

STREAMING VIDEO TO AWS ELEMENTAL
MEDIACONNECT USING ZIXI SDK – PART 3



CONTENTS

INTRODUCTION	3
PRE-REQUISITES.....	3
COST OF PROVISIONING RESOURCES USED IN THIS POST.....	3
STEP 1: CONFIGURE ENCRYPTION AND PERMISSIONS.....	4
STEP 2: OBTAIN ZIXI-MEDIACONNECT RECEIVER LICENSE (FREE).....	4
STEP 3: CONFIGURE ZIXI-MEDIACONNECT RECEIVER.....	4
STEP 4: CONFIGURE MEDIACONNECT OUTPUT AND ZIXI-MEDIACONNECT INPUT.....	5
OPTION 1: ADD A ZIXI PUSH STREAM.....	5
OPTION 2 -- ADD A ZIXI PULL STREAM.....	7
STEP 5: CONFIGURE ZIXI-MEDIACONNECT OUTPUT AND TEST	9
STEP 6: CONFIGURE AN AWS ELEMENTAL MEDIALIVE INPUT AND OPTIONALLY ADD OTT WORKFLOW	11
STEP 7: CLEAN UP	12
CONCLUSION.....	12
RESOURCES	13

Introduction

In [Part 1](#) of this 3-part series, we demonstrated how to install and configure Open Broadcaster Software (OBS) Studio and Zixi SDK for [AWS Elemental MediaConnect](#) (Zixi-MediaConnect) Feeder on a local system, create a MediaConnect flow, and stream a local VOD asset to end-user mobile devices. In [Part 2](#), we demonstrated how to configure a local encoder (either [AWS Elemental Live](#) or FFmpeg) and MediaConnect to encode and stream [Amazon Simple Storage Service \(Amazon S3\)](#) or local VOD assets to end-user mobile devices.

In this document, we explore streaming to Zixi-MediaConnect Receiver and OTT workflows.

Zixi-MediaConnect Receiver resides at the point of distribution. With Zixi-MediaConnect Receiver, you can receive streams from MediaConnect flow to your on-premises or cloud systems, generate UDP multicast/unicast or TCP outputs, and integrate the streams to your lineup channels. You can also use professional Integrated Receiver/Decoders (IRDs) that support the Zixi protocol natively.

A description of Zixi-MediaConnect Feeder/Receiver, supported protocols, latency considerations, and limitations are available in [Part 1](#) of this blog.

The following protocols are supported:

Inputs:

- Push – the stream is pushed from MediaConnect to Zixi-MediaConnect Receiver.
- Pull – the stream is pulled by the Zixi-MediaConnect Receiver from MediaConnect

Outputs:

- UDP
- TCP

Pre-requisites

- This post assumes that you already have a MediaConnect Flow configured, you can follow [Part 1](#) or [Part 2](#) of this blog series to configure a MediaConnect flow.
- External firewall that allows communication on UDP ports 2088 and 2077 (inbound and outbound).
- Ubuntu Linux system (this post was tested with Ubuntu Focal Fossa).

Cost of provisioning resources used in this post

- There is no charge for Zixi-MediaConnect Feeder/Receiver.
- MediaConnect is billed at an hourly rate for each running flow, plus either a per-gigabyte charge for data transferred using the flow or, with [reserved outbound bandwidth pricing](#), a per-hour charge based on bandwidth for video sent to the internet. A flow that is in standby mode is not an active resource, and does not incur active resource costs even if it has outputs or entitlements configured. See <https://aws.amazon.com/mediaconnect/pricing/> for more information.
- AWS Secrets Manager comes with a 30-day free trial. After the trial is over, Secrets Manager is billed \$0.40 per secret per month. For secrets that are stored for less than a month, the price is prorated (based on the number of hours). See <https://aws.amazon.com/secrets-manager/pricing/> for more information.

- To keep charges to a minimum, be sure to clean up resources once you are finished testing by following the tasks in Step 6 of this post.

Step 1: Configure encryption and permissions

Follow Step 1 from [Part 1](#) to setup AES-256 encryption and MediaConnect permissions.

Step 2: Obtain Zixi-MediaConnect Receiver license (free)

- In a web browser, navigate to <https://portal.zixi.com/self-registration/MEDIACONNECT> (create an account).
- Follow instructions in confirmation email to set your password.
- Navigate to <https://portal.zixi.com> and login.
- Click **My Settings**.
- Enable subscription to MediaConnect.

You can subscribe to the following package(s)



AWS MediaConnect

This package grants access to installation files and documentation related to AWS MediaConnect. It also gives you the ability to generate license key(s) needed to activate a Zixi MediaConnect Feeder/Receiver, once installed.

To generate a key, see the "License Keys" section of this application.

- Click **License Keys**.
- Click **Generate a Key** for MediaConnect Receiver.

License Keys

License Keys You Can Generate



Generate a Key AWS MediaConnect : MediaConnect Receiver Key

You don't have any keys.

MediaConnect Receiver license key is displayed.

Step 3: Configure Zixi-MediaConnect Receiver

Download, install, and configure Zixi-MediaConnect

- Follow Step 5 from [Part 1](#) to download, install, and configure Zixi-MediaConnect on your Ubuntu system.
- To license the Zixi-MediaConnect Receiver, click **Status, Options**.



- Click **Activate License** in the pull-down menu.
- Paste the MediaConnect Receiver license key from Step 2 and click **Activate**.

Zixi-MediaConnect is now licensed as a Receiver.

Step 4: Configure MediaConnect Output and Zixi-MediaConnect Input

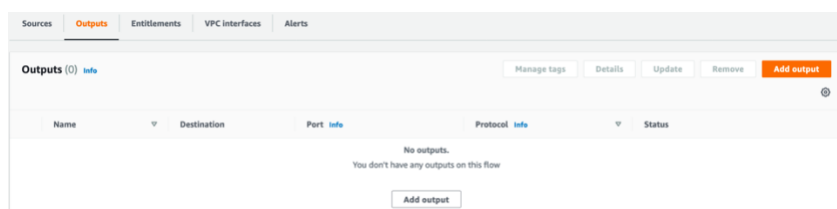
Configure the MediaConnect Output to the Zixi-MediaConnect Receiver

We explore two options:

- Zixi push (the stream is pushed from MediaConnect to Zixi-MediaConnect Receiver).
- Zixi pull (the stream is pulled from MediaConnect by the Zixi-MediaConnect Receiver).

Option 1: Add a Zixi push stream

a. In the MediaConnect console, select an existing flow, then click **Outputs, Add Output**.



b. Fill in the values as shown:

Name: demo-output

Output type: Standard source

Description: output to be consumed by Zixi Receiver

Protocol: Zixi push

Stream ID: demo-stream

Maximum latency: 2000

IP address: <Public IP of the Zixi Receiver host>

Port: 2088

Add output
✕

Name

The output name must be unique within the current flow. The output name can have up to 64 characters. Valid characters: A-Z, a-z, and 0-9

Output type

Description

Protocol

Stream ID - optional
The stream ID that is set in the Zixi receiver.

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

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

IP address

Sample format: 192.0.2.0

Port

- c. Under **Encryption**, check the **Enable** box
Encryption Type: Static key
Encryption algorithm: aes256
Role ARN: click the pull-down arrow. The ARN for the role you created in Step 1 should be displayed.
Secret ARN: click the pull-down arrow. The ARN for the secret that you created in Step 1 should be displayed.

Encryption

If you want to add a layer of protection to your output, encrypt the output here. You must store the encryption key in AWS Secrets Manager before you can complete this section.

Enable

Encryption type: Static key Encryption algorithm: aes256

Role ARN
 The ARN of the IAM role that you created when you set up AWS Elemental MediaConnect as a trusted entity.
 List of ARNs: arn:aws:iam::<Account ID>:role/demo-role-MediaConnect

Secret ARN
 The ARN that Secrets Manager assigned when you created the secret to store the encryption key.
 List of ARNs: arn:aws:secretsmanager:us-east-1:<Account ID>:secret:de

Cancel **Add output**

- d. Click **Add output**.
- e. Click **Start** button of the flow.
- f. On the Ubuntu system, in the Zixi-MediaConnect Receiver **Inputs** page, click **New Input**.



- g. The **Add Input** window is displayed:

Add new input stream ✕

Stream ID:

Push Pull

Stream parameters

Password:

Latency [ms]: Remote configuration

Decrypt stream

*Automatic encryption does not require manual settings.

Decryption type: AES 256

Decryption key:

Recording parameters

Ok Cancel

Fill in the values as shown:

Stream ID: stream-option1 (a unique internal name for the stream)

Select **Push**

Password: <leave blank>

Latency [ms]: Remote configuration (default - use the latency set in MediaConnect)

Check the **Decrypt Stream** box

Decryption type: AES 256

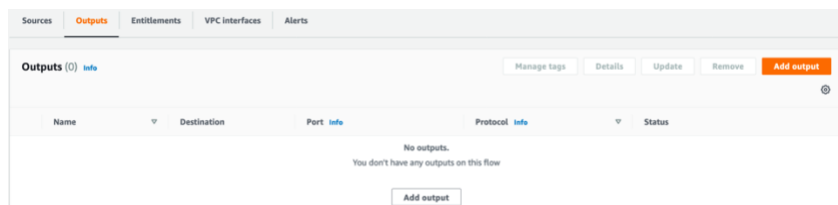
Decryption key: paste the AES 256 hexadecimal key that you saved in Secrets Manager in Step 1.

h. Click **OK**.

The “Input added” message appears on the top of the screen. The Push Stream now appears in the list of Input streams available.

Option 2: Add a Zixi pull stream

a. In the MediaConnect console, select an existing flow, then click **Outputs, Add Output**.



b. Fill in the values as shown:

Name: demo-output

Output type: Standard output

Description: output to be consumed by Zixi Receiver

Protocol: Zixi pull

Stream ID: demo-stream

Remote ID: demo-id

Maximum latency: 2000

CIDR allow list: <Zixi-MediaConnect Receiver public IP address> or use 0.0.0.0/0 if you are using a non-static IP address

Add output

Name

The output name must be unique within the current flow. The output name can have up to 64 characters. Valid characters: A-Z, a-

Output type

Description

Protocol

Stream ID

The stream ID that is set in the Zixi receiver.

Remote ID

The identifier that is assigned to the receiver.

Maximum latency - *optional*

The size of the buffer (in ms) that you want.

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

CIDR allow list

Sample format: 10.24.34.0/23

- c. Under **Encryption**, check the **Enable** box
Encryption Type: Static key
Encryption algorithm: aes256
Role ARN: click the pull-down arrow. The ARN for the role you created in Step 1 should be displayed.
Secret ARN: click the pull-down arrow. The ARN for the secret that you created in Step 1 should be displayed.

Encryption

If you want to add a layer of protection to your output, encrypt the output here. You must store the encryption key in AWS Secrets Manager before you can complete this section.

Enable

Encryption type: Static key Encryption algorithm: aes256

Role ARN
 The ARN of the IAM role that you created when you set up AWS Elemental MediaConnect as a trusted entity.
 List of ARNs: arn:aws:iam::<Account ID>:role/demo-role-MediaConnect

Secret ARN
 The ARN that Secrets Manager assigned when you created the secret to store the encryption key.
 List of ARNs: arn:aws:secretsmanager:us-east-1:<Account ID>:secret:de

Cancel **Add output**

- d. Click **Add output**.
- e. Click **Start** button of the flow.
- f. On the Ubuntu system, in the Zixi-MediaConnect Receiver **Inputs** page, click **New Input**.



- g. The **Add Input** window is displayed:

Add new input stream

Stream ID:

Push Pull

Stream parameters

Host: +

Port:

Stream:

Password:

Latency [ms]:

Ignore TLS certificate:

Bind to IP:

Decrypt stream
 *Automatic encryption does not require manual settings.

Decryption type:

Decryption key:

Recording parameters

Ok Cancel

Fill in the values as shown:

Stream ID: stream-option2 (a unique internal name for the stream)

Select **Pull**

Host: <AWS Elemental MediaConnect Public Outbound IP address>

Port: 2077

Stream: demo-stream (this is the AWS Elemental MediaConnect **Stream ID** configured on the zixi pull output)

Password: <leave blank>

Latency [ms]: 6000 (default)

Ignore TLS certificate: unchecked (default)

Bind to IP: Any (default)

Check the **Decrypt Stream** box

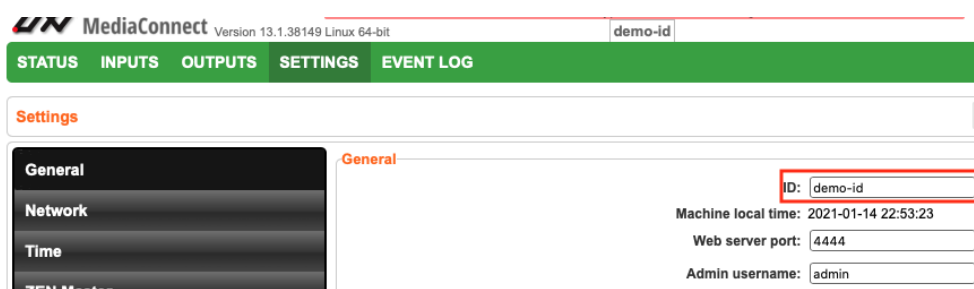
Decryption type: AES 256

Decryption key: paste the AES 256 hexadecimal key that you saved in Secrets Manager in Step 1.

- h. Click **OK**.

The “Input added” message appears on the top of the screen. The Pull Input is added and now appears in the list of streams available in the inputs screen.

- i. In the Zixi-MediaConnect **Settings** menu, click **General** and modify your **ID** to be **demo-id** (this matches the MediaConnect Remote ID configured on the Zixi pull output).



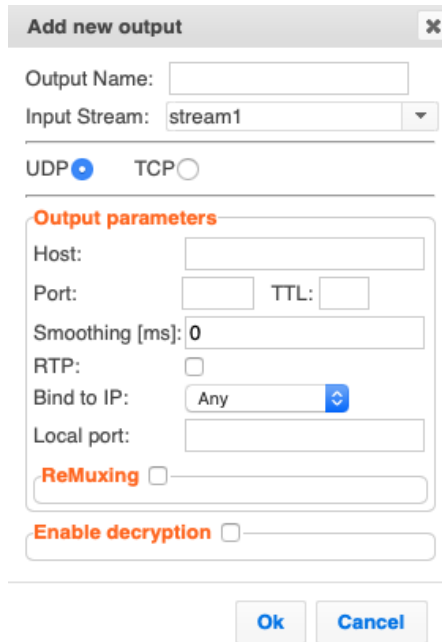
Step 5: Configure Zixi-MediaConnect Output and test

Add Output in Zixi-MediaConnect Receiver

- a. In the **Outputs** page of Zixi-MediaConnect Receiver, click **New Output**.



- b. The **Add Output** window is displayed.



Fill in the values as shown:

Output Name: output1 (a unique internal name for the stream)

Input Stream: Select the Input, which was previously configured as Input

Select **UDP**

Host: 224.1.1.1 (Use this IP or specify another multi-cast destination IP address)

Port: 1234 (Use this port or specify another target UDP port)

TTL: <leave blank> (Time-to-live (TTL) is used mainly to limit the range (or “scope”) of a multicast transmission. Leaving it as blank uses the default value specified by the operating system)

Smoothing [ms]: 100 (Enables transmission of the output at the correct rate. Required when the receiving device is sensitive and cannot lock on to the stream)

RTP: Uncheck (Enables use of the Real-time Transport Protocol)

Bind to IP: Any (default)

Local Port: <leave blank>

ReMuxing: You may want to remux the stream to use Constant Bitrate (CBR). This is used with professional IRDs in order to avoid bandwidth waste. If so, check the box and configure the respective parameters.

Enable Decryption: leave unchecked

- c. Click **OK**.

The New Output with its new settings takes effect.

- d. Open a terminal window and install VLC on the Ubuntu system to test the UDP output stream.

```
[Gnu Bash]
sudo apt install vlc
```

- e. Open the VLC app. Click **Media, Open Network Stream**, and specify the network URL: `udp://@Host:Port` (example `udp://@224.1.1.1:1234`). Click **Play**.



You can now distribute your stream using the AWS Cloud and go global from one stream to multiple destinations.

Step 6: Configure an AWS Elemental MediaLive input and optionally add OTT workflow

Configure the MediaConnect flow as a MediaLive input

- a. In the MediaLive Console, click **Inputs**, then **Create Input**. Select the **MediaConnect** input type.

Input details

Input name – *required*

Input type – *required*

- RTP
Push your source to fixed endpoints with the real-time transport protocol.
- RTMP (push)
Push your source to fixed endpoints with the real-time messaging protocol.
- RTMP (pull)
Pull your source from external endpoints with the real-time messaging protocol.
- HLS
Pull your source from external endpoints with the HTTP protocol.
- MP4
Ingest file content from an MP4 file that is on the public internet
- MediaConnect**
Ingest streaming content from one or two MediaConnect flows
- AWS CDI
Push your uncompressed video over AWS Cloud Digital Interface.
- Elemental Link
Ingest streaming content from an AWS Elemental Link device

- b. Select either the STANDARD_INPUT or SINGLE_INPUT **Input class** and select your MediaConnect flow ARN from the Flow pull-down (Note: STANDARD_INPUT is a redundant

configuration workflow and requires two flows on the same AWS Region).

MediaConnect flows

The unique ARN of the MediaConnect flow being used as a source.

Input class

For a standard channel, you must choose STANDARD_INPUT. For a single-pipeline channel, you or STANDARD_INPUT.

STANDARD_INPUT ▼

Flow A ARN

us-east-1b ▼

Flow B ARN

us-east-1b ▼

- c. MediaConnect automatically generates a MediaLive output, visible in the MediaConnect console flow **Outputs**.

Name	Destination	Port	Protocol	Status
Pull1	-	-	zixi-pull	-
MediaLive-13a80d4a534f	-	-	-	-

Optionally, you can create an OTT workflow following this [Workshop](#) or send the stream to social media following this [blog](#).

Step 7: Clean up

Once you finish testing, clean up your environment

- In the [MediaConnect console](#), **Stop** and **Delete** the flow.
- If you performed the OTT workflow tasks in Step 6, navigate to the MediaLive console and **Delete** the Input that you created.
- In the [AWS Secrets Manager console](#), select the demo-MediaConnect-AES256 secret and click **Actions, Delete Secret**. Choose a 7-day waiting period (this is the minimum) and click **Schedule Deletion**.

Conclusion

In this post, we installed and licensed Zixi-MediaConnect Receiver on an Ubuntu system, created two types of encrypted MediaConnect outputs (push and pull), and configured Zixi-MediaConnect Receiver to receive and decrypt the streams.

This 3-part blog explored ways to bring on-premises video contribution to the AWS Cloud using MediaConnect with the Zixi protocol for secure, reliable, and scalable streaming. We have many options for professional encoders, with or without the integration of the Zixi protocol. Using Zixi-MediaConnect, we can broadcast by encapsulating a traditional protocol, such as UDP, in the Zixi protocol. Zixi-MediaConnect also allows us to receive the Zixi protocol and encapsulate it in a UDP

transmission protocol. The capabilities of having the signals in the cloud allow you to distribute these signals globally and to [other AWS accounts](#) without the need for a dedicated infrastructure or specialized devices.

Resources

- [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](#)