Video Intelligence: Qwik Start

Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

1. Click **Activate Cloud Shell** at the top of the Google Cloud console.

2. Click through the following windows:

    - Continue through the Cloud Shell information window.
    - Authorize Cloud Shell to use your credentials to make Google Cloud API calls.

When you are connected, you are already authenticated, and the project is set to your **Project_ID**, `qwiklabs-gcp-00-0e2c8dc69a39`. The output contains a line that declares the **Project_ID** for this session:

```
Your Cloud Platform project in this session is set to qwiklabs-gcp-00-
0e2c8dc69a39
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

3. (Optional) You can list the active account name with this command:

```
gcloud auth list
```

4. Click **Authorize**.

Output:

```
ACTIVE: *
ACCOUNT: student-01-91d7fd156d5d@qwiklabs.net

To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

5. (Optional) You can list the project ID with this command:

```
gcloud config list project
```

Output:

```
[core]
project = qwiklabs-gcp-00-0e2c8dc69a39
```

Note: For full documentation of `gcloud`, in Google Cloud, refer to the gcloud CLI overview guide.

Task 1. Set up authorization

For this lab, you create and use a service account that is tied to your Google Cloud project for authorization.

1. In Cloud Shell, run the following command to create a new service account named `quickstart`:

```
gcloud iam service-accounts create quickstart
```

2. Create a service account key file, replacing `<your-project-123>` with your Project ID:

```
gcloud iam service-accounts keys create key.json --iam-account
quickstart@<your-project-123>.iam.gserviceaccount.com
```

3. Now authenticate your service account, passing the location of your service account key file:

```
gcloud auth activate-service-account --key-file key.json
```

4. Obtain an authorization token using your service account:

```
gcloud auth print-access-token
```

The token will print in the output, and you'll be using it in a future step.

Task 2. Make an annotate video request

Note: In this lab, the Cloud Video Intelligence API is already enabled for you.

1. Run this command to create a JSON request file with the following text, and save it as `request.json` :

```
cat > request.json <<EOF
{
   "inputUri":"gs://spls/gsp154/video/train.mp4",
   "features": [
       "LABEL_DETECTION"
   ]
}
EOF
```

Note: To make the process simpler, a public video of a train available to your project is used as the value for your `inputUri`. If preferred or running in a personal project, any video can be used in place by uploading it to Cloud Storage and providing its Cloud Storage URI (format: `gs://bucket/object`) for the value of `inputUri`.

2. Use `curl` to make a `videos:annotate` request passing the filename of the entity request:

```
curl -s -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer '$(gcloud auth print-access-token)'' \
    'https://videointelligence.googleapis.com/v1/videos:annotate' \
    -d @request.json
```

The Video Intelligence API creates an operation to process your request. You should now see a response that includes your operation name, which should look similar to this one:

```
{
  "name": "projects/474887704060/locations/asia-
east1/operations/16366331060670521152"
}
```

You will use this operation name, locations and projects in the future step.

3. Use this script to request information on the operation by calling the `v1.operations` endpoint. Replace the `PROJECTS`, `LOCATIONS` and `OPERATION_NAME` with the value you just received in the previous command:

```
curl -s -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer '$(gcloud auth print-access-token)'' \

'https://videointelligence.googleapis.com/v1/projects/PROJECTS/location
s/LOCATIONS/operations/OPERATION_NAME'
```

You'll now see information related to your operation. If the operation has completed, a `done` field is included and set to `true`:

```
{
  "name": "projects/425437283751/locations/asia-
east1/operations/17938636079131796601",
  "metadata": {
    "@type":
"type.googleapis.com/google.cloud.videointelligence.v1.Annota
tionProgressMetadata",
    "progressMetadata": [
      {
        "inputUri": "gs://spls/gsp154/video/train.mp4",
        "startTime": "2016-09-22T21:41:56.766091Z",
        "lastUpdateTime": "2016-09-22T21:42:03.889743Z"
      }
    ]
  },
  ...
}
```

4. After giving the request some time (about a minute, typically), re-run the command and the same request returns annotated results:

```
{
  "name": "projects/425437283751/locations/asia-
east1/operations/17938636079131796601",
  "metadata": {
    "@type":
"type.googleapis.com/google.cloud.videointelligence.v1.AnnotateVideoPro
gress",
    "annotationProgress": [
      {
        "inputUri": "/spls/gsp154/video/train.mp4",
        "progressPercent": 100,
        "startTime": "2017-02-17T22:39:00.333942Z",
        "updateTime": "2017-02-17T22:39:11.414399Z"
      }
    ]
  },
  "done": true,
  "response": {
    "@type":
"type.googleapis.com/google.cloud.videointelligence.v1.AnnotateVideoRes
ponse",
    "annotationResults": [
      {
        "inputUri": "/spls/gsp154/video/train.mp4",
        "segmentLabelAnnotations": [
          {
            "entity": {
              "entityId": "/m/01yrx",
              "languageCode": "en-US"
            },
            "segments": [
              {
                "segment": {
                  "startTimeOffset": "0s",
                  "endTimeOffset": "14.833664s"
                },
                "confidence": 0.98509187
              }
            ]
```

```
           },
          ...
```

You've sent your first request to Cloud Video Intelligence API.