

EZDRM Configuration for Google LIVE Packager Transcoding

Table of Contents

Prerequisites	3
Project CreationEnable Live Stream API	3
Create Editor and Storage Roles	3
Create Bucket (Public) and Set Permissions	4
Google Virtual Machine	6
Create Input Endpoint	7
Create a Channel	9
Start the Channel.....	10
Utilize Cloud Shell Editor to send (push) the rtmp stream. Run the following:	11
Set CORS	12
Stopping the Channel	12
Troubleshooting - Config Issues	14
Additional Information	15

Version 1.0 (published August 5, 2024)

Prerequisites

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

For more information visit:

<https://cloud.google.com/livestream/docs/quickstarts/quickstart-dash>

<https://cloud.google.com/livestream/docs/locations>

Project Creation Enable Live Stream API

Follow the Quickstart for an MPEG-DASH live stream instructions to install Google Cloud CLI, initialize and create a Google Cloud project. Note your *PROJECT_ID* and set the project. <https://cloud.google.com/livestream/docs/quickstarts/quickstart-dash>

Run this command to enable the Live Stream API:

```
gcloud services enable livestream.googleapis.com
```

Create Editor and Storage Roles

Within your project, you will need to create two access roles using the following CLI commands for each IAM role.

```
gcloud projects add-iam-policy-binding PROJECT_ID --member="user:EMAIL" --role=ROLE
```

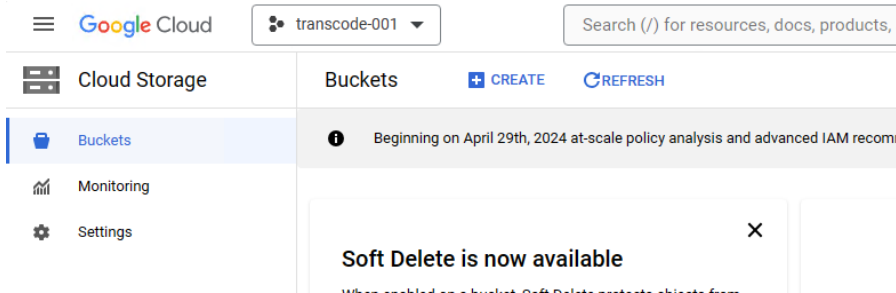
Examples:

```
gcloud projects add-iam-policy-binding live-001-416618 --member="user:sample@ezdrm.com" --role=roles/livestream.editor
```

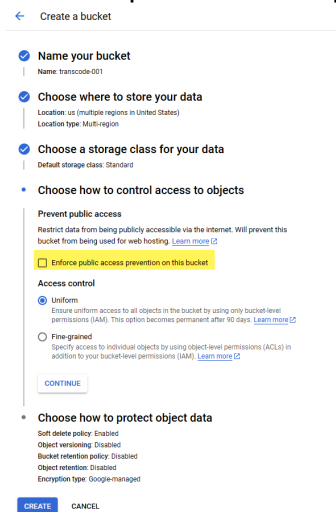
```
gcloud projects add-iam-policy-binding live-001-416618 --member="user:sample@ezdrm.com" --role=roles/storage.admin
```

Create Bucket (Public) and Set Permissions

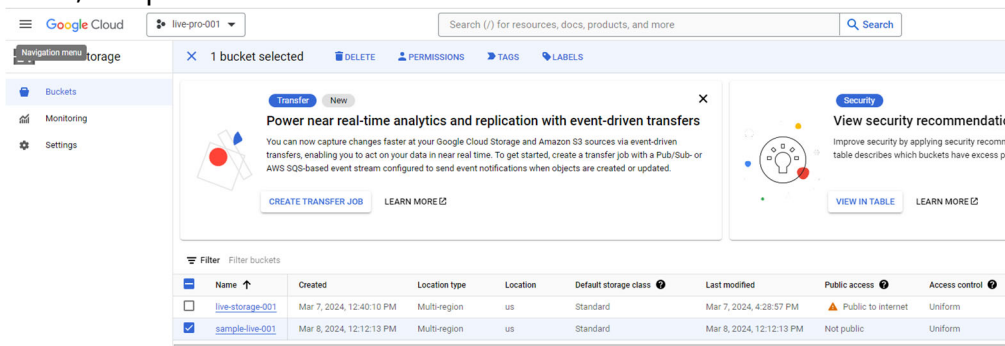
Navigate to Cloud Storage, and click Create.



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 + 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:

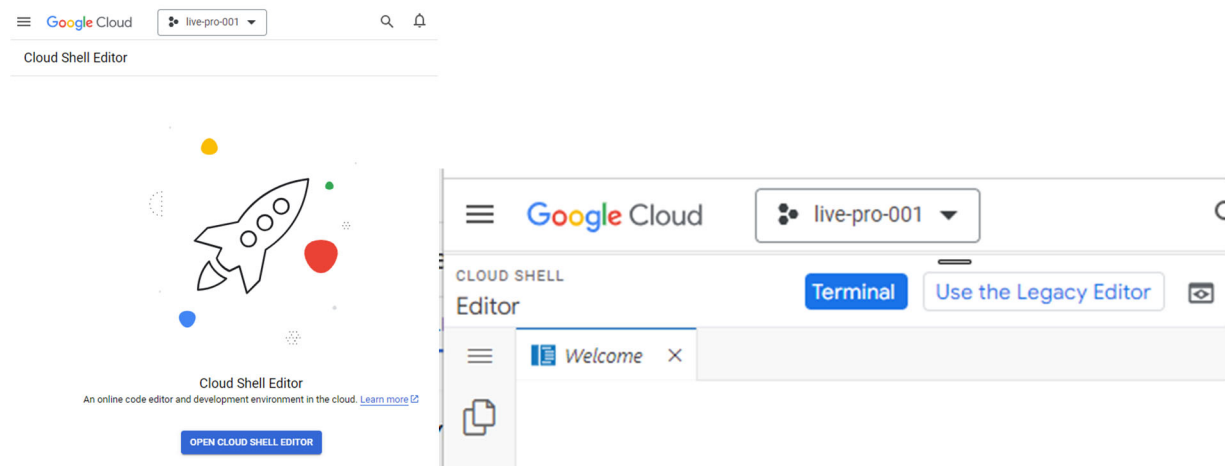
```
gcloud storage buckets add-iam-policy-binding gs://vod-transcoder-walkthrough --member=allUsers --role=roles/storage.objectViewer
```

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

Google Virtual Machine

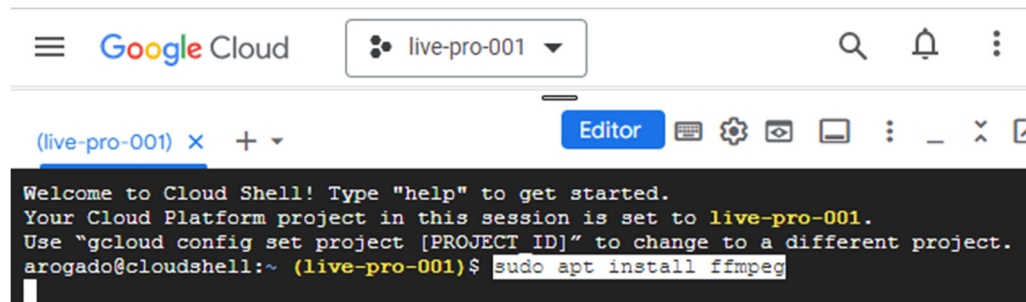
Utilize Cloud Shell Editor to create a Google virtual machine that will run ffmpeg to send a sample stream for ingestion by the live stream API. Here's the sample URL for the Cloud Shell Editor:

<https://console.cloud.google.com/cloudshelleditor?referrer=search&project=PROJECT ID&cloudshell=true>



Run this command within the editor:

```
sudo apt install ffmpeg
```




```
-InFile live-create-input-endpoint.json `
-Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/inputs?inputId=live-endpoint-test-001" | Select-Object -Expand Content
```

The response should be as follows:

```
PS D:\google> $cred = gcloud auth print-access-token
PS D:\google> $headers = @{ "Authorization" = "Bearer $cred" }
PS D:\google> Invoke-WebRequest `
>> -Method POST `
>> -Headers $headers `
>> -ContentType: "application/json; charset=utf-8" `
>> -InFile live-create-input-endpoint.json `
>> -Uri "https://livestream.googleapis.com/v1/projects/779860544924/locations/us-east1/inputs?inputId=live-endpoint-test-001" | Select-Object -Expand Content
{
  "name": "projects/779860544924/locations/us-east1/operations/operation-1709924785307-6132ae36bbf2b-6ab032bc-02826c0d"
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.video.livestream.v1.OperationMetadata",
    "createTime": "2024-03-08T19:06:27.234812114Z",
    "target": "projects/779860544924/locations/us-east1/inputs/live-endpoint-test-001",
    "verb": "create",
    "requestedCancellation": false,
    "apiVersion": "v1"
  },
  "done": false
}
```

Next, check the input endpoint status utilizing the PowerShell command below with the **Operation ID** assigned in the response (in this example: operation-1709924785307-6132ae36bbf2b-6ab032bc-02826c0d as shown above).

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }
Invoke-WebRequest `
-Method GET `
-Headers $headers `
-Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/operations/operation-1709924785307-6132ae36bbf2b-6ab032bc-02826c0d" | Select-Object -Expand Content
```

The “Done” status should be returned, Google may take up to 10 minutes to create the new endpoint.

```
PS D:\google> $cred = gcloud auth print-access-token
PS D:\google> $headers = @{ "Authorization" = "Bearer $cred" }
PS D:\google> Invoke-WebRequest `
>> -Method GET `
>> -Headers $headers `
>> -Uri "https://livestream.googleapis.com/v1/projects/779860544924/locations/us-east1/operations/operation-1709924785307-6132ae36bbf2b-6ab032bc-02826c0d" | Select-Object -Expand Content
{
  "name": "projects/779860544924/locations/us-east1/operations/operation-1709924785307-6132ae36bbf2b-6ab032bc-02826c0d",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.video.livestream.v1.OperationMetadata",
    "createTime": "2024-03-08T19:06:27.234812114Z",
    "target": "projects/779860544924/locations/us-east1/inputs/live-endpoint-test-001",
    "verb": "create",
    "requestedCancellation": false,
    "apiVersion": "v1"
  },
  "done": false
}
```


Once successful, the response should look like this.

```
PS D:\google> Invoke-WebRequest `
>> -Method GET `
>> -Headers $headers `
>> -Uri "https://livestream.googleapis.com/v1/projects/779868544924/locations/us-east1/operations/operation-1789924785307-6132ae36bbf2b-6ab032bc-02826c0d " | Select-Object -Expand Content
{
  "name": "projects/779868544924/locations/us-east1/operations/operation-1789924785307-6132ae36bbf2b-6ab032bc-02826c0d",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.video.livestream.v1.OperationMetadata",
    "createTime": "2024-03-08T19:06:27.234812114Z",
    "endTime": "2024-03-08T19:08:19.2954144593Z",
    "target": "projects/779868544924/locations/us-east1/inputs/live-endpoint-test-001",
    "verb": "create",
    "requestedCancellation": false,
    "apiVersion": "v1"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.video.livestream.v1.Input",
    "name": "projects/live-001-416618/locations/us-east1/inputs/live-endpoint-test-001",
    "createTime": "2024-03-08T19:06:27.231027243Z",
    "updateTime": "2024-03-08T19:07:42.295974651Z",
    "type": "rtmp-push",
    "uri": "rtmp://34.138.94.235/live/eaafd733-5095-4aa2-b4ac-d82cb98a7e85",
    "cred": "RD"
  }
}
```

Note the **Input_Stream_URI** as shown in the response, example:

```
rtmp://34.138.94.235/live/eaafd733-5095-4aa2-b4ac-d82cb98a7e85
```

Create a Channel

To create the channel, utilize the “live-create-channel-drm.json”. Assign an **channelID** to name the endpoint.

<https://accounts.ezdrm.com/downloads/google/live/live-create-channel-drm.zip>

Run the following **PowerShell** command:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }
Invoke-WebRequest `
  -Method POST `
  -Headers $headers `
  -ContentType: "application/json; charset=utf-8" `
  -InFile live-create-channel.json `
  -Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/channels?channelId=live-channel-test-001" | Select-Object -Expand Content
```

Next, check the channel status:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }cls
Invoke-WebRequest `
  -Method GET `
  -Headers $headers `
  -Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/channels/live-channel-test-001" | Select-Object -Expand Content
```

*Note: **streamingState** is the part of the response we are interested in.*

```
"streamingState": "STOPPED"
"streamingState": "STARTING"
```

Start the Channel

In **PowerShell**, run this command to start the channel:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
  -Method POST `
  -Headers $headers `
  -Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/channels/live-channel-test-001:start" | Select-Object -Expand Content
```

Note: "streamingState": "AWAITING_INPUT" means we're ready to receive input.

Utilize **Cloud Shell Editor** to send (push) the rtmp stream. Run the following:

```
ffmpeg -re -f lavfi -i "testsrc=size=1280x720 [out0]; sine=frequency=500 [out1]" \
-acodec aac -vcodec h264 -f flv INPUT_STREAM_URI

ffmpeg -re -f lavfi -i "testsrc=size=1280x720 [out0]; sine=frequency=500 [out1]" \
-acodec aac -vcodec h264 -f flv rtmp://34.138.94.235/live/ea4dd733-5095-4aa2-b4ac-d82cb98a7e85
```

```
walkthrough@cloudshell: (live-001-416618) $ ffmpeg -re -f lavfi -i "testsrc=size=1280x720 [out0]; sine=frequency=500 [out1]" -acodec aac -vcodec h264 -f flv rtmp://34.138.94.235/live/ea4dd733-5095-4aa2-b4ac-d82cb98a7e85
ffmpeg version 4.3.6-0+deb11u1 Copyright (c) 2000-2023 the FFmpeg developers
  built with gcc 10 (Debian 10.2.1-9)
  configuration: --prefix=/usr --extra-version=0+deb11u1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=resample
  --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbrotli --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdav1d --enable-libdrm --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgsm --enable-libharfbuzz --enable-libid3tag --enable-libiec61883 --enable-libjvarkit --enable-liblame --enable-libltdl --enable-libmp3lame --enable-libmysofa --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librav1e --enable-librist --enable-librubio --enable-librtmp --enable-libshaderc --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-ltlib --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libxml --enable-libyuv --enable-libzmq --enable-libzimg --enable-libzix --enable-libzstd --enable-libzvonk --enable-libzvbi --enable-lv2 --enable-libvmaf --enable-libvpl --enable-libvrtx --enable-libvulkan --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxvid --enable-libyasm --enable-libzmq --enable-libzimg --enable-libzix --enable-libzstd --enable-libzvonk --enable-libzvbi --enable-lv2 --enable-libvmaf --enable-libvpl --enable-libvrtx --enable-libvulkan --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxvid
  enabled-libiec61883 --enable-chromaprint --enable-frei0r --enable-libaom --enable-libass --enable-libbrotli --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libdav1d --enable-libdrm --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgsm --enable-libharfbuzz --enable-libid3tag --enable-libiec61883 --enable-libjvarkit --enable-liblame --enable-libltdl --enable-libmp3lame --enable-libmysofa --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librav1e --enable-librist --enable-librubio --enable-librtmp --enable-libshaderc --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-ltlib --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libxml --enable-libyuv --enable-libzmq --enable-libzimg --enable-libzix --enable-libzstd --enable-libzvonk --enable-libzvbi --enable-lv2 --enable-libvmaf --enable-libvpl --enable-libvrtx --enable-libvulkan --enable-libx264 --enable-libx265 --enable-libxavs2 --enable-libxvid
  libavutil 56. 51.100 / 56. 51.100
  libavcodec 58. 91.100 / 58. 91.100
  libavformat 58. 45.100 / 58. 45.100
  libavdevice 58. 10.100 / 58. 10.100
  libavfilter 7. 65.100 / 7. 65.100
  libavresample 4. 0. 0 / 4. 0. 0
  libswscale 5. 7.100 / 5. 7.100
  libswresample 3. 7.100 / 3. 7.100
  libpostproc 55. 7.100 / 55. 7.100
Input #0: lavfi, from 'testsrc=size=1280x720 [out0]; sine=frequency=500 [out1]':
  Duration: N/A, start: 0.000000, bitrate: 705 kb/s
  Stream #0:0: Video: rawvideo (RGB24) / 0x18429730, rgb24, 1280x720 [SAR 1:1 DAR 16:9], 25 tbr, 25 tbn, 25 tbc
  Stream #0:1: Audio: pcm_s16le, 44100 Hz, mono, s16, 705 kb/s
Stream mapping:
  Stream #0:0 -> #0:0 (rawvideo (native) -> h264 (libx264))
  Stream #0:1 -> #0:1 (pcm_s16le (native) -> aac (native))
Press [q] to stop, [?] for help
[libx264 @ 0x56e77ae2740] using x86-1/1
[libx264 @ 0x56e77ae2740] using cpu capabilities: MMX2 SSE2Fast SSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[libx264 @ 0x56e77ae2740] profile High 4:4:4 Predictive, level 3.1, 4:4:4, 8-Bit
[libx264 @ 0x56e77ae2740] 644 - core 140 25011 cdec9a93 - h264/avc1-4 AVC codec - Copyleft 2003-2020 - http://www.videolan.org/vlc.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_re
c=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=4 threads=3 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt
=1 b_bias=0 direct=1 weight=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc_2pass=0 rc_strategy=1 crf=23.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 qg=111.00
Output #0: flv, to 'rtmp://34.138.94.235/live/ea4dd733-5095-4aa2-b4ac-d82cb98a7e85':
  Metadata:
    encoder         : Lavf58.45.100
  Stream #0:0: Video: h264 (libx264) ([7][0][0][0] / 0x0007), yuv444p, 1280x720 [SAR 1:1 DAR 16:9], q=1-1, 25 fps, 1k tbn, 25 tbc
  Metadata:
    encoder         : Lavf58.91.100 libx264
  Side data:
    qg: bitrate max/min/avg: 0/0/0 buffer size: 0 vbr_delay: N/A
  Stream #0:1: Audio: aac (LC) ([10][0][0][0] / 0x0000A), 44100 Hz, mono, fltp, 69 kb/s
  Metadata:
    encoder         : Lavf58.91.100 aac
  Stream #0:2: Audio: pcm_s16le, 44100 Hz, stereo, s16, 176 kb/s
  Metadata:
    encoder         : Lavf58.91.100 pcm_s16le
  Frame= 227 fps= 25 q=29.0 size= 138kB time=00:00:09.01 bitrate= 125.6kb/s speed=0.993x
```

Now that the stream is running and sending to the API, you can view the MPD in the storage bucket.

Google Cloud | live-001 | Search (/) for resources, docs, products, and more | Search

Cloud Storage | Bucket details | REF

Buckets | live-storage-sample-001

Monitoring | Public to internet: This bucket is publicly accessible because allUsers or allAuthenticatedUsers have one or more permissions. Remove these principals to stop public access. EDIT A

Settings

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Public to internet	None

OBJECTS | CONFIGURATION | PERMISSIONS | PROTECTION | LIFECYCLE | OBSERVABILITY | INVENTORY REPORTS | OPERATIONS

Buckets > live-storage-sample-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 obje

Name	Size	Type	Created	Storage class	Last modified	Public access
main.mpd	1.6 KB	application/dash+xml	Mar 8, 2024, 3:05:05 PM	Standard	Mar 8, 2024, 3:05:05 PM	Public to internet Copy URL
mux_audio/	-	Folder	-	-	-	-
mux_video/	-	Folder	-	-	-	-

Copy the .mpd URL for playback.

```
https://storage.googleapis.com/live-storage-sample-001/main.mpd
```

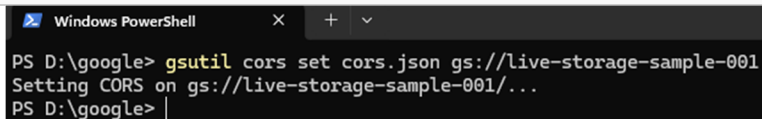
For details on Testing playback, please visit our website **ezdrm.com/documentation > EZDRM Testing Playback:**
<https://hs.ezdrm.com/hubfs/Documentation/EZDRM-Testing-Playback-v2.2.pdf?hsLang=en>

Set CORS

Run this command in PowerShell to set CORS:

```
gsutil cors set JSON_FILE_NAME.json gs://STORAGE_BUCKET
```

```
gsutil cors set cors.json gs://live-storage-sample-001
```



```
Windows PowerShell x + v  
PS D:\google> gsutil cors set cors.json gs://live-storage-sample-001  
Setting CORS on gs://live-storage-sample-001/...  
PS D:\google> |
```

Stopping the Channel

In PowerShell run the following command to stop the channel:

```
In powershell  
$cred = gcloud auth print-access-token  
$headers = @{ "Authorization" = "Bearer $cred" }  
Invoke-WebRequest ` -Method POST `
```

```
-Headers $headers `
-Uri "https://livestream.googleapis.com/v1/projects/PROJECT_ID/locations/us-east1/channels/Live-channel-test-001:stop" | Select-Object -Expand Content
```

The response should be as follows:

```
PS D:\google> $cred = gcloud auth print-access-token
PS D:\google> $headers = @{ "Authorization" = "Bearer $cred" }
PS D:\google> Invoke-WebRequest `
>> -Method POST `
>> -Headers $headers `
>> -Uri "https://livestream.googleapis.com/v1/projects/779860544924/locations/us-east1/channels/live-channel-test-001:stop" | Select-Object -Expand Content
{
  "name": "projects/779860544924/locations/us-east1/operations/operation-1709928995079-6132bde57c60a-1ba74065-826bb2ff",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.video.livestream.v1.OperationMetadata",
    "createTime": "2024-03-08T20:16:35.096332041Z",
    "target": "projects/779860544924/locations/us-east1/channels/live-channel-test-001",
    "verb": "stop",
    "requestedCancellation": false,
    "apiVersion": "v1"
  },
  "done": false
}
```

Stopping the channel will also stop ffmpeg.

Troubleshooting - Config Issues

To troubleshoot configuration issues, run in **PowerShell**:

```
gcloud init
```

The following response will help with troubleshooting:

```
PS D:\google> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
PS D:\google> gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [new] are:
accessibility:
  screen_reader: 'False'
core:
  account: arogado@ezdrm.com
  disable_usage_reporting: 'True'
  project: live-pro-001

Pick configuration to use:
[1] Re-initialize this configuration [new] with new settings
[2] Create a new configuration
[3] Switch to and re-initialize existing configuration: [default]
[4] Switch to and re-initialize existing configuration: [walk]
Please enter your numeric choice: 2

Enter configuration name. Names start with a lower case letter and contain only lower case letters a-z, digits 0-9, and
hyphens '-': walking
Your current configuration has been set to: [walking]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
[1] arogado@ezdrm.com
[2] walkthrough@ezdrm.com
[3] Log in with a new account
Please enter your numeric choice: 2

You are logged in as: [walkthrough@ezdrm.com].

Pick cloud project to use:
[1] just-genius-415714
[2] live-001-416618
[3] transcode-001
[4] Enter a project ID
[5] Create a new project
Please enter numeric choice or text value (must exactly match list item): 2

Your current project has been set to: [live-001-416618].
```

Additional Information

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