

EZDRM Configuration for Google VOD Transcoding Packager

Table of Contents

Prerequisites	3
Create Transcoder and Storage Roles	3
Google Packager Encryption	4
<i>Generating Keys.....</i>	<i>4</i>
<i>Encryption Keys file - Secret Manager.....</i>	<i>6</i>
<i>Create Keys File.....</i>	<i>9</i>
CENC-Widevine.....	9
CENC-PlayReady.....	10
Apple Fairplay	11
Create Bucket (Public) and Set Permissions	13
Transcoder – Create Jobs.....	15
Create Templates	16
Authorization	18
Testing Playback.....	19
Additional Information	19

Prerequisites

Download Google CLI – <https://cloud.google.com/sdk/docs/install>

For more information visit:

<https://cloud.google.com/transcoder/docs/how-to/encrypt-content>

Create Transcoder and Storage Roles

Within your project, you will need to create two access roles using the following CLI commands. For this example, our project is *transcode-001*. Specify your **member user login** and create the roles for **transcoder.admin** and **storage.admin**.

```
gcloud projects add-iam-policy-binding transcode-001 --member="user:youruser@company.com" --role=roles/transcoder.admin
```

```
gcloud projects add-iam-policy-binding transcode-001 --member="user:youruser@company.com" --role=roles/storage.admin
```

Google Packager Encryption

Generating Keys

Below are the steps to create the DRM Keys for CENC-PlayReady or CENC-Widevine encryption for Google Packager, as well as Apple FairPlay DRM.

EZDRM Apple FairPlay DRM is a hosted Apple FairPlay Streaming (DRM). This enables a content owner to encrypt the media with Apple FPS DRM keys and deliver content Apple devices with native support MAC Safari browser via HTML 5 player or iOS via native App or Safari.

To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

For more details on CPIX, please visit our website:

<https://hs.ezdrm.com/hubfs/Documentation/EZDRM%20CPIX%202.x%20Keys%20Guide%20V1.pdf?hsLang=en>

Option 1: Request DRM keys using EZDRM CPIX Web Service

1. Call the EZDRM web service in a browser:
<https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=resourceName>

The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID - generic resource name/identifier (client generated) - passed into id field

* To generate a GUID for the k value, you can use a GUID generator like the one found here: <https://www.guidgenerator.com/>.

2. The response from EZDRM will look like this:

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId=" " version="2.3">
  <cpix:ContentKeyList>
    <cpix:ContentKey explicitIV="IiI4" >P5A==" kid="222238c8-1c8921e3f1e4" commonEncryptionScheme="cenc">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>DTNvniU5XXXXXXxXvUkNjQ==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSystemList>
    <cpix:DRMSystem id="222238c8-1c8921e3f1e4" systemId="edef8ba9-939f-4746-8010-4d51d21ed">
      <cpix:PSSH AAAAP3Bcz2gAAAAA7eLqX V6ZH3150Pc1ZgC</cpix:PSSH>
      <cpix:ContentProtectionData>PHBzc2gg6G1sbm9Iny6jptcGVnoM1bmM6HjAwly1+QUFBQVAzQnpjImdBQUFBQzI1K0x0xw55XU3H2an1DzWfXU; PUGHslnNHC9vc3HoPg==</cpix:ContentProtectionData>
      <cpix:HLSSignalingData>
        <playList>"media">I0VVVCLVUfFuTpnRVRI7Q9u0FNUeXFLUfFujyLDVFIs58V25UQ9HgyYjYjYtzh00DEWITE0Q9Y; xW5XU3H2an1DzWfXUjBoN1FBQUFCOFNFQ01pT01nukVVeJf
        <playList>"master">I0VVVCLVUfFuTpnRVRI7Q9u0FNUeXFLUfFujyLDVFIs58V25UQ9HgyYjYjYtzh00DEWITE0Q9Y; VVeJly0EFjaVNIajh1UwFUV2khhkDFNUGHslnNHC9vc3HoPg==
      </cpix:DRMSystem>
    </cpix:DRMSystemList>
  </cpix:CPIX>
```

- o **id** – c value returned, generic resource name/identifier (client generated)
- o **kid** – Key ID in GUID format (client generated)*
- o **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value.
- o **explicitIV** – the Apple FairPlay explicit IV value
- o **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

* To generate a GUID for the k value, you can use a GUID generator like the one found here: <https://www.guidgenerator.com/>.

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" contentId=" "
" version="2.3">
<cpix:ContentKeyList>
<cpix:ContentKey explicitIV="IiI4xXXXXxxXwByJiEpx5A==" kid="222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4" commonEn
ryptionScheme="cenc">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>DTNvniU5XXXXXXxXvUkNjQ==</pskc:PlainValue>
</pskc:Secret>
```

```
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
>
<cpix:PSSH>AAAP3Bzc2gXXXXXe+LqXnWSs6jyCfc1R0h7QAAAXXXXXXXXXMgREUz3r8AcISHj8eQaBWV6ZHJtSOPc1ZsG</cpix:PSSH
>
```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v "https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=drm-001"
```

Encryption Keys file - Secret Manager

Once you have the DRM values to encrypt the content, you can add them to the Google packager for CENC-PlayReady, CENC-Widevine, and Apple FairPlay.

The sample JSON file for key encryption for all types outlined in this section. This file will be uploaded into Secrets.

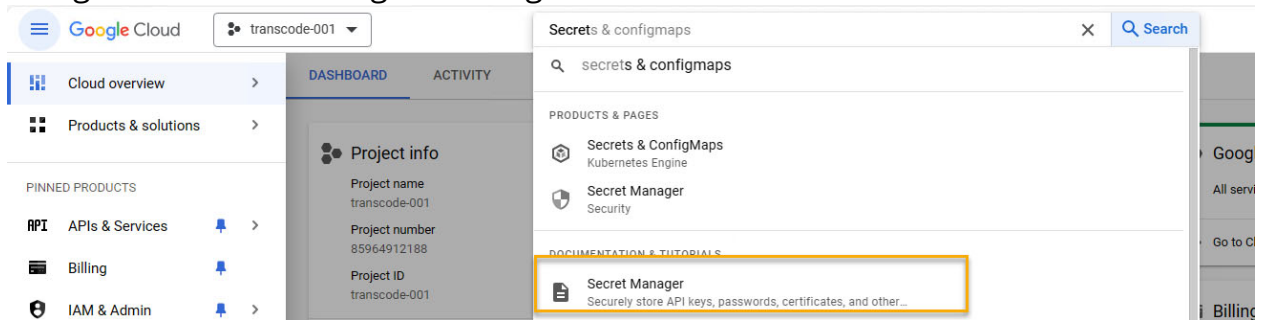
1. Create the keys JSON file as outlined in the following section.

```

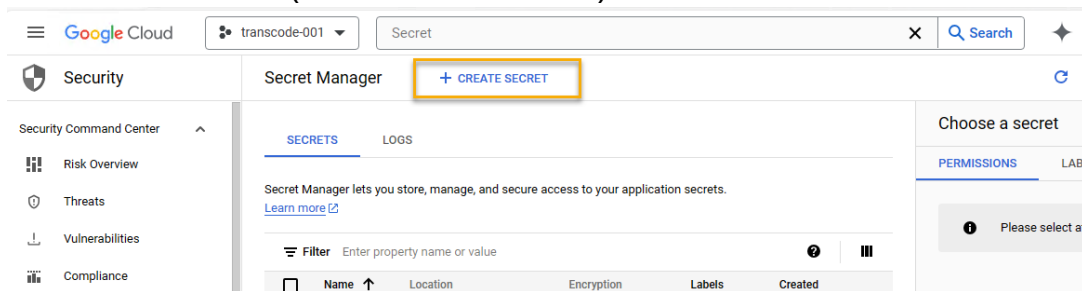
all_keys_sample.json
1 {
2   "encryptionKeys": [
3     {
4       "keyId": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
5       "key": "0d33XxXx25395d15538f8712bd49XxXx",
6       "iv": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
7       "keyUri": "skd://222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4:222238c8XXXX4cf7XXXX1c8921e3f1e4",
8       "matchers": [
9         {
10          "muxStreams": [
11            "ts_fairplay"
12          ]
13        }
14      ]
15    },
16    {
17      "keyId": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
18      "key": "0d33XxXx25395d15538f8712bd49XxXx",
19      "iv": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
20      "keyUri": "skd://222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4:222238c8XXXX4cf7XXXX1c8921e3f1e4",
21      "matchers": [
22        {
23          "muxStreams": [
24            "mp4_widevine_cenc_video",
25            "mp4_widevine_cenc_audio",
26            "mp4_widevine_cbcs_video",
27            "mp4_widevine_cbcs_audio"
28          ]
29        }
30      ]
31    }
32  ]
33 }

```

2. Navigate to Secret Manager in Google Cloud.



3. Create New Secret (click + Create Secret).



Enable the key for the transcoding service by running this CLI command:

```
gcloud projects add-iam-policy-binding transcode-001 service-85964912188@gcp-sa-transcoder.iam.gserviceaccount.com --role=roles/secretmanager.secretAccessor
```

Create Keys File

Note – the command requires an iv value, even when running for only Widevine or PlayReady.

CENC-Widevine

- **“keyId”** - the EZDRM **kid** value (no dashes)
- **“key”** - use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = DTNvniU5XXXXXxxXvUkNjQ==



(keyID - HEX) = 0d33XxXx25395d15538f8712bd49XxXx

- **keyURI** - specify the license URL for encryption. Build by appending the **kid** value to base URL as shown in this example:
skd://222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4
- **“iv”** - decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at:
https://tomeko.net/online_tools/base64.php?lang=en

explicitIV (Base 64) = liI4xXXXXxxXwByJlePx5A==



iv (HEX no dashes) = 222238c8XXXX4cf7XXXX1c8921e3f1e4

```
{
  "keyId": "222238c8XXXX4cf7XXX1c8921e3f1e4",
  "key": "0d33XxXx25395d15538f8712bd49XxXx",
  "keyUri": "skd://222238c8XXXX4cf7XXX1c8921e3f1e4",
  "iv": "222238c8XXXX4cf7XXX1c8921e3f1e4",
  "matchers": [
    {
      "muxStreams": [
        "fmp4_widevine_cenc_video",
        "fmp4_widevine_cenc_audio",
        "fmp4_widevine_cbcs_video",
        "fmp4_widevine_cbcs_audio"
      ]
    }
  ]
},
```

CENC-PlayReady

- **“keyId”** - the EZDRM **kid** value (no dashes)
- **“key”** - use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = DTNvniU5XXXXXxxXvUkNjQ==



(keyID - HEX) = 0d33XxXx25395d15538f8712bd49XxXx

- **keyURI** - specify the license URL for encryption. Build by appending the **kid** value to base URL as shown in this example:
skd://222238c8-XXXX-4cf7-XXX-1c8921e3f1e4
- **“iv”** - decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

explicitIV (Base 64) = lil4xXXXXxxXwByJlePx5A==



iv (HEX no dashes) = 222238c8XXXX4cf7XXXX1c8921e3f1e4

```
{
  "keyId": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
  "key": "0d33XxXx25395d15538f8712bd49XxXx",
  "keyUri": "skd://222238c8XXXX4cf7XXXX1c8921e3f1e4",
  "iv": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
  "matchers": [
    {
      "muxStreams": [
        "fmp4_playready_cenc_video",
        "fmp4_playready_cenc_audio",
        "fmp4_playready_cbcs_video",
        "fmp4_playready_cbcs_audio"
      ]
    }
  ]
},
```

Apple Fairplay

- **"keyId"** - the EZDRM **kid** value (no dashes)
- **"key"** - use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = DTNvniU5XXXXXxxXvUkNjQ==



(keyID - HEX) = 0d33XxXx25395d15538f8712bd49XxXx

- **“iv”** - decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at:

https://tomeko.net/online_tools/base64.php?lang=en

explicitIV (Base 64) = liI4xXXXXxxXwByJlePx5A==



iv (HEX no dashes) = 222238c8XXXX4cf7XXXX1c8921e3f1e4

- **keyURI** - specify the license URL for encryption. Build by appending the **kid** value to base URL and the **iv** value separated by a colon as shown in this example:

skd://222238c8-XXXX-4cf7-XXXX-

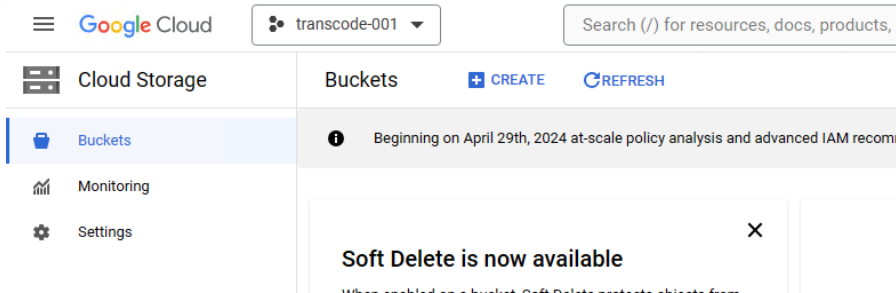
1c8921e3f1e4:222238c8XXXX4cf7XXXX1c8921e3f1e4

```

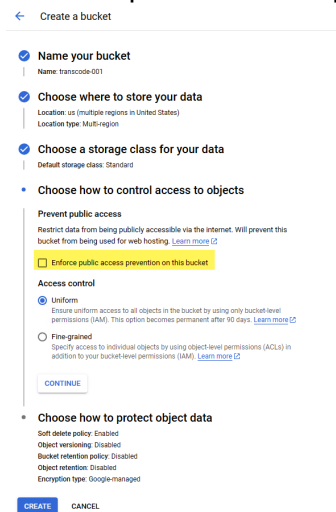
    {
      "keyId": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
      "key": "0d33XxXx25395d15538f8712bd49XxXx",
      "iv": "222238c8XXXX4cf7XXXX1c8921e3f1e4",
      "keyUri": "skd://222238c8-XXXX-4cf7-XXXX-1c8921e3f1e4:222238c8XXXX4cf7XXXX1
c8921e3f1e4",
      "matchers": [
        {
          "muxStreams": [
            "ts_fairplay"
          ]
        }
      ]
    },
  
```

Create Bucket (Public) and Set Permissions

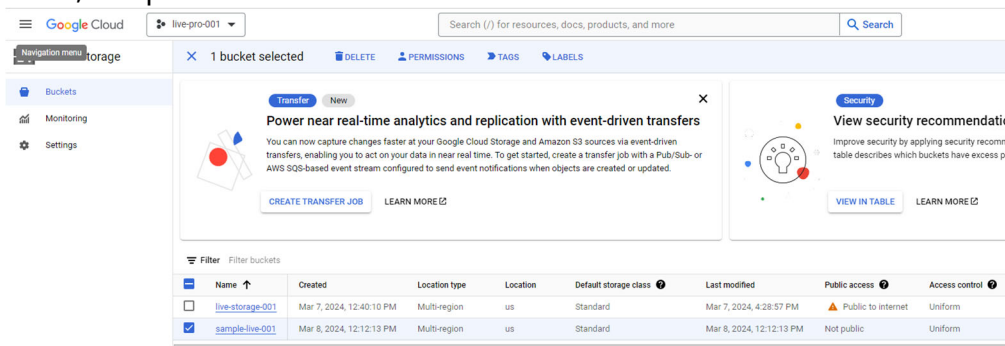
Navigate to Cloud Storage, and click Create.



Under settings enter your bucket name, region, class and access control – turn off enforce public access prevention.



Next, set permissions for the bucket.



Add allUsers to principals.

Permissions for sample-live-001 X LEARN

Public access

Not public v

PREVENT PUBLIC ACCESS

Access control

Uniform: No object-level ACLs enabled v

SWITCH TO FINE-GRAINED

Edit or delete permissions below, or select "Add Principal" to grant new access.

+ ADD PRINCIPAL

Show inherited permissions

Filter Enter property name or value ?

Grant access to "sample-live-001"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. [Learn more about IAM conditions](#)

Resource

sample-live-001

Add principals

Principals are users, groups, domains, or service accounts. [Learn more about principals in IAM](#)

New principals * ?

allUsers v

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

Role * ?

Storage Object Viewer v IAM condition (optional) ?

Grants access to view objects and their metadata, excluding ACLs. Can also list the objects in a bucket. + ADD IAM CONDITION

+ ADD ANOTHER ROLE

SAVE CANCEL

Confirm public access:

Are you sure you want to make this resource public?

Adding allUsers or allAuthenticatedUsers to this resource will make it publicly accessible to anyone on the internet. If this resource contains data that should not be made public to everyone, cancel this action to prevent public access. [Learn more](#)

CANCEL ALLOW PUBLIC ACCESS

The CLI to complete these tasks are as follows:

```
gcloud storage buckets add-iam-policy-binding gs://vod-transcoder-walkthrough --member=allUsers --role=roles/storage.objectViewer
```

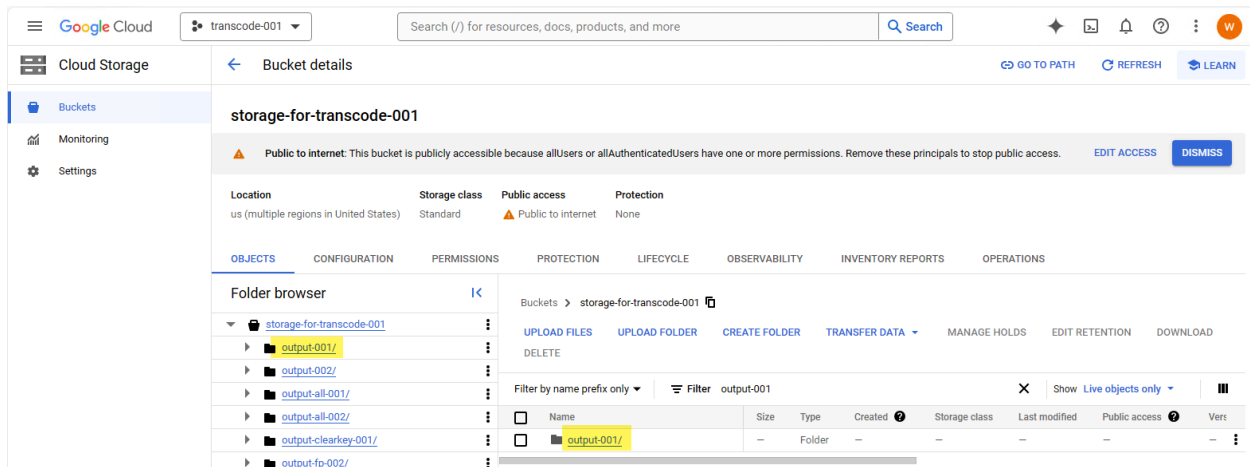
```
gcloud storage objects update gs://vod-transcoder-walkthrough/objectname --add-acl-grant=entity=AllUsers,role=READER
```

Transcoder – Create Jobs

Create the Transcoder jobs with the input, location, and output for the transcoded file:

```
gcloud transcoder jobs create \
  --input-uri="gs://storage-for-transcode-001/fragmented-bunny.mp4" \
  --location=us-east1 \
  --output-uri="gs://storage-for-transcode-001/output-001/"
```

```
gcloud transcoder jobs create --input-uri="gs://storage-for-transcode-001/fragmented-bunny.mp4" --location=us-east1 --output-uri="gs://storage-for-transcode-001/output-001/"
```



The screenshot shows the Google Cloud Storage console. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'transcode-001', a search bar, and utility icons. The main content area is titled 'Bucket details' for 'storage-for-transcode-001'. It displays metadata such as Location (us), Storage class (Standard), Public access (Public to internet), and Protection (None). Below this, there are tabs for OBJECTS, CONFIGURATION, PERMISSIONS, PROTECTION, LIFECYCLE, OBSERVABILITY, INVENTORY REPORTS, and OPERATIONS. The 'Folder browser' view shows a tree structure with folders like 'output-001', 'output-002', 'output-all-001', 'output-clearkey-001', and 'output-fp-002'. The 'output-001' folder is selected, and its contents are displayed in a table below.

Buckets > storage-for-transcode-001 > output-001

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER TRANSFER DATA MANAGE HOLDS EDIT RETENTION DOWNLOAD DELETE

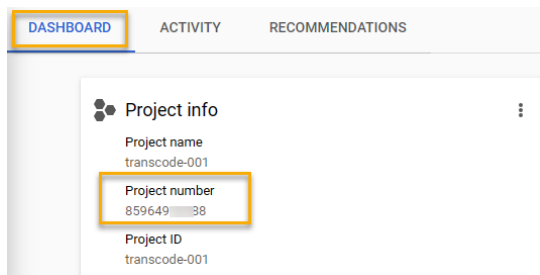
Filter by name prefix only Filter Filter objects and folders Show Live objects only

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	
<input type="checkbox"/>	audio-only0000000000.m4s	1 MB	audio/mp4	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	hd.mp4	37.1 MB	video/mp4	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	manifest.m3u8	319 B	application/vnd.apple.mpegurl	Feb 26, 2024, 6:00:19 AM	Standard	↓ ⋮
<input type="checkbox"/>	manifest.mpd	4.5 KB	application/dash+xml	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	media-hd.m3u8	1.5 KB	application/vnd.apple.mpegurl	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	media-hd0000000000.ts	39.1 MB	video/mp2t	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	media-sd.m3u8	1.5 KB	application/vnd.apple.mpegurl	Feb 26, 2024, 6:00:19 AM	Standard	↓ ⋮
<input type="checkbox"/>	media-sd0000000000.ts	10.5 MB	video/mp2t	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	sd.mp4	9 MB	video/mp4	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	video-only-hd0000000000.m4s	36.1 MB	video/mp4	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮
<input type="checkbox"/>	video-only-sd0000000000.m4s	8 MB	video/mp4	Feb 26, 2024, 6:00:18 AM	Standard	↓ ⋮

Create Templates

Here is an example of the CLI to create templates. This makes the processing of many jobs easier.

Reference the Project number from the Project Dashboard:



Create a Job Definitions JSON file to reference the secretVersion within the project and the job settings, using the template provided by Google. The encoder will then use the proper secret key version.

```
"config": {
  "elementaryStreams": [
    {
      "key": "es_video",
      "videoStream": {
        "h264": {
          "profile": "main",
          "heightPixels": 600,
          "widthPixels": 800,
          "bitrateBps": 1000000,
          "frameRate": 60
        }
      }
    },
    {
      "key": "es_audio",
      "audioStream": {
        "codec": "aac",
        "channelCount": 2,
        "bitrateBps": 160000
      }
    }
  ]
}
```



```

    }
  },
],
"encryptions": [
  {
    "id": "fairplay",
    "secretManagerKeySource": {
      "secretVersion": "projects/85964912XX8/secrets/widevine/versions/17"
    },
    "drmSystems": {
      "fairplay": {}
    },
    "sampleAes": {}
  },
  {
    "id": "widevine-cenc",
    "secretManagerKeySource": {
      "secretVersion": "projects/85964912XX8/secrets/widevine/versions/17"
    }
  },
]

```

Note: setting *segmentDuration* to "3s" or more is recommended when using the Google template.

```

"muxStreams": [
  {
    "key": "ts_fairplay",
    "container": "ts",
    "elementaryStreams": [
      "es_video",
      "es_audio"
    ],
    "segmentSettings": {
      "segmentDuration": "3s"
    }
  },
]

```

Create the template with the template ID name and indicate the location:

```
gcloud transcoder templates create walk-001 --file="drm.json" --location=us-east1
```

Identify the input, output and template ID:

```
gcloud transcoder jobs create --input-uri="gs://storage-for-transcode-001/fragmented-bunny.mp4" --location=us-east1 --output-uri="gs://storage-for-transcode-001/output-all-002/" --template-id="walk-001"
```

To see the progress of a job, utilize the jobs ID provided in the JSON response:

```
{  
  "createTime": "2024-02-28T15:15:44.027334153Z",  
  "name": "projects/85964912188/locations/us-east1/jobs/1b9e68ae-ebaa-4d45-82bf-9c861356affa",  
  "optimization": "AUTODETECT",  
  "state": "PENDING",  
  "ttlAfterCompletionDays": 30  
}
```

```
gcloud transcoder jobs describe 1b9e68ae-ebaa-4d45-82bf-9c861356affa --location=us-east1
```

Authorization

For reference, here is an example of the authorization through CLI:

```
https://accounts.google.com/o/oauth2/auth/oauthchooseaccount?response_type=code&client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fsqlservice.login%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&state=Zs5oL7b2TUVV9C1QhIoV6XaaTjxcnF&access_type=offline&code_challenge=qw-I3EHvXh90BARxC6pKNVpJdSR3lQ_oatB1gxVMdA&code_challenge_method=S256&service=iso&o2v=1&theme=glif&flowName=GeneralOAuthFlow
```

Here is an example of the connection to secrets manager:

```
https://console.cloud.google.com/security/secret-manager?referrer=search&project=transcode-001
```

Testing Playback

For details on Testing playback, please visit our website

ezdrm.com/documentation > EZDRM Testing Playback:

<https://hs.ezdrm.com/hubfs/Documentation/EZDRM-Testing-Playback-v2.2.pdf?hsLang=en>

Additional Information

For additional questions and comments please contact: simplify@ezdrm.com