



開發人員指南

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: 開發人員指南

Copyright © 2025 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 Amazon Kinesis Video Streams ?	1
區域可用性	1
運作方式	2
API 和生產者程式庫	3
資料模型	6
系統要求	12
攝影機需求	13
已測試的作業系統	13
SDK 儲存需求	14
配額	14
控制平面 API 服務配額	14
媒體和封存媒體 API 服務配額	17
片段中繼資料和片段媒體配額	21
串流中繼資料服務配額	23
生產者開發套件配額	24
設定 帳戶	26
註冊 AWS 帳戶	27
建立具有管理存取權的使用者	27
建立 AWS 帳戶 金鑰	28
優勢	29
開始使用	30
建立 Amazon Kinesis 影片串流	30
使用主控台建立影片串流	30
使用 建立影片串流 AWS CLI	31
將資料傳送至 Amazon Kinesis 影片串流	31
建置 SDK 和範例	31
執行範例以將媒體上傳至 Kinesis Video Streams	35
檢閱確認物件	37
使用媒體資料	37
在 主控台中檢視媒體	37
使用 HLS 使用媒體資料	37
上傳至 Kinesis Video Streams	38
Kinesis Video Streams 生產者用戶端	38
Kinesis Video Streams 生產者程式庫	39

了解什麼是生產者程式庫	39
Java	40
程序：使用 Java 生產者 SDK	40
先決條件	40
下載並設定程式碼	41
撰寫並檢查程式碼	42
清除資源	44
執行並驗證程式碼	44
Android	45
程序：使用 Android 生產者 SDK	45
先決條件	46
下載並設定程式碼	49
檢查程式碼	50
執行並驗證程式碼	52
C++	53
物件模型	54
將媒體放入串流	54
回呼界面	54
程序：使用 C++ 生產者 SDK	55
先決條件	55
下載並設定程式碼	56
撰寫並檢查程式碼	59
執行並驗證程式碼	65
使用 SDK 做為 GStreamer 外掛程式	66
使用 C++ 生產者 SDK 做為 Docker 容器中的 GStreamer 外掛程式	66
使用記錄	66
C 生產者	67
物件模型	67
將媒體放入串流	68
程序：使用 C 生產者 SDK	68
先決條件	68
下載程式碼	69
撰寫並檢查程式碼	70
執行並驗證程式碼	72
Raspberry Pi 上的 C++	74
先決條件	74

建立使用者	75
加入您的網路	76
遠端連線	77
設定攝影機	77
安裝軟體先決條件	79
下載並建置 C++ 生產者 SDK	80
串流即時串流	83
播放媒體	91
故障診斷	93
錯誤碼參考	97
PutFrame 回呼傳回的錯誤和狀態碼 - 平台獨立代碼 (PIC)	98
PutFrame 回呼 - C 生產者程式庫傳回的錯誤和狀態碼	135
NAL 適應旗標	142
生產者結構	143
DeviceInfo/DefaultDeviceInfoProvider	143
StorageInfo	143
串流結構	144
StreamDefinition/StreamInfo	145
ClientMetrics	154
StreamMetrics	155
回呼	157
ClientCallbackProvider	157
StreamCallbackProvider	158
ClientCallbacks	158
要重試串流的回呼實作	162
使用串流中繼資料	163
將中繼資料新增至 Kinesis 影片串流	163
影片播放	166
播放需求	166
GetClip	167
GetDASHStreamingSessionURL	168
GetHLSStreamingSessionURL	168
GetImages	170
使用 HLS 播放	170
使用 AWS CLI 擷取 HLS 串流工作階段 URL	170
範例：在 HTML 和 JavaScript 中使用 HLS	174

故障診斷 HLS 問題	178
使用 MPEG-DASH 播放	179
範例：在 HTML 和 JavaScript 使用 MPEG-DASH	180
設定通知	184
管理通知組態	184
UpdateNotificationConfiguration	184
DescribeNotificationConfiguration	185
關於生產者 MKV 標籤	185
生產者 MKV 標籤的語法	185
MKV 標籤限制	185
Amazon SNS 訊息	186
Amazon SNS 主題承載	186
檢視您的 Amazon SNS 訊息	187
從影片串流擷取影像	188
自動化映像產生 (Amazon S3 交付)	188
UpdateImageGenerationConfiguration	188
DescribeImageGenerationConfiguration	190
生產者 MKV 標籤	185
使用 PutEventMetaData 在生產者 SDK 中新增中繼資料標籤	192
限制	192
S3 物件中繼資料	193
Amazon S3 物件路徑 (影像)	193
Amazon S3 URI 建議，以防止限流	194
存取影片分析	195
使用中繼資料	195
使用剖析器程式庫進行串流	196
先決條件	196
下載程式碼	197
檢查程式碼	197
執程式碼	204
監控	204
使用 CloudWatch 監控指標	205
使用 CloudWatch 監控 Amazon Kinesis Video Streams Edge 代理程式	220
使用 CloudTrail 記錄 API 呼叫	225
串流中繼資料限制	229
排程影片錄製和儲存	230

Amazon Kinesis Video Streams Edge Agent API 操作	231
監控 Amazon Kinesis Video Streams Edge 代理程式	231
在非AWS IoT Greengrass 模式下部署	231
安裝相依項目	232
為您的 IP 攝影機 RTSP URLs建立資源	234
建立 IAM 許可政策	235
建立 IAM 角色	237
建立 AWS IoT 角色別名	238
建立 AWS IoT 政策	239
建立 AWS IoT 物件並取得 AWS IoT Core 登入資料	241
建置 Edge 代理程式	243
安裝 CloudWatch 代理程式	253
將 Edge Agent 執行為原生程序	256
部署至 AWS IoT Greengrass	258
建立 Ubuntu 執行個體	258
設定 AWS IoT Greengrass 核心裝置	260
為您的 IP 攝影機 RTSP URLs建立資源	261
將許可新增至 TES 角色	263
安裝 Secret Manager 元件	265
在裝置上部署 Edge Agent	268
安裝 AWS IoT Greengrass 日誌管理員元件	275
常見問答集	279
Amazon Kinesis Video Streams Edge Agent 支援哪些作業系統？	279
Amazon Kinesis Video Streams Edge Agent 是否支援 H.265 媒體？	280
Amazon Kinesis Video Streams Edge Agent 是否可在 AL2 中運作？	280
如何在 AWS IoT 實物或裝置中執行多個串流？	280
傳送StartEdgeConfigurationUpdate後如何編輯？	280
您是否有任何常見的範例ScheduleConfigs？	280
是否有最大串流限制？	281
如何重新啟動發生錯誤的任務？	281
如何監控 Amazon Kinesis Video Streams Edge Agent 的運作狀態？	281
透過 VPC 串流影片	282
其他資訊	282
VPC 端點程序	282
範例	285
範例：將資料傳送至 Kinesis Video Streams	285

範例：從 Kinesis Video Streams 擷取資料	285
範例：播放影片資料	285
先決條件	285
GStreamer 外掛程式 - kvssink	286
下載、建置和設定 GStreamer 元素	287
執行 GStreamer 元素	287
啟動命令	288
在 Docker 容器中執行 GStreamer 元素	289
參數參考	292
PutMedia API	304
下載並設定程式碼	304
撰寫並檢查程式碼	305
執行並驗證程式碼	307
RTSP 和 Docker	308
教學課程影片	309
先決條件	309
建置 Docker 映像	309
執行 RTSP 範例應用程式	310
轉譯器	311
先決條件	312
執行轉譯器範例	312
運作方式	313
API 參考	315
動作	315
Amazon Kinesis Video Streams	316
Amazon Kinesis Video Streams 媒體	438
Amazon Kinesis Video Streams 封存媒體	453
Amazon Kinesis Video Signaling 頻道	498
Amazon Kinesis Video WebRTC 儲存體	506
資料類型	515
Amazon Kinesis Video Streams	517
Amazon Kinesis Video Streams 媒體	555
Amazon Kinesis Video Streams 封存媒體	558
Amazon Kinesis Video Signaling 頻道	575
Amazon Kinesis Video WebRTC 儲存體	577
常見錯誤	577

常見參數	579
安全	581
資料保護	581
什麼是 Kinesis Video Streams 的伺服器端加密？	582
成本、區域和效能考量事項	582
如何開始使用伺服器端加密？	583
建立和使用客戶受管金鑰	583
使用客戶受管金鑰的許可	584
使用 IAM 控制對 Kinesis Video Streams 資源的存取	585
政策語法	586
Kinesis Video Streams 的動作	587
Kinesis Video Streams 的 Amazon Resource Name (ARNs)	587
授予其他 IAM 帳戶對 Kinesis 影片串流的存取權	587
範例政策	591
使用 控制對 Kinesis Video Streams 資源的存取 AWS IoT	593
串流名稱為AWS IoT ThingName	593
AWS IoT CertificateId 做為串流名稱	599
使用 AWS IoT 登入資料串流至硬式編碼的串流名稱	601
合規驗證	602
恢復能力	602
基礎設施安全性	603
安全最佳實務	603
實作最低權限存取	603
使用 IAM 角色	603
使用 CloudTrail 監控 API 呼叫	604
故障診斷	605
一般問題	605
延遲過高	605
API 問題	606
錯誤：「未知的選項」	606
錯誤：「無法判斷要授權的服務/操作名稱」	606
錯誤：「無法將影格放入串流」	607
錯誤：「服務在接收到最終 AckEvent 之前已關閉連線」	607
錯誤：「STATUS_STORE_OUT_OF_MEMORY」	607
錯誤：「憑證應範圍限定為有效區域。」	607
HLS 問題	608

Java 問題	608
啟用 Java 日誌	608
生產者程式庫問題	609
無法編譯生產者 SDK	609
影片串流不會顯示在主控台	610
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」	610
錯誤：「無法將影格提交到 Kinesis Video 用戶端」	610
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	611
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」	611
在 Raspberry Pi 上執行 GStreamer 示範時出現錯誤：「非法指示」	611
攝影機無法載入 Raspberry Pi	612
在 macOS High Sierra 上找不到攝影機	612
在 macOS High Sierra 上編譯時找不到 jni.h 檔案	613
執行 GStreamer 示範應用程式時 Curl 錯誤	613
在 Raspberry Pi 執行時間的時間戳記/範圍聲明	613
在 Raspberry Pi 上的 gst_value_set_fraction_range_full 聲明	613
在 Android 上發生 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) 錯誤	613
已達到片段持續時間上限錯誤	614
使用 IoT 授權時的「傳遞的物件名稱無效」錯誤	614
串流剖析器程式庫問題	614
無法從串流存取單一影格	614
片段解碼錯誤	615
網路問題	615
文件歷史紀錄	616
.....	dcxx

什麼是 Amazon Kinesis Video Streams ？

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

Kinesis Video Streams 不只是影片資料的儲存體。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，或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影片資料，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、空拍機和其他來源。您也可以傳送非影片的時間序列化資料，例如音訊資料、熱影像、深度資料和 RADAR 資料。當即時影片從這些來源串流到 Kinesis 影片串流時，您可以建置應用程式來即時 frame-by-frame 存取資料，以進行低延遲處理。Kinesis Video Streams 與來源無關。您可以使用 [GStreamer 外掛程式 - kvssink](#) 程式庫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。

您也可以設定 Kinesis 影片串流，以在指定的保留期間內長期存放媒體資料。Kinesis Video Streams 會自動儲存此資料並將其靜態加密。此外，Kinesis Video Streams 會根據生產者時間戳記和擷取時間戳記來為儲存的資料建立索引。您可以建置定期批次處理視訊資料的應用程式，也可以建立需要存取不同使用案例歷史資料的應用程式。

您的自訂應用程式，無論是即時或批次導向，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。

區域可用性

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

區域名稱	AWS 區域碼
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

區域名稱	AWS 區域碼
AWS GovCloud (美國東部)	us-gov-east-1
AWS GovCloud (美國西部)	us-gov-west-1
Africa (Cape Town)	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
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

Kinesis Video Streams : 運作方式

主題

- [Kinesis Video Streams API 和生產者程式庫支援](#)
- [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，並永久存放。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 運作方式的概觀。

下圖示範以下元件間的互動：

- 生產者 – 將資料放入 Kinesis 影片串流的任何來源。生產者可以是任何影片產生裝置，例如安全攝影機、體戴式攝影機、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，影片攝影機可以將影片資料推送至一個 Kinesis 影片串流，並將音訊資料推送至另一個 Kinesis 影片串流。

- Kinesis Video Streams 生產者程式庫 – 您可以在裝置上安裝和設定的一組軟體和程式庫。您可以使用這些程式庫，以不同方式安全地連接和可靠地串流影片，包括即時、緩衝幾秒鐘後，或作為 after-the-fact 媒體上傳。
- Kinesis 影片串流 – 可用來傳輸即時影片資料、選擇性地存放，以及提供資料以供即時、批次或一次性使用的資源。在典型組態中，Kinesis 影片串流只有一個生產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 SDKs 建立 Kinesis 影片串流。

多個獨立應用程式可以平行使用 Kinesis 影片串流。

- 取用者 – 從 Kinesis 影片串流取得片段和影格等資料，以進行檢視、處理或分析。這些消費者通常稱為 Kinesis Video Streams 應用程式。您可以在 Kinesis Video Streams 中即時寫入使用和處理資料的應用程式，或在資料儲存後，不需要低延遲處理時，以時間索引的方式進行。您可以建立這些取用者應用程式，以在 Amazon EC2 執行個體上執行。
- [使用剖析器程式庫觀察攝影機的輸出](#) – 讓 Kinesis Video Streams 應用程式以低延遲的方式可靠地從 Kinesis Video Streams 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 API 和生產者程式庫支援

Kinesis Video Streams 提供 APIs，可讓您建立和管理串流，以及讀取或寫入串流的媒體資料。除了管理功能之外，Kinesis Video Streams 主控台也支援即時和 video-on-demand 播放。Kinesis Video

Streams 也提供一組生產者程式庫，您可以在應用程式程式碼中使用，從媒體來源擷取資料並上傳至 Kinesis 影片串流。

主題

- [Kinesis Video Streams API](#)
- [端點探索模式](#)
- [生產者程式庫](#)

Kinesis Video Streams API

Kinesis Video Streams 提供用於建立和管理 Kinesis Video Streams APIs。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：

- 生產者 API – Kinesis Video Streams 提供 PutMedia API，可將媒體資料寫入 Kinesis 影片串流。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](#)。

隨著片段送達，Kinesis Video Streams 會依增加的順序指派唯一的片段編號。它也會儲存每個片段的生產者端和伺服器端時間戳記，做為 Kinesis Video Streams 特定的中繼資料。

- 消費者 APIs – 消費者可以使用下列 APIs 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。當必須傳回尚未封存的較新片段時，會 GetMedia 切換到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，消費者可以選擇將清單分割為子清單，並讓工作者（例如多個 Amazon EC2 執行個體）使用平行擷取片段 `GetMediaFromFragmentList`，並平行處理。

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。收到資料時，Kinesis Video Streams 會將傳入媒體資料儲存為 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 Video Streams 特定的中繼資料；例如，片段編號、伺服器端和生產者端時間戳記

當取用者請求媒體中繼資料時，Kinesis Video Streams 會傳回區塊串流，從您在請求中指定的片段編號開始。

如果您啟用串流的資料持久性，則在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。

端點探索模式

控制平面 REST APIs

若要存取 [Kinesis Video Streams 控制平面 REST APIs](#)，請使用 [Kinesis Video Streams 服務端點](#)。

Data Plane REST APIs

Kinesis Video Streams 使用 [行動架構](#) 建置，以確保更好的擴展和流量隔離屬性。由於每個串流都會映射到區域中的特定儲存格，因此您的應用程式必須使用串流已映射到的正確儲存格特定端點。存取 Data Plane REST APIs 時，您需要自行管理和映射正確的端點。此程序為端點探索模式，如下所述：

1. 端點探索模式會從呼叫其中一個 `GetEndpoints` 動作開始。這些動作屬於控制平面。
 1. 如果您要擷取 [the section called “Amazon Kinesis Video Streams 媒體”](#) 或 [the section called “Amazon Kinesis Video Streams 封存媒體”](#) 服務的端點，請使用 [the section called “GetDataEndpoint”](#)。
 2. 如果您要擷取 [the section called “Amazon Kinesis Video Signaling 頻道”](#)、[the section called “Amazon Kinesis Video WebRTC 儲存體”](#) 或 [Kinesis Video Signaling](#) 的端點，請使用 [the section called “GetSignalingChannelEndpoint”](#)。
2. 快取和重複使用端點。
3. 如果快取的端點無法再運作，請對 `GetEndpoints` 進行新的呼叫以重新整理端點。

生產者程式庫

建立 Kinesis 影片串流後，您可以開始將資料傳送至串流。在您的應用程式程式碼中，您可以使用這些程式庫從媒體來源擷取資料，並上傳至 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 [上傳至 Kinesis Video Streams](#)。

Kinesis Video Streams 資料模型

[上傳至 Kinesis Video Streams](#) 和 [the section called “使用剖析器程式庫進行串流”](#) 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會以 MKV 格式傳送或接收資料。

[上傳至 Kinesis Video Streams](#) 使用 `StreamDefinition` 和 `Frame` 類型來產生 MKV 串流標頭、影格標頭和影格資料。

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++](#) 產生的 MKV 格式化之資料的元件。

主題

- [串流標頭元素](#)
- [串流追蹤資料](#)
- [框架標頭元素](#)
- [MKV 影格資料](#)

串流標頭元素

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 影片串流的名稱。	my-stream
retention_period	Kinesis Video Streams 會保留該串流資料的持續時間，以小時為單位。0 為未保留資料的串流指定。	24
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，使用者定義的 AWS KMS 金鑰會用來加密串流上的資料。如果不存在，資料會由 Kinesis 提供的金鑰 () 加密aws/kinesisvideo。	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/h264

Element	描述	典型值
	影片資料，內容類型必須是 video/h264 。	
max_latency	此值目前未使用，且應設為 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
timecode_scale	<p>指示影格時間戳記所使用的比例。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 文件中的 TimecodeScale。</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 會使用所接收影格的呈現時間戳記 (pts) 和解碼時間戳記 (dts) 值。如果 false，Kinesis Video Streams 會在收到時以系統產生的時間值對影格加上戳記。	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

Element	描述	典型值
fragment_acks	如果 true , Kinesis Video Streams 收到資料時會傳送確認 (ACKs)。您可以使用 KinesisVideoStream FragmentAck 或 KinesisVideoStream ParseFragmentAck 回呼來接收 ACK。	true
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_ANN EXB_NALS 與 NAL_ADAPTATION_ANNEXB_CPD_NALS 。	NAL_ADAPTATION_ANN EXB_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	以 Mbps 為單位的內容頻寬估計值。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

Element	描述	典型值
buffer_duration	內容在生產者上緩衝的持續時間。如果網路延遲低，可以降低此值。如果網路延遲很高，增加此值可防止畫面在傳送之前遭到捨棄，因為配置無法將畫面放入較小的緩衝區。	
replay_duration	如果連線中斷，影片資料串流「復原」的時間量。如果因為連線遺失而導致影格遺失，則此值可以是零。如果耗用的應用程式可以移除備援影格，則可以增加該值。此值應小於緩衝持續時間，否則會使用緩衝持續時間。	
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 CodecID 。	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 C++ 生產者程式庫 中，中的 <code>gMkvTrackVideoBits</code> 陣列 <code>MkvStatics.cpp</code> 包含影格的像素寬度和高度。	
codecPrivateDataSize	<code>codecPrivateData</code> 參數中的資料大小。	

Element	描述	典型值
track_type	串流的軌道類型。	MKV_TRACK_INFO_TYPE_AUDIO 或 MKV_TRACK_INFO_TYPE_VIDEO
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

串流追蹤資料

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 C++ 生產者程式庫 中，MkvStatics.cpp 中的 gMkvTrackVideoBits 陣列包含影格的像素寬度和高度。 MkvStatics.cpp	
cpd_size	codecPrivateData 參數中的資料大小。	
track_type	軌道的類型。例如，您可以針對音訊使用 MKV_TRACK	MKV_TRACK_INFO_TYPE_AUDIO

Element	描述	典型值
	<code>_INFO_TYPE_AUDIO</code> 的列舉值。	

框架標頭元素

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：解碼此影格時的時間戳記。如果先前的影格依賴此影格進行解碼，則此時間戳記可能早於先前影格的時間戳記。這個值相對於該片段的開始。
- 呈現時間戳記：顯示此影格時的時間戳記。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 影格資料

`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 AWS Management Console 中，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。

Amazon Kinesis Video Streams 系統需求

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

主題

- [攝影機需求](#)

- [已測試的作業系統](#)
- [SDK 儲存需求](#)

攝影機需求

用於執行 Kinesis Video Streams 生產者 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。
- 此範例應用程式預設組態為 512 MB。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。

生產者 SDK 支援下列類型的攝影機：

- 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。

已測試的作業系統

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 Pro 筆記型電腦
 - 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 儲存需求

安裝 [上傳至 Kinesis Video Streams](#) 有 170 MB 的最低儲存要求，以及 512 MB 的建議儲存要求。

Amazon Kinesis Video Streams 服務配額

Kinesis Video Streams 具有下列服務配額：

Important

下列服務配額為軟式 **【s】**，可透過提交支援票證進行升級，或硬式 **【h】**，無法增加。您會在下表中看到個別服務配額旁的 **【s】** 和 **【h】**。

控制平面 API 服務配額

下一節說明控制平面 APIs 的服務配額。TPS 代表每秒的交易。

達到帳戶層級或資源層級請求限制時，`ClientLimitExceededException` 會擲出。

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外狀況和備註
CreateStream	50 TPS [s]	在所有支援的區域中，每個帳戶 【s】 有 10,000 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。

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外狀況和備註
		<p> Note</p> <p>此限制可以提高到每個帳戶 100,000 個 (或更多) 串流。登入 AWS Management Console 位於的 https://console.aws.amazon.com/，並請求提高此限制。</p>		
DeleteEdgeConfiguration	10 TPS [h]	N/A	1 TPS [h]	

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外狀況和備註
DeleteStream	50 TPS [h]	N/A	5 TPS [h]	
DescribeEdgeConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeImageGenerationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeMappedResourceConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeNotificationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
DescribeStream	300 TPS [h]	N/A	5 TPS [h]	
GetDataEndpoint	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。
ListEdgeAgentConfigurations	50 TPS [h]	N/A	N/A	
ListStreams	50 TPS [h]	N/A		

API	帳戶限制：請求	帳戶限制：串流	串流層級限制	相關例外狀況和備註
ListTagsForStream	50 TPS [h]	N/A	5 TPS [h]	
StartEdgeConfigurationUpdate	10 TPS [h]	N/A	1 TPS [h]	
TagStream	50 TPS [h]	N/A	5 TPS [h]	
UntagStream	50 TPS [h]	N/A	5 TPS [h]	
UpdateDataRetention	50 TPS [h]	N/A	5 TPS [h]	
UpdateImageGenerationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
UpdateNotificationConfiguration	50 TPS [h]	N/A	5 TPS [h]	
UpdateStream	50 TPS [h]	N/A	5 TPS [h]	

媒體和封存媒體 API 服務配額

下一節說明媒體和封存媒體 APIs 的服務配額。

達到帳戶層級或資源層級請求限制時，ClientLimitExceededException 會擲出。

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	相關例外狀況和備註
PutMedia	5 TPS [h]	1 [h]	每秒 12.5 MB，或每個串流 100 Mbps [s]	<ul style="list-style-type: none"> • 片段持續時間下限：1 秒 (h) • 片段持續時間上限：20 秒 [h] • 片段大小上限：50 MB (h) • 軌道數目上限：3 [s] • 每秒傳送的最大片段數：5 [h] • 片段中繼資料上限：10 個標籤 [h] 	PutMedia 請求正在串流、長時間執行的連線。您不需要為每個資料開啟新的連線，因為您可以在單一持久性連線中傳送多個片段。如果您嘗試多個並行PutMedia連線，Kinesis Video Streams 會使用 <code>ConnectionLimitExceededException</code> 錯誤訊息調節最新的連線。

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外狀況和備註
GetClip	N/A	N/A	100 MB 的大小限制 [h]	片段數量上限：200 [h]	
GetDASHStreamingSessionURL	25 TPS 【h】	N/A	N/A	N/A	
GetHLSStreamingSessionURL	25 TPS 【h】	N/A	N/A	N/A	
GetImages	N/A	N/A	100 MB 【h】	N/A	每個請求的影像數目上限為 100 【h】。

 **Note**
 的最小值 `SamplingInterval` 為 200 毫秒 (ms)，即每秒 5 個影像。

API	串流層級限制	連接層級限制	頻寬限制	片段層級限制	相關例外狀況和備註
GetMedia	5 TPS [h]	3 【h】	25 MB/秒或 200 Mbps (s)	每秒最多傳送 5 個片段 【h】	<p>GetMedia 請求是串流、長時間執行的連線。您不需要為每個資料開啟新的連線，因為您可以在單一持久性連線中傳送多個片段。如果您嘗試三個以上的並行GetMedia連線，Kinesis Video Streams 會使用ConnectionLimitExceededException 錯誤訊息調節最新的連線。</p> <p>如果典型片段約為 5 MB，則此限制表示每個Kinesis 影片串流約 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 style="border: 1px solid #00aaff; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>GetMedia 不會用於 HLS/DASH 播放。</p> </div>
GetMediaForFragmentList	N/A	5 (s)	25 MB/秒或 200 Mbps (s)	片段數量上限：1000 [h]	五個片段型消費應用程式可以同時叫用GetMediaForFragmentList。多出的連接都會遭到拒絕。

影片播放通訊協定 API 服務配額

API	工作階段層級限制	片段層級限制
GetDASHManifestPlaylist	5 TPS [h]	每個播放清單的片段數目上限：5,000 【h】
GetHLSMasterPlaylist	5 TPS [h]	N/A
GetHLSMediaPlaylist	5 TPS [h]	每個播放清單的片段數目上限：5,000 【h】
GetMP4InitFragment	5 TPS [h]	N/A
GetMP4MediaFragment	20 TPS 【h】	N/A
GetTSFragment	20 TPS 【h】	N/A

片段中繼資料和片段媒體配額

[用於存取封存媒體的 Kinesis Video Streams APIs](#) 會根據請求的片段數目而非 API 呼叫數目進行調節。APIs 受片段中繼資料數量和請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套用至每個串流。換句話說，一個串流中片段中繼資料或媒體的請求不適用於另一個串流的配額。不過，在指定的串流中，每個配額會跨多個 APIs 共用。這表示對於指定的串流，不同 APIs 的片段請求會使用相同配額。當超過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 APIs 如何從這兩種類型的配額中取用。對於這些表格中的第二欄，假設如果串流具有 N 的配額，則表示 APIs 從該串流的配額類型中要使用 N 點。GetClip API 會顯示在兩個資料表中。

片段中繼資料配額消耗

API	每個請求消耗的配額點數量	共用配額 (N)
ListFragments	MaxResults 參數的值	每個串流每秒 10,000 個配額【h】
GetClip	產生的剪輯中的片段數量	

API	每個請求消耗的配額點數量	共用配額 (N)
GetHLSMediaPlaylist	MaxMediaPlaylistFragmentResults 參數的值	
GetDASHManifest	MaxManifestFragmentResults 參數的值	
GetImages	值為 400 + 請求的影像數量上限	

片段媒體配額消耗

API	每個請求消耗的配額點數量	共用配額 (N)
GetMediaForFragmentList	Fragments 參數中的片段數量	每個串流每秒 500 個配額【h】
GetClip	產生的剪輯中的片段數量	
GetMP4MediaFragment	1	
GetTSFragment	1	
GetImages	請求的影像數量上限	

例如，配額為每秒 500 個片段媒體時，支援特定串流的下列呼叫模式：

- 每秒 5 個請求至 GetClip，每個剪輯中有 100 個片段。
- 每秒 100 個請求至 GetClip，每個剪輯中有 5 個片段。
- 每秒 2 個請求到 GetClip，每個剪輯有 100 個片段，GetMediaForFragmentList 每個剪輯有 3 個請求到。
- 每秒 400 個請求至 GetMP4MediaFragment，每秒 100 個請求至 GetTSFragment。

這些配額對於每個串流可支援的 HLS 和 MPEG-DASH 工作階段數量具有重要意義。媒體播放器在特定時間可以使用的 HLS 和 DASH 工作階段數量沒有限制。因此，播放應用程式不允許太多工作階段同時使用。下列兩個範例說明如何判斷可支援的並行播放工作階段數量：

範例 1：即時串流

在具有 1 秒持續時間片段的 HLS、音訊和視訊軌，並 `MaxMediaPlaylistFragmentResults` 設為 5 的即時串流案例中，媒體播放器通常會 `GetHLSMediaPlaylist` 每秒對進行兩次呼叫。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。這兩個呼叫各耗用五個片段中繼資料配額點。它也會 `GetMP4MediaFragment` 每秒對進行兩次呼叫：一個呼叫最新的視訊，另一個呼叫對應的音訊。每個呼叫都會耗用單一片段媒體字符，因此總共會耗用兩個字符。

在此案例中，最多可支援 250 個並行播放工作階段。透過 250 個工作階段，此案例每秒耗用 2,500 個片段中繼資料配額（遠低於 10,000 個配額）和每秒 500 個片段媒體配額。

範例 2：隨需播放

在 MPEG-DASH 過去事件的隨需播放案例中，音訊和視訊軌 `MaxManifestFragmentResults` 設定為 1,000，媒體播放器通常會在工作階段開始時呼叫 `GetDASHManifest` 一次（耗用 1,000 個片段中繼資料配額點），並以每秒最多 5 次的 `GetMP4MediaFragment` 速率呼叫（耗用 5 個片段媒體配額點），直到載入所有片段為止。在此案例中，每秒最多可以啟動 10 個新工作階段（右側為每秒 10,000 個片段中繼資料配額），而最多 100 個工作階段可以每秒 5 個速率主動載入片段媒體（右側為每秒 500 個片段媒體配額）。

您可以使用 `ArchivedFragmentsConsumed.Metadata` 和 `ArchivedFragmentsConsumed.Media` 分別監控片段中繼資料和片段媒體配額點的消耗。如需監控的資訊，請參閱 [the section called “監控”](#)。

串流中繼資料服務配額

下列服務配額適用於將串流中繼資料新增至 Kinesis 影片串流：

- 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。

生產者開發套件配額

下表包含 SDK 中值的目前配額。如需詳細資訊，請參閱[上傳至 Kinesis Video Streams](#)。

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以遞迴方式建立串流。
最高裝置名稱長度	128 個字元	
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	

Value	限制	備註
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 Amazon Kinesis Video Streams 服務配額 。
最長區段持續時間	20 秒	如需詳細資訊，請參閱 Amazon Kinesis Video Streams 服務配額 。
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	

Value	限制	備註
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	
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	

設定帳戶

第一次使用 Amazon Kinesis Video Streams 之前，請先完成下列任務。

主題

- [註冊 AWS 帳戶](#)
- [建立具有管理存取權的使用者](#)
- [建立 AWS 帳戶 金鑰](#)

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

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

註冊程序的一部分包括接聽電話或文字訊息，以及在電話鍵盤上輸入驗證碼。

當您註冊時 AWS 帳戶，AWS 帳戶根使用者會建立。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，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 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center 和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶您的電子郵件地址，以帳戶擁有者 [AWS Management Console](#) 身分登入。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)。

如需說明，請參閱《IAM 使用者指南》中的 [為您的 AWS 帳戶根使用者（主控台）啟用虛擬 MFA 裝置](#)。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的 [啟用 AWS IAM Identity Center](#)。

- 在 IAM Identity Center 中，將管理存取權授予使用者。

如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center 《使用者指南》中的[使用預設值設定使用者存取 IAM Identity Center 目錄](#)。

以具有管理存取權的使用者身分登入

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

如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《使用者指南》中的[登入 AWS 存取入口網站](#)。

指派存取權給其他使用者

- 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[建立許可集](#)。

- 將使用者指派至群組，然後對該群組指派單一登入存取權。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[新增群組](#)。

建立 AWS 帳戶 金鑰

您需要 AWS 帳戶 金鑰，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：

- 登入 AWS Management Console 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
- 在導覽列中選擇 Users (使用者)，然後選擇 Administrator (管理員) 使用者。
- 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)。
- 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)。

優勢

使用 Kinesis Video Streams 的優點如下：

- 從數百萬部裝置連線和串流 – 您可以使用 Kinesis Video Streams 從消費者智慧型手機、空拍機和行車記錄器等數百萬部裝置連線和串流視訊、音訊和其他資料。您可以使用 Kinesis Video Streams 生產者程式庫來設定您的裝置並即時可靠地串流，或做為after-the-fact媒體上傳。
- 長期存放、加密和索引資料 – 您可以設定 Kinesis 影片串流，以長期存放自訂保留期間的媒體資料。Kinesis Video Streams 也會根據生產者產生的或服務端時間戳記，在儲存的資料上產生索引。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設施 – Kinesis Video Streams 是無伺服器，因此無需設定或管理基礎設施。您不需要擔心基礎設施的部署、組態或彈性擴展，因為您的資料串流和耗用應用程式的數量會成長和縮減。Kinesis Video Streams 會自動執行管理串流所需的所有管理和維護，因此您可以專注於應用程式，而不是基礎設施。
- 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 來建置在即時資料串流上操作的自訂即時應用程式，並建立在持久資料上操作的批次或一次性應用程式，而不需要嚴格的延遲要求。您可以使用 建置、部署和管理自訂應用程式：開放原始碼 (Apache MXNet、OpenCV)、自家或第三方解決方案 AWS Marketplace，以處理和分析串流。您可以使用 Kinesis Video Streams Get APIs 來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資料持續存在時加密所有資料。Kinesis Video Streams 會對來自裝置的資料串流強制執行 Transport Layer Security (TLS) 型加密，並使用 AWS Key Management Service () 加密所有靜態資料AWS KMS。此外，您可以使用 AWS Identity and Access Management (IAM) 管理對資料的存取。
- 隨需付費 – 如需詳細資訊，請參閱 [AWS 定價計算工具](#)。

Amazon Kinesis Video Streams 入門

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

- 如果您尚未設定，請設定 AWS 帳戶 並建立管理員。
- 建立 Kinesis 視訊串流。
- 從相機傳送資料至 Kinesis 影片串流，並在 主控台中檢視媒體。

如果您是初次使用 Amazon Kinesis Video Streams，建議您[Kinesis Video Streams：運作方式](#)先閱讀。

Note

依照入門範例，您的 不會產生任何費用 AWS 帳戶。如需 區域中的資料成本，請參閱 [Amazon Kinesis Video Streams 定價](#)。

主題

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

建立 Amazon Kinesis 影片串流

本節說明如何建立 Kinesis 影片串流。

本節包含下列程序：

- [the section called “使用主控台建立影片串流”](#)
- [the section called “使用 建立影片串流 AWS CLI”](#)

使用主控台建立影片串流

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. 在建立新的影片串流頁面上，輸入串流名稱的 *YourStreamName*。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 Video Streams 建立串流後，請檢閱 *YourStreamName* 頁面上的詳細資訊。

使用 建立影片串流 AWS CLI

1. 確認您已安裝 AWS CLI 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至您在上一節中建立的 Kinesis 影片串流。本區段使用 [使用 C++ 生產者程式庫](#) 做為 [範例：Kinesis Video Streams 生產者 SDK GStreamer 外掛程式 - kvssink 外掛程式](#)。

若要從各種作業系統上的各種裝置傳送媒體，本教學課程會使用 Kinesis Video Streams C++ 生產者程式庫和 [GStreamer](#)，這是一個開放原始碼媒體架構，可標準化對攝影機和其他媒體來源的存取。

主題

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

建置 SDK 和範例

您可以在電腦或 [中](#)建置 SDK 和範例 AWS Cloud9。請遵循以下適當的程序。

Build on your computer

使用 [讀我檔案](#) 中的指示來建置生產者程式庫和範例應用程式。

其中包含：

- 安裝相依項目
- 複製儲存庫
- 使用 CMake 產生 makefile
- 使用 make 建置二進位檔案

Build in AWS Cloud9

請依照這些程序上傳至 Kinesis Video Streams AWS Cloud9。您不需要將任何內容下載到您的電腦。

1. 在 [AWS Management Console](#) 中，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 - 選取 Ubuntu Server 22.04 LTS。

您可以使用預設選擇保留其他欄位。

3. 建立環境後，請在 Cloud9 IDE 欄中選取開啟。

在畫面的中下區域，您會看到 `Admin:~/environment $`。這是 AWS Cloud9 (Amazon EC2) 終端機。

Note

如果您不小心關閉終端機，請選取視窗、新終端機。

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. 給予指令碼執行許可。

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

- c. 執行指令碼。

```
./resize_volume.sh
```

4. 取得所有軟體的最新資訊，您可以透過進階封裝工具 (APT) 安裝或更新。

此命令不會更新軟體本身，但請確定您的系統知道哪些是最新的可用版本。

```
sudo apt-get update
```

5. 安裝 C++ 生產者 SDK 相依性。

```
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 複製 C++ 生產者 SDK。

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

7. 準備建置目錄。

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

8. 使用 CMake 產生 makefile。

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

預期輸出的結尾如下所示：

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

9. 使用 `make` 編譯 SDK 和範例應用程式，以及建置最終可執行檔。

```
make
```

預期輸出的結尾如下所示：

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

10. 確認已建立範例檔案。列出目前目錄中的檔案：

```
ls
```

確認下列檔案存在：

- `kvs_gstreamer_sample`
 - `libgstkvssink.so`
11. (選用) 您可以將 `GST_PLUGIN_PATH` 環境變數的設定新增至 `shell` 的啟動指令碼。這可確保在新的終端機工作階段期間正確設定 `GST_PLUGIN_PATH`。在中 AWS Cloud9，`shell` 的啟動指令碼為：`~/.bashrc`。

執行下列命令，將命令附加到 Shell 啟動指令碼的結尾。

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

輸入下列內容以執行 `shell` 的啟動指令碼：

```
source ~/.bashrc
```

確認已設定 `GST_PLUGIN_PATH`。

```
echo $GST_PLUGIN_PATH
```

如果您正確設定輸出，您會看到下列輸出。如果輸出為空白，環境變數未正確設定。

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

執行範例以將媒體上傳至 Kinesis Video Streams

範例應用程式不支援 IMDS 登入資料。在終端機中，匯出 IAM 使用者或角色的 AWS 登入資料，以及串流所在的區域。

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

如果您使用的是臨時 AWS 憑證，也請匯出工作階段字符：

```
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
```

影片規格：

- 解析度 - 1280 x 720 像素
- 影格速率 - 每秒 30 個影格
- 持續時間 - 14.0 秒
- 影片編碼 - H.264，軌道 1
- 關鍵影格 - 每 3 秒產生一個片段持續時間（也稱為一組圖片 (GoP) 大小）3 秒，最終片段長度為 2 秒。

使用您先前建立的串流名稱執行下列命令。如果您尚未建立串流，請參閱 [the section called “建立 Amazon Kinesis 影片串流”](#)。

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

Sample video from GStreamer

使用下列命令來產生使用 GStreamer 的影片。

告訴 GStreamer 要在哪裡找到 kvssink GStreamer 外掛程式。在建置目錄中，指定包含 `libgstkvssink.so` 檔案的資料夾路徑。

從您的建置目錄中，執行下列命令：

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

此 GStreamer 管道會產生即時測試影片串流，其標準測試模式每秒執行 10 個影格，解析度為 640x480 像素。新增了重疊，顯示目前的系統時間和日期。然後，影片會編碼為 H.264 格式，而金鑰影格最多每 10 個影格產生一次，導致片段持續時間（也稱為一組圖片 (GoP) 大小）為 1 秒。會 kvssink 採用 H.264 編碼的影片串流，將其封裝為 Matroska (MKV) 容器格式，並將其上傳至 Kinesis 影片串流。

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 管道，請選取終端機視窗，然後按 CTRL+C。

Note

如需使用 GStreamer 外掛程式從攝影機或 USB 攝影機串流影片的詳細資訊，請參閱 [範例：Kinesis Video Streams 生產者 SDK GStreamer 外掛程式 - kvssink](#)。

檢閱確認物件

在上傳期間，Kinesis Video Streams 會將確認物件傳回至執行上傳的用戶端。您應該會在命令輸出中看到這些項目。範例如下所示：

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

如果確認EventType是 PERSISTED，表示 Kinesis Video Streams 已永久儲存和加密此媒體區塊，以供擷取、分析和長期儲存。

如需確認的詳細資訊，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，或使用 Hypertext Live Streaming (HLS) 建立從串流讀取媒體資料的應用程式，以取用媒體資料。

在 主控台中檢視媒體

在另一個瀏覽器索引標籤中，開啟 AWS Management Console。在 Kinesis Video Streams Dashboard 中，選取 [影片串流](#)。

在串流清單中選取串流的名稱。如有必要，請使用搜尋列。

展開媒體播放區段。如果影片仍在上傳，則會顯示。如果上傳已完成，請選取左雙箭頭。

使用 HLS 使用媒體資料

您可以建立用戶端應用程式，使用 HLS 耗用來自 Kinesis 影片串流的資料。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 [影片播放](#)。

上傳至 Kinesis Video Streams

Amazon Kinesis Video Streams 生產者程式庫是 Kinesis Video Streams 生產者開發套件中的一組程式庫。用戶端使用 程式庫和 SDK 建置裝置內應用程式，以安全地連線至 Kinesis Video Streams，並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：

- 即時
- 緩衝幾秒鐘後
- 媒體上傳之後

建立 Kinesis Video Streams 串流後，您可以開始將資料傳送至該串流。您可以使用 SDK 建立應用程式碼，從媒體來源擷取稱為影格的影片資料，並將其上傳至 Kinesis Video Streams。這些應用程式也稱為生產者應用程式。

生產者程式庫包含下列元件：

- [Kinesis Video Streams 生產者用戶端](#)
- [Kinesis Video Streams 生產者程式庫](#)

Kinesis Video Streams 生產者用戶端

Kinesis Video Streams 生產者用戶端包含單一 `KinesisVideoClient` 類別。此類別會管理媒體來源、從來源接收資料，並在資料從媒體來源流向 Kinesis Video Streams 時管理串流生命週期。它還提供了一個 `MediaSource` 介面，用於定義 Kinesis Video Streams 與專屬硬體和軟體之間的互動。

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，您可以選擇將下列項目傳送至 Kinesis Video Streams：

- 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 Video Streams 不會為攝影機等產生媒體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](#)，將資料上傳至 Kinesis Video Streams。

Kinesis Video Streams 生產者用戶端適用於 Java 和 Android 應用程式。如需詳細資訊，請參閱 [使用 Java 生產者程式庫](#) 及 [使用 Android 生產者程式庫](#)。

Kinesis Video Streams 生產者程式庫

Kinesis Video Streams 生產者程式庫包含在 Kinesis Video Streams 生產者用戶端中。此程式庫也可供希望與 Kinesis Video Streams 更深入整合的使用者直接使用。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 Video Streams 生產者程式庫實作狀態機器，以串流至 Kinesis Video Streams。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。

您可以選擇直接使用 Kinesis Video Streams 生產者程式庫，原因如下：

- 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。
- 由於記憶體和處理能力等限制，您想要減少程式碼中的額外負荷，並將其限制為裸露的最低抽象層級。

目前，Kinesis Video Streams 生產者程式庫適用於 Android、C、C++ 和 Java 應用程式。如需詳細資訊，請參閱下列相關主題中支援的語言。

了解什麼是生產者程式庫

[使用 Java 生產者程式庫](#)

[使用 Android 生產者程式庫](#)

[使用 C++ 生產者程式庫](#)

[使用 C 生產者程式庫](#)

[在 Raspberry Pi 上使用 C++ 生產者 SDK](#)

使用 Java 生產者程式庫

您可以使用 Amazon Kinesis Video Streams 提供的 Java 生產者程式庫，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至 Kinesis 影片串流。

執行下列步驟，將程式碼與 Kinesis Video Streams 整合，讓您的應用程式可以開始將資料串流至 Kinesis 影片串流：

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 生產者 SDK

此程序示範如何在 Java 應用程式中使用 Kinesis Video Streams Java 生產者用戶端，將資料傳送至 Kinesis 影片串流。

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：

- [下載並設定程式碼](#)
- [撰寫並檢查程式碼](#)
- [執行並驗證程式碼](#)

先決條件

設定 Java 生產者 SDK 之前，請確定您具備下列先決條件：

- 在範例程式碼中，您可以透過指定您在憑證設定檔檔案中設定的設定檔來提供 AWS 憑證。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[設定開發的 AWS 登入資料和區域適用於 Java 的 AWS SDK](#)。

Note

Java 範例使用 `SystemPropertiesCredentialsProvider` 物件來取得您的登入資料。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 `KinesisVideoProducerJNI` 檔案，其可在 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp> 取得。此檔案的副檔名取決於您的作業系統：
 - Linux 中為 `KinesisVideoProducerJNI.so`
 - macOS 中為 `KinesisVideoProducerJNI.dylib`
 - 適用於 Windows 的 `KinesisVideoProducerJNI.dll`

Note

macOS、Ubuntu、Windows 和 Raspbian 的預先建置程式庫可在 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java> `src/main/resources/lib` 中取得。若是其他環境，則編譯 [C++](#)。

下載並設定 Java 生產者程式庫程式碼

在 Java 生產者程式庫程序的本節中，您可以下載 Java 範例程式碼、將專案匯入您的 Java IDE，以及設定程式庫位置。

如需此範例的先決條件和其他詳細資訊，請參閱[使用 Java 生產者程式庫](#)。

1. 建立目錄，然後從 GitHub 儲存庫複製範例原始碼。

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

2. 開啟您使用的 Java 整合開發環境 (IDE) (如 [Eclipse](#) 或 [JetBrains IntelliJ IDEA](#))，並匯入您下載的 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 範例程式碼使用目前的 AWS 登入資料。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"));
```

如需詳細資訊，請參閱 <https://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/auth/profile/ProfileCredentialsProvider.html> 參考中的適用於 Java 的 AWS SDK ProfileCredentialsProvider。

撰寫並檢查程式碼

在 [Java 生產者程式庫程序](#) 的本節中，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，然後開始將資料傳送至用戶端。

下列各節將詳細說明。

建立 KinesisVideoClient 執行個體

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 = ABCDEFGHIJKLMNOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

建立 MediaSource 執行個體

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

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 影片串流，但您可以使用它來測試程式碼。

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

向用戶端註冊 MediaSource

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，使其可以開始產生資料並將其傳送至用戶端。

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

清除資源

為了避免記憶體洩漏，請執行下列動作，從用戶端取消註冊媒體來源並釋放用戶端。

```
try {
    kinesisVideoClient.unregisterMediaSource(mediaSource);
    kinesisVideoClient.free();
} catch (final KinesisVideoException e) {
    throw new RuntimeException(e);
}
```

如果您使用 `將任何項目新增至快取` [CachedInfoMultiAuthServiceCallbacks](#)，例如：

```
serviceCallbacks.addStreamInfoToCache(streamName, streamInfo);
serviceCallbacks.addStreamingEndpointToCache(streamName, dataEndpoint);
```

完成時清除快取：

```
serviceCallbacks.removeStreamFromCache(streamName);
```

執行並驗證程式碼

若要執行 Java [生產者程式庫的 Java](#) 測試工具，請執行下列動作。

1. 選擇 DemoAppMain。
2. 選擇 Run (執行)、Run 'DemoAppMain (執行「DemoAppMain」)。
3. 將您的登入資料新增至應用程式的 JVM 引數：
 - 對於非暫時 AWS 登入資料：`"-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`。然後，用戶端會將資料傳送至您的 Kinesis 影片串流。

使用 Android 生產者程式庫

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

執行下列步驟，將程式碼與 Kinesis Video Streams 整合，讓您的應用程式可以開始將資料串流至 Kinesis 影片串流：

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 生產者 SDK

此程序示範如何在 Android 應用程式中使用 Kinesis Video Streams Android 生產者用戶端，將資料傳送至 Kinesis 影片串流。

此程序包含以下步驟：

- [the section called “先決條件”](#)
- [the section called “下載並設定程式碼”](#)
- [the section called “檢查程式碼”](#)
- [the section called “執行並驗證程式碼”](#)

先決條件

建議使用 [Android Studio](#) 檢查、編輯和執行應用程式的程式碼。建議使用最新的穩定版本。

在範本程式碼中，您會提供 Amazon Cognito 登入資料。

請依照這些程序來設定 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](#) 並確認區域正確無誤。
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. 完成下列各節：
 - 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：設定安全需求 - 選取適當的選項。
選取下一步。
 - c. 步驟 3：設定註冊體驗 - 選取適當的選項。
選取下一步。
 - d. 步驟 4：設定訊息傳遞 - 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：整合您的應用程式 - 選取適當的選項。
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 - 檢閱先前區段的選擇，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 ID 以供稍後使用。在 `awsconfiguration.json` 檔案中，這是 `CognitoUserPool.Default.PoolId`。
6. 選取應用程式整合索引標籤，然後移至頁面底部。
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 並記下此 ID 以供稍後使用。在 `awsconfiguration.json` 檔案中，這是 `CognitoUserPool.Default.AppClientId`。
8. 顯示用戶端秘密，並記下此秘密以供稍後使用。在 `awsconfiguration.json` 檔案中，這是 `CognitoUserPool.Default.AppClientSecret`。

設定身分集區

設定身分集區

1. 登入 [Amazon Cognito 主控台](#) 並確認區域正確無誤。
2. 在左側導覽中，選擇身分集區。
3. 選擇 建立身分池。
4. 設定身分集區。
 - a. 步驟 1：設定身分集區信任 - 完成下列區段：
 - 使用者存取 - 選取已驗證的存取
 - 已驗證的身分來源 - 選取 Amazon Cognito 使用者集區

選取下一步。
 - b. 步驟 2：設定許可 - 在已驗證角色區段中，完成下列欄位：
 - IAM 角色 - 選取建立新的 IAM 角色
 - IAM 角色名稱 - 輸入名稱並將其記下以供後續步驟使用。

選取下一步。
 - c. 步驟 3：連接身分提供者 - 在使用者集區詳細資訊區段中，完成下列欄位：

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

選取下一步。

- d. 步驟 4：設定屬性 - 在身分集區名稱欄位中輸入名稱。

選取下一步。

- e. 步驟 5：檢閱並建立 - 檢閱每個區段中的選擇，然後選取建立身分集區。

5. 在身分集區頁面上，選取新的身分集區。

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

6. 更新 IAM 角色的許可。
 - a. 登入 AWS Management Console 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
 - b. 在左側導覽中，選擇角色。
 - c. 尋找並選取您在上面建立的角色。

 Note

如有需要，請使用搜尋列。

- d. 選取連接的許可政策。

選擇 Edit (編輯)。

- e. 選取 JSON 索引標籤，並以下列內容取代政策：

JSON

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

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

選取下一步。

- f. 如果尚未選取，請選取此新版本設定為預設值旁的方塊。

選取儲存變更。

下載並設定 Android 生產者程式庫程式碼

在 Android 生產者程式庫程序的本節中，您可以下載 Android 範例程式碼，並在 Android Studio 中開啟專案。

如需此範例的先決條件和其他詳細資訊，請參閱[使用 Android 生產者程式庫](#)。

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

```
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，提供來自 [先決條件](#) 區段中設定身分集區程序的身分集區 ID，並提供您的 AWS 區域 (例如，**us-west-2**)。

在 CognitoUserPool 節點中，提供來自 [先決條件](#) 區段中設定使用者集區程序的應用程式用戶端秘密、應用程式用戶端 ID 和集區 ID，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. 將更新AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java為您的 區域 (在下列範例中，其設定為 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](#)。

檢查程式碼

在 [Android 生產者程式庫程序](#)的本節中，您會檢查範例程式碼。

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，並開始將資料傳送至用戶端。

下列各節將詳細說明。

建立 KinesisVideoClient 的執行個體

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。您會傳入執行個體 `AWSCredentialsProvider`，該執行個體會從您在上一節修改 `awsconfiguration.json` 的檔案讀取 Amazon Cognito 登入資料。

建立 MediaSource 執行個體

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

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();

```

執行並驗證程式碼

若要執行 Android [生產者程式庫的 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. 選擇加號圖示 (+)、Android 應用程式。在 Name (名稱) 欄位中，輸入 **AmazonKinesisVideoDemoApp**。在 Module (模組) 下拉式選單中，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 Create new account (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條件。
- 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代碼，然後選擇確認。選擇確定。
9. 在下一頁中，保留預設值，然後選擇串流。
10. 登入 AWS Management Console，並在美國西部（奧勒岡）區域開啟 [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 影片串流。

使用 C++ 生產者程式庫

您可以使用 Amazon Kinesis Video Streams 提供的 C++ 生產者程式庫來撰寫應用程式程式碼，將媒體資料從裝置傳送至 Kinesis 影片串流。

物件模型

C++ 程式庫提供下列物件來管理將資料傳送至 Kinesis 影片串流：

- **KinesisVideoProducer**：包含媒體來源和 AWS 登入資料的相關資訊，並維護回呼以報告 Kinesis Video Streams 事件。
- **KinesisVideoStream**：代表 Kinesis 影片串流。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++ 程式庫提供的方法（例如 `PutFrame`）將資料放入 `KinesisVideoStream` 物件。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。

回呼界面

此層公開一組回呼界面，使其能與應用程式層溝通。這些回呼界面包含以下項目：

- **服務回呼界面 (`CallbackProvider`)**：程式庫會在建立串流、取得串流描述，以及刪除串流時，叫用透過此界面取得的事件。
- **用戶端就緒狀態或低儲存事件界面 (`ClientCallbackProvider`)**：當用戶端就緒時，或當程式庫偵測到其可能耗盡可用儲存體或記憶體時，程式庫會叫用此界面上的事件。
- **串流事件回呼界面 (`StreamCallbackProvider`)**：當串流事件發生時，程式庫會在此界面上叫用事件，例如串流進入就緒狀態、捨棄影格或串流錯誤。

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，如果您需要自訂聯網邏輯，或想要向使用者介面公開低儲存條件。

如需生產者程式庫中回呼的詳細資訊，請參閱 [生產者開發套件回呼](#)。

程序：使用 C++ 生產者 SDK

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 影片串流。

此程序包含以下步驟：

先決條件

設定 C++ 生產者 SDK 之前，請確定您具備下列先決條件：

- 登入資料：在範本程式碼中，您可以透過指定您在登入資料設定檔檔案中設定的 AWS 設定檔來提供登入資料。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[設定用於開發的 AWS 登入資料和區域](#)。

- 憑證存放區整合：Kinesis Video Streams 生產者程式庫必須與呼叫的服務建立信任。這是透過驗證公有憑證存放區中的憑證授權單位 (CAs) 來完成。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](#)
 - xCode (macOS)/clang/gcc (xcode-select 2347 版)
 - Java 開發套件 (JDK) (適用於 Java JNI 編譯)
 - [Lib-Pkg](#)
- 安裝下列 Ubuntu 的建置相依性：
 - Git: `sudo apt install git`
 - [CMake](#) : `sudo apt install cmake`
 - G++ : `sudo apt install g++`
 - pkg-config : `sudo apt install pkg-config`
 - [OpenJDK](#) : `sudo apt install openjdk-8-jdk`

Note

只有在您要建置 Java 原生界面 (JNI) 時才需要此項目。

- 設定 JAVA_HOME 環境變數：`export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

下載並設定 C++ 生產者程式庫程式碼

如需有關如何下載和設定 C++ 生產者程式庫的資訊，請參閱 [Amazon Kinesis Video Streams CPP Producer、GStreamer Plugin 和 JNI](#)。

如需此範例的先決條件和詳細資訊，請參閱 [the section called “C++”](#)。

CMake 引數

以下是 C++ Producer SDK 特定 CMake 引數的參考表。您也可以將 [標準 CMake 選項](#) 傳遞給 CMake。

Important

這些都是選用的。

包含或排除特定功能的旗標

CMake 引數	Type	預設	說明
BUILD_DEPENDENCIES	Boolean	ON	從來源建置相依性。否則，請使用已安裝在系統上的相依性。如果找不到其中一個必要的相依性，則會傳回錯誤。
BUILD_GSTREAMER_PLUGIN	Boolean	OFF	建置 kvssink GStreamer 外掛程式。

CMake 引數	Type	預設	說明
BUILD_JNI	Boolean	OFF	建置 Java 原生界面 (JNI)，以便能夠從 Java 執行時間環境呼叫此程式碼。
ALIGNED_MEMORY_MODEL	Boolean	OFF	如果記憶體配置應對齊 8 位元組邊界。有些架構不允許未對齊的記憶體存取。
CONSTRAINED_DEVICE	Boolean	OFF	僅限非 Windows。開啟時，將執行緒堆疊大小設定為 0.5 MiB。 Alpine Linux 組建需要。否則，會使用作業系統預設值。
BUILD_STATIC	Boolean	OFF	將程式庫和可執行檔建置為 共用 (OFF) 或 靜態 (ON)。
ADD_MUCLIBC	Boolean	OFF	連結至 uClibc 而非標準 C 程式庫，這是專為內嵌系統設計的小型 C 標準程式庫。
OPEN_SRC_INSTALL_PREFIX	字串	../open-source/local	如果從來源建置，安裝開放原始碼相依性的位置。

跨編譯的旗標

Important

如果您的目標和主機機器 CPU 架構不同，請設定這些屬性。

CMake 引數	Type	預設	說明
BUILD_LOG 4CPLUS_HOST	字串	""	為指定的 CPU 架構建置log4cplus 相依性。如果未設定，log4cplus 會自動偵測並使用主機的 CPU 架構。
BUILD_OPE NSSL_PLATFORM	字串	""	為指定的 CPU 架構建置OpenSSL相依性。如果未設定，OpenSSL會自動偵測並使用主機的 CPU 架構。

與測試相關的旗標

CMake 引數	Type	預設	說明
BUILD_TEST	Boolean	OFF	建置單元和整合測試。若要執行所有測試，./tst/producerTest 請從建置目錄執行。AWS 需要登入資料才能執行測試。
CODE_COVERAGE	Boolean	OFF	僅適用於 GNU/Clang 編譯器。使用 gcov 和產生報告來啟用程式碼涵蓋範圍集合。
COMPILER_WARNINGS	Boolean	OFF	僅適用於 GNU/Clang 編譯器。啟用所有編譯器警告。

CMake 引數	Type	預設	說明
ADDRESS_SANITIZER	Boolean	OFF	僅適用於 GNU/Clang 編譯器。使用 AddressSanitizer 建置。
MEMORY_SANITIZER	Boolean	OFF	僅適用於 GNU/Clang 編譯器。使用 MemorySanitizer 建置。
THREAD_SANITIZER	Boolean	OFF	僅適用於 GNU/Clang 編譯器。使用 ThreadSanitizer 建置。
UNDEFINED_BEHAVIOR_SANITIZER	Boolean	OFF	僅適用於 GNU/Clang 編譯器。使用 UndefinedBehaviorSanitizer 建置。

若要使用這些 CMake 引數，請依照 `cmake ..` 命令，將它們做為以空格分隔的 `-Dkey=value` 配對清單傳遞。例如：

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -
DALIGNED_MEMORY_MODEL=ON
```

CMake 會遵循 `$PATH` 變數來尋找編譯器工具鏈。在執行 CMake 之前，請設定 `CC` 和 `CXX` 環境變數，以明確設定用於跨編譯的工具鏈。

撰寫並檢查程式碼

在本節中 [the section called "C++"](#)，您會檢查 C++ 測試工具中的程式碼 (`tst/ProducerTestFixture.h` 和其他檔案)。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：

- 建立執行個體 `KinesisVideoProducer` 以存取 Kinesis Video Streams。
- 建立 `KinesisVideoStream` 的執行個體。AWS 帳戶 如果同名的串流尚未存在，這會在 中建立 Kinesis 影片串流。
- 當可用時，在 `KinesisVideoStream` 上呼叫每個資料畫面的 `putFrame`，將其傳送到串流。

下列各節提供此編碼模式的詳細資訊。

建立 `KinesisVideoProducer` 的執行個體

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 $3 \text{ (串流)} * 3 \text{ (分鐘)} * 60 \text{ (分鐘內秒)} * 10000 \text{ (kb)} * 1.2 \text{ (重組額度)} = 194.4 \text{ Mb} \sim 200 \text{ Mb}$ 。

- `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，提供登入資料環境變數的 AWS 存取權。
- The AWS 區域 ("us-west-2")。服務端點取決於區域。

建立 KinesisVideoStream 執行個體

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[the section called “資料保護”](#)。
- 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，且應設為 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。

將音軌新增至 Kinesis 影片串流

您可以使用的 `addTrack` 方法，將音軌詳細資訊新增至視訊軌串流定義 `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 (做為音訊的 ID)。這應該是唯一且非零的數值。
- 使用者定義的音軌名稱 (例如，音訊音軌的「音訊」)。
- 此音軌的轉碼器 ID (例如，用於音軌 "A_AAC")。
- 音軌類型 (例如，將 `MKV_TRACK_INFO_TYPE_AUDIO` 的列舉值用於音訊)。

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。您也可以在建立 `KinesisVideoStream` 物件之後傳送轉碼器建立私有資料，同時在 `KinesisVideoStream` 中呼叫開始方法。

將影格放入 Kinesis 影片串流

您可以使用將媒體放入 Kinesis 影片串流 `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++ 生產者範例會傳送測試資料的緩衝區。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。
- 簡報時間戳記。
- 畫面的持續時間 (到 100 ns 單位)。
- 畫面的大小 (位元組)。
- 畫面資料。

如需影格格式的詳細資訊，請參閱 [the section called “資料模型”](#)。

將 KinesisVideoFrame 放入 KinesisVideoStream 的特定軌道

您可以使用 PutFrameHelper 類別將影格資料放入特定軌道。首先，呼叫 getFrameDataBuffer 以取得其中一個預先配置緩衝區的指標，以填入 KinesisVideoFrame 資料。然後，您可以呼叫 putFrameMultiTrack 來傳送 KinesisVideoFrame 以及布林值，以指出影格資料類型。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。此 putFrameMultiTrack 方法使用佇列機制，以確保 MKV 片段維持單調增加的影格時間戳記，且任兩個片段都不會重疊。例如，片段第一個影格的 MKV 時間戳記應一律大於先前片段最後一個影格的 MKV 時間戳記。

PutFrameHelper 有下列欄位：

- 佇列中的音訊影格數目上限。

- 佇列中的影片影格數目上限。
- 要為單一音訊影格配置的大小。
- 要為單一影片影格配置的大小。

存取指標和指標記錄

C++ 生產者 SDK 包含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 Video Streams 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 (用於存放串流資料的記憶體) 的總大小 (位元組)。
- `contentStoreAvailableSize`：內容存放區中可用的記憶體，以位元組為單位。
- `contentStoreAllocatedSize`：內容存放區的配置記憶體。
- `totalContentViewsSize`：用於內容檢視的總記憶體。內容檢視是內容存放區中一系列的資訊索引。

- `totalFrameRate` : 所有作用中串流的每秒畫面彙總數量。
- `totalTransferRate` : 所有串流中每秒傳送的總位元數 (bps)。

由 `getKinesisVideoStreamMetrics` 填滿的 `PStreamMetrics` 物件包含下列資訊：

- `currentViewDuration` : 內容檢視的前端 (編碼影格時) 與目前位置 (將影格資料傳送至 Kinesis Video Streams 時) 之間 100 ns 單位的差異。
- `overallViewDuration` : 內容檢視前端 (編碼影格時) 到尾端 (從記憶體排清影格時, 因為超過內容檢視的總配置空間, 或因為從 Kinesis Video Streams 接收 `PersistedAck` 訊息, 而未知保留的影格) 之間 100 ns 單位的差異。
- `currentViewSize` : 內容檢視從頭部 (當影格編碼時) 到目前位置 (當影格傳送到 Kinesis Video Streams 時) 的大小, 以位元組為單位。
- `overallViewSize` : 內容檢視的總大小 (位元組)。
- `currentFrameRate` : 串流的最新測量速率 (每秒畫面數)。
- `currentTransferRate` : 串流的最新測量速率 (每秒位元組)。

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);
```

執行並驗證程式碼

若要執行和驗證的程式碼 [the section called “C++”](#), 請參閱下列作業系統特定指示:

- [Linux](#)
- [macOS](#)

- [Windows](#)
- [Raspberry Pi 作業系統](#)

您可以在 Amazon CloudWatch 主控台中監看與串流相關聯的指標，例如 `PutMedia.IncomingBytes`，以監控串流上的流量。

使用 C++ 生產者 SDK 做為 GStreamer 外掛程式

[GStreamer](#) 是多個攝影機和視訊來源使用的熱門媒體架構，可透過結合模組化外掛程式來建立自訂媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。

如需使用 C++ 生產者 SDK 做為 GStreamer 外掛程式的詳細資訊，請參閱 [範例：Kinesis Video Streams 生產者 SDK GStreamer 外掛程式 - kvssink](#)。

使用 C++ 生產者 SDK 做為 Docker 容器中的 GStreamer 外掛程式

[GStreamer](#) 是多個攝影機和視訊來源使用的熱門媒體架構，可透過結合模組化外掛程式來建立自訂媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。

此外，使用 [Docker](#) 建立 GStreamer 管道會將 Kinesis Video Streams 的操作環境標準化，以簡化應用程式的建置和執行。

如需在 Docker 容器中使用 C++ 生產者 SDK 做為 GStreamer 外掛程式的詳細資訊，請參閱 [在 Docker 容器中執行 GStreamer 元素](#)。

搭配 C++ 生產者 SDK 使用記錄

您可以在 `kinesis-video-native-build` 資料夾中的 `kvs_log_configuration` 檔案中設定 C++ 生產者 SDK 應用程式的記錄。

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日誌項目寫入日誌檔案，請將檔案中的第一行更新為下列內容：

```
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 生產者程式庫

您可以使用 Amazon Kinesis Video Streams 提供的 C 生產者程式庫來撰寫應用程式程式碼，將媒體資料從裝置傳送至 Kinesis 影片串流。

物件模型

Kinesis Video Streams C 生產者程式庫是以稱為 Platform Independent Codebase (PIC) 的常見元件為基礎，可在 GitHub 上取得，網址為 <https://github.com/aws-labs/amazon-kinesis-video-streams-pic/>。PIC 包含基礎元件的平台獨立商業邏輯。Kinesis Video Streams C 生產者程式庫使用額外的 API 層包裝 PIC，允許案例和平台特定的回呼和事件。Kinesis Video Streams C 生產者程式庫在 PIC 上建置了下列元件：

- 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 數量設定的所需緩衝量，來最佳化內容存放區。
- 串流資訊提供者 – 公開可直接提供給 PIC API 的 `StreamInfo` 結構。有一組供應商，專門針對應用程式類型和常見的串流案例類型。這些包括視訊、音訊和音訊和視訊多軌等供應商。這些案例都有預設值，您可以根據應用程式的需求自訂。
- 回呼提供者 – 公開可直接提供給 PIC API 的 `ClientCallbacks` 結構。這包括一組用於聯網的回呼提供者 (CURL 型 API 回呼)、授權 (AWS 憑證 API)，以及在錯誤回呼上重試串流。回呼

提供者 API 需要設定許多引數，例如 AWS 區域 和 授權資訊。方法是使用 IoT 憑證或使用 AWS AccessKeyId、SecretKey 或 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 – 協助處理多軌案例的音訊和視訊同步。它具有預設行為，您可以自訂此行為來處理應用程式的特定邏輯。它也會簡化 PIC Frame 結構中的影格中繼資料封裝，然後再將其提交至底層 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 – 包含您的裝置相關資訊，並維護回呼以報告 Kinesis Video Streams 事件。
- KinesisVideoStream – 代表影片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 程式庫提供的方法（例如 PutKinesisVideoFrame）將資料放入 KinesisVideoStream 物件。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 生產者 SDK

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 編碼的影片影格傳送至您的 Kinesis 影片串流。

此程序包含以下步驟：

- [下載 C 生產者程式庫程式碼](#)
- [撰寫並檢查程式碼](#)
- [執行並驗證程式碼](#)

先決條件

設定 C 生產者 SDK 之前，請確定您具備下列先決條件：

- 登入資料 – 在範例程式碼中，您可以指定您在登入資料設定檔檔案中設定的設定檔，以提供 AWS 登入資料。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[設定用於開發的 AWS 登入資料和區域](#)。

- 憑證存放區整合 – Kinesis Video Streams 生產者程式庫必須與呼叫的服務建立信任。這是透過驗證公有憑證存放區中的憑證授權機構 (CAs) 來完成。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](#)
 - xCode (macOS)/clang/gcc (xcode-select 2347 版)
 - Java 開發套件 (JDK) (適用於 Java JNI 編譯)
 - [Lib-Pkg](#)
- 安裝下列 Ubuntu 的建置相依性：
 - Git: `sudo apt install git`
 - [CMake](#) : `sudo apt install cmake`
 - G++ : `sudo apt install g++`
 - pkg-config : `sudo apt install pkg-config`
 - OpenJDK : `sudo apt install openjdk-8-jdk`
 - 設定 JAVA_HOME 環境變數 : `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

下載 C 生產者程式庫程式碼

在本節中，您會下載低階程式庫。如需此範例的先決條件和其他詳細資訊，請參閱 [the section called “C++”](#)。

1. 建立目錄，然後從 GitHub 儲存庫複製範例原始碼。

下載程式碼

```
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`。您也必須安裝 `pkg-config`、`CMake` 和建置環境。

如需詳細資訊，請參閱 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git> README.md 中的。

2. 在您選擇的整合開發環境 (IDE) 中開啟程式碼 (例如 [Eclipse](#))。

撰寫並檢查程式碼

在本節 `KvsVideoOnlyStreamingSample.c` 中，您會在 GitHub 上 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c> 儲存庫的 `samples` 資料夾中檢查範例應用程式的程式碼。您在上一個步驟中下載了此程式碼。此範例示範如何使用 C 生產者程式庫，將資料夾內的 H.264 編碼影片影格 `samples/h264SampleFrames` 傳送至 Kinesis 影片串流。

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。
 - 初始化及設定管道的 `KinesisVideoClient` 和 `KinesisVideoStream`、設定回呼、整合案例特定驗證、擷取和提交轉碼器私有資料，以及讓串流成為「就緒」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。
 - 提交影格至 `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 AWS 登入資料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));

```

執行並驗證程式碼

若要執行和驗證的程式碼 [the section called “C++”](#)，請執行下列動作：

1. 執行下列命令，在 [下載的 C 開發套件](#) 中建立 build 目錄，並從 cmake 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` - 是否根據來源的程式庫建置。
- `-DBUILD_TEST=TRUE` - 建置單位和整合測試。確認支援您的裝置可能很有用。

```
./tst/webrtc_client_test
```

- `-DCODE_COVERAGE` - 啟用涵蓋範圍報告。
- `-DCOMPILER_WARNINGS` - 啟用所有編譯器警告。
- `-DADDRESS_SANITIZER` - 使用 AddressSanitizer 建置。
- `-DMEMORY_SANITIZER` - 使用 MemorySanitizer 建置。
- `-DTHREAD_SANITIZER` - 使用 ThreadSanitizer 建置。
- `-DUNDEFINED_BEHAVIOR_SANITIZER` - 使用 UndefinedBehaviorSanitizer 建置。
- `-DALIGNED_MEMORY_MODEL` - 僅針對對齊記憶體模型的裝置建立。預設值為 OFF。

2. 導覽至您剛使用上一個步驟建立的 `build` 目錄，然後執行 `make` 來建置 WebRTC C 開發套件及其提供的範例。

```
make
```

3. 範例應用程式 `kinesis_video_cproducer_video_only_sample` 會將資料夾內的 h.264 編碼影片影格 `samples/h264SampleFrames` 傳送至 Kinesis Video Streams。下列命令會將迴圈中的影片影格 10 秒傳送至 Kinesis Video Streams：

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

如果您想要從另一個資料夾傳送 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
```

4. 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 Amazon CloudWatch 主控台中觀看與串流相關聯的指標，例如 `PutMedia.IncomingBytes`，來監控串流上的流量。

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 上使用 C++ 生產者 SDK

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。本教學課程說明如何在 Raspberry Pi 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

- [先決條件](#)
- [建立具有寫入 Kinesis Video Streams 許可的 IAM 使用者](#)
- [將 Raspberry Pi 加入 Wi-Fi 網路](#)
- [遠端連線至 Raspberry Pi](#)
- [設定 Raspberry Pi 攝影機](#)
- [安裝軟體先決條件](#)
- [下載並建置 Kinesis Video Streams C++ 生產者 SDK](#)
- [將影片串流到您的 Kinesis 影片串流](#)
- [從 Kinesis 影片串流播放媒體](#)
- [針對適用於 Raspberry Pi 的 C++ 生產者 SDK 的建置問題進行故障診斷](#)

先決條件

在 Raspberry Pi 上設定 C++ 生產者 SDK 之前，請確定您有下列先決條件：

- 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](#) 或連接的 USB 攝影機（網路攝影機）。
 - 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)。您可以從 Raspberry Pi [網站下載最新的 Raspberry Pi OS](#) (先前稱為 Raspbian) 映像。遵循 Raspberry Pi 指示 [將下載的映像安裝在 SD 卡上](#)。
- AWS 帳戶 具有 Kinesis 影片串流的 。如需更多資訊，請參閱 [Kinesis 視訊串流入門](#)。

建立具有寫入 Kinesis Video Streams 許可的 IAM 使用者

如果您尚未這麼做，請設定具有寫入 Kinesis 影片串流許可的 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 存取金鑰對。裝置可以使用 X.509 憑證來連線 AWS IoT。 [the section called “使用 控制對 Kinesis Video Streams 資源的存取 AWS IoT”](#) 如需如何設定裝置以使用憑證型身分驗證的詳細資訊，請參閱。

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. 選擇下一步：許可。
7. 在 Set permissions for kinesis-video-raspberry-pi-producer (設定 kinesis-video-raspberry-pi-producer 許可) 下，選擇 Attach existing policies directly (直接貼附現有的政策)。
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。
9. 選擇 JSON 標籤。
10. 將下列 JSON 政策複製並貼入文字區域。此政策提供使用者建立資料並將其寫入 Kinesis 影片串流的許可。

JSON

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

```
"kinesisvideo:DescribeStream",
"kinesisvideo:CreateStream",
"kinesisvideo:GetDataEndpoint",
"kinesisvideo:PutMedia"
],
"Resource": [
  "*"
]
}]
}
```

11. 選擇檢閱政策。
12. 為您的政策提供一個名稱，例如 **kinesis-video-stream-write-policy**。
13. 選擇建立政策。
14. 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)。
15. 在搜尋方塊中，輸入您建立之政策的名稱。
16. 在清單中，選取新政策旁邊的核取方塊。
17. 選擇 Next:Review (下一步：檢閱)。
18. 選擇 Create user (建立使用者)。
19. 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。

將 Raspberry Pi 加入 Wi-Fi 網路

如果您使用連接的螢幕和鍵盤，請前往 [設定 Raspberry Pi 攝影機](#)。

這些指示旨在協助您在無周邊模式下執行時設定 Raspberry Pi，也就是不使用連接的鍵盤、監視器或網路纜線。請依照下列指示設定 Raspberry Pi，以自動嘗試連線到指定的網路，讓您的主機機器能夠 SSH 連入其中。

1. 在您的電腦上，建立名為 `wpa_supplicant.conf` 的檔案。
2. 複製下列文字並將其貼入 `wpa_supplicant.conf` 檔案：

```
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。

遠端連線至 Raspberry Pi

如果您使用連接的螢幕和鍵盤，請前往 [設定 Raspberry Pi 攝影機](#)。

這些指示旨在協助您在無頭模式下執行時設定 Raspberry Pi，也就是不使用連接的鍵盤、監視器或網路纜線。請依照下列指示，在網路上找到您的 Raspberry Pi，並從主機機器將 SSH 放入其中。

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](#)。

設定 Raspberry Pi 攝影機

請依照下列步驟設定 [Raspberry Pi 攝影機模組](#)，將影片從裝置傳送至 Kinesis 影片串流。

Note

如果您使用的是 USB 網路攝影機，請跳至 [the section called “安裝軟體先決條件”](#)。

Camera module 1

請依照這些指示更新模組檔案、啟用相機界面，以及驗證相機的功能。更新模組檔案會告知 Raspberry Pi 在開機時要載入哪些核心模組。預設不會載入攝影機驅動程式，以便在未使用攝影機的 Raspberry Pi 裝置上節省系統資源。

1. 開啟編輯器以變更模組檔案。開啟終端機，並使用下列命令，使用 nano 編輯器編輯檔案：

```
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，然後退出編輯器。若要使用 nano 編輯器儲存和結束，請使用 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、舊版攝影機。在 Raspbian 作業系統的較舊組建中，此功能表選項可能位於介面選項相機下。

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
```

如果您的攝影機設定正確，此命令會從攝影機擷取影像，並將其儲存至名為 test.jpg 的檔案，並顯示資訊性訊息。

Camera module 2 or 3

如果您使用的是攝影機模組 2，您可以使用 `bcm2835-v4l2` (舊版) 或 `libcamera` (現代)。不過，建議使用 `libcamera` 堆疊以獲得更好的支援和功能。請依照下列步驟，確認在您的系統上 `libcamera` 是 up-to-date。

1. [libcamera](#) 應該預先安裝在 Raspberry Pi 上。檢查是否有任何更新，並更新至最新版本，以取得錯誤修正和安全性更新。開啟終端機並輸入下列命令：

```
sudo apt-get update
sudo apt-get upgrade
```

2. 重新啟動您的系統，讓更新生效。

```
sudo reboot
```

3. 測試您的相機。此應用程式會啟動攝影機預覽串流，並在螢幕上顯示。

```
libcamera-hello
```

如果您的相機模組發生問題，請參閱 [Raspberry Pi 文件](#) 進行故障診斷。

安裝軟體先決條件

C++ 生產者 SDK 需要您在 Raspberry Pi 上安裝下列軟體先決條件。

1. 更新套件清單並安裝建置 SDK 所需的程式庫。開啟終端機並輸入下列命令：

```
sudo apt-get update
sudo apt-get 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. 如果您使用的是 libcamera 堆疊，也請安裝 libcamerasrc GStreamer 外掛程式。預設不會安裝此 GStreamer 外掛程式。

```
sudo apt-get install gstreamer1.0-libcamera
```

3. 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++ 生產者 SDK

請依照下列程序下載並建置 [Kinesis Video Streams C++ 生產者 SDK](#)。請確定您已安裝軟體先決條件；如需這些步驟 [the section called “安裝軟體先決條件”](#)，請參閱。

1. 導覽至下載目錄。開啟終端機並變更為您偏好的下載目錄。

例如：

```
cd ~/Downloads
```

2. 複製 SDK 儲存庫。使用 `git clone` 命令從 GitHub 儲存庫下載 SDK。類型：

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-  
cpp.git --single-branch -b master kvs-producer-sdk-cpp
```

此命令只會複製單一分支 (master 分支)，以減少下載大小和時間。它也會將下載的內容放入目前目錄中名為 `kvs-producer-sdk-cpp` 的資料夾。

3. 驗證下載。複製程序完成後，請列出 `kvs-producer-sdk-cpp` 資料夾的內容，以確認已下載 SDK。

```
ls kvs-producer-sdk-cpp
```

4. 準備建置目錄。類型：

```
mkdir -p kvs-producer-sdk-cpp/build
cd kvs-producer-sdk-cpp/build
```

5. 設定建置。執行下列cmake命令，以使用特定選項設定建置環境：

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=OFF -
DALIGNED_MEMORY_MODEL=ON
```

[CMake](#) 使用以下選項來產生適當的 Makefiles：

- 使用專案資料夾 (..) 做為來源目錄。
- 使用目前的目錄 (.) (build/) 進行建置輸出。
- -DBUILD_GSTREAMER_PLUGIN=ON 可建置 GStreamer 外掛程式 kvssink。
- -DBUILD_DEPENDENCIES=OFF 停用從來源建置外部相依性。專案會尋找並使用上一個步驟中安裝的外部相依性。
- -DALIGNED_MEMORY_MODEL=ON 會停用未對齊的記憶體模型。某些 Raspberry Pi 裝置不支援未對齊的記憶體存取。

Note

如需 CMake 引數的完整清單，請參閱 [the section called “下載並設定程式碼”](#)。

6. 建置專案。設定組建後，請使用 make 命令，使用 Makefile 產生的 編譯 cmake。

```
make -j$(nproc)
```

的 -j 引數make允許它平行執行多個編譯任務。若要減少建置時間，請使用 nproc 命令來動態計算 Raspberry Pi 上的 CPU 核心數量。

7. 確認 libgstkvssink.so 存在。

列出目前目錄中的檔案。

提示：

```
ls
```

回應：

```
CMakeCache.txt      dependency      kvs_gstreamer_sample
CMakeFiles          kvs_gstreamer_audio_video_sample  kvssink_gstreamer_sample
Makefile            kvs_gstreamer_file_uploader_sample libKinesisVideoProducer.so
cmake_install.cmake kvs_gstreamer_multistream_sample  libgstkvssink.so
```

8. 確認 GStreamer 可以載入 kvssink。

將 GST_PLUGIN_PATH 環境變數設定為包含的目錄 `libgstkvssink.so`。

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

讓 GStreamer 載入 kvssink：

```
gst-inspect-1.0 kvssink
```

您應該會看到一些有關的文件 kvssink。使用方向鍵導覽並按 q 退出。

9. (選用) 更新 shell 的啟動指令碼，以包含設定 GST_PLUGIN_PATH 環境變數。這可確保 GST_PLUGIN_PATH 在新的終端機工作階段期間正確設定。在 Raspberry Pi 裝置上，殼層的啟動指令碼為 `~/.bashrc`。

執行下列命令，將命令附加到 shell 啟動指令碼的結尾。

```
echo "export GST_PLUGIN_PATH=~/Downloads/kvs-producer-sdk-cpp/build" >> ~/.bashrc
```

輸入以下內容以執行 shell 的啟動指令碼，或關閉目前的 shell 並開啟新的 shell。

```
source ~/.bashrc
```

確認 GST_PLUGIN_PATH 已設定，您可以載入 kvssink。

```
echo $GST_PLUGIN_PATH
```

```
gst-inspect-1.0 kvssink
```

將影片串流到您的 Kinesis 影片串流

若要執行範本應用程式，您需要以下資訊：

- 在 [先決條件](#) 區段中建立之串流的名稱。
- 在 [建立具有寫入 Kinesis Video Streams 許可的 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)。
- GStreamer 能夠找到kvssink外掛程式。如需詳細資訊，請參閱 [the section called “下載並建置 C++ 生產者 SDK”](#)。

1. 設定登入資料和區域。

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

如需其他身分驗證方法，請參閱 [the section called “提供登入資料給 kvssink”](#)。

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，在美國西部 (奧勒岡) 區域中 AWS 區域 建立 Kinesis 影片串流。
若要為 Kinesis 影片串流使用不同的區域，請將下列環境變數設定為您的區域 (例如 *us-east-1*)：

```
export AWS_DEFAULT_REGION=us-east-1
```

2. 根據您的輸入媒體，選擇下列其中一項：

Sample GStreamer video

此 GStreamer 管道會產生即時測試影片串流，其標準測試模式以每秒 10 個影格執行，解析度為 640x480 像素。新增浮水印，顯示目前的系統時間和日期。然後影片會編碼為 H.264 格式，金鑰影格最多每 10 個影格產生一次，導致片段持續時間 (也稱為一組圖片 (GoP) 大小) 為 1 秒。kvssink 採用 H.264 編碼的影片串流，將其封裝為 Matroska (MKV) 容器格式，並將其上傳至 Kinesis 影片串流。

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 管道，請選取終端機視窗，然後按 CTRL+C。

範例影片 GStreamer 管道如下所示：

USB web cam

執行下列命令，讓 GStreamer 自動偵測您的 USB 攝影機：

```
gst-launch-1.0 autovideosrc \  
! videoconvert \  
! video/x-raw,format=I420,width=640,height=480 \  
! x264enc bframes=0 key-int-max=45 tune=zerolatency byte-stream=true speed-  
preset=ultrafast \  
! h264parse \  
! video/x-h264,stream-format=avc,alignment=au,profile=baseline \  
! kvssink stream-name="YourStreamname"
```

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

您可以 `v4l2src` 搭配特定裝置識別符使用，而不是讓 GStreamer 自動偵測。執行以下命令：

```
gst-device-monitor-1.0
```

在輸出中，您會看到一些裝置和 GStreamer 管道的啟動，說明如何使用裝置：

```
Device found:  
  
name : H264 USB Camera: USB Camera  
class : Video/Source  
caps : video/x-h264, stream-format=(string)byte-stream,  
alignment=(string)au, width=(int)1920, height=(int)1080, pixel-aspect-  
ratio=(fraction)1/1, colorimetry=(string){ 2:4:7:1 }, framerate=(fraction)  
{ 30/1, 25/1, 15/1 };
```

```

...
properties:
  device.path = /dev/video4
  udev-probed = false
  device.api = v4l2
  v4l2.device.driver = uvcvideo
  v4l2.device.card = "H264\ USB\ Camera:\ USB\ Camera"
  v4l2.device.bus_info = usb-3f980000.usb-1.3
  v4l2.device.version = 265767 (0x00040e27)
  v4l2.device.capabilities = 2216689665 (0x84200001)
  v4l2.device.device_caps = 69206017 (0x04200001)
gst-launch-1.0 v4l2src device=/dev/video4 ! ...

```

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

Raspberry Pi camera module 1

如果您使用 Pi 攝影機模組 1 或 Pi 攝影機模組 2 搭配 bcm2835-v4l2，請使用下列項目：

```

gst-launch-1.0 v4l2src device=/dev/video0 \
! videoconvert \
! video/x-raw,format=I420,width=640,height=480 \
! x264enc bframes=0 key-int-max=45 bitrate=500 tune=zerolatency \
! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \
! kvssink stream-name="YourStreamname"

```

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

Raspberry Pi camera module 2 or 3

如果您使用的是現代libcamera堆疊，請使用下列 GStreamer 管道：

```

gst-launch-1.0 libcamerasrc \
! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \
! videoconvert \
! x264enc speed-preset=ultrafast tune=zerolatency byte-stream=true key-int-
max=75 \
! video/x-h264,level='(string)4' \
! h264parse \
! video/x-h264,stream-
format=avc,alignment=au,width=640,height=480,framerate=30/1 \
! kvssink stream-name="YourStreamname"

```

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

利用硬體

有些 Raspberry Pi 模型隨附硬體加速 H.264 編碼器。您可以使用它們來取代 x264enc，這是軟體編碼器。

1. 請確定已安裝 GStreamer 外掛程式：

```
sudo apt-get install gstreamer1.0-tools gstreamer1.0-plugins-bad
```

2. 類型：

```
gst-inspect-1.0 | grep h264
```

判斷下列元素是否可用：

- omxh264enc
- v4l2h264enc

如果可用，您可以使用它們。以下是使用這些元素的一些管道範例：

omxh264enc:

```
gst-launch-1.0 v4l2src device=/dev/video0 \  
! videoconvert \  
! video/x-raw,format=I420,width=640,height=480 \  
! omxh264enc control-rate=2 target-bitrate=512000 periodicity-idr=45 inline-  
header=FALSE \  
! h264parse ! video/x-h264,stream-format=avc,alignment=au,profile=baseline \  
! kvssink stream-name="raspberry"
```

v4l2h264enc 和 v4l2convert：

```
gst-launch-1.0 libcamerasrc \  
! video/x-raw,width=640,height=480,framerate=30/1,format=I420 \  
! v4l2convert \  
! v4l2h264enc extra-controls="controls,repeat_sequence_header=1" \  
! video/x-h264,level=(string)4 \  
!
```

```
! h264parse \  
! video/x-h264,stream-format=avc,alignment=au,width=640,height=480,framerate=30/1  
\  
! kvssink stream-name="test-stream"
```

執行時間問題

以下是一些經常遇到的執行時間問題，以及如何進行故障診斷。

沒有這類元素 "xxxxxxxxx"

如果您收到類似以下的錯誤，表示您缺少 GStreamer 外掛程式：

```
WARNING: erroneous pipeline: no element "videoconvert"
```

解決方法：

根據缺少哪個元素，判斷適當的動作：

- kvssink：請參閱 [the section called “下載並建置 C++ 生產者 SDK”](#)。
- libcamerasrc：請參閱 [the section called “「緩衝集區啟用失敗」錯誤](#) 安裝 libcamerasrc GStreamer 元素。
- omxh264enc 或 v4l2h264enc：

遵循 [the section called “安裝軟體先決條件”](#) 安裝所有 GStreamer 程式庫。如果您已全部安裝它們，而且這些元素未顯示，這表示您的 Raspberry Pi 沒有硬體。請 x264enc 改用軟體編碼器。

- 其他：遵循 [the section called “安裝軟體先決條件”](#) 安裝所有 GStreamer 程式庫。在各種 GStreamer 外掛程式群組中找到不同的 GStreamer 元素（好、壞、醜），因此請務必全部安裝。

「緩衝集區啟用失敗」錯誤

如果您收到類似以下的錯誤，表示使用的管道正在使用 v4l2src，但應該 libcamerasrc 改用。

```
ERROR bufferpool gstbufferpool.c:572:gst_buffer_pool_set_active:source:pool0:src start  
failed  
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Failed to allocate  
required memory.  
WARN v4l2src gstv4l2src.c:976:gst_v4l2src_decide_allocation: error: Buffer pool  
activation failed
```

```
WARN basesrc gstbasesrc.c:3352:gst_base_src_prepare_allocation: Subclass failed to
decide allocation
Error received from element source: Failed to allocate required memory.
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: Internal data stream error.
Debugging information: ../sys/v4l2/gstv4l2src.c(976): gst_v4l2src_decide_allocation
(): /GstPipeline:live-kinesis-pipeline/GstV4l2Src:source:
Buffer pool activation failed
WARN basesrc gstbasesrc.c:3132:gst_base_src_loop: error: streaming stopped, reason not-
negotiated (-4)
```

例如，如果您在未安裝 `libcamerasrc` 攝影機模組 2 的情況下使用下列管道，當 GStreamer 嘗試自動偵測要使用的元素時，您可能會遇到此錯誤。

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

解決方法：

確定 `libcamerasrc` 已安裝，並將其用作來源元素，而不是 `v4l2src`。輸入以下內容以安裝 `libcamerasrc` GStreamer 元素：

```
sudo apt-get update
sudo apt-get install gstreamer1.0-libcamera
```

安裝 `libcamerasrc` 完成後，如果您使用的是 `autovideosrc` 元素，GStreamer 應該自動切換為使用正確的來源，`libcamerasrc` 而不是 `v4l2src`。

匯流排錯誤

如果您在啟動後不久收到匯流排錯誤 `kvssink`（通常是 HTTP 呼叫 `PutMedia` 完成的前後），這表示您的 Raspberry Pi 不支援未對齊的記憶體存取。日誌看起來如下：

```
INFO Camera camera.cpp:1197 configuring streams: (0) 640x480-YUV420
INFO RPI pisp.cpp:1450 Sensor: /base/axi/pcie@1200000/rp1/i2c@880000/imx708@1a - Selected
sensor format: 1536x864-SBGG10_1X10 - Selected CFE format: 1536x864-PC1B
[INFO ] kinesisVideoStreamFormatChanged(): Stream format changed.
[DEBUG] setRequestHeader(): Appending header to request: user-agent -> AWS-SDK-KVS-CPP-
CLIENT/3.4.2/1.5.3 GCC/12.2.0 Linux/6.6.51+rpt-rpi-v8 aarch64 CPPSDK
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-stream-name -> demo-
stream
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-producer-start-
timestamp -> 1732012345.678
```

```
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-
acknowledgment-required -> 1
[DEBUG] setRequestHeader(): Appending header to request: x-amzn-fragment-timecode-type
-> ABSOLUTE
[DEBUG] setRequestHeader(): Appending header to request: transfer-encoding -> chunked
[DEBUG] setRequestHeader(): Appending header to request: connection -> keep-alive
[INFO ] putStreamResultEvent(): Put stream result event. New upload handle 0
[WARN ] notifyDataAvailable(): [demo-stream] Failed to un-pause curl with error: 43.
  Curl object 0xe2f6f418
Bus error
```

Kinesis Video Streams PIC 使用未對齊的記憶體存取來最佳化記憶體用量，這並非所有裝置都支援。

解決方法：

若要在對齊的記憶體存取模式中使用 SDK，您需要在編譯 ON 時明確將 `ALIGNED_MEMORY_MODEL` CMake 旗標設定為 `kvssink`，因為它預設為 `OFF`。如需更詳細的說明 [the section called “下載並建置 C++ 生產者 SDK”](#)，請參閱。

時間戳記凍結且管道停滯

在 GStreamer 管道 `x264enc` 中使用時，您可能會遇到管道的時間軸在幾秒鐘內明顯或完全停止緩慢的情況。

這是因為 `x264enc` 預設設定可能會帶來高編碼延遲，這超過預設輸入緩衝區的容量。因此，輸入緩衝區會填滿，導致上游元素封鎖且管道停滯。

如需詳細資訊，請參閱 [GStreamer 文件](#)。

解決方法：

`x264enc` 使用 `zerolatency` 調校選項設定。這可針對即時案例進行最佳化，確保更快地處理和輸出影格，進而大幅降低編碼延遲。

範例組態：

```
... ! x264enc tune=zerolatency byte-stream=true speed-preset=ultrafast bframes=0 key-
int-max=60 ! ...
```

Note

雖然此解決方案可有效防止管道停滯，但可能會影響編碼效率和品質。對於同時需要低延遲和高品質的情況，請考慮替代方法，例如使用硬體最佳化或尋找直接輸出 H.264 的網路攝影機，略過此編碼步驟。

如需詳細資訊，請參閱 [the section called “利用硬體”](#)。

無法同時從相同 v4l2 裝置執行多個管道

這類裝置一次 /dev/video0 只能透過一個程序存取。如果多個程序嘗試同時存取它，第二個程序會等到第一個程序完成。

解決方法：

建立迴路裝置，允許多個程序同時使用迴路界面。如需詳細資訊，請參閱 [Stack Exchange](#)。

內部資料串流錯誤

當您建立 GStreamer 管道時，您可以透過將一個元素的來源鍵盤連結至另一個元素的接收器鍵盤來連接元素。此連結程序允許資料從來源元素流向接收器元素，形成資料管道。

日誌中的錯誤訊息「Pad 連結失敗」表示 GStreamer 在嘗試在管道中兩個元素的 Pad 之間建立連線（連結）時遇到問題。

```
Pad link failed
Error received from element udpsrc0: Internal data stream error.
```

解決方法：

判斷哪些元素無法互相連結。若要縮小管道範圍，請從管道中移除元素。將最右側的元素取代之為 fakesink，並一次移除一個元素。

您可能需要調整 [capsfilter](#) 元素，和/或變更管道使用的元素。

常見的情況是要求 framerate 或攝影機不支援 resolution。在終端機 gst-device-monitor-1.0 中使用來取得支援的 framerates、resolutions 和 formats。您可以使用 [影片規模](#) GStreamer 元素來調整影片解析度，並使用 [影片速率](#) 來調整影片影格速率。

若要檢查個別 GStreamer 元素支援的格式，請在終端機 gst-inspect-1.0 element-name 中輸入。

為了改善載入時間，請考慮減少關鍵影格間隔。這將建立較短的片段，減少延遲，但也會增加影片檔案的大小。

對於 x264enc GStreamer 元素，您可以透過 key-int-max 屬性明確設定關鍵影格間隔：

```
x264enc bframes=0 key-int-max=60
```

檢閱日誌輸出時，請注意上傳用戶端從 Kinesis Video Streams 接收 ACKs 的頻率。產生的關鍵影格越多，傳回ACKs 越多。

媒體失真或有成品

若要疑難排解此問題，請確定所有纜線都已緊密連接。檢閱攝影機模組的輸出 libcamera-hello (或 raspistill 舊版 Pi 攝影機的輸出)。

在您的 GStreamer 管道中，將取代 kvssink 為 autovideosink 或 matroskamux 和 filesink。例如：

```
... x264enc tune=zerolatency speed-preset=ultrafast bframes=0 key-int-max=60 byte-stream=true ! h264parse ! matroskamux ! filesink location=output.mkv
```

檢閱的輸出檔案，filesink 或是使用時開啟的媒體播放器 autovideosink，以查看成品是否也存在。

同時檢閱下列管道的輸出：

```
gst-launch-1.0 autovideosrc ! videoconvert ! autovideosink
```

將元素新增至管道，例如 [去彎曲](#)，可以修正魚眼攝影機輸出。

檢閱攝影機支援的輸出轉碼器，並視需要調整元素。

例如，如果您的 USB 攝影機僅支援 JPEG 輸出，則您將需要使用 jpegparse 和 jpegdec 元素來轉換媒體，然後再使用 將其編碼為 H.264x264enc。在 GStreamer 論壇上搜尋具有類似管道和/或網路攝影機設定的其他使用者的協助。

針對適用於 Raspberry Pi 的 C++ 生產者 SDK 的建置問題進行故障診斷

如果您遇到建置問題，並想要嘗試不同的 CMake 引數，請務必執行乾淨的建置。請先刪除 open-source、和 build 資料夾 dependency，然後再試一次。

使用 OpenSSL 建置問題

如果您收到類似以下的輸出，則表示 OpenSSL 未正確偵測到您的系統架構。

```
crypto/md5/md5-aarch64.S: Assembler messages:
crypto/md5/md5-aarch64.S:3: Error: unrecognized symbol type ""
crypto/md5/md5-aarch64.S:6: Error: bad instruction `stp x19,x20,[sp,#-80]!'
crypto/md5/md5-aarch64.S:7: Error: bad instruction `stp x21,x22,[sp,#16]'
crypto/md5/md5-aarch64.S:8: Error: bad instruction `stp x23,x24,[sp,#32]'
crypto/md5/md5-aarch64.S:9: Error: bad instruction `stp x25,x26,[sp,#48]'
```

在此範例中，嘗試在此 Raspberry Pi 實際為 32 位元時建置 64 位元版本 (linux-aarch64)。有些 Raspberry Pi 裝置具有 64 位元的核心，但有 32 位元的使用者空間。

決定 OpenSSL 嘗試建置的架構。您可以在 OpenSSL configure 的步驟中找到日誌列：

```
[ 33%] Performing update step for 'project_libopenssl'
-- Already at requested tag: OpenSSL_1_1_1t
[ 44%] No patch step for 'project_libopenssl'
[ 55%] Performing configure step for 'project_libopenssl'
Operating system: x86_64-whatever-linux2
Configuring OpenSSL version 1.1.1t (0x1010114fL) for linux-x86_64
Using os-specific seed configuration
Creating configdata.pm
Creating Makefile
```

驗證系統的架構：

- 檢閱核心位元：執行 `uname -m`
- 檢閱使用者空間位元：執行 `getconf LONG_BIT`

您也可以使用 `cat /proc/cpuinfo` 或 `lscpu` 命令檢閱 CPU 資訊。

解決方法：

若要解決此問題，請在建置時新增下列 CMake 引數，以確保 OpenSSL 針對 32 位元 ARM 架構正確建置：

```
-DBUILD_OPENSSL_PLATFORM=linux-armv4
```

對 GStreamer 中的 kvssink 載入問題進行故障診斷

確認 GST_PLUGIN_PATH

確保目前 Shell 工作階段中的 GST_PLUGIN_PATH 環境變數指向包含的目錄 kvssink。環境變數是工作階段特定的，因此您需要為每個新工作階段設定它們。若要讓此變更永久存在，請參閱「更新 shell 的啟動指令碼以包含設定 GST_PLUGIN_PATH 環境變數」。

錯誤：無法開啟共用物件檔案：沒有此類檔案或目錄

如果您遇到錯誤 `Cannot open shared object file: No such file or directory`，請執行下列命令：

```
gst-inspect-1.0 /path/to/libgstkvssink.so
```

如果您收到下列輸出，表示動態連結器找不到所需的程式庫 kvssink。這通常是因為下列原因發生：

- kvssink 移至與建置位置不同的位置。
- 交叉編譯錯誤的 CPU 架構。
- 缺少必要的相依性。

輸出：

```
WARNING: erroneous pipeline: no element "kvssink"  
error while loading shared libraries: libcproducer.so: cannot open shared object file:  
No such file or directory
```

解決方法：

對於移動的程式庫，請將包含遺失程式庫的目錄新增至 LD_LIBRARY_PATH。

從原始儲存庫的根目錄中，您可以使用 find 公用程式來尋找遺失的程式庫。在終端機中，輸入：

```
find . -name "*libcproducer*"
```

輸出：

```
./build/dependency/libkvsproducer/kvsproducer-src/libcproducer.so
```

Linux 裝置上的檔案路徑分隔符號為 `:`。以下命令會將新的資料夾路徑附加至現有的 `LD_LIBRARY_PATH` 環境變數，以保留任何先前的值。

在終端機中，輸入：

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path/to/build/dependency/libkvsproducer/
kvsproducer-src
```

Important

環境變數是工作階段特定的。若要保留跨工作階段的變更，請修改 shell 的啟動指令碼。

您可能還需要將 `open-source/local/lib` 新增至您的 `$LD_LIBRARY_PATH`。

錯誤：`./path/to/libcproducer.so.1`：無效的 ELF 標頭

如果您在載入共用程式庫時收到此錯誤，可能是由於符號連結 (`ld`) 損壞所致 `symlinks`。如果主機機器的作業系統不符合目標機器的作業系統，`Symlinks` 可能會中斷。例如，Raspberry Pi 的 MacBook 上的交叉編譯。

另一個可能的原因是建置的二進位檔用於錯誤的架構。例如，如果二進位檔是針對 x86 而建置 (Raspberry Pi 使用 ARM CPUs)。

導覽至錯誤和類型中指定的程式庫位置：`ls -la` 以檢查程式庫 `symlinks`。

回應：

```
drwxr-xr-x 16 me staff 512 Sep 10 17:16 .
drwxr-xr-x 7 me staff 224 Jan 6 23:46 ..
drwxr-xr-x 4 me staff 128 Sep 10 17:16 engines-1.1
-rwxr-xr-x 1 me staff 2294496 Sep 10 17:16 libcrypto.1.1.so
-rw-r--r-- 1 me staff 4002848 Sep 10 17:16 libcrypto.a
lrwxr-xr-x 1 me staff 19 Sep 10 17:16 libcrypto.so -> libcrypto.1.1.so
-rwxr-xr-x 1 me staff 631176 Sep 10 17:12 liblog4cplus-2.0.3.so
lrwxr-xr-x 1 me staff 24 Sep 10 17:12 liblog4cplus.so -> liblog4cplus-2.0.3.so
-rwxr-xr-x 1 me staff 1012 Sep 10 17:12 liblog4cplus.a
-rwxr-xr-x 1 me staff 694328 Sep 10 17:12 liblog4cplusU-2.0.3.so
lrwxr-xr-x 1 me staff 25 Sep 10 17:12 liblog4cplusU.dylib ->
liblog4cplusU-2.0.3.so
-rwxr-xr-x 1 me staff 1017 Sep 10 17:12 liblog4cplusU.a
-rwxr-xr-x 1 me staff 536416 Sep 10 17:16 libssl.1.1.so
```

```
-rw-r--r--  1 me  staff   795184 Sep 10 17:16 libssl.a
lrwxr-xr-x  1 me  staff      16 Sep 10 17:16 libssl.so -> libssl.1.1.so
drwxr-xr-x  6 me  staff    192 Sep 10 17:16 pkgconfig
```

在上述範例輸出中，`symlinks` 不會中斷。中斷`symlinks`不會有指向其目標的箭頭。

解決方法：

有兩種選項可以修正符號連結：

- 建議：`symlink`使用 `ln`命令重新建立。類型：

```
ln -s /path/to/actual/library /path/to/symlink
```

- 複製實際的程式庫檔案，並將其重新命名以符合 `symlink`。

Note

此選項會導致儲存用量增加。

最佳實務是使用 Docker 等工具在相同的作業系統上編譯，以避免跨編譯問題。

缺少相依性：

如果遺失的程式庫名稱以開頭`libkvs`，請參閱上述「移動的程式庫」一節，將 Kinesis Video Streams 程式庫從主機裝置安裝到目標裝置。

否則，請遵循 [the section called “安裝軟體先決條件”](#)以確保目標裝置上已安裝所有開放原始碼軟體先決條件。

錯誤碼參考

本節包含 [上傳至 Kinesis Video Streams](#) 的錯誤碼與狀態碼資訊。

有關常見問題的解決方法，請參閱 [故障診斷](#)。

主題

- [PutFrame 回呼傳回的錯誤和狀態碼 - 平台獨立代碼 \(PIC\)](#)
- [PutFrame 回呼 - C 生產者程式庫傳回的錯誤和狀態碼](#)

PutFrame 回呼傳回的錯誤和狀態碼 - 平台獨立代碼 (PIC)

下列各節包含平台獨立程式碼 (PIC) 內 PutFrame 操作的回呼所傳回的錯誤和狀態資訊。

主題

- [用戶端程式庫傳回的錯誤和狀態碼](#)
- [持續時間程式庫傳回的錯誤和狀態碼](#)
- [常見程式庫傳回的錯誤和狀態碼](#)
- [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](#)
- [檢視程式庫傳回的錯誤和狀態碼](#)

用戶端程式庫傳回的錯誤和狀態碼

下表包含 Kinesis Video Streams Client 程式庫中方法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 生產者開發套件配額 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。	在中指定大於零的串流數目上限 DeviceInfo 。
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	無效裝置名稱長度。	請參閱 中指定的字元的最大裝置名稱長度 生產者開發套件配額 。
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。

代碼	訊息	描述	建議的動作
0x52000005	STATUS_MAX_TAG_COUNT	已達標籤次數上限。	請參閱 中指定的目前最大標籤計數 生產者開發套件配額 。
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 中所指定字元的最大串流名稱長度 生產者開發套件配額 。
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 生產者開發套件配額 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 中指定的根目錄路徑長度上限 生產者開發套件配額 。
0x5200000c	STATUS_INVALID_SPILL_RATIO	無效的溢出率。	以 0–100 的百分比表示溢出率。

代碼	訊息	描述	建議的動作
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。	最常見的情況是，當 SDK 無法達到執行請求的操作所需的狀態時，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_CALLBACKS_MISSING	對於部分必要函數來說，Callbacks 架構含有遺失的函數入口點。	確認用戶端應用程式中已實作強制回呼。此錯誤只會向平台獨立程式碼 (PIC) 用戶端公開。C++ 和其他更高級別的包裝函式可滿足這些呼叫的要求。
0x52000010	STATUS_SERVICE_NOT_AUTHORIZED_ERROR	未獲授權。	驗證安全字符、憑證、安全字符整合和過期。確認權杖具有與其相關的正確權限。對於 Kinesis Video Streams 範例應用程式，請確認環境變數已正確設定。
0x52000011	STATUS_DESCRIBE_STREAM_CALL_FAILED	DescribeStream API 失敗。	此錯誤會在 DescribeStream API 重試失敗後傳回。PIC 用戶端會在停止重試後傳回此錯誤。

代碼	訊息	描述	建議的動作
0x52000012	STATUS_INVALID_DESCRIBE_STREAM_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。確認沒有其他程序在使用串流時嘗試刪除串流。
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。	確認在使用中時未刪除裝置。
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	該裝置未佈建。	確認裝置已佈建。
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 會確認先建立串流。

代碼	訊息	描述	建議的動作
0x52000018	STATUS_INVALID_AUTH_LEN	無效授權資訊長度。	請參考 生產者開發套件配額 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 生產者開發套件配額 中指定的目前最高數值。
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。

代碼	訊息	描述	建議的動作
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。我們建議您捨棄影格，因為它們掉出內容檢視視窗，而不是只是被放置 (內容存放區記憶體壓力)。這是因為捨棄影格會啟動串流壓力通知回呼。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。較高層級的用戶端 (例如 C++、Java 或 Android) 不會看到此警告，因為它是在內部處理。
0x52000030	STATUS_INVALID_TAG_VERSION	無效的 Tag 架構版本。	指定正確的目前架構版本。

代碼	訊息	描述	建議的動作
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。
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。
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。
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。
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。
0x52000037	STATUS_SERVICE_CALL_TIMEOUT_ERROR	服務呼叫逾時。	呼叫特定服務 API 產生逾時。確認您擁有有效的網路連線。PIC 會自動重試操作。

代碼	訊息	描述	建議的動作
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_DEVICE_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。確認標籤已指定或計數為零。
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 生產者開發套件配額 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 生產者開發套件配額 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。
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。

代碼	訊息	描述	建議的動作
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 生產者開發套件配額 中指定的值)。
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	客戶端授權失敗。	多次重試後，PIC 無法取得適當的身分驗證資訊 (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。
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。

代碼	訊息	描述	建議的動作
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。	高階 SDKs (例如 C++ 或 Java) 尚未實作裝置或生產者建立 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。
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	無效的安全字元過期。	安全性字元過期應具有未來大於目前時間戳記的絕對時間戳記，並具有寬限期。對於寬限期的限制，請參閱 生產者開發套件配額 。

代碼	訊息	描述	建議的動作
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，或者工作階段字符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 生產者開發套件配額 。
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。
0x5200004e	STATUS_INVALID_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。

代碼	訊息	描述	建議的動作
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。
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_CLIENT_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。

代碼	訊息	描述	建議的動作
0x52000056	STATUS_STATE_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影格和片段的大小。
0x5200005b	STATUS_VIEW_ITEM_SIZE_GREATER_THAN_ALLOCATION	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_ACK_ERROR_STREAM_READ_ERROR	串流讀取錯誤 ACK。	ACK 從後端傳回的錯誤，指出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。

代碼	訊息	描述	建議的動作
0x5200005d	STATUS_ACK_ERR_FRAGMENT_SIZE_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 生產者開發套件配額 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。檢查編碼器設定並確認關鍵影格是否正確產生。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_ACK_ERR_DURATION_REACHED	已達最高區段持續時間上限。	最高區段持續時間定義於 生產者開發套件配額 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。檢查編碼器設定，並確認金鑰影格是否定期正確產生。
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 中指定的最長連線持續時間 生產者開發套件配額 。製作者開發套件會在達到上限之前自動輪換串流或字符。使用 SDK 的用戶端不應收到此錯誤。

代碼	訊息	描述	建議的動作
0x52000060	STATUS_AK_ERR_FRAGMENT_TIMCODE_NOT_MONOTONIC	時間碼並非以單調遞增。	Producer SDK 會強制執行時間戳記，因此使用 SDK 的用戶端不應收到此錯誤。
0x52000061	STATUS_AK_ERR_MULTITRACK_MKV	在 MKV 中的多個音軌。	Producer SDK 會強制執行單一追蹤串流，因此使用 SDK 的用戶端不應收到此錯誤。
0x52000062	STATUS_AK_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。如果轉換期間串流損毀，使用 SDK 的用戶端可能會遇到此錯誤。如果緩衝壓力強制 SDK 捨棄部分傳輸的尾影格，也會發生這種情況。在後一種情況下，我們建議您減少 FPS 和解析度、提高壓縮率，或（如果有「爆量」網路）允許較大的內容存放區和緩衝區持續時間，以適應暫時壓力。

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。確認系統時鐘已正確設定。直接使用 PIC 的用戶端應驗證其回呼函數是否傳回正確的時間戳記。
0x52000064	STATUS_ACK_ERROR_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 AWS Key Management Service API 參考 。

代碼	訊息	描述	建議的動作
0x52000068	STATUS_ACK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key 無法使用。	金鑰無法使用。如需詳細資訊，請參閱 AWS Key Management Service API 參考 。
0x52000069	STATUS_ACK_ERR_KMS_KEY_INVALID_USAGE	KMS 金鑰的使用無效。	AWS KMS key 未設定為在此內容中使用。如需詳細資訊，請參閱 AWS Key Management Service API 參考 。
0x5200006a	STATUS_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS 無效狀態。	如需詳細資訊，請參閱 AWS Key Management Service API 參考 。
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 AWS Key Management Service API 參考 。
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 生產者開發套件配額 。
0x52000074	STATUS_MAX_FRAGMENT_METADATA_COUNT	片段中繼資料項目新增數量已達上限。	Kinesis 影片串流最多可以將 10 個中繼資料項目新增至片段，方法是將非持久性項目新增至片段，或將持久性項目新增至中繼資料佇列。如需詳細資訊，請參閱 搭配 Kinesis Video Streams 使用串流中繼資料 。

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 Producer SDK 程式碼中的限制變更，否則不會發生此錯誤。如需詳細資訊，請參閱 搭配 Kinesis Video Streams 使用串流中繼資料 。
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，中繼資料項目不會新增至片段或中繼資料佇列。如需詳細資訊，請參閱 搭配 Kinesis Video Streams 使用串流中繼資料 。
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	片段影格的結尾為無效狀態。	片段結尾不應在 non-key-frame 分段串流中傳送。
0x52000079	STATUS_TRACK_INFO_MISSING	遺失軌道資訊。	軌跡編號必須大於零，且符合軌跡 ID。
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。	離線串流模式保留時間不應設定為零。
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_HAND LE_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_PUTMEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		

代碼	訊息	描述	建議的動作
0x52000089	STATUS_NO N_ALIGNED _HEAP_WIT H_IN_CONT ENT_STORE _ALLOCATORS		
0x5200008a	STATUS_MU LTIPLE_CO NSECUTIVE_EOFR		
0x5200008b	STATUS_DU PLICATE_S TREAM_EVENT_TYPE		
0x5200008c	STATUS_ST REAM_NOT_STARTED		
0x5200008d	STATUS_IN VALID_IMA GE_PREFIX_LENGTH		
0x5200008e	STATUS_IN VALID_IMA GE_METADA TA_KEY_LENGTH		
0x5200008f	STATUS_IN VALID_IMA GE_METADA TA_VALUE_LENGTH		

持續時間程式庫傳回的錯誤和狀態碼

下表包含程式Duration庫中方法傳回的錯誤和狀態資訊。

代碼	訊息
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

常見程式庫傳回的錯誤和狀態碼

下表包含程式Common庫中方法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	沒有前導 0 的程式碼	訊息	描述
0x00000001	0x1	STATUS_NULL_ARG	為必要引數傳遞 NULL。
0x00000002	0x2	STATUS_INVALID_ARG	為引數指定了無效的值。
0x00000003	0x3	STATUS_INVALID_ARG_LEN	指定了無效的引數長度。
0x00000004	0x4	STATUS_NOT_ENOUGH_MEMORY	無法配置足夠的記憶體。
0x00000005	0x5	STATUS_BUFFER_TOO_SMALL	指定的緩衝大小太小。
0x00000006	0x6	STATUS_UNEXPECTED_EOF	達到未預期的檔案結尾。
0x00000007	0x7	STATUS_FORMAT_ERROR	遇到無效的格式。

代碼	沒有前導 0 的程式碼	訊息	描述
0x00000008	0x8	STATUS_INVALID_HANDLE_ERROR	無效的處理常式值。
0x00000009	0x9	STATUS_OPEN_FILE_FAILED	無法開啟檔案。
0x0000000a	0xa	STATUS_READ_FILE_FAILED	無法從檔案讀取。
0x0000000b	0xb	STATUS_WRITE_TO_FILE_FAILED	無法寫入檔案。
0x0000000c	0xc	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	0xd	STATUS_INVALID_OPERATION	具有無效的操作，或者不允許執行該操作。
0x0000000e	0xe	STATUS_NOT_IMPLEMENTED	未套用功能。
0x0000000f	0xf	STATUS_OPERATION_TIMED_OUT	此操作逾時。
0x00000010	0x10	STATUS_NOT_FOUND	找不到需要的資源。
0x00000011	0x11	STATUS_CREATE_THREAD_FAILED	無法建立執行緒。

代碼	沒有前導 0 的程式碼	訊息	描述
0x00000012	0x12	STATUS_TH READ_NOT_ ENOUGH_RE SOURCES	資源不足以建立另一個執行緒，或遇到系統對執行緒數量施加的限制。
0x00000013	0x13	STATUS_TH READ_INVA LID_ARG	指定的執行緒屬性無效，或另一個執行緒已等待加入此執行緒。
0x00000014	0x14	STATUS_TH READ_PERM ISSIONS	沒有許可來設定執行緒屬性中指定的排程政策和參數。
0x00000015	0x15	STATUS_TH READ_DEAD LOCKED	偵測到死結或聯結執行緒指定呼叫執行緒。
0x00000016	0x16	STATUS_TH READ_DOES _NOT_EXIST	找不到具有指定執行緒 ID 的執行緒。
0x00000017	0x17	STATUS_JO IN_THREAD _FAILED	從執行緒聯結操作傳回未知或一般錯誤。
0x00000018	0x18	STATUS_WA IT_FAILED	超過等待條件式變數的時間上限。
0x00000019	0x19	STATUS_CA NCEL_THRE AD_FAILED	從執行緒取消操作傳回未知或一般錯誤。
0x0000001a	0x1a	STATUS_TH READ_IS_N OT_JOINABLE	在不可聯結的執行緒上請求執行緒聯結操作。

代碼	沒有前導 0 的程式碼	訊息	描述
0x0000001b	0x1b	STATUS_DETACH_THREAD_FAILED	從執行緒分離操作傳回未知或一般錯誤。
0x0000001c	0x1c	STATUS_THREAD_ATTR_INIT_FAILED	無法初始化執行緒屬性物件。
0x0000001d	0x1d	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	無法設定執行緒屬性物件的堆疊大小。
0x0000001e	0x1e	STATUS_MEMORY_NOT_FREED	僅用於測試。表示尚未釋放所有請求的記憶體。
0x0000001f	0x1f	STATUS_INVALID_THREAD_PARAMS_VERSION	無效的 "ThreadParams" 結構版本。指定正確的目前架構版本。

堆積程式庫傳回的錯誤和狀態碼

下表包含程式Heap庫中方法傳回的錯誤和狀態資訊。

代碼	訊息	描述
0x10000001	STATUS_HEAP_FLAGS_ERROR	指定的標籤組合無效。
0x10000002	STATUS_HEAP_NOT_INITIALIZED	堆積初始化前已嘗試操作。
0x10000003	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

代碼	訊息	描述
		的緩衝區溢位可能導致堆積損壞。
0x10000004	STATUS_HEAP_VRAM_L IB_MISSING	無法載入或遺失 VRAM (影片 RAM) 使用者或核心模式程式庫。檢查基礎平台是否支援 VRAM 分配。
0x10000005	STATUS_HEAP_VRAM_L IB_REOPEN	無法開啟 VRAM 程式庫。
0x10000006	STATUS_HEAP_VRAM_I NIT_FUNC_SYMBOL	無法載入 INIT 函數匯出。
0x10000007	STATUS_HEAP_VRAM_A LLOC_FUNC_SYMBOL	無法載入 ALLOC 函數匯出。
0x10000008	STATUS_HEAP_VRAM_F REE_FUNC_SYMBOL	無法載入 FREE 函數匯出。
0x10000009	STATUS_HEAP_VRAM_L OCK_FUNC_SYMBOL	無法載入 LOCK 函數匯出。
0x1000000a	STATUS_HEAP_VRAM_U NLOCK_FUNC_SYMBOL	無法載入 UNLOCK 函數匯出。
0x1000000b	STATUS_HEAP_VRAM_U NINIT_FUNC_SYMBOL	無法載入 UNINIT 函數匯出。
0x1000000c	STATUS_HEAP_VRAM_G ETMAX_FUNC_SYMBOL	無法載入 GETMAX 函數匯出。
0x1000000d	STATUS_HEAP_DIRECT _MEM_INIT	無法初始化在混合堆積中的主要堆積集區。
0x1000000e	STATUS_HEAP_VRAM_I NIT_FAILED	VRAM 動態初始化失敗。

代碼	訊息	描述
0x1000000f	STATUS_HEAP_LIBRARY_FREE_FAILED	無法取消分配和釋放 VRAM 程式庫。
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	VRAM 分配失敗。
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	VRAM 釋放失敗。
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	VRAM 對應失敗。
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	VRAM 取消對應失敗。
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	VRAM 取消初始化失敗。
0x10000015	STATUS_INVALID_ALLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

MKVGen 程式庫傳回的錯誤和狀態碼

下表包含程式MKVGen庫中方法傳回的錯誤和狀態資訊。

代碼	訊息	描述/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

代碼	訊息	描述/建議的動作
		資料有效，且在 中指定的限制 內 生產者開發套件配額 。
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 錯誤：「無法將影格提交到 Kinesis Video 用戶端」 。
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 生產者開發套件配額 。
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 生產者開發套件配額 。
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 生產者開發套件配額 。
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 生產者開發套件配額 。
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 生產者開發套件配額 。

代碼	訊息	描述/建議的動作
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	轉碼器私有資料 (CPD) 為 NULL，而 CPD 大小大於零。
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	時間碼比例值無效。如需詳細資訊，請參閱 生產者開發套件配額 。
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 生產者開發套件配額 。
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	描述/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 在 Android 上發生 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA (0x3200000d) 錯誤 。
0x3200000e	STATUS_MKV_INVALID_AVCC_NALU_IN_FRAME_DATA	指定調整 AVCC 旗標時，無效的 AVCC NALU 封裝。確認位元組串流是有效的 AVCC 格式。如需詳細資訊，請參閱 MPEG 規格。
0x3200000f	STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED	已指定調整 AVCC 和 Annex-B NALUs。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。確認 CPD 為有效的 Annex-B 格式。如果不是，請移除 CPD Annex-B 適應旗標。

代碼	訊息	描述/建議的動作
0x32000011	STATUS_MKV_PTS_DTS _ARE_NOT_SAME	Kinesis Video Streams 會強制執行片段開始影格的 PTS (呈現時間戳記) 和 DTS (解碼時間戳記) 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 SPS NALU。
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	描述/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	AnnexBb 轉碼器私有 Beta 值大小下限。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	描述/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

下表包含程式 Trace 庫中方法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

下表包含程式Utils庫中方法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

檢視程式庫傳回的錯誤和狀態碼

下表包含程式View庫中方法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 生產者開發套件配額 。
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 生產者開發套件配額 。
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前一個影格時間戳記，加上前一個影

代碼	訊息	描述
		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 StreamInfo.StreamCaps 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。不過，若要處理編碼器「抖動」，建議您使用該影格持續時間的一半 (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。

PutFrame 回呼 - C 生產者程式庫傳回的錯誤和狀態碼

下一節包含 C 生產者程式庫中 PutFrame 操作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_MAX_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_MAX_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_DESCRIBE_STREAM_RETURN_JSON	針對 DescribeStream 傳回無效的 JSON。	
0x15000010	STATUS_HMAC_GENERATION_ERROR	HMAC 產生錯誤。	
0x15000011	STATUS_IOT_FAILED	IoT 授權失敗。	
0x15000012	STATUS_MAX_ROLE_ALIAS_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 過期時間戳記發生在過去。	
0x15000018	STATUS_IOT_EXPIRATION_PARSING_FAILED	IoT 過期剖析失敗。	
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_BEING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RESET_FAILED		

代碼	訊息	描述	建議的動作
0x16000001	STATUS_CURL_PERFORM_FAILED	CURL 傳回非成功代碼。	<p>檢閱日誌以取得其他資訊。常見的 CURL 錯誤是「無法解析主機名稱」，請檢查裝置的網際網路連線。</p> <p>另一個常見的錯誤是 403 錯誤代碼。這表示未正確建立或指定 IoT 憑證。檢查 IoT 憑證和許可的檔案路徑是否已正確設定。如需詳細資訊，請參閱the section called “使用 控制對 Kinesis Video Streams 資源的存取 AWS IoT”。</p>
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH	擷取 IoT 登入資料時收到 0 長度的回應。	檢閱 AWS 運作狀態儀表板，稍後再試一次。
0x16000003	STATUS_IOT_NULL_AWS_CREDS	從 IoT 登入資料端點傳回的 JSON 不包含登入資料物件。	如需其他資訊，請參閱 JSON 中的「訊息」項目。
0x16000004	STATUS_IOT_INVALID_URI_LEN	傳入擷取 IoT 登入資料函數的 URL 長度不介於 1 到 10,000 之間。	檢閱傳遞至此函數的 URL。
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT	JSON 中從擷取 IoT 登入資料的「過期」項目的格式不是：YYYY-MM-DDTHH:mm:ssZ 。	檢閱 AWS 運作狀態儀表板，稍後再試一次。

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。某些高階位元串流處理器 (例如播放引擎或中的[媒體來源延伸 \(MSE\)](#) 播放器 AWS Management Console) 會針對其影格使用 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
<code>NAL_ADAPTATION_FLAG_NONE</code>	不適應。
<code>NAL_ADAPTATION_ANNEXB_NALS</code>	將 Annex-B NALUs 調整為 AVCC NALUs。
<code>NAL_ADAPTATION_AVCC_NALS</code>	將 AVCC NALUs 調整為 Annex-B NALUs。
<code>NAL_ADAPTATION_ANNEXB_CPD_NALS</code>	將轉碼器私有資料的 Annex-B NALUs 調整為 AVCC 格式 NALUs。
<code>NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS</code>	將轉碼器和影格私有資料的 Annex-B NALUs 調整為 AVCC 格式 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](#)。

生產者開發套件結構

本節包含您可以用來將資料提供給 Kinesis Video Streams Producer 物件的結構相關資訊。

主題

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

DeviceInfo 和 DefaultDeviceInfoProvider 物件控制 Kinesis Video Streams Producer 物件的行為。

成員欄位

- 版本 – 整數值，用來確保結構的正確版本與程式碼基底的目前版本搭配使用。目前版本指定使用 DEVICE_INFO_CURRENT_VERSION 巨集。
- name – 裝置的人類可讀名稱。
- tagCount/tags – 目前未使用。
- streamCount – 裝置可處理的串流數目上限。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 主要儲存體的組態。

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的記憶體，建議您將更多記憶體新增至儲存體，以避免嚴重分段。

確認儲存體大小足以容納高編碼複雜性（因高動作而增加影格大小）或頻寬低的所有串流的完整緩衝區。如果生產者達到記憶體壓力，它會發出儲存溢出壓力回呼 (StorageOverflowPressureFunc)。不過，當內容存放區中沒有可用的記憶體時，它會捨棄推送到 Kinesis Video Streams 的影格，並顯示錯誤 (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。在此情況下，緩衝區會在前一個影格開始捨棄之前填充至「緩衝持續時間」容量。

成員欄位

- 版本 – 整數值，用來確保結構的正確版本與程式碼基底的目前版本搭配使用。
- storageType – DEVICE_STORAGE_TYPE 指定儲存體基礎備份和實作的列舉。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 – 要預先配置的儲存體大小，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，而不是次要溢出儲存體（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 影片串流結構

您可以使用下列結構，將資料提供給 Kinesis 影片串流的執行個體。

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
stream_name	string	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 生產者開發套件配額 。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。
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	用於加密串流的 AWS KMS 金鑰 ID。如需詳細資訊，請參閱 Kinesis Video Streams 中的資料保護 。	預設 KMS 金鑰 (aws/kinesisvideo)。
streaming_type	STREAMING_TYPE 列舉	唯一支援的值為 STREAMING_TYPE_REALTIME。	

欄位	資料類型	描述	預設值
content_type	string	串流的內容格式。Kinesis Video Streams 主控台可以 video/h264 格式播放內容。	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()
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，Kinesis Video Streams 會在經過此持續時間後，在金鑰影格上產生片段。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。確認框架時間碼可以使用指定的時間碼比例來表示。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 Amazon Kinesis Video Streams 服務配額 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。如果為 false，Kinesis Video Streams 至少會等待，fragment_duration 並在其後面的金鑰影格上產生新的片段。	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，指定此參數可確保影格在放入 Kinesis Video Streams 時加上時間戳記。	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	是否接收應用程式層級片段 ACKs (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，我們不建議使用 false 做為此值。	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。大多數基本串流生產者和消費者（讀取編碼器和解碼器）都使用 Annex-B 格式，因為它具有優勢，例如錯誤復原。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: 5px; margin-top: 10px;"> <p>Ab(Sps)Ab (Pps)Ab(I- frame)Ab(P/B-</p> </div>	<p>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。</p>

欄位	資料類型	描述	預設值
		<p>frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame) Ab(P/B-frame)</p> <p>在 H.264 的情況下，轉碼器私有資料 (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。它可以透過尋找第一個 IDR 影格 (應包含 SPS 和 PPS)、擷取兩個 NALUs (即 Ab(Sps)Ab(Pps))，並在的 CPD 中設定它來執行此操作StreamDefinition。</p> <p>如需詳細資訊，請參閱the section called “NAL 適應旗標”。</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

欄位	資料類型	描述	預設值
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
buffer_duration	duration<uint64_t>	串流緩衝持續時間，以秒為單位。開發套件會將內容存放區中的影格保留至 <code>buffer_duration</code> ，之後會隨著視窗向前移動而捨棄先前的影格。如果正在捨棄的影格尚未傳送至後端，則會呼叫捨棄的影格回呼。如果目前緩衝區持續時間大於 <code>max_latency</code> ，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，則在錯誤期間向後滾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並未指示無效主機 (主機仍然有效，並在其內部緩衝區中包含影格資料)，則轉返會在最後收到的 ACK 影格停止。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，則會呼叫串流過時回呼的時間，以秒為單位。它表示正在從裝置傳送影格，但後端未確認它們。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。不過，如果資料在建立串流時無法使用，則可以在 KinesisVideoStream.start(cpd) 函數的其中一個過載中設定。	null
codecPrivateDataSize	uint32_t	編解碼器私有資料緩衝區大小。	0

ClientMetrics

ClientMetrics 物件透過呼叫 `getKinesisVideoMetrics` 來填充。

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。
contentStoreAvailableSize	UINT64	目前可用的儲存大小，以位元組為單位。
contentStoreAllocatedSize	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。
totalContentViewsSize	UINT64	所有串流的所有內容視圖的已分配記憶體大小。這不會計入儲存體大小。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

透過呼叫 `StreamMetricsGetKinesisVideoMetrics` 物件。

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。在快速聯網案例中，此持續時間為零或影格持續時間（正在傳輸影格時）。如果持續時間超過中max_latency 指定的持續時間StreamDefinition，則會在指定時呼叫串流延遲回呼。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 ACKs或持久性，則此值會隨著影格放入 Kinesis 影片串流而增加，並等於 buffer_duration 中的 StreamDefinition。啟用 ACKs並收到持久 ACK 時，緩衝區會修剪為下一個金鑰影格。這是因為 ACK 時間戳記表示整個片段的開頭。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。
overallViewSize	UINT64	整體檢視大小，以位元組為單位。

欄位	資料類型	描述
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。

生產者開發套件回呼

Amazon Kinesis Video Streams Producer SDK 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](#) – 此物件會從平台獨立程式碼 (PIC) 元件向應用程式公開每個回呼。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](#) – 這些物件公開串流特定和用戶端特定回呼，而 SDK 的 C++ 層公開其餘回呼。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](#)
- [要重試串流的回呼實作](#)

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 [the section called “ClientCallbacks”](#) 節。

回呼方法：

- `getClientReadyCallback` – 報告用戶端的就緒狀態。
- `getStorageOverflowPressureCallback` – 報告儲存溢出或壓力。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` – 報告捨棄的影格。
- `getFragmentAckReceivedCallback` – 報告已接收串流的片段 ACK。
- `getStreamClosedCallback` – 報告串流關閉條件。
- `getStreamConnectionStaleCallback` – 報告過時的連線條件。在這種情況下，生產者正在將資料傳送至服務，但沒有收到確認。
- `getStreamDataAvailableCallback` – 報告資料可在串流中使用。
- `getStreamErrorReportCallback` – 報告串流錯誤條件。
- `getStreamLatencyPressureCallback` – 報告串流延遲條件，即累積的緩衝區大小大於 `max_latency` 值時。如需詳細資訊，請參閱 [StreamDefinition/StreamInfo](#)。
- `getStreamReadyCallback`：–報告串流就緒條件。
- `getStreamUnderflowReportCallback` – 報告串流下溢條件。此函數目前未使用，並保留供日後使用。

如需原始程式碼詳細資訊 `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

函式	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
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

函式	Description (描述)	Type
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。此函數目前未使用，並保留供日後使用。	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
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

函式	Description (描述)	Type
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 StreamDefinition/StreamInfo 。	通知
StreamReadyFunc	當串流進入就緒狀態時呼叫。	通知
StreamUnderflowReportFunc	此函數目前未使用，並保留供日後使用。	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。此函數目前未使用，並保留供日後使用。	平台整合
GetDeviceFingerprintFunc	傳回裝置指紋。此函數目前未使用，並保留供日後使用。	平台整合
GetRandomNumberFunc	傳回一隨機編號介於 0 和 RAND_MAX 之間。	平台整合
GetSecurityTokenFunc	傳回傳遞給與後端 API 通訊之函數的安全性字符。此實作可指定序列化 AccessKeyId, SecretKeyId 以及工作階段字符。	平台整合

函式	Description (描述)	Type
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。

要重試串流的回呼實作

Kinesis Video Producer SDK 提供透過回呼函數串流的狀態。我們建議您實作下列回呼機制，從串流期間遇到的任何暫時性網路問題中復原。

- 串流延遲壓力回呼 - 此回呼機制會在 SDK 遇到串流延遲條件時啟動。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。您可以視需要使用 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/StreamLatencyStateMachine.c> 的範例實作。請注意，由於網路中斷，無法將未交付的影格存放在次要儲存中，以供回填。

- 串流過時回呼 - 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（預設為 60 秒），就會啟動此回呼。根據網路設定，串流延遲壓力回呼或串流過時回呼，或兩者都可以啟動。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。您可以視需要使用 <https://github.com/awslabs/amazon-kinesis-video-streams-producer-c/blob/master/src/source/ConnectionStateStateMachine.c> 的範例實作。
- 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。
- 捨棄影格回呼 - 當儲存大小已滿時，由於網路速度緩慢或串流錯誤，會啟動此回呼。如果網路速度導致影格中斷，您可以增加儲存體大小、減少影片影格大小或影格速率以符合網路速度。

搭配 Kinesis Video Streams 使用串流中繼資料

您可以使用 Amazon Kinesis Video Streams 生產者 SDK，在 Kinesis 影片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鍵/值對。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感應器讀數，或滿足其他自訂需求。中繼資料提供為 [the section called “GetMedia”](#) 或 [the section called “GetMediaForFragmentList”](#) API 操作的一部分。它會在整個串流保留期間與片段一起存放。您的取用應用程式可以根據中繼資料使用來讀取、處理和反應 [使用剖析器程式庫觀察攝影機的輸出](#)。

中繼資料可以用兩種模式內嵌於串流中的片段：

- 非持久性 – 您可以根據已發生的業務特定條件，將中繼資料一次性或臨時附加至串流中的片段。例如，智慧攝影機可偵測動作，並在將片段傳送至其 Kinesis 影片串流之前，將中繼資料新增至包含動作的對應片段。您可以將中繼資料套用至下列格式的片段中：`Motion = true`。
- 持久性 – 您可以根據持續的需求，將中繼資料附加至串流中的連續片段。例如，智慧攝影機會傳送與其傳送至其 Kinesis 影片串流之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。

將中繼資料新增至 Kinesis 影片串流

您新增至 Kinesis 影片串流的中繼資料會建模為 MKV 標籤，這些標籤會實作為索引鍵/值對。

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。持久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 [上傳至 Kinesis Video Streams](#) 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

您可以在生產者 SDK 中使用下列操作來實作串流中繼資料。

主題

- [PIC](#)
- [C++ 生產者 SDK](#)
- [Java 生產者 SDK](#)
- [持久性和非持久性中繼資料](#)

PIC

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

C++ 生產者 SDK

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

Java 生產者 SDK

您可以使用 Java 生產者 SDK，MediaSource 使用 `將中繼資料新增至 MediaSourceSink.onCodecPrivateData`：

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

持久性和非持久性中繼資料

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。生產者 SDK 會收集中繼資料佇列中的中繼資料項目，直到它們在下一個片段前面。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`。

對於持久性中繼資料，生產者 SDK 會以與非持久性中繼資料相同的方式收集中繼資料佇列中的中繼資料項目。不過，中繼資料項目在加上下一個片段時，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。

Kinesis Video Streams 播放

您可以使用下列方法檢視 Kinesis 影片串流：

- **GetMedia** – 您可以使用 GetMedia API 建置自己的應用程式來處理 Kinesis Video Streams。GetMedia 是一種低延遲的即時 API。若要建立使用的玩家 GetMedia，您必須自行建置。如需有關如何使用開發顯示 Kinesis 影片串流的應用程式的資訊 GetMedia，請參閱 [使用剖析器程式庫進行串流](#)。
- **HLS** – [HTTP Live Streaming \(HLS\)](#) 是業界標準的 HTTP 型媒體串流通訊協定。您可以使用 HLS 來檢視 Kinesis 影片串流，無論是即時播放或檢視封存的影片。

您可以使用 HLS 進行即時播放。延遲通常介於 3–5 秒之間，但也可能介於 1–10 秒之間，具體取決於使用案例、玩家和網路條件。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。您也可以可以在 [Apple Safari](#) 或 [Microsoft Edge](#) 瀏覽器的位置列中輸入 HLS 串流工作階段 URL 來播放影片。

- **MPEG-DASH** – [動態自適應 HTTP 串流 \(DASH\)](#)，也稱為 MPEG-DASH，是一種自適應位元速率串流通訊協定，可透過傳統 HTTP Web 伺服器提供的網際網路，高品質串流媒體內容。

您可以使用 MPEG-DASH 進行即時播放。延遲通常介於 3–5 秒之間，但也可能介於 1–10 秒之間，具體取決於使用案例、玩家和網路條件。您可以使用第三方播放器 (例如 [dash.js](#) 或 [Google Shaka Player](#))，以程式設計或手動方式提供 MPEG-DASH 串流工作階段 URL，以顯示影片串流。

- **GetClip** – 您可以使用 GetClip API 下載剪輯 (在 MP4 檔案中)，其中包含在指定時間範圍內從指定影片串流封存的隨需媒體。如需詳細資訊，請參閱 [GetClip API 參考](#)。

主題

- [影片播放軌跡需求](#)
- [使用 HLS 播放影片](#)
- [使用 MPEG-DASH 播放影片](#)

影片播放軌跡需求

Amazon Kinesis Video Streams 支援以多種格式編碼的媒體。如果您的 Kinesis 影片串流使用下列四個 APIs 之一不支援的格式，請使用 [GetMedia](#) 或 [GetMediaForFragmentList](#)，因為它們沒有追蹤類型限制。

主題

- [GetClip 需求](#)
- [GetDASHStreamingSessionURL 要求](#)
- [GetHLSStreamingSessionURL 要求](#)
- [GetImages 需求](#)

GetClip 需求

如需這種 API 的詳細資訊，請參閱 [GetClip](#)。

軌道 1 描述	軌道 1 轉碼器 ID	軌道 2 描述	軌道 2 轉碼器 ID
H.264 影片	V_MPEG/ISO/AVC	N/A	N/A
H.264 影片	V_MPEG/ISO/AVC	AAC 音訊	A_AAC
H.264 影片	V_MPEG/ISO/AVC	G.711 音訊 (僅限 A-Law)	A_MS/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

Important

每個片段中包含的編解碼器私有資料 (CPD) 包含編解碼器特定的初始化資訊，例如影格速率、解析度和編碼描述檔，這些是正確解碼片段的必要項目。結果剪輯的目標片段之間不支援 CPD 變更。CPD 必須透過查詢媒體保持一致性，否則會傳回錯誤。

Important

不支援追蹤變更。追蹤在整個查詢媒體中必須保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則會傳回錯誤。

GetDASHStreamingSessionURL 要求

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

軌道 1 描述	軌道 1 轉碼器 ID	軌道 2 描述	軌道 2 轉碼器 ID
H.264 影片	V_MPEG/ISO/AVC	N/A	N/A
H.264 影片	V_MPEG/ISO/AVC	AAC 音訊	A_AAC
H.264 影片	V_MPEG/ISO/AVC	G.711 音訊 (僅限 A-Law)	A_MS/ACM
H.264 影片	V_MPEG/ISO/AVC	G.711 音訊 (僅限 U-Law)	A_MS/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

Important

每個片段中包含的編解碼器私有資料 (CPD) 包含編解碼器特定的初始化資訊，例如影格速率、解析度和編碼描述檔，這些是正確解碼片段的必要項目。串流工作階段期間不支援 CPD 變更。CPD 必須透過查詢的媒體保持一致性。

Important

不支援追蹤變更。追蹤在整個查詢媒體中必須保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則串流將會失敗。

GetHLSStreamingSessionURL 要求

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

HLS Mp4

軌道 1 描述	軌道 1 轉碼器 ID	軌道 2 描述	軌道 2 轉碼器 ID
H.264 影片	V_MPEG/ISO/AVC	N/A	N/A
H.264 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
H.265 影片	V_MPEGH/ISO/HEVC	AAC 音訊	A_AAC

HLS TS

軌道 1 描述	軌道 1 轉碼器 ID	軌道 2 描述	軌道 2 轉碼器 ID
H.264 影片	V_MPEG/ISO/AVC	N/A	N/A
H.264 影片	V_MPEG/ISO/AVC	AAC 音訊	A_AAC
AAC 音訊	A_AAC	N/A	N/A

Note

每個片段中包含的編解碼器私有資料 (CPD) 包含編解碼器特定的初始化資訊，例如影格速率、解析度和編碼描述檔，這些是正確解碼片段的必要條件。對於 TS 和 MP4，串流工作階段期間支援 CPD 變更。因此，工作階段中的片段在 CPD 中可以有不同的資訊，而不會中斷播放。對於每個串流工作階段，僅允許 500 個 CPD 變更。

Important

不支援追蹤變更。追蹤在整個查詢媒體中必須保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則串流將會失敗。

GetImages 需求

如需這種 API 的詳細資訊，請參閱 [GetImages](#)。

Note

GetImages 媒體應該包含軌道 1 中的影片軌跡。

使用 HLS 播放影片

[HTTP 即時串流 \(HLS\)](#) 是業界標準的 HTTP 型媒體串流通訊協定。您可以使用 HLS 來檢視 Kinesis 影片串流，無論是即時播放或檢視封存的影片。

您可以使用 HLS 進行即時播放。延遲通常介於 3 到 5 秒之間，但也可能介於 1 到 10 秒之間，視使用案例、播放器和網路條件而定。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。您也可以使用 [Apple Safari](#) 或 [Microsoft Edge](#) 瀏覽器的位置列中輸入 HLS 串流工作階段 URL 來播放影片。

若要使用 HLS 檢視 Kinesis 影片串流，請先使用 [GetHLSStreamingSessionURL](#) 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

並非所有傳送到 Kinesis Video Streams 的媒體都可以透過 HLS 播放。如需特定上傳需求 [the section called “GetHLSStreamingSessionURL”](#)，請參閱。

主題

- [使用 AWS CLI 擷取 HLS 串流工作階段 URL](#)
- [範例：在 HTML 和 JavaScript 中使用 HLS](#)
- [故障診斷 HLS 問題](#)

使用 AWS CLI 擷取 HLS 串流工作階段 URL

下列程序示範如何使用 AWS CLI 來產生 Kinesis 影片串流的 HLS 串流工作階段 URL。

如需安裝說明，請參閱[AWS Command Line Interface 《使用者指南》](#)。安裝後，使用登入資料和區域設定 [AWS CLI](#)。

或者，開啟 AWS CLI 已安裝並設定的 AWS CloudShell 終端機。如需詳細資訊，請參閱 [AWS CloudShell 使用者指南](#)。

擷取 Kinesis 影片串流的 HLS URL 端點。

1. 在終端機中輸入下列項目：

```
aws kinesisisvideo 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. 向傳回的端點提出 HLS 串流工作階段 URL 請求。

Live

針對即時播放，HLS 媒體播放清單會持續更新為可用的最新媒體。當您在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示「即時」通知，而沒有用於選擇要顯示之播放視窗中的位置的清理程式控制。

執行此命令時，請務必將媒體上傳至此串流。

```
aws kinesisisvideo 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 媒體播放清單也會隨著最新的媒體可用而持續更新。工作階段將繼續包含新擷取的媒體，直到工作階段過期，或直到指定的結束時間，以先到者為準。此模式有助於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。

決定開始時間戳記。

在此範例中，我們使用秒格式的 Unix Epoch 時間。如需[時間戳記](#)格式的詳細資訊，請參閱 AWS Command Line Interface 《使用者指南》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 2024 年 2 月 20 日下午 3 : 30 : 00 GMT-08 : 00

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

使用LIVE_REPLAY播放模式和指定的 [HLS Fragment Selector](#) 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 \  
  
"FragmentSelectorType=SERVER_TIMESTAMP,TimestampRange={StartTimestamp=1708471800}"
```

On-demand

對於隨需播放，HLS 媒體播放清單包含 HLS 片段選取器指定的媒體。在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示清除器控制項，以選擇要在播放視窗中顯示的位置。

若要為串流的特定區段建立 URL，請先判斷開始和結束時間戳記。

在此範例中，我們使用秒格式的 Unix Epoch 時間。如需[時間戳記](#)格式的詳細資訊，請參閱 AWS Command Line Interface 《使用者指南》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 2024 年 2 月 20 日下午 3 : 30 : 00 GMT-08 : 00
- 1708471860 等於 2024 年 2 月 20 日下午 3 : 31 : 00 GMT-08 : 00

使用 ON_DEMAND 播放模式和指定的 [HLS Fragment Selector](#) 叫用 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, EndTime
```

 Note

時間戳記必須在彼此的 24 小時內，如 [the section called “HLSTimestampRange”](#) 文件所述。

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...DkREGM~"
}
```

 Important

不要在未經授權的實體可以存取它的地方共用或存放此字符。權杖可讓您存取串流的內容。使用與 AWS 登入資料相同的量值來保護字符。

您可以使用此 URL 和任何 HLS 播放器來檢視 HLS 串流。

例如，使用 VLC 媒體播放器。

您也可以在 Apple Safari 或 Microsoft Edge 瀏覽器的位置列中輸入 HLS 串流工作階段 URL，以播放 HLS 串流。

範例：在 HTML 和 JavaScript 中使用 HLS

下列範例示範如何使用適用於 JavaScript 的 AWS SDK v2 擷取 Kinesis 影片串流的 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 GitHub 中檢視[完整的範例程式碼](#)和[託管網頁](#)。此靜態網頁可簡化測試和實驗來自 Amazon Kinesis 影片串流的 HLS 和 MPEG-DASH 輸出。範例頁面提供下列參數的輸入欄位：

- AWS 區域：Kinesis 影片串流所在的區域
- 串流名稱：Kinesis 影片串流的名稱
- 播放模式：HLS 播放模式 (LIVE、LIVE_REPLAY 或 ON_DEMAND)
- 片段選取器類型：用於選取片段的方法 (SERVER_TIMESTAMP 或 PRODUCER_TIMESTAMP)
- 片段編號：開始片段編號（如適用）
- 容器格式：媒體容器的格式 (FRAGMENTED_MP4 或 MPEG_TS)

應用程式會從 HTML 頁面上的輸入方塊中擷取這些值，並使用它們來建構頁面上顯示的 HLS 串流工作階段請求。

程式碼演練主題：

- [匯入適用於瀏覽器 AWS 的 JavaScript 開發套件](#)
- [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 串流](#)

匯入適用於瀏覽器 AWS 的 JavaScript 開發套件

在 網頁中，包含下列指令碼標籤，將適用於 JavaScript 的 AWS SDK v2 匯入專案。

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

如需詳細資訊，請參閱適用於 [AWS JavaScript 的 SDK](#) 文件。

設定 Kinesis Video Streams 用戶端

若要使用 HLS 存取串流影片，請先建立和設定 Kinesis Video Streams 用戶端。如需其他身分驗證方法，請參閱在 [Web 瀏覽器中設定登入](#) 資料。

```
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 用戶端來叫用 [the section called “GetDataEndpoint”](#) API 來擷取端點。

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

此程式碼會將端點存放在 `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 封存媒體用戶端來叫用 [the section called “GetHLSStreamingSessionURL”](#) API 以擷取 HLS 播放 URL。

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

在網頁上顯示 HLS 串流

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

執行下列動作，將 [Video.js](#) 及其 CSS 類別匯入至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>
```

建立 video HTML 元素以顯示影片：

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

將 HLS URL 設定為 HTML 影片元素來源：

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

Shaka

執行下列動作，將 [Google Shaka 播放器](#) 匯入我們的瀏覽器指令碼：

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

建立 video HTML 元素以顯示影片：

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

建立指定影片元素的 Shaka 播放器並呼叫載入方法：

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

hls.js

執行下列動作，將 [hls.js](#) 匯入至我們的瀏覽器指令碼：

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

建立 video HTML 元素以顯示影片：

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

建立 hls.js 播放器，為其提供 HLS URL，並指示其播放：

```
const playerElement = document.getElementById('hlsjs');
const player = new Hls();
```

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

故障診斷 HLS 問題

本節說明將 HTTP 即時串流 (HLS) 與 Kinesis Video Streams 搭配使用時可能遇到的問題。

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：

- 判斷影片串流是否在 Kinesis Video Streams 主控台播放。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，服務可能會調節播放器，因為它對影片片段的請求太頻繁。
- 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 和自訂播放器) 可能僅支援具有 MPEG TS 容器格式的 HLS。建議您試用 ContainerFormat 參數，以開始故障診斷。

- 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擁有音訊和視訊音軌，以及只有視訊音軌之間沒有變更。同時確認編碼器設定（解析度和影格率）在每個軌跡中的片段之間沒有變更。

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 – 片段持續時間是由影片編碼器產生的關鍵影格頻率所控制串流中分割之間的影片量。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：位元速率較低的影片串流讀取、寫入和傳輸所需的時間較少。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 – 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。大多數玩家改為從最早的片段開始。透過減少播放清單中的片段數量，您可以減少先前片段和新片段之間的時間間隔。如果延遲將新片段新增至播放清單，或延遲玩家取得更新後的播放清單，則播放期間可以略過片段。建議使用 3-5 個片段，並使用設定為僅從播放清單中載入最新片段的玩家。
- 播放器緩衝大小 – 大多數影片播放器具有可設定的最短緩衝持續時間，通常預設為 10 秒。如需將延遲降到最低，您可以將此值設為 0 秒。不過，這樣做表示如果有任何延遲產生片段，則玩家會重新緩衝，因為玩家將沒有緩衝來吸收延遲。
- 播放器「追上」 – 如果緩衝區填滿，影片播放器通常不會自動擷取播放到影片緩衝區的正面，例如延遲片段導致片段的待處理項目播放時。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 影片串流，請先使用 [GetDASHStreamingSessionURL](#) 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 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 影片串流對於透過 MPEG-DASH 提供影片有下列要求：

- 如需串流影片播放追蹤需求，請參閱 [the section called “GetDASHStreamingSessionURL”](#)。
- 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](#)。

範例：在 HTML 和 JavaScript 使用 MPEG-DASH

下列範例示範如何擷取 Kinesis 影片串流的 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 Kinesis Video Streams 封存的內容端點以進行 MPEG-DASH 播放](#)
- [擷取 MPEG-DASH 串流工作階段 URL](#)
- [使用 MPEG-DASH 播放顯示串流影片](#)
- [已完成的範例](#)

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

若要使用 MPEG-DASH 存取串流影片，請先建立和設定 Kinesis Video Streams 用戶端（擷取服務端點）和封存媒體用戶端（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 Kinesis Video Streams 封存的內容端點以進行 MPEG-DASH 播放

啟動用戶端後，請擷取 Kinesis Video Streams 封存內容端點，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

當您有封存的內容端點時，請呼叫 [GetDASHStreamingSessionURL](#) 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);
}

```

使用 MPEG-DASH 播放顯示串流影片

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');
}
```

已完成的範例

您可以在 GitHub [下載或檢視完整的範例程式碼](#)。

在 Kinesis Video Streams 中設定通知

當媒體片段可供取用時，Kinesis Video Streams 會使用 Amazon Simple Notification Service (Amazon SNS) 通知來通知客戶。

Note

Amazon Kinesis Video Streams 使用 Amazon SNS 標準主題進行通訊。目前不支援 FIFO 主題。

下列主題說明如何開始使用通知。

主題

- [管理通知組態](#)
- [關於生產者 MKV 標籤](#)
- [Amazon SNS 訊息](#)

管理通知組態

若要管理通知組態，請使用 `UpdateNotificationConfiguration` 和 `DescribeNotificationConfiguration`。如需詳細資訊，請參閱下方。

UpdateNotificationConfiguration

使用此 API 操作更新串流的通知資訊。如需 `UpdateNotificationConfiguration` 此功能的詳細資訊，請參閱《Amazon Kinesis Video Streams 開發人員指南》中的 [UpdateNotificationConfiguration](#)。

Note

更新通知組態後，至少需要一分鐘的時間來啟動通知。等待至少一分鐘，再叫用 `PutMedia` 更新呼叫。

DescribeNotificationConfiguration

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

關於生產者 MKV 標籤

您可以使用 Kinesis Video Streams 生產者 SDK，透過在 SDK 中公開發 API 操作來標記特定感興趣的片段。請參閱 [本節程式碼中](#) 運作方式的範例。呼叫此 API 時，軟體開發套件會新增一組預先定義的 MKV 標籤以及片段資料。Kinesis Video Streams 會辨識這些特殊 MKV 標籤，並啟動已標記片段的通知。

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

生產者 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。如果中繼資料標籤限制可調整，您可以透過帳戶管理員請求增加。

限制	最大值	可調整
選用中繼資料金鑰長度	128	否

限制	最大值	可調整
選用中繼資料值長度	256	否
選用中繼資料的數量上限	10	是

Amazon SNS 訊息

本主題包含有關 Amazon SNS 訊息和主題承載的詳細資訊。

主題

- [Amazon SNS 主題承載](#)
- [檢視您的 Amazon SNS 訊息](#)

Amazon SNS 主題承載

透過先前工作流程啟動的任何通知都會交付 Amazon SNS 主題承載，如下列範例所示。此範例是在從 Amazon Simple Queue Service (Amazon SQS) 佇列耗用通知資料後發生的 Amazon SNS 訊息。
Amazon SQS

```
{
  "Type" : "Notification",
  "MessageId" : Message ID,
  "TopicArn" : SNS ARN,
  "Subject" : "Kinesis Video Streams Notification",
  "Message" : "{\"StreamArn\": \"Stream Arn\", \"FragmentNumber\": \"Fragment Number\",
  \"FragmentStartProducerTimestamp\": \"FragmentStartProducerTimestamp\",
  \"FragmentStartServerTimestamp\": \"FragmentStartServerTimestamp\",
  \"NotificationType\": \"PERSISTED\", \"NotificationPayload\": {\" CUSTOM_KEY_1:
  \"CUSTOM_VALUE_1\",
  \"CUSTOM_KEY_2: \"CUSTOM_VALUE_2\"}}",
  "Timestamp" : "2022-04-25T18:36:29.194Z",
  "SignatureVersion" : Signature Version,
  "Signature" : Signature,
  "SigningCertURL" : Signing Cert URL,
  "UnsubscribeURL" : Unsubscribe URL
}
```

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

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 可供讀取。若要檢視訊息，請訂閱 SQS 佇列至 SNS 主題，或選擇任何其他 [Amazon SNS 支援的目的地](#)。不過，檢視訊息最有效率的選項是使用 Amazon SQS。

使用 Amazon SQS 檢視 Amazon SNS 訊息 Amazon SQS

1. 建立 [Amazon SQS 佇列](#)。
2. 從 AWS Management Console，開啟 Amazon SNS 主題集，做為下的目的地 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. 在啟用通知組態並新增通知 MKV 標籤至片段的情況下執行 PutMedia 工作階段。
5. 在 Amazon SQS 主控台中選擇 Amazon SQS 佇列，然後選取傳送和接收 Amazon SQS 佇列的訊息。Amazon SQS
6. 訊息輪詢。此命令應會顯示 PutMedia 工作階段產生的所有通知。如需輪詢的相關資訊，請參閱 [Amazon SQS 短輪詢和長輪詢](#)。

從影片串流擷取影像

您可以使用 Amazon Kinesis Video Streams APIs 和 SDKs，協助您從影片串流擷取映像。您可以使用這些影像來增強播放應用程式，例如縮圖或增強的清理，或用於機器學習管道。Kinesis Video Streams 透過 API 提供隨需映像擷取，或從擷取的影片中的中繼資料標籤自動擷取映像。

如需使用 Kinesis Video Streams 受管影像支援的詳細資訊，請參閱：

- [隨需影像產生 \(GetImages\)](#) - 此 API 可讓客戶從存放在 Kinesis Video Streams 中的影片擷取單一影像或多個影像。
- [the section called “自動化映像產生 \(Amazon S3 交付\)”](#) - 設定 Kinesis Video Streams 以根據上傳影片中的標籤即時從影片資料中自動擷取影像，並將影像交付至客戶指定的 S3 儲存貯體。

自動化映像產生 (Amazon S3 交付)

目前，客戶會執行和管理自己的映像轉碼管道，以建立映像，用於各種用途，例如清除、映像預覽、在映像上執行 ML 模型等。Kinesis Video Streams 提供轉碼和交付映像的功能。Kinesis Video Streams 會根據標籤即時從影片資料中擷取影像，並將影像交付給客戶指定的 S3 儲存貯體。

主題

- [UpdateImageGenerationConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [生產者 MKV 標籤](#)
- [使用 PutEventMetaData 在生產者 SDK 中新增中繼資料標籤](#)
- [限制](#)
- [S3 物件中繼資料](#)
- [Amazon S3 物件路徑 \(影像\)](#)
- [Amazon S3 URI 建議，以防止限流](#)

UpdateImageGenerationConfiguration

若要設定 Kinesis 影片串流以啟用對 Amazon S3 產生映像：

1. 使用新的 API，根據 SDK 中新增的標籤建立用於產生映像的 S3 儲存貯體。請注意，在更新串流的影像產生組態時，下一步需要 S3 URI。

2. 使用下列內容做為輸入，建立名為 update-image-generation-input.json 的 JSON 檔案。

```
{
  "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 來叫用 [UpdateImageGenerationConfiguration](#) API 操作，以新增先前建立的 Amazon S3 ARN，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。等待至少 1 分鐘再叫用更新呼叫PutMedia。

DescribeImageGenerationConfiguration

若要檢視已為串流設定的影像產生組態，客戶可以提出DescribeImageGenerationConfiguration請求，如下所示。

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'
  }
}
```

若要進一步了解DescribeImageGenerationConfiguration此功能，請參閱《Amazon Kinesis Video Streams 開發人員指南》中的 [DescribeImageGenerationConfiguration](#)。Amazon Kinesis Video Streams

生產者 MKV 標籤

您可以使用 Kinesis Video Streams 生產者 SDK，透過在 SDK 中公開發 API 操作來標記特定感興趣的片段。如需標籤的範例，請參閱[此程式碼](#)。呼叫此 API 時，軟體開發套件會新增一組預先定義的 MKV 標

籤以及片段資料。Kinesis Video Streams 會辨識這些特殊 MKV 標籤，並根據該串流的影像處理組態啟動影像產生工作流程。

Amazon S3 影像產生標籤隨附的任何片段中繼資料都會儲存為 Amazon S3 中繼資料。

生產者 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
```

使用 PutEventMetaData 在生產者 SDK 中新增中繼資料標籤

PutEventMetaData 函數會附加與事件相關聯的 MKV 檔案。PutEventMetaData 會採用兩個參數。第一個參數是事件，其值來自 STREAM_EVENT_TYPE 列舉。第二個參數是選用的 [pStreamEventMetadata](#)，可用來包含其他中繼資料做為索引鍵/值對。可以新增的中繼資料金鑰值對限制為 5 個。

限制

下表列出與中繼資料標籤相關聯的限制。如果中繼資料標籤限制可調整，您可以透過帳戶管理員請求增加。

限制	最大值	可調整
影像字首長度	256	編號
選用中繼資料金鑰長度	128	編號
選用中繼資料值長度	256	編號
選用中繼資料的數量上限	10	是

S3 物件中繼資料

根據預設，Kinesis Video Streams 會設定產生為 Amazon S3 物件中繼資料之映像的片段編號、生產者和伺服器時間戳記。如果在 MKV 標籤中指定任何其他片段資料，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,
}
```

Amazon 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_AccountID_StreamName_ImageTimecode_RandomID.file-extension

1. ImagePrefix - 的值 AWS_KINESISVIDEO_IMAGE_PREFIX。

2. AccountID - 建立串流的帳戶 ID。

3. StreamName - 產生映像的串流名稱。
4. ImageTimecode - 產生映像的片段中的 Epoch 時間碼。
5. RandomID - 隨機 GUID。
6. file-extension - JPG 或 PNG，根據請求的影像格式。

Amazon S3 URI 建議，以防止限流

如果您將數千張影像寫入 Amazon S3，會有限流風險。如需詳細資訊，請參閱 [S3 字首賣權請求限制](#)。

Amazon S3 字首以每秒 3,500 個 PUT 請求的 PUT 限制開始，並會逐漸增加唯一字首的速率。避免使用日期和時間做為 Amazon S3 字首。時間編碼資料一次會影響一個字首，也會定期變更，使先前的字首擴展失效。若要啟用更快速、一致的 Amazon S3 擴展，我們建議您將隨機字首，例如十六進位碼或 UUID 新增至 Amazon S3 目的地 URI。例如，十六進位碼字首會自然地將您的請求隨機分割為 16 個不同的字首（每個唯一的十六進位字元的字首），在 Amazon S3 自動擴展之後，每秒允許 56,000 個 PUT 請求。

存取影片分析

本節包含如何使用剖析器程式庫和 Amazon CloudWatch 存取影片分析的相關資訊。

主題

- [使用 Kinesis 影片串流中內嵌的中繼資料](#)
- [使用剖析器程式庫觀察攝影機的輸出](#)
- [監控 Amazon Kinesis Video Streams](#)
- [串流中繼資料限制](#)

使用 Kinesis 影片串流中內嵌的中繼資料

若要使用 Kinesis 影片串流中的中繼資料，請使用 的實作 `MkvTagProcessor`：

```
public interface MkvTagProcessor {
    default void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
    default void clear() {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
}
```

此界面位於 [使用剖析器程式庫觀察攝影機的輸出](#) 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，請先呼叫 `clear` 清除收集的中繼資料，然後再次擷取中繼資料項目。

使用剖析器程式庫觀察攝影機的輸出

Kinesis 影片串流剖析器程式庫是一組工具，可用於 Java 應用程式，以使用 Kinesis 影片串流中的 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 影片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。

先決條件

您必須具備下列項目，才能檢查和使用 Kinesis 影片串流剖析器程式庫：

- Amazon Web Services (AWS) 帳戶。如果您還沒有 AWS 帳戶，請參閱 [the section called “註冊 AWS 帳戶”](#)。
- Java 整合開發環境 (IDE)，例如 [Eclipse Java Neon](#) 或 [JetBrains IntelliJ Idea](#)。
- Java 11，例如 [Amazon Corretto 11](#)。

下載程式碼

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 [the section called “使用剖析器程式庫進行串流”](#)。

1. 建立目錄，並從 GitHub 儲存庫複製程式庫原始程式碼 (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>)。

```
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](#))，並匯入您下載的 Apache Maven 專案：
 - 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。

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 影片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，例如轉碼器私有資料、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().getElementMetaData().getTypeInfo()))
{
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame =
((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
            MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
```

```

        assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
    }
    if
(MkvTypeInfos.SEGMENT.equals(mkvElement.get().getElementMetaData().getTypeInfo())) {
        if (mkvElement.get() instanceof MkvEndMasterElement) {
            if (segmentCount < continuationTokens.size()) {
                Optional<String> continuationToken =
fragmentVisitor.getContinuationToken();
                Assert.assertTrue(continuationToken.isPresent());
                Assert.assertEquals(continuationTokens.get(segmentCount),
continuationToken.get());
            }
            segmentCount++;
        }
    }
}
}
}
}
}
}
}
}

```

前述範例顯示下列編碼模式：

- 建立 `FragmentMetadataVisitor` 剖析資料，以及建立 [StreamingMkvReader](#) 提供資料。
- 對於串流之中的各個 `MkvElement`，請測試其中繼資料是否為 `SIMPLEBLOCK` 類型。
- 如果是，請由 `MkvElement` 擷取 `MkvDataElement`。
- 由 `MkvDataElement` 擷取 `Frame` (媒體資料)。
- 由 `FragmentMetadataVisitor` 擷取 `MkvTrackMetadata` 用於 `Frame`。
- 由 `Frame` 及 `MkvTrackMetadata` 物件擷取和驗證下列資料：
 - 軌道編號。
 - 影格像素高度。
 - 影格像素寬度。
 - 用於編碼影格轉碼器的轉碼器 ID。
 - 此影格依序抵達。確認前一個影格的軌跡編號是否存在，小於目前影格的軌跡編號。

如欲在專案使用 `FragmentMetadataVisitor`，請將 `MkvElement` 物件以其 `accept` 方式傳送至訪客：

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

此類別將串流之中不同軌道的中繼資料合併，成為單一區段的串流。

下列程式碼範例 (來自 `FragmentMetadataVisitorTest` 檔案) 顯示如何使用 `OutputSegmentMerger`，由名為 `inputBytes` 的位元組陣列合併軌道中繼資料：

```
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 影片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 影片串流。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](#) 將影片片段串流至 Kinesis 影片串流。
- 呼叫 [GetMedia](#) 將影片片段串流出 Kinesis 影片串流。
- 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）會刪除指定的 Kinesis 影片串流：

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

下列程式碼範例（從 `StreamOps.java` 檔案）會建立具有指定名稱的 Kinesis 影片串流：

```
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）在串流呼叫 [PutMedia](#)：

```
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) 在串流呼叫 [GetMedia](#) :

```
GetMediaResult result = videoMedia.getMedia(new
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

剖析 GetMedia 結果

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。

以 `StreamingMkvReader` 讀取 `GetMedia` 輸出

下列程式碼範例 (來自 `GetMediaWorker.java` 檔案) 建立 [StreamingMkvReader](#)，並將其用於剖析 [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。在下節之中，元素將傳送至 [FragmentMetadataVisitor](#)。

使用 `FragmentMetadataVisitor` 擷取片段

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。
- 將 [FragmentMetadataVisitor](#)、[the section called “OutputSegmentMerger”](#) 和 組成 `LogVisitor` 單一元素訪客 `CompositeMkvElementVisitor` 的。

```
//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，媒體處理引數會傳遞至 `GetMediaWorker`，再傳遞至 `ExecutorService`，而會在不同的執行緒上執行工作者：

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getMkvElementVisitor());
executorService.submit(getMediaWorker);
```

執行程式碼

Kinesis 影片串流剖析器程式庫包含可供您用於自己專案的工具。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

監控 Amazon Kinesis Video Streams

監控是維護 Amazon Kinesis Video Streams 和您的 AWS 解決方案可靠性、可用性和效能的重要部分。我們建議您從 AWS 解決方案的所有部分收集監控資料，以便在發生多點故障時協助您偵錯。在

您開始監控 Amazon Kinesis Video Streams 之前，我們建議您建立監控計畫，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？

在您定義監控目標並建立監控計畫之後，下一步是建立您環境中正常 Amazon Kinesis Video Streams 效能的基準。您應該在不同時間以及不同的載入條件下測量 Amazon Kinesis Video Streams 效能。當您監控 Amazon Kinesis Video Streams 時，請存放您收集的監控資料歷史記錄。您可以比較目前的 Amazon Kinesis Video Streams 效能與此歷史資料，以協助您識別正常效能模式和效能異常，並設計方法來解決可能發生的問題。

主題

- [使用 CloudWatch 監控 Amazon Kinesis Video Streams 指標](#)
- [使用 CloudWatch 監控 Amazon Kinesis Video Streams Edge 代理程式](#)
- [使用 記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail](#)

使用 CloudWatch 監控 Amazon Kinesis Video Streams 指標

您可以使用 Amazon CloudWatch 監控 Kinesis 影片串流，該串流會收集原始資料，並將 Amazon Kinesis Video Streams 處理為可讀且近乎即時的指標。這些統計資料會記錄 15 個月的時間，以便您可以存取歷史資訊，並更好地了解 Web 應用程式或服務的表現。

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，選擇目前區域帳戶層級指標區段中的影片串流索引標籤。
- 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	<p>所有 APIs 消耗的片段媒體配額點數量。如需配額點概念的說明，請參閱 the section called “片段中繼資料和片段媒體配額”。</p> <p>單位：計數</p>
ArchivedFragmentsConsumed.Metadata	<p>所有 APIs 使用的片段中繼資料配額點數量。如需配額點概念的說明，請參閱 the section called “片段中繼資料和片段媒體配額”。</p> <p>單位：計數</p>
PutMedia.Requests	<p>指定串流的 PutMedia API 請求數量。</p> <p>單位：計數</p>
PutMedia.IncomingBytes	<p>作為串流一部分接收PutMedia的位元組數。</p> <p>單位：位元組</p>
PutMedia.IncomingFragments	<p>作為 PutMedia 串流一部分接收的完整片段數量。</p> <p>單位：計數</p>
PutMedia.IncomingFrames	<p>作為 PutMedia 串流一部分收到的完整影格數量。</p> <p>單位：計數</p>
PutMedia.ActiveConnections	<p>服務主機的連線總數。</p> <p>單位：計數</p>
PutMedia.ConnectionErrors	<p>建立串流PutMedia連線時發生錯誤。</p> <p>單位：計數</p>
PutMedia.FragmentIngestionLatency	<p>Amazon Kinesis Video Streams 收到片段的第一個位元組和最後一個位元組之間的時間差。</p>

指標	描述
	單位：毫秒
PutMedia.FragmentPersistLatency	接收和封存完整片段資料所花費的時間。 單位：計數
PutMedia.Latency	建立連線時，請求與來自 InletService 的 HTTP 回應之間的時間差。 單位：計數
PutMedia.BufferingAckLatency	Amazon Kinesis Video Streams 接收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。 單位：毫秒
PutMedia.ReceivedAckLatency	Amazon Kinesis Video Streams 收到新片段的最後位元組與傳送片段的 Received ACK 之間的時間差。 單位：毫秒
PutMedia.PersistedAckLatency	Amazon Kinesis Video Streams 收到新片段的最後位元組與傳送片段持久性 ACK 之間的時間差。 單位：毫秒
PutMedia.ErrorAckCount	PutMedia 為串流執行時傳送的錯誤 ACKs 數量。 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
GetMedia.Requests	指定串流的 GetMedia API 請求數。 單位：計數

指標	描述
GetMedia.OutgoingBytes	從服務傳送的位元組總數，做為指定串流 GetMedia API 的一部分。 單位：位元組
GetMedia.OutgoingFragments	GetMedia 為串流執行時傳送的片段數量。 單位：計數
GetMedia.OutgoingFrames	在指定串流 GetMedia 上於 期間傳送的影格數。 單位：計數
GetMedia.MillisBehindNow	目前伺服器時間戳記與傳送之最後一個片段的伺服器時間戳記之間的時間差。 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 1251 1507 1562" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> 單位：計數

指標	描述
GetMediaForFragmentList.OutgoingBytes	從服務傳送的位元組總數，做為指定串流 GetMediaForFragmentList API 的一部分。 單位：位元組
GetMediaForFragmentList.OutgoingFragments	從服務傳送的片段總數，做為指定串流 GetMediaForFragmentList API 的一部分。 單位：計數
GetMediaForFragmentList.OutgoingFrames	從服務傳送的影格總數，做為指定串流 GetMediaForFragmentList API 的一部分。 單位：計數
GetMediaForFragmentList.Requests	指定串流的 GetMediaForFragmentList API 請求數量。 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 #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。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> 單位：計數
ListFragments.Latency	ListFragments API 的延遲會呼叫給定的串流名稱。 單位：毫秒

指標	描述
ListFragments.Requests	指定串流的 ListFragments API 請求數量。 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> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> 單位：計數
GetHLSStreamingSessionURL.Latency	GetHLSStreamingSessionURL API 的延遲會呼叫給定的串流名稱。 單位：毫秒
GetHLSStreamingSessionURL.Requests	指定串流的 GetHLSStreamingSessionURL API 請求數量。 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 style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> <p>單位：計數</p>
GetHLSMasterPlaylist.Latency	<p>GetHLSMasterPlaylist API 的延遲會呼叫給定的串流名稱。</p> <p>單位：毫秒</p>
GetHLSMasterPlaylist.Requests	<p>指定串流的 GetHLSMasterPlaylist API 請求數量。</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 style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> <p>單位：計數</p>

指標	描述
GetHLSMediaPlaylist.Latency	GetHLSMediaPlaylist API 的延遲會呼叫給定的串流名稱。 單位：毫秒
GetHLSMediaPlaylist.Requests	指定串流的 GetHLSMediaPlaylist API 請求數量。 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 #00a0e3; 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。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> 單位：計數
GetMP4InitFragment.Latency	GetMP4InitFragment API 的延遲會呼叫給定的串流名稱。 單位：毫秒
GetMP4InitFragment.Requests	指定串流的 GetMP4InitFragment API 請求數量。 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 1508 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> <p>單位：計數</p>
GetMP4MediaFragment.Latency	<p>GetMP4MediaFragment API 的延遲會呼叫給定的串流名稱。</p> <p>單位：毫秒</p>
GetMP4MediaFragment.Requests	<p>指定串流的 GetMP4MediaFragment API 請求數量。</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 1508 1675" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> <p>單位：計數</p>

指標	描述
GetMP4MediaFragment.OutgoingBytes	<p>從服務傳送的位元組總數，做為指定串流 GetMP4MediaFragment API 的一部分。</p> <p>單位：位元組</p>
GetTSFragment.Latency	<p>GetTSFragment API 的延遲會呼叫給定的串流名稱。</p> <p>單位：毫秒</p>
GetTSFragment.Requests	<p>指定串流的 GetTSFragment API 請求數量。</p> <p>單位：計數</p>
GetTSFragment.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="748 930 1507 1241" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p> </div> <p>單位：計數</p>
GetTSFragment.OutgoingBytes	<p>從服務傳送的位元組總數，做為指定串流 GetTSFragment API 的一部分。</p> <p>單位：位元組</p>
GetDASHStreamingSessionURL.Latency	<p>GetDASHStreamingSessionURL API 的延遲會呼叫給定的串流名稱。</p> <p>單位：毫秒</p>

指標	描述
GetDASHStreamingSessionURL. Requests	指定串流的 GetDASHStreamingSessionURL API 請求數量。 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> Note</p><p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p></div> 單位：計數
GetDASHManifest.Latency	GetDASHManifest API 的延遲會呼叫給定的串流名稱。 單位：毫秒
GetDASHManifest.Requests	指定串流的 GetDASHManifest API 請求數量。 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 data-bbox="748 352 1508 667"><p> Note</p><p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p></div> <p>單位：計數</p>
GetClip.Latency	<p>指定影片串流名稱的 GetClip API 呼叫延遲。</p> <p>單位：毫秒</p>
GetClip.Requests	<p>特定影片串流的 GetClip API 請求數量。</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 data-bbox="748 1262 1508 1577"><p> Note</p><p>失敗包括 400 個 (使用者) 錯誤和 500 個 (系統) 錯誤。如需啟用請求和回應摘要的詳細資訊，包括 AWS 請求 IDs，請參閱請求/回應摘要記錄。</p></div> <p>單位：計數</p>

指標	描述
GetClip.OutgoingBytes	做為指定影片串流 GetClip API 的一部分，從服務傳送的位元組總數。 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

- [資料是否到達 Amazon Kinesis Video Streams 服務？](#)
- [為什麼 Amazon Kinesis Video Streams 服務未成功擷取資料？](#)
- [為什麼無法以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 影片串流讀取資料，並以什麼速率讀取資料？](#)
- [為什麼用戶端無法從 Kinesis 影片串流讀取資料？](#)

資料是否到達 Amazon Kinesis Video Streams 服務？

相關指標：

- PutMedia.IncomingBytes
- PutMedia.IncomingFragments
- PutMedia.IncomingFrames

動作項目：

- 如果這些指標下降，請檢查您的應用程式是否仍在將資料傳送至服務。
- 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 服務未成功擷取資料？

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 回應和錯誤代碼，以查看建立連線時發生哪些錯誤。
- 如果 出現下降PutMedia.Success或增加PutMedia.ErrorAckCount，請查看服務傳送的 ack 回應中的 ack 錯誤碼，以了解資料擷取失敗的原因。如需詳細資訊，請參閱 [AckErrorCode.Values](#)。

為什麼無法以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或減少PutMedia.IncomingFragments，請檢查網路頻寬以及資料是否仍在傳送中。
- 如果 中有下降PutMedia.Success，請檢查 ack 錯誤碼。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。如果條件長時間持續存在，請洽詢您的客戶服務聯絡人，以了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，消費者可能會落後於讀取串流，因為經常嘗試重新連線至串流。查看 GetMedia 請求傳回的 HTTP 回應/錯誤代碼。
- 如果 出現下降GetMedia.Success，可能是由於服務無法將資料傳送給消費者，這會導致連線中斷，以及消費者重新連線，這會導致消費者落後於串流的前端。
- 如果 增加GetMedia.MillisBehindNow，請查看您的頻寬限制，查看您是否因為頻寬較低而以較慢的速度接收資料。

用戶端是否從 Kinesis 影片串流讀取資料，並以什麼速率讀取資料？

相關指標：

- GetMedia.OutgoingBytes
- GetMedia.OutgoingFragments
- GetMedia.OutgoingFrames
- GetMediaForFragmentList.OutgoingBytes
- GetMediaForFragmentList.OutgoingFragments

- `GetMediaForFragmentList.OutgoingFrames`

動作項目：

- 這些指標指出即時和封存資料的讀取速率。

為什麼用戶端無法從 Kinesis 影片串流讀取資料？

相關指標：

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

動作項目：

- 如果增加 `GetMedia.ConnectionErrors`，請查看 `GetMedia` 請求傳回的 HTTP 回應和錯誤代碼。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，請檢查串流 `PutMedia.IncomingBytes` 中是否有資料可供服務傳送給消費者。
- 如果 `GetMedia.Success` 或出現下降 `GetMediaForFragmentList.Success`，可能是由於服務無法將資料傳送給消費者。如果條件長時間持續存在，請洽詢您的客戶服務聯絡人，以了解您的服務是否有問題。

使用 CloudWatch 監控 Amazon Kinesis Video Streams Edge 代理程式

您可以使用 Amazon CloudWatch 監控 Amazon Kinesis Video Streams Edge Agent，該程式會收集原始資料並將其處理為可讀且近乎即時的指標。Amazon CloudWatch 這些統計資料會記錄 15 個月的時間。透過此歷史資訊，您可以更清楚地了解 Web 應用程式或 Amazon Kinesis Video Streams Edge Agent 服務的效能。

若要檢視指標，請執行下列動作：

1. 登入 AWS Management Console，並在 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。

2. 在左側導覽的指標下，選取所有指標。
3. 選擇瀏覽索引標籤，然後選擇 EdgeRuntimeAgent 自訂命名空間。

Amazon Kinesis Video Streams Edge Agent 會在命名空間 下發佈下列指標EdgeRuntimeAgent：

維度	州	描述
串流名稱、RecordJob	執行中	<p>在 RecordJob 執行時持續發佈。</p> <p>單位：無。只要RecordJob 處於此狀態，「1」就會發佈。</p>
	FatalError	<p>如果RecordJob 嚴重錯誤，則發佈。</p> <p>單位：無。當發生此事件時，「1」會發佈一次。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>如需其他資訊，請參閱日誌。</p> </div>
	已完成	<p>RecordJob 完成時發佈。</p> <p>單位：無。當發生此事件時，「1」會發佈一次。</p>
串流名稱、UploadJob	執行中	<p>在 UploadJob 執行時持續發佈。</p> <p>單位：無。只要UploadJob 處於此狀態，「1」就會發佈。</p>
	FatalError	<p>如果UploadJob 嚴重錯誤，則發佈。</p> <p>單位：無。當發生此事件時，「1」會發佈一次。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>如需其他資訊，請參閱日誌。</p> </div>
	已完成	<p>UploadJob 完成時發佈。</p> <p>單位：無。當發生此事件時，「1」會發佈一次。</p>

維度	州	描述
串流名稱	PercentageSpaceUsed	<p>這是在錄製媒體的 Amazon Kinesis Video Streams Edge Agent 組態中配置的總空間中所使用的百分比。如需更多資訊，請參閱the section called "LocalSizeConfig"。</p> <p>單位：百分比（比例 0–1）。</p>
實物名稱	Alive	<p>每分鐘從 Amazon Kinesis Video Streams Edge 代理程式發佈，無論其上執行的任何組態為何。</p> <p>這可用來了解 Amazon Kinesis Video Streams Edge 代理程式是否存活並準備好接受組態。</p> <p>單位：無。「1」每分鐘發佈一次。</p>
	RecordJobs.HealthyJobCount	<p>在 Amazon Kinesis Video Streams Edge Agent 上執行和排程記錄任務的總計數。</p> <p>單位：計數。</p>
	UploadJobs.HealthyJobCount	<p>在 Amazon Kinesis Video Streams Edge Agent 上執行和排程上傳任務的總計數。</p> <p>單位：計數。</p>
	RecordJobs.UnhealthyJobCount	<p>目前錯誤的記錄任務總數。</p> <p>單位：計數。</p>
	UploadJobs.UnhealthyJobCount	<p>目前錯誤上傳任務的總計數。</p> <p>單位：計數。</p>
	RecordJobs.RunningJobCount	<p>主動執行的記錄任務總數。</p> <p>單位：計數。</p>

維度	州	描述
	UploadJobs.RunningJobCount	主動執行上傳任務的總計數。 單位：計數。
	RecordJobs.EdgeConfigCount	Amazon Kinesis Video Streams Edge Agent 上處理中的記錄組態總數。 單位：計數。
	UploadJobs.EdgeConfigCount	在 Amazon Kinesis Video Streams Edge Agent 上進行中的上傳組態總數。 單位：計數。

Amazon Kinesis Video Streams Edge Agent 的 CloudWatch 指標指南

CloudWatch 指標對於尋找下列問題的答案很有用：

主題

- [Amazon Kinesis Video Streams Edge Agent 是否有足夠的空間可供記錄？](#)
- [Amazon Kinesis Video Streams Edge 代理程式是否有效？](#)
- [是否有任何運作狀態不佳的任務？](#)
- [是否有任何任務需要外部介入？](#)

Amazon Kinesis Video Streams Edge Agent 是否有足夠的空間可供記錄？

相關指標：PercentageSpaceUsed

動作：不需要任何動作。

Amazon Kinesis Video Streams Edge 代理程式是否有效？

相關指標：Alive

動作：如果您在任何時候停止接收此指標，這表示 Amazon Kinesis Video Streams Edge Agent 遇到下列一或多個問題：

- 應用程式執行期問題：記憶體或其他資源限制、錯誤等
- 代理程式在關機、當機或終止時正在執行 AWS IoT 的裝置
- AWS IoT 裝置沒有網路連線

是否有任何運作狀態不佳的任務？

相關指標：

- `RecordJobs.UnhealthyJobCount`
- `UploadJobs.UnhealthyJobCount`

動作：檢查日誌並尋找 `FatalError` 指標。

- 如果 `FatalError` 指標存在，則遇到嚴重錯誤，您需要手動重新啟動任務。使用 `StartEdgeConfigurationUpdate` 手動重新啟動任務之前，請先檢查日誌並修正問題。
- 如果 `FatalError` 指標不存在，則遇到暫時性（非嚴重）錯誤，Amazon Kinesis Video Streams Edge Agent 正在重試任務。

Note

若要讓客服人員重新嘗試嚴重發生錯誤的任務，請使用 [the section called “StartEdgeConfigurationUpdate”](#)。

是否有任何任務需要外部介入？

相關指標：

- `PercentageSpaceUsed` – 如果超過特定值，則記錄任務會暫停，並只在有可用空間時（媒體停止保留時）繼續。您可以傳送更新後的組態與更新的版本，`MaxLocalMediaSizeInMB` 以立即更新任務。
- `RecordJob.FatalError / UploadJob.FatalError` – 調查客服人員的日誌，然後再次傳送組態，讓任務繼續。

動作：使用組態進行 API 呼叫，以重新啟動遇到此問題的任務。

使用 記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail

Amazon Kinesis Video Streams 可與 搭配使用 AWS CloudTrail，此服務提供 Amazon Kinesis Video Streams AWS 服務 中使用者、角色或 所採取動作的記錄。CloudTrail 會將 Amazon Kinesis Video Streams 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 Amazon Kinesis Video Streams 主控台的呼叫，以及對 Amazon Kinesis Video Streams API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 Amazon Kinesis Video Streams 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 收集的資訊，判斷對 Amazon Kinesis Video Streams 提出的請求、提出請求的 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，包括如何設定及啟用，請參閱 [《AWS CloudTrail 使用者指南》](#)。

Amazon Kinesis Video Streams 和 CloudTrail

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當 Amazon Kinesis Video Streams 中發生支援的事件活動時，該活動會記錄於 CloudTrail 事件，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱《使用 CloudTrail 事件歷史記錄檢視事件》<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/view-cloudtrail-events.html>。

若要持續記錄您 AWS 帳戶中的事件，包括 Amazon Kinesis Video Streams 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 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。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務 來進一步分析 CloudTrail 日誌中收集的事件資料，並對其採取行動。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案](#)，以及 [從多個帳戶接收 CloudTrail 日誌檔案](#)

Amazon Kinesis Video Streams 支援將下列動作記錄為 CloudTrail 日誌檔案中的事件：

- [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 userIdentity 元素](#)。

範例：Amazon Kinesis Video Streams 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。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 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例顯示示範 [CreateStream](#) 動作的 CloudTrail 日誌項目。

```
{
  "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": "b7acfd0-6ca9-4ee1-a3d7-c4e8d420d99b"
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-05-25:17:06Z",
  "eventSource": " kinesisvideo.amazonaws.com",
  "eventName": "DeleteStream",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
  "requestParameters": {
    "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/
VideoStream/12345",
    "currentVersion": "keqrjeqkj9"
  },
  "responseElements": null,
  "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
  "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
},
{

```

```

    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:02Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "DescribeStream",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream"
    },
    "responseElements": null,
    "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:03Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "GetDataEndpoint",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream",
      "apiName": "LIST_FRAGMENTS"
    },
    "responseElements": null,

```

```
    "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2018-05-25T00:16:56Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "ListStreams",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "maxResults": 100,
      "streamNameCondition": {"comparisonValue": "MyVideoStream"
comparisonOperator": "BEGINS_WITH"}}
    },
    "responseElements": null,
    "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
    "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
  }
]
}
```

串流中繼資料限制

[the section called “串流中繼資料服務配額”](#) 如需將串流中繼資料新增至 Kinesis 影片串流之限制的詳細資訊，請參閱。

使用 Amazon Kinesis Video Streams Edge Agent 排程影片錄製和儲存

Amazon Kinesis Video Streams 提供一種高效、經濟實惠的方式來連接到客戶現場部署的 IP 攝影機。使用 Amazon Kinesis Video Streams Edge Agent，您可以根據客戶定義的排程，在本機記錄和存放攝影機影片，並將影片串流到雲端，以進行長期儲存、播放和分析處理。

Note

若要存取 Amazon Kinesis Video Streams Edge 代理程式，請完成此[簡短表單](#)。

您可以下載 Amazon Kinesis Video Streams Edge 代理程式，並將其部署到您的內部部署邊緣運算裝置。您也可以 Amazon EC2 執行個體上執行的 Docker 容器中輕鬆部署它們。部署之後，您可以使用 Amazon Kinesis Video Streams API 來更新影片錄製和雲端上傳組態。此功能適用於可透過 RTSP 通訊協定串流的任何 IP 攝影機。它不需要對攝影機進行任何額外的韌體部署。

我們為 Amazon Kinesis Video Streams Edge Agent 提供下列安裝：

- 作為 AWS IoT Greengrass V2 元件：您可以將 Amazon Kinesis Video Streams Edge 代理程式安裝為任何 AWS IoT Greengrass 已認證裝置上的 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 邊緣：您可以在 Snowball Edge 裝置上執行 Amazon Kinesis Video Streams Edge 代理程式。若要進一步了解，請參閱 [AWS Snowball Edge 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原生安裝 Amazon Kinesis Video Streams Edge Agent。Edge SDK 使用透過 [AWS IoT Core](#) 管理邊緣 [the section called “Amazon Kinesis Video Streams”](#)。

若要開始使用 Amazon Kinesis Video Streams Edge Agent，請繼續執行以下適當的程序。

主題

- [Amazon Kinesis Video Streams Edge Agent API 操作](#)
- [監控 Amazon Kinesis Video Streams Edge 代理程式](#)
- [在非 AWS IoT Greengrass 模式下部署](#)
- [將 Amazon Kinesis Video Streams Edge 代理程式部署至 AWS IoT Greengrass](#)

- [Amazon Kinesis Video Streams Edge 代理程式常見問答集](#)

Amazon Kinesis Video Streams Edge Agent API 操作

使用下列 API 操作來設定 Amazon Kinesis Video Streams Edge Agent :

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

監控 Amazon Kinesis Video Streams Edge 代理程式

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

在非AWS IoT Greengrass 模式下部署

本節提供在 AWS IoT Greengrass 環境外使用 Amazon Kinesis Video Streams 的完整指南。無論您是使用邊緣裝置或其他平台，此資訊都可協助您有效設定和利用 Kinesis Video Streams。

您可以找到下列詳細資訊：

- 設定開發環境
- 建立 Kinesis 影片串流
- 下載和編譯 Kinesis Video Streams Producer SDK
- 撰寫和檢查範例應用程式
- 執行範例應用程式

繼續執行下列步驟，以 AWS IoT MQTT 執行 Amazon Kinesis Video Streams Edge 代理程式做為獨立部署。

主題

- [安裝相依項目](#)
- [為您的 IP 攝影機 RTSP URLs 建立資源](#)

- [建立 IAM 許可政策](#)
- [建立 IAM 角色](#)
- [建立 AWS IoT 角色別名](#)
- [建立 AWS IoT 政策](#)
- [建立 AWS IoT 物件並取得 AWS IoT Core 登入資料](#)
- [建置 Amazon Kinesis Video Streams Edge 代理程式](#)
- [在裝置上安裝 CloudWatch 代理程式](#)
- [執行 Amazon Kinesis Video Streams Edge 代理程式作為原生程序](#)

安裝相依項目

您需要使用必要的相依性來設定開發環境，才能開始使用 Amazon Kinesis Video Streams 生產者 SDK。此頁面會引導您在系統上安裝所需的軟體元件和程式庫。

Note

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

在裝置上安裝相依性

1. 若要執行 Amazon Kinesis Video Streams Edge Agent，請在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://dlcdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-
bin.tar.gz
RUN tar -xzvf apache-maven-3.9.2-bin.tar.gz -C /opt
```

Important

如果您看到畫面告訴您某些服務需要重新啟動，請按 Enter 鍵選取確定。

如需詳細資訊，請參閱 [Amazon Corretto 11 使用者指南](#)。

2. 安裝 AWS Command Line Interface。請參閱 AWS Command Line Interface 《使用者指南》中的 [安裝或更新最新版 AWS CLI](#) 的程序。

為您的 IP 攝影機 RTSP URLs 建立資源

請依照這些程序，在 中建立所需的串流和秘密 AWS Secrets Manager。請先執行此步驟，因為您需要政策中已建立資源ARNs。

建立 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](#)。在左側導覽中選擇影片串流。

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

在 中建立秘密 AWS Secrets Manager

在 中 AWS Management Console，開啟 [AWS Secrets Manager 主控台](#)。在左側導覽中選擇秘密。

確認已選取適當的區域。

1. 選擇儲存新機密。

a. 步驟 1：選擇秘密類型

- 選取其他機密類型。
- 在鍵/值對區段中，新增鍵/值對。

索引鍵：MediaURI

Note

金鑰必須是 MediaURI。這是區分大小寫的。如果您輸入不正確，應用程式將無法運作。

值：*Your MediaURI*。

Example

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

- b. 步驟 2：設定秘密。為此秘密命名。隨心所欲命名。
 - c. 步驟 3：設定輪換 - 選用。選擇下一步。
 - d. 步驟 4：檢閱。選擇儲存。
2. 如果您的秘密未立即顯示，請選取重新整理按鈕。

選擇秘密的名稱。請記下秘密 ARN。

3. 針對您要串流的每個 MediaURI 重複此程序。

Note

AWS 網路會封鎖一些公有 RTSP 來源。您無法從 Amazon EC2 執行個體內或在連線到 VPN 時執行未受管理的執行個體存取這些項目。

Important

您的相機 RTSP URL 應以 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 “生產者開發套件配額”](#)。

Amazon Kinesis Video Streams Edge Agent 僅支援視訊。

執行 `gst-discoverer-1.0 Your RtspUrl` 以確保您的攝影機可從您的裝置連線。

儲存您建立的所有串流和秘密 ARNs。您需要這些項目才能進行下一個步驟。

建立 IAM 許可政策

請遵循這些程序來建立 IAM 政策。此許可政策允許 AWS 資源的選擇性存取控制（支援操作的子集）。在此情況下，AWS 資源是您希望 Amazon Kinesis Video Streams Edge Agent 串流的目標影片串流。資源也包含 Amazon Kinesis Video Streams Edge Agent 可以擷取的 AWS Secrets Manager 秘密。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，並在 <https://console.aws.amazon.com/iam/> 開啟 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：

JSON

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

```

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

```

Note

arn:aws:kinesisvideo:*:*:stream/streamName2/* 將 arn:aws:kinesisvideo:*:*:stream/streamName1/* 和 取代為視訊串流ARNs，並將 取代arn:aws:secretsmanager:*:*:secret:* 為包含您在 中建立的 MediaURI 秘密ARNs [the section called “為您的 IP 攝影機 RTSP URLs 建立資源”](#)。將 ARNs 用於您希望 Amazon Kinesis Video Streams Edge 代理程式存取的秘密。

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. 在檢閱和建立頁面上，輸入您正在建立之政策的政策名稱和選用的描述。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。

建立 IAM 角色

您在此步驟中建立的角色可由 擔任，AWS IoT 以便從 AWS Security Token Service () 取得臨時登入資料AWS STS。這會在從 Amazon Kinesis Video Streams Edge Agent 執行憑證授權請求時完成。

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

1. 登入 AWS Management Console，並在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。

2. 在 IAM 主控台的導覽窗格中，選擇角色，然後選擇建立角色。
3. 選擇自訂信任政策角色類型，並貼上下列政策：

JSON

```
{
  "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 政策旁的方塊 [the section called “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目的。

Example

範例：KvsEdgeAgentRole

7. (選用) 在 Description (說明) 中，輸入新角色的說明。
8. (選用) 透過將標籤附加為索引鍵/值對，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，然後選擇建立角色。

建立 AWS IoT 角色別名

請依照這些程序，為您在 中建立的 IAM 角色建立 AWS IoT 角色別名 [the section called “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，以指出要擔任哪個 IAM 角色，以便從 AWS Security Token Service () 取得臨時登入資料 AWS STS。如需詳細資訊，請參閱 [如何使用憑證來取得安全字符](#)。

建立 AWS IoT 角色別名

1. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/iot/> 開啟 AWS IoT Core 主控台。
2. 確認已選取適當的區域。
3. 在左側導覽上，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 “建立 IAM 角色”](#)。
7. 選擇建立。在下一頁，您會看到您的角色別名已成功建立的備註。
8. 搜尋並選取新建立的角色別名。請記下角色別名 ARN。在下一個步驟中，您需要 AWS IoT 此政策。

建立 AWS IoT 政策

請依照這些程序來建立將連接到裝置憑證 AWS IoT 的政策。這會授予 AWS IoT 功能的許可，並允許使用憑證擔任角色別名。

透過 AWS IoT Core 政策，您可以控制對 AWS IoT Core 資料平面的存取。AWS IoT Core 資料平面包含您可以用來執行下列動作的操作：

- 連接至 AWS IoT Core 訊息中介裝置
- 傳送和接收 MQTT 訊息
- 取得或更新物件的裝置影子

如需詳細資訊，請參閱 [AWS IoT Core 政策](#)。

使用 AWS IoT 政策編輯器來建立 AWS IoT 政策

1. 登入 AWS Management Console ，並在 <https://console.aws.amazon.com/iot/> 開啟 AWS IoT Core 主控台。
2. 在左側導覽上，選取安全性，然後選擇政策。

3. 選擇建立政策。
4. 輸入政策的名稱。

Example

政策名稱的範例為 KvsEdgeAccessIoTPolicy。

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[標記您的 AWS IoT 資源](#)。

6. 選擇 JSON 標籤。
7. 貼上下列 JSON 政策文件：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "your-role-alias-arn"
    }
  ]
}
```

Note

`your-role-alias-arn` 以您在 中建立的角色別名 ARN 取代 [the section called “建立 AWS IoT 角色別名”](#)。

8. 選擇建立以儲存您的工作。

建立 AWS IoT 物件並取得 AWS IoT Core 登入資料

此時您已建立：

- IAM 許可政策。請參閱 [the section called “建立 IAM 許可政策”](#)。
- 已連接許可政策的 IAM 角色。請參閱 [the section called “建立 IAM 角色”](#)。
- IAM AWS IoT 角色的角色別名。請參閱 [the section called “建立 AWS IoT 角色別名”](#)。
- AWS IoT 政策，目前未連接到任何 AWS 資源。請參閱 [the section called “建立 AWS IoT 政策”](#)。

建立和註冊 AWS IoT 物件並取得 AWS IoT Core 存取登入資料

1. 將裝置註冊為 AWS IoT 物件，並為裝置產生 X.509 憑證。
 - a. 登入 AWS Management Console 並在 <https://console.aws.amazon.com/iot/> 開啟 AWS IoT Core 主控台。
 - b. 選取適當的區域。
 - c. 在左側導覽上，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. 指定物件屬性
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），然後選擇下一步。
 3. 步驟 3. 將政策連接至憑證

搜尋您在 中建立的許可政策 [the section called “建立 AWS IoT 政策”](#)。

選取政策旁的核取方塊，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。
- 公有金鑰檔案
- 私有金鑰檔案
- Amazon 信任服務端點 (RSA 2048 位元金鑰：Amazon Root CA 1)

請記下每個這些檔案的位置，以供後續步驟使用。

g. 選擇完成。在下一頁，您會看到您的物件已成功建立的備註。

h. 如果上面下載的檔案尚未存在，請將檔案傳輸至您的 AWS IoT 物件。

2. 取得您 AWS 帳戶的登入資料提供者端點。

AWS CLI

執行以下命令：

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

AWS Management Console

在 [AWS CloudShell](#) 中，執行下列命令：

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

請記下此資訊以供後續步驟使用。

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. 登入 AWS Management Console 並在 AWS IoT Core <https://console.aws.amazon.com/iot/> 開啟 主控台。
2. 在左側導覽中，選取設定。
3. 找到裝置資料端點。

請記下此資訊以供後續步驟使用。

4. (選用) 確認您的憑證已正確產生。

執行下列命令來驗證您的項目是否正確產生。

```
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[如何使用憑證來取得安全字符](#)。

建置 Amazon Kinesis Video Streams Edge 代理程式

建置 Amazon Kinesis Video Streams Edge 代理程式

1. 使用提供給您的連結下載 tar 檔案。

如果您已完成 Amazon Kinesis Video Streams Edge Agent 興趣表單，請檢查您的電子郵件以取得下載連結。如果您尚未完成表單，請[在此處](#)完成。

2. 驗證檢查總和。
3. 在裝置中解壓縮二進位檔和 jar。

類型：`tar -xvf kvs-edge-agent.tar.gz`。

擷取之後，您的資料夾結構會如下所示：

```
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

發行資料夾名稱的設定方式應該反映最新的二進位發行編號。例如，1.0.0 版本會將資料夾名稱設定為 1.0.0。

4. 建置相依性 jar。

Note

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

導覽至包含的 `kvs-edge-agent` 資料夾 `pom.xml`。

輸入 `mvn clean package`。

這會產生 jar 檔案，其中包含 Amazon Kinesis Video Streams Edge Agent 在所需的相依性kvs-edge-agent/target/libs.jar。

- 將 libs.jar放入包含元件成品的資料夾。

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

- 使用先前步驟的值設定環境變數。下表提供變數的說明。

環境變數名稱	必要	描述
AWS_REGION	是	使用的區域。 範例：us-west-2
AWS_IOT_CA_CERT	是	用於透過 TLS 與後端服務建立信任的 CA 憑證檔案路徑。 範例： <code>/file/path/to/AmazonRootCA1.pem</code>
AWS_IOT_CORE_CERT	是	X.509 憑證的檔案路徑。 範例： <code>/file/path/to/certificateID-certificate.pem.crt</code>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	是	您 AWS 帳戶的 AWS IoT Core 登入資料端點提供者端點 。 範例： <code>credential-account-specific-prefix.credentials.iot.amazonaws.com</code>

環境變數名稱	必要	描述
AWS_IOT_CORE_DATA_ATS_ENDPOINT	是	<p>您 AWS 帳戶 AWS IoT Core 的資料平面端點。</p> <p>範例：<i>data-account-specific-prefix.iot.aws-region.amazonaws.com</i></p>
AWS_IOT_CORE_PRIVATE_KEY	是	<p>公有/私有金鑰對中使用的私有金鑰的檔案路徑。如需詳細資訊，請參閱 中的金鑰管理 AWS IoT。</p> <p>範例：<i>/file/path/to/certificateID-private.pem.key</i></p>
AWS_IOT_CORE_ROLE_ALIAS	是	<p>連接至 時，指向要使用之 AWS IAM 角色的角色別名名稱 AWS IoT Core。</p> <p>範例：<i>kvs-edge-role-alias</i></p>
AWS_IOT_CORE_THING_NAME	是	<p>正在執行應用程式的 AWS IoT 物件名稱。</p> <p>範例：<i>my-edge-device-thing</i></p>

環境變數名稱	必要	描述
GST_PLUGIN_PATH	是	<p>檔案路徑指向包含 <code>gstkvssink</code> 和 <code>IngstorPipelineJNI</code> 平台相依程式庫的資料夾。讓 <code>GStreamer</code> 載入這些外掛程式。如需詳細資訊，請參閱 the section called “下載、建置和設定 GStreamer 元素”。</p> <p>範例：<code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/</code></p>
LD_LIBRARY_PATH	是	<p>檔案路徑指向包含 <code>cproducer</code> 和 <code>KinesisVideoProducer</code> 平台相依程式庫的目錄。</p> <p>範例：<code>/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/</code></p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	否	<p>決定 Amazon Kinesis Video Streams Edge 代理程式是否會發佈任務運作狀態指標 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>範例：FALSE</p>
AWS_KVS_EDGE_LOG_LEVEL	否	<p>記錄 Amazon Kinesis Video Streams Edge Agent 輸出的層級。</p> <p>接受的值：</p> <ul style="list-style-type: none">• OFF• ALL• 嚴重• ERROR• WARN• INFO，預設，若未提供• DEBUG• TRACE <p>範例：INFO</p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	否	<p>一旦日誌檔案達到此大小，就會發生輪換。</p> <ul style="list-style-type: none"> • 最小值：0 • 最大值：10000 • 預設：20，若未提供 • 單位：MB (MB) <p>範例：5</p>
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	否	<p>檔案路徑指向輸出 Amazon Kinesis Video Streams Edge Agent 日誌的目錄。./log 如果未提供，則預設為。</p> <p>範例：<i>/file/path/</i></p>
AWS_KVS_EDGE_LOG_ROTATION_COUNT	否	<p>刪除之前要保留的復原日誌數目。</p> <ul style="list-style-type: none"> • 最小值：1 • 上限：100 • 預設：10，若未提供 <p>範例：20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	否	<p>指向目錄錄製媒體的檔案路徑將寫入。如果未提供，則預設為目前的目錄。</p> <p>範例：<i>/file/path/</i></p>

環境變數名稱	必要	描述
GST_DEBUG	否	指定要輸出的 GStreamer 日誌層級。如需詳細資訊，請參閱 GStreamer 文件 。 範例：0
GST_DEBUG_FILE	否	指定 GStreamer 除錯日誌的輸出檔案。如果取消設定，偵錯日誌會取得標準錯誤的輸出。如需詳細資訊，請參閱 GStreamer 文件 。 範例： <code>/tmp/gstreamer-logging.log</code>

7. 清除 GStreamer 快取。類型：

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

如需詳細資訊，請參閱 [GStreamer 登錄檔文件](#)。

8. 準備並執行 java 命令。Amazon Kinesis Video Streams Edge 代理程式接受下列引數：

Java 屬性名稱	必要	描述
java.library.path	否	檔案路徑指向包含 gstkvssink 和 IngestorPipelineJNI 相依程式庫的資料夾。如果未提供，Amazon Kinesis Video Streams Edge 代理程式將在目前目錄中搜尋它們。

Important

如果找不到這些檔案，Amazon Kinesis

Java 屬性名稱

必要

描述

Video Streams Edge Agent 將無法正常運作。

範例：*/file/path/*

若要設定這些命令，請將 `-Djava-property-name=value` 新增至用來執行 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

從與相同的目錄執行上述 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. 建立新的檔案 *example-edge-configuration.json*。

將以下程式碼貼到檔案。這是每日上午 9:00:00 至下午 4:59:59 記錄的範例組態（根據您 AWS IoT 裝置上的系統時間）。它也會每天從下午 7:00:00 到晚上 9:59:59 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 Edge 代理程式，請在 中輸入下列項目
AWS CLI：

```

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

```

10. 針對 Amazon Kinesis Video Streams Edge Agent 的每個串流重複上述步驟。

在裝置上安裝 CloudWatch 代理程式

Note

請注意 [CloudWatch 配額](#)。

請依照這些程序安裝和設定 CloudWatch 代理程式，以自動將 Amazon Kinesis Video Streams Edge 代理程式產生的日誌上傳至 CloudWatch。此為選用步驟。

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

出現組態提示時，請選取下列其中一個組態。

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 位置執行 Amazon Kinesis Video Streams Edge

Agent : `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version`。

- 若要設定 CloudWatch 代理程式上傳日誌並發佈裝置 RAM 和 CPU 指標，請將以下內容貼到組態檔案中。

```
{
  "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"
        }
      ]
    }
  }
}

```

```
}
}
```

- 若要僅上傳日誌，而不收集裝置的 RAM 和 CPU，請使用下列組態：

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

執行 Amazon Kinesis Video Streams Edge 代理程式作為原生程序

將 Amazon Kinesis Video Streams Edge Agent 設定為系統化服務。此為選用步驟。

systemd 是 Linux 裝置上的系統和服務管理員。systemd 是管理程序的建議方式，因為它會在應用程式發生錯誤或執行應用程式的裝置斷電時重新啟動 Amazon Kinesis Video Streams Edge Agent。

請執行下列操作：

執行 Amazon Kinesis Video Streams Edge 代理程式作為原生程序

1. 在 `/etc/systemd/system` 中建立新的檔案，並將其命名為 `aws.kinesisvideo.edge-runtime-agent.service`。

貼上下列項目：

```
[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 服務組態檔案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 “建置 Edge 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 Edge Agent 服務的狀態，以確認是否正在執行。

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. `systemctl --help` 輸入 以取得使用 管理程序的完整選項清單 `systemctl`。

以下是管理 Amazon Kinesis Video Streams Edge Agent 的一些常見命令：

- 若要重新啟動，請輸入 `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`。
- 若要停止，請輸入 `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`。

- 若要在每次裝置重新啟動時自動啟動，請輸入 `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`。

將 Amazon Kinesis Video Streams Edge 代理程式部署至 AWS IoT Greengrass

本節提供使用 Amazon Kinesis Video Streams 的完整指南 AWS IoT Greengrass。透過結合這些服務，您可以有效率地將影片從邊緣裝置串流到雲端，在 IoT、監控等領域啟用各種應用程式。

您可以找到下列詳細資訊：

- [設定開發環境](#)
- [建立 Kinesis 影片串流](#)
- [建立和封裝 Lambda 函數](#)
- [設定 Kinesis Video Streams 核心裝置](#)
- [部署至核心裝置](#)
- [驗證您的串流](#)

請依照下列步驟，將 Amazon Kinesis Video Streams Edge 代理程式部署至 AWS IoT Greengrass，以從 IP 攝影機記錄和上傳媒體。

主題

- [建立 Ubuntu Amazon EC2 執行個體](#)
- [在裝置上設定 AWS IoT Greengrass V2 核心裝置](#)
- [為您的 IP 攝影機 RTSP URLs 建立 Amazon Kinesis Video Streams AWS Secrets Manager 和資源](#)
- [將許可新增至字符交換服務 \(TES\) 角色](#)
- [在裝置上安裝 AWS IoT Greengrass Secret Manager 元件](#)
- [在裝置上部署 Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass 元件](#)
- [在裝置上安裝 AWS IoT Greengrass 日誌管理員元件](#)

建立 Ubuntu Amazon EC2 執行個體

執行下列動作來建立 Ubuntu Amazon EC2 執行個體。

建立 Ubuntu Amazon EC2 執行個體

1. 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 – 輸入執行個體的名稱。
- 應用程式和作業系統映像 (Amazon Machine Image) – 選取 Ubuntu。
- 執行個體類型 – 選取 t2.large。
- 金鑰對登入 – 建立您自己的金鑰對。
- 網路設定 – 保留預設值。
- 設定儲存 – 將磁碟區增加到 256 GiB。
- 進階設定 – 保留預設值。

3. 在其中啟動執行個體和 SSH。

請執行下列操作：

1. 選取左側導覽中的執行個體，然後選取執行個體 ID。
 2. 選擇右上角的連線。
 3. 選擇 SSH 用戶端並遵循畫面上的指示。
 4. 開啟終端機並導覽至下載.pem的檔案（可能在 中~/Downloads）。
 5. 第一次遵循這些程序時，您會收到「無法建立主機的真實性 (...)」訊息。輸入 yes。
4. 安裝系統程式庫，以在執行個體上建置 Amazon Kinesis Video Streams Edge Agent。

```
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

如果您看到畫面告訴您某些服務需要重新啟動，請按 Enter 鍵選取確定。

如需詳細資訊，請參閱 [Amazon Corretto 11 使用者指南](#)。

在裝置上設定 AWS IoT Greengrass V2 核心裝置

請依照這些程序，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. 登入 AWS Management Console，<https://console.aws.amazon.com/iot/>。

確認已選取適當的區域。

2. 在左側導覽中，選取 Greengrass 裝置、Core 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 Greengrass 核心裝置。輸入裝置的名稱。
 - 步驟 2：新增至物件群組以套用連續部署。選取無群組。
 - 步驟 3：安裝 Greengrass Core 軟體。選取 Linux。
 - 步驟 3.1：在裝置上安裝 Java

Java 已安裝為的一部分 [the section called “建立 Ubuntu 執行個體”](#)。如果您尚未安裝 Java，請返回該步驟。

- 步驟 3.2：將 AWS 登入資料複製到裝置

開啟 bash/zsh 選項，並將匯出命令貼到 Amazon EC2 執行個體中。

- 步驟 3.3：執行安裝程式

1. 複製並執行下載安裝程式，並在 Ubuntu Amazon EC2 執行個體中執行安裝程式命令。

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. 記下已建立的字符交換服務 (TES) 角色。供稍後使用。

Note

根據預設，建立的角色稱為 GreengrassV2TokenExchangeRole。

為您的 IP 攝影機 RTSP URLs 建立 Amazon Kinesis Video Streams AWS Secrets Manager 和資源

請依照這些程序，在 中建立所需的串流和秘密 AWS Secrets Manager。請先執行此步驟，因為您需要政策中已建立資源ARNs。

建立 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](#)。在左側導覽中選擇影片串流。

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

在 中建立秘密 AWS Secrets Manager

在 中 AWS Management Console，開啟 [AWS Secrets Manager 主控台](#)。在左側導覽中選擇秘密。

確認已選取適當的區域。

1. 選擇儲存新機密。

- a. 步驟 1：選擇秘密類型

- 選取其他機密類型。
- 在鍵/值對區段中，新增鍵/值對。

索引鍵：MediaURI

Note

金鑰必須為 MediaURI。這區分大小寫。如果您輸入不正確，應用程式將無法運作。

值：*Your MediaURI*。

Example

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

- b. 步驟 2：設定秘密。為此秘密命名。隨心所欲命名。
 - c. 步驟 3：設定輪換 - 選用。選擇下一步。
 - d. 步驟 4：檢閱。選擇儲存。
2. 如果您的秘密未立即顯示，請選取重新整理按鈕。

選擇秘密的名稱。請記下秘密 ARN。

3. 針對您要串流的每個 MediaURI 重複此程序。

Note

AWS 網路會封鎖一些公有 RTSP 來源。您無法從 Amazon EC2 執行個體內或在連線到 VPN 時執行未受管理的執行個體存取這些項目。

Important

您的相機 RTSP URL 應以 h.264 格式串流影片。片段持續時間不得超過 [中所述的限制](#)the section called “生產者開發套件配額”。

Amazon Kinesis Video Streams Edge Agent 僅支援視訊。

執行 `gst-discoverer-1.0 Your RtspUrl` 以確保您的攝影機可從您的裝置連線。

儲存您建立的所有串流和秘密 ARNs。您需要這些項目才能進行下一個步驟。

將許可新增至字符交換服務 (TES) 角色

將字符交換服務 (TES) 角色授予取得查看秘密許可的裝置。這是元件正常運作的必要條件 AWS Secrets Manager AWS IoT Greengrass 。

將許可新增至 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，並搜尋您在程序中稍早建立的 TES 角色。
3. 在新增許可下拉式清單中，選取連接政策。
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 並編輯政策。

將政策取代為下列項目：

Note

arn:aws:kinesisvideo:*:*:stream/streamName2/* 將
arn:aws:kinesisvideo:*:*:stream/streamName1/* 和 取代為您在上一個步驟中
建立的串流 ARNs。

JSON

```
{
  "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. 為您的政策命名，然後選擇建立政策。

政策名稱的範例為 KvsEdgeAccessPolicy。

9. 關閉標籤並返回您要將政策連接到 TES 角色的標籤。

選擇重新整理按鈕，然後搜尋新建立的政策。

選取核取方塊，然後選擇連接政策。

在下一個畫面上，您會看到備註，指出政策已成功連接到角色。

10. 建立並連接另一個政策，這次為您的秘密。

將政策取代為下列項目：

Note

arn:aws:secretsmanager:*:*:secret:* 以包含您在 中建立之 MediaURI 秘密 ARNs 取代 [the section called “為您的 IP 攝影機 RTSP URLs 建立資源”](#)。

JSON

```

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

```

```

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

```

11. 建立並連接另一個政策，這次用於 Amazon CloudWatch 指標。將政策取代為下列項目：

JSON

```

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

```

在裝置上安裝 AWS IoT Greengrass Secret Manager 元件

Amazon Kinesis Video Streams Edge Agent 需要先在裝置上安裝 AWS IoT Greengrass Secret Manager 元件。

安裝 Secret Manager 元件

1. 登入 AWS Management Console，並在 <https://console.aws.amazon.com/iot/> 開啟 AWS IoT Core 主控台。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 “設定 AWS IoT Greengrass 核心裝置”](#)。

3. 在右上角的動作下拉式清單中，選擇修訂。

在出現的快顯視窗中，選擇修訂部署。

4. 完成下列各節：

- 步驟 1：指定目標。選擇下一步。
- 步驟 2：選取元件。
 - 確認已選取 `aws.greengrass.Cli` 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`。
 - 勾選 `aws.greengrass.SecretManager` 旁的方塊，然後選擇下一步。
- 步驟 3：設定元件。設定 AWS IoT Greengrass Secret Manager 元件，從 AWS IoT Greengrass 環境中下載秘密。

選取 `aws.greengrass.SecretManager` 元件，然後選擇設定元件。

在出現的畫面中，更新組態中的 AWS Secrets Manager ARNs 以合併方塊。

Note

`arn:aws:secretsmanager:*:*:secret:*` 將取代為您在 中建立之秘密的 ARNs [the section called “為您的 IP 攝影機 RTSP URLs 建立資源”](#)。

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

 Note

cloudSecrets 是具有金鑰的物件清單arn。如需詳細資訊，請參閱《AWS IoT Greengrass Version 2 開發人員指南》中的 [Secret Manager 組態](#) 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 元件和許可已正確安裝。

在 Ubuntu Amazon EC2 執行個體上，輸入 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` 來驗證元件是否收到更新後的組態。

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，請修改部署以移除 `aws.greengrass.SecretManager` 元件。

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

如果部署錯誤與缺少許可有關，請檢閱 [the section called “將許可新增至 TES 角色”](#) 區段，以確保 TES 角色具有適當的許可。然後，重複本節。

更新 AWS IoT Greengrass Secret Manager 元件上的秘密

 Important

AWS IoT Greengrass Secret Manager 元件只會在更新部署時擷取和快取秘密。

若要更新 AWS IoT Greengrass Secret Manager 元件上的秘密，請遵循上述步驟 1–6，並執行下列變更。

步驟 3：設定元件。設定 AWS IoT Greengrass Secret Manager 元件，從 AWS IoT Greengrass 環境中下載秘密。

選取 `aws.greengrass.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，並更新要合併組態方塊中 AWS Secrets Manager ARNs。

如需詳細資訊，請參閱[重設更新](#)。

在裝置上部署 Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass 元件

執行下列動作以在裝置上部署 Amazon Kinesis Video Streams Edge Agent AWS IoT Greengrass 元件：

部署 元件

1. 使用提供的連結下載 tar 檔案。

如果您已完成 Amazon Kinesis Video Streams Edge Agent 興趣表單，請檢查您的電子郵件以取得下載連結。如果您尚未完成表單，請[在此處](#)完成。

2. 驗證檢查總和。
3. 在裝置中解壓縮二進位檔和 jar。

類型：`tar -xvf kvs-edge-agent.tar.gz`。

擷取之後，您的資料夾結構會如下所示：

```
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

發行資料夾名稱的設定方式應該反映最新的二進位發行編號。例如，1.0.0 版本會將資料夾名稱設定為 1.0.0。

4. 建置相依性 jar。

Note

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

導覽至包含的 kvs-edge-agent 資料夾 pom.xml。

輸入 mvn clean package。

這將產生 jar 檔案，其中包含 Amazon Kinesis Video Streams Edge Agent 在所需的相依性 kvs-edge-agent/target/libs.jar。

5. 將 libs.jar 放入包含元件成品的資料夾。

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

6. 「選用」。設定屬性。Amazon Kinesis Video Streams Edge Agent 在模式中 AWS IoT Greengrass 接受下列環境變數：

環境變數名稱	必要	描述
AWS_REGION	是	<p>使用的 區域。</p> <p>範例：us-west-2</p> <p>AWS IoT Greengrass 核心軟體會自動為您設定此值。如需詳細資訊，請參閱《AWS IoT Greengrass Version 2 開發人員指南》中的元件環境變數參考主題。</p>
GST_PLUGIN_PATH	是	<p>檔案路徑指向包含 gstkvssink 和IngestorPipelineJNI 平台相依程式庫的資料夾。這可讓 GStreamer 載入這些外掛程式。如需詳細資訊，請參閱the section called “下載、建置和設定 GStreamer 元素”。</p> <p>範例：<i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ <i>EdgeAgentVersion</i> /</i></p>

環境變數名稱	必要	描述
LD_LIBRARY_PATH	是	<p>檔案路徑指向包含 <code>cproducer</code> 和 <code>KinesisVideoProducer</code> 平台相依程式庫的目錄。</p> <p>範例：<code>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion /lib/</code></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	否	<p>決定 Amazon Kinesis Video Streams Edge 代理程式是否會發佈任務運作狀態指標 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>範例：FALSE</p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_LEVEL	否	<p>記錄 Amazon Kinesis Video Streams Edge Agent 輸出的層級。</p> <p>接受的值：</p> <ul style="list-style-type: none"> • OFF • ALL • 嚴重 • ERROR • WARN • INFO, 預設, 若未提供 • DEBUG • TRACE <p>範例：INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	否	<p>一旦日誌檔案達到此大小，就會發生輪換。</p> <ul style="list-style-type: none"> • 最小值：1 • 上限：100 • 預設：20, 若未提供 • 單位：MB (MB) <p>範例：5</p>
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	否	<p>檔案路徑指向輸出 Amazon Kinesis Video Streams Edge Agent 日誌的目錄。./log 如果未提供，則預設為。</p> <p>範例：<i>/file/path/</i></p>

環境變數名稱	必要	描述
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	否	<p>刪除之前要保留的復原日誌數目。</p> <ul style="list-style-type: none"> • 最小值：1 • 上限：100 • 預設：10，若未提供 <p>範例：20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	否	<p>指向目錄錄製媒體的檔案路徑將寫入。如果未提供，則預設為目前的目錄。</p> <p>範例：<i>/file/path/</i></p>
GREENGRASS_ROOT_DIRECTORY	否	<p>AWS IoT Greengrass 根目錄的檔案路徑。</p> <p><i>/greengrass/v2/</i> 如果未提供，則預設為。</p> <p>範例：<i>/file/path/</i></p>
GST_DEBUG	否	<p>指定要輸出的 GStreamer 日誌層級。如需詳細資訊，請參閱 GStreamer 文件。</p> <p>範例：0</p>
GST_DEBUG_FILE	否	<p>指定 GStreamer 除錯日誌的輸出檔案。如果取消設定，偵錯日誌會取得標準錯誤的輸出。如需詳細資訊，請參閱 GStreamer 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，以新增任何上述環境變數。

⚠ Important

請確定修改後的執行指令碼不包含任何標籤字元。AWS IoT Greengrass 核心軟體將無法讀取配方。

7. 部署 Amazon Kinesis Video Streams Edge Agent 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"
```

如需詳細資訊，請參閱《AWS IoT Greengrass Version 2 開發人員指南》中的下列章節：

- [AWS IoT Greengrass CLI 命令](#)
- [將 AWS IoT Greengrass 元件部署至裝置](#)

8. 使用 將組態傳送至應用程式 AWS CLI。

a. 建立新的檔案 `example-edge-configuration.json`。

將以下程式碼貼到檔案。這是每日上午 9:00:00 至下午 4:59:59 記錄的範例組態（根據您 AWS IoT 裝置上的系統時間）。它也會每天從下午 7:00:00 到晚上 9:59:59 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 `awscli` 中輸入以下內容 AWS CLI，將檔案傳送至 Amazon Kinesis Video Streams Edge Agent：

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

9. 針對 Amazon Kinesis Video Streams Edge Agent 的每個串流重複上述步驟。

在裝置上安裝 AWS IoT Greengrass 日誌管理員元件

Note

請注意 [CloudWatch 配額](#)。

請依照這些程序來設定 Amazon Kinesis Video Streams Edge Agent 日誌，以使用 AWS IoT Greengrass 日誌管理員元件自動上傳至 CloudWatch。此為選用步驟。

安裝 AWS IoT Greengrass 日誌管理員元件

1. 確認 AWS IoT Greengrass 裝置角色具有適當的許可。
 - a. 登入 AWS Management Console ，並在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
 - b. 按一下左側導覽中的角色。
 - c. 選擇在 中建立的 TES 角色名稱 [the section called “設定 AWS IoT Greengrass 核心裝置”](#)。如有必要，請使用搜尋列。
 - d. 選取 GreengrassV2TokenExchangeRoleAccess 政策。
 - e. 選取 JSON 索引標籤，並確認政策看起來如下：

JSON

```
{
  "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，或缺少一些必要的許可，請使用這些許可建立新的 IAM 政策，並將其連接到在 中建立的 TES 角色 [the section called “設定 AWS IoT Greengrass 核心裝置”](#)。
2. 登入 AWS Management Console 並開啟位於 <https://console.aws.amazon.com/iot/> 的 AWS IoT Core 主控台。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 中建立的物件具有相同目標的部署 [the section called “設定 AWS IoT Greengrass 核心裝置”](#)。

4. 在右上角，選取動作，然後選擇修訂。

在出現的快顯視窗中，選擇修訂部署。

5. 完成下列各節：

- a. 步驟 1：指定目標。選擇下一步。

- b. 步驟 2：選取元件。

- i. 確認仍然選取 `aws.greengrass.Cli` 元件和 `aws.greengrass.SecretManager` 元件。

⚠ 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 `aws.greengrass.LogManager` 旁的方塊，然後選擇下一步。

- c. 步驟 3：設定元件。設定 AWS IoT Greengrass 日誌管理員元件以上傳 Amazon Kinesis Video Streams Edge Agent 產生的日誌。

選取 `aws.greengrass.LogManager` 元件，然後選擇設定元件。

在出現的畫面中，將下列日誌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_kvssink.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

如需日誌管理員組態之每個參數的詳細資訊，請參閱《AWS IoT Greengrass Version 2 開發人員指南》中的 [日誌管理員](#) 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 日誌管理員元件和許可已正確安裝。

7. 在 Ubuntu Amazon EC2 執行個體上，輸入 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` 來驗證元件是否收到更新後的組態。
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，請修改部署以移除 `aws.greengrass.LogManager` 元件。

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

如果部署錯誤與缺少許可有關，請檢閱 [the section called “將許可新增至 TES 角色”](#) 以確保 TES 角色具有適當的許可。然後，重複本節。

Amazon Kinesis Video Streams Edge 代理程式常見問答集

以下是 Amazon Kinesis Video Streams Edge Agent 服務的一些常見問題。

Amazon Kinesis Video Streams Edge Agent 支援哪些作業系統？

Amazon Kinesis Video Streams Edge Agent 目前支援下列作業系統：

Ubuntu

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

AL2

- amzn2
 - AMD64 amazonlinux : 2.0.20210219.0-amd64 (Snowball)

Amazon Kinesis Video Streams Edge Agent 是否支援 H.265 媒體？

Amazon Kinesis Video Streams Edge Agent 僅支援 H.264 基本串流。

Amazon Kinesis Video Streams Edge Agent 是否可在 AL2 中運作？

是。

如何在 AWS IoT 實物或裝置中執行多個串流？

將另一個 [the section called "StartEdgeConfigurationUpdate"](#) 傳送到相同的 HubDeviceArn，但不同的 Amazon Kinesis Video Streams/AWS Secrets Manager ARNs。

傳送 **StartEdgeConfigurationUpdate** 後如何編輯？

HubDeviceArn 使用相同的 Amazon Kinesis Video Streams ARN，[the section called "StartEdgeConfigurationUpdate"](#) 將更新後的 傳送至相同的。當應用程式收到來自 Amazon Kinesis Video Streams 的訊息時，它會覆寫該串流的先前組態。屆時將發生變更。

您是否有任何常見的範例 **ScheduleConfigs**？

Amazon Kinesis Video Streams Edge Agent 會使用其執行所在裝置的系統時間。

描述	ScheduleExpression	DurationInSeconds
全年無休錄製，每小時上傳	(null ScheduleConfig)	
每天上午 9 : 00 : 00 - 下午 4 : 59 : 59	0 0 9-16 * * # *	3599
9:00:00 AM - 4:59:59 PM weekdays	0 0 9-16 # * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599

描述	ScheduleExpression	DurationInSeconds
週末上午 9 : 00 : 00 - 下午 4 : 59 : 59	0 0 9-16 ? * SAT,SUN *	3599
工作日下午 10 : 00 : 00 - 晚上 11 : 59 : 59	0 0 22,23 ? * MON-FRI *	3599
每天上午 9 : 00 : 00 - 上午 10 : 00 : 00	0 0 9 * * ? *	3600
每天下午 4 : 00 : 00 - 下午 5 : 59 : 59	0 0 16-17 * * ? *	3599

如需更多範例，請參閱 [Jetson 文件](#)。

是否有最大串流限制？

Amazon Kinesis Video Streams Edge Agent 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 更新相同串流的組態 [the section called “StartEdgeConfigurationUpdate”](#) 不會增加裝置的串流計數。

如何重新啟動發生錯誤的任務？

如果發生錯誤，Amazon Kinesis Video Streams Edge 代理程式會嘗試重新啟動任務。不過，若發生某些錯誤（例如組態錯誤），您必須手動重新啟動任務。

若要判斷哪些任務需要手動重新啟動，請參閱 中的 FatalError 指標 [the section called “使用 CloudWatch 監控 Amazon Kinesis Video Streams Edge 代理程式”](#)。

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

如何監控 Amazon Kinesis Video Streams Edge Agent 的運作狀態？

如需詳細資訊，請參閱 [the section called “使用 CloudWatch 監控 Amazon Kinesis Video Streams Edge 代理程式”](#)。

透過 VPC 串流影片

此 Beta 版可在歐洲（巴黎）區域 eu-west-3 進行預覽。若要存取這些元件和我們的入門指南，請[傳送電子郵件給我們](#)。

Amazon Kinesis Video Streams VPC 端點服務可讓您透過 Amazon 網路串流和使用影片，而不需透過公有網際網路進行任何資料。

若要請求存取，請[透過電子郵件將下列資訊傳送給我們](#)：

- 帳戶 ID
- 串流 ARNs
- VPC ID

Note

我們最多可能需要一週的時間才能將您新增至服務。

如果您過去未曾使用過 VPC 端點，請檢閱下列資訊以熟悉概念：

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

其他資訊

您新增至 Beta 版後，我們會透過電子郵件將此功能的其他資訊連結傳送給您。

VPC 端點程序

配額

主要配額差異為：

- 所有頻寬 APIs 的配額較低 (2 mbps)：

- PutMedia
- GetMedia
- GetMediaForFragmentList
- 每個客戶允許 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 Kinesis Video Streams [VPC 主控台或 \(\)](#) 為 Amazon Kinesis Video Streams 建立介面 [VPC 端點](#) AWS CLI。AWS Command Line Interface

在 中 AWS CLI，輸入下列項目：

```
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 來透過端點路由。如果您未啟用此功能，則需要實作自己的 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](#)。

控制對端點的存取

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

- 可以執行動作的委託人、
- 可執行的動作，以及
- 可對其執行動作的資源。

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的 [使用端點政策控制對 VPC 端點的服務存取](#)。

以下是 Amazon Kinesis Video Streams 的端點政策範例。連接至端點時，此政策會拒絕所有資源上所有主體的所列PutMedia動作的存取權。

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

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 生產者 SDK GStreamer 外掛程式 - kvssink](#)：示範如何建置 Kinesis Video Streams 生產者 SDK 以用作 GStreamer 目的地。
- [在 Docker 容器中執行 GStreamer 元素](#)：示範如何使用預先建置的 Docker 映像，將即時串流通訊協定 (RTSP) 影片從 IP 攝影機傳送至 Kinesis Video Streams。
- [範例：從 RTSP 來源串流](#)：示範如何建置自己的 Docker 影像，並將 RTSP 影片從 IP 攝影機傳送至 Kinesis Video Streams。
- [範例：使用 PutMedia API 將資料傳送至 Kinesis Video Streams](#)：示範如何使用 [PutMedia API](#) [使用 Java 生產者程式庫](#)將資料傳送至已採用容器格式 (MKV) 的 Kinesis Video Streams。

範例：從 Kinesis Video Streams 擷取資料

- [KinesisVideoExample](#)：示範如何使用 Kinesis Video Streams Parser Library 剖析和記錄影片片段。
- [範例：剖析和轉譯 Kinesis Video Streams 片段](#)：示範如何使用 [JCodec](#) 和 [JFrame](#) 剖析和轉譯 Kinesis 影片串流片段。

範例：播放影片資料

- [範例：在 HTML 和 JavaScript 中使用 HLS](#)：說明如何擷取 Kinesis 影片串流的 HLS 串流工作階段，並在網頁中播放。

先決條件

- 在範例程式碼中，您可以指定您在登入資料設定檔檔案中設定的 AWS 設定檔，或在整合式開發環境 (IDE) 的 Java 系統屬性中提供登入資料。如果您尚未這麼做，請先設定您的登入資料。如需詳細資訊，請參閱[設定用於開發的 AWS 登入資料和區域](#)。
- 我們建議您使用 Java IDE 檢視及執行程式碼，例如：
 - [Eclipse Java Neon](#)

- [JetBrains IntelliJ IDEA](#)

範例：Kinesis Video Streams 生產者 SDK GStreamer 外掛程式 - kvssink

本主題說明如何建置 Amazon Kinesis Video Streams 生產者 SDK 以用作 GStreamer 外掛程式。

主題

- [下載、建置和設定 GStreamer 元素](#)
- [執行 GStreamer 元素](#)
- [GStreamer 啟動命令範例](#)
- [在 Docker 容器中執行 GStreamer 元素](#)
- [GStreamer 元素參數參考](#)

[GStreamer](#) 是多個攝影機和視訊來源使用的熱門媒體架構，可透過結合模組化外掛程式來建立自訂媒體管道。Kinesis Video Streams GStreamer 外掛程式可簡化現有 GStreamer 媒體管道與 Kinesis Video Streams 的整合。整合 GStreamer 之後，您可以將視訊從網路攝影機或即時串流通訊協定 (RTSP) 攝影機串流到 Kinesis Video Streams，以進行即時或更新版本的播放、儲存和進一步分析。

GStreamer 外掛程式會自動管理影片串流到 Kinesis Video Streams 的傳輸，方法是在 GStreamer 接收器元素中封裝 Kinesis Video Streams 生產者 SDK 所提供的功能 kvssink。GStreamer 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 的外掛程式元素接著會將影片串流傳送至 Kinesis Video Streams。

本主題說明如何建構能夠從影片來源串流影片的 GStreamer 媒體管道，例如 Web 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。當您的影片串流可作為 Kinesis 影片串流使用時，您可以使用 [the section called “使用剖析器程式庫進行串流”](#) 進一步處理、播放、儲存或分析影片串流。

下載、建置和設定 GStreamer 元素

GStreamer 外掛程式範例包含在 Kinesis Video Streams C++ 生產者 SDK 中。如需開發套件先決條件和下載的相關資訊，請參閱 [下載並設定 C++ 生產者程式庫程式碼](#)。

您可以在 macOS、Ubuntu、Raspberry Pi 或 Windows 上將生產者 SDK GStreamer 接收器建置為動態程式庫。GStreamer 外掛程式位於您的 build 目錄中。若要載入此外掛程式，它必須位於您的中 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。

執行 GStreamer 元素

若要使用 Kinesis Video Streams 生產者 SDK 元素作為接收器執行 GStreamer，請使用 `gst-launch-1.0` 命令。使用適合 GStreamer 外掛程式使用的上游元素。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。

除了 [提供登入](#) 資料和 [提供區域](#) 之外，`kvssink` 元素還具有下列必要參數：

- `stream-name` – 目的地 Kinesis Video Streams 的名稱。

如需選用參數 `kvssink` 的詳細資訊，請參閱 [GStreamer 元素參數參考](#)。

如需 GStreamer 外掛程式和參數的最新資訊，請參閱 [GStreamer 外掛程式](#)。您也可以使用 GStreamer 元素或外掛程式的名稱 `gst-inspect-1.0` 後面的 來列印其資訊，並驗證裝置是否可用：

```
gst-inspect-1.0 kvssink
```

如果建置 `kvssink` 失敗或 `GST_PLUGIN_PATH` 未正確設定，您的輸出會如下所示：

```
No such element or plugin 'kvssink'
```

GStreamer 啟動命令範例

下列範例示範如何使用 kvssink GStreamer 外掛程式從不同類型的裝置串流影片。

範例 1：從 Ubuntu 上的 RTSP 攝影機串流影片

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
```

範例 2：從 Ubuntu 上的 USB 攝影機編碼和串流視訊

下列命令會在 Ubuntu 上建立 GStreamer 管道，以 H.264 格式從 USB 攝影機編碼串流，並將其串流至 Kinesis Video Streams。此範例使用 [v4l2src](#) GStreamer 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影片

下列命令會在 Ubuntu 上建立 GStreamer 管道，將相機已編碼為 H.264 格式的影片串流至 Kinesis Video Streams。此範例使用 [v4l2src](#) GStreamer 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"
```

範例 4：從 macOS 上的網路攝影機串流影片

下列命令會在 macOS 上建立 GStreamer 管道，將影片從網路攝影機串流至 Kinesis Video Streams。此範例使用 [rtspsrc](#) GStreamer 外掛程式。

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
  rtph264depay ! h264parse ! video/x-h264, format=avc,alignment=au ! kvssink
  stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-
key="YourSecretKey" aws-region="YourAWSRegion"
```

範例 5：從 Windows 上的網路攝影機串流視訊

下列命令會在 Windows 上建立 GStreamer 管道，將影片從網路攝影機串流至 Kinesis Video Streams。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"
```

範例 6：從 Raspberry Pi 上的攝影機串流影片

下列命令會在 Raspberry Pi 上建立 GStreamer 管道，將影片串流至 Kinesis Video Streams。此範例使用 [v4l2src](#) GStreamer 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"
```

範例 7：在 Raspberry Pi 和 Ubuntu 中串流音訊和視訊

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。

範例 8：從 macOS 中的裝置來源串流音訊和視訊

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 9：上傳同時包含音訊和視訊的 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](#)。您需要具有 h.264 和 AAC 編碼媒體的 MKV 測試檔案。

在 Docker 容器中執行 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 管道會將 Kinesis Video Streams 的操作環境標準化，以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，您可以使用下列其中一個提供的 `docker pull` 命令，從 Amazon Elastic Container Registry 下載 Kinesis Video Streams C++ Producer SDK（和 GStreamer 外掛程式）。

若要使用 Kinesis Video Streams 生產者 SDK 元素作為 Docker 容器中的接收器來執行 GStreamer，請執行下列動作：

主題

- [驗證您的 Docker 用戶端](#)
- [下載 Ubuntu、macOS、Windows 或 Raspberry Pi 的 Docker 影像](#)
- [執行 Docker 映像](#)

驗證您的 Docker 用戶端

向打算提取映像的 Amazon ECR 登錄檔驗證您的 Docker 用戶端。您必須取得每個使用的登錄檔的身分驗證字符。字符有效期為 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

若要使用 Amazon ECR 進行身分驗證，請複製並貼上下列命令，如下所示。

```
sudo aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

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

```
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
```

執行 Docker 映像

視您的作業系統而定，使用下列其中一個命令執行 Docker 映像：

在 Ubuntu 上執行 Docker 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 上執行 Docker 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 Windows 上執行 Docker 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 Raspberry Pi 上執行 Docker 映像

```
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，讓您使用容器內的命令。

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
```

kvssink 使用 開始串流至 `gst-launch-1.0`，以執行適合您裝置和視訊來源的管道。如需管道範例，請參閱 [GStreamer 啟動命令範例](#)。

GStreamer 元素參數參考

若要將影片傳送至 Amazon Kinesis Video Streams 生產者 C++ SDK，您可以將指定 kvssink 為管道的目的地或最終目的地。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 外掛程式 - kvssink”](#)。

主題

- [the section called “提供登入資料給 kvssink”](#)
- [the section called “提供區域給 kvssink”](#)
- [the section called “kvssink 選用參數”](#)

提供登入資料給 kvssink

若要允許 kvssink GStreamer 元素向 發出請求 AWS，請在呼叫 Amazon Kinesis Video Streams 服務時提供 AWS 登入資料以供其使用。登入資料提供者鏈結會依下列順序尋找登入資料：

1. AWS IoT 憑證

若要設定 AWS IoT 登入資料，請參閱 [the section called “使用 控制對 Kinesis Video Streams 資源的存取 AWS IoT”](#)。

`iot-credentials` 參數值必須以開頭 `iot-certificate`，後面接著下列 *key=value* 對的逗號分隔清單。

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的 CA 憑證檔案路徑。 Example 範例： <code>/file/path/to/certificate.pem</code>
<code>cert-path</code>	是	X.509 憑證的檔案路徑。 Example 範例： <code>/file/path/to/certificateID -certificate.pem.crt</code>
<code>endpoint</code>	是	您 AWS 帳戶的 AWS IoT Core 登入資料端點提供者端點。請參閱 AWS IoT 開發人員指南 。 Example 範例： <code>credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com</code>

金鑰	必要	描述
key-path	是	公有/私有金鑰對中使用的私有金鑰的檔案路徑。 Example 範例： <code>/file/path/to/certificateID-private.pem.key</code>
role-aliases	是	連接至時，指向要使用之AWS IAM 角色的角色別名名稱AWS IoT Core。 Example 範例： <code>KvsCameraIoTRoleAlias</code>
iot-thing-name	否	iot-thing-name 是選用的。如果iot-thing-name未提供，則會使用stream-name 參數值。 Example 範例： <code>kvs_example_camera</code>

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

若要讓 kvssink 使用環境中的登入資料，請設定下列環境變數：

環境變數名稱	必要	描述
AWS_ACCESS_KEY_ID	是	用來存取 Amazon Kinesis Video Streams 的 AWS 存取金鑰。
AWS_SECRET_ACCESS_KEY	是	與存取金鑰相關聯的 AWS 私密金鑰。
AWS_SESSION_TOKEN	否	如果您直接從 AWS STS 操作使用臨時安全登入資料，請指定必要的工作階段字符值。

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。若要讓變數在未來的工作階段中持續存在，請在 shell 的啟動指令碼中設定它們。

3. access-key、secret-key 參數

若要直接指定登入資料做為 kvssink 參數，請設定下列參數：

kvssink 參數名稱	必要	描述
access-key	是	用來存取 Amazon Kinesis Video Streams 的 AWS 存取金鑰。
secret-key	是	與存取金鑰相關聯的 AWS 私密金鑰。
session-token	否	如果您直接從 AWS STS 操作使用臨時安全登入資料，請指定必要的工作階段字符值。

Example

使用靜態登入資料：

```
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. 登入資料檔案

Important

如果您已選取上述其中一種方法，則無法使用 `credential-path kvssink` 參數。

kvssink 參數名稱	必要	描述
<code>credential-path</code>	是	文字檔案的路徑，其中包含特定格式的登入資料。

文字檔案必須包含下列其中一種格式的登入資料：

- CREDENTIALS *YourAccessKey YourSecretKey*
- CREDENTIALS *YourAccessKey ## YourSecretKey SessionToken*

Example

範例：您的 `credentials.txt` 檔案位於 `/home/ubuntu`，並包含下列項目：

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEEXAMPLE STEXAMPLE
```

若要在 `gst-launch-1.0` 中使用它 `kvssink`，請輸入：

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
credential-path="/home/ubuntu/credentials.txt"
```

Note

過期時間應該至少為 $5 + 30 + 3 =$ 未來 38 秒。寬限期在 中定義為 IOT_CREDENTIAL_FETCH_GRACE_PERIOD 變數 [IotCredentialProvider.h](#)。如果登入資料在您啟動 時太接近過期 kvssink，您會收到錯誤碼 0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION。

Important

kvssink 不會修改登入資料檔案。如果您使用的是暫時登入資料，則必須在過期時間減去寬限期之前，由外部來源更新登入資料檔案。

提供區域給 kvssink

以下是區域查詢順序：

1. AWS_DEFAULT_REGION 環境變數會先檢閱。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. 如果之前的方法都未使用，則 kvssink 預設為 us-west-2。

kvssink 選用參數

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 影片串流結構](#)。

參數	描述	單位/類型	預設
stream-name	目的地 Amazon Kinesis 視訊串流的名稱。		

參數	描述	單位/類型	預設
	<p> Important</p> <p>如果未指定串流名稱，將使用預設串流名稱：「DEFAULT_STREAM」。如果具有該預設名稱的串流尚不存在，則會建立該串流。</p>		
absolute-fragment-times	是否使用絕對片段時間。	Boolean	true
access-key	<p>用來存取 Kinesis Video Streams 的 AWS 存取金鑰。</p> <p>您必須設定 AWS 登入資料或提供此參數。若要提供此資訊，請輸入下列內容：</p> <pre>export AWS_ACCESS_KEY_ID=</pre>		
avg-bandwidth-bps	串流的預期平均頻寬。	每秒位元數	4194304

參數	描述	單位/類型	預設
aws-region	<p>AWS 區域 要使用的。</p> <div data-bbox="472 352 792 953" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note 您也可以為區域提供 AWS_DEFAULT_REGION 環境變數。如果同時設定環境變數和 kvssink 參數，則環境變數優先。</p> </div> <div data-bbox="472 1020 792 1335" style="border: 1px solid #ff9999; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Important us-west-2 如果未另外指定，則區域預設為。</p> </div>	字串	"us-west-2"
buffer-duration	串流緩衝持續時間。	秒鐘	120
codec-id	串流的編解碼器 ID。	字串	"V_MPEG4/ISO/AVC"
connection-staleness	之後呼叫串流過時回呼的時間。	秒鐘	60
content-type	串流的內容類型。	字串	"video/h264"

參數	描述	單位/類型	預設
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。如需詳細資訊，請參閱 輪換 AWS KMS 金鑰 。	秒鐘	3600
secret-key	用來存取 Kinesis Video Streams 的 AWS 私密金鑰。 您必須設定 AWS 登入資料或提供此參數。 <pre>export AWS_SECRET_ACCESS_KEY=</pre>		
session-token	如果您直接從 AWS STS 操作使用臨時安全登入資料，請指定必要的工作階段字符值。		
storage-size	以 MB (MiB) 為單位的裝置儲存體大小。如需設定裝置儲存的詳細資訊，請參閱 StorageInfo 。	Mebibyte (MiB)	128
streaming-type	串流類型。有效值包含： <ul style="list-style-type: none"> • 0：即時 • 1：近乎即時 (目前不支援) • 2：離線 	列舉 GstKvsSinkStreamingType	0：即時
timecode-scale	MKV 時間碼比例。	毫秒	1

參數	描述	單位/類型	預設
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 要用於 kvssink 元素的 憑證。</p> <p>iot-certificate 接受下列金鑰和值：</p> <div data-bbox="472 527 792 1035" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>iot-thing -name 是選用的。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"> • endpoint=iotcredentialsproviderendpoint • cert-path=/localdirectorypath/to/certificate • key-path=/localdirectorypath/to/private/key • ca-path=/localdir 	字串	無

參數	描述	單位/類型	預設
	ectorypath/ to/ca-cert <ul style="list-style-type: none"> role-aliases =role-aliases iot-thing-name=YourIotThingName 		

範例：使用 PutMedia API 將資料傳送至 Kinesis Video Streams

本範例示範如何使用 [PutMedia](#) API。它說明如何傳送已採用容器格式 (MKV) 的資料。如果您的資料必須在傳送之前組合成容器格式（例如，如果您要將攝影機視訊資料組合成影格），請參閱 [上傳至 Kinesis Video Streams](#)。

Note

PutMedia 此操作僅適用於 C++ 和 Java SDKs。這是因為連線、資料流程和確認的全雙工管理。其他語言不支援此功能。

本範例包含下列步驟：

- [下載並設定程式碼](#)
- [撰寫並檢查程式碼](#)
- [執行並驗證程式碼](#)

下載並設定程式碼

依照步驟下載 Java 範例程式碼、將專案匯入您的 Java IDE、設定程式庫位置，以及設定程式碼以使用您的登入資料 AWS。

1. 建立目錄，由 GitHub 儲存庫複製範例原始碼。PutMedia 範例為 [Java](#) 的一部分。

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

2. 開啟您正在使用的 Java IDE (例如 [Eclipse](#) 或 [IntelliJ IDEA](#))，並匯入您下載的 Apache Maven 專案：

- 在 Eclipse 中：請選擇 File (檔案)、Import (匯入)、Maven、Existing Maven Projects (現有 Maven 專案)，並巡覽至下載套件的根目錄。選取 pom.xml 檔案。
- 在 IntelliJ Idea：選擇 Import (匯入)。巡覽至已下載套件根目錄的 pom.xml 檔案。

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。

撰寫並檢查程式碼

PutMedia API 範例 (PutMediaDemo) 顯示下列編碼模式：

主題

- [建立 PutMediaClient](#)
- [串流媒體並暫停執行緒](#)

本節程式碼範例來自 PutMediaDemo 類別。

建立 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 生產者程式庫](#) (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

PutMedia API 範例不會建立串流。您必須使用的測試應用程式 [使用 Java 生產者程式庫](#)、Kinesis Video Streams 主控台或來建立串流 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();
```

執行並驗證程式碼

如欲執行 PutMedia API 範例，請執行下列步驟：

1. 在 `my-stream` Kinesis Video Streams 主控台或使用 `建立名為` 的串流 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，並提供您的 AWS 登入資料：

```
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 來源串流

[C++](#) 包含連接至即時串流通訊協定 (RTSP) 網路攝影機之 [Docker](#) 容器的定義。使用 Docker 標準化 Kinesis Video Streams 的操作環境，可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

- [教學課程影片](#)
- [先決條件](#)
- [建置 Docker 映像](#)
- [執行 RTSP 範例應用程式](#)

教學課程影片

此影片說明如何設定 Raspberry Pi，將 RTSP 摘要傳送至 AWS 雲端和 Amazon Kinesis Video Streams。這是end-to-end示範。

此影片示範如何從摘要擷取影像，以使用電腦視覺和 Amazon Rekognition 來處理影像並傳送提醒。

先決條件

若要執行 Kinesis Video Streams RTSP 範例應用程式，您必須具有下列項目：

- 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](#)。

建置 Docker 映像

首先，建置示範應用程式將在其中執行的 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. 變更為包含 Dockerfile 的目錄。在這種情況下，它是 [docker-rtsp](#) 目錄。

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. 使用下列命令來建置 Docker 映像。此命令會建立映像並將其標記為 rtspdockertest。

```
docker build -t rtspdockertest .
```

4. 執行 `docker images` 並搜尋以標記的影像 ID `rtspdockertest`。

例如，在下面的範例輸出中，IMAGE ID是 54f0d65f69b2。

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB

在後續步驟中，您將需要此項目。

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 Docker 容器中](#)
- [Docker 容器外部](#)

在 Docker 容器中

執行 RTSP 範例應用程式

1. 使用以下命令啟動 Amazon Kinesis Video Streams Docker 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 影片串流的名稱，以及 RTSP 網路攝影機的 URL。

Important

如果您使用的是臨時登入資料，您還需要提供您的 AWS_SESSION_TOKEN。請參閱以下第二個範例。

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId  
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId  
export AWS_DEFAULT_REGION=YourAWSRegion
```

```
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

暫時登入資料：

```
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
```

- 登入 AWS Management Console 並開啟 [Kinesis Video Streams 主控台](#)。

檢視串流。

- 若要結束 Docker 容器，請關閉終端機視窗或輸入 `exit`。

Docker 容器外部

從 Docker 容器外部，使用下列命令：

```
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=Your AWS Region; ./  
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

範例：剖析和轉譯 Kinesis Video Streams 片段

[使用剖析器程式庫進行串流](#) 包含名為 `KinesisVideoRendererExample` 的示範應用程式，示範剖析和轉譯 Amazon Kinesis 影片串流片段。範例使用 [JCodec](#) 解碼 H.264 編碼影格 (以 [範例：Kinesis Video Streams 生產者 SDK GStreamer 外掛程式 - kvssink](#) 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 影片串流擷取影格，並轉譯串流以供檢視。
- 在自訂應用程式中檢視串流的影片內容，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 影片串流內容，例如在顯示之前不需要解碼的 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。

先決條件

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](#)。
- Java 整合開發環境 (IDE)，例如 [Eclipse Java Neon](#) 或 [JetBrains IntelliJ Idea](#)。

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)。導覽至 `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/KinesisVideoRendererExampleTest`。
4. 由檔案移除 `@Ignore` 指令。

5. 使用 Kinesis 影片串流的名稱更新 `.stream` 參數。
6. 執行 `KinesisVideoRendererExample` 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 片段剖析為影格](#)
- [解碼和顯示影格](#)

傳送 MKV 資料

範例傳送 `rendering_example_video.mkv` 檔案的 MKV 資料，使用 `PutMedia` 傳送影片資料至名為 `render-example-stream` 的串流。

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 [使用剖析器程式庫進行串流](#) 文件)。

將 MKV 片段剖析為影格

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 [使用剖析器程式庫進行串流](#) 文件)。

解碼和顯示影格

範例之後使用 [`JFrame`](#) 解碼及顯示影格。

下列程式碼範例來自 `KinesisVideoFrameViewer` 類別，延伸 `JFrame`：

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
    repaint();
}
```

影像以 [`java.awt.image.BufferedImage`](#) 的執行個體顯示。如需如何使用 `BufferedImage` 的範例，請參閱 [讀取/載入影像](#)。

API 參考

此節點下的區段包含 API 參考文件。使用左側窗格中的目錄，前往不同的 API 參考區段。

動作

Amazon Kinesis Video Streams 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 Video Streams Media 支援下列動作：

- [GetMedia](#)
- [PutMedia](#)

Amazon Kinesis Video Streams Archived Media 支援下列動作：

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

Amazon Kinesis Video Signaling Channels 支援下列動作：

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

Amazon Kinesis Video WebRTC Storage 支援下列動作：

- [JoinStorageSession](#)
- [JoinStorageSessionAsViewer](#)

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams 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

建立訊號頻道。

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。每個 AWS 帳戶 和 都必須是唯一的 AWS 區域。

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頻道類型組態的結構。頻道訊息的存留時間為 60 秒 (1 分鐘)。

類型：[SingleMasterConfiguration](#) 物件

必要：否

Tags

您要與此頻道建立關聯的一組標籤（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

所建立頻道的 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]+`

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

AccountChannelLimitExceededException

您已達到此區域中作用中 AWS 帳戶 訊號頻道的上限。

HTTP 狀態碼：400

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入 StreamARN 或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。

2. 用來判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream 或 DescribeSignalingChannel API。

HTTP 狀態碼：400

TagsPerResourceExceededLimitException

您已超過可與資源建立關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

CreateStream

服務：Amazon Kinesis Video Streams

建立新的 Kinesis 影片串流。

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](#)

您想要在串流中保留資料的時數。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。最短為 1 小時。

當DataRetentionInHours值為 0 時，消費者仍然可以使用保留在服務主機緩衝區中的片段，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，片段會從緩衝區中移除。

類型：整數

有效範圍：最小值為 0。

必要：否

DeviceName

正在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 Video Streams 用來加密串流資料的 AWS Key Management Service (AWS KMS) 金鑰 ID。

如果未指定金鑰 ID，則會使用預設的 Kinesis Video 受管金鑰 (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。串流的消費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 準則的 [命名要求](#)。

範例有效值包括 "video/h264" 和 "video/h264 , audio/aac"。

此參數為選用；預設值為 null (或在 JSON 中為空)。

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符，而且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

要與指定串流建立關聯的標籤清單。每個標籤都是索引鍵/值對 (該值為選用)。

類型：字串到字串映射

映射項目：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：長度下限為 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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccountStreamLimitExceededException

為帳戶建立的串流數目太高。

HTTP 狀態碼：400

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用來判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel API。

HTTP 狀態碼：400

TagsPerResourceExceededLimitException

您已超過可與資源建立關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)

- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DeleteEdgeConfiguration

服務：Amazon Kinesis Video Streams

非同步 API，可刪除串流的現有邊緣組態，以及 Edge Agent 中對應的媒體。

當您呼叫此 API 時，同步狀態會設為 DELETING。刪除程序會開始，其中作用中的邊緣任務會停止，且所有媒體都會從邊緣裝置刪除。刪除的時間會有所不同，取決於儲存媒體的總量。如果刪除程序失敗，同步狀態會變更為 DELETE_FAILED。您需要重試刪除。

當刪除程序成功完成時，就無法再存取邊緣組態。

Note

此 API 不適用於 AWS 非洲（開普敦）區域 af-south-1。

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

要從中刪除邊緣組態的串流名稱。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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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 Video Stream 找不到您指定的串流邊緣組態時呈現的例外狀況。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DeleteSignalingChannel

服務：Amazon Kinesis Video Streams

刪除指定的訊號頻道。DeleteSignalingChannel 是一種非同步操作。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

您要刪除之訊號頻道的 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：是

CurrentVersion

您要刪除之訊號頻道的目前版本。您可以透過叫用 DescribeSignalingChannel 或 ListSignalingChannels API 操作來取得目前版本。

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

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

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入 StreamARN 或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。

3. 用於判斷資源狀態的 DescribeStream 或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DeleteStream

服務：Amazon Kinesis Video Streams

刪除 Kinesis 影片串流和串流中包含的資料。

此方法會將串流標記為刪除，並使串流中的資料無法立即存取。

若要確保在刪除之前擁有最新版本的串流，您可以指定串流版本。Kinesis Video Streams 會為每個串流指派版本。當您更新串流時，Kinesis Video Streams 會指派新的版本編號。若要取得最新的串流版本，請使用 DescribeStream API。

這項操作需要 KinesisVideo:DeleteStream 動作的許可。

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：您要刪除的串流版本。

指定 版本作為保護，以確保您的 正在刪除正確的串流。若要取得串流版本，請使用 DescribeStream API。

如果未指定，CreationTime則只會在刪除串流之前檢查。

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

您要刪除之串流的 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：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](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)

- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeEdgeConfiguration

服務：Amazon Kinesis Video Streams

描述使用 `StartEdgeConfigurationUpdate` API 設定的串流邊緣組態，以及邊緣代理程式記錄器和上傳器任務的最新狀態。使用此 API 取得組態的狀態，以判斷組態是否與 Edge Agent 同步。使用此 API 來評估 Edge Agent 的運作狀態。

Note

此 API 不適用於 AWS 非洲（開普敦）區域 `af-south-1`。

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

您要更新其邊緣組態的串流名稱。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組態的時間戳記。

類型：Timestamp

EdgeAgentStatus

包含邊緣代理程式記錄器和上傳器任務最新狀態詳細資訊的物件。使用此資訊來判斷邊緣代理程式目前的運作狀態。

類型：[EdgeAgentStatus](#) 物件

[EdgeConfig](#)

串流邊緣組態的描述，將用於與 Edge Agent IoT Greengrass 元件同步。Edge Agent 元件將在您現場部署的 IoT Hub 裝置設定上執行。

類型：[EdgeConfig](#) 物件

[FailedStatusDetails](#)

所產生失敗狀態的描述。

類型：字串

[LastUpdatedTime](#)

上次更新串流邊緣組態的時間戳記。

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](#)

邊緣組態更新來源的串流名稱。

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_.-]+`

[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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

Amazon Kinesis Video Stream 找不到您指定的串流邊緣組態時出現的例外狀況。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeImageGenerationConfiguration

服務：Amazon Kinesis Video Streams

取得指定 Kinesis 影片串流 ImageGenerationConfiguration 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

要從中擷取映像產生組態的 Kinesis 視訊串流的 Amazon Resource Name (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 影片串流 (KVS) 影像交付所需資訊的結構。如果此結構為 null，則會從串流中刪除組態。

類型：[ImageGenerationConfiguration](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)

- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeMappedResourceConfiguration

服務：Amazon Kinesis Video Streams

傳回串流的最新資訊。輸入中streamARN應該提供 streamName或。

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

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

在您的下一個請求中提供的字符，以取得另一批結果。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

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 請求中用來擷取下一組結果的字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeMediaStorageConfiguration

服務：Amazon Kinesis Video Streams

傳回頻道的最新資訊。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](#)

頻道的 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](#)

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組態屬性的結構。

類型：[MediaStorageConfiguration](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeNotificationConfiguration

服務：Amazon Kinesis Video Streams

取得指定 Kinesis 影片串流 NotificationConfiguration 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

您要從中擷取通知組態之 Kinesis 視訊串流的 Amazon Resource Name (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

要從中擷取通知組態的串流名稱。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

{
  "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，則會從串流中刪除組態。

類型：[NotificationConfiguration](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeSignalingChannel

服務：Amazon Kinesis Video Streams

傳回有關訊號頻道的最新資訊。您必須指定要描述之頻道的名稱或 Amazon Resource Name (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

您要描述之訊號頻道的 ARN。

類型：字串

長度限制：長度下限為 1。長度上限為 1024。

模式：arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

必要：否

ChannelName

您要描述的訊號頻道名稱。

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

{
  "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](#)

封裝指定訊號頻道中繼資料和屬性的結構。

類型：[ChannelInfo](#) 物件

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DescribeStream

服務：Amazon Kinesis Video Streams

傳回指定串流的最新資訊。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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetDataEndpoint

服務：Amazon Kinesis Video Streams

取得指定串流的端點以進行讀取或寫入。在應用程式中使用此端點從指定的串流讀取（使用 `GetMedia` 或 `GetMediaForFragmentList` 操作）或寫入（使用 `PutMedia` 操作）。

Note

傳回的端點未附加 API 名稱。用戶端需要將 API 名稱新增至傳回的端點。

在請求中，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

要取得端點的 API 動作名稱。

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

您要取得端點之串流的 Amazon Resource Name (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

您要取得端點的串流名稱。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其中，請在應用程式中指定此端點。

類型：字串

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)

- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetSignalingChannelEndpoint

服務：Amazon Kinesis Video Streams

提供指定訊號頻道的端點，以傳送和接收訊息。此 API 使用 `SingleMasterChannelEndpointConfiguration` 輸入參數，其中包含 `Protocols` 和 `Role` 屬性。

`Protocols` 用來判斷通訊機制。例如，如果您指定 WSS 做為通訊協定，此 API 會產生安全的 Websocket 端點。如果您指定 HTTPS 做為通訊協定，此 API 會產生 HTTPS 端點。如果您將指定 WEBRTC 為通訊協定，但訊號頻道未設定為擷取，您將會收到錯誤 `InvalidArgumentException`。

`Role` 會決定簡訊許可。MASTER 角色會導致此 API 產生端點，用戶端可用來與頻道上的任何檢視器進行通訊。VIEWER 角色會導致此 API 產生端點，用戶端只能用來與 通訊 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

您要取得端點之訊號管道的 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：是

[SingleMasterChannelEndpointConfiguration](#)

包含SINGLE_MASTER頻道類型端點組態的結構。

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](#)

指定訊號頻道的端點清單。

類型：[ResourceEndpointListItem](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)

- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListEdgeAgentConfigurations

服務：Amazon Kinesis Video Streams

傳回與指定 Edge Agent 相關聯的邊緣組態陣列。

在請求中，您必須指定 Edge Agent HubDeviceArn。

Note

此 API 不適用於 AWS 非洲（開普敦）區域 af-south-1。

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) 物件」。

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

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ListEdgeAgentConfigurations操作結果截斷時，呼叫會在回應NextToken中傳回。若要取得另一批邊緣組態，請在下一個請求中提供此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

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](#)

單一串流邊緣組態的描述。

類型：[ListEdgeAgentConfigurationsEdgeConfig](#) 物件陣列

[NextToken](#)

如果回應被截斷，呼叫會傳回具有指定字符的此元素。若要取得下一批邊緣組態，請在下一個請求中使用此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

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 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListSignalingChannels

服務：Amazon Kinesis Video Streams

傳回 ChannelInfo 物件陣列。每個物件都會描述訊號頻道。若要僅擷取滿足特定條件的頻道，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：僅傳回滿足特定條件的頻道。

類型：[ChannelNameCondition](#) 物件

必要：否

MaxResults

回應中傳回的頻道數目上限。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ListSignalingChannels操作結果截斷時，呼叫會在回應NextToken中傳回。若要取得另一批頻道，請在下一個請求中提供此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

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](#)

如果回應被截斷，呼叫會傳回此元素與字符。若要取得下一批串流，請在下一個請求中使用此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)

- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListStreams

服務：Amazon Kinesis Video Streams

傳回 StreamInfo 物件陣列。每個物件都會描述串流。若要僅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

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 ListStreams 操作結果截斷時，呼叫會在回應 NextToken 中傳回。若要取得另一批串流，請在下一個請求中提供此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

必要：否

StreamNameCondition

選用：僅傳回滿足特定條件的串流。目前，您只能將串流名稱的字首指定為條件。

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](#)

如果回應被截斷，呼叫會傳回此元素與字符。若要取得下一批串流，請在下一個請求中使用此字符。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

[StreamInfoList](#)

StreamInfo 物件的陣列。

類型：[StreamInfo](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)

- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListTagsForResource

服務：Amazon Kinesis Video Streams

傳回與指定訊號通道相關聯的標籤清單。

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呼叫的結果被截斷，則回應會包含一個字符，您可以在下一個請求中使用它來擷取下一批標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：[a-zA-Z0-9+/=]*

必要：否

[ResourceARN](#)

您要列出標籤之訊號通道的 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：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` 呼叫的結果被截斷，則回應會包含一個字符，您可以在下一個請求中使用它來擷取下一組標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

[Tags](#)

與指定訊號通道相關聯的標籤索引鍵和值映射。

類型：字串到字串映射

映射項目：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：長度下限為 0。長度上限為 256。

值模式：`[\p{L}\p{Z}\p{N}_.:/+\\-@]*`

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)

- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListTagsForStream

服務：Amazon Kinesis Video Streams

傳回與指定串流相關聯的標籤清單。

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 呼叫的結果被截斷，則回應會包含權杖，您可以在下一個請求中使用它來擷取下一批標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：[a-zA-Z0-9+/=]*

必要：否

[StreamARN](#)

您要列出標籤之串流的 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

您要列出標籤的串流名稱。

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
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` 呼叫的結果遭到截斷，則回應會包含權杖，您可以在下一個請求中使用它來擷取下一組標籤。

類型：字串

長度限制：長度下限為 0。長度上限為 1024。

模式：`[a-zA-Z0-9+/=]*`

Tags

與指定串流相關聯的標籤索引鍵和值映射。

類型：字串到字串映射

映射項目：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：長度下限為 0。長度上限為 256。

值模式：`[\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*`

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

StartEdgeConfigurationUpdate

服務：Amazon Kinesis Video Streams

非同步 API，可更新串流的現有邊緣組態。Kinesis Video Stream 會將串流的邊緣組態與在 IoT Hub 裝置上執行的 Edge 代理程式 IoT Greengrass 元件同步，該元件會在您現場部署中設定。同步的時間可能有所不同，取決於中樞裝置的連線。當邊緣組態確認並與 Edge Agent 同步時，SyncStatus 將會更新。

如果第一次叫用此 API，則會為串流建立新的邊緣組態，並將同步狀態設定為 SYNCING。您必須等待同步狀態達到終端狀態，例如：IN_SYNC 或 SYNC_FAILED，才能再次使用此 API。如果您在同步程序期間叫用此 API，ResourceInUseException 則會擲出。串流邊緣組態和 Edge Agent 的連線將會重試 15 分鐘。15 分鐘後，狀態會轉換為 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](#) 刪除目前的邊緣組態。然後，您可以使用更新的 Hub 裝置 ARN 叫用 StartEdgeConfigurationUpdate。

Note

此 API 不適用於 AWS 非洲（開普敦）區域 af-south-1。

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](#)

叫用更新程序所需的邊緣組態詳細資訊。

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

您要更新其邊緣組態的串流名稱。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組態的時間戳記。

類型：Timestamp

[EdgeConfig](#)

串流邊緣組態的描述，將用於與 Edge Agent IoT Greengrass 元件同步。Edge Agent 元件將在您現場部署的 IoT Hub 裝置設定上執行。

類型：[EdgeConfig](#) 物件

[FailedStatusDetails](#)

所產生失敗狀態的描述。

類型：字串

[LastUpdatedTime](#)

上次更新串流邊緣組態的時間戳記。

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

邊緣組態更新來源的串流名稱。

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_.-]+`

SyncStatus

串流邊緣組態的目前同步狀態。當您叫用此 API 時，同步狀態會設為 SYNCING 狀態。使用 DescribeEdgeConfiguration API 取得邊緣組態的最新狀態。

類型：字串

有效值:SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NoDataRetentionException

串流資料保留小時數等於零。

HTTP 狀態碼：400

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

TagResource

服務：Amazon Kinesis Video Streams

將一或多個標籤新增至訊號頻道。標籤是您可以定義和指派給 AWS 資源的鍵/值對（值為選用）。如果您指定已存在的標籤，您在請求中指定的值會取代標籤值。如需詳細資訊，請參閱《AWS 帳單與成本管理 和成本管理使用者指南》中的[使用成本分配標籤](#)。

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](#)

您要新增標籤之訊號通道的 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：是

Tags

要與指定訊號頻道建立關聯的標籤清單。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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

您已超過可與資源建立關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

TagStream

服務：Amazon Kinesis Video Streams

將一或多個標籤新增至串流。標籤是您可以定義和指派給 AWS 資源的鍵/值對（值為選用）。如果您指定已存在的標籤，您在請求中指定的值會取代標籤值。如需詳細資訊，請參閱《AWS 帳單與成本管理 和成本管理使用者指南》中的[使用成本分配標籤](#)。

您必須提供 StreamName 或 StreamARN。

這項操作需要 KinesisVideo:TagStream 動作的許可。

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之資源的 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

您要新增標籤或標籤的串流名稱。

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

Tags

要與指定串流建立關聯的標籤清單。每個標籤都是索引鍵/值對（該值為選用）。

類型：字串到字串映射

映射項目：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： $^{\wedge}([\backslash\{L\}\backslash\{Z\}\backslash\{N\}_\cdot\:/=\+\backslash-\@]^*)^{\$}$

值長度限制：長度下限為 0。長度上限為 256。

值模式： $[\backslash\{L\}\backslash\{Z\}\backslash\{N\}_\cdot\:/=\+\backslash-\@]^*$

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](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

TagsPerResourceExceededLimitException

您已超過可與資源建立關聯的標籤限制。Kinesis 影片串流最多可支援 50 個標籤。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)

- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UntagResource

服務：Amazon Kinesis Video Streams

從訊號頻道移除一或多個標籤。在請求中，僅指定標籤索引鍵或索引鍵；不要指定值。如果您指定不存在的標籤金鑰，則會予以忽略。

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

您要從中移除標籤之訊號通道的 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：是

TagKeyList

您要移除之標籤的索引鍵清單。

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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UntagStream

服務：Amazon Kinesis Video Streams

從串流移除一或多個標籤。在請求中，僅指定標籤索引鍵或索引鍵；請勿指定值。如果您指定不存在的標籤金鑰，則會予以忽略。

在請求中，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

您要從中移除標籤之串流的 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

您要從中移除標籤的串流名稱。

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

TagKeyList

您要移除之標籤的索引鍵清單。

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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

Amazon Kinesis Video Streams 找不到您指定的串流。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UpdateDataRetention

服務：Amazon Kinesis Video Streams

依您指定的值增加或減少串流的資料保留期間。若要指出您是否要增加或減少資料保留期間，請在請求內文中指定 `Operation` 參數。在請求中，您必須指定 `StreamName` 或 `StreamARN`。

這項操作需要 `KinesisVideo:UpdateDataRetention` 動作的許可。

變更資料保留期間會影響串流中的資料，如下所示：

- 如果資料保留期增加，現有資料會保留在新的保留期內。例如，如果資料保留期間從 1 小時增加到 7 小時，則所有現有的資料都會保留 7 小時。
- 如果資料保留期間減少，則會在新的保留期間保留現有資料。例如，如果資料保留期間從 7 小時縮短為 1 小時，則所有現有的資料都會保留 1 小時，而且任何超過 1 小時的資料都會立即刪除。

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之串流的版本。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

要調整目前保留的時數。您指定的值會根據新增至目前值或從中減去 operation。

資料保留的最小值為 0，最大值為 87600（十年）。

類型：整數

有效範圍：最小值為 1。

必要：是

Operation

指出您是否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

您要變更其保留期間之串流的 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

您要變更其保留期間之串流的名稱。

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
```

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](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceInUseException

當輸入 StreamARN 或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入 StreamARN 或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。

2. 用於判斷指定串流所對應之頻道的 `DescribeMappedResourceConfiguration` API。
3. 用於判斷資源狀態的 `DescribeStream`或 `DescribeSignalingChannel` 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

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 映像交付所需資訊的結構。如果結構為 null，則會從串流中刪除組態。

類型：[ImageGenerationConfiguration](#) 物件

必要：否

StreamARN

您要更新映像產生組態之 Kinesis 視訊串流的 Amazon Resource Name (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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NoDataRetentionException

串流資料保留小時數等於零。

HTTP 狀態碼：400

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UpdateMediaStorageConfiguration

服務：Amazon Kinesis Video Streams

將 SignalingChannel 與串流建立關聯以存放媒體。您可以指定兩種訊號模式：

- 如果 StorageStatus 啟用，資料將存放在 StreamARN 提供的 中。為了讓 WebRTC 擷取正常運作，串流必須啟用資料保留。
- 如果 StorageStatus 已停用，則不會儲存任何資料，而且不需要 StreamARN 參數。

Important

如果 StorageStatus 啟用，則不會再發生 peer-to-peer (主要檢視者) 連線。對等直接連線到儲存工作階段。您必須呼叫 JoinStorageSession API 來觸發 SDP 優惠傳送，並在對等和儲存工作階段之間建立連線。

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

頻道的 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：是

[MediaStorageConfiguration](#)

封裝或包含媒體儲存組態屬性的結構。

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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NoDataRetentionException

串流資料保留小時數等於零。

HTTP 狀態碼：400

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用來判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

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，則會從串流中刪除組態。

類型：[NotificationConfiguration](#) 物件

必要：否

StreamARN

您要更新通知組態之 Kinesis 視訊串流的 Amazon Resource Name (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

要從中更新通知組態的串流名稱。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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NoDataRetentionException

串流資料保留小時數等於零。

HTTP 狀態碼：400

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用來判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)

- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UpdateSignalingChannel

服務：Amazon Kinesis Video Streams

更新現有的訊號頻道。這是非同步操作，需要一些時間才能完成。

如果MessageTtlSeconds值已更新（增加或減少），則僅適用於更新後透過此頻道傳送的新訊息。現有訊息仍會根據先前的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

您要更新的訊號頻道的 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：是

CurrentVersion

您要更新的訊號頻道目前版本。

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

包含您要更新之訊號頻道SINGLE_MASTER類型的組態的結構。具有頻道類型的頻道需要此參數和SINGLE_MASTER頻道訊息的time-to-live。

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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

AccessDeniedException

您沒有執行此操作所需的許可。

HTTP 狀態碼：401

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用於判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)

- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

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。

您必須提供串流的名稱或串流的 Amazon Resource Name (ARN)。

若要確保在更新之前擁有最新版本的串流，您可以指定串流版本。Kinesis Video Streams 會為每個串流指派版本。當您更新串流時，Kinesis Video Streams 會指派新的版本編號。若要取得最新的串流版本，請使用 DescribeStream API。

UpdateStream 是一種非同步操作，需要一些時間才能完成。

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

正在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。使用 `MediaType` 指定串流包含給串流消費者的內容類型。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](#)。

若要在主控台上播放影片，您必須指定正確的影片類型。例如，如果串流中的視訊是 H.264，請將指定 `video/h264` 為 `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

StreamARN

您要更新其中繼資料之串流的 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-]+: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符，而且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

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceInUseException

當輸入StreamARN或 ChannelARN 中的 CLOUD_STORAGE_MODE 已映射至不同的 Kinesis Video Stream 資源，或提供的輸入StreamARN或 ChannelARN 未處於作用中狀態時，請嘗試下列其中一個：

1. 用於判斷串流指定頻道所對應的 DescribeMediaStorageConfiguration API。
2. 用來判斷指定串流所對應之頻道的 DescribeMappedResourceConfiguration API。
3. 用於判斷資源狀態的 DescribeStream或 DescribeSignalingChannel 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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Streams 媒體

Amazon Kinesis Video Streams Media 支援下列動作：

- [GetMedia](#)
- [PutMedia](#)

GetMedia

服務：Amazon Kinesis Video Streams Media

使用此 API 從 Kinesis 影片串流擷取媒體內容。在請求中，您可以識別串流名稱或串流 Amazon Resource Name (ARN) 和開始區塊。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 才能取得端點。然後使用 `--endpoint-url` 參數將 `GetMedia` 請求傳送至此端點。

當您將媒體資料（片段）放在串流時，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」的內容中。如需詳細資訊，請參閱 [PutMedia](#)。GetMedia API 會從您在請求中指定的區塊開始傳回這些區塊的串流。

GetMedia API 旨在透過長時間執行的連線做為串流 API 運作。它不適用於以傳統 RESTful 方式使用，其中會為每個片段建立新的 HTTP 連線並關閉。當您叫用 GetMedia API 時，Kinesis Video Streams 會使用 HTTP 區塊傳輸編碼，透過持久性長時間執行連線持續交付片段。

使用 GetMedia API 時適用下列限制：

- 用戶端每秒最多可呼叫每個串流 GetMedia 五次。
- Kinesis Video Streams 會在 GetMedia 工作階段期間以每秒高達 25 MB（或每秒 200 MB）的速率傳送媒體資料。

Note

使用 GetMedia 做為串流長時間執行的連線，擷取單一持久性連線中的多個片段。請勿在建立和關閉每個片段的新 HTTP 連線時，以傳統 RESTful 方式使用 GetMedia API。如果您嘗試三個以上的並行 GetMedia 連線，Kinesis Video Streams 會使用 `ConnectionLimitExceededException` 錯誤調節最新的連線。

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，則讀取 HTTP 回應承載會在 3 秒後逾時。

Note

如果在叫用 Kinesis Video Streams 媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 - 如果您想要向回報問題 AWS，支援團隊可以在指定請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。如需詳細資訊，請參閱本主題底部的錯誤區段，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

識別要從指定串流取得的開始區塊。

類型：[StartSelector](#) 物件

必要：是

StreamARN

您想要從中取得媒體內容的串流 ARN。如果您未指定 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

您想要從中取得媒體內容的 Kinesis 影片串流名稱。如果您未指定 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

請求媒體的內容類型。

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 傳回是來自指定串流的一系列區塊。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額外的 Matroska (MKV) 標籤：

- `AWS_KINESISVIDEO_CONTINUATION_TOKEN` (UTF-8 字串) - 如果您的 GetMedia 呼叫終止，您可以在下一個請求中使用此接續字符，以取得最後一個請求終止的下一個區塊。
- `AWS_KINESISVIDEO_MILLIS_BEHIND_NOW` (UTF-8 字串) - 用戶端應用程式可以使用此標籤值來判斷回應中傳回的區塊落後程度，是來自串流上的最新區塊。
- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - 區塊中傳回的片段編號。
- `AWS_KINESISVIDEO_SERVER_TIMESTAMP` - 片段的伺服器時間戳記。
- `AWS_KINESISVIDEO_PRODUCER_TIMESTAMP` - 片段的生產者時間戳記。

如果發生錯誤，將出現下列標籤：

- `AWS_KINESISVIDEO_ERROR_CODE` - 導致 GetMedia 停止之錯誤的字串描述。
- `AWS_KINESISVIDEO_ERROR_ID`：錯誤整數代碼。

錯誤代碼如下所示：

- 3002 - 寫入串流時發生錯誤
- 4000 - 找不到請求的片段
- 4500 - 拒絕存取串流的 KMS 金鑰
- 4501 - 串流的 KMS 金鑰已停用
- 4502 - 串流 KMS 金鑰上的驗證錯誤
- 4503 - 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
- 5000 - 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 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

ConnectionLimitExceededException

Kinesis Video Streams 已調節請求，因為您已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

來電者使用錯誤的端點將資料寫入串流。收到這類例外狀況時，使用者必須呼叫 `GetDataEndpoint`，並將 `APIName` 設定為 `PUT_MEDIA`，並使用回應中的端點來叫用下一個 `PutMedia` 呼叫。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權對指定的串流執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

狀態碼：404，具有指定名稱的串流不存在。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

PutMedia

服務：Amazon Kinesis Video Streams Media

使用此 API 將媒體資料傳送至 Kinesis 影片串流。

Note

您必須先呼叫 `GetDataEndpoint` API 才能取得端點。然後使用 `--endpoint-url` 參數將 `PutMedia` 請求傳送至此端點。

在請求中，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。您可以使用請求內文來傳送媒體資料。Kinesis Video Streams 僅支援使用此 API 傳送媒體資料的 Matroska (MKV) 容器格式。

您有下列選項可使用此 API 傳送資料：

- 即時傳送媒體資料：例如，安全攝影機可以在產生影格時即時傳送影格。此方法可將視訊錄製與線路上傳送的資料之間的延遲降至最低。這稱為連續生產者。在這種情況下，消費者應用程式可以即時或在需要時讀取串流。
- 離線傳送媒體資料（分批傳送）：例如，主體攝影機可能會錄製影片數小時，並將其存放在裝置上。稍後，當您將攝影機連接到停駐連接埠時，攝影機可以啟動 `PutMedia` 工作階段，將資料傳送至 Kinesis 影片串流。在此案例中，延遲不是問題。

使用此 API 時，請注意下列考量：

- 您必須指定 `streamName` 或 `streamARN`，但不能同時指定兩者。
- 若要能夠在主控台上或透過 HLS 播放媒體，每個片段的追蹤 1 應該包含 h.264 編碼的視訊、片段中繼資料中的 `CodecID` 應該是 "V_MPEG/ISO/AVC"，而片段中繼資料應該包含 AVCC 格式的 h.264 轉碼器私有資料。或者，每個片段的軌道 2 應該包含 AAC 編碼音訊，片段中繼資料中的 `CodecID` 應該是 "A_AAC"，而片段中繼資料應該包含 AAC 轉碼器私有資料。
- `PutMedia` API 旨在透過長時間執行的連線做為串流 API 運作。它不適用於以傳統 RESTful 方式使用，其中會為每個片段建立新的 HTTP 連線並關閉。使用 `PutMedia` API 時，請使用 HTTP 區塊傳輸編碼，透過持久性連線持續傳送片段。
- 對於 `PutMedia` 工作階段中接收的每個片段，Kinesis Video Streams 會傳送一或多個確認。潛在的用戶端網路考量可能會導致您無法在產生所有這些確認時取得這些確認。

Note

使用 PutMedia 做為串流長時間執行的連線，在單一持久性連線中傳送多個片段。如果您嘗試多個並行 PutMedia 連線，Kinesis Video Streams 會調節 ConnectionLimitExceededException 發生錯誤的最新連線。

使用 PutMedia API 時適用下列限制：

- 用戶端每秒每個串流 PutMedia 最多可呼叫 5 次。
- 用戶端每秒每個串流最多可傳送五個片段。
- Kinesis Video Streams 會在 PutMedia 工作階段期間以高達 12.5 MB/秒或 100 Mbps 的速率讀取媒體資料。

請注意下列限制條件。在這些情況下，Kinesis Video Streams 會在回應中傳送錯誤確認。

- 不允許時間代碼範圍超過允許上限且包含超過 50 MB 資料的片段。
- 不允許包含超過三個音軌的片段。每個片段中的每個影格都必須具有與片段標頭中定義的其中一個軌跡相同的軌跡編號。此外，每個片段在片段標頭中定義的每個軌道必須至少包含一個影格。
- 對於片段中繼資料中定義的每個軌跡，每個片段必須至少包含一個影格。
- 片段中最早的影格時間戳記必須晚於前一個片段中最新的影格時間戳記。
- 包含多個 MKV 區段或包含不允許 MKV 元素（例如 track*）的 MKV 串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存放在稱為「區塊」的內容中。片段中繼資料包括下列項目：

- PutMedia 請求開始時提供的 MKV 標頭
- 片段的下列 Kinesis Video Streams 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 - 生產者開始記錄片段時的時間戳記。Kinesis Video Streams 使用請求中接收的三個資訊片段來計算此值。
 - 請求內文中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（承載中的片段時間碼是絕對還是相對）。

Kinesis Video Streams 接著會運算片段 `producer_timestamp` 的，如下所示：

如果 `fragmentTimeCodeType` 是相對的，則

`producer_timestamp = producerStartTimeStamp + 片段時間碼`

如果 `fragmentTimeCodeType` 是絕對值，則

`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` 請求時，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

此操作僅適用於適用於 Java 的 AWS SDK。其他語言不支援 AWS SDKs。

Note

Kinesis Video Streams 不會在透過 `PutMedia` API 擷取和封存期間剖析和驗證轉碼器私有資料。KVS 透過 HLS APIs 使用串流時，會從 MPEG-TS 和 MP4 片段封裝的編解碼器私有資料擷取並驗證必要資訊。

Note

如果在叫用 Kinesis Video Streams 媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 - 如果您想要向 回報問題 AWS，支援團隊可以在指定請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。如需詳細資訊，請參閱本主題底部的錯誤區段，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](#)

您可以將此值做為 `x-amzn-fragment-timecode-type` HTTP 標頭傳遞。

指出片段中的時間碼（承載、HTTP 請求內文）是絕對還是相對於 `producerStartTimestamp`。Kinesis Video Streams 會使用此資訊來計算請求中接收 `producer_timestamp` 之片段的，如 API 概觀中所述。

有效值:ABSOLUTE | RELATIVE

必要：是

[ProducerStartTimestamp](#)

您可以將此值傳遞為 `x-amzn-producer-start-timestamp` HTTP 標頭。

這是生產者開始記錄媒體的生產者時間戳記（而非請求中特定片段的時間戳記）。

[StreamARN](#)

您傳遞此值做為 `x-amzn-stream-arn` HTTP 標頭。

您要寫入媒體內容之 Kinesis 影片串流的 Amazon Resource Name (ARN)。如果您未指定 `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

您可以將此值傳遞為 `x-amzn-stream-name` HTTP 標頭。

您要寫入媒體內容的 Kinesis 影片串流名稱。如果您未指定 `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 影片串流的媒體內容。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。區段可以包含一或多個叢集。

Note

每個 MKV 叢集都會映射至 Kinesis 影片串流片段。無論您選擇哪個叢集持續時間，都會變成片段持續時間。

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請求後，服務會驗證請求標頭。然後，服務會開始讀取承載，並先傳送 HTTP 200 回應。

然後，服務會傳回串流，其中包含一系列以換行分隔的 JSON 物件 (Acknowledgement 物件)。在傳送媒體資料的相同連線上收到確認。PutMedia 請求可以有許多確認。每個 Acknowledgement 包含下列鍵值對：

- AckEventType - 確認代表的事件類型。
 - 緩衝：Kinesis Video Streams 已開始接收片段。收到第一個片段資料位元組時，Kinesis Video Streams 會傳送第一個緩衝確認。
 - 已接收：Kinesis Video Streams 收到整個片段。如果您未將串流設定為保留資料，生產者可以在收到此確認時停止緩衝片段。
 - 持久性：Kinesis Video Streams 已保留片段 (例如 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。您可以檢閱錯誤代碼並判斷下一個動作。
 - 閒置：PutMedia工作階段正在進行中。不過，Kinesis Video Streams 目前未接收資料。Kinesis Video Streams 會在上次接收資料後最多 30 秒內定期傳送此認可。如果在 30 秒內未收到任何資料，Kinesis Video Streams 會關閉請求。

Note

此認可可協助生產者判斷PutMedia連線是否有效，即使未傳送任何資料也一樣。

- 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 - 傳送確認的 Kinesis Video Streams 產生的片段編號。
- 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 IDs 及其對應的錯誤代碼和錯誤訊息清單：

- 4000 - STREAM_READ_ERROR - 讀取資料串流時發生錯誤。
- 4001 - MAX_FRAGMENT_SIZE_REACHED - 片段大小大於上限，允許 50 MB。
- 4002 - MAX_FRAGMENT_DURATION_REACHED - 片段持續時間大於允許的上限。
- 4003 - MAX_CONNECTION_DURATION_REACHED - 連線持續時間大於允許的閾值上限。
- 4004 - FRAGMENT_TIMECODE_LESSER_THAN_PREVIOUS - 片段時間碼小於時間碼先前的時間碼（在PutMedia通話中，您無法依序傳送片段）。
- 4005 - more_THAN_ALLOWED_TRACKS_FOUND - 在 MKV 中找到多個軌道。（已棄用）
- 4006 - INVALID_MKV_DATA - 無法將輸入串流剖析為有效的 MKV 格式。
- 4007 - INVALID_PRODUCER_TIMESTAMP - 無效的生產者時間戳記。
- 4008 - STREAM_NOT_ACTIVE - 串流不再存在（已刪除）。
- 4009 - FRAGMENT_METADATA_LIMIT_REACHED - 已達到片段中繼資料限制。請參閱 開發人員指南的限制<https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/limits.html>一節。
- 4010 - TRACK_NUMBER_MISMATCH - MKV 影格中的軌跡編號與 MKV 標頭中的軌跡不相符。
- 4011 - FRAMES_MISSING_FOR_TRACK - 片段不包含 MKV 標頭中至少一個軌道的任何影格。
- 4012 - INVALID_FRAGMENT_METADATA - 片段中繼資料名稱不能以字串 開頭 AWS_。
- 4500 - KMS_KEY_ACCESS_DENIED - 拒絕存取串流指定的 KMS 金鑰。
- 4501 - KMS_KEY_DISABLED - 串流指定的 KMS 金鑰已停用。
- 4502 - KMS_KEY_VALIDATION_ERROR - 串流指定的 KMS 金鑰驗證失敗。
- 4503 - KMS_KEY_UNAVAILABLE - 串流指定的 KMS 金鑰無法使用。
- 4504 - KMS_KEY_INVALID_USAGE - 串流指定 KMS 金鑰的使用無效。
- 4505 - KMS_KEY_INVALID_STATE - 串流指定的 KMS 金鑰處於無效狀態。
- 4506 - KMS_KEY_NOT_FOUND - 找不到串流指定的 KMS 金鑰。
- 5000 - INTERNAL_ERROR - 內部服務錯誤。
- 5001 - ARCHIVAL_ERROR - Kinesis Video Streams 無法將片段保留到資料存放區。

 Note

生產者在傳送長時間執行PutMedia請求的承載時，應讀取回應以進行確認。由於中繼代理伺服器上的緩衝，生產者可能會同時收到確認區塊。想要接收及時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 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

HTTP 狀態碼：400

ConnectionLimitExceededException

Kinesis Video Streams 已調節請求，因為您已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

來電者使用錯誤的端點將資料寫入串流。收到這類例外狀況時，使用者必須呼叫 `GetDataEndpoint`，並將 `APIName` 設定為 `PUT_MEDIA`，並使用回應中的端點來叫用下一個 `PutMedia` 呼叫。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權對指定的串流執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

狀態碼：404，具有指定名稱的串流不存在。

HTTP 狀態碼：404

範例

確認格式

確認的格式如下：

```
{
  Acknowledgement : {
    "EventType": enum
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
    "ErrorId" : String
  }
}
```

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Streams 封存媒體

Amazon Kinesis Video Streams Archived Media 支援下列動作：

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

GetClip

服務：Amazon Kinesis Video Streams Archived Media

在指定的時間範圍內，從指定的影片串流下載包含封存隨需媒體的 MP4 檔案 (clip)。

StreamName 和 StreamARN 參數都是選用的，但您必須在調用此 API 操作時指定 StreamName 或 StreamARN。

Note

您必須先呼叫 GetDataEndpoint API 才能取得端點。然後使用 `--endpoint-url` 參數將 GetClip 請求傳送至此端點。

Amazon Kinesis 影片串流對於透過 MP4 提供資料有下列需求：

- [影片播放軌跡需求](#)。
- 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 Amazon CloudWatch GetClip.OutgoingBytes 指標來監控傳出資料的數量。如需使用 CloudWatch 監控 Kinesis Video Streams 的資訊，請參閱 [監控 Kinesis Video Streams](#)。如需定價資訊，請參閱 [Amazon Kinesis Video Streams 定價](#) 和 [AWS 定價](#)。需支付傳出 AWS 資料的費用。

Important

每個片段中包含的轉碼器私有資料 (CPD) 包含轉碼器特定的初始化資訊，例如影格率、解析度和編碼描述檔，這些是正確解碼片段的必要項目。結果剪輯的目標片段之間不支援 CPD 變更。CPD 必須透過查詢的媒體保持一致，否則將會傳回錯誤。

⚠ Important

不支援軌跡變更。追蹤必須在整個查詢媒體中保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則會傳回錯誤。

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

請求剪輯的時間範圍和時間戳記的來源。

類型：[ClipFragmentSelector](#) 物件

必要：是

StreamARN

要擷取媒體剪輯之串流的 Amazon Resource Name (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

要擷取媒體剪輯的串流名稱。

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: 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

請求剪輯中媒體的內容類型。

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 MP4 檔案，其中包含來自指定影片串流的媒體剪輯。輸出將包含來自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 配額](#)。

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 已調節請求，因為您已超過限制。稍後嘗試進行呼叫。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

InvalidCodecPrivateDataException

影片串流中至少一個音軌中的轉碼器私有資料不適用於此操作。

HTTP 狀態碼：400

InvalidMediaFrameException

無法根據指定的編解碼器剖析所請求剪輯中的一個或多個影格。

HTTP 狀態碼：400

MissingCodecPrivateDataException

在影片串流的至少一個軌跡中找不到轉碼器私有資料。

HTTP 狀態碼：400

NoDataRetentionException

GetImages 請求的串流不會保留資料 (也就是 DataRetentionInHours 的 為 0)。

HTTP 狀態碼 : 400

NotAuthorizedException

狀態碼 : 403 , 發起人未獲授權在指定的串流上執行 操作 , 或權杖已過期。

HTTP 狀態碼 : 401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時 , 會擲回此錯誤。

GetHLSStreamingSessionURL 如果LIVE_REPLAY請求的串流具有 ON_DEMAND或 PlaybackMode的工作階段在請求的時間範圍內沒有片段 , 或請求的串流具有 PlaybackMode的工作階段在過去 30 秒內沒有片段 , LIVE則GetDASHStreamingSessionURL擲回此錯誤。

HTTP 狀態碼 : 404

UnsupportedStreamMediaTypeException

媒體的類型 (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第一個片段中軌道的編解碼器 IDs 決定。軌道 1 的轉碼器 ID 應該是 , V_MPEG/ISO/AVC並且選擇性地 , 軌道 2 的轉碼器 ID 應該是 A_AAC。

HTTP 狀態碼 : 400

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊 , 請參閱下列內容 :

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)

- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetDASHStreamingSessionURL

服務：Amazon Kinesis Video Streams Archived Media

擷取串流的 MPEG 動態自適應串流透過 HTTP (DASH) URL。然後，您可以在媒體播放器中開啟 URL 以檢視串流內容。

StreamName 和 StreamARN 參數都是選用的，但您必須在叫用此 API 操作StreamARN時指定 StreamName或。

Amazon Kinesis 影片串流對於透過 MPEG-DASH 提供資料有下列需求：

- [影片播放軌跡需求](#)。
- 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 [--endpoint-url 參數](#)將GetDASHStreamingSessionURL請求傳送至此端點。
2. 使用 擷取 MPEG-DASH URLGetDASHStreamingSessionURL。Kinesis Video Streams 會建立 MPEG-DASH 串流工作階段，用於使用 MPEG-DASH 通訊協定存取串流中的內容。GetDASHStreamingSessionURL會傳回工作階段 MPEG-DASH 資訊清單（使用 MPEG-DASH 串流所需的根資源）的已驗證 URL（包括加密工作階段字符）。

Note

請勿將此字符共用或存放於未經授權的實體可以存取的地方。權杖可讓您存取串流的內容。使用與 AWS 登入資料搭配使用的相同量值來保護權杖。

透過資訊清單提供的媒體僅包含請求的串流、時間範圍和格式。不提供其他媒體資料（例如請求視窗外的影格或替代位元速率）。

3. 將 MPEG-DASH 資訊清單的 URL (包含加密的工作階段字符) 提供給支援 MPEG-DASH 通訊協定的媒體播放器。Kinesis Video Streams 透過資訊清單 URL 提供初始化片段和媒體片段。初始化片段包含串流的轉碼器私有資料，以及設定視訊或音訊解碼器和轉譯器所需的其他資料。媒體片段包含編碼的影片影格或編碼的音訊範例。
4. 媒體播放器通常會接收已驗證的 URL 並請求串流中繼資料和媒體資料。當媒體播放器請求資料時，它會呼叫下列動作：
 - GetDASHManifest：擷取 MPEG DASH 資訊清單，其中包含您要播放之媒體的中繼資料。
 - GetMP4InitFragment：擷取 MP4 初始化片段。媒體播放器通常會先載入初始化片段，再載入任何媒體片段。此片段包含「fyp」和「moov」MP4 原子，以及初始化媒體播放器解碼器所需的子原子。

初始化片段不會對應至 Kinesis 影片串流中的片段。它只包含串流和個別軌跡的轉碼器私有資料，媒體播放器需要這些資料來解碼媒體影格。

- GetMP4MediaFragment：擷取 MP4 媒體片段。這些片段包含「moof」和「mdat」MP4 原子及其子原子，其中包含編碼片段的媒體影格及其時間戳記。

Important

每個片段中包含的轉碼器私有資料 (CPD) 包含轉碼器特定的初始化資訊，例如影格率、解析度和編碼描述檔，這些是正確解碼片段的必要項目。在串流工作階段期間不支援 CPD 變更。CPD 必須透過查詢的媒體保持一致。

Important

不支援軌跡變更。追蹤必須在整個查詢媒體中保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則串流將會失敗。

使用此動作擷取的資料是計費的。如需詳細資訊，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 配額](#)。

您可以透過監控 `GetMP4MediaFragment.OutgoingBytesAmazon CloudWatch` 指標來監控媒體播放器使用的資料量。如需有關使用 CloudWatch 監控 Kinesis Video Streams 的資訊，請參閱[監控 Kinesis Video Streams](#)。如需定價資訊，請參閱 [Amazon Kinesis Video Streams 定價](#)和[AWS 定價](#)。同時收取 HLS 工作階段和傳出 AWS 資料的費用。

如需 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 Video Streams 封存媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 – 如果您想要向支援團隊報告問題 AWS，如果指定請求 ID，則更能診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進行程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。

如需詳細資訊，請參閱本主題底部的錯誤區段，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 PlaybackMode 是 ON_DEMAND 或 LIVE_REPLAY，則需要此參數 LIVE_REPLAY。如果 PlaybackMode 為 LIVE，則此參數為選用 LIVE。如果 PlaybackMode 是 LIVE，FragmentSelectorType 則可以設定，但 TimestampRange 不應設定。如果 PlaybackMode 是 ON_DEMAND 或 LIVE_REPLAY，TimestampRange 則必須同時設定 FragmentSelectorType 和 TimestampRange。

類型：[DASHFragmentSelector](#) 物件

必要：否

DisplayFragmentNumber

根據片段在工作階段中的序號，在資訊清單檔案中識別片段。如果 DisplayFragmentNumber 設定為 ALWAYS，Kinesis Video Streams 片段編號會新增至清單檔案中具有屬性名稱 "kvs:fn" 的每個 S 元素。這些片段編號可用於記錄或與其他 APIs (例如 GetMedia 和 GetMediaForFragmentList) 搭配使用。自訂 MPEG-DASH 媒體播放器是利用這些自訂屬性的必要項目。

預設值為 NEVER。

類型：字串

有效值: ALWAYS | NEVER

必要：否

DisplayFragmentTimestamp

根據 MPEG-DASH 規格，可以使用資訊清單本身中的屬性衍生資訊清單檔案中片段的時鐘時間。不過，MPEG-DASH 相容媒體播放器通常無法正確處理媒體時間軸中的差距。Kinesis Video

Streams 會調整資訊清單檔案中的媒體時間軸，以啟用具有不連續性的媒體播放。因此，從資訊清單檔案衍生的時鐘時間可能不準確。如果 `DisplayFragmentTimestamp` 設定為 `ALWAYS`，則準確的片段時間戳記會新增至資訊清單檔案中具有屬性名稱 “kvs : ts” 的每個 S 元素。自訂 MPEG-DASH 媒體播放器是利用此自訂屬性的必要項目。

預設值為 `NEVER`。當 [DASHFragmentSelector](#) 為 `SERVER_TIMESTAMP`，時間戳記將是伺服器啟動時間戳記。同樣地，當 [DASHFragmentSelector](#) 為 `PRODUCER_TIMESTAMP`，時間戳記將是生產者起始時間戳記。

類型：字串

有效值：`ALWAYS` | `NEVER`

必要：否

[Expires](#)

請求的工作階段過期前的秒數。此值可以介於 300 (5 分鐘) 和 43200 (12 小時) 之間。

當工作階段過期時，就 `GetMP4MediaFragment` 無法對該工作階段進行新的 `GetDashManifest`、`GetMP4InitFragment` 或 呼叫。

預設值為 300 (5 分鐘)。

類型：整數

有效範圍：最小值為 300。最大值為 43200。

必要：否

[MaxManifestFragmentResults](#)

MPEG-DASH 資訊清單中傳回的片段數量上限。

當 `PlaybackMode` 為 `LIVE`，最新的片段會傳回至此值。當 `PlaybackMode` 為 `ON_DEMAND`，會傳回最舊的片段，最多到此最大數量。

當即時 MPEG-DASH 資訊清單中可用的片段數量較高時，影片播放器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，但可減少播放期間重新緩衝的可能性。我們建議即時 MPEG-DASH 資訊清單至少有 3 個片段，最多有 10 個片段。

如果 `PlaybackMode` 是 `LIVE` 或 `LIVE_REPLAY`，則預設為 5 個片段，如果 `PlaybackMode` 是 `ON_DEMAND`，則預設為 1,000 個片段。

1,000 個片段的最大值對應於具有 1 秒片段之串流上的影片超過 16 分鐘，以及具有 10 秒片段之串流上的影片超過 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需資料。

三種工作階段類型的功能包括：

- **LIVE**：對於此類型的工作階段，MPEG-DASH 資訊清單會持續更新為可用的最新片段。我們建議媒體播放器每隔一秒擷取新的資訊清單。在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示「即時」通知，而沒有用於選擇要顯示之播放視窗中的位置的清除器控制。

Note

在 LIVE 模式中，最新的可用片段會包含在 MPEG-DASH 資訊清單中，即使片段之間有間隙（亦即，如果片段遺失）。這種差距可能會導致媒體播放器停止或導致播放跳躍。在此模式中，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 MPEG-DASH 資訊清單。如果遺失的片段在後續片段新增至資訊清單後變成可用，則不會新增較舊的片段，也不會填充間隙。

- **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，只是從包含指定開始時間的片段開始。片段會新增為下一個片段經過的持續時間，而不是在擷取片段時新增的片段。例如，如果工作階段中的片段長度為兩秒，則會每兩秒將新的片段新增至資訊清單。此模式有助於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。此模式也適用於串流先前封存的媒體，而不會受限於 ON_DEMAND 模式中的 1,000 個片段限制。
- **ON_DEMAND**：對於此類型的工作階段，MPEG-DASH 資訊清單包含工作階段的所有片段，最高可達中指定的數字 `MaxManifestFragmentResults`。每個工作階段只能擷取一次資訊清單。在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示清除器控制項，以選擇要顯示之播放視窗中的位置。

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，則具有較大片段編號（即較新的片段）的片段會包含在 MPEG-

DASH 資訊清單中。不包含其他片段。具有不同時間戳記但具有重疊持續時間的片段仍會包含在 MPEG-DASH 資訊清單中。這可能會導致媒體播放器發生意外行為。

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 之串流的 Amazon Resource Name (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 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
```

```
{  
  "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](#)

媒體播放器可用來擷取 MPEG-DASH 資訊清單的 URL（包含工作階段字符）。

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 已調節請求，因為您已超過限制。稍後嘗試撥打電話。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

InvalidCodecPrivateDataException

影片串流中至少一個音軌中的轉碼器私有資料不適用於此操作。

HTTP 狀態碼：400

MissingCodecPrivateDataException

在影片串流的至少一個軌跡中找不到轉碼器私有資料。

HTTP 狀態碼：400

NoDataRetentionException

GetImages 請求的串流不會保留資料 (也就是 DataRetentionInHours 的 為 0)。

HTTP 狀態碼 : 400

NotAuthorizedException

狀態碼 : 403 , 發起人未獲授權在指定的串流上執行 操作 , 或權杖已過期。

HTTP 狀態碼 : 401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時 , 會擲回此錯誤。

GetHLSStreamingSessionURL 如果LIVE_REPLAY請求的串流具有 ON_DEMAND或 PlaybackMode的工作階段在請求的時間範圍內沒有片段 , 或請求的串流具有 PlaybackMode的工作階段在過去 30 秒內沒有片段 , LIVE則GetDASHStreamingSessionURL擲回此錯誤。

HTTP 狀態碼 : 404

UnsupportedStreamMediaTypeException

媒體的類型 (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第一個片段中軌道的編解碼器 IDs 決定。軌道 1 的轉碼器 ID 應該是 , V_MPEG/ISO/AVC並且選擇性地 , 軌道 2 的轉碼器 ID 應該是 A_AAC。

HTTP 狀態碼 : 400

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊 , 請參閱下列內容 :

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)

- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetHLSStreamingSessionURL

服務：Amazon Kinesis Video Streams Archived Media

擷取串流的 HTTP 即時串流 (HLS) URL。然後，您可以在瀏覽器或媒體播放器中開啟 URL，以檢視串流內容。

StreamName 和 StreamARN 參數都是選用的，但您必須在叫用此 API 操作StreamARN時指定 StreamName或。

Amazon Kinesis 影片串流對於透過 HLS 提供資料有下列需求：

- [影片播放軌跡需求](#)。
- 資料保留必須大於 0。
- 每個片段的視訊軌必須包含 H.264 格式的編解碼器私有資料 (AVC) 或 H.265 格式的 HEVC ([MPEG-4 規格 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](#)。
- 每個片段的音軌（如果有）必須包含 AAC 格式的轉碼器私有資料 ([AAC 規格 ISO/IEC 13818-7](#))。

Kinesis Video Streams HLS 工作階段包含分段 MPEG-4 格式（也稱為 fMP4 或 CMAF）或 MPEG-2 格式（也稱為 HLS 規格也支援的 TS 區塊）的片段。如需 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 [--endpoint-url 參數](#)將GetHLSStreamingSessionURL請求傳送至此端點。
2. 使用 擷取 HLS URLGetHLSStreamingSessionURL。Kinesis Video Streams 會建立 HLS 串流工作階段，用於使用 HLS 通訊協定存取串流中的內容。會針對工作階段的 HLS 主播放清單（使用 HLS 串流所需的根資源）GetHLSStreamingSessionURL傳回已驗證的 URL（包括加密的工作階段字符）。

Note

請勿將此字符共用或存放於未經授權的實體可以存取它的位置。權杖可讓您存取串流的內容。使用與 AWS 登入資料相同的措施來保護字符。

透過播放清單提供的媒體僅包含請求的串流、時間範圍和格式。不提供其他媒體資料（例如請求視窗外的影格或替代位元速率）。

3. 將 HLS 主播放清單的 URL（包含加密的工作階段字符）提供給支援 HLS 通訊協定的媒體播放器。Kinesis Video Streams 透過主播放清單 URL 提供 HLS 媒體播放清單、初始化片段和媒體片段。初始化片段包含串流的轉碼器私有資料，以及設定視訊或音訊解碼器和轉譯器所需的其他資料。媒體片段包含 H.264 編碼的影片影格或 AAC 編碼的音訊範例。
4. 媒體播放器通常會接收已驗證的 URL 並請求串流中繼資料和媒體資料。當媒體播放器請求資料時，它會呼叫下列動作：
 - GetHLSMasterPlaylist：擷取 HLS 主播放清單，其中包含每個軌跡 GetHLSMediaPlaylist 的動作 URL，以及媒體播放器的其他中繼資料，包括估計位元速率和解析度。
 - GetHLSMediaPlaylist：擷取 HLS 媒體播放清單，其中包含使用 GetMP4InitFragment 動作存取 MP4 初始化片段的 URL，以及使用 GetMP4MediaFragment 動作存取 MP4 媒體片段的 URLs。HLS 媒體播放清單也包含播放器播放所需的串流中繼資料，例如 PlaybackMode 是 LIVE 還是 ON_DEMAND。對於具有的工作階段，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為具有 PlaybackType 之工作階段的新片段 LIVE。視訊音軌和音訊音軌（如果適用）有不同的 HLS 媒體播放清單，其中包含特定音軌的 MP4 媒體 URLs。
 - GetMP4InitFragment：擷取 MP4 初始化片段。媒體播放器通常會先載入初始化片段，再載入任何媒體片段。此片段包含「fytp」和「moov」MP4 原子，以及初始化媒體播放器解碼器所需的子原子。

初始化片段不會對應至 Kinesis 影片串流中的片段。它只包含串流和個別軌跡的轉碼器私有資料，媒體播放器需要這些資料來解碼媒體影格。

- GetMP4MediaFragment：擷取 MP4 媒體片段。這些片段包含「moof」和「mdat」MP4 原子及其子原子，其中包含編碼片段的媒體影格及其時間戳記。

Note

每個片段中包含的轉碼器私有資料 (CPD) 包含轉碼器特定的初始化資訊，例如影格率、解析度和編碼描述檔，這些是正確解碼片段的必要資訊。對於 TS 和 MP4，串流工作階段期間支援 CPD 變更。因此，工作階段中的片段在 CPD 中可以有不同的資訊，而不會中斷播放。對於每個串流工作階段，僅允許 500 個 CPD 變更。

⚠ Important

不支援軌跡變更。追蹤必須在整個查詢媒體中保持一致。如果串流中的片段從只有視訊變更為同時擁有音訊和視訊，或者 AAC 音軌變更為 A-Law 音軌，則串流將會失敗。

使用此動作擷取的資料是計費的。如需相關資訊，請參閱 [定價](#)。

- GetTSFragment：擷取 MPEG TS 片段，其中包含串流中所有軌道的初始化和媒體資料。

ℹ Note

如果 ContainerFormat 是 MPEG_TS，則會使用此 API 而非 GetMP4InitFragment 和 GetMP4MediaFragment 來擷取串流媒體。

使用此動作擷取的資料是計費的。如需詳細資訊，請參閱 [Kinesis Video Streams 定價](#)。

不得在玩家之間共用串流工作階段 URL。如果多個媒體播放器共用工作階段，服務可能會調節工作階段。如需連線限制，請參閱 [Kinesis Video Streams 配額](#)。

您可以透過監控 GetMP4MediaFragment.OutgoingBytesAmazon CloudWatch 指標來監控媒體播放器使用的資料量。如需使用 CloudWatch 監控 Kinesis Video Streams 的資訊，請參閱 [監控 Kinesis Video Streams](#)。如需定價資訊，請參閱 [Amazon Kinesis Video Streams 定價](#) 和 [AWS 定價](#)。同時收取 HLS 工作階段和傳出 AWS 資料的費用。

請參閱 文件指南中的影片播放範例：[使用 AWS CLI 擷取 HLS 串流工作階段 URL](#) 和 [範例：在 HTML 和 JavaScript 中使用 HLS](#)。

如需 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 Video Streams 封存媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 – 如果您想要向 報告問題 AWS，支援團隊可以在指定請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進行程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。如需詳細資訊，請參閱本主題底部的錯誤區段，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

指定應用於封裝媒體的格式。指定FRAGMENTED_MP4容器格式會將媒體封裝為 MP4 片段 (fMP4 或 CMAF)。這是建議的封裝，因為封裝負荷最少。另一個容器格式選項是 MPEG_TS。HLS 自推出以來就支援 MPEG TS 區塊，有時是較舊的 HLS 播放器上唯一支援的封裝。MPEG TS 通常具有 5-25% 的封裝額外負荷。這表示 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

指定何時標記標記片段之間的不連續性新增到媒體播放清單。

媒體播放器通常會根據每個片段的時間戳記，建置要播放的媒體內容時間軸。這表示如果片段之間有任何重疊或間隙（如果 [HLSFragmentSelector](#) 設定為 `SERVER_TIMESTAMP`），媒體播放器時間軸在某些位置也會有片段之間的小間隙，在其他位置會覆寫影格。媒體播放器時間軸中的間隙可能會導致播放停滯，重疊可能會導致播放抖動。當片段之間存在不連續旗標時，媒體播放器預期會重設時間軸，導致下一個片段在前一個片段之後立即播放。

支援下列模式：

- ALWAYS：不連續標記會放置在 HLS 媒體播放清單中的每個片段之間。ALWAYS 如果片段時間戳記不正確，建議使用 `NEVER` 值。
- NEVER：不會將不連續標記放在任何位置。建議使用 `NEVER` 值，以確保媒體播放器時間軸最準確地對應到生產者時間戳記。
- ON_DISCONTINUITY：不連續標記會放置在間隙或重疊超過 50 毫秒的片段之間。對於大多數播放案例，建議使用 `ON_DISCONTINUITY` 值，以便媒體播放器時間軸只有在媒體時間軸（例如缺少片段）發生重大問題時才會重設。

預設值為ALWAYS。[HLSFragmentSelector](#)當 設定為 `SERVER_TIMESTAMP`，以及當 設定為 `NEVER`時`PRODUCER_TIMESTAMP`。

類型：字串

有效值:ALWAYS | NEVER | ON_DISCONTINUITY

必要：否

[DisplayFragmentTimestamp](#)

指定何時應將片段開始時間戳記包含在 HLS 媒體播放清單中。一般而言，媒體播放器會將播放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。不過，當 HLS 媒體播放清單中包含啟動時間戳記時，有些媒體播放器可能會根據片段時間戳記，將目前的播放頭報告為絕對時間。這有助於建立播放體驗，向觀眾顯示媒體的時鐘時間。

預設值為 NEVER。當 [HLSFragmentSelector](#) 為 時 SERVER_TIMESTAMP，時間戳記將是伺服器啟動時間戳記。同樣地，當 [HLSFragmentSelector](#) 為 時 PRODUCER_TIMESTAMP，時間戳記將是生產者起始時間戳記。

類型：字串

有效值: ALWAYS | NEVER

必要：否

[Expires](#)

請求的工作階段過期前的秒數。此值可以介於 300 (5 分鐘) 和 43200 (12 小時) 之間。

當工作階段過期時，就 GetTSFragment 無法對該工作階段進行新的 GetHLSMasterPlaylist、GetMP4MediaFragment、GetHLSMediaPlaylist 或 GetMP4InitFragment 或 呼叫。

預設值為 300 (5 分鐘)。

類型：整數

有效範圍：最小值為 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 PlaybackMode 是 ON_DEMAND 或 LIVE_REPLAY，則需要此參數 LIVE_REPLAY。如果 PlaybackMode 為 LIVE，則此參數為選用 LIVE。如果 PlaybackMode 是 LIVE，FragmentSelectorType 則可以設定，但 TimestampRange 不應設定。如果 PlaybackMode 是 ON_DEMAND 或 LIVE_REPLAY，TimestampRange 則必須同時設定 FragmentSelectorType 和 TimestampRange。

類型：[HLSFragmentSelector](#) 物件

必要：否

[MaxMediaPlaylistFragmentResults](#)

HLS 媒體播放清單中傳回的片段數量上限。

當 PlaybackMode 為 時LIVE，最新的片段會傳回至此值。當 PlaybackMode 為 時ON_DEMAND，會傳回最舊的片段，最多到此最大數量。

當即時 HLS 媒體播放清單中有更多可用的片段時，影片播放器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，但可減少播放期間重新緩衝的可能性。我們建議即時 HLS 媒體播放清單至少有 3 個片段，最多有 10 個片段。

如果 PlaybackMode 是 LIVE 或 ，則預設為 5 個片段LIVE_REPLAY，如果 PlaybackMode 是 ，則預設為 1,000 個片段ON_DEMAND。

5,000 個片段的最大值對應於具有 1 秒片段之串流上的影片超過 80 分鐘，以及具有 10 秒片段之串流上的影片超過 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需資料。

三種工作階段類型的功能包括：

- **LIVE**：對於此類型的工作階段，HLS 媒體播放清單會持續更新為可用的最新片段。我們建議媒體播放器每隔一秒擷取新的播放清單。在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示「即時」通知，而沒有用於選擇要顯示之播放視窗中的位置的清除器控制。

Note

在 LIVE 模式中，最新的可用片段會包含在 HLS 媒體播放清單中，即使片段之間存在間隙（亦即，如果片段遺失）。這種差距可能會導致媒體播放器停止或導致播放跳躍。在此模式中，如果片段比播放清單中的最新片段舊，則不會將其新增至 HLS 媒體播放清單。如果遺失的片段在後續片段新增至播放清單後變成可用，則不會新增較舊的片段，也不會填補間隙。

- **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模式的更新方式類似，只是從包含指定開始時間的片段開始。片段會新增為下一個片段經過的持續時間，而不是在擷取片段時新增的片段。例如，如果工作階段中的片段長度為兩秒，則會每兩秒將新的片段新增至媒體播放清單。此模式有助於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。此模式也適用於串流先前封存的媒體，而不會受限於 ON_DEMAND 模式中的 1,000 個片段限制。
- **ON_DEMAND**：對於此類型的工作階段，HLS 媒體播放清單包含工作階段的所有片段，最多到中指定的數字MaxMediaPlaylistFragmentResults。每個工作階段只能擷取播放清單一次。在媒體播放器中播放此類型的工作階段時，使用者介面通常會顯示清除器控制項，以選擇要顯示之播放視窗中的位置。

在所有播放模式中，如果 FragmentSelectorType 為 PRODUCER_TIMESTAMP，且如果有多個片段具有相同的開始時間戳記，則具有最大片段編號（即最新片段）的片段會包含在 HLS 媒體播放清單中。不包含其他片段。具有不同時間戳記但持續時間重疊的片段仍會包含在 HLS 媒體播放清單中。這可能會導致媒體播放器發生意外行為。

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 之串流的 Amazon Resource Name (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](#)

媒體播放器可用來擷取 HLS 主播放清單的 URL（包含工作階段字符）。

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 已調節請求，因為您已超過限制。稍後嘗試進行呼叫。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

InvalidCodecPrivateDataException

影片串流中至少一個音軌中的轉碼器私有資料不適用於此操作。

HTTP 狀態碼：400

MissingCodecPrivateDataException

在影片串流的至少一個軌跡中找不到轉碼器私有資料。

HTTP 狀態碼：400

NoDataRetentionException

GetImages 請求的串流不會保留資料（也就是 DataRetentionInHours 的為 0）。

HTTP 狀態碼：400

NotAuthorizedException

狀態碼：403，發起人未獲授權在指定的串流上執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時，會擲回此錯誤。

GetHLSStreamingSessionURL 如果 LIVE_REPLAY 請求的串流具有 ON_DEMAND 或 PlaybackMode 的工作階段在請求的時間範圍內沒有片段，或請求的串流具有 PlaybackMode 的工作階段在過去 30 秒內沒有片段，LIVE 則 GetDASHStreamingSessionURL 擲回此錯誤。

HTTP 狀態碼：404

UnsupportedStreamMediaTypeException

媒體的類型（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第一個片段中軌道的編解碼器 IDs 決定。軌道 1 的轉碼器 ID 應該是 V_MPEG/ISO/AVC 並且選擇性地，軌道 2 的轉碼器 ID 應該是 A_AAC。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetImages

服務：Amazon Kinesis Video Streams Archived Media

影像的受管支援提供一種完全受管的方式，從 Kinesis Video Streams 中串流和存放的影片資料取得影像。您可以使用映像來執行機器學習 (ML) 工作負載，例如人員、寵物或車輛偵測。影像也可以用來新增互動式元素到播放，例如動作事件的影像預覽和影片剪輯的清除。

GetImages 也會擷取與指定時間範圍、取樣間隔和影像格式組態的每個時間戳記對應的影像清單。

Note

您必須先呼叫 GetDataEndpoint API 才能取得端點。然後使用 [--endpoint-url](#) 參數將 GetImages 請求傳送至此端點。

[影片播放軌跡需求。](#)

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

要產生之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 StartTimeStamp，您將收到 `IllegalArgumentException`。

類型：Timestamp

必要：是

Format

用來編碼影像的格式。

類型：字串

有效值: JPEG | PNG

必要：是

FormatConfig

金鑰值對結構的清單，其中包含可在產生映像時套用的額外參數。FormatConfig 金鑰是 `JPEGQuality`，表示要用來產生映像的 JPEG 品質金鑰。此 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的壓縮產生影像。如果值為 100，則會產生具有最佳品質且壓縮較少的影像。如果未提供值，`JPEGQuality` 金鑰的預設值會設為 80。

類型：字串到字串映射

映射項目：最多 1 個項目。

有效金鑰：`JPEGQuality`

值長度限制：長度下限為 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映像高度。同時提供 `HeightPixels` 和 `WidthPixels` 參數時，影像會延伸以符合指定的長寬比。如果只提供 `HeightPixels` 參數，則會使用其原始長寬比來計算 `WidthPixels` 比率。如果未提供參數，則會傳回原始影像大小。

類型：整數

有效範圍：最小值為 1。最大值為 2160。

必要：否

ImageSelectorType

用於產生映像的伺服器或生產者時間戳記來源。

類型：字串

有效值:PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

必要：是

MaxResults

API 要傳回的映像數量上限。

Note

預設限制為每個 API 回應 25 個映像。提供MaxResults大於此值會導致頁面大小為 25。任何其他結果都會進行分頁。

類型：Long

有效範圍：最小值為 1。最大值為 100。

必要：否

NextToken

指定在何處開始分頁下一組影像的權杖。這是先前截斷的回應GetImages:NextToken中的。

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，以毫秒 (ms) 為單位。可以提供的最小值為 200 毫秒（每秒 5 張影像）。如果時間戳記範圍小於取樣間隔，startTimestamp則會在可用時傳回來自的映像。

類型：整數

必要：否

StartTimestamp

應產生影像的起點。這StartTimestamp必須在時間戳記的包含範圍內，才能傳回影像。

類型：Timestamp

必要：是

StreamARN

要從中擷取映像之串流的 Amazon Resource Name (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 參數搭配使用的輸出映像寬度。同時提供 WidthPixels和 HeightPixels 參數時，影像會延伸以符合指定的長寬比。如果只提供 WidthPixels 參數，或只HeightPixels提供，ValidationException則會擲出。如果未提供參數，則會傳回串流的原始影像大小。

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

從影片串流產生的影像清單。如果指定的時間戳記沒有可用的媒體，則會在輸出中列出 NO_MEDIA 錯誤。如果在產生映像時發生錯誤，輸出中 MEDIA_ERROR 會列出 作為遺失映像的原因。

類型：[Image](#) 物件陣列

NextToken

請求中用於取得更多映像的加密字符。

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 已調節請求，因為您已超過限制。稍後嘗試撥打電話。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

NoDataRetentionException

GetImages 請求的串流不會保留資料（也就是 DataRetentionInHours 的為 0）。

HTTP 狀態碼：400

NotAuthorizedException

狀態碼：403，發起人未獲授權在指定的串流上執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時，會擲回此錯誤。

GetHLSStreamingSessionURL 如果 LIVE_REPLAY 請求的串流具有 ON_DEMAND 或 PlaybackMode 的工作階段在請求的時間範圍內沒有片段，或請求的串流具有 PlaybackMode 的工作階段在過去 30 秒內沒有片段，LIVE 則 GetDASHStreamingSessionURL 擲回此錯誤。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

GetMediaForFragmentList

服務：Amazon Kinesis Video Streams Archived Media

從 Amazon Kinesis 影片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 才能取得端點。然後使用 `--endpoint-url` 參數將 `GetMediaForFragmentList` 請求傳送至此端點。

如需限制，請參閱 [Kinesis Video Streams 配額](#)。

Important

如果在叫用 Kinesis Video Streams 封存媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 – 如果您想要向 報告問題 AWS，支援團隊可以在指定請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。如需詳細資訊，請參閱本主題底部的錯誤區段，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量清單。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之串流的 Amazon Resource Name (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串流名稱。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

請求媒體的內容類型。

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 傳回的承載是來自指定串流的一系列區塊。如需區塊的相關資訊，請參閱 [PutMedia](#)。Kinesis Video Streams 在 `GetMediaForFragmentList` 呼叫中傳回的區塊也包含下列額外的 Matroska (MKV) 標籤：

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - 區塊中傳回的片段編號。
- `AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP` - 片段的伺服器端時間戳記。
- `AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP` - 片段的生產者端時間戳記。

如果發生例外狀況，將包含下列標籤：

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - 擲出例外狀況的片段數量。
- `AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE` - 錯誤的整數代碼。
- `AWS_KINESISVIDEO_EXCEPTION_MESSAGE` - 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 已調節請求，因為您已超過限制。稍後嘗試進行呼叫。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

NotAuthorizedException

狀態碼：403，發起人未獲授權在指定的串流上執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時，會擲回此錯誤。

GetHLSStreamingSessionURL 如果LIVE_REPLAY請求的串流具有 ON_DEMAND或 PlaybackMode的工作階段在請求的時間範圍內沒有片段，或請求的串流具有 PlaybackMode的工作階段在過去 30 秒內沒有片段，LIVE則GetDASHStreamingSessionURL擲回此錯誤。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)

- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListFragments

服務：Amazon Kinesis Video Streams Archived Media

從封存資料中指定的串流和時間戳記範圍傳回[Fragment](#)物件清單。

列出片段最終是一致的。這表示即使生產者收到片段保留的確認，結果可能不會立即從對的請求傳回ListFragments。不過，結果通常可在一秒內取得。

Note

您必須先呼叫 GetDataEndpoint API 才能取得端點。然後使用 `--endpoint-url` 參數將ListFragments請求傳送至此端點。

Important

如果在調用 Kinesis Video Streams 封存媒體 API 後擲出錯誤，除了 HTTP 狀態碼和回應內文之外，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 – 如果您想要向回報問題 AWS，支援團隊可以在指定請求 ID 時更好地診斷問題。

HTTP 狀態碼和 `ErrorType` 標頭都可以用來對錯誤是否可以重試以及在何種條件下進程式設計決策，並提供用戶端程式設計人員可能需要採取的動作的相關資訊，以便成功重試。如需詳細資訊，請參閱本主題底部的錯誤區段，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 API 中未傳遞時，才需要這樣做。

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超過中指定的值max-results，則會在輸出中提供[ListFragments : NextToken](#)，供您用來恢復分頁。

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。這是先前截斷的回應中的 [ListFragments : NextToken](#)。

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之串流的 Amazon Resource Name (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串流名稱。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

串流中符合選取器條件的封存 [Fragment](#) 物件清單。結果沒有特定順序，即使是跨頁面也一樣。

如果串流中沒有符合選取器條件的片段，則會傳回空清單。

類型：[Fragment](#) 物件陣列

NextToken

如果已截斷傳回的清單，操作會傳回此字符，以用於擷取下一頁的結果。此值 null 表示沒有更多結果可傳回。

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 已調節請求，因為您已超過限制。稍後嘗試撥打電話。如需限制的相關資訊，請參閱 [Kinesis Video Streams 配額](#)。

HTTP 狀態碼：400

InvalidArgumentException

指定的參數超過其限制、不支援或無法使用。

HTTP 狀態碼：400

NotAuthorizedException

狀態碼：403，發起人未獲授權在指定的串流上執行操作，或權杖已過期。

HTTP 狀態碼：401

ResourceNotFoundException

GetImages 當 Kinesis Video Streams 找不到您指定的串流時，會擲回此錯誤。

GetHLSStreamingSessionURL 如果LIVE_REPLAY請求的串流具有 ON_DEMAND或 PlaybackMode的工作階段在請求的時間範圍內沒有片段，或請求的串流具有 PlaybackMode的工作階段在過去 30 秒內沒有片段，LIVE則GetDASHStreamingSessionURL擲回此錯誤。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)

- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Signaling 頻道

Amazon Kinesis Video Signaling Channels 支援下列動作：

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

GetIceServerConfig

服務：Amazon Kinesis Video Signaling Channels

注意：使用此 API 之前，您必須呼叫 `GetSignalingChannelEndpoint` API 來請求 HTTPS 端點。然後，您可以在 `GetIceServerConfig` API 請求中指定端點和區域。

取得互動式連線建立 (ICE) 伺服器組態資訊，包括 URI URIs、使用者名稱和密碼，可用於設定 WebRTC 連線。ICE 元件使用此組態資訊來設定 WebRTC 連線，包括使用圍繞 NAT 的轉送 (TURN) 轉送伺服器進行周遊驗證。

TURN 是一種通訊協定，用於改善 peer-to-peer 應用程式的連線能力。透過提供雲端型轉送服務，TURN 可確保即使一或多個對等連線無法直接 peer-to-peer 連線也能建立連線。如需詳細資訊，請參閱 [REST API for Access To TURN Services](#)。

您可以叫用此 API 來建立備用機制，以防任一對等無法透過訊號管道建立直接 peer-to-peer 連線。您必須指定訊號頻道的 Amazon Resource Name (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

訊號通道的 ARN，用於已設定 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

檢視器的唯一識別符。在訊號頻道中必須是唯一的。

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 是唯一的有效值。

類型：字串

有效值:TURN

必要：否

Username

要與登入資料建立關聯的選用使用者 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 伺服器資訊物件的清單。

類型：[IceServer](#) 物件陣列

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

您的請求已受到調節，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

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

發起人無權執行此操作。

HTTP 狀態碼：401

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。用戶端應重新連線至頻道，以繼續傳送/接收訊息。

HTTP 狀態碼：400

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

SendAlexaOfferToMaster

服務：Amazon Kinesis Video Signaling Channels

Note

使用此 API 之前，您必須呼叫 `GetSignalingChannelEndpoint` API 以取得端點。然後，您可以在 `SendAlexaOfferToMaster` API 請求中指定端點和區域。

此 API 可讓您將已啟用 WebRTC 的裝置與 Alexa 顯示裝置連線。調用時，它會將 Alexa 工作階段描述協定 (SDP) 優惠傳送給主對等。一旦主伺服器連接到指定的訊號頻道，就會立即交付優惠。此 API 會從連線的主伺服器傳回 SDP 答案。如果主伺服器未連線至訊號頻道，則會在訊息過期之前提出重新傳遞請求。

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 和主要對等通訊的訊號管道的 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：是

MessagePayload

base64 編碼的 SDP 提供內容。

類型：字串

長度限制：長度下限為 1。長度上限為 10000。

模式：`[a-zA-Z0-9+/=]+`

必要：是

SenderClientId

寄件者用戶端的唯一識別符。

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

base64 編碼的 SDP 回答內容。

類型：字串

長度限制：長度下限為 1。長度上限為 10000。

錯誤

如需所有動作常見錯誤的資訊，請參閱[常見錯誤](#)。

ClientLimitExceededException

您的請求已受到調節，因為您已超過允許的用戶端呼叫限制。稍後嘗試進行呼叫。

HTTP 狀態碼：400

InvalidArgumentException

此輸入參數的值無效。

HTTP 狀態碼：400

NotAuthorizedException

發起人無權執行此操作。

HTTP 狀態碼：401

ResourceNotFoundException

找不到指定的資源。

HTTP 狀態碼：404

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video WebRTC 儲存體

Amazon Kinesis Video WebRTC Storage 支援下列動作：

- [JoinStorageSession](#)
- [JoinStorageSessionAsViewer](#)

JoinStorageSession

服務：Amazon Kinesis Video WebRTC Storage

Note

使用此 API 之前，您必須呼叫 `GetSignalingChannelEndpoint` API 來請求 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訊和/或多向音訊 WebRTC 工作階段，做為輸入頻道的視訊產生裝置。如果頻道沒有現有的工作階段，請建立新的串流工作階段，並提供訊號頻道的 Amazon Resource Name (ARN)。

對於 `SINGLE_MASTER` 類型，產生視訊的裝置目前能夠同時將音訊和視訊媒體擷取到串流中。只有產生影片的裝置可以加入工作階段並記錄媒體。

Important

WebRTC 擷取目前需要音訊和視訊音軌。

目前需求：

- 視訊軌：H.264
- 音軌：Ops

Kinesis 影片串流中產生的擷取影片將具有下列參數：H.264 影片和 AAC 音訊。

一旦主要參與者透過 WebRTC 交涉連線，擷取的媒體工作階段將存放在 Kinesis 影片串流中。然後，多個檢視器可以透過我們的播放 APIs 播放即時媒體。

您也可以搭配擷取的 WebRTC 媒體，使用現有的 Kinesis Video Streams 功能，例如 HLS 或 DASH 播放、透過 [GetImages](#) 產生影像等等。

Note

目前不支援 S3 映像交付和通知。

Note

假設只有一個產生視訊的裝置用戶端可以與頻道的工作階段建立關聯。如果多個用戶端將特定頻道的工作階段聯結為影片製作裝置，則以最新的用戶端請求為優先。

其他資訊

- 等冪 - 此 API 不是等冪。
- 重試行為 - 這計為新的 API 呼叫。
- 並行呼叫 - 允許並行呼叫。每次呼叫會傳送一次提議。

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)。

⚠ Important

請注意此輸入參數的大寫。

類型：字串

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

您沒有執行此操作所需的許可。

HTTP 狀態碼：403

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

JoinStorageSessionAsViewer

服務：Amazon Kinesis Video WebRTC Storage

Note

具有多檢視器支援（預覽版）的 WebRTC 擷取正在預覽版中提供，如 AWS 服務條款所定義，可能會有所變更。它目前僅適用於 us-east-1 (IAD)。

若要參加預覽版，請傳送電子郵件至 kvs-webrtc-multi-view-preview@amazon.com。

Note

使用此 API 之前，您必須呼叫 GetSignalingChannelEndpoint API 來請求 WEBRTC 端點。然後，您可以在 JoinStorageSessionAsViewer API 請求中指定端點和區域。

JoinStorageSessionAsViewer 可讓檢視器加入進行中的雲端錄製 WebRTC 串流工作階段。此 API 會透過在檢視器和錄製代理程式之間傳送 SDP 優惠和 ICE 候選項目來啟動 WebRTC 連線，讓檢視器透過錄製代理程式接收來自主伺服器的即時視訊，並透過錄製代理程式參與雙向音訊通訊。連線後，如果提供，檢視器的音訊會轉送至所有其他連線的對等，包括主要參與者，並併入要儲存到 Kinesis Video 串流的 WebRTC 串流。

Important

觀眾目前無法傳送任何視訊軌。觀眾可以傳送選用的音軌，或完全不傳送任何音軌。目前的瀏覽者參與者傳送要求：

- 視訊軌：不支援
- 音軌（選用）：Ops

如果主要參與者目前連接到視訊產生裝置，則 Kinesis 視訊串流中產生的擷取視訊將具有下列參數：H.264 視訊和 AAC 音訊。

Note

檢視器參與者直接連接到儲存工作階段，而不是直接連接到主要參與者。儲存工作階段負責混合、複製和路由媒體到適當的目的地。

Note

如果主要參與者不存在，檢視器將無法互相聆聽。

檢視器參與者透過 WebRTC 協商連線後，只要主參與者也連線至儲存工作階段，擷取的音訊工作階段就會存放在 Kinesis 視訊串流中。

您也可以搭配擷取的 WebRTC 媒體，使用現有的 Kinesis Video Streams 功能，例如 HLS 或 DASH 播放、透過 [GetImages](#) 產生影像等等。

Note

目前不支援 S3 映像交付和通知。

Note

假設只有一個產生影片的裝置用戶端可以與頻道的工作階段建立關聯。如果多個用戶端將特定頻道的工作階段聯結為影片製作裝置，則以最新的用戶端請求為優先。

限制

目前限制已就緒：

- 檢視器數量上限：3
- 檢視器參與者保持連線至儲存工作階段的時間上限，但沒有主要參與者：3 分鐘

Important

如果檢視器與儲存工作階段中斷連線（關閉對等連線），其配額（檢視器限制）會維持 1 分鐘。在此 1 分鐘期間，檢視器可以使用相同的用戶端 ID 叫用此 API，以重新加入工作階段，而不需要額外的檢視器配額。1 分鐘後，檢視器配額會釋出，並可供其他檢視器加入。

其他資訊

- 等冪 - 此 API 不是等冪。
- 重試行為 - 這計為新的 API 呼叫。
- 並行呼叫 - 允許並行呼叫。每次呼叫會傳送一次提議。

請求語法

```
POST /joinStorageSessionAsViewer HTTP/1.1
Content-type: application/json
```

```
{
  "channelArn": "string",
  "clientId": "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)。

Important

請注意此輸入參數的大寫。

類型：字串

模式：`arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

必要：是

clientId

寄件者用戶端的唯一識別符。

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
```

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

您沒有執行此操作所需的許可。

HTTP 狀態碼：403

ClientLimitExceededException

Kinesis Video Streams 已調節請求，因為您已超過允許的用戶端呼叫限制。稍後嘗試撥打電話。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 命令列介面](#)
- [AWS 適用於 .NET 的 SDK](#)
- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Go 的 SDK v2](#)
- [AWS 適用於 Java V2 的開發套件](#)
- [AWS 適用於 JavaScript V3 的 SDK](#)
- [AWS 適用於 Kotlin 的 SDK](#)
- [AWS 適用於 PHP V3 的 SDK](#)
- [AWS 適用於 Python 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

資料類型

Amazon Kinesis Video Streams 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 Video Streams Media 支援下列資料類型：

- [StartSelector](#)

Amazon Kinesis Video Streams Archived Media 支援下列資料類型：

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Amazon Kinesis Video Signaling Channels 支援下列資料類型：

- [IceServer](#)

Amazon Kinesis Video WebRTC Storage 支援下列資料類型：

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams 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

封裝訊號頻道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

訊號頻道的名稱。

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

訊號頻道的目前狀態。

類型：字串

有效值:CREATING | ACTIVE | UPDATING | DELETING

必要：否

ChannelType

訊號頻道的類型。

類型：字串

有效值:SINGLE_MASTER | FULL_MESH

必要：否

CreationTime

訊號頻道建立的時間。

類型：Timestamp

必要：否

SingleMasterConfiguration

包含SINGLE_MASTER頻道類型組態的結構。

類型：[SingleMasterConfiguration](#) 物件

必要：否

Version

目前版本的訊號頻道。

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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ChannelNameCondition

服務：Amazon Kinesis Video Streams

ListSignalingChannels API 的選用輸入參數。當呼叫時指定此參數時ListSignalingChannels，API 只會傳回滿足中指定條件的頻道ChannelNameCondition。

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開頭為指定字首的訊號頻道。

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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DeletionConfig

服務：Amazon Kinesis Video Streams

從 Edge Agent 刪除串流連線所需的組態詳細資訊。

目錄

DeleteAfterUpload

用來表示是否要在媒體上傳至 Kinesis Video Stream 雲端後，將媒體標記為刪除boolean的值。如果任何刪除組態值設為 true，例如當 EdgeRetentionInHours或已達到限制時MaxLocalMediaSizeInMB，可以刪除媒體檔案。

由於預設值設為 true，請設定上傳者排程，以便在媒體檔案最初上傳至 AWS 雲端之前不會將其刪除。

類型：布林值

必要：否

EdgeRetentionInHours

您想要在 Edge Agent 上的串流中保留資料的時數。保留時間的預設值為 720 小時，即 30 天。

類型：整數

有效範圍：最小值為 1。最大值為 720。

必要：否

LocalSizeConfig

刪除邊緣組態所需的本機大小值。

類型：[LocalSizeConfig](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

EdgeAgentStatus

服務：Amazon Kinesis Video Streams

物件，其中包含邊緣代理程式記錄器和上傳器任務的最新狀態詳細資訊。使用此資訊來判斷邊緣代理程式的目前運作狀態。

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

串流邊緣到雲端上傳程式任務的最新狀態。

類型：[LastUploaderStatus](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

EdgeConfig

服務：Amazon Kinesis Video Streams

串流邊緣組態的描述，將用於與 Edge Agent IoT Greengrass 元件同步。Edge Agent 元件會在您現場部署的 IoT Hub 裝置設定上執行。

目錄

HubDeviceArn

串流的「物聯網 (IoT) 物件」。

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

記錄器組態包含本機 `MediaSourceConfig` 詳細資訊，做為登入資料來存取在攝影機上串流的本機媒體檔案。

類型：[RecorderConfig](#) 物件

必要：是

DeletionConfig

刪除組態是由用於進行刪除的保留時間 (`EdgeRetentionInHours`) 和本機大小組態 (`LocalSizeConfig`) 詳細資訊所組成。

類型：[DeletionConfig](#) 物件

必要：否

UploaderConfig

上傳程式組態包含用於將錄製媒體檔案從 Edge Agent 上傳到 Kinesis Video Stream `ScheduleExpression` 的詳細資訊。

類型：[UploaderConfig](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ImageGenerationConfiguration

服務：Amazon Kinesis Video Streams

包含 KVS 影像交付所需資訊的結構。如果為 null，則會從串流中刪除組態。

目錄

DestinationConfig

包含將映像交付給客戶所需的資訊之結構。

類型：[ImageGenerationDestinationConfig](#) 物件

必要：是

Format

接受的影像格式。

類型：字串

有效值:JPEG | PNG

必要：是

ImageSelectorType

用於產生映像的伺服器或生產者時間戳記來源。

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，以毫秒 (ms) 為單位。可以提供的最小值為 200 毫秒。如果時間戳記範圍小於取樣間隔，StartTimestamp則會在可用時傳回來自的影像。

類型：整數

必要：是

Status

指出 ContinuousImageGenerationConfigurations API 是啟用或停用。

類型：字串

有效值:ENABLED | DISABLED

必要：是

FormatConfig

索引鍵/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。FormatConfig 金鑰是 JPEGQuality，表示要用來產生影像的 JPEG 品質金鑰。此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的壓縮產生影像。如果值為 100，則會以最佳品質和較少的壓縮產生影像。如果未提供值，JPEGQuality則金鑰的預設值會設為 80。

類型：字串到字串映射

映射項目：最多 1 個項目。

有效金鑰：JPEGQuality

值長度限制：長度下限為 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映像高度。同時提供 HeightPixels和 WidthPixels 參數時，影像會延伸以符合指定的長寬比。如果只提供 HeightPixels 參數，則會使用其原始長寬比來計算WidthPixels比率。如果未提供參數，則會傳回原始影像大小。

類型：整數

有效範圍：最小值為 1。最大值為 2160。

必要：否

WidthPixels

與 HeightPixels 參數搭配使用的輸出映像寬度。同時提供 WidthPixels和 HeightPixels 參數時，影像會延伸以符合指定的長寬比。如果只提供 WidthPixels 參數，則會使用其原始長寬比來計算HeightPixels比率。如果未提供參數，則會傳回原始影像大小。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列各項：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ImageGenerationDestinationConfig

服務：Amazon Kinesis Video Streams

包含將映像交付給客戶所需的資訊之結構。

目錄

DestinationRegion

S3 儲存貯體的 AWS 區域，其中將交付影像。這DestinationRegion必須符合串流所在的區域。

類型：字串

長度限制：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

統一資源識別符 (URI)，用於識別影像的交付位置。

類型：字串

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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

LastRecorderStatus

服務：Amazon Kinesis Video Streams

串流邊緣錄製工作的最新狀態。

目錄

JobStatusDetails

記錄器任務最新狀態的描述。

類型：字串

必要：否

LastCollectedTime

記錄器任務上次執行的時間戳記，以及存放到本機磁碟的媒體。

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

最新記錄器任務的狀態。

類型：字串

有效值:SUCCESS | USER_ERROR | SYSTEM_ERROR

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

LastUploaderStatus

服務：Amazon Kinesis Video Streams

串流邊緣到雲端上傳程式任務的最新狀態。

目錄

JobStatusDetails

上傳者任務最新狀態的說明。

類型：字串

必要：否

LastCollectedTime

上次執行上傳者任務的時間戳記，以及收集至雲端的媒體。

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

最新上傳者任務的狀態。

類型：字串

有效值:SUCCESS | USER_ERROR | SYSTEM_ERROR

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ListEdgeAgentConfigurationsEdgeConfig

服務：Amazon Kinesis Video Streams

單一串流邊緣組態的描述。

目錄

CreationTime

串流第一次建立邊緣組態時的時間戳記。

類型：Timestamp

必要：否

EdgeConfig

串流邊緣組態的描述，將用於與 Edge Agent IoT Greengrass 元件同步。Edge Agent 元件會在您現場部署的 IoT Hub 裝置設定上執行。

類型：[EdgeConfig](#) 物件

必要：否

FailedStatusDetails

所產生失敗狀態的描述。

類型：字串

必要：否

LastUpdatedTime

串流上次更新邊緣組態時的時間戳記。

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

串流邊緣組態的目前同步狀態。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

LocalSizeConfig

服務：Amazon Kinesis Video Streams

組態詳細資訊，其中包含您要在 Edge Agent 上存放串流的媒體大小上限 (MaxLocalMediaSizeInMB)，以及當串流大小已達到上限時應使用的策略 (StrategyOnFullSize)。

目錄

MaxLocalMediaSizeInMB

您想要在 Edge Agent 上存放串流的媒體整體大小上限。

類型：整數

有效範圍：最小值為 64。最大值為 2000000。

必要：否

StrategyOnFullSize

達到串流MaxLocalMediaSizeInMB限制時要執行的策略。

類型：字串

有效值:DELETE_OLDEST_MEDIA | DENY_NEW_MEDIA

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

MappedResourceConfigurationListItem

服務：Amazon Kinesis Video Streams

封裝或包含媒體儲存組態屬性的結構。

目錄

ARN

與串流相關聯的 Kinesis Video Stream 資源的 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

Type

kinesis 影片串流的關聯資源類型。

類型：字串

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

MediaSourceConfig

服務：Amazon Kinesis Video Streams

組態詳細資訊，包含存取串流至攝影機之媒體檔案所需的登入資料 (MediaUriSecretArn 和 MediaUriType)。

目錄

MediaUriSecretArn

相機使用者名稱和密碼的 AWS Secrets Manager ARN，或本機媒體檔案位置。

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

統一資源識別符 (URI) 類型。FILE_URI 值可用來串流本機媒體檔案。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)

- [AWS 適用於 Ruby V3 的 SDK](#)

MediaStorageConfiguration

服務：Amazon Kinesis Video Streams

封裝或包含媒體儲存組態屬性的結構。

- 如果StorageStatus已啟用，資料將存放在StreamARN提供的 中。為了讓 WebRTC 擷取運作，串流必須啟用資料保留。
- 如果 StorageStatus 已停用，則不會儲存任何資料，而且不需要 StreamARN 參數。

目錄

Status

媒體儲存組態的狀態。

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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

NotificationConfiguration

服務：Amazon Kinesis Video Streams

使用此 API 可設定 Amazon Simple Notification Service (Amazon SNS) 通知，以便在串流中提供片段時通知。如果此參數為 null，則會從串流中刪除組態。

如需詳細資訊，請參閱 [Kinesis Video Streams 中的通知](#)。

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

指出通知組態是啟用或停用。

類型：字串

有效值:ENABLED | DISABLED

必要：是

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

NotificationDestinationConfig

服務：Amazon Kinesis Video Streams

包含向客戶傳送通知所需資訊的結構。

目錄

Uri

統一資源識別符 (URI)，用於識別影像的交付位置。

類型：字串

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

RecorderConfig

服務：Amazon Kinesis Video Streams

錄製器組態包含本機MediaSourceConfig詳細資訊，這些詳細資訊會做為登入資料，用來存取攝影機上串流的本機媒體檔案。

目錄

MediaSourceConfig

組態詳細資訊，包含存取串流至攝影機的媒體檔案所需的登入資料 (MediaUriSecretArn 和 MediaUriType)。

類型：[MediaSourceConfig](#) 物件

必要：是

ScheduleConfig

包含的組態，ScheduleExpression以及指定從攝影機或本機媒體檔案記錄到 Edge Agent 的排程DurationInMinutes的詳細資訊。如果未提供 ScheduleExpression 屬性，則 Edge Agent 一律會設定為錄製模式。

類型：[ScheduleConfig](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ResourceEndpointListItem

服務：Amazon Kinesis Video Streams

描述 GetSignalingChannelEndpoint API 傳回之訊號頻道端點的物件。

媒體伺服器端點將對應至WEBRTC通訊協定。

目錄

Protocol

GetSignalingChannelEndpoint API 傳回的訊號頻道通訊協定。

類型：字串

有效值:WSS | HTTPS | WEBRTC

必要：否

ResourceEndpoint

GetSignalingChannelEndpoint API 傳回的訊號頻道端點。

類型：字串

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

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 包含 ScheduleExpression 和 DurationInMinutes 屬性。

如果未在 ScheduleConfig 中提供 RecorderConfig，則 Edge Agent 一律會設定為錄製模式。

如果未在 ScheduleConfig 中提供 UploaderConfig，則 Edge Agent 會定期上傳（每 1 小時）。

目錄

DurationInSeconds

記錄媒體的總持續時間。如果提供 ScheduleExpression 屬性，則應該指定 DurationInSeconds 屬性。

類型：整數

有效範圍：最小值為 60。最大值為 3600。

必要：是

ScheduleExpression

負責從攝影機或本機媒體檔案將任務排程到 Edge Agent 的 Quartz cron 表達式。如果 ScheduleExpression 未提供 RecorderConfig，則 Edge Agent 一律會設定為錄製模式。

如需關於 Quartz 的詳細資訊，請參閱 [Cron Trigger 教學](#) 課程頁面以了解有效的表達式及其用途。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

SingleMasterChannelEndpointConfiguration

服務：Amazon Kinesis Video Streams

包含SINGLE_MASTER頻道類型的端點組態的物件。

目錄

Protocols

此屬性用於判斷透過此SINGLE_MASTER訊號頻道進行通訊的性質。如果指定 WSS ，則此 API 會傳回 WebSocket 端點。如果指定 HTTPS ，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

此屬性用於判斷此SINGLE_MASTER訊號頻道中的訊息許可。如果指定 MASTER ，此 API 會傳回端點，用戶端可用來接收來自此訊號頻道上任何檢視器的優惠，並將答案傳送給該檢視器。如果指定 VIEWER ，此 API 會傳回端點，用戶端只能使用此端點將優惠傳送到此訊號頻道上的另一個MASTER用戶端。

類型：字串

有效值:MASTER | VIEWER

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

SingleMasterConfiguration

服務：Amazon Kinesis Video Streams

包含SINGLE_MASTER頻道類型組態的結構。

目錄

MessageTtlSeconds

訊號頻道在捨棄未傳遞的訊息之前，會保留一段時間（以秒為單位）。使用 [UpdateSignalingChannel](#) 更新此值。

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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

StreamInfo

服務：Amazon Kinesis Video Streams

描述 Kinesis 影片串流的物件。

目錄

CreationTime

指出何時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。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

StreamNameCondition

服務：Amazon Kinesis Video Streams

指定當您列出串流時，串流必須滿足才能傳回的條件（請參閱 `ListStreams` API）。條件具有比較操作和值。目前，您只能指定運算 `BEGINS_WITH` 子，該運算子會尋找名稱開頭為指定字首的串流。

目錄

ComparisonOperator

比較運算子。目前，您只能指定運算 `BEGINS_WITH` 子，該運算子會尋找名稱開頭為指定字首的串流。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Tag

服務：Amazon Kinesis Video Streams

與指定訊號通道相關聯的金鑰和值對。

目錄

Key

與指定訊號通道相關聯的標籤索引鍵。

類型：字串

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

與指定訊號通道相關聯的標籤值。

類型：字串

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

UploaderConfig

服務：Amazon Kinesis Video Streams

包含的組態，ScheduleExpression以及指定從攝影機或本機媒體檔案記錄到 Edge Agent 的排程DurationInMinutes的詳細資訊。如果未在 ScheduleConfig中提供 UploaderConfig，則 Edge Agent 會定期上傳（每 1 小時）。

目錄

ScheduleConfig

包含的組態，ScheduleExpression以及指定從攝影機或本機媒體檔案記錄到 Edge Agent 的排程DurationInMinutes的詳細資訊。如果未在此 ScheduleConfig中提供 UploaderConfig，則 Edge Agent 會定期上傳（每 1 小時）。

類型：[ScheduleConfig](#) 物件

必要：是

另請參閱

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Streams 媒體

Amazon Kinesis Video Streams Media 支援下列資料類型：

- [StartSelector](#)

StartSelector

服務：Amazon Kinesis Video Streams Media

識別 Kinesis 影片串流上的區塊，您希望 GetMedia API 開始傳回媒體資料。您有下列選項來識別開始區塊：

- 選擇最新（或最舊）區塊。
- 識別特定區塊。您可以提供片段編號或時間戳記（伺服器或生產者）來識別特定區塊。
- 每個區塊的中繼資料都包含接續字符做為 Matroska (MKV) 標籤 (`AWS_KINESISVIDEO_CONTINUATION_TOKEN`)。如果您之前的 GetMedia 請求已終止，您可以在下一個 GetMedia 請求中使用此標籤值。然後，API 會從最後一個 API 結束的位置開始傳回區塊。

目錄

StartSelectorType

識別 Kinesis 影片串流上您要開始從中取得資料的片段。

- 立即 - 從串流上的最新區塊開始。
- EARLIEST - 從串流上最早的可用區塊開始。
- FRAGMENT_NUMBER - 從特定片段之後的區塊開始。您也必須指定 `AfterFragmentNumber` 參數。
- PRODUCER_TIMESTAMP 或 SERVER_TIMESTAMP - 從包含具有指定生產者或伺服器時間戳記之片段的區塊開始。您可以新增 `StartTimeStamp` 來指定時間戳記。
- CONTINUATION_TOKEN - 使用指定的接續字符讀取。

Note

如果您選擇 `NOW`、`EARLIEST` 或 `CONTINUATION_TOKEN` 做為 `startSelectorType`，則不會在 `startSelector` 中提供任何其他資訊。

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 在上一個 GetMedia 回應中傳回的連續字符。然後，GetMedia API 會從接續字符識別的區塊開始。

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 PRODUCER_TIMESTAMP 或 SERVER_TIMESTAMP 作為 `startSelectorType`，則需要此值。GetMedia API 接著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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Streams 封存媒體

Amazon Kinesis Video Streams Archived Media 支援下列資料類型：

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

ClipFragmentSelector

服務：Amazon Kinesis Video Streams Archived Media

描述一系列片段的時間戳記範圍和時間戳記原始伺服器。

具有重複生產者時間戳記的片段會重複刪除。這表示，如果生產者生產的片段串流具有大約等於真實時間的生產者時間戳記，則剪輯將包含請求時間戳記範圍內的所有片段。如果某些片段是在相同的時間範圍內和非常不同的時間點擷取，則只會傳回擷取時間最久的片段集合。

目錄

FragmentSelectorType

要使用的時間戳記來源（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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

ClipTimestampRange

服務：Amazon Kinesis Video Streams Archived Media

要傳回片段的時間戳記範圍。

目錄

EndTimeStamp

所請求媒體的時間戳記範圍結束。

此值必須在指定的 24 小時內 StartTimestamp，且必須晚於該 StartTimestamp 值。如果請求 FragmentSelectorType 的是 SERVER_TIMESTAMP，則此值必須是過去的。

此值包含。EndTimeStamp 會與片段的（開始）時間戳記進行比較。在 EndTimeStamp 值之前開始並繼續超過的片段會包含在工作階段中。

類型：Timestamp

必要：是

StartTimestamp

傳回片段之時間戳記範圍內的起始時間戳記。

只有完全在 或之後開始的片段 StartTimestamp 才會包含在工作階段中。在之前開始 StartTimestamp 並繼續超過的片段不會包含在工作階段中。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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DASHFragmentSelector

服務：Amazon Kinesis Video Streams Archived Media

包含所請求媒體的時間戳記範圍，以及時間戳記的來源。

目錄

FragmentSelectorType

所請求媒體的時間戳記來源。

當 `FragmentSelectorType` 設定為 `PRODUCER_TIMESTAMP` 且

[GetDASHStreamingSessionURL : PlaybackMode](#) 為 `ON_DEMAND` 或 `LIVE_REPLAY`，在指定的 [FragmentSelector : TimestampRange](#) 內擷取的第一個片段會包含在媒體播放清單中。此外，在第一個片段之後立即 `TimestampRange` 擷取的內具有生產者時間戳記的片段（最高為 [GetDASHStreamingSessionURL : MaxManifestFragmentResults](#) 值）也會包含在內。

具有重複生產者時間戳記的片段會重複刪除。這表示，如果生產者生產的片段串流具有大約等於真實時鐘時間的生產者時間戳記，MPEG-DASH 資訊清單將包含請求時間戳記範圍內的所有片段。如果某些片段是在相同的時間範圍內和非常不同的時間點擷取，則只會傳回擷取時間最久的片段集合。

當 `FragmentSelectorType` 設為 `PRODUCER_TIMESTAMP` 且 [GetDASHStreamingSessionURL : PlaybackMode](#) 為 `LIVE`，生產者時間戳記會用於 MP4 片段和重複資料刪除。但是，根據伺服器時間戳記的最新擷取片段會包含在 MPEG-DASH 資訊清單中。這表示即使過去擷取的片段現在具有具有值的生產者時間戳記，它們也不會包含在 HLS 媒體播放清單中。

預設值為 `SERVER_TIMESTAMP`。

類型：字串

有效值：`PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

必要：否

TimestampRange

所請求媒體的時間戳記範圍開始和結束。

如果 `PlaybackType` 為 `LIVE`，則此值不應存在。

類型：[DASHTimestampRange](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

DASHTimestampRange

服務：Amazon Kinesis Video Streams Archived Media

所請求媒體的時間戳記範圍開始和結束。

如果 PlaybackType 為 `LIVE`，則此值不應存在 LIVE。

中的值 DASHTimestampRange 包含在內。工作階段中會包含開始時間或之後的確切片段。在開始時間之前開始並繼續超過的片段不會包含在工作階段中。

目錄

EndTimeStamp

所請求媒體的時間戳記範圍結束。此值必須在指定的 24 小時內 StartTimestamp，且必須晚於該 StartTimestamp 值。

如果請求 FragmentSelectorType 的是 `SERVER_TIMESTAMP`，則此值必須是過去的。

`ON_DEMAND` 模式需要此 EndTimeStamp 值，但 `LIVE_REPLAY` 模式為選用。如果 EndTimeStamp 未將設定為 `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。

Note

此值包含。EndTimeStamp 會與片段的（開始）時間戳記進行比較。在 EndTimeStamp 值之前開始並繼續超過的片段會包含在工作階段中。

類型：Timestamp

必要：否

StartTimeStamp

所請求媒體的時間戳記範圍開始。

如果指定 DASHTimestampRange 值，則需要該 StartTimestamp 值。

只有完全在或之後開始的片段 StartTimestamp 才會包含在工作階段中。在之前開始 StartTimestamp 並繼續超過的片段不會包含在工作階段中。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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Fragment

服務：Amazon Kinesis Video Streams Archived Media

代表影片區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

片段的唯一識別符。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，包括片段和包含媒體資料的相關資訊。

類型：Long

必要：否

ProducerTimestamp

生產者對應至片段的時間戳記，以毫秒為單位。

類型：Timestamp

必要：否

ServerTimestamp

對應至片段之 AWS 伺服器的時間戳記，以毫秒為單位。

類型：Timestamp

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

FragmentSelector

服務：Amazon Kinesis Video Streams Archived Media

描述一系列片段的時間戳記範圍和時間戳記原始伺服器。

只會傳回起始時間戳記大於或等於指定開始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啟動時間戳記的片段：

- 00 : 00 : 00
- 00 : 00 : 02
- 00 : 00 : 04
- 00 : 00 : 06

開始時間為 00 : 00 : 01 且結束時間為 00 : 00 : 04 的片段選取器範圍會傳回開始時間為 00 : 00 : 02 和 00 : 00 : 04 的片段。

目錄

FragmentSelectorType

要使用的時間戳記來源（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

如需在其中一種語言特定 AWS SDKs中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)

- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

HLSFragmentSelector

服務：Amazon Kinesis Video Streams Archived Media

包含所請求媒體的時間戳記範圍，以及時間戳記的來源。

目錄

FragmentSelectorType

所請求媒體的時間戳記來源。

當 `FragmentSelectorType` 設為 `PRODUCER_TIMESTAMP` 且 [GetHLSStreamingSessionURL : PlaybackMode](#) 為 `ON_DEMAND` 或 `LIVE_REPLAY`，在指定的 [FragmentSelector : TimestampRange](#) 內擷取的第一個片段會包含在媒體播放清單中。此外，在第一個片段之後立即 `TimestampRange` 擷取的內具有生產者時間戳記的片段（最高為 [GetHLSStreamingSessionURL : MaxMediaPlaylistFragmentResults](#) 值）也會包含在內。

具有重複生產者時間戳記的片段會重複刪除。這表示如果生產者生產的片段串流具有大約等於真實時鐘時間的生產者時間戳記，則 HLS 媒體播放清單將包含請求時間戳記範圍內的所有片段。如果某些片段是在相同的時間範圍內和非常不同的時間點擷取，則只會傳回擷取時間最久的片段集合。

當 `FragmentSelectorType` 設為 `PRODUCER_TIMESTAMP` 且 [GetHLSStreamingSessionURL : PlaybackMode](#) 為 `LIVE`，生產者時間戳記會用於 MP4 片段和重複資料刪除。但根據伺服器時間戳記擷取的最新片段會包含在 HLS 媒體播放清單中。這表示即使過去擷取的片段現在具有具有值的生產者時間戳記，它們也不會包含在 HLS 媒體播放清單中。

預設值為 `SERVER_TIMESTAMP`。

類型：字串

有效值：`PRODUCER_TIMESTAMP` | `SERVER_TIMESTAMP`

必要：否

TimestampRange

所請求媒體的時間戳記範圍開始和結束。

如果 `PlaybackType` 為 `LIVE`，則此值不應存在。

類型：[HLSTimestampRange](#) 物件

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

HLSTimestampRange

服務：Amazon Kinesis Video Streams Archived Media

所請求媒體的時間戳記範圍開始和結束。

如果 PlaybackType 為 `LIVE`，則此值不應存在 LIVE。

目錄

EndTimeStamp

所請求媒體的時間戳記範圍結束。此值必須在指定的 24 小時內 StartTimestamp，且必須晚於該 StartTimestamp 值。

如果請求 FragmentSelectorType 的是 `SERVER_TIMESTAMP`，則此值必須是過去的。

`ON_DEMAND` 模式需要此 EndTimestamp 值，但 `LIVE_REPLAY` 模式為選用。如果 EndTimestamp 未將設定為 `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。

Note

此值包含。EndTimeStamp 會與片段的（開始）時間戳記進行比較。在 EndTimestamp 值之前開始並繼續超過的片段會包含在工作階段中。

類型：Timestamp

必要：否

StartTimestamp

所請求媒體的時間戳記範圍開始。

如果指定 HLSTimestampRange 值，則需要該 StartTimestamp 值。

只有完全在 或之後開始的片段 StartTimestamp 才會包含在工作階段中。在之前開始 StartTimestamp 並繼續超過的片段不會包含在工作階段中。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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Image

服務：Amazon Kinesis Video Streams Archived Media

包含 Timestamp、Error 和 的結構 ImageContent。

目錄

Error

當因為無法嘗試的錯誤而未擷取所提供時間戳記的影像時，會顯示錯誤訊息。如果出現下列情況，將會傳回錯誤：

- 指定的 不存在媒體 Timestamp。
- 指定時間的媒體不允許擷取映像。在此情況下，媒體僅是音訊，或已擷取不正確的媒體。

類型：字串

有效值: NO_MEDIA | MEDIA_ERROR

必要：否

ImageContent

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

物件的屬性 Image，用於從影片串流擷取影像。此欄位用於管理映像上的差距，或更了解分頁視窗。

類型：Timestamp

必要：否

另請參閱

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

TimestampRange

服務：Amazon Kinesis Video Streams Archived Media

要傳回片段的時間戳記範圍。

目錄

EndTimeStamp

傳回片段之時間戳記範圍內的結束時間戳記。

類型：Timestamp

必要：是

StartTimeStamp

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱以下內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video Signaling 頻道

Amazon Kinesis Video Signaling Channels 支援下列資料類型：

- [IceServer](#)

IceServer

服務：Amazon Kinesis Video Signaling Channels

ICE 伺服器連線資料的結構。

目錄

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

使用者名稱和密碼有效的期間，以秒為單位。

類型：整數

有效範圍：固定值為 300。

必要：否

Uris

URI 陣列 URIs，採用 [I-D.petithuguenin-behave-turn-uris](#) 規格中指定的格式。這些 URIs 提供可用於連接 TURN 伺服器的不同地址和/或通訊協定。

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

如需在其中一種語言特定 AWS SDKs 中使用此 API 的詳細資訊，請參閱下列內容：

- [AWS 適用於 C++ 的 SDK](#)
- [AWS 適用於 Java V2 的 SDK](#)
- [AWS 適用於 Ruby V3 的 SDK](#)

Amazon Kinesis Video WebRTC 儲存體

Amazon Kinesis Video WebRTC Storage 支援下列資料類型：

常見錯誤

本節列出所有 AWS 服務的 API 動作常見的錯誤。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

請求簽章不符合 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

輸入無法滿足 AWS 服務指定的限制條件。

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 Signature 第 4 版的詳細資訊，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 [AWS API 請求簽章元素](#)。

類型：字串

必要：有條件

X-Amz-Security-Token

透過呼叫 AWS Security Token Service () 取得的臨時安全字串 AWS STS。如需支援 AWS STS 的臨時安全憑證的服務清單，請參閱《IAM 使用者指南》中的 [可搭配 IAM 運作的 AWS 服務](#)。

條件：如果您使用來自的臨時安全登入資料 AWS STS，則必須包含安全字串。

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。如需指定簽章標頭的詳細資訊，請參閱《IAM 使用者指南》中的 [建立簽章 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

Amazon Kinesis Video Streams 的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您將受益於資料中心和網路架構，該架構旨在滿足最安全敏感組織的需求。

安全性是 AWS 與您之間共同責任。[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 Cloud AWS 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。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](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

本文件可協助您了解如何在使用 Kinesis Video Streams 時套用共同責任模型。下列主題說明如何設定 Kinesis Video Streams 以符合您的安全與合規目標。您也將了解如何使用其他 AWS 服務，協助您監控和保護 Kinesis Video Streams 資源。

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 控制對 Kinesis Video Streams 資源的存取 AWS IoT](#)
- [Amazon Kinesis Video Streams 的合規驗證](#)
- [Amazon Kinesis Video Streams 中的彈性](#)
- [Kinesis Video Streams 中的基礎設施安全性](#)
- [Kinesis Video Streams 的安全最佳實務](#)

Kinesis Video Streams 中的資料保護

您可以使用 AWS Key Management Service () 金鑰使用伺服器端加密 (SSE AWS KMS)，透過在 Amazon Kinesis Video Streams 中加密靜態資料來滿足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、區域和效能考量事項](#)

- [如何開始使用伺服器端加密？](#)
- [建立和使用客戶受管金鑰](#)
- [使用客戶受管金鑰的許可](#)

什麼是 Kinesis Video Streams 的伺服器端加密？

伺服器端加密是 Kinesis Video Streams 的一項功能，在使用 AWS KMS 您指定的金鑰儲存靜態資料之前，會自動加密資料。資料會在寫入 Kinesis Video Streams 串流儲存層之前加密，並在從儲存體擷取後解密。因此，您的資料一律會在 Kinesis Video Streams 服務內進行靜態加密。

透過伺服器端加密，您的 Kinesis 影片串流生產者和消費者不需要管理 KMS 金鑰或密碼編譯操作。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，因此靜態資料會加密。AWS KMS 提供伺服器端加密功能所使用的所有金鑰。會 AWS KMS 簡化 KMS 金鑰的使用，用於由匯入 AWS KMS 服務 AWS 的使用者指定 AWS KMS 金鑰所管理的 Kinesis Video Streams。

成本、區域和效能考量事項

當您套用伺服器端加密時，您需要支付 AWS KMS API 用量和金鑰成本。與自訂 AWS KMS 金鑰不同，預設 `aws/kinesisvideo` KMS 金鑰是免費提供的。不過，您仍必須支付 Kinesis Video Streams 代您產生的 API 使用成本。

API 使用成本適用於每個 KMS 金鑰，包括自訂金鑰。AWS KMS 成本會隨著您在資料生產者和消費者在使用的使用者登入資料數量而擴展，因為每個使用者登入資料都需要唯一的 API 呼叫 AWS KMS。

以下依據資源說明各項成本：

金鑰

- 由管理之 Kinesis Video Streams 的 KMS 金鑰 AWS (別名 = `aws/kinesisvideo`) 不收費。
- 使用者產生的 KMS 金鑰需支付 AWS KMS key 費用。如需詳細資訊，請參閱 [AWS Key Management Service 定價](#)。

AWS KMS API 用量

產生新資料加密金鑰或擷取現有加密金鑰的 API 請求會隨著流量增加而增加，並需支付 AWS KMS 使用成本。如需詳細資訊，請參閱 [AWS Key Management Service 定價：用量](#)。

即使保留設為 0 (無保留)，Kinesis Video Streams 也會產生金鑰請求。

區域伺服器端加密的可用性

Kinesis 影片串流的伺服器端加密可在提供 AWS 區域 Kinesis 影片串流的所有 中使用。

如何開始使用伺服器端加密？

Kinesis Video Streams 一律啟用伺服器端加密。如果建立串流時未指定使用者提供的金鑰，則會使用 AWS 受管金鑰（由 Kinesis Video Streams 提供）。

使用者提供的 KMS 金鑰必須在建立時指派給 Kinesis 影片串流。您稍後無法使用 [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 影片串流：

- 在 中建立 Kinesis 影片串流時 AWS Management Console，請在建立新影片串流頁面的加密索引標籤中指定 KMS 金鑰。
- 使用 [CreateStream](#) API 建立 Kinesis 影片串流時，請在 `KmsKeyId` 參數中指定金鑰 ID。

建立和使用客戶受管金鑰

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。

建立客戶受管金鑰

如需有關如何建立自有金鑰的資訊，請參閱《AWS Key Management Service 開發人員指南》中的[建立金鑰](#)。為您的帳戶建立金鑰後，Kinesis Video Streams 服務會在客戶受管金鑰清單中傳回這些金鑰。

使用客戶受管金鑰

將正確的許可套用至您的消費者、生產者和管理員之後，您可以在自己的 AWS 帳戶 或其他 中使用自訂 KMS 金鑰 AWS 帳戶。您帳戶中的所有 KMS 金鑰都會出現在 主控台上的客戶受管金鑰清單中。

若要使用位於另一個帳戶中的自訂 KMS 金鑰，您必須具有使用這些金鑰的許可。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`操作之前，不會存取 KMS 金鑰。如此會產生下列結果：

- 如果您指定的金鑰不存在，CreateStream操作會成功，但串流上的 PutMedia和 GetMedia操作會失敗。
- 如果您使用提供的金鑰 (aws/kinesisvideo)，在執行第一個 PutMedia或 GetMedia操作之前，金鑰不會出現在您的帳戶中。

使用客戶受管金鑰的許可

您必須先設定 KMS 金鑰政策以允許加密串流，以及加密和解密串流記錄，才能搭配客戶受管金鑰使用伺服器端加密。如需 AWS KMS 許可的範例和詳細資訊，請參閱 [AWS KMS API 許可：動作和資源參考](#)。

Note

使用預設服務金鑰進行加密不需要套用自訂 IAM 許可。

在使用客戶受管金鑰之前，請確認您的 Kinesis 影片串流生產者和消費者 (IAM 主體) 是 AWS KMS 預設金鑰政策中的使用者。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](#)。

範例生產者許可

您的 Kinesis 影片串流製作者必須擁有 kms:GenerateDataKey 許可：

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

範例取用者許可

您的 Kinesis 影片串流取用者必須擁有 `kms:Decrypt` 許可：

```
{
  "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

您可以搭配 Amazon Kinesis Video Streams 使用 AWS Identity and Access Management (IAM)，以控制組織中的使用者是否可以使用特定 Kinesis Video Streams API 操作來執行任務，以及是否可以使用特定 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](#)
- [Kinesis Video Streams 的動作](#)
- [Kinesis Video Streams 的 Amazon Resource Name \(ARNs\)](#)
- [授予其他 IAM 帳戶對 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 – 動作是您授予或拒絕許可的特定 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 – 條件為選用。您可以使用它們來控制何時政策開始生效。

當您建立和管理 IAM 政策時，建議您使用 [IAM 政策產生器](#) 和 [IAM 政策模擬器](#)。

Kinesis Video Streams 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，請使用下列字首搭配 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 操作，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](#)。

Kinesis Video Streams 的 Amazon Resource Name (ARNs)

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 使用以下 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
```

您可以使用 [DescribeStream](#) 取得串流的 ARN。

授予其他 IAM 帳戶對 Kinesis 影片串流的存取權

您可能需要將許可授予其他 IAM 帳戶，才能對 Kinesis Video Streams 中的串流執行操作。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. 取得您要授予許可的帳戶的 12 位數帳戶 ID，以對帳戶中建立的串流資源執行操作。

範例：在下列步驟中，我們將使用 111111111111 做為您要授予許可的帳戶的帳戶 ID，並使用 999999999999 做為 Kinesis Video Streams 的 ID

2. 在擁有串流 (999999999999) 的帳戶中建立 IAM 受管政策，以允許您要授予的存取層級。

範例政策：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia"
      ],
      "Resource": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-stream-name/1613732218179"
    }
  ]
}
```

如需 Kinesis Video Streams 資源的其他範例政策，請參閱下一節[範例政策](#)中的。

3. 在擁有串流 (999999999999 的帳戶中建立角色)，並指定您要授予 (111111111111 許可的帳戶)。這會將信任的實體新增至角色。

信任政策範例：

JSON

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

```
        "Principal": {
            "AWS": "arn:aws:iam::111111111111:root"
        },
        "Action": "sts:AssumeRole"
    }
}
}
```

將您在上一個步驟中建立的政策連接到此角色。

您現在已在帳戶 999999999999 中建立角色，該帳戶在受管政策中的串流資源 ARN PutMedia 上具有 DescribeStream、GetDataEndpoint 和 等操作的許可。這個新角色也會信任另一個帳戶 111111111111 擔任此角色。

Important

請記下角色 ARN，您將在下一個步驟中使用該角色。

4. 在另一個帳戶 111111111111 中建立受管政策，允許對您在上一個步驟中在帳戶 999999999999 中建立的角色執行 AssumeRole 動作。您需要提及上一個步驟中的角色 ARN。

範例政策：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
  }
}
```

5. 將上一個步驟中建立的政策連接到 IAM 實體，例如帳戶 111111111111 中的角色或使用者。此使用者現在具有在帳戶 999999999999 CustomRoleName 中擔任角色的許可。

此使用者的登入資料會呼叫 AWS STS AssumeRole API 以取得工作階段登入資料，之後會用來在帳戶 999999999999 中建立的串流上呼叫 Kinesis Video Streams APIs。

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/CustomRoleName" --
role-session-name "kvs-cross-account-assume-role"
{
  "Credentials": {
    "AccessKeyId": "",
    "SecretAccessKey": "",
    "SessionToken": "",
    "Expiration": ""
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "",
    "Arn": ""
  }
}
```

6. 根據環境中的前一組設定，設定存取金鑰、私密金鑰和工作階段登入資料。

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=
```

7. 執行 Kinesis Video Streams APIs 來描述和取得帳戶 999999999999 中串流的資料端點。

```
aws kinesismedia describe-stream --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179"
{
  "StreamInfo": {
    "StreamName": "custom-stream-name",
    "StreamARN": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-
stream-name/1613732218179",
    "KmsKeyId": "arn:aws:kms:us-west-2:999999999999:alias/aws/kinesisvideo",
    "Version": "abcd",
    "Status": "ACTIVE",
    "CreationTime": "2018-02-19T10:56:58.179000+00:00",
    "DataRetentionInHours": 24
  }
}

aws kinesismedia get-data-endpoint --stream-arn "arn:aws:kinesisvideo:us-
west-2:999999999999:stream/custom-stream-name/1613732218179" --api-name "PUT_MEDIA"
{
  "DataEndpoint": "https://s-b12345.kinesisvideo.us-west-2.amazonaws.com"
```

```
}
```

如需授予跨帳戶存取權的一般step-by-step說明，請參閱[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存取 Kinesis Video Streams

Example 1：允許使用者從任何 Kinesis 影片串流取得資料

此政策允許使用者或群組在任何 Kinesis 影片串流上執行 DescribeStream、ListStreams、GetDataEndpoint GetMedia和 ListTagsForStream操作。此政策適用於可以取得任何視訊串流資料的使用者。

JSON

```
{
  "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 影片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

此政策允許使用者或群組在任何資源上執行任何 Kinesis Video Streams 操作。此政策適用於管理員。

JSON

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

Example 4 : 允許使用者將資料寫入特定的 Kinesis 影片串流

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。

JSON

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

```
"Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/
your_stream/0123456789012"
    }
  ]
}
```

使用 控制對 Kinesis Video Streams 資源的存取 AWS IoT

本節說明如何讓裝置（例如攝影機）僅將音訊和視訊資料傳送至特定 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，AWS IoT 使用 TLS 交互身分驗證通訊協定連線至。其他 AWS 服務（例如 Kinesis Video Streams）不支援以憑證為基礎的身分驗證，但可以使用 AWS Signature 第 4 版格式的憑證進行 AWS 呼叫。Signature 第 4 版演算法通常要求發起人擁有存取金鑰 ID 和私密存取金鑰。AWS IoT 具有登入資料提供者，可讓您使用內建的 X.509 憑證做為唯一裝置身分來驗證 AWS 請求（例如，對 Kinesis Video Streams 的請求）。這不需要在您的裝置上存放存取金鑰 ID 和私密存取金鑰。

登入資料提供者會使用 X.509 憑證來驗證用戶端（在此案例中是 Kinesis Video Streams 開發套件，該開發套件正在您要將資料傳送至影片串流的攝影機上執行），並發出暫時、有限權限的安全字符。您可以使用字符來簽署和驗證任何 AWS 請求（在此情況下是對 Kinesis Video Streams 的呼叫）。如需詳細資訊，請參閱[授權直接呼叫 AWS 服務](#)。

以這種方式驗證攝影機對 Kinesis Video Streams 的請求，需要您建立和設定 IAM 角色，並將適當的 IAM 政策連接到角色，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

- [串流名稱為AWS IoT ThingName](#)
- [AWS IoT CertificateId 做為串流名稱](#)
- [使用 AWS IoT 登入資料串流至硬式編碼的串流名稱](#)

串流名稱為AWS IoT ThingName

主題

- [步驟 1：建立 AWS IoT 物件類型和 AWS IoT 物件](#)
- [步驟 2：建立要由 擔任的 IAM 角色 AWS IoT](#)
- [步驟 3：建立和設定 X.509 憑證](#)
- [步驟 4：使用 Kinesis 影片串流測試 AWS IoT 登入資料](#)
- [步驟 5：在相機的檔案系統上部署 AWS IoT 憑證和登入資料，並將資料串流到您的影片串流](#)

步驟 1：建立 AWS IoT 物件類型和 AWS IoT 物件

在中 AWS IoT，物件是特定裝置或邏輯實體的表示。在這種情況下，AWS IoT 物件代表您要設定資源層級存取控制的 Kinesis 影片串流。若要建立物件，您必須先建立 AWS IoT 物件類型。您可以使用 AWS IoT 物件類型來存放與相同物件類型相關聯之所有物件常見的描述和組態資訊。

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` 實物類型的 `kvs_example_camera` 實物：

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

步驟 2：建立要由 擔任的 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，可決定身分可以和不可以執行的操作 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由 擔任 AWS IoT，以便在從用戶端執行登入資料授權請求時，從安全字符服務 (STS) 取得臨時登入資料。在這種情況下，用戶端是在相機上執行的 Kinesis Video Streams 開發套件。

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：

JSON

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

- 接著，將許可政策連接至您先前建立的 IAM 角色。此許可政策允許 AWS 資源的選擇性存取控制（支援操作的子集）。在這種情況下，AWS 資源是您希望相機傳送資料的影片串流。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 使用下列 IAM 政策 JSON `iam-permission-document.json`：

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，此政策僅在預留位置 指定的視訊串流 (AWS 資源) 上授權指定的動作(`${credentials-iot:ThingName}`)。當登入資料提供者在請求中傳送視訊串流名稱ThingName時，此 AWS IoT 預留位置會接受 AWS IoT 物件屬性的值。

3. 接著，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角色別名，以指出要擔任哪個 IAM 角色，才能從 STS 取得臨時登入資料。

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，您可以建立 政策，AWS IoT 讓 使用角色別名擔任具有憑證的角色（連接後）。

下列範例命令會建立 AWS IoT 名為 的政策 `KvsCameraIoTPolicy`。

```

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

```

您可以使用下列命令來建立 `iot-policy-document.json` 文件 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：建立和設定 X.509 憑證

裝置（您的影片串流）與之間的通訊 AWS IoT 會透過使用 X.509 憑證來保護。

1. 建立憑證，您必須為 AWS IoT 先前建立的政策附加憑證。

```
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 AWS IoT (KvsCameraIoTPolicy 先前建立) 的政策連接至此憑證。

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

3. 將 AWS IoT 物件 (kvs_example_camera_stream) 連接至您剛建立的憑證：

```
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 登入資料提供者授權請求，您需要 AWS IoT 登入資料端點，這對於您的 AWS 帳戶 ID 是唯一的。您可以使用下列命令來取得 AWS IoT 登入資料端點。

```
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 影片串流測試 AWS IoT 登入資料

現在，您可以測試到目前為止設定的 AWS IoT 登入資料。

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中建立的 AWS IoT 物件名稱相同的名稱來建立影片串流 (kvs_example_camera_stream)。

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

2. 接著，呼叫 AWS IoT 登入資料提供者以取得臨時登入資料：

```
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

您可以使用下列命令來取得 IOT_GET_CREDENTIAL_ENDPOINT：

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

輸出 JSON 包含 accessKey、secretKey 和 sessionToken，您可以用來存取 Kinesis Video Streams。

3. 針對您的測試，您可以使用這些登入資料來叫用範例影片串流的 Kinesis kvs_example_camera_stream Video Streams 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：在相機的檔案系統上部署 AWS IoT 憑證和登入資料，並將資料串流到您的影片串流

Note

本節中的步驟說明從使用的攝影機傳送媒體到 Kinesis 影片串流 [the section called "C++"](#)。

1. 將先前步驟中產生的 X.509 憑證、私有金鑰和 CA 憑證複製到相機的檔案系統。指定存放這些檔案的路徑、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 登入資料端點。
2. 下列範例命令使用 AWS IoT 憑證授權將影片傳送至 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 做為串流名稱

若要透過 AWS IoT 物件代表您的裝置（例如，您的攝影機），但授權不同的串流名稱，您可以使用屬性做為串流名稱，AWS IoT `certificateId` 並在串流上使用提供 Kinesis Video Streams 許可 AWS IoT。完成此操作的步驟與先前概述的步驟類似，但有一些變更。

- 將許可政策修改為您的 IAM 角色 (`iam-permission-document.json`)，如下所示：

JSON

```
{
  "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 許可將起作用。從憑證取得憑證 ID，以便您可以在以下描述串流 API 呼叫中使用它做為串流名稱。

```
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}

```

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

```

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 AWS IoT 傳遞 thingname 給登入資料提供者。您可以使用將物件名稱 getenv 傳遞至示範應用程式，類似於傳遞其他 AWS IoT 屬性。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 登入資料串流至硬式編碼的串流名稱

若要透過 AWS IoT 物件代表您的裝置（例如，您的攝影機），但授權串流到特定的 Amazon Kinesis 影片串流，請使用在串流上提供 Amazon Kinesis Video Streams 許可 AWS IoT。程序類似於前面的章節，但有一些變更。

將許可政策修改為您的 IAM 角色 (`iam-permission-document.json`)，如下所示：

JSON

```
{
  "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、私有金鑰和 CA 憑證複製到相機的檔案系統。

指定存放這些檔案的路徑、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 登入資料端點。

下列範例命令使用 AWS IoT 憑證授權將影片傳送至 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"
```

Amazon Kinesis Video Streams 的合規驗證

若要了解 是否 AWS 服務 在特定合規計劃的範圍內，請參閱[AWS 服務 合規計劃](#)範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱[AWS 合規計劃](#)。

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[在 中下載報告 AWS Artifact](#)。

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 提供下列資源來協助合規：

- [安全合規與治理](#) - 這些解決方案實作指南內容討論了架構考量，並提供部署安全與合規功能的步驟。
- [HIPAA 合格服務參考](#) - 列出 HIPAA 合格服務。並非所有 AWS 服務 都符合 HIPAA 資格。
- [AWS 合規資源](#) - 此工作手冊和指南的集合可能適用於您的產業和位置。
- [AWS 客戶合規指南](#) - 透過合規的角度了解共同責任模型。本指南摘要說明跨多個架構（包括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)) 保護 AWS 服務 和映射指南至安全控制的最佳實務。
- 《AWS Config 開發人員指南》中的[使用 規則評估資源](#) - AWS Config 服務會評估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 可讓您全面檢視其中的安全狀態 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 控制參考](#)」。
- [Amazon GuardDuty](#) - 這會監控您的環境是否有可疑和惡意活動，以 AWS 服務 偵測對您 AWS 帳戶、工作負載、容器和資料的潛在威脅。GuardDuty 可滿足特定合規架構所規定的入侵偵測需求，以協助您因應 PCI DSS 等各種不同的合規需求。
- [AWS Audit Manager](#) - 這 AWS 服務 可協助您持續稽核 AWS 用量，以簡化您管理風險的方式，以及符合法規和業界標準的方式。

Amazon Kinesis Video Streams 中的彈性

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置的。AWS 區域提供多個實體隔離和隔離的可用區域，這些區域以低延遲、高輸送量和高度備援聯網連接。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 AWS 區域和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

Kinesis Video Streams 中的基礎設施安全性

Amazon Kinesis Video Streams 是受管服務，受到 [Amazon Web Services：安全程序概觀](#) 白皮書中所述的 AWS 全球網路安全程序的保護。

您可以使用 AWS 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 主體相關聯的私密存取金鑰來簽署。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 的安全最佳實務

Amazon Kinesis Video Streams 提供許多安全功能，供您在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

授予許可時，您可以決定誰要取得哪些 Kinesis Video Streams 資源的許可。您還需針對這些資源啟用允許執行的動作，因此，您只應授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 的生產者只需要 PutMedia、GetStreamingEndpoint 和 DescribeStream。請勿授予生產者應用程式所有動作 (*) 或其他動作 (例如 GetMedia) 的許可。

如需詳細資訊，請參閱 [什麼是最低權限以及為什麼需要它？](#)

使用 IAM 角色

製作者和用戶端應用程式必須具有有效的登入資料，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。這些是不會自動輪換的長期登入資料，如果遭到入侵，可能會對業務產生重大影響。

反之，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。當您使用角色時，您不需要使用長期登入資料（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](#)

使用 CloudTrail 監控 API 呼叫

Kinesis Video Streams 可與 搭配使用 AWS CloudTrail，此服務提供由使用者、角色或 Kinesis Video Streams AWS 服務 中所採取動作的記錄。

您可以使用 CloudTrail 所收集的資訊來判斷對 Kinesis Video Streams 提出的請求、提出請求的 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。

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

對 Kinesis Video Streams 進行故障診斷

使用下列資訊對 Amazon Kinesis Video Streams 遇到的常見問題進行故障診斷。

主題

- [一般問題](#)
- [API 問題](#)
- [HLS 問題](#)
- [Java 問題](#)
- [生產者程式庫問題](#)
- [串流剖析器程式庫問題](#)
- [網路問題](#)

一般問題

本節說明使用 Kinesis Video Streams 時可能遇到的一般問題。

問題

- [延遲過高](#)

延遲過高

延遲可能是由於傳送至 Kinesis Video Streams 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量，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。

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：「服務在接收到最終 AckEvent 之前已關閉連線」](#)
- [錯誤：「STATUS_STORE_OUT_OF_MEMORY」](#)
- [錯誤：「憑證應範圍限定為有效區域。」](#)

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 AWS CLI 使用 output 類型設定，則會發生此錯誤 json。AWS CLI 使用預設輸出類型 () 重新設定 none。如需設定的詳細資訊 AWS CLI，請參閱 AWS CLI 命令參考中的[設定](#)。

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。當您取得端點時，請務必在 GetDataEndpoint 呼叫中包含下列參數，視要呼叫的 API 而定：

```
--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 `aws` 中執行下列項目 AWS CLI，並確認可以擷取串流資訊：

```
aws kinesishvideo describe-stream --stream-name StreamName --endpoint https://  
ServiceEndpoint.kinesisvideo.region.amazonaws.com
```

如果呼叫失敗，請參閱[故障診斷 AWS CLI 錯誤](#)以取得詳細資訊。

錯誤：「服務在接收到最終 AckEvent 之前已關閉連線」

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。確認 `unread()` 方法已正確實作。

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](#)。

錯誤：「憑證應範圍限定為有效區域。」

如果簽署區域與端點區域不相符，則會發生此錯誤。

例如，如果您指定 `us-west-2` 做為簽署區域，但嘗試連線到 `kinesisvideo.us-east-1.amazonaws.com(us-east-1)` 端點，則會收到此錯誤。

在某些應用程式中，例如 [kvssink](#)，區域後援鏈預設為 us-west-2。確認您已根據您使用的應用程式正確設定您的區域。

HLS 問題

如果您的影片串流未正確播放，請參閱 [the section called “故障診斷 HLS 問題”](#)。

Java 問題

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

若要對 Java 範例和程式庫的問題進行故障診斷，啟用和檢查偵錯日誌會很有幫助。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

本節說明使用 [上傳至 Kinesis Video Streams](#) 時可能遇到的問題。

問題

- [無法編譯生產者 SDK](#)
- [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](#)
- [已達到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](#)

無法編譯生產者 SDK

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)
```

如果 H.264 在串流中做為 SPS 和 PPS，CPD (Codec Private Data) 可以調整為 AvCC 格式。除非媒體管道單獨提供 CPD，否則應用程式可以透過尋找第一個 Idr 影格 (應包含 SPS 和 PPS) 來從影格擷取 CPD，擷取兩個 NALUs (將是 Ab(Sps)Ab(Pps))，並在其中的 CPD 中設定它 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。
- 將高品質串流設定為較高的位元速率，如果攝影機支援，則修正來源串流中的任何抖動。

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，請確認您已編譯應用程式以取得正確版本的裝置。（例如，當您在 Raspberry Pi 2 上執行時，請確認您沒有為 Raspberry Pi 3 編譯。）

```
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：

```
vcgencmd 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](#) 一行中提供此旗標：

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/StreamConfigurationFragment.java#L169>

已達到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 [the section called “媒體和封存媒體 API 服務配額”](#) 區段中的片段持續時間上限。

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：
 - 如果您使用的是金鑰影格型分割，請將編碼器設定為在 10 秒內提供金鑰影格。
 - 如果您未使用金鑰影格型分割，則在 中定義串流時 [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 管道中使用軟體編碼器（例如 x264），您可以在 10 秒內將 key-int-max 屬性設定為值。例如，將 key-int-max 設定為 60，fps 設定為 30，以每 2 秒啟用金鑰影格。
- 如果您使用的是 RPI 攝影機，請將金鑰影格間隔屬性設定為小於 10 秒。
- 如果您使用的是 IP (RTSP) 攝影機，請將 GOP 大小設定為 60。

使用 IoT 授權時的「傳遞的物件名稱無效」錯誤

為了避免在使用 IoT 憑證進行授權時出現此錯誤 (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 [GStreamer 元素參數參考](#)。

串流剖析器程式庫問題

本節說明使用 [使用剖析器程式庫進行串流](#) 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影格，請確認串流包含正確的編解碼器私有資料。如需有關串流資料格式的詳細資訊，請參閱 [資料模型](#)。

若要了解如何使用轉碼器私有資料存取影格的詳細資訊，請參閱 GitHub 網站上的以下測試檔案：[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。

網路問題

如果您在嘗試連線至 Kinesis Video Streams 時看到連線錯誤，例如「連線逾時」或「連線失敗」，可能是網路設定中的 IP 地址範圍限制所致。

如果您的設定有 Kinesis Video Streams 的 IP 地址範圍限制，請更新您的網路組態以允許列出 Kinesis Video Streams [IP 地址範圍](#)。

Important

IP 範圍清單不是 Kinesis Video Streams IP 地址的完整清單。包含您看到的 IP 地址範圍，並請注意 IP 地址可能會隨著時間而變更。

如需詳細資訊，請參閱 [AWS IP 範圍](#)。若要在 IP 範圍變更時收到通知，請遵循[訂閱程序](#)。

Amazon Kinesis Video Streams 的文件歷史記錄

下表說明自上次發行 Amazon Kinesis Video Streams 以來文件的重要變更。

- 最新的 API 版本：2017-11-29
- 文件最近更新時間：2025 年 1 月 6 日

變更	描述	日期
Raspberry Pi 上的 C++	重新整理在 Raspberry Pi 上使用 C++ 生產者 SDK 的文件。	2025 年 1 月 6 日
Amazon Kinesis Video Streams Edge Agent Edge-to-Cloud 連線	新功能版本。如需詳細資訊，請參閱 排程影片錄製和儲存 。	2023 年 6 月 27 日
入門：將資料傳送至 Kinesis 影片串流	將媒體資料從攝影機傳送至 Kinesis 影片串流的基本教學課程。如需詳細資訊，請參閱 將資料傳送至 Amazon Kinesis 影片串流 。	2019 年 1 月 21 日
串流中繼資料	您可以使用 Producer SDK 將中繼資料嵌入 Kinesis 影片串流。如需詳細資訊，請參閱 搭配 Kinesis Video Streams 使用串流中繼資料 。	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 搭配 C++ 生產者 SDK 使用記錄 。	2018 年 7 月 18 日
HLS 視訊串流	您現在可以使用 HTTP Live Streaming 檢視 Kinesis 影片串流。如需詳細資訊，請參	2018 年 7 月 13 日

變更	描述	日期
	閱 Kinesis Video Streams 播放 。	
使用 RTSP 來源進行串流	Kinesis Video Streams 的範例應用程式，會在 Docker 容器中執行，並從 RTSP 來源串流影片。如需詳細資訊，請參閱 RTSP 和 Docker 。	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 C++ 做為 GStreamer 目的地。如需詳細資訊，請參閱 GStreamer 外掛程式 - kvssink 。	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 上傳至 Kinesis Video Streams 。如需詳細資訊，請參閱 生產者開發套件回呼 。	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 Amazon Kinesis Video Streams 系統需求 。	2018 年 5 月 30 日
CloudTrail 支援	使用 CloudTrail 來監控 API 用量的文件。如需詳細資訊，請參閱 使用記錄 Amazon Kinesis Video Streams API 呼叫 AWS CloudTrail 。	2018 年 5 月 24 日
製作者開發套件結構參考文件	上傳至 Kinesis Video Streams 使用之結構的參考文件。如需詳細資訊，請參閱 生產者開發套件結構 和 Kinesis 影片串流結構 。	2018 年 5 月 7 日

變更	描述	日期
轉譯器範例文件	Renderer 範例應用程式的文件，說明如何從 Kinesis 影片串流解碼和顯示影格。如需詳細資訊，請參閱 範例：剖析和轉譯 Kinesis Video Streams 片段 。	2018 年 3 月 15 日
製作者開發套件限制參考文件	有關在 C++ 中操作限制之資訊。如需詳細資訊，請參閱 生產者開發套件配額 。	2018 年 3 月 13 日
監控	有關使用 Amazon CloudWatch 和 監控 Kinesis Video Streams 指標和 API 呼叫的資訊 AWS CloudTrail。如需詳細資訊，請參閱 監控 Amazon Kinesis Video Streams 。	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 NAL 適應旗標 。	2018 年 1 月 15 日
Android 支援串流視訊	Kinesis Video Streams 現在支援從 Android 裝置串流影片。如需詳細資訊，請參閱 Android 。	2018 年 1 月 12 日
Kinesis 影片範例文件	Kinesis Video 範例應用程式的文件，其中顯示如何在 使用剖析器程式庫觀察攝影機的輸出 應用程式中使用。如需詳細資訊，請參閱 KinesisVideoExample 。	2018 年 1 月 9 日

變更	描述	日期
Kinesis Video Streams 文件已發行	這是 Amazon Kinesis Video Streams 開發人員指南的最初版本。	2017 年 11 月 29 日

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。