EZDRM Configuration
AWS MediaConvert
Table of Contents

Prerequisites ........................................................................................................................................3

STEP 1 - EZDRM AWS Speke 2.0 Server Deployment .....................................................................3

  Create API ...................................................................................................................................3
  Create API Resource ....................................................................................................................5
  Create Method ..............................................................................................................................5
  Integration Request .......................................................................................................................7
  Deploy API ....................................................................................................................................8

STEP 2 - Create Role – MediaConvert ............................................................................................10

STEP 3 - Creating an AWS MediaConvert Job ...............................................................................13

  Widevine and PlayReady ............................................................................................................13
  Apple FairPlay Streaming ..........................................................................................................20
  CMAF (Apple HLS and MPEG-DASH) ......................................................................................29

Additional Information ...................................................................................................................36

Version 5.0 / Updated July 19, 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

To download Python 3.6: https://www.python.org/downloads/

STEP 1 - EZDRM AWS Speke 2.0 Server Deployment

We will utilize AWS SPEKE 2.0 with backward compliance for SPEKE 1.0.

Create API

1. Under API Gateway click Create API.
2. Build a **REST API** by clicking the **Build** button.

   ![REST API Build](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.
4. Click **Create API**.

Create API Resource

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

![Create Resource](image)

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**.

![Create Method](image)
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://cpix.ezdrm.com/speke2/speke2.aspx** (SPEKE 2.0 is backwards compliant with all SPEKE 1.0 jobs)

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

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

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

To create a the MediaConvert 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 and click the Next button.

   Select trusted entity
   
   Trusted entity type
   
   Use case
   
   Use cases for other AWS services:
   
   MediaConvert
   
   Allows MediaConvert service to call S3 APIs and API Gateway on your behalf.
4. Review attached permissions and click **Next** button.

5. Enter the **Role name**, review Permissions and click **Create role**.
6. Now that the MediaPackage role is created, click on the link to open the role details.
STEP 3 - Creating an AWS MediaConvert Job

Widevine and PlayReady

1. Launch the AWS S3 console by searching for S3.

2. Create a new bucket or use an existing bucket.

3. Select the bucket and click the Create folder button.

4. Create two folders, one named input and one named output.
5. Select the **input** folder and click the **Upload** button. Then select the **Add Files. Upload** the test mp4. For this example we used **BigBuckBunny_320x180.mp4**. The file will now show in the input folder.

6. Copy the **S3 Destination URL** for the next step.
7. Go to MediaConvert and under Create a job, click Get Started.

8. For Input 1 enter the S3 Destination URL from Step 6.

9. Next click the Add for Output groups. For this example, select DASH ISO and click the Select button.
10. The DASH ISO group settings will open. The **Custom Group Name** is optional. Under **Destination** enter the **S3 URI for the Output folder** in the Bucket previously created.
11. Scroll down and select the toggle for **DRM Encryption**.

![DRM Encryption Toggle](image)

The parameters are as follows:

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

  *Note: The first time you send a Resource ID to run a job, the ID will be tied to the DRM keys for that job. Jobs can use the same Resource ID to reference the same keys or for new DRM Keys send a new Resource ID. It is best not to use a Resource ID 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-a3c8-27cd51d21ed**
  (PlayReady) **9a04f079-9840-4286-ab92-e65be0885f95**

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

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

12. For the Video output “**H.264, _output1**”, be sure to enter a **Max bitrate (bits/s)**, this value is required. For this example, we used 128000.

13. For Audio output “**AAC, _output2**”, be sure to set the appropriate **Bitrate (kbit/s)**.
14. Confirm the **Service Role** (created in **Step 2**) under Job Settings / AWS integration.

15. Once all of the settings are complete, scroll to the bottom of the page and click the **Create** button to create the job.
16. The job will show Submitted and then if you click the **Refresh** button, you will see the job listed as Complete.

![Job summary table]

**Apple FairPlay Streaming**

1. Launch the AWS S3 console by searching for S3.

![AWS S3 console]

2. Create a new bucket or use an existing bucket.

3. Select the bucket and click the **Create folder** button.

![AWS S3 bucket creation]

![AWS S3 media conversion]

4. Create two folders, one named **input** and one named **output**.
5. Select the **input** folder and click the **Upload** button. Then select the **Add Files.** **Upload** the test mp4. For this example, we used **BigBuckBunny_320x180.mp4**. The file will now show in the input folder.

![Upload](image)

6. Copy the **S3 Destination URL** for the next step.

7. Go to **MediaConvert** and under **Create a job**, click **Get Started**.
8. For **Input 1** enter the **S3 Destination URL** from **Step 6**.

   ![AWS Elemental MediaConvert](image1)

9. Next click the **Add** for **Output groups**. For this example, select **Apple HLS** and click the **Select** button.

   ![AWS Elemental MediaConvert](image2)
10. The Apple HLS group settings will open. The **Custom Group Name** is optional. Under **Destination** enter the **S3 URI for the Output folder** in the Bucket previously created.
11. Scroll down and select the toggle for **DRM Encryption**.

The parameters are as follows:

- **Encryption method**: select `SAMPLE_AES`.
- **Key provider type**: select `SPEKE`.
- **Initialization vector in manifest**: select `Include`.
- **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. 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.

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

  Sample URL: https://i2xXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

12. Next, select “**H.264, AAC**” and then select **Video**. Enter a **Max bitrate (bits/s)** – this is a required field. For this example, we used 128000.

13. Next select **Audio 1** output, be sure to set the appropriate **Bitrate (kbit/s)**. For this example, it is set to 96.0.
14. Confirm the **Service Role** (created in **Step 2**) under Job Settings / AWS integration.

15. Once all of the settings are complete, scroll to the bottom of the page and click the **Create** button to create the job.

16. The job will show Submitted and then if you click the **Refresh** button, you will see the job listed as Complete.
CMAF (Apple HLS and MPEG-DASH)

1. Launch the AWS S3 console by searching for S3.

2. Create a new bucket or use an existing bucket.

3. Select the bucket and click the Create folder button.

4. Create two folders, one named input and one named output.
5. Select the input folder and click the **Upload** button. Then select the **Add Files.** Upload the test mp4. For this example we used **BigBuckBunny_320x180.mp4.** The file will now show in the input folder.

6. Copy the **S3 Destination URL** for the next step.
7. Go to MediaConvert and under Create a job, click Get Started.

8. For Input 1 enter the S3 Destination URL from Step 6.

9. Next click the Add for Output groups. For this example, select CMAF and click the Select button.
10. The CMAF group settings will open. The **Custom Group Name** is optional. Under **Destination** enter the **S3 URI for the Output folder** in the Bucket previously created.
11. Scroll down and select the toggle for **DRM Encryption**.

The parameters are as follows:

- **Encryption method**: select **AES-CBC subsample**
- **Initialization vector in manifest**: select **Include**
- **Key provider type**: select **SPEKE**
- **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 signaled in HLS:** Unique identifiers for the DRM system to use. 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.

- **System ID signaled in DASH:** 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.

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

  Sample URL: https://i2xXXjdb1e.execute-api.us-east-1.amazonaws.com/production/copyProtection

12. Next, select the Video “H.264” under CMAF Output groups. Enter a Max bitrate (bits/s) – this is a required field. For this example we entered 128000.
13. Go back up to the top left side of the screen to change the settings for AAC. Under the Audio 1 output, be sure to set the appropriate Bitrate (kbit/s). For this example, it is set to 96.0.

14. Confirm the Service Role (created in Step 2) under Job Settings / AWS integration.
15. Once all of the settings are complete, scroll to the bottom of the page and click the **Create** button to create the job.

![ Nielsen configuration](image)

16. The job will show Submitted and then if you click the **Refresh** button, you will see the job listed as Complete.

![ Job summary](image)

**Additional Information**

For additional questions and comments please contact: simplify@ezdrm.com
Updated 3/8/22 – V5.0
Speke 2.0 Server updates / updated AWS screen flows

4/27/22 – V5.0 – Certificate ARN correction and added Service Role screencap

5/10/22 – updated screenshots and added CMAF