybackMode的LIVE流請求具有 a 的會話, 則GetDASHStreamingSessionURL拋出此錯誤。

HTTP 狀態碼：404

### UnsupportedStreamMediaTypeException

媒體類型 (例如 h.264 或 h.265 視訊或 AAC 或 G.711 音訊) 無法從播放工作階段的第一個片段中曲目的編解碼器 ID 來決定。音軌 1 的轉碼器 ID 應該是V\_MPEG/ISO/AVC音軌 2 的編解碼器識別碼 (選擇性) A\_AAC。

HTTP 狀態碼：400

### 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊, 請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)



- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)

## GetImages

服務：Amazon Kinesis Video Streams Archived Media

檢索對應於給定時間範圍，採樣間隔和圖像格式配置的每個時間戳的圖像列表。

### Note

您必須先呼叫 `GetDataEndpoint` API 以取得端點。然後使用 `--end` 點 url 參數將 `GetImages` 請求發送到此端點。

[視頻播放軌道要求](#)。

### 請求語法

```
POST /getImages HTTP/1.1
Content-type: application/json

{
  "EndTimeStamp": number,
  "Format": "string",
  "FormatConfig": {
    "string" : "string"
  },
  "HeightPixels": number,
  "ImageSelectorType": "string",
  "MaxResults": number,
  "NextToken": "string",
  "SamplingInterval": number,
  "StartTimestamp": number,
  "StreamARN": "string",
  "StreamName": "string",
  "WidthPixels": number
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

## [EndTimestamp](#)

要產生的影像範圍的結束時間戳記。如果StartTimeStamp和EndTimeStamp之間的時間範圍超過 300 秒以上StartTimeStamp，您將收到一個IllegalArgumentException。

類型：Timestamp

必要：是

## [Format](#)

將用於編碼圖像的格式。

類型：字串

有效值:JPEG | PNG

必要：是

## [FormatConfig](#)

鍵值對結構的清單，其中包含可在產生影像時套用的額外參數。關FormatConfig鍵是JPEGQuality，表示用於生成圖像的 JPEG 質量鍵。該FormatConfig值接受從 1 到 100 的整數。如果值為 1，則會以較低的品質和最佳壓縮方式產生影像。如果值為 100，則會以最佳品質和較少的壓縮產生影像。如果未提供任何值，JPEGQuality金鑰的預設值將設定為 80。

類型：字串到字串映射

地圖項目：1 個項目的最大數量。

有效金鑰：JPEGQuality

值長度限制：最小長度為 0。長度上限為 256。

值模式：^[a-zA-Z\_0-9]+

必要：否

## [HeightPixels](#)

與WidthPixels參數搭配使用的輸出影像高度。當同時提供HeightPixels和WidthPixels參數時，圖像將被拉伸以適應指定的縱橫比。如果僅提供HeightPixels參數，則將使用其原始縱橫比來計算WidthPixels比例。如果兩個參數都沒有提供，原始圖像大小將被返回。

類型：整數

有效範圍：最小值為 1。最大值為 2160。

必要：否

### ImageSelectorType

用來產生影像的伺服器或製作者時間戳記的來源。

類型：字串

有效值:PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

必要：是

### MaxResults

API 要傳回的影像數目上限。

#### Note

預設限制為每個 API 回應 25 張影像。如果提供MaxResults大於此值，頁面大小將為 25。任何額外的結果將被分頁。

類型：Long

有效範圍：最小值為 1。最大值為 100。

必要：否

### NextToken

指定從何處開始分頁下一組影像的權杖。這是先前截斷的GetImages:NextToken回應。

類型：字串

長度限制：長度下限為 1。長度上限為 4096。

模式：`[a-zA-Z0-9+/\]=\{0,2\}`

必要：否

### SamplingInterval

需要從串流產生影像的時間間隔 (毫秒)。可提供的最小值為 200 ms (每秒 5 張影像)。如果時間戳記範圍小於取樣間隔，則startTimestamp會傳回來自的影像 (如果可用)。

類型：整數

必要：是

### StartTimestamp

應從中生成圖像的起點。這StartTimestamp必須在包含的時間戳記範圍內，才能傳回影像。

類型：Timestamp

必要：是

### StreamARN

要從中擷取映像的資料流的 Amazon 資源名稱 (ARN)。您必須指定StreamName或StreamARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### StreamName

要從中擷取影像的串流名稱。您必須指定StreamName或StreamARN。

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

### WidthPixels

與HeightPixels參數一起使用的輸出影像寬度。當同時提供WidthPixels和HeightPixels參數時，圖像將被拉伸以適應指定的縱橫比。如果僅提供了WidthPixels參數，或者只提HeightPixels供了參數，ValidationException將拋出一個。如果兩個參數都沒有提供，從流的原始圖像大小將被返回。

類型：整數

有效範圍：最小值為 1。最大值為 3840。

必要：否

## 回應語法

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

{
  "Images": [
    {
      "Error": "string",
      "ImageContent": "string",
      "TimeStamp": number
    }
  ],
  "NextToken": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### Images

從視訊串流產生的影像清單。如果給定的時間戳沒有可用的媒體，則NO\_MEDIA錯誤將列在輸出中。如果在產生影像時發生錯誤，MEDIA\_ERROR將會列在輸出中做為遺失影像的原因。

類型：[Image](#) 物件陣列

### NextToken

要求中用來取得更多影像的加密權杖。

類型：字串

長度限制：長度下限為 1。長度上限為 4096。

模式：`[a-zA-Z0-9+/\]{0,2}`

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱 [Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

### InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

### NotAuthorizedException

狀態碼：403，調用者未授權在給定流上執行操作，或令牌已過期。

狀態碼：

### ResourceNotFoundException

GetImages 當 Kinesis 影片串流找不到您指定的串流時，就會擲回此錯誤。

GetHLSStreamingSessionURL 如果在請求的時間範圍內沒有片段PlaybackMode的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話，或者如果在過去 30 秒內沒有片段PlaybackMode的LIVE流請求具有 of 的會話，則GetDASHStreamingSessionURL拋出此錯誤。

HTTP 狀態碼：404

## 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)

- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於紅寶石 V3 的 SDK](#)



## GetMediaForFragmentList

服務：Amazon Kinesis Video Streams Archived Media

從 Amazon Kinesis 視訊串流中的封存資料取得片段清單 (由片段編號指定) 的媒體。

### Note

您必須先呼叫 `GetDataEndpoint` API 以取得端點。然後使用 `--end` 點 url 參數將 `GetMediaForFragmentList` 請求發送到此端點。

如需限制，請參閱 [Kinesis Video Streams 限制](#)。

### Important

如果在叫用 Kinesis 影片串流封存媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- `x-amz-ErrorTypeHTTP` 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- `x-amz-RequestIdHTTP` 標頭 — 如果您想向其報告問題 AWS，支持團隊可以在給出請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可用於對錯誤是否可重試以及在什麼情況下進程式設計決策，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及 [常見錯誤](#)。

### 請求語法

```
POST /getMediaForFragmentList HTTP/1.1
Content-type: application/json
```

```
{
  "Fragments": [ "string" ],
  "StreamARN": "string",
  "StreamName": "string"
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### Fragments

要擷取媒體的片段數目清單。您可以使用擷取這些值[ListFragments](#)。

類型：字串陣列

陣列成員：項目數下限為 1。項目數上限為 1000。

長度限制：長度下限為 1。長度上限為 128。

模式：`^[0-9]+$`

必要：是

### StreamARN

要從中擷取片段媒體的資料流的 Amazon 資源名稱 (ARN)。指定此參數或 `StreamName` 參數。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

要從中擷取片段媒體的資料流名稱。指定此參數或 `StreamARN` 參數。

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

## 回應語法

```
HTTP/1.1 200  
Content-Type: ContentType
```

```
Payload
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

回應會傳回下列 HTTP 標頭。

### ContentType

要求的媒體的內容類型。

長度限制：長度下限為 1。長度上限為 128。

模式：`^[a-zA-Z0-9_\. \-]+$`

回應傳回以下內容作為 HTTP 主體。

### Payload

Kinesis Video Streams 傳回的承載是指定串流中的一系列區塊。如需有關區塊的資訊，請參閱 [PutMedia](#)。Kinesis Video Streams 在 `GetMediaForFragmentList` 呼叫中傳回的區塊也包含下列其他 Matroska (MKV) 標記：

- AW\_ 動態片段-區塊中傳回的片段編號。
- 伺服器端時間戳記-片段的伺服器端時間戳記。
- AW\_ 生產者 \_ 側面時間戳記-片段的生產者端時間戳記。

如果發生異常，將包括以下標籤：

- AW\_ 中國視頻片段 \_ 號碼-拋出異常的片段的編號。
- AW\_ 中國異常 \_ 錯誤代碼-錯誤的整數碼。
- AW\_ 中國例外 \_ 訊息-例外狀況的文字描述。

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱 [Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

### InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

### NotAuthorizedException

狀態碼：403，調用者未授權在給定流上執行操作，或令牌已過期。

狀態碼：4

### ResourceNotFoundException

GetImages 當 Kinesis 影片串流找不到您指定的串流時，就會擲回此錯誤。

GetHLSStreamingSessionURL 如果在請求的時間範圍內沒有片段PlaybackMode的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話，或者如果在過去 30 秒內沒有片段PlaybackMode的LIVE流請求具有 of 的會話，則GetDASHStreamingSessionURL拋出此錯誤。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)

- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

## ListFragments

服務：Amazon Kinesis Video Streams Archived Media

傳回封存資料內指定串流和時間戳記範圍的[Fragment](#)物件清單。

列出片段最終是一致的。這意味著即使生產者收到片段已持續存在的確認，結果也可能不會立即從請求返回。ListFragments但是，通常不到一秒鐘的時間就可以獲得結果。

### Note

您必須先呼叫 GetDataEndpoint API 以取得端點。然後使用 `--end` 點 url 參數將ListFragments請求發送到此端點。

### Important

如果在叫用 Kinesis 影片串流封存媒體 API 之後擲回錯誤，除了 HTTP 狀態碼和回應本文之外，它還包含下列資訊：

- `x-amz-ErrorTypeHTTP` 標頭 — 除了 HTTP 狀態碼提供的內容之外，還包含更具體的錯誤類型。
- `x-amz-RequestIdHTTP` 標頭 — 如果您想向其報告問題AWS，支持團隊可以在給出請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可用於對錯誤是否可重試以及在什麼情況下進程式設計決策，以及提供用戶端程式設計人員可能需要採取哪些動作才能成功再試一次的資訊。如需詳細資訊，請參閱本主題底部的「錯誤」一節，以及[常見錯誤](#)。

## 請求語法

```
POST /listFragments HTTP/1.1
Content-type: application/json

{
  "FragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimestamp": number,
```

```
    "StartTimeStamp": number
  }
},
"MaxResults": number,
"NextToken": "string",
"StreamARN": "string",
"StreamName": "string"
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

## 請求主體

請求接受採用 JSON 格式的下列資料。

## FragmentSelector

描述要傳回之片段範圍的時間戳記範圍和時間戳記來源。

### Note

只有在NextToken未傳入 API 時才需要此功能。

類型：[FragmentSelector](#) 物件

必要：否

## MaxResults

要傳回的片段總數。如果可用的片段總數大於中指定的值max-results，則輸出中會NextToken提供[ListFragments](#)，您可以用來繼續分頁。

預設值為 100。

類型：Long

有效範圍：最小值為 1。最大值為 1000。

必要：否

## NextToken

用以指定分頁開始位置的字符。這是NextToken來自先前截斷的回應的[ListFragments](#)。

類型：字串

長度限制：長度下限為 1。長度上限為 4096。

模式：`[a-zA-Z0-9+/\]{0,2}`

必要：否

## StreamARN

要從中擷取片段清單的資料流的 Amazon 資源名稱 (ARN)。指定此參數或StreamName參數。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

## StreamName

要從中擷取片段清單的資料流名稱。指定此參數或StreamARN參數。

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

## 回應語法

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

{
  "Fragments": [
    {
      "FragmentLengthInMilliseconds": number,
```



```
    "FragmentNumber": "string",
    "FragmentSizeInBytes": number,
    "ProducerTimestamp": number,
    "ServerTimestamp": number
  }
],
"NextToken": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

## Fragments

串流中符合選取器準則的封存 [Fragment](#) 物件清單。結果沒有特定的順序，即使跨頁面也是如此。

如果流中沒有符合選擇條件的片段，則返回一個空列表。

類型：[Fragment](#) 物件陣列

## NextToken

如果傳回的清單遭到截斷，則作業會傳回此 Token，以使用來擷取下一頁結果。這個值是 null 當沒有更多的結果要返回時。

類型：字串

長度限制：長度下限為 1。長度上限為 4096。

模式：`[a-zA-Z0-9+/\ ]+= {0,2}`

## 錯誤

如需所有動作常見錯誤的資訊，請參閱 [常見錯誤](#)。

## ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已超過限制。稍後再嘗試撥打電話。如需有關限制的資訊，請參閱 [Kinesis Video Streams 限制](#)。

HTTP 狀態碼：400

## InvalidArgumentException

指定的參數超出其限制、不受支援或無法使用。

HTTP 狀態碼：400

## NotAuthorizedException

狀態碼：403，調用者未授權在給定流上執行操作，或令牌已過期。

狀態碼：

## ResourceNotFoundException

GetImages當 Kinesis 影片串流找不到您指定的串流時，就會擲回此錯誤。

GetHLSStreamingSessionURL如果在請求的時間範圍內沒有片段PlaybackMode的流請求具有 ON\_DEMAND or LIVE\_REPLAY 的會話，或者如果在過去 30 秒內沒有片段PlaybackMode的LIVE流請求具有 of 的會話，則GetDASHStreamingSessionURL拋出此錯誤。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列介面](#)
- [適用於 .NET 的 AWS 軟體開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 軟體開發套件第 2 版](#)
- [AWS適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 軟體開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 軟體開發套件第 3 版](#)

## Amazon Kinesis Video Streams

Amazon Kinesis 視訊訊號通道支援下列動作：

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

## GetIceServerConfig

服務: Amazon Kinesis Video Signaling Channels

注意：在使用此 API 之前，您必須呼叫 `GetSignalingChannelEndpoint` API 來要求 HTTPS 端點。然後，您可以在 `GetIceServerConfig` API 請求中指定端點和區域。

獲取交互式連接建立 (ICE) 服務器配置信息，包括 URI，用戶名和密碼，可用於配置 WebRTC 技術連接。ICE 元件會使用此組態資訊來設定 WebRTC 連線，包括使用周圍 NAT (TURN) 中繼伺服器的周遊使用繼電器進行驗證。

TURN 是一種用於改善 peer-to-peer 應用程式連接性的協議。TURN 通過提供基於雲的轉送服務，確保即使一個或多個對等端點無法直接連接，也可以建立 peer-to-peer 連接。有關更多信息，請參閱 [用於訪問轉彎服務的 REST API](#)。

如果其中一個對等方無法透過訊號通道建立直接 peer-to-peer 連線，您可以叫用此 API 來建立後援機制。您必須指定信令通道的 Amazon 資源名稱 (ARN)，才能叫用此 API。

### 請求語法

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ClientId": "string",
  "Service": "string",
  "Username": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

#### ChannelARN

信令通道的 ARN，用於設定的對等之間的 peer-to-peer 連線。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：是

### ClientId

檢視器的唯一識別碼。在信號通道中必須是唯一的。

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

### Service

指定所需的服務。目前，TURN是唯一有效的值。

類型：字串

有效值: TURN

必要：否

### Username

要與認證相關聯的選用使用者 ID。

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

### 回應語法

```
HTTP/1.1 200
```

```
Content-type: application/json

{
  "IceServerList": [
    {
      "Password": "string",
      "Ttl": number,
      "Uris": [ "string" ],
      "Username": "string"
    }
  ]
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### [IceServerList](#)

ICE 伺服器資訊物件清單。

類型：[IceServer](#) 物件陣列

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

您的請求被限制，因為您已超過允許的用戶端通話限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### InvalidClientException

指定的用戶端無效。

HTTP 狀態碼：400

NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：4

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

SessionExpiredException

如果用戶端工作階段已過期。用戶端連線後，工作階段的有效期為 45 分鐘。客戶端應重新連接到通道以繼續發送/接收消息。

HTTP 狀態碼：400

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## SendAlexaOfferToMaster

服務: Amazon Kinesis Video Signaling Channels

### Note

在使用此 API 之前，您必須調用 `GetSignalingChannelEndpoint` API 以獲取端點。然後，您可以在 `SendAlexaOfferToMaster` API 請求中指定端點和區域。

此 API 允許您將啟用 WebRTC 的設備與 Alexa 顯示設備連接起來。當調用時，它發送 Alexa 會話描述協議 (SDP) 提供給主對等。只要主機連線到指定的訊號通道，就會立即傳送選件。此 API 從連接的主服務器返回 SDP 答案。如果主機未連接到信令通道，則會發出重新傳遞請求，直到消息過期為止。

### 請求語法

```
POST /v1/send-alex-a-offer-to-master HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "MessagePayload": "string",
  "SenderClientId": "string"
}
```

### URI 請求參數

請求不會使用任何 URI 參數。

### 請求主體

請求接受採用 JSON 格式的下列資料。

### ChannelARN

Alexa 和主對等通訊所使用的信號通道的 Amazon 資源名稱 (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`



必要：是

### MessagePayload

由 64 編碼的 SDP 提供內容。

類型：字串

長度限制：長度下限為 1。最大長度為 1 萬。

模式：[a-zA-Z0-9+/=]+

必要：是

### 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：是

## 回應語法

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

{
  "Answer": "string"
}
```

## 回應元素

如果動作成功，則服務傳回 HTTP 200 回應。

服務會傳回下列 JSON 格式的資料。

### Answer

以 64 編碼的 SDP 回答內容。

類型：字串

長度限制：長度下限為 1。最大長度為 1 萬。

## 錯誤

如需所有動作常見的錯誤相關資訊，請參閱[常見錯誤](#)。

### ClientLimitExceededException

您的請求被限制，因為您已超過允許的用戶端通話限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### NotAuthorizedException

呼叫者沒有執行此操作的授權。

狀態碼：

### ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [AWS 命令列界面](#)
- [適用於 .NET 的 AWS 開發套件](#)
- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)

- [適用於 PHP 的 AWS 開發套件第 3 版](#)
- [適用於 Python 的 AWS 開發套件](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Amazon Kinesis Video Streams

Amazon Kinesis 動影片 WebRTC 儲存支援下列動作：

- [JoinStorageSession](#)

## JoinStorageSession

服務：Amazon Kinesis Video WebRTC Storage

### Note

在使用此 API 之前，您必須呼叫 `GetSignalingChannelEndpoint` API 來要求 WebRTC 端點。然後，您可以在 `JoinStorageSession` API 請求中指定端點和區域。

加入正在進行的單向視頻和/或多路音頻 WebRTC 會話作為用於輸入信道的視頻生產設備。如果該通道沒有現有的工作階段，則需要建立新的串流工作階段，並且必須提供信號通道的 Amazon 資源名稱 (ARN)。

目前，對於該 `SINGLE_MASTER` 類型，視頻生成設備能夠將音頻和視頻媒體吸收到流中。只有視訊製作裝置可以加入工作階段並錄製媒體。

### Important

目前需要 WebRTC 技術攝取的音訊和視訊軌道。

一旦主參與者透過 WebRTC 協商連線，擷取的媒體工作階段將會儲存在 Kinesis 視訊串流中。然後，多個觀眾可以通過我們的播放 API 播放實時媒體。

客戶也可以使用現有的 Kinesis Video Streams 功能，例如 HLS DASH 播放、產生影像等，搭配擷取的 WebRTC 媒體。

### Note

目前不支援 S3 影像交付。

### Note

假設只有一個視訊產生裝置用戶端可以與頻道的工作階段相關聯。如果有多個用戶端將特定頻道的工作階段加入為視訊產生裝置，則會優先使用最新的用戶端要求。

## 其他資訊

- 幂等-這個 API 不是幂等的。
- 重試行為-這被計為一個新的 API 調用。
- 並發呼叫-允許並發呼叫。每次呼叫會傳送一次提議。

## 請求語法

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json

{
  "channelArn": "string"
}
```

## URI 請求參數

請求不會使用任何 URI 參數。

## 請求主體

請求接受採用 JSON 格式的下列資料。

### [channelArn](#)

訊號通道的 Amazon Resource Name (ARN)。

類型：String

模式：`^arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+$`

必要：是

## 回應語法

```
HTTP/1.1 200
```

## 回應元素

如果動作成功，則服務會傳回具空 HTTP 內文的 HTTP 200 回應。

## 錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

### AccessDeniedException

您沒有執行此作業所需的權限。

HTTP 狀態碼：403

### ClientLimitExceededException

Kinesis Video Streams 已限制要求，因為您已經超過允許的用戶端呼叫限制。稍後再嘗試撥打電話。

HTTP 狀態碼：400

### InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

### ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

## 另請參閱

如需在其中一個特定語言 AWS SDK 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2 的开发](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS SDK for PHP](#)
- [AWS 適用於 Python 的 SDK](#)

- [AWS 適用於紅寶石 V3 的 SDK](#)

## 資料類型

Amazon Kinesis 影片串流支援下列資料類型：

- [ChannelInfo](#)
- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)
- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)
- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)
- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

Amazon Kinesis 影片串流媒體支援下列資料類型：

- [StartSelector](#)

Amazon Kinesis 影片串流封存媒體支援下列資料類型：

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Amazon Kinesis 視訊訊號通道支援下列資料類型：

- [IceServer](#)

Amazon Kinesis 影片 WebRTC 儲存支援下列資料類型：

## Amazon Kinesis Video Streams

Amazon Kinesis 影片串流支援下列資料類型：

- [ChannelInfo](#)
- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)



- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)
- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)
- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

## ChannelInfo

服務: Amazon Kinesis Video Streams

封裝信號通道的中繼資料和屬性的結構。

目錄

### ChannelARN

訊號通道的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### ChannelName

訊號通道的名稱。

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

### ChannelStatus

訊號通道的目前狀態。

類型：字串

有效值: `CREATING | ACTIVE | UPDATING | DELETING`

必要：否

### ChannelType

訊號通道的類型。

類型：字串

有效值: SINGLE\_MASTER | FULL\_MESH

必要：否

#### CreationTime

訊號通道建立的時間。

類型：Timestamp

必要：否

#### SingleMasterConfiguration

SINGLE\_MASTER頻道類型的組態設定結構。

類型：[SingleMasterConfiguration](#) 物件

必要：否

#### Version

訊號通道的目前版本。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：[a-zA-Z0-9]+

必要：否

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ChannelNameCondition

服務: Amazon Kinesis Video Streams

ListSignalingChannelsAPI 的可選輸入參數。如果在叫用時指定此參數ListSignalingChannels，API 只會傳回滿足中ChannelNameCondition指定條件的通道。

目錄

### ComparisonOperator

比較運算子。目前，您只能指定BEGINS\_WITH運算子，該運算子會尋找名稱以指定前置字元開頭的信號通道。

類型：字串

有效值: BEGINS\_WITH

必要：否

### ComparisonValue

要比較的值。

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

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## DeletionConfig

服務: Amazon Kinesis Video Streams

從 Edge 代理程式刪除串流連線所需的組態詳細資料。

內容

### DeleteAfterUpload

一旦媒體上傳到 Kinesisboolean Video Stream 雲端，用來指出是否要標示要刪除的媒體。如果有任何刪除組態值設定為 (例如true，或) 的限制已達到時EdgeRetentionInHours，就可以刪除媒體檔案。MaxLocalMediaSizeInMB

由於預設值設定為true，因此請設定上傳程式排程，使媒體檔案在初始上傳到AWS雲端之前不會遭到刪除。

類型：布林值

必要：否

### EdgeRetentionInHours

您想要在 Edge 代理程式上保留串流中的資料的小時數。保留時間的預設值為 720 小時，也就是 30 天。

類型：整數

有效範圍：最小值為 1。最大值為 720。

必要：否

### LocalSizeConfig

刪除邊組態所需的本機大小值。

類型：[LocalSizeConfig](#) 物件

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## EdgeAgentStatus

服務: Amazon Kinesis Video Streams

包含 Edge 代理程式的記錄器和上傳工作之最新狀態詳細資料的物件。使用此資訊可判斷 Edge 代理程式目前的健全狀況。

### 目錄

#### LastRecorderStatus

串流邊緣錄製工作的最新狀態。

類型 : [LastRecorderStatus](#) 物件

必要 : 否

#### LastUploaderStatus

串流邊緣至雲端上傳工作的最新狀態。

類型 : [LastUploaderStatus](#) 物件

必要 : 否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## EdgeConfig

服務: Amazon Kinesis Video Streams

將用來與邊緣代理程式 IoT Greengrass 元件同步處理之串流邊緣設定的說明。Edge 代理程式元件將在您的內部部署的 IoT 中樞裝置設定上執行。

### 目錄

#### HubDeviceArn

「物聯網 ( IoT ) 事情」流的 Arn。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9\_.-]+

必要：是

#### RecorderConfig

錄製程式組態包含本機MediaSourceConfig詳細資料，這些詳細資料用作存取在攝影機上串流的本機媒體檔案的認證。

類型：[RecorderConfig](#) 物件

必要：是

#### DeletionConfig

刪除組態是由用來進行刪除的保留時間 (EdgeRetentionInHours) 和本機大小組態 (LocalSizeConfig) 詳細資料所組成。

類型：[DeletionConfig](#) 物件

必要：否

#### UploaderConfig

上傳程式組態包含用於排程從 Edge 代理程式到 Kinesis 視ScheduleExpression訊串流上傳錄製媒體檔案的工作的詳細資料。

類型：[UploaderConfig](#) 物件



必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ImageGenerationConfiguration

服務: Amazon Kinesis Video Streams

包含 KVS 影像傳送所需資訊的結構。如果為 null，則配置將從流中刪除。

### 目錄

#### DestinationConfig

包含將影像傳送給客戶所需資訊的結構。

類型：[ImageGenerationDestinationConfig](#) 物件

必要：是

#### Format

接受的圖像格式。

類型：字串

有效值: JPEG | PNG

必要：是

#### ImageSelectorType

用來產生影像的伺服器或製作者時間戳記的來源。

類型：字串

有效值: SERVER\_TIMESTAMP | PRODUCER\_TIMESTAMP

必要：是

#### SamplingInterval

需要從串流產生影像的時間間隔 (毫秒)。可以提供的最小值為 200 毫秒。如果時間戳記範圍小於採樣間隔，則來自StartTimestamp如果可用，將退回。

類型：整數

有效範圍：最小值為 3000。最大值為 2 萬。

必要：是

## Status

指示是否ContinuousImageGenerationConfigurationsAPI 已啟用或停用。

類型：字串

有效值: ENABLED | DISABLED

必要：是

## FormatConfig

鍵值對結構的清單，其中包含可在產生影像時套用的額外參數。該FormatConfig關鍵是JPEGQuality，表示要用來產生影像的 JPEG 品質金鑰。該FormatConfig值接受從 1 到 100 的整數。如果值為 1，則會以較低的品質和最佳壓縮方式產生影像。如果值為 100，則會以最佳品質和較少的壓縮產生影像。如果未提供任何值，則為JPEGQuality密鑰將被設置為 80。

類型：字串到字串映射

地圖項目：1 個項目的最大數量。

有效金鑰：JPEGQuality

值長度限制：最小長度為 0。長度上限為 256。

值模式：^[a-zA-Z\_0-9]+

必要：否

## HeightPixels

結合使用的輸出圖像的高度WidthPixels參數。當兩者HeightPixels和WidthPixels提供了參數，圖像將被拉伸以適應指定的縱橫比。如果只有HeightPixels提供了參數，其原始縱橫比將用於計算WidthPixels比率。如果兩個參數都沒有提供，原始圖像大小將被返回。

類型：整數

有效範圍：最小值為 1。最大值為 2160。

必要：否

## WidthPixels

輸出影像的寬度，與HeightPixels參數。當兩者WidthPixels和HeightPixels提供了參數，圖像將被拉伸以適應指定的縱橫比。如果只有WidthPixels提供了參數，其原始縱橫比將用於計算HeightPixels比率。如果兩個參數都沒有提供，原始圖像大小將被返回。

類型：整數

有效範圍：最小值為 1。最大值為 3840。

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ImageGenerationDestinationConfig

服務: Amazon Kinesis Video Streams

包含將影像傳送給客戶所需資訊的結構。

內容

### DestinationRegion

將交付映像的 S3 儲存貯體AWS區域。這DestinationRegion必須與串流所在的區域相符。

類型 : String

長度下限 : 長度下限為 9。長度上限為 14。

模式 : `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

必要 : 是

Uri

識別符 (URI) , 識別符 (URI)。

類型 : String

長度限制 : 長度下限為 1。長度上限為 255。

模式 : `^[a-zA-Z_0-9]+:(//)?(^[^/]+)/?(^[^*]*)$`

必要 : 是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## LastRecorderStatus

服務: Amazon Kinesis Video Streams

串流邊緣錄製工作的最新狀態。

目錄

### JobStatusDetails

記錄器工作最新狀態的描述。

類型：字串

必要：否

### LastCollectedTime

上次執行記錄器工作以及儲存媒體至本機磁碟的時間戳記。

類型：Timestamp

必要：否

### LastUpdatedTime

上次更新記錄器狀態的時間戳記。

類型：Timestamp

必要：否

### RecorderStatus

最新記錄器工作的狀態。

類型：字串

有效值: SUCCESS | USER\_ERROR | SYSTEM\_ERROR

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## LastUploaderStatus

服務: Amazon Kinesis Video Streams

串流邊緣至雲端上傳工作的最新狀態。

目錄

### JobStatusDetails

上傳者工作的最新狀態說明。

類型：字串

必要：否

### LastCollectedTime

上次執行上傳工作以及將媒體收集到雲端的時間戳記。

類型：Timestamp

必要：否

### LastUpdatedTime

上次更新上傳者狀態的時間戳記。

類型：Timestamp

必要：否

### UploaderStatus

最新上傳者工作的狀態。

類型：字串

有效值: SUCCESS | USER\_ERROR | SYSTEM\_ERROR

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：



- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ListEdgeAgentConfigurationsEdgeConfig

服務: Amazon Kinesis Video Streams

單一資料流邊緣設定的描述。

### 目錄

#### CreationTime

流首次創建邊緣配置時的時間戳。

類型：Timestamp

必要：否

#### EdgeConfig

將用來與邊緣代理程式 IoT Greengrass 元件同步處理之串流邊緣設定的說明。Edge 代理程式元件將在您的內部部署的 IoT 中樞裝置設定上執行。

類型：[EdgeConfig](#) 物件

必要：否

#### FailedStatusDetails

產生失敗狀態的描述。

類型：字串

必要：否

#### LastUpdatedTime

流上次更新邊緣配置時的時間戳。

類型：Timestamp

必要：否

#### StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

#### StreamName

串流的名稱。

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

#### SyncStatus

串流邊緣設定的目前同步狀態。

類型：字串

有效值: SYNCING | ACKNOWLEDGED | IN\_SYNC | SYNC\_FAILED | DELETING | DELETE\_FAILED | DELETING\_ACKNOWLEDGED

必要：否

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## LocalSizeConfig

服務: Amazon Kinesis Video Streams

組態詳細資料，包括您想要在 Edge Agent 上儲存串流的媒體大小上限 (MaxLocalMediaSizeInMB)，以及達到串流的大小上限時應使用的策略 (StrategyOnFullSize)。

### 目錄

#### MaxLocalMediaSizeInMB

您想要在 Edge 代理程式上儲存串流的媒體的整體大小上限。

類型：整數

有效範圍：最小值為 64。最大值為 2000。

必要：否

#### StrategyOnFullSize

達到串流MaxLocalMediaSizeInMB限制時執行的策略。

類型：字串

有效值: DELETE\_OLDEST\_MEDIA | DENY\_NEW\_MEDIA

必要：否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## MappedResourceConfigurationListItem

服務: Amazon Kinesis Video Streams

封裝或包含媒體儲存裝置組態屬性的結構。

內容

ARN

Kinesis Data Streams 資源的 Amazon Resource Name (ARN)。

類型 : String

長度限制 : 長度下限為 1。長度上限為 1024。

模式 : `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要 : 否

Type

室壁運動視訊串流的相關資源類型。

類型 : 字串

必要 : 否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## MediaSourceConfig

服務: Amazon Kinesis Video Streams

組態詳細資料，包含存取串流至攝影機之媒體檔案所需的認證 (MediaUriSecretArn和MediaUriType)。

### 目錄

#### MediaUriSecretArn

相機的使用者名稱和密碼或本機媒體檔案位置的 AWS Secrets Manager ARN。

類型：字串

長度限制：長度下限為 20。長度上限為 2048。

模式：arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9\_.-]+

必要：是

#### MediaUriType

統一資源識別碼 (URL) 類別碼 (URL)。該FILE\_URI值可用於流式傳輸本地媒體文件。

#### Note

預覽僅支援RTSP\_URI媒體來源 URI 格式。

類型：字串

有效值: RTSP\_URI | FILE\_URI

必要：是

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)

- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## MediaStorageConfiguration

服務: Amazon Kinesis Video Streams

封裝或包含媒體儲存裝置組態屬性的結構。

- 如StorageStatus果啟用，資料將儲存在StreamARN提供的。為了讓 WebRTC 技術擷取能夠運作，串流必須啟用資料保留。
- 如果StorageStatus停用，則不會儲存任何資料，也不需要StreamARN參數。

### 目錄

#### Status

媒體儲存設定的狀態。

類型：字串

有效值: ENABLED | DISABLED

必要：是

#### StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9\_.-]+/[0-9]+

必要：否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)



- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## NotificationConfiguration

服務: Amazon Kinesis Video Streams

包含 KVS 影像傳送通知資訊的結構。如果此參數為 null，則系統會將設定至串流中刪除。

### 目錄

#### DestinationConfig

傳送通知給客戶所需的目的地資訊。

類型：[NotificationDestinationConfig](#) 物件

必要：是

#### Status

指出通知組態為啟用或停用。

類型：字串

有效值: ENABLED | DISABLED

必要：是

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## NotificationDestinationConfig

服務: Amazon Kinesis Video Streams

包含傳遞通知給客戶所需資訊的結構。

內容

Uri

識別符 (URI)。

類型 : String

長度限制 : 長度下限為 1。長度上限為 255。

模式 : `^[a-zA-Z_0-9]+:(//)?( [/]+ )/?([ ^* ]*)$`

必要 : 是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## RecorderConfig

服務: Amazon Kinesis Video Streams

錄製程式組態包含本機MediaSourceConfig詳細資料，這些詳細資料可用來存取在攝影機上串流的本機媒體檔案。

內容

### MediaSourceConfig

組態詳細資料，包含存取串流至攝影機的媒體檔案所需的認證(MediaUriSecretArn和MediaUriType)。

類型：[MediaSourceConfig](#) 物件

必要：是

### ScheduleConfig

包含指定要從攝影機或本機媒體檔案錄製到 Edge Agent 的排程ScheduleExpression和DurationInMinutes詳細資料的組態。如果未提供ScheduleExpression屬性，則 Edge 代理程式將一律設定為錄製模式。

類型：[ScheduleConfig](#) 物件

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ResourceEndpointListItem

服務: Amazon Kinesis Video Streams

描述 GetSignalingChannelEndpoint API 傳回的端點的物件。

媒體伺服器端點將對應至WEBRTC通訊協定。

### 目錄

#### Protocol

GetSignalingChannelEndpointAPI 所傳回之訊號通道的通訊協定。

類型：字串

有效值: WSS | HTTPS | WEBRTC

必要：否

#### ResourceEndpoint

GetSignalingChannelEndpointAPI 傳回的訊號通道端點。

類型：字串

必要：否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ScheduleConfig

服務: Amazon Kinesis Video Streams

此 API 可讓您指定攝影機或本機媒體檔案應記錄到 Edge Agent 的持續時間。ScheduleConfig 由 ScheduleExpression 和 DurationInMinutes 屬性組成。

如果未 ScheduleConfig 在中提供 RecorderConfig，則 Edge 代理程式將永遠設定為錄製模式。

如果未 ScheduleConfig 在中提供 UploaderConfig，則 Edge 代理程式會定期上傳一次 (每 1 小時)。

內容

### DurationInSeconds

記錄媒體的總持續時間。如果提供了 ScheduleExpression 屬性，那麼該 DurationInSeconds 屬性也應該被指定。

類型：整數

有效範圍：最小值為 60。最大值為 3600。

必要：是

### ScheduleExpression

Quartz cron 運算式可處理排程工作，以便從攝影機或本機媒體檔案記錄到 Edge Agent。如果未 ScheduleExpression 提供給 RecorderConfig，則 Edge 代理程式將永遠設定為錄製模式。

有關 Quartz 的更多信息，請參閱 [Cron 觸發器教程](#) 頁面以了解有效表達式及其用法。

類型：字串

長度限制：長度下限為 11。長度上限為 100。

模式：`[^\n]{11,100}`

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## SingleMasterChannelEndpointConfiguration

服務: Amazon Kinesis Video Streams

包含 SINGLE\_MASTER 通道類型之端點組態的物件。

### 目錄

### Protocols

此屬性用於確定通過此 SINGLE\_MASTER 信令通道的通信的性質。如果 WSS 指定，這個 API 會傳回一個網路通訊端點。如 HTTPS 指定，此 API 會傳回 HTTPS 端點。

類型：字串陣列

陣列成員：項目數下限為 1。項目數上限為 5。

有效值: WSS | HTTPS | WEBRTC

必要：否

### Role

這個屬性是用來判斷此 SINGLE\_MASTER 信令通道中的訊息權限。如果指定，MASTER 則此 API 會傳回用戶端可用來接收來自此訊號通道上任何檢視者的選件，並將答案傳送給任何檢視器的端點。如果 VIEWER 指定，此 API 會傳回用戶端只能用來將選件傳送給此訊號通道上的另一個用 MASTER 用戶端的端點。

類型：字串

有效值: MASTER | VIEWER

必要：否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)





## SingleMasterConfiguration

服務: Amazon Kinesis Video Streams

包含 SINGLE\_MASTER 通道類型組態的結構。

內容

### MessageTtlSeconds

訊號通道在捨棄未傳遞的訊息之前保留這些訊息的時長。

類型：整數

有效範圍：最小值為 5。最大值為 120。

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## StreamInfo

服務: Amazon Kinesis Video Streams

Kinesis 視訊串流的組態資訊串流。

目錄

### CreationTime

指出建立串流的時間戳記。

類型：Timestamp

必要：否

### DataRetentionInHours

串流保留資料的時長 (以小時為單位)。

類型：整數

有效範圍：最小值為 0。

必要：否

### DeviceName

與串流相關聯的裝置名稱。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：[a-zA-Z0-9\_.-]+

必要：否

### KmsKeyId

Kinesis Video Streams 用來加密串流上資料的 AWS Key Management Service (AWS KMS) 金鑰 ID。

類型：字串

長度限制：長度下限為 1。長度上限為 2048。

模式：.+

必要：否

### MediaType

串流的 MediaType。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：`[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

必要：否

### Status

串流的狀態資訊串流。

類型：字串

有效值: CREATING | ACTIVE | UPDATING | DELETING

必要：否

### StreamARN

串流的 Amazon Resource Name (ARN)。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：`arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：否

### StreamName

串流的名稱。

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

## Version

此串流的版本。

類型：字串

長度限制：長度下限為 1。長度上限為 64。

模式：`[a-zA-Z0-9]+`

必要：否

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## StreamNameCondition

服務: Amazon Kinesis Video Streams

指定列出串流時必須滿足才會傳回的條件 (請參閱 `ListStreams` API)。條件具有比較運算和值。目前，您只能指定 `BEGINS_WITH` 運算子，該運算子會尋找名稱以指定前置詞開頭的串流。

### 目錄

#### ComparisonOperator

比較運算子。目前，您只能指定 `BEGINS_WITH` 運算子，該運算子會尋找名稱以指定前置詞開頭的串流。

類型：字串

有效值: `BEGINS_WITH`

必要：否

#### ComparisonValue

要比較的值。

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

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Tag

服務: Amazon Kinesis Video Streams

與指定信令通道相關聯的鍵/值對。

內容

Key

與指定的信令通道相關聯的標籤的索引鍵。

類型：String

長度限制：長度下限為 1。長度上限為 128。

模式：`^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`

必要：是

Value

與指定的信令通道相關聯的標籤值。

類型：String

長度限制：長度下限為 0。長度上限為 256。

模式：`[\p{L}\p{Z}\p{N}_.:/+\\-@]*`

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## UploaderConfig

服務: Amazon Kinesis Video Streams

包含指定要從攝影機或本機媒體檔案錄製到 Edge Agent 的排程ScheduleExpression和DurationInMinutes詳細資料的組態。如果未ScheduleConfig在中提供UploaderConfig，則 Edge 代理程式會定期上傳一次 (每 1 小時)。

內容

### ScheduleConfig

包含指定要從攝影機或本機媒體檔案錄製到 Edge Agent 的排程ScheduleExpression和DurationInMinutes詳細資料的組態。如果未ScheduleConfig在此提供UploaderConfig，則 Edge 代理程式將會定期上傳一次 (每 1 小時)。

類型：[ScheduleConfig](#) 物件

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Amazon Kinesis Video Streams 媒體

Amazon Kinesis Video Streams 媒體支持以下數據類型：

- [StartSelector](#)



## StartSelector

服務: Amazon Kinesis Video Streams Media

識別 Kinesis 視訊串流上您希望 GetMedia API 開始傳回媒體資料的區塊。您有以下選項：

- 選擇最新 (或最舊) 的區塊。
- 識別特定區塊。您可以通過提供片段編號或時間戳記 ( 服務器或生產者 ) 來識別特定塊。
- 每個塊的元數據包括一個繼續令牌作為 Matroska ( MKV ) 標記 ( )。AWS\_KINESISVIDEO\_CONTINUATION\_TOKEN如果您先前的GetMedia請求終止，您可以在下一個GetMedia請求中使用此標籤值。然後，API 開始返回從最後一個 API 結束的位置開始返回塊。

### 目錄

#### StartSelectorType

識別 Kinesis 視訊串流上您要從中開始取得資料的片段。

- 現在-從流上的最新塊開始。
- 最早-從串流上最早的可用區塊開始。
- 片段編號-從特定片段之後的塊開始。您也必須指定 AfterFragmentNumber 參數。
- 生產者時間戳或服務器時間戳-從包含具有指定生產者或服務器時間戳的片段的塊開始。您可以透過新增來指定時間戳記StartTimestamp。
- 繼續令牌-使用指定的繼續令牌讀取。

#### Note

如果您選擇「立即」、「最早」或「繼續\_TOKEN」作為startSelectorType，則不會在中提供任何其他資訊。startSelector

類型：字串

有效值: FRAGMENT\_NUMBER | SERVER\_TIMESTAMP | PRODUCER\_TIMESTAMP | NOW | EARLIEST | CONTINUATION\_TOKEN

必要：是

## AfterFragmentNumber

指定您希望 GetMedia API 開始傳回片段的片段編號。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：`^[0-9]+$`

必要：否

## ContinuationToken

Kinesis Video Streams 在先前回 GetMedia 應中傳回的接續權杖。接著，GetMediaAPI 會從延續權杖所識別的區塊開始。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：`^[a-zA-Z0-9_\.\\-]+$`

必要：否

## StartTimestamp

時間戳記值。如果您選擇生產者時間戳記或伺服器時間戳記作為，則需要此值。startSelectorType 接著，GetMediaAPI 會從包含具有指定時間戳記之片段的區塊開始。

類型：Timestamp

必要：否

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Amazon Kinesis Video Streams 歸檔媒體

Amazon Kinesis Video Streams 存檔媒體支持以下數據類型：

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

## ClipFragmentSelector

服務: Amazon Kinesis Video Streams Archived Media

描述片段範圍的時間戳記範圍和時間戳記來源。

具有重複製作者時間戳記的片段會刪除重複資料。這意味著，如果生產者正在生產具有生產者時間戳的片段流大約等於真實時鐘時間，則剪輯將包含請求的時間戳範圍內的所有片段。如果某些片段在相同的時間範圍內攝入，並且時間點非常不同，則只會返回最早攝入的片段集合。

### 目錄

#### FragmentSelectorType

要使用的時間戳記的來源 (伺服器或生產者)。

類型：字串

有效值: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

必要：是

#### TimestampRange

要傳回的時間戳記範圍。

類型：[ClipTimestampRange](#) 物件

必要：是

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## ClipTimestampRange

服務: Amazon Kinesis Video Streams Archived Media

要傳回片段的時間戳記範圍。

內容

### EndTimeStamp

要求之媒體的時間戳記範圍結束。

此值必須在指定值的 24 小時內 StartTimestamp，且必須晚於 StartTimestamp 值。如果 FragmentSelectorType 請求是 SERVER\_TIMESTAMP，則此值必須是過去。

此值包含在內。會 EndTimeStamp 與片段的 (開始) 時間戳記進行比較。在值之前開始並繼續超過該 EndTimeStamp 值的片段會包含在工作階段中。

類型：Timestamp

必要：是

### StartTimestamp

要傳回片段之時間戳記範圍內的起始時間戳記。

只有正好在或之後開始的片段才 StartTimestamp 會包含在工作階段中。在會話中不包含在之前開始 StartTimestamp 並繼續過去的片段。如果 FragmentSelectorType 是 SERVER\_TIMESTAMP，則 StartTimestamp 必須晚於串流頭。

類型：Timestamp

必要：是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## DASHFragmentSelector

服務: Amazon Kinesis Video Streams Archived Media

包含要求媒體的時間戳記範圍，以及時間戳記的來源。

目錄

### FragmentSelectorType

要求之媒體的時間戳記來源。

當FragmentSelectorType設定為PRODUCER\_TIMESTAMP且 [getDash StreamingSession URL: PlaybackMode](#) 是ON\_DEMAND或時LIVE\_REPLAY，媒體播放清單中會包含以指定[FragmentSelector](#):內的生產者時間戳記擷取TimestampRange的第一個片段。此外，包括緊接在第一個片段之後TimestampRange攝入的生產者時間戳的片段（直到 [GetDash StreamingSession URL : MaxManifestFragmentResultsvalue](#)）。

具有重複製作者時間戳記的片段會刪除重複資料。這意味著，如果生產者正在生產具有生產者時間戳的片段流大約等於真正的時鐘時間，MPEG-DASH 清單將包含請求的時間戳範圍內的所有片段。如果某些片段在相同的時間範圍內攝入，並且時間點非常不同，則只會返回最早攝入的片段集合。

當設定FragmentSelectorType為PRODUCER\_TIMESTAMP且 [GetDash StreamingSession URL: PlaybackMode](#) 是時LIVE，製作者時間戳記會用於 MP4 片段和重複資料刪除。但是 MPEG-DASH 資訊清單中包含基於伺服器時間戳記的最新擷取片段。這意味著，即使過去攝入的片段現在具有帶值的生產者時間戳，它們也不會包含在 HLS 媒體播放列表中。

預設值為 SERVER\_TIMESTAMP。

類型：字串

有效值: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

必要：否

### TimestampRange

要求之媒體的時間戳記範圍的開始與結束。

如果是，則不應該存在PlaybackType此值LIVE。

類型：[DASHTimestampRange](#) 物件

必要：否

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## DASHTimestampRange

服務: Amazon Kinesis Video Streams Archived Media

要求之媒體的時間戳記範圍的開始與結束。

如果是，則不應該存在PlaybackType此值LIVE。

中的值包括DASHTimestampRange在內。正好在開始時間或之後開始的片段會包含在工作階段中。在開始時間之前開始並繼續過去的片段不會包含在工作階段中。

內容

### EndTimeStamp

要求之媒體的時間戳記範圍結束。此值必須在指定值的 24 小時內StartTimestamp，且必須晚於StartTimestamp值。

如果FragmentSelectorType請求是SERVER\_TIMESTAMP，則此值必須是過去。

該EndTimeStamp值對於ON\_DEMAND模式是必需的，但對於LIVE\_REPLAY模式是可選的。如果未設定為LIVE\_REPLAY模式，則工作階段將繼續包含新擷取的片段，直到工作階段過期為止。EndTimeStamp

#### Note

此值包含在內。會EndTimeStamp與片段的 (開始) 時間戳記進行比較。在值之前開始並繼續超過該EndTimeStamp值的片段會包含在工作階段中。

類型：Timestamp

必要：否

### StartTimestamp

要求之媒體的時間戳記範圍開始。

如果指定了DASHTimestampRange值，則需要該StartTimestamp值。

只有正好在或之後開始的片段才StartTimestamp會包含在工作階段中。在會話中不包含在之前開始StartTimestamp並繼續過去的片段。如果FragmentSelectorType是SERVER\_TIMESTAMP，則StartTimestamp必須晚於串流頭。

類型：Timestamp

必要：否

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Fragment

服務：Amazon Kinesis Video Streams Archived Media

代表視訊片段或其他時間分隔資料。

### 目錄

#### FragmentLengthInMilliseconds

播放持續時間或與片段相關聯的其他時間值。

類型：Long

必要：否

#### FragmentNumber

片段的唯一標識符。此值會根據擷取順序單調增加。

類型：字串

長度限制：長度下限為 1。長度上限為 128。

模式：`^[0-9]+$`

必要：否

#### FragmentSizeInBytes

總片段大小，包括有關片段的信息和包含的媒體數據。

類型：Long

必要：否

#### ProducerTimestamp

生產者對應於片段的時間戳記，以毫秒為單位。

類型：Timestamp

必要：否

#### ServerTimestamp

來自AWS伺服器對應於片段的時間戳記，以毫秒為單位。

類型：Timestamp

必要：否

#### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 軟體開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## FragmentSelector

服務: Amazon Kinesis Video Streams Archived Media

描述片段範圍的時間戳記範圍和時間戳記來源。

只有開始時間戳記大於或等於預期時間，且小於或等於結束時間的片段。例如，如果串流包含具有下列開始時間戳記的片段：

- 00:00:00
- 00:02
- 00:04
- 00:06

開始時間為 00:00:01 和結束時間 00:04 的片段選取器範圍會傳回開始時間為 00:02 和 00:00:04 的片段。

### 目錄

#### FragmentSelectorType

要使用的時間戳記的來源 (伺服器或生產者)。

類型：字串

有效值: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

必要：是

#### TimestampRange

要傳回的時間戳記範圍。

類型：[TimestampRange](#) 物件

必要：是

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)

- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## HLSFragmentSelector

服務: Amazon Kinesis Video Streams Archived Media

包含要求媒體的時間戳記範圍，以及時間戳記的來源。

目錄

### FragmentSelectorType

要求之媒體的時間戳記來源。

當FragmentSelectorType設定為PRODUCER\_TIMESTAMP且 [GetHls StreamingSession URL: PlaybackMode](#) 是ON\_DEMAND或時LIVE\_REPLAY，媒體播放清單中會包含以指定[FragmentSelector](#):內的製作者時間戳記擷取的第TimestampRange一個片段。此外，包括緊接在第一個片段之後TimestampRange攝入的帶有生產者時間戳的片段（直到 [GetHls StreamingSession URL : MaxMediaPlaylistFragmentResultsvalue](#)）。

具有重複製作者時間戳記的片段會刪除重複資料。這意味著，如果生產者正在生產具有生產者時間戳的片段流大約等於真實時鐘時間，則 HLS 媒體播放列表將包含請求的時間戳範圍內的所有片段。如果某些片段在相同的時間範圍內攝入，並且時間點非常不同，則只會返回最早攝入的片段集合。

當設置FragmentSelectorType為PRODUCER\_TIMESTAMP和 [GetHls StreamingSession URL : PlaybackMode](#)是時LIVE，生產者時間戳記用於 MP4 片段和重複數據刪除。但是，基於服務器時間戳記的最近攝入的片段包含在 HLS 媒體播放列表中。這意味著，即使過去攝入的片段現在具有帶值的生產者時間戳，它們也不會包含在 HLS 媒體播放列表中。

預設值為 SERVER\_TIMESTAMP。

類型：字串

有效值: PRODUCER\_TIMESTAMP | SERVER\_TIMESTAMP

必要：否

### TimestampRange

要求之媒體的時間戳記範圍的開始與結束。

如果是，則不應該存在PlaybackType此值LIVE。

類型：[HLSTimestampRange](#) 物件

必要：否

## 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)



## HLSTimestampRange

服務: Amazon Kinesis Video Streams Archived Media

要求之媒體的時間戳記範圍的開始與結束。

如果是，則不應該存在PlaybackType此值LIVE。

內容

### EndTimeStamp

要求之媒體的時間戳記範圍結束。此值必須在指定值的 24 小時內StartTimestamp，且必須晚於該StartTimestamp值。

如果FragmentSelectorType請求是SERVER\_TIMESTAMP，則此值必須是過去。

該EndTimeStamp值對於ON\_DEMAND模式是必需的，但對於LIVE\_REPLAY模式是可選的。如果未設定為LIVE\_REPLAY模式，則工作階段將繼續包含新擷取的片段，直到工作階段過期為止。EndTimeStamp

#### Note

此值包含在內。會EndTimeStamp將其與片段的 (開始) 時間戳記進行比較。在值之前開始並繼續超過該EndTimeStamp值的片段會包含在工作階段中。

類型：Timestamp

必要：否

### StartTimestamp

要求之媒體的時間戳記範圍開始。

如果指定了HLSTimestampRange值，則需要該StartTimestamp值。

只有正好在或之後開始的片段才StartTimestamp會包含在工作階段中。在會話中不包含在之前開始StartTimestamp並繼續過去的片段。如果FragmentSelectorType是SERVER\_TIMESTAMP，則StartTimestamp必須晚於串流頭。

類型：Timestamp

必要：否

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Image

服務: Amazon Kinesis Video Streams Archived Media

包含Timestamp、Error和的結構ImageContent。

### 目錄

#### Error

由於不可嘗試的錯誤而未提取提取提供的時間戳記的圖像時顯示的錯誤消息。如果出現以下情況，將返回錯誤：

- 沒有指定的媒體存在Timestamp。
- 指定時間內的媒體不允許擷取影像。在這種情況下，媒體僅是音頻，或者不正確的媒體已被攝入。

類型：字串

有效值: NO\_MEDIA | MEDIA\_ERROR

必要：否

#### ImageContent

以 Base64 編碼之Image物件的屬性。

類型：字串

長度限制：長度下限為 1。長度上限為 6291456。

必要：否

#### TimeStamp

Image物件的屬性，用來從視訊串流中擷取影像。此欄位用於管理影像上的間隙或更好地瞭解分頁視窗。

類型：Timestamp

必要：否

### 另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## TimestampRange

服務: Amazon Kinesis Video Streams Archived Media

要傳回片段的時間戳記範圍。

內容

### EndTimeStamp

要傳回片段的時間戳記範圍內的結束時間戳記。

類型 : Timestamp

必要 : 是

### StartTimeStamp

要傳回片段之時間戳記範圍內的起始時間戳記。

類型 : Timestamp

必要 : 是

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Amazon Kinesis Video Streams

Amazon Kinesis 視訊訊號通道支援下列資料類型：

- [IceServer](#)

## IceServer

服務: Amazon Kinesis Video Signaling Channels

用於 ICE 伺服器連線資料的結構。

內容

Password

登入 ICE 伺服器的密碼。

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

Ttl

以秒為單位的時間量，若使用者名稱和密碼有效。

類型：整數

有效範圍：最小值為 30。最大值為 86400。

必要：否

Uris

URI 陣列，採用 [I-D 中指定的格式](#)。 [petithuguenin-behave-turn-uris](#) 規格。這些 URI 提供了可用於到達轉彎服務器的不同地址和/或協議。

類型：字串陣列

長度限制：長度下限為 1。長度上限為 256。

必要：否

Username

登入 ICE 伺服器的使用者名稱。

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

另請參閱

如需在語言特定的 AWS 開發套件之一中使用此 API 的詳細資訊，請參閱下列說明：

- [適用於 C++ 的 AWS 開發套件](#)
- [適用於 Go 的 AWS 開發套件](#)
- [適用於 Java 的 AWS 開發套件第 2 版](#)
- [適用於 Ruby 的 AWS 開發套件第 3 版](#)

## Amazon Kinesis Video Streams

Amazon Kinesis 影片 WebRTC 儲存支援下列資料類型：

### 常見錯誤

本部分列出所有 AWS 服務 API 動作的常見錯誤。如需此服務之 API 動作的特定錯誤，請參閱該 API 動作的主題。

#### AccessDeniedException

您沒有足夠存取權可執行此動作。

HTTP 狀態碼：400

#### IncompleteSignature

請求簽署不符合 AWS 標準。

HTTP 狀態碼：400

#### InternalFailure

由於不明的錯誤、例外狀況或故障，處理請求失敗。

HTTP 狀態碼：500

## InvalidAction

請求的動作或操作無效。確認已正確輸入動作。

HTTP 狀態碼：400

## InvalidClientId

提供的 X.509 憑證或 AWS 存取金鑰 ID 不存在於我們的記錄中。

HTTP 狀態碼：403

## NotAuthorized

您沒有執行此動作的許可。

HTTP 狀態碼：400

## OptInRequired

AWS 存取金鑰 ID 需要訂閱服務。

HTTP 狀態碼：403

## RequestExpired

請求送達服務已超過戳印日期於請求上之後的 15 分鐘，或者已超過請求過期日期之後的 15 分鐘 (例如預先簽章的 URL)，或者請求上的日期戳印在未來將超過 15 分鐘。

HTTP 狀態碼：400

## ServiceUnavailable

由於伺服器暫時故障，請求失敗。

HTTP 狀態碼：503

## ThrottlingException

由於請求調節，因此請求遭到拒絕。

HTTP 狀態碼：400

## ValidationError

輸入不符合 AWS 服務規定的限制。

HTTP 狀態碼：400



## 常見參數

以下清單內含所有動作用來簽署 Signature 第 4 版請求的參數以及查詢字串。任何專屬於特定動作的參數則列於該動作的主題中。如需簽名版本 4 的詳細資訊，請參閱 IAM 使用者指南中的[簽署 AWS API 請求](#)。

### Action

要執行的動作。

類型：字串

必要：是

### Version

編寫請求所憑藉的 API 版本，以 YYYY-MM-DD 格式表示。

類型：字串

必要：是

### X-Amz-Algorithm

建立請求簽章時所使用的雜湊演算法。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

有效值: AWS4-HMAC-SHA256

必要：有條件

### X-Amz-Credential

憑證範圍值，此為一個字串，其中包含您的存取金鑰、日期、您的目標區域、您請求的服務，以及終止字串（“aws4\_request”）。值以下列格式表示：access\_key/YYYYMMDD/region/service/aws4\_request。

如需詳細資訊，請參閱 IAM 使用者指南中的[建立已簽署的 AWS API 請求](#)。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

## X-Amz-Date

用來建立簽署的日期。格式必須是 ISO 8601 基本格式 (YYYYMMDD'T'HHMMSS'Z')。例如，以下日期時間是有效的 X-Amz-Date 值：20120325T120000Z

條件：對所有請求而言，X-Amz-Date 皆為選用，可用來覆寫用於簽署請求的日期。如果規定日期標頭採用 ISO 8601 基本格式，則不需要 X-Amz-Date。當使用 X-Amz-Date 時，其一律會覆寫日期標頭的值。如需詳細資訊，請參閱 IAM 使用者指南中的 [AWSAPI 請求簽名元素](#)。

類型：字串

必要：有條件

## X-Amz-Security-Token

透過呼叫 AWS Security Token Service (AWS STS) 所取得的臨時安全字符。如需支援 IAM 使用者指南中的臨時安全憑證的服務清單 AWS STS [AWS 服務，請參閱 IAM 使用者指南中的《可搭配 IAM 運作》](#)。

條件：如果您使用 Security Token Token Token.AWS STS

類型：字串

必要：有條件

## X-Amz-Signature

指定從要簽署的字串和衍生的簽署金鑰中計算出的十六進位編碼簽章。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

## X-Amz-SignedHeaders

指定納入作為標準請求一部分的所有 HTTP 標頭。如需有關指定已簽署標頭的詳細資訊，請參閱 IAM 使用者指南中的 [建立已簽署 AWS API 請求](#)。

條件：當您在查詢字串中而非 HTTP 授權標頭中納入驗證資訊時，應指定此參數。

類型：字串

必要：有條件

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。