aws elemental

OBS STUDIO TO AWS ELEMENTAL MEDIALIVE TO AWS ELEMENTAL MEDIAPACKAGE

Workflow Example

CONTENTS

INTRODUCTION

This workflow example illustrates how to use a workstation running OBS Studio to send a high definition (HD) feed to AWS Elemental MediaLive, where we will encode an ABR stream set using an HLS output group and send the content to AWS Elemental MediaPackage.

In the text below the OBS Studio workstation is referred to as “the appliance.”

Warning: This workflow is intended as an example and for testing purposes only. AWS Elemental MediaLive requires two inputs as part of its redundancy model. OBS Studio does not have a provision to send its output to multiple destinations at the same time. As such, OBS Studio can’t provide redundancy and you should use caution running this example in a production workflow.

Note: To use this workflow in production, it is highly recommended you use the AWS Elemental MediaPackage endpoint as an origin for a CDN such as Amazon CloudFront. The AWS Elemental MediaPackage console includes an option to create a CloudFront distribution during channel creation.

REQUIREMENTS

To perform this procedure, you must be familiar with the configuration of the OBS Studio software. You also must have all of the required information for your particular source, excluding the configuration of the streaming settings directed towards AWS Elemental MediaLive.

ORDER OF WORK

1. Obtain needed information.
2. Create two channels in AWS Elemental MediaPackage.
3. Create an input in AWS Elemental MediaLive.
4. Configure the OBS Studio software (“the appliance”).
5. Create a channel in AWS Elemental MediaLive.
6. Start the video stream.

PREREQUISITE: OBTAIN NEEDED INFORMATION

Obtain the public IP address (or addresses) from the appliance that you will use to send the feed to the AWS Elemental MediaLive input.

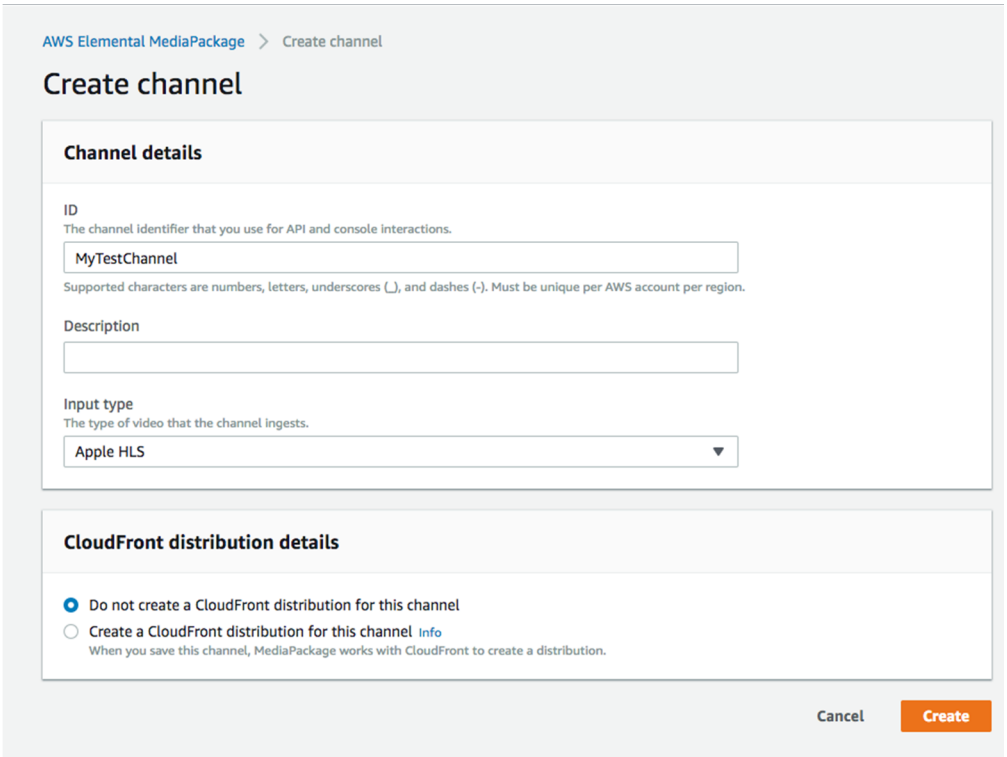
Note: If there is a firewall between the appliance and the internet (highly recommended), the public IP addresses will likely be different from those that the appliance reports. If so, you need to determine the external address being used. The appliance network may also be configured to utilize a pool of external IP addresses. In this case, you will need the CIDR range for the entire pool to include in the Input Security Group.

STEP A: CREATE A CHANNEL IN AWS ELEMENTAL MEDIAPACKAGE

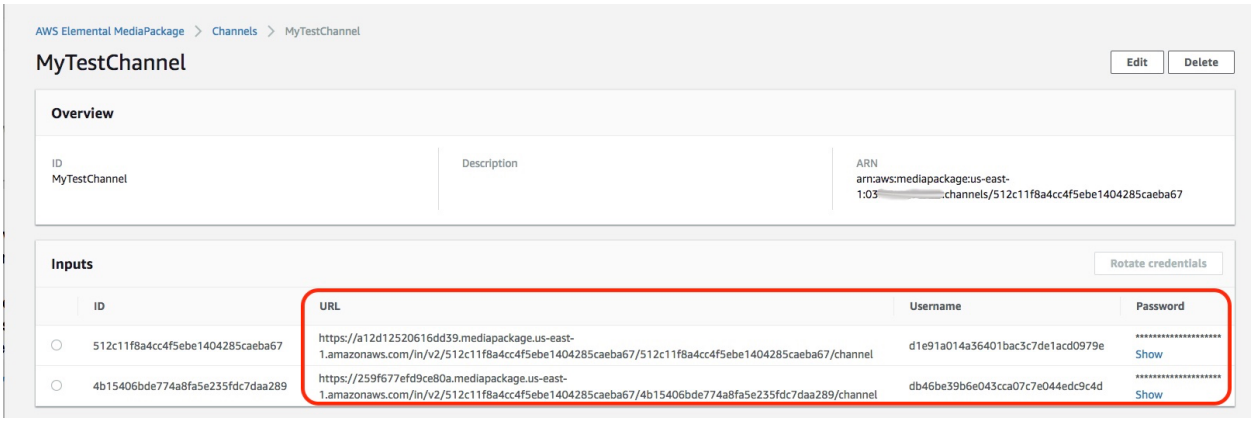
In order to create your AWS Elemental MediaLive channel, you must first know the destination URLs and credentials for your output(s). For this example, use AWS Elemental MediaPackage as your destination.

MediaLive requires two output destinations per output group, and each AWS Elemental MediaPackage channel has two inputs.

1. Log into the AWS Elemental MediaPackage console for the same region where you will be using AWS Elemental MediaLive.
2. If you have previously created channels in MediaPackage, the channel listing view will appear. Otherwise the introductory landing page will appear.
    a. From the landing page, enter a channel name and choose **Next Step**.
    b. From the Channel Listing page, choose **Create Channel**.
3. For either case above, you should now see the Create channel page:

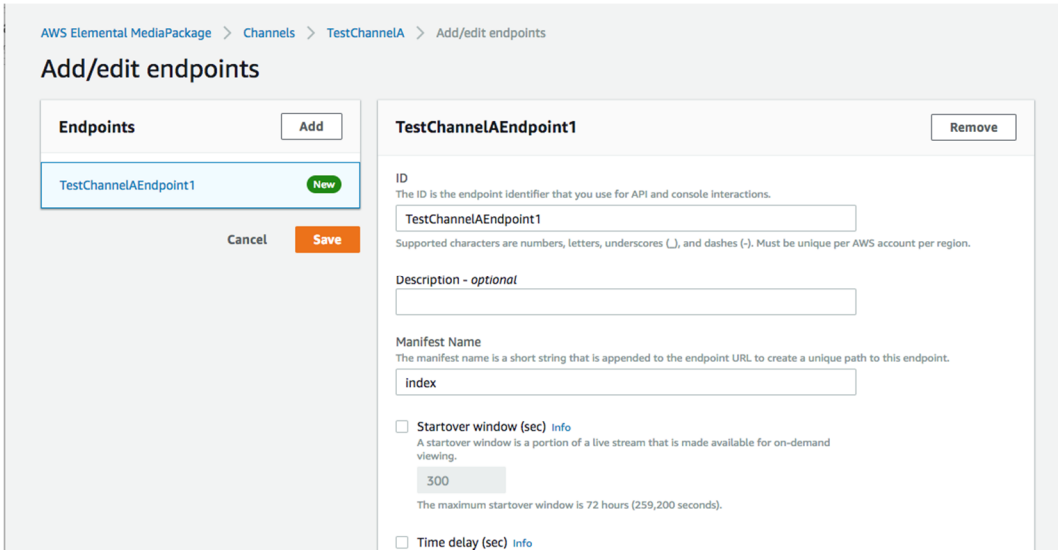


4. Add a description if desired. Note that there is an option to create a CloudFront distribution to work with this channel. For production workloads it is important to place a content distribution network (CDN) in front of the MediaPackage endpoints. Choose **Create** to save and create the channel. The channel detail page appears.



Make a note of the **Input URL**, **Username**, and **Password** for each of the two inputs (use the **show** button to reveal the password) as you will need to use these values when creating your AWS Elemental MediaLive channel.

5. Just below the channel detail tile choose **Add endpoints** to create an appropriate endpoint to be able to view your channel. For this example it is sufficient to create a simple HLS endpoint so just give it a unique name in the **ID** field and choose **Save** to create the endpoint.



When the MediaLive channel is up and running you will be able to point an HLS compatible player or browser at the endpoint to view the channel, or you can preview it from inside the MediaPackage console.

6. Keep this browser session active so you can easily come back later to check your channel.

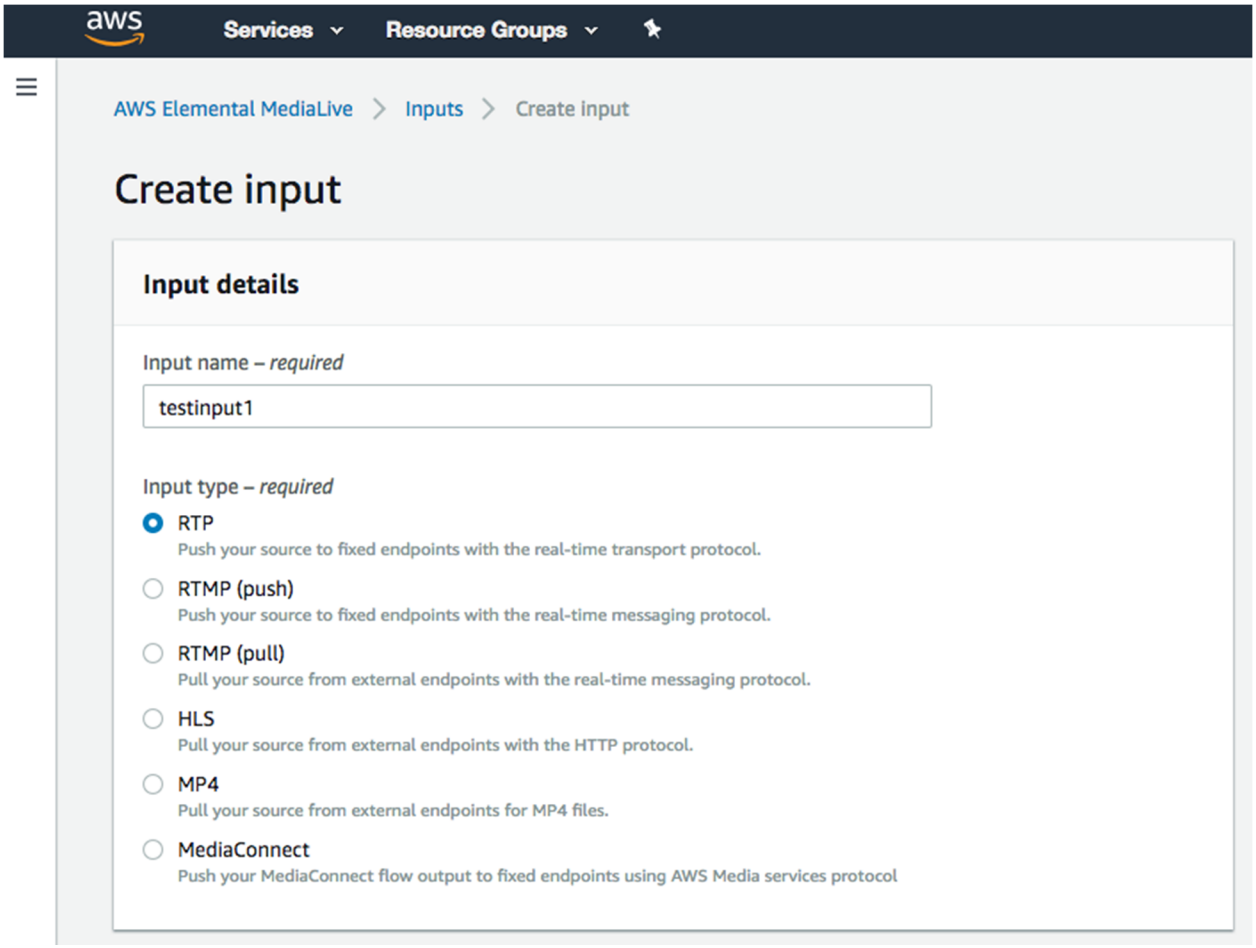
STEP B: SET UP INPUTS IN AWS ELEMENTAL MEDIALIVE

1. In a new browser tab or window, log into the AWS Elemental MediaLive console for the same region you just used to create your AWS Elemental MediaPackage channels and endpoints.
2. Take the appropriate action:

- If the standard service page appears, choose **Inputs** from the navigation panel on the left side.
- If the service landing page appears, expand the left-hand menu by choosing the three horizontal lines near the top just below the AWS icon. Choose **Inputs**.

The Input listing page appears.

3. Choose **Create input**. The Create input page appears.

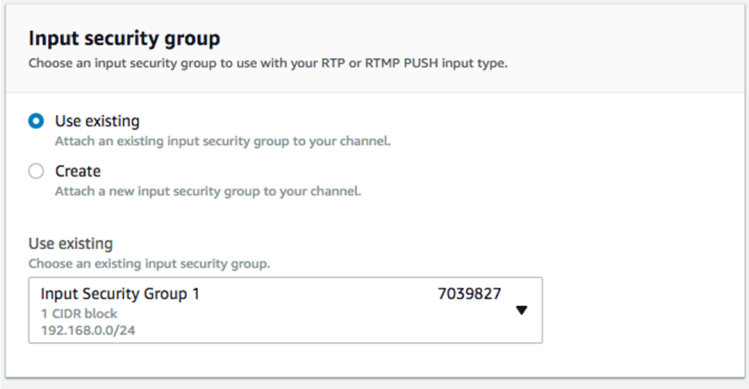


4. Complete the fields as follows:

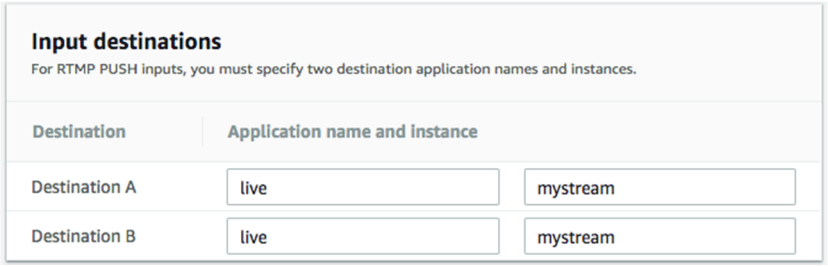
   - **Input name**: Assign a meaningful name.
   - **Input type**: Choose **RTMP (push)**.
   - **Input security group**: Choose **Create**.

     New security group: Using CIDR format, type the set of IP addresses from the Prerequisite step in this document. If you're entering a range, specify a mask that encompasses all of the addresses, or enter several CIDR entries to encompass all of the addresses.

5. Choose **Create input security group**. The tile changes to show the newly created group.

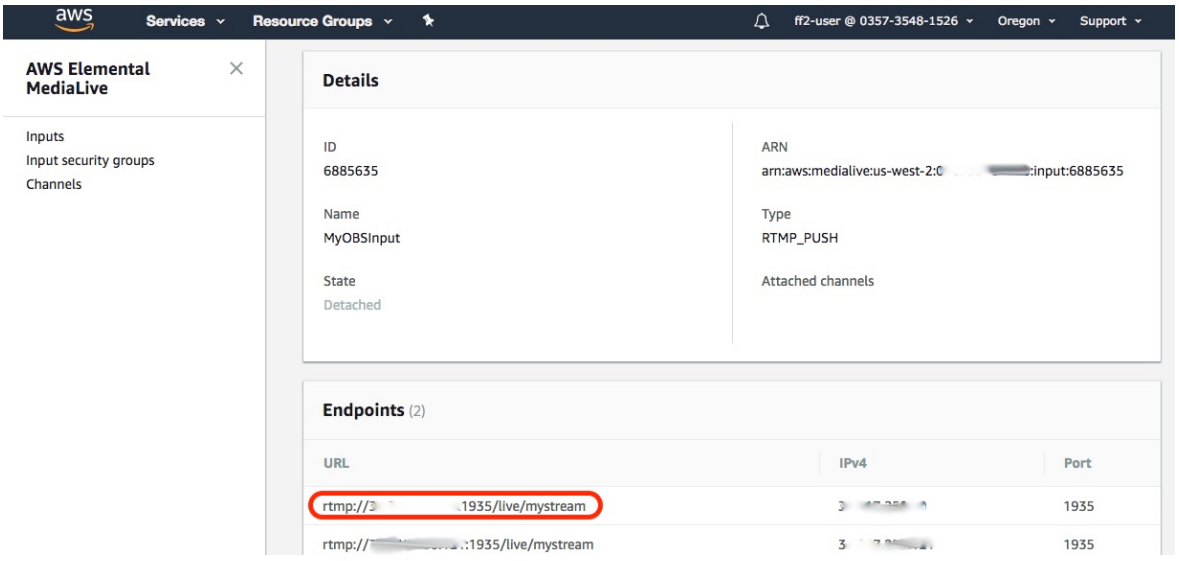


6. In the **Input destinations** section, enter **application name** and application **instance** (stream name), in the fields provided. You will use these names in the OBS setup in Step C.



7. Choose **Create**. The new input appears in the list of inputs.

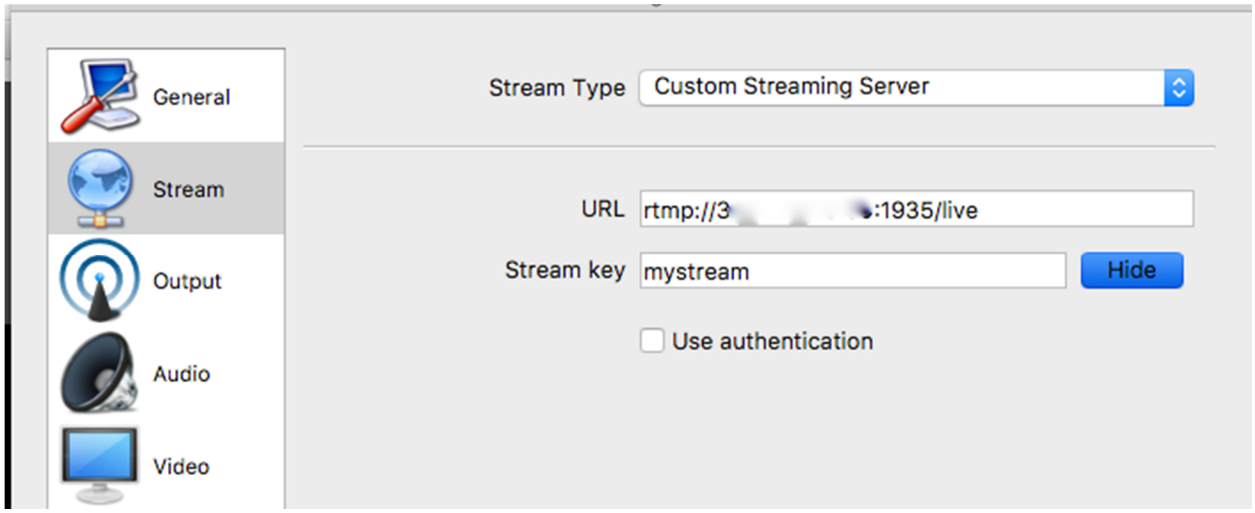
8. Open the detail page for the newly-created inputs, and make a note of the endpoint URLs, as you will need to enter them in the OBS streaming configuration in Step C.



9. Leave this page open. You will return to it in a later step.

STEP C: CONFIGURE THE APPLIANCE

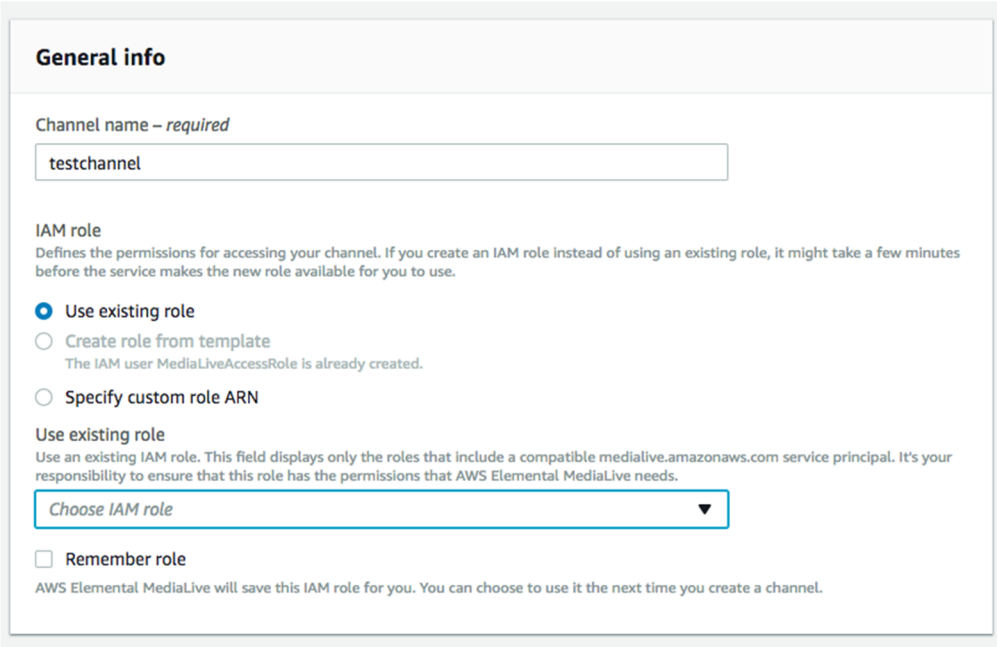
1. Launch OBS Studio on the source system. Choose **Settings** to open the settings window. Choose **Stream** to access the streaming settings.
2. Complete the fields as follows and choose **Apply** to save your changes:
   - For **Stream Type**, choose **Custom Streaming Server**.
   - For **URL**, copy one of the endpoint URLs from the input you created in Step B. Remove the /<stream_name> at the end of the URL.
   - For **Stream key**, type the stream name.
   - Leave the **Use authentication** box unchecked.



3. Choose **Output** from the left-hand menu and in **Output Mode**, choose **Advanced.**
4. Complete the Streaming tab as follows and choose **Apply** to save your changes:
   - For the **Encoder**, choose **x264**.
   - Select **Rescale Output** box and type 1**920x1080** in the drop-down (it's not a drop-down option, but you can enter it manually).
   - Change **Bitrate** to **10000** (this assumes your uplink bandwidth is sufficient).
   - Leave the remaining settings at their defaults.
5. Choose **Audio** from the left-hand menu, confirm the following, and choose **Apply** if you had to make any changes:
   - **Sample rate** is **44.1 kHz**.
   - **Channels** are set to **Stereo**.
6. Choose **Video** from the left-hand menu complete the fields as follows. Choose **Apply** to save your changes.
   - For **Base (Canvas) Resolution**, choose **1920x1080**.
   - For **Output (Scaled) Resolution**, choose **1920x1080**.
   - Set the desired frame rate in **Common FPS Values**.
7. Choose **OK** to dismiss the settings window.

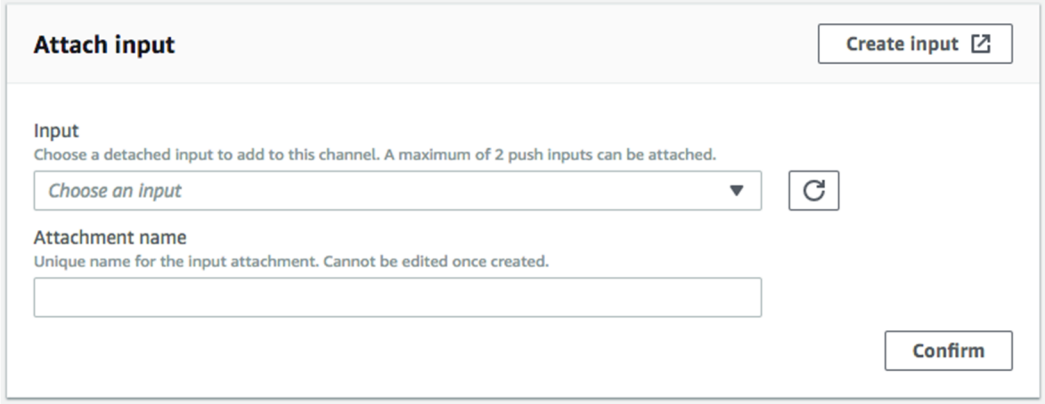
STEP D: CREATE A CHANNEL IN AWS ELEMENTAL MEDIALIVE

1. Switch back to the AWS Elemental MediaLive console.
2. Choose **Channels** from the left-hand column, then choose **Create channel**. The Create channel page appears.
3. For **Channel name**, type a meaningful identifier for the channel.
4. In the **Channel template** section at the bottom, choose **HTTP Live Streaming**. The Channel navigation panel is populated with:
   - One output group named TN2224 (HLS)
   - Ten outputs that all belong to that output group.
5. In the IAM role section, take the appropriate action:
   - If the **Create role from template** option is *enabled*, select that option and choose **Create IAM role**. The role is created. Once the creation process is complete, the role is automatically selected from the **Use existing role** drop-down.
   - If the **Create role from template** option is *grayed out*, select **Use existing role** and then select **MediaLiveAccessRole** from the dropdown.



6. In the Channel template section, choose **HTTP live streaming** from the drop-down list. The output group and outputs appear in the left-hand column.
7. Under **Input specifications**, choose the **Input codec** (**AVC** for our example settings**), Input resolution** (select **HD** for our example), and **Maximum input bitrate** (use **MAX_10_MBPS** in our example).
8. In the left-hand column choose the Add button beside **Input attachments**. The Attach input card appears to the right. Choose the input you created earlier from the drop-down and then choose Confirm. Additional options appear to configure the network input settings, which you can adjust if

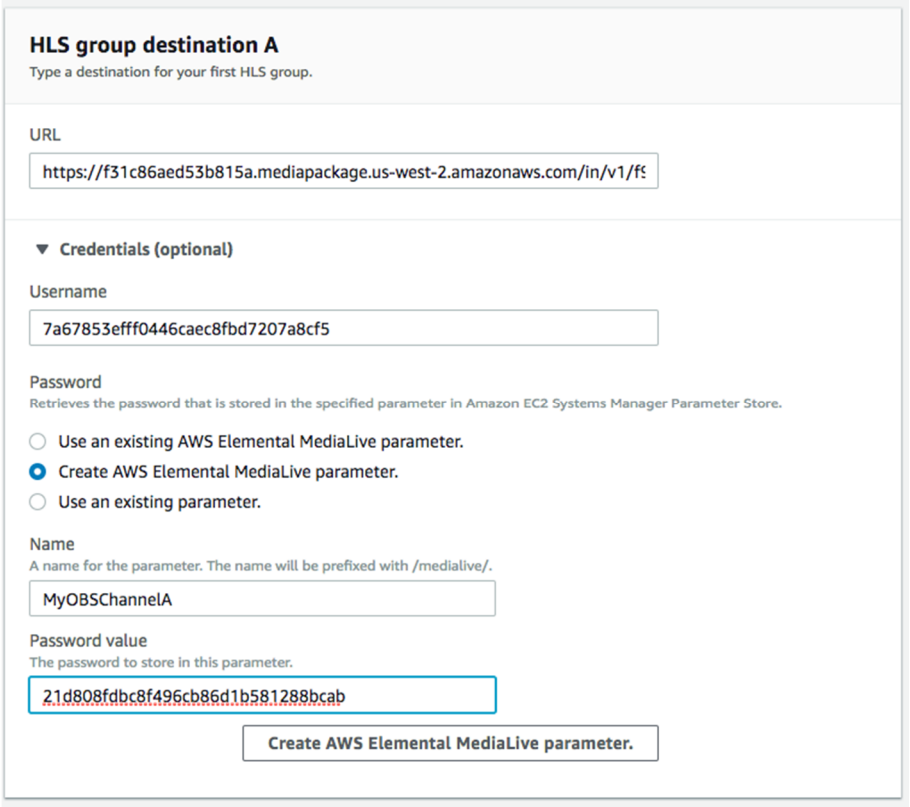
necessary for your particular source.



9. In the left hand column, navigate to “output groups” and choose the group named TN2224 (HLS). The Output Group details appear to the right.
10. In the **HLS group destination A** section, expand the **Credentials** sub-section, then complete the fields with the information from the *first* input of the AWS Elemental MediaPackage channel you created earlier (as described in Step A, part 4).

    - **URL**: Type the first URL for the channel you created in AWS Elemental MediaPackage.
    - **Username**: Type the first username for the channel you created in AWS Elemental MediaPackage.
    - From the list of **Password** options, select the **Create AWS Elemental MediaLive parameter** radio button. In **Name**, enter a meaningful name for the EC2 parameter store entry where your credentials will be stored.
    - **Password**: Type the first password for the channel you created in AWS Elemental MediaPackage. The password will be stored securely in the AWS EC2 parameter store under the name medialive/<name you entered above>.

    Choose the **Create AWS Elemental MediaLive parameter** button to create it.



11. Repeat step 9 for **HLS group destination B**, using the information from the *second* input of the channel that you created in AWS Elemental MediaPackage. You must specify two destinations to create a valid AWS Elemental MediaLive channel.
12. Under HLS Settings, change “Input Loss Action” from EMIT_OUTPUT to PAUSE_OUTPUT. This will allow AWS Elemental MediaPackage to detect a loss of input on one of the MediaLive pipelines and switch any endpoints using the failed pipeline to use the other redundant pipeline.
13. This channel template includes a WebVTT captions output. However, we didn’t define a caption selector on the input, nor did we configure captions on the source appliance. Navigate to the HLS outputs card and choose the X to the right of Output 10 (_webvtt) to delete the captions output.
14. Choose **Create channel**. The page with the list of channels appears, showing the new channel. The status of the channel changes from Creating to Idle.

STEP E: START STREAMING THE VIDEO

You must start the event on the appliance and the AWS Elemental MediaLive channel in the correct order. This example uses RTMP, so you must start the AWS Elemental MediaLive channel *first*. If the channel is not in a **Running** state when you start the OBS stream, the handshake attempt from OBS to the channel will fail.

1. In AWS Elemental MediaLive, on the **Channels** page, choose the radio button next to your new channel. The buttons along the top are enabled.
2. Choose **Start**. The channel state changes to Starting, and then to Running.
3. Switch to OBS and start the stream connection.

Video should begin streaming from the appliance through to AWS Elemental MediaLive, then to AWS Elemental MediaPackage, where you can view it in a preview window. Because only one pipeline is being supplied with input, there will be alerts showing in the MediaLive console, but the MediaPackage endpoint will use the working pipeline to supply the video.