

STREAMING DE VIDEO A AWS ELEMENTAL
MEDIACONNECT USANDO EL SDK DE ZIXI –
PARTE 2



CONTENTS

INTRODUCCIÓN	3
PRERREQUISITOS	3
COSTO DE APROVISIONAR LOS RECURSOS USADOS EN ESTA PUBLICACIÓN	3
PASO 1: CONFIGURACIÓN DEL CIFRADO Y PERMISOS	4
PASO 2: CREACIÓN DE POLÍTICAS Y USUARIO EN AWS IDENTITY AND ACCESS MANAGEMENT (IAM) PARA AWS ELEMENTAL LIVE	4
PASO 3: CONFIGURACIÓN DEL FLUJO DE ENTRADA Y SALIDA EN AWS ELEMENTAL MEDIACONNECT	4
PASO 4: ELIJA UN ENCODER (AWS ELEMENTAL LIVE O FFMPEG)	5
PASO 5: AWS ELEMENTAL LIVE – CONFIGURACIÓN DE LAS ENTRADAS DEL EVENTO	5
PASO 6: AWS ELEMENTAL LIVE – CONFIGURACIÓN DE LAS SALIDAS DEL EVENTO	6
PASO 7: TRANSMITA CONTENIDO VOD (ARCHIVOS .MP4)	7
PASO 8: FFMPEG – CREACIÓN DEL FLUJO DE AWS ELEMENTAL MEDIACONNECT	7
PASO 9: COMPILE FFMEG PARA SOPORTAR FEC (FORWARD ERROR CORRECTION) E INSTALE FFMPEG	7
STEP 10: INSTALE ZIXI-MEDIACONNECT Y CONFIGURE EL FEEDER	8
PASO 11: TRANSMITA CONTENIDO VOD (ARCHIVOS .MP4)	9
PASO 12: VISUALICE EL CONTENIDO VOD EN UN DISPOSITIVO ANDROID O IOS	10
STEP 13: TERMINAR LOS RECURSOS	11
CONCLUSIONES	11
RECURSOS	11

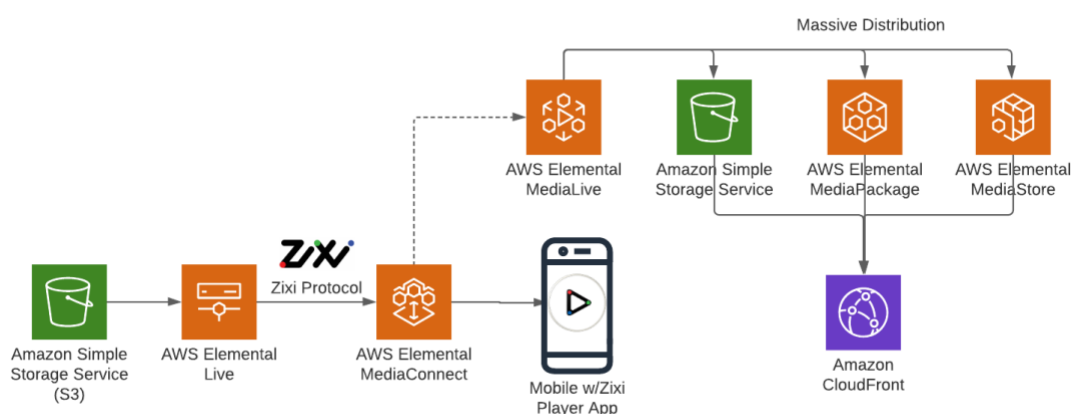
Introducción

El contenido VOD (Video on Demand) se puede retransmitir a dispositivos móviles mediante un encoder local, el servicio AWS Elemental MediaConnect (EMX) y Zixi Player.

En la parte 1 de esta serie de publicaciones, demostramos como instalar y configurar Open Broadcaster Software (OBS) Studio y el SDK de Zixi para contribuir contenido de video de un sistema local hacia AWS Elemental MediaConnect (Zixi-MediaConnect Feeder), crear un flujo de AWS Elemental MediaConnect y retransmitir contenido VOD a dispositivos móviles.

En esta publicación, demostramos como configurar un encoder local (ya sea [AWS Elemental Live](#) o FFmpeg) y el servicio de AWS Elemental MediaConnect para codificar y transmitir contenido VOD desde [Amazon Simple Storage Service \(Amazon S3\)](#) o desde un sistema local a dispositivos móviles.

La descripción de Zixi-MediaConnect Feeder/Receiver, protocolos compatibles, consideraciones de seguridad y latencia así cómo limitaciones están disponibles en la parte 1 de esta publicación.



Prerrequisitos

En esta publicación (Parte 2) se requiere el acceso a un encoder (ya sea el encoder AWS Elemental Live, un encoder de terceros o un encoder de software. Como alcance de esta publicación, se demostrará el uso de AWS Elemental Live y FFmpeg instalado en un sistema local con sistema operativo Ubuntu Focal Fossa).

Costo de aprovisionar los recursos usados en esta publicación

- No existen cargos por utilizar FFmpeg, OBS, Zixi-MediaConnect Feeder/Receiver, ni por el uso de la aplicación Zixi Player
- AWS Elemental MediaConnect se factura por uso, es decir; por hora de uso de cada flujo en ejecución, además de un cargo por gigabyte transferido utilizando a través del flujo, ya sea por un precio reservado de ancho de banda saliente, por hora, basado en el ancho de banda utilizado por el video enviado a internet o bajo demanda. Un flujo que está en modo de espera no es considerado un recurso activo y no incurre en costo, incluso si tiene salidas o derechos configurados.

Revisa <https://aws.amazon.com/mediacconnect/pricing/> para más información.

- [AWS Secrets Manager](#) incluye un periodo de prueba gratuito de 30 días. Después de que termine el plazo, Secrets Manager se factura \$0.40 por secreto al mes. Para secretos que se almacenan por menos de un mes, el precio se prorratea (en función del número de horas). Consulta <https://aws.amazon.com/secrets-manager/pricing/> para más información.
- Para mantener los costos al mínimo, asegúrese de terminar los recursos una vez que termine sus pruebas siguiendo las tareas del Paso 8 de esta publicación.

Paso 1: Configuración del cifrado y permisos

Realizar las tareas del Paso 1 de la Parte 1 de la publicación anterior. Estas tareas crean un secreto, una política y el rol que permite a AWS Elemental MediaConnect conectarse a AWS Secrets Manager.

Paso 2: Creación de políticas y usuario en AWS Identity and Access Management (IAM) para AWS Elemental Live

Ejecute las tareas del paso A de esta [guía de configuración de AWS Elemental Live](#) para crear dos nuevas políticas de IAM y un nuevo usuario de IAM. Nota: ya hemos cubierto las tareas en el paso B (configuración de cifrado) de la guía de configuración de AWS Elemental Live. Los pasos C y D (configuración de flujo de entrada y salida MediaConnect) se cubren a continuación.

Paso 3: Configuración del flujo de entrada y salida en AWS Elemental MediaConnect

- Ejecute las tareas del paso 2 de la Parte 1 [«Link Paso 2 en Parte 1 pdf»](#) de la publicación, usando los siguientes valores:

Source type: Standard source
Name: feed-from-elemental-live
Protocol: Zixi push
Inbound port: 2088 (default)
Source description: feed from AWS Elemental Live
Stream ID: 1

Source

Source type [Info](#)

Standard source
 Choose this option if your content comes from a non-entitlement based source, such as an on-premises encoder.

Entitled source
 Choose this option if your content comes from an AWS Elemental MediaConnect flow that is owned by another AWS account.

Name

The source name can have up to 64 characters. Valid characters: A-Z, a-z, and 0-9

Protocol

Inbound port [Info](#)

The source must use port 2088 if the protocol is set to Zixi.

Whitelist CIDR block [Info](#)

Sample format: 10.24.34.0/23

Source description
 The description can help you distinguish one flow from another.

Maximum latency - optional [Info](#)
 The size of the buffer (in ms) that you want.

Valid range: 0-60,000. Valid characters: 0-9

Stream ID - optional [Info](#)

- b. Ejecute los pasos restantes del paso 2 de la Parte 1 <<Link Paso 2 en Part 1 pdf>> de la publicación para configurar el descifrado y el flujo de salida, e iniciar el flujo de AWS Elemental MediaConnect.

Paso 4: Elija un encoder (AWS Elemental Live o FFmpeg)

Debe elegir si desea utilizar AWS Elemental Live o FFmpeg como su encoder. Si utiliza AWS Elemental Live como su encoder, continúe con el Paso 5. Si prefiere utilizar FFmpeg como encoder, vaya directamente al Paso 8.

Paso 5: AWS Elemental Live – configuración de las entradas del evento

Vaya al Paso 8 si no usa AWS Elemental Live

- a. Consiga dos o más archivos .mp4 files en esta demo y súbalos a un bucket de Amazon S3 (puede ser nuevo o ya existente) dentro de su cuenta de AWS.
- b. Diríjase a la consola de AWS Elemental Live encoder.

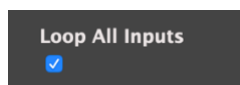
El primer paso es configurar un nuevo evento en vivo (Live Event).

- c. Haga click en **Event Control**, posteriormente haga click en **Create**
- d. Complete los valores como se muestran a continuación:
 - Name:** live-event-blog-part2
 - Priority:** <default>
 - Restart on Failure:** <default>
 - Input 1:** choose **File Input**
 - Name:** Seleccione cualquier nombre – este nombre servirá para identificar el archivo.
 - File Location:** Escriba o copie la URI del archivo cargado en Amazon S3, asegúrese de que tenga un formato similar al siguiente: s3://<nombre del bucket>/<nombre del archivo>

- e. Haga click en el icono del candado para seleccionar las credenciales de AWS IAM
 - User name:** Copie la AWS access key ID que se generó en el Paso 2
 - Password:** Copie el Secret access key que se generó en el Paso 2

- f. Haga click en **Add Input**

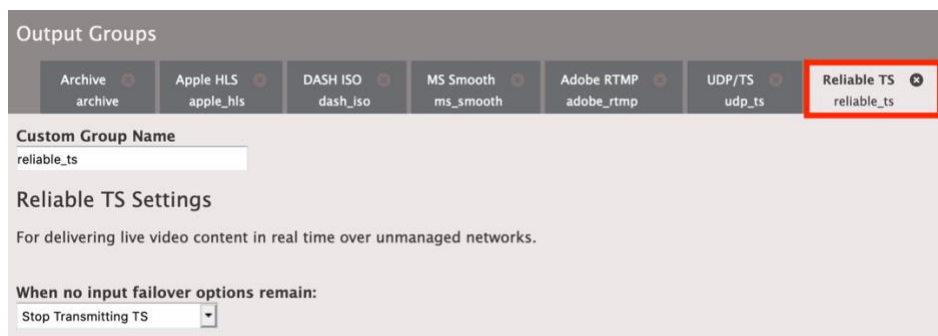
- g. Configure el archivo de entrada, (nombre), Ubicación del archivo (Ruta Amazon S3), y las credenciales para el resto de los archivos .mp4 en Amazon S3
- h. Cuando termine de crear la última entrada, active la casilla **Loop All Inputs**.



Paso 6: AWS Elemental Live – configuración de las salidas del evento

Vaya al Paso 8 si no está usando AWS Elemental Live

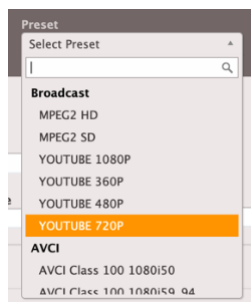
- a. Diríjase hacia abajo hasta llegar a **Output Groups** y seleccione **Reliable TS**
 Complete los valores como se muestra a continuación:
Custom Group Name: reliable_ts (default)
When no input failover options remain: Stop Transmitting TS (default)



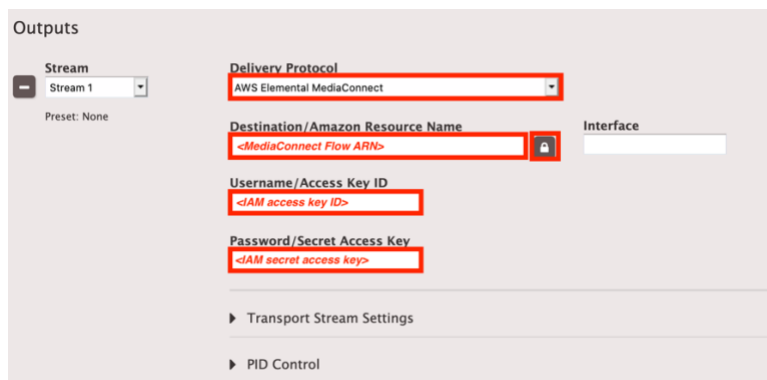
- b. Haga click en **Add Output**.



- c. En la opción de **Preset**, seleccione **YOUTUBE 720P**



- d. Complete los valores como se muestra a continuación:
Delivery Protocol: AWS Elemental MediaConnect
Destination: Copie el ARN del flujo de AWS Elemental MediaConnect del Paso 3
User name: Copie la IAM access key ID del Paso 2
Password: Copie el IAM secret access key del Paso 2



e. Haga click en **Create**

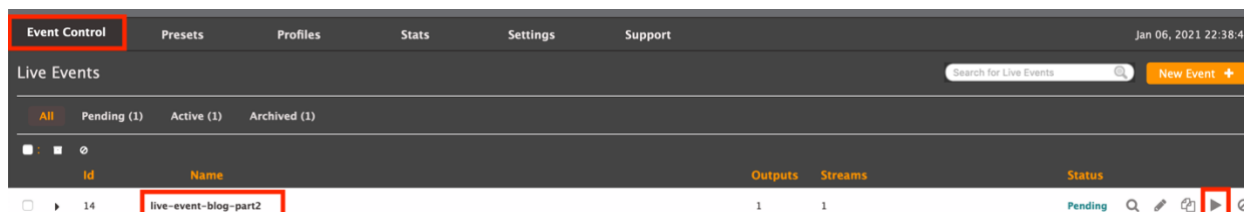


Paso 7: Transmite contenido VOD (archivos .mp4)

Diríjase al Paso 8 si no está usando AWS Elemental Live

Comencemos a transmitir nuestro contenido VOD desde AWS Elemental Live hacia AWS Elemental MediaConnect.

- a. En AWS Elemental Live, haga click en Event Control, localice el evento recién creado, y haga click en el **icono de la flecha** (Play) que se encuentra a la derecha de nuestro evento.



En este paso concluimos la configuración de nuestro encoder usando AWS Elemental Live. AWS Elemental Live ahora transmitirá los archivos .mp4 (de forma continua y cifrada) hacia AWS Elemental MediaConnect.

- b. Salte los Pasos del 8-11 y diríjase al Paso 12 para visualizar la salida en su dispositivo móvil.

Paso 8: FFMpeg – Creación del flujo de AWS Elemental MediaConnect

Diríjase al Paso 12 si no está usando FFMpeg

FFMpeg transmitirá hacia AWS Elemental MediaConnect. Para crear un flujo a AWS Elemental MediaConnect, ejecute las tareas del Paso 2 de la Parte 1 de esta publicación

Paso 9: Compile FFMpeg para soportar FEC (forward error correction) e instale FFMpeg

Diríjase al Paso 12 si no está usando FFMpeg

- a. En el sistema Ubuntu abra una terminal.

FFmpeg es software open-source distribuido bajo la licencia [LGPL2.1](#).

- b. Si tiene una versión de FFmpeg instalada, desinstálela usando el siguiente comando:

[Gnu Bash]

```
$ sudo apt-get remove ffmpeg
```

- c. Ahora debe compilar FFmpeg para habilitar FEC:

[Gnu Bash]

```
$ cd ~
$ sudo apt -y update
$ sudo apt -y upgrade
$ wget https://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2
$ tar -jxvf ffmpeg-snapshot.tar.bz2
$ cd ffmpeg/
$ sudo apt -y install autoconf automake build-essential cmake git-core libass-
dev libfreetype6-dev libgnutls28-dev libsdl2-dev libtool libva-dev libvdpau-dev libvorbis-
dev libxcb1-dev libxcb-shm0-dev libxcb-xfixes0-dev pkg-config texinfo wget yasm zlib1g-
dev nasm libx264-dev libx265-dev libnuma-dev libvpx-dev libfdk-aac-dev libmp3lame-
dev libopus-dev
$ sudo ./configure --pkg-config-flags="--static" --extra-libs="-lpthread -lm" --enable-gpl --enable-
gnutls --disable-libaom --enable-libass --enable-libfdk-aac --enable-libfreetype --enable-
libmp3lame --enable-libopus --disable-libsvtav1 --enable-libvorbis --enable-libvpx --enable-
libx264 --enable-libx265 --enable-nonfree
$ sudo make
```

- d. Instale FFmpeg:

[Gnu Bash]

```
$ sudo make install
```

- e. Valide que la instalación fue satisfactoria:

[Gnu Bash]

```
$ ffmpeg -version
```

El resultado de este comando le mostrará la información de la versión instalada.

Step 10: Instale Zixi-MediaConnect y configure el Feeder

Vaya al Paso 12 si no está usando FFMpeg

- a. Ejecute las tareas del Paso 3 de la Parte 1 de la publicación para obtener una licencia de Zixi-MediaConnect Feeder.

- b. Ejecute las tareas del Paso 5 (a-i) de la Parte 1 de la publicación para descargar, instalar, activar la licencia, y configurar Zixi-MediaConnect. Especifique el **Stream ID**: stream-ffmpeg-to-zixi.

- c. Continúe ejecutando las tareas del Paso 5 (j-m) de la Parte 1 de la publicación para configurar la salida en Zixi, especificando **Input Stream**: stream-ffmpeg-to-zixi

Paso 11: Transmite contenido VOD (archivos .mp4)

Diríjase al Paso 12 si no está usando FFMpeg

- a. Utilice dos o más archivos .mp4 files para probar esta demo y guárdelos en el directorio home de su sistema Ubuntu.

- b. Abra una terminal y cree un Shell script para reproducir cada video de forma continua en un loop:

[Gnu Bash]

```
$ vim ~/stream-videos-loop.sh
```

- c. Copie el siguiente script en el archivo en el archivo que acaba de abrir y reemplace el nombre de los archivos mp4 con el nombre de sus archivos:

[plain text]

```
#!/bin/bash
```

```
while true ; do
  for video in <filename-video-1>.mp4 <filename-video-2>.mp4 <filename-video-3>.mp4 ; do
    ffmpeg -re -i $video -c copy -map 0 -f rtp_mpegts -fec prompeg=l=5:d=20 rtp://127.0.0.1:5000
    test $? -gt 128 && exit
  done
done
```

- d. Cambie los permisos y ejecute el Shell script:

[Gnu Bash]

```
$ chmod +x stream-videos-loop.sh
$ ./stream-videos-loop.sh
```

- e. Deje el script en ejecución hasta que finalice las pruebas en su dispositivo móvil. Concluidas sus pruebas, detenga la ejecución del script presionando, **Ctrl-C** en la terminal.
- f. En Zixi-MediaConnect, haga click en Inputs y verifique que el Input de Zixi aparezca como Conectado a ffmpeg.

Zixi-MediaConnect enviará la transmisión cifrada a AWS Elemental MediaConnect.

- g. Continúe con el Paso 12 para visualizar el contenido VOD en su dispositivo móvil.

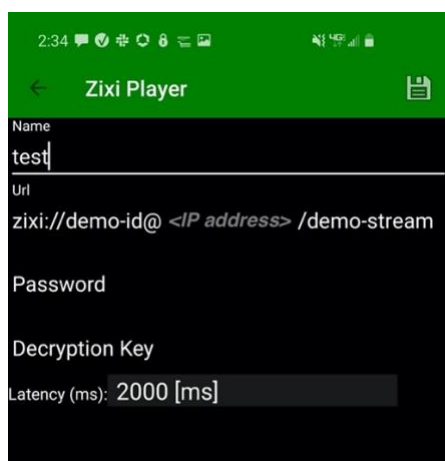
Paso 12: Visualice el contenido VOD en un dispositivo Android o iOS

- a. En su dispositivo móvil, instale la app de Zixi Player, disponible en <https://android-apk.org/com.zixi.player/35756756-zixiplayer/> o en la App Store de iOS.

Configure Zixi Player para recibir la transmisión de AWS Elemental MediaConnect.

- b. Abra la app Zixi Player en el dispositivo móvil y llene los campos con los siguientes valores:
Name: test
URL: zixi://demo-id@<AWS Elemental MediaConnect Public Outbound IP address>/demo-stream
Password: <blank>
Decryption key: <blank>

Latency: (default)



- c. Presione **Save (icono de disco)** en la parte superior.
- d. Presione el nombre de la configuración recién creada para reproducir la transmisión.

Step 13: Terminar los recursos

- a. Si usó el encoder FFmpeg, presione **Ctrl-C** en la terminal de Ubuntu para detener el script stream-videos.sh que hace la transmisión.
- b. Si usó el encoder de AWS Elemental Live, **Detenga** el evento en vivo: “live-event-blog-part2”.
- c. En la consola de [AWS Elemental MediaConnect](#), **Detenga** el flujo “demo-part2” y luego **Bórralo**.
- d. En la consola de [AWS Secrets Manager](#), seleccione el secreto “demo-MediaConnect-AES256” haga click en **Actions, Delete Secret**. Seleccione 7-day waiting period (periodo mínimo) y haga click en **Schedule Deletion**.

Conclusiones

En esta publicación utilizó el encoder de AWS Elemental Live o de FFmpeg en un equipo local con sistema operativo Ubuntu Focal Fossa para leer contenido VOD y retransmitirlo de forma segura a AWS Elemental MediaConnect. El contenido VOD fue visualizado desde la app de Zixi Player en su dispositivo móvil. En la Parte 3 de esta publicación, exploraremos la transmisión a Zixi-MediaConnect Receiver y los flujos de trabajo OTT (Over the top).

Recursos

- [AWS Elemental MediaConnect Source Failover](#)
- [AWS Elemental MediaConnect Zixi portal](#)
- [Services and tools for monitoring video workflows on AWS](#)
- [The evolution of the television live event ecosystem and how AWS and TAGVS are shaping the future](#)
- [Zen master and AWS Elemental MediaConnect. Scale technical operations with global visibility](#)
- [AWS Elemental MediaConnect user guide](#)
- [Content Syndication using AWS Elemental MediaConnect](#)