EZDRM Configuration
AWS SPEKE 2.0 for MediaLive and MediaPackage
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>3</td>
</tr>
<tr>
<td><strong>STEP 1 - EZDRM AWS Speke 2.0 Server Deployment</strong></td>
<td>3</td>
</tr>
<tr>
<td>Create API</td>
<td>3</td>
</tr>
<tr>
<td>Create API Resource</td>
<td>5</td>
</tr>
<tr>
<td>Create Method</td>
<td>5</td>
</tr>
<tr>
<td>Integration Request</td>
<td>7</td>
</tr>
<tr>
<td>Deploy API</td>
<td>8</td>
</tr>
<tr>
<td><strong>STEP 2 - Create Role – MediaPackage</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>STEP 3 - Creating an AWS MediaLive &amp; MediaPackage Job</strong></td>
<td>17</td>
</tr>
<tr>
<td>Create a Channel in MediaLive</td>
<td>17</td>
</tr>
<tr>
<td>Create Channels in MediaPackage</td>
<td>22</td>
</tr>
<tr>
<td>DASH-ISO Output example</td>
<td>28</td>
</tr>
<tr>
<td>HLS Output example</td>
<td>31</td>
</tr>
<tr>
<td><strong>Step 4 - Create Endpoints in MediaPackage</strong></td>
<td>34</td>
</tr>
<tr>
<td>No DRM Endpoint example</td>
<td>34</td>
</tr>
<tr>
<td>DASH-ISO Widevine Endpoint example</td>
<td>36</td>
</tr>
<tr>
<td>DASH-ISO Widevine &amp; PlayReady Endpoint example</td>
<td>40</td>
</tr>
<tr>
<td>DASH-ISO PlayReady Endpoint example</td>
<td>44</td>
</tr>
<tr>
<td>CMAF Apple HLS Endpoint example</td>
<td>48</td>
</tr>
<tr>
<td>CMAF Widevine &amp; PlayReady Endpoint example</td>
<td>52</td>
</tr>
<tr>
<td>CMAF Widevine, PlayReady &amp; Apple FairPlay Endpoint example</td>
<td>56</td>
</tr>
<tr>
<td>Microsoft Smooth Streaming for PlayReady Endpoint example</td>
<td>60</td>
</tr>
<tr>
<td><strong>STEP 5 - Starting a MediaLive Channel</strong></td>
<td>64</td>
</tr>
<tr>
<td><strong>Appendix 1 – Error Log Set-up</strong></td>
<td>65</td>
</tr>
<tr>
<td><strong>Additional Information</strong></td>
<td>71</td>
</tr>
</tbody>
</table>

Version 1.2 / Updated April 25, 2022
**Prerequisites**

Installation of AWS Command Line Interface (CLI) pip install is required prior to configuration. Python 3.6 or higher is required.

For more information on requirements set up, visit this link in a browser: [https://docs.aws.amazon.com/cli/latest/userguide/installing.html](https://docs.aws.amazon.com/cli/latest/userguide/installing.html)

To download Python 3.6: [https://www.python.org/downloads/](https://www.python.org/downloads/)

**STEP 1 - EZDRM AWS Speke 2.0 Server Deployment**

We will utilize AWS SPEKE 2.0 to support their multi-key infrastructure.

**Create API**

1. Under API Gateway click **Create API**.

![Create API](image)
2. Build a **REST API** by clicking the **Build** button.

   ![Build REST API](image)

   *Note – do not use REST API Private, only the **REST API** option shown.*

3. Select **REST** protocol, and under Create new API select **New API**. Enter the **API name**, **Description** and select the **Endpoint Type** – **Edge Optimized**. Edge Optimized allows the endpoint to be geo-balanced.

   ![Create API](image)

4. Click **Create API**.
Create API Resource

5. Under Resources **Actions** menu, select **Create Resource**.

6. Leave **Configure as proxy resource** unchecked. Enter **Resource Name**, we recommend **copyProtection** (case sensitive). Leave **Enable API Gateway CORS** unchecked.

7. Click **Create Resource**.

Create Method

8. Under Resources **Actions** menu, select **Create Method**.
9. The Method type is **POST**.

10. Under **Integration Type** select **HTTP**. Select the checkbox for **Use HTTP Proxy Integration**. **HTTP Method** is **POST**. The **Endpoint URL** is [https://cpxx.ezdrm.com/speke2/speke2.aspx](https://cpxx.ezdrm.com/speke2/speke2.aspx)

11. **Content Handling** is **Passthrough**. Select **Use Default Timeout**.

12. Click **Save**.
13. Next, select **Integration Request** link.

**Integration Request**

14. Specify an **HTTP Header**, this is how access to the endpoint is validated with EZDRM.

15. Enter a **Name**, for this example we suggest `ezdrmSecureToken` (case insensitive).

16. Enter the `ezdrmSecureToken` provided through your EZDRM admin portal under **Mapped From** in single quotes (see example).

17. Click **checkmark** to save.
Deploy API

18. Select **Deploy API** from the Actions menu.

19. Select **[New Stage]** under **Stage Name**.
20. Enter the **Stage Name**. This name is used as part of the API URL to identify the version of the API. For example, you can name based on a test or stage version, as well as production, etc. For our example we used “production”.
21. The **Stage Description** can be used to notate the version of the API. For this example we used “deploying production endpoint”.

22. Click **Deploy**.

23. You will copy the **API URL** at the top of the Editor page labeled **Invoke URL**. Paste this URL in a notepad for editing in a future step.
STEP 2 - Create Role – MediaPackage

To create a MediaPackage Role in AWS complete the following steps:

1. Launch the AWS IAM console by searching for IAM.

2. Go to the Roles menu. Click the Add role button.

3. Under AWS service select the MediaConvert role (there isn’t currently a role for MediaPackage) and click the Next: Permissions button.

4. Review attached permissions and click the Next: Tags button.
5. Add any application tags (optional) and then click **Next: Review**

6. Enter the Role name and click Create role.
7. Now that the MediaPackage role is created, click on the link to open the role details.

8. Because a role doesn't exist for MediaPackage, you will need to add an inline policy and change the settings of these role. This gives permission to execute the copy protection.

   First, note your AWS Account ID as part of the Role ARN value (you can also find this value under the My Account menu under Account Settings). Click the link to Add inline policy.

9. Next select the JSON tab and replace with the following code:

```json
{
  "Version": "2012-10-17",
  "Statement": []
}
```
The yellow highlighted value is your **AWS Account ID**, the purple highlighted value is from the **EzDRM SPEKE 2.0 API Invoke URL** created in Step 1 (this value would change if you redeploy the Speke server).

10. Once you’ve entered the correct code in the JSON tab, click the **Review policy** button.
11. On the Review policy page, fill in the policy **Name** and click **Create policy.**
12. Now you will detach the two default policies from the role by clicking the “x” next to AmazonS3FullAccess and AmazonAPIGatewayInvokeFullAccess. Click **Detach** on the Detach policy confirmation screen for both.

13. Then click on the **Trust relationships** tab and click the **Edit trust relationship** button.
14. Edit line 8 from “mediaconvert.amazonaws.com” to “mediapackage.amazonaws.com” and click the Update trust policy button.

The Trust relationships tab should be updated as follows:
15. Once the MediaPackage role is created, make note of the **Role ARN** value for use in a later step. You can copy this value using the doc copy shortcut.

**STEP 3 - Creating an AWS MediaLive & MediaPackage Job**

**Create a Channel in MediaLive**

1. Through **AWS Services** go to **MediaLive** and under **Get Started**, click **Create Channel**.

**Channel and Input Details**

2. The channel is the input for your live broadcast. Enter the **Channel Name** (this is a required value).
3. Under IAM Role, the first time you create a channel, you can select Create Role from Template and click Create IAM role. The MediaLiveAccessRole will be created. You can select to Remember role and it will be available as the existing role for future channels.

4. Under Input attachments, click the Add button.
5. Under **Attach input**, click the **Create input** button.

![Create input button](image)

6. This will pull the source and type of stream pushing up to **MediaLive**, for this example we are using the HLS input type.

![Create input dialog](image)
7. Channel **Input source A** and **Input source B** will be the same for redundancy. For this example, we are using a publicly available HLS stream provided by Apple for testing. You will enter your encoders publishing point URL for both Input sources and click **Create** button.

![Input Source A and B](image)

8. Once the Input is created, it can be selected from the Input dropdown menu.

![Input Selected](image)

9. Click **Confirm**.
10. Under Output groups click **Add**.

![Create channel](image)

11. Select **HLS** and then click **Confirm**.

12. Now you will keep this tab open while you complete the next few steps.
Create Channels in MediaPackage

13. The next step is to create a new channel in **MediaPackage** to ingest the stream that is coming from MediaLive.

*Note: It is helpful to have multiple tabs open during this process, for ease of copying settings between MediaLive and MediaPackage.*

14. Click **Next Step** under **Create a new channel**.

15. Enter the Channel details including the **ID** channel identifier and select the **Input type “Apple HLS”** (this is the only supported type). Click **Create**.
16. This will create the MediaPackage channel. For redundancy, a second channel will need to be created. Select **Channels**.

17. Click create on the **Channels** page and click the **Create** button to create the second redundant channel.

18. Enter the **Channel details** and click **Create**.

19. Now we have the URL and Channel details we will need for the Output Groups in MediaLive.
**MediaLive Output Groups**

20. Back in **MediaLive**, click the **Add** button under **Output groups** and select **HLS**. Click the **Confirm** button.  

![Create channel](image)

**Note: MediaPackage only accepts HLS streams.**

21. Copy and Paste the **Input URL**, **Username** and **Password** from the first **MediaPackage** channel you created to input in the next step.

![Copy and Paste](image)

22. In **MediaLive**, in the first **HLS Group destination A**, enter the copied values for **URL** and **Username**. The first time you set up a password in the Output groups, you will select **Create AWS Elemental MediaLive parameter**. This will allow the password to be saved by AWS for future use. We recommend entering the password **Name** with something that will help you select the correct one when you have multiple channels created in the future. Enter the **Password** value and click to **Create the AWS Elemental MediaLive Parameter**.
23. Copy and Paste the **Input URL**, **Username** and **Password** from the second **MediaPackage** channel you created to input in the next step.
24. Back in MediaLive, in HLS Group destination B for redundancy, repeat the process in Step 17 to enter parameters for URL and Username, Password Name and Password from MediaPackage.
25. Next in MediaLive under HLS Settings enter a Name, and for CDN Settings select Hls webdav. Keep the other settings as default.
**DASH-ISO Output example**

26. This is the Output set up for DASH-ISO. See the next section for HLS Output settings. Under **HLS outputs** click the **Add output** button to create Output 2. You can name Output 1 to represent the video output, and Output 2 to represent the audio output.

![HLS outputs](image)

27. The rest of the settings under **Channel and Input Details** keep as default.

**Input Settings**

28. Click on the input link.

![Input Settings](image)
29. Scroll down to **Add audio selectors** button.

30. Enter the **Audio Selector Name** and copy it to paste in the next section.
31. Under **Output Groups**, select **Output 2 (_a)** and click on the **Video** tab. Click the **Remove video** button.

![Image of Output Groups](image1)

32. In the **Audio 1** section, enter the **Audio Description Name** (we recommend AAC Audio), then paste the **Audio Selector Name** that you entered in Step 24. Select **Aac** under Codec Settings.

![Image of Audio 1 section](image2)

33. Next, select **Output 1 (_v)** and **Remove audio 1**.

![Image of Output 1](image3)
34. You can rename the Video Description Name if you prefer and leave the default settings. Then click Create channel.

35. The MediaLive channel should now be created.

**HLS Output example**

36. This is the Output setup for HLS. Under HLS outputs rename Output 1 to represent the video output.

37. The rest of the settings under Channel and Input Details keep as default.
**Input Settings**

38. Click on the **Input settings** link and click the **Add audio selectors** button.

39. Enter the **Audio Selector Name** and copy it to paste in the next section.
40. Under **Output Groups**, in the **Audio 1** section, enter the **Audio Description Name** (we recommend AAC Audio), then paste the **Audio Selector Name** that you entered in Step 33. Select **Aac** under Codec Settings.

![Audio 1 Settings](image)

41. Next, select **Output 1 (v)**. You can rename the **Video Description Name** if you prefer and leave the default settings. Then click **Create channel**.

![Video Description](image)

42. The MediaLive channel should now be created.

![Create Channel](image)
Step 4 - Create Endpoints in MediaPackage

Endpoints are the outputs for the live stream for viewing. You can have multiple endpoints for each channel.

No DRM Endpoint example
1. In MediaPackage, from the first MediaLive channel you created, click the **Add endpoints** button.

2. Edit the **Endpoint ID** and **Manifest Name** to a unique identifier.

3. Once these settings are completed, click the **Save** button to create the endpoint.
4. Now for redundancy, from your second MediaLive channel, create a 001_no_drm endpoint with the same settings as the one we just created, but change the ID name to indicate the redundant endpoint.

For this example, we called our first channel **MediaLive1** and created the DASH-ISO endpoint **001_no_drm**. Under **MediaLive2** we will create a duplicate DASH-ISO endpoint but name it **001_no_drm_2**.

Duplicate ALL the same settings for the second No DRM endpoint under the second channel and click **Save**.

**Note:** It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

5. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.
DASH-ISO Widevine Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add/edit endpoints button.

2. Click Add. Edit the Endpoint ID and Manifest Name to a unique identifier.

3. Under Packager Settings, select the Type DASH-ISO, and update Segment duration (sec) to 20 seconds.
4. Scroll down and select the toggle for **Encrypt Content**.

The parameters are as follows:

- **ResourceID**: this will be the ID that references your DRM Keys. This is a required field.

  *Note*: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID's for Widevine (one ID per line):
  (Widevine) **edef8ba9-79d6-4ace-a3c8-27dcd51d21ed**
Note: The System ID values need to be lowercase.

- **URL:** The URL is the API URL copied from **Step 1:**
  
  Sample URL:  
  https://i2xXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

- **Role ARN:** This value is from the MediaPackage Role ARN created in **Step 2.**

- **SPEKE version:** Select **Version 2.0**

5. Under **Additional configuration** uncheck **Key rotation interval (sec).**
6. Once these settings are completed, click the **Save** button to create the endpoint.

7. Now for redundancy, from your second MediaLive channel, create a DASH-ISO endpoint with the same settings as the one we just created, but change the **ID** name to indicate the redundant endpoint.

For this example, we called our first channel **MediaLive1** and created the DASH-ISO endpoint **002_dash_widevine**. Under **MediaLive2** we will create a duplicate DASH-ISO endpoint but name it **002_dash_widevine_2**.

Duplicate ALL the same settings for the second DASH-ISO endpoint under the second channel and click **Save**.

**Note:** *It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.*

8. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.
DASH-ISO Widevine & PlayReady Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add/edit endpoints button.

2. Click Add. Edit the Endpoint ID and Manifest Name to a unique identifier.

3. Under Packager Settings, select the Type DASH-ISO, and update Segment duration (sec) to 20 seconds.
4. Scroll down and select the toggle for **Encrypt Content**.

The parameters are as follows:

- **ResourceId**: this will be the ID that references your DRM Keys. This is a required field.

  **Note**: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID’s for Widevine and PlayReady, one ID per line:
  - (Widevine) `edef8ba9-79d6-4ace-a3c8-27dcd51d21ed`
  - (PlayReady) `9a04f079-9840-4286-ab92-e65be0885f95`

  **Note**: The System ID values need to be lowercase.
• **URL**: The URL is the **API URL** copied from **Step 1**: 

Sample URL:  
https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

• **Role ARN**: This value is from the **MediaPackage Role ARN** created in **Step 2**.

![Role ARN Image](image)

• **SPEKE version**: Select Version 2.0

5. Under **Additional configuration** uncheck **Key rotation interval (sec).**
6. Once these settings are completed, click the **Save** button to create the endpoint.

   ![Add/edit endpoints](image)

7. Now for redundancy, from your second MediaLive channel, create a DASH-ISO endpoint with the same settings as the one we just created, but change the **ID** name to indicate the redundant endpoint.

   For this example, we called our first channel **MediaLive1** and created the DASH-ISO endpoint **003_dash_widevine_playready**. Under **MediaLive2** we will create a duplicate DASH-ISO endpoint but name it **003_dash_widevine_playready_2**.

   Duplicate ALL the same settings for the second DASH-ISO endpoint under the second channel and click **Save**.

   **Note**: It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

8. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.
DASH-ISO PlayReady Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add/edit endpoints button.

2. Click Add. Edit the Endpoint ID and Manifest Name to a unique identifier.

9. Under Packager Settings, select the Type DASH-ISO, and update Segment duration (sec) to 20 seconds.
10. Scroll down and select the toggle for Encrypt Content.

The parameters are as follows:

- **ResourceID**: this will be the ID that references your DRM Keys. This is a required field.

  *Note*: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID's for PlayReady (one ID per line):

  (PlayReady) 9a04f079-9840-4286-ab92-e65be0885f95

  *Note*: The System ID values need to be lowercase.

- **URL**: The URL is the API URL copied from Step 1:
Sample URL: https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

- **Role ARN**: This value is from the MediaPackage Role ARN created in Step 2.

- **SPEKE version**: Select Version 2.0

11. Under **Additional configuration** uncheck **Key rotation interval (sec).**
12. Once these settings are completed, click the **Save** button to create the endpoint.

![Add/edit endpoints](image)

13. Now for redundancy, from your second MediaLive channel, create a DASH-ISO endpoint with the same settings as the one we just created, but change the **ID name** to indicate the redundant endpoint.

   For this example, we called our first channel **MediaLive1** and created the DASH-ISO endpoint **004_dash_playready**. Under **MediaLive2** we will create a duplicate DASH-ISO endpoint but name it **004_dash_playready_2**.

   Duplicate ALL the same settings for the second DASH-ISO endpoint under the second channel and click **Save**.

   **Note:** It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

14. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.

   ![URL](image)
CMAF Apple HLS Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add endpoints button.

2. Click Add. Edit the Endpoint ID and Manifest Name to a unique identifier.

3. Under Packager Settings, select the Common Media Application Format (CMAF) and leave the other settings as default.
4. Enter the same manifest name for the **HLS Manifest**.

5. Scroll down and select the toggle for **Encrypt Content**.

The parameters are as follows:

- **ResourceId**: this will be the ID that references your DRM Keys. This is a required field.

  *Note: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.*
• **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID for Apple FairPlay (one ID per line):
  94ce86fb-07ff-4f43-adb8-93d2fa968ca2

*Note: The System ID values need to be lowercase.*

• **URL**: The URL is the API URL copied from Step 1:

  Sample URL:  
  https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

• **Role ARN**: This value is from the MediaPackage Role created in Step 2.

  ![Role ARN Image]

  6. Once these settings are completed, click the **Save** button to create the endpoint.
7. Now for redundancy, from your second MediaLive channel, create an Apple HLS endpoint with the same settings as the one we just created, but change the ID name to indicate the redundant endpoint.

For this example, we called our first channel MediaLive1 and created the Apple HLS endpoint 005_cmaf_fairplay. Under MediaLive2 we will create a duplicate Apple HLS endpoint but name it 005_cmaf_fairplay_2.

Duplicate ALL the same settings for the second CMAF endpoint under the second channel and click Save.

Note: It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

8. Once MediaLive is running and publishing to MediaPackage, you will be able to access the URL created to play the encrypted Media.
CMAF Widevine & PlayReady Endpoint example

6. In MediaPackage, from the first MediaLive channel you created, click the **Add endpoints** button.

![Add endpoints](image)

7. Click **Add**. Edit the **Endpoint ID** and **Manifest Name** to a unique identifier.

![Add/edit endpoints](image)

8. Under **Packager Settings**, select the **Common Media Application Format (CMAF)** and leave the other settings as default.

![Packager settings](image)
9. Enter the same manifest name for the **HLS Manifest**.

![HLS manifest](image)

10. Scroll down and select the toggle for **Encrypt Content**.

![Package encryption](image)

The parameters are as follows:

- **ResourceID**: this will be the ID that references your DRM Keys. This is a required field.

*Note: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference*
the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID:** Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID’s for Widevine and PlayReady, one ID per line:
  - (Widevine) edef8ba9-79d6-4ace-a3c8-27dcd51d21ed
  - (PlayReady) 9a04f079-9840-4286-ab92-e65be0885f95

**Note:** The System ID values need to be lowercase.

- **URL:** The URL is the API URL copied from Step 1:

  Sample URL: https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

- **Role ARN:** This value is from the MediaPackage Role created in Step 2.

9. Once these settings are completed, click the **Save** button to create the endpoint.

10. Now for redundancy, from your second MediaLive channel, create an Apple HLS endpoint with the same settings as the one we just created, but change the ID name to indicate the redundant endpoint.
For this example, we called our first channel **MediaLive1** and created the Apple HLS endpoint **006_cmaf_widevine_playready**. Under **MediaLive2** we will create a duplicate Apple HLS endpoint but name it **006_cmaf_widevine_playready_2**.

Duplicate ALL the same settings for the second CMAF endpoint under the second channel and click **Save**.

**Note:** *It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.*

11. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.

![URL](http://example.com/mediapackage-url)
CMAF Widevine, PlayReady & Apple FairPlay Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add endpoints button.

2. Click Add. Edit the Endpoint ID and Manifest Name to a unique identifier.

3. Under Packager Settings, select the Common Media Application Format (CMAF) and leave the other settings as default.
4. Enter the same manifest name for the **HLS Manifest**.

   ![HLS manifest](image)

   **ID**
   
   ```
   007_cmaf_widevine_playready_fairplay
   Must be unique within the endpoint and it cannot be changed after it is created.
   ```

5. Scroll down and select the toggle for **Encrypt Content**.

   ![Package encryption](image)

   **Resource ID**
   
   The resource ID is the identifier that you send to the key server to identify this endpoint.
   
   ```
   007_cmaf_widevine_playready_fairplay
   ```

   **System IDs**
   
   A system ID is a unique identifier for the DRM system to use. Type one per line.

   ![System IDs](image)

   **Role ARN**
   
   The Amazon Resource Name (ARN) for the IAM role that you created that allows communication between SPEKE and AWS Elemental MediaPackage.
   
   ```
   arn:aws:iam::123456789012:role/MediaPackage
   ```

   **SPEKE version**
   
   Select the version of SPEKE to use for encryption on this endpoint. SPEKE 1.0 is the legacy version that uses CPDK 2.0, and supports single key encryption. SPEKE 2.0 uses CPDK 2.5, and supports multiple key encryption.
   
   ![SPEKE version](image)

The parameters are as follows:

- **ResourceID**: this will be the ID that references your DRM Keys. This is a required field.

  **Note**: *The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference*
the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID’s for Widevine, PlayReady and FairPlay, one ID per line:
  - **Widevine**: edef8ba9-79d6-4ace-a3c8-27dcd51d21ed
  - **PlayReady**: 9a04f079-9840-4286-ab92-e65be0885f95
  - **FairPlay**: 94ce86fb-07ff-4f43-adb8-93d2fa968ca2

  *Note*: The System ID values need to be lowercase.

- **URL**: The URL is the API URL copied from Step 1:

  Sample URL: https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

- **Role ARN**: This value is from the MediaPackage Role created in Step 2.

12. Once these settings are completed, click the **Save** button to create the endpoint.

13. Now for redundancy, from your second MediaLive channel, create an Apple HLS endpoint with the same settings as the one we just created, but change the ID name to indicate the redundant endpoint.
For this example, we called our first channel **MediaLive1** and created the Apple HLS endpoint **007_cmaf_widevine_playready_fairplay**. Under **MediaLive2** we will create a duplicate Apple HLS endpoint but name it **007_cmaf_widevine_playready_fairplay_2**.

Duplicate ALL the same settings for the second CMAF endpoint under the second channel and click **Save**.

**Note:** It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

14. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the encrypted Media.

https://630eff2268.mediapackage.us-east-1.amazonaws.com/out/v1/603f0a911294cd5610072/007_cmaf_widevine_playready_fairplay/007_cmaf_widevine_playready_fairplay.m3u8

QR code
Microsoft Smooth Streaming for PlayReady Endpoint example

1. In MediaPackage, from the first MediaLive channel you created, click the Add endpoints button.

2. Click the Add button. Edit the Endpoint ID and Manifest Name to a unique identifier.
3. Under **Packager Settings**, select the **Type Smooth** and **Segment duration (sec)** to **20** seconds.

![Packager settings](image)

4. Scroll down and select the toggle for **Encrypt Content**.

![Package encryption](image)

The parameters are as follows:

- **ResourceID**: this will be the ID that references your DRM Keys. This is a required field.
Note: The first time you send a ResourceID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same ResourceID to reference the same keys or for new DRM Keys send a new ResourceID. It is best not to use a ResourceID from a failed job.

- **System ID**: Unique identifiers for the DRM system to use. These System IDs are industry standard, must be utilized for encryption. Insert the System ID for Smooth Streaming (*uses PlayReady’s System ID*), one ID per line:

  9a04f079-9840-4286-ab92-e65be0885f95

  Note: The System ID values need to be lowercase.

- **URL**: The URL is the API URL copied from Step 1:

  Sample URL: https://i2qXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

- **Role ARN**: This value is from the MediaPackage Role created in Step 2.

5. Once these settings are completed, click the **Save** button to create the endpoint.

6. Now for redundancy, from your second MediaLive channel, create a Smooth Streaming endpoint with the same settings as the one we just created, but change the **ID** name to indicate the redundant endpoint.
For this example, we called our first channel **MediaLive1** and created the Smooth Streaming endpoint **smooth-001**. Under **MediaLive2** we will create a duplicate Smooth Streaming endpoint but name it **smooth-002**.

Duplicate ALL the same settings for the second Smooth Streaming endpoint under the second channel and click **Save**.

*Note:* It is helpful to have multiple tabs open during this process, for ease of copying settings from one channel to the other.

7. Once **MediaLive** is running and publishing to **MediaPackage**, you will be able to access the URL created to play the DRM encrypted Media.
STEP 5 - Starting a MediaLive Channel

Open MediaLive and select the channel. Click the Start button to start the channel.

Once the channel is started, data for the stream will be shown in the Health section.

If Input video frame rate is ever not running, you know that there is a problem with the stream.

Same on the MediaPackage side, there will be data showing under Operational metrics.
Appendix 1 – Error Log Set-up

1. To set up an error log, go to Simple Notification Service in AWS.

2. Click Create topic from the SNS dashboard.

3. To create a Standard topic, enter the Topic name and Display name and click Create topic.

4. The Topic details will open, then click Create subscription.
5. Change the **Protocol** to **Email** and enter the **email address** in the **Endpoint** field. Click **Create subscription**.

6. There will now be a **Pending Confirmation** line item, and an email will be sent to confirm the subscription.
7. Next, open **Amazon EventBridge** under AWS Services.

8. Under the **Rules** menu, click **Create rule**.

9. Enter the Rule **Name** and **Description** (optional).
10. Under Define pattern select Event pattern.

![Define pattern screenshot]

11. Select:
   - Pre-defined pattern by service
   - Service Provider: AWS
   - Service name: MediaConvert (there isn't a service for MediaPackage)
   - Event Type: All Events

![Define pattern screenshot with pre-defined pattern]

12. Click the **Edit** button.

13. Update “aws.mediaconvert” to “aws.mediapackage” and click **Save**.
14. Under Select targets, click **Add target**.

15. Select **SNS Topic** from the dropdown and select the **Topic** you created in Step 3, for this example “ErrorfromMediaPackage”.
16. Click the **Create** button.

You will now get an error message in the event that there is a connection issue.
Additional Information
For additional questions and comments please contact: simplify@ezdrm.com