**Getting Environment Ready**

* Turn on the following APIs in Google Cloud Console.
  1. Vision API
  2. BigQuery API
  3. Cloud Run API
  4. Functions API
  5. Storage API
* Create 3 Buckets in the GCS as shown in the figure below
  1. First one is where the user will upload the pdf files.
     + Ex. I named it “Dark-foundry-340620-companydata”
  2. Second one is where the processed files will be stored.
     + Ex. I named it “Dark-foundry-340620-companydata-processed”
  3. Third one is used to store Cloud Vision Result temporarily (Must be always empty)
     + Ex. I named it “dark-foundry-340620-tmp”

Graphical user interface, application

Description automatically generated

* Create a BigQuery dataset to store results. Ex. I named it “companydata”

**Graphical user interface, text, application

Description automatically generated**

* Create a new service account and give it the role of “Owner” and download its “key”. I have downloaded it and named it “dark-foundry-340620-ebf4ab8b7ad2.json”. Replace this key file with your own.
* Note: Giving “Owner” Role is necessary to use Vision API. So, no workaround exists yet.

**Making Necessary Code Changes**

* In the given folder “data-cleaner” open extract.py.
* Change the following lines as shown in figure with your own info.
  + “BUCKET” = “your bucket name where user will store pdf files”
  + “PROCESSED\_BUCKET” = “your bucket name where files will be stored after processing”
  + “TABLE\_ID” = “your project name.your bq dataset name.your bq table name”
  + “LOGIN\_FILE” = “The key file that you generated in the previous section” Make sure that it is the same path as extract.py

**Text

Description automatically generated**

* In target.py, change the following things as shown in figure.
  + “gcs\_destination\_uri” = “name of the temporary bucket to be used by VISION API prefixed with gs://”
  + “LOGIN\_FILE” = “name of your json key file generated in the previous section”
  + “TMP\_BUCKET\_NAME = “name of the temporary bucket to be used by VISION API”

**Text

Description automatically generated**

**Deployment**

* Install gcloud CLI on your local pc.
* Open Command Prompt and navigate to the folder where you made your code changes.
* Type this command: “gcloud run deploy”
* Press return.
* Login to your google account if prompted.
* Allow unauthorized access.
* When deployment is succeeded, you’ll receive a URI that will run the code if it is hit by a get request. **COPY THAT URL.**

**Trigger Function**

* Go to the main.py in function-source folder.
  + Change URL in the following line as your own URL that you got from Cloud Run.
  + It should be like “requests.get(‘your url’)”



* Create a new Cloud Function select Python 3.x runtime.
* Select Trigger Type: Cloud Storage and Event Type: Finalize/Create.

Application

Description automatically generated with low confidence

* Select Bucket as your bucket where user will store PDF files.
* Click Save.
* Copy the code from “function-source/main.py” to the cloud function’s main.py.
* Copy the code from “function-source/requirements.txt” to cloud function’s requirements.txt.
* Deploy the function.
* Test it by uploading the PDF file in your PDF bucket and refresh the page after a couple of minutes it will be gone and new rows will be added to your big query table.