Integrate with Machine Learning APIs Challenge Lab

Author: Vedant Kakde | **GitHub Profile:** github.com/vedant-kakde | **LinkedIn Profile:** linkedin.com/in/vedant-kakde/

Task 1: Configure a service account to access the Machine Learning APIs, BigQuery, and Cloud Storage

In the Cloud Shell, create a new service account that provides credentials for the script using the following commands. (Remember to replace `<Your_Project_ID>` with your GCP project ID)

```
export PROJECT=<Your_Project_ID>


gcloud iam service-accounts create my-account --display-name my-account
```

Once you have created the account, bind the BigQuery Admin and Cloud Storage Admin roles to the Service Account to provide the IAM permissions required to process files from Cloud Storage and insert the result data into a BigQuery table.

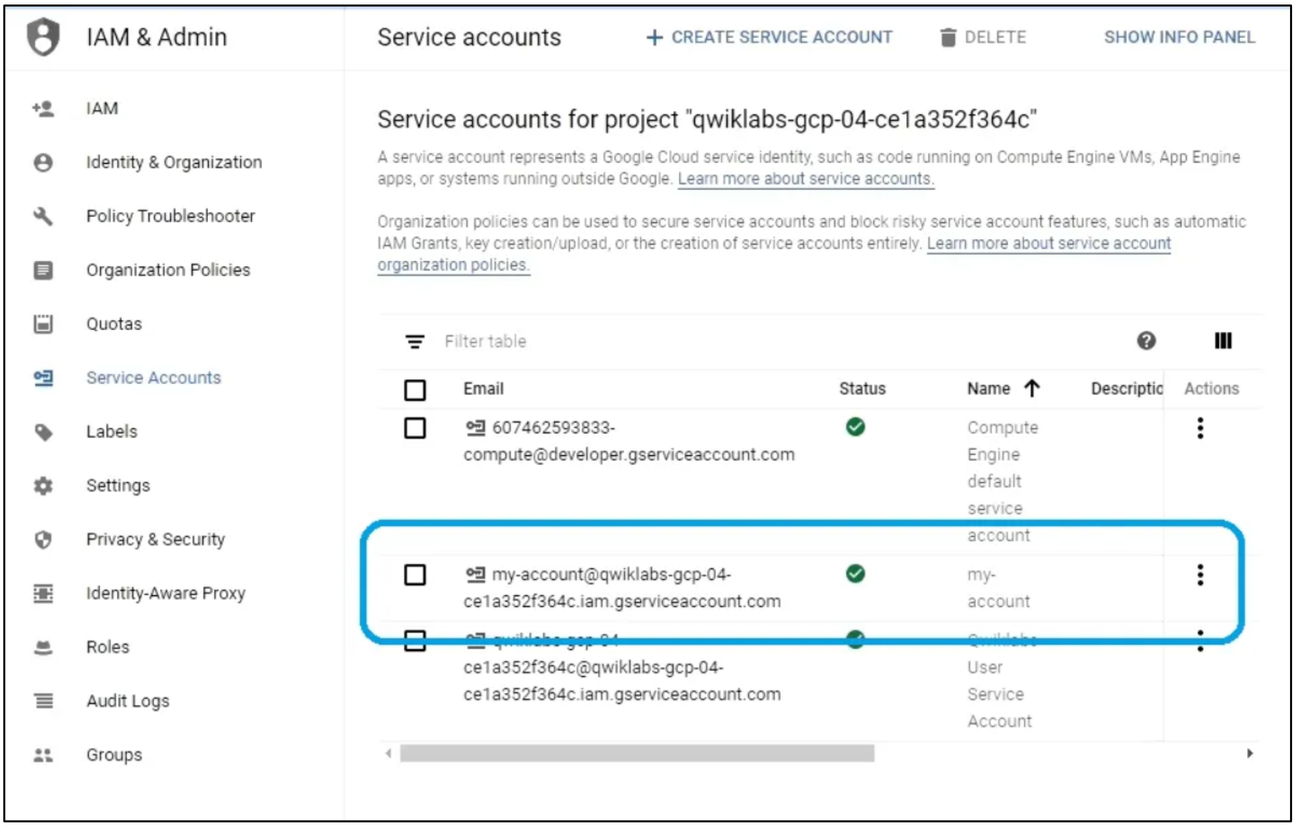
```
gcloud projects add-iam-policy-binding $PROJECT --member=serviceAccount:my-account@$PROJECT.iam.gserviceaccount.com --role=roles/bigquery.admin


gcloud projects add-iam-policy-binding $PROJECT --member=serviceAccount:my-account@$PROJECT.iam.gserviceaccount.com --role=roles/storage.admin
```

Task 2: Create and download a credential file for your Service Account

Run the following commands to download the JSON format IAM credentials file for the service account, and configure the name of the credential file as an environment variable.

```
gcloud iam service-accounts keys create key.json --iam-account=my-account@$PROJECT.iam.gserviceaccount.com


export GOOGLE_APPLICATION_CREDENTIALS=key.json
```

In the Cloud Console, navigate to **IAM & Admin** > **Service Accounts** to confirm the status of the service account (`my-account`).



IAM & Admin

Service accounts   + CREATE SERVICE ACCOUNT   DELETE   SHOW INFO PANEL

- IAM
- Identity & Organization
- Policy Troubleshooter
- Organization Policies
- Quotas
- Service Accounts
- Labels
- Settings
- Privacy & Security
- Identity-Aware Proxy
- Roles
- Audit Logs
- Groups

Service accounts for project "qwiklabs-gcp-04-ce1a352f364c"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

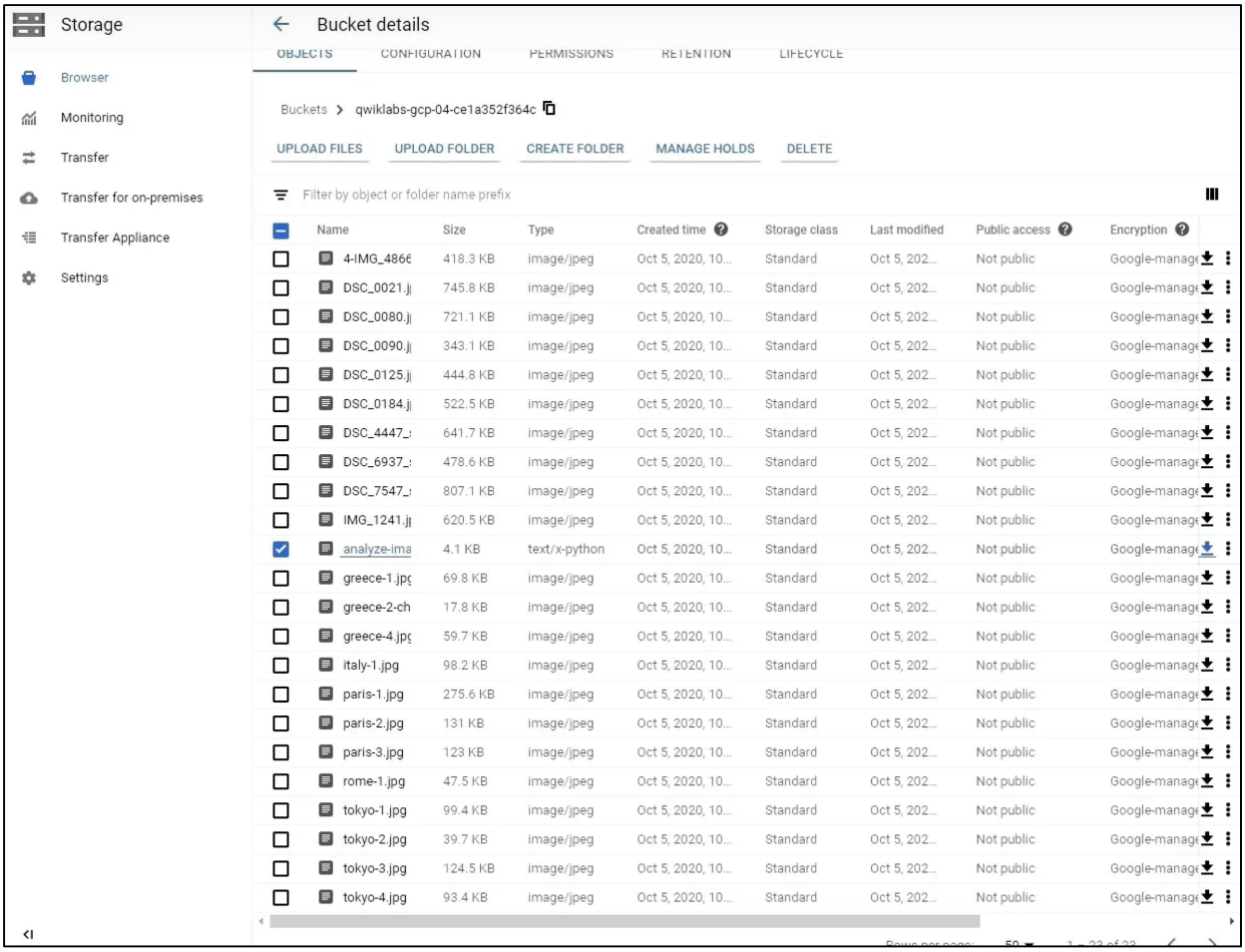
Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

Filter table

□	Email	Status	Name ↑	Descriptio	Actions
□	607462593833-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		⋮
□	my-account@qwiklabs-gcp-04-ce1a352f364c.iam.gserviceaccount.com	✓	my-account		⋮
□	qwiklabs-gcp-04-ce1a352f364c@qwiklabs-gcp-04-ce1a352f364c.iam.gserviceaccount.com	✓	Qwiklabs User Service Account		⋮

Task 3: Modify the Python script to extract text from image files

Navigate to **Storage** in the Cloud Console, then click on the bucket name to explore the image files and the Python script that have been provided for you.



Storage

- Browser
- Monitoring
- Transfer
- Transfer for on-premises
- Transfer Appliance
- Settings

← Bucket details

OBJECTS   CONFIGURATION   PERMISSIONS   RETENTION   LIFECYCLE

Buckets > qwiklabs-gcp-04-ce1a352f364c

UPLOAD FILES   UPLOAD FOLDER   CREATE FOLDER   MANAGE HOLDS   DELETE

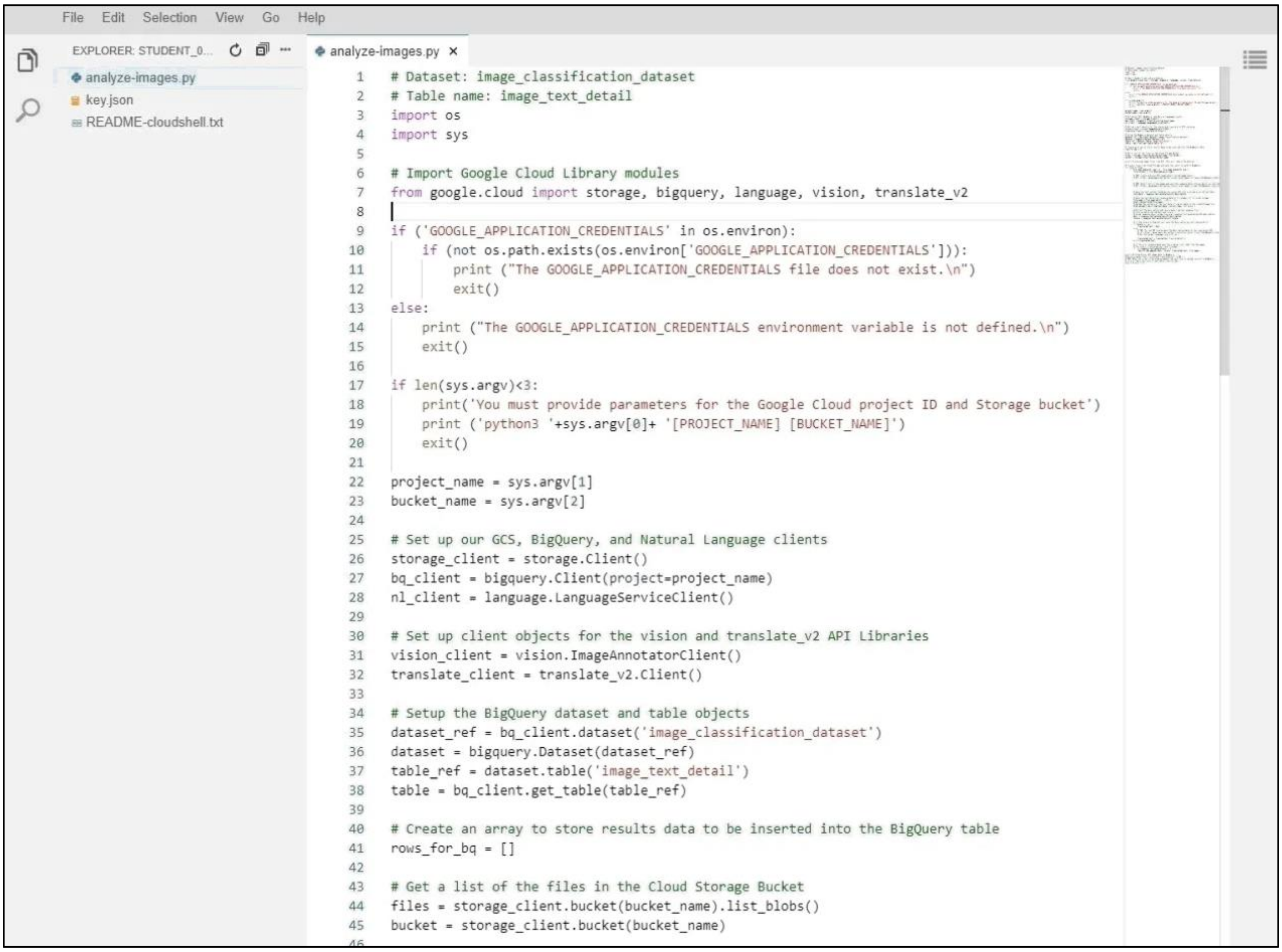
Filter by object or folder name prefix

	Name	Size	Type	Created time	Storage class	Last modified	Public access	Encryption
□	4-IMG_4866	418.3 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_0021.j	745.8 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_0080.j	721.1 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_0090.j	343.1 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_0125.j	444.8 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_0184.j	522.5 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_4447_	641.7 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_6937_	478.6 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	DSC_7547_	807.1 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	IMG_1241.j	620.5 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
☑	analyze-ima	4.1 KB	text/x-python	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	greece-1.jpg	69.8 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	greece-2-ch	17.8 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	greece-4.jpg	59.7 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	italy-1.jpg	98.2 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	paris-1.jpg	275.6 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	paris-2.jpg	131 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	paris-3.jpg	123 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	rome-1.jpg	47.5 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	tokyo-1.jpg	99.4 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	tokyo-2.jpg	39.7 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	tokyo-3.jpg	124.5 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage
□	tokyo-4.jpg	93.4 KB	image/jpeg	Oct 5, 2020, 10...	Standard	Oct 5, 202...	Not public	Google-manage

Run the following **gsutil** command to copy the file `analyze-images.py` from the Cloud Storage bucket into the Cloud Shell.

```
gsutil cp gs://$PROJECT/analyze-images.py .
```

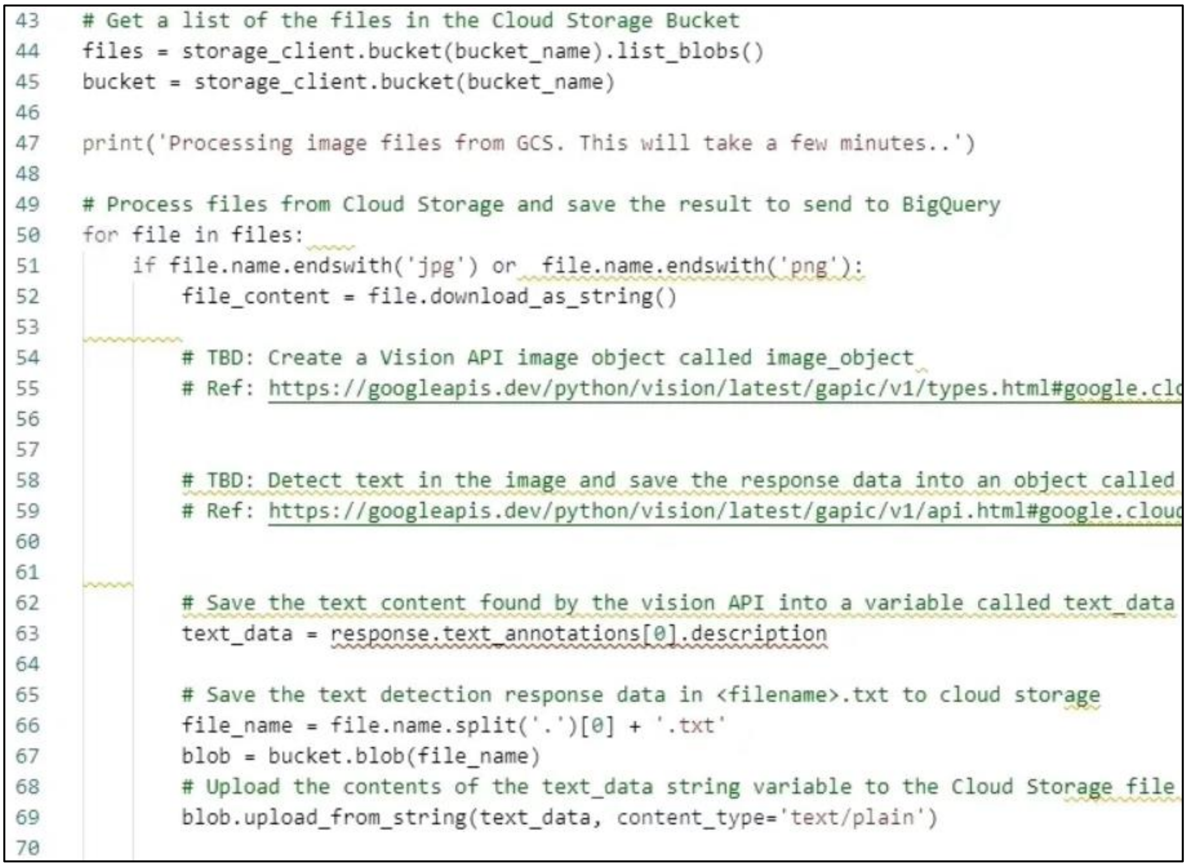
Open the Cloud Shell Editor to review and edit the script file.



```
1   # Dataset: image_classification_dataset
2   # Table name: image_text_detail
3   import os
4   import sys
5
6   # Import Google Cloud Library modules
7   from google.cloud import storage, bigquery, language, vision, translate_v2
8
9   if ('GOOGLE_APPLICATION_CREDENTIALS' in os.environ):
10      if (not os.path.exists(os.environ['GOOGLE_APPLICATION_CREDENTIALS'])):
11          print ("The GOOGLE_APPLICATION_CREDENTIALS file does not exist.\n")
12          exit()
13  else:
14      print ("The GOOGLE_APPLICATION_CREDENTIALS environment variable is not defined.\n")
15      exit()
16
17  if len(sys.argv)<3:
18      print('You must provide parameters for the Google Cloud project ID and Storage bucket')
19      print ('python3 '+sys.argv[0]+ '[PROJECT_NAME] [BUCKET_NAME]')
20      exit()
21
22  project_name = sys.argv[1]
23  bucket_name = sys.argv[2]
24
25  # Set up our GCS, BigQuery, and Natural Language clients
26  storage_client = storage.Client()
27  bq_client = bigquery.Client(project=project_name)
28  nl_client = language.LanguageServiceClient()
29
30  # Set up client objects for the vision and translate_v2 API Libraries
31  vision_client = vision.ImageAnnotatorClient()
32  translate_client = translate_v2.Client()
33
34  # Setup the BigQuery dataset and table objects
35  dataset_ref = bq_client.dataset('image_classification_dataset')
36  dataset = bigquery.Dataset(dataset_ref)
37  table_ref = dataset.table('image_text_detail')
38  table = bq_client.get_table(table_ref)
39
40  # Create an array to store results data to be inserted into the BigQuery table
41  rows_for_bq = []
42
43  # Get a list of the files in the Cloud Storage Bucket
44  files = storage_client.bucket(bucket_name).list_blobs()
45  bucket = storage_client.bucket(bucket_name)
46
```

There are three unfinished parts in the script that you must complete to make the correct Machine Learning API calls. All of them are preceded with a comment using the label `# TBD:`. You will need to make use of the Vision API, the Translation API, and the BigQuery API. The import of the Google Cloud Library modules for the required APIs has been done in line 7 of the script file. Note down the name of their API clients that have also been declared from lines 25 to 32.

In Task 3, you need to add your codes to the following part of the script file.



```
43   # Get a list of the files in the Cloud Storage Bucket
44   files = storage_client.bucket(bucket_name).list_blobs()
45   bucket = storage_client.bucket(bucket_name)
46
47   print('Processing image files from GCS. This will take a few minutes..')
48
49   # Process files from Cloud Storage and save the result to send to BigQuery
50   for file in files:
51       if file.name.endswith('jpg') or  file.name.endswith('png'):
52           file_content = file.download_as_string()
53
54           # TBD: Create a Vision API image object called image_object
55           # Ref: https://googleapis.dev/python/vision/latest/gapic/v1/types.html#google.cl
56
57
58           # TBD: Detect text in the image and save the response data into an object called
59           # Ref: https://googleapis.dev/python/vision/latest/gapic/v1/api.html#google.cloud
60
61
62           # Save the text content found by the vision API into a variable called text_data
63           text_data = response.text_annotations[0].description
64
65           # Save the text detection response data in <filename>.txt to cloud storage
66           file_name = file.name.split('.')[0] + '.txt'
67           blob = bucket.blob(file_name)
68           # Upload the contents of the text_data string variable to the Cloud Storage file
69           blob.upload_from_string(text_data, content_type='text/plain')
70
```

`TBD:` `Create a Vision API image object called image_object` `Ref:` `google.cloud.vision_v1.types.Image`

```
        image_object = vision.types.Image()

        image_object.content = file_content
```

where `file_content` is the image content extracted from a JEPG or a PNG file via lines 51 and 52 of the script.

`TBD:` `Detect text in the image and save the response data into an object called response` `Ref:` `google.cloud.vision_v1.ImageAnnotatorClient.document_text_detection`

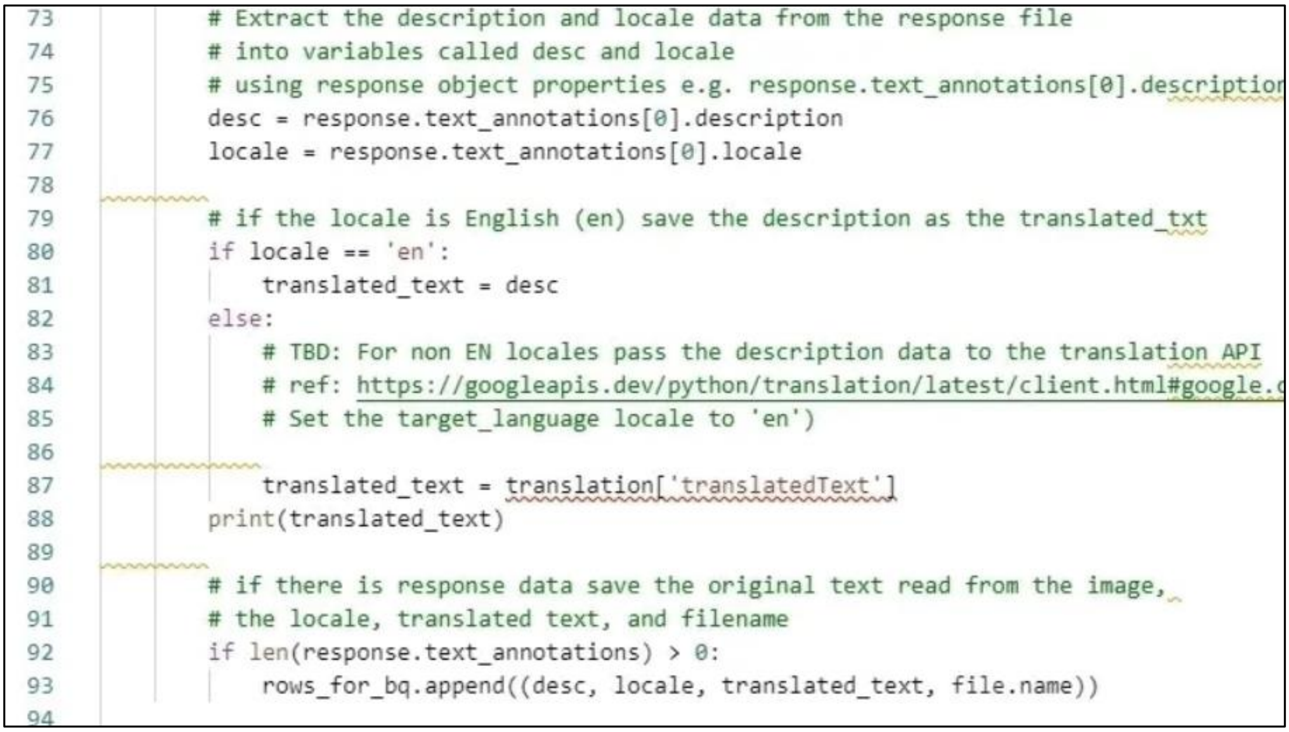
```
        response = vision_client.document_text_detection(image=image_object)
```

This line uses the Cloud Vision API to extract text data from each image inside the for loop.

Note: Make sure that you indent the codes correctly.

Task 4: Modify the Python script to translate the text using the Translation API

In Task 4, you need to add your codes to the following part of the script file.



```
        # Extract the description and locale data from the response file
        # into variables called desc and locale
        # using response object properties e.g. response.text_annotations[0].descriptior
        desc = response.text_annotations[0].description
        locale = response.text_annotations[0].locale

        # if the locale is English (en) save the description as the translated_txt
        if locale == 'en':
            translated_text = desc
        else:
            # TBD: For non EN locales pass the description data to the translation API
            # ref: https://googleapis.dev/python/translation/latest/client.html#google.c
            # Set the target_language locale to 'en')

            translated_text = translation['translatedText']
        print(translated_text)

        # if there is response data save the original text read from the image,
        # the locale, translated text, and filename
        if len(response.text_annotations) > 0:
            rows_for_bq.append((desc, locale, translated_text, file.name))
```

TBD: For non EN locales pass the description data to the translation API **Ref:** [google.cloud.translate_v2.client.Client.translate]()

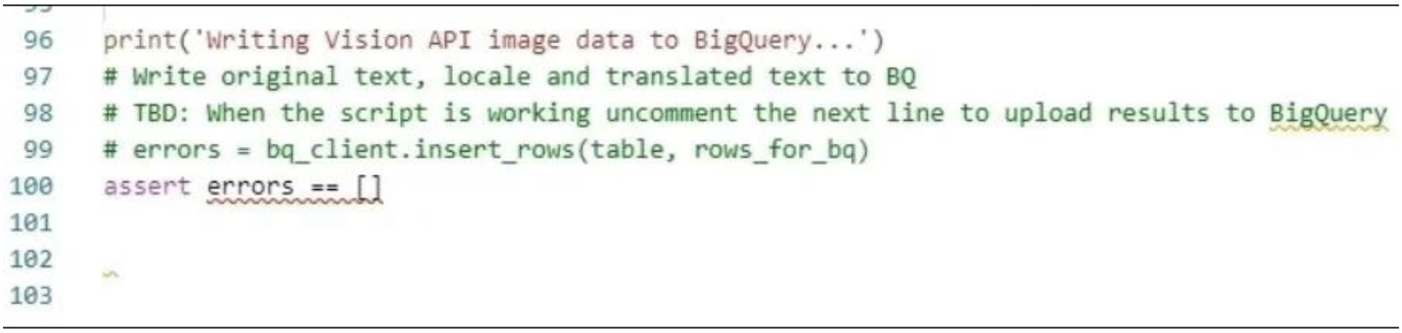
```
            # Set the target_language locale to 'en')

            translation = translate_client.translate(desc, target_language='en')
```

This line uses the Translation API to translate the non-English text into English.

Note: Make sure that you indent the codes correctly.

Task 5: Identify the most common non-English language used in the signs in the data set

In Task 5, you need to remove the comment characters to enable the line of code in the following part of the script.



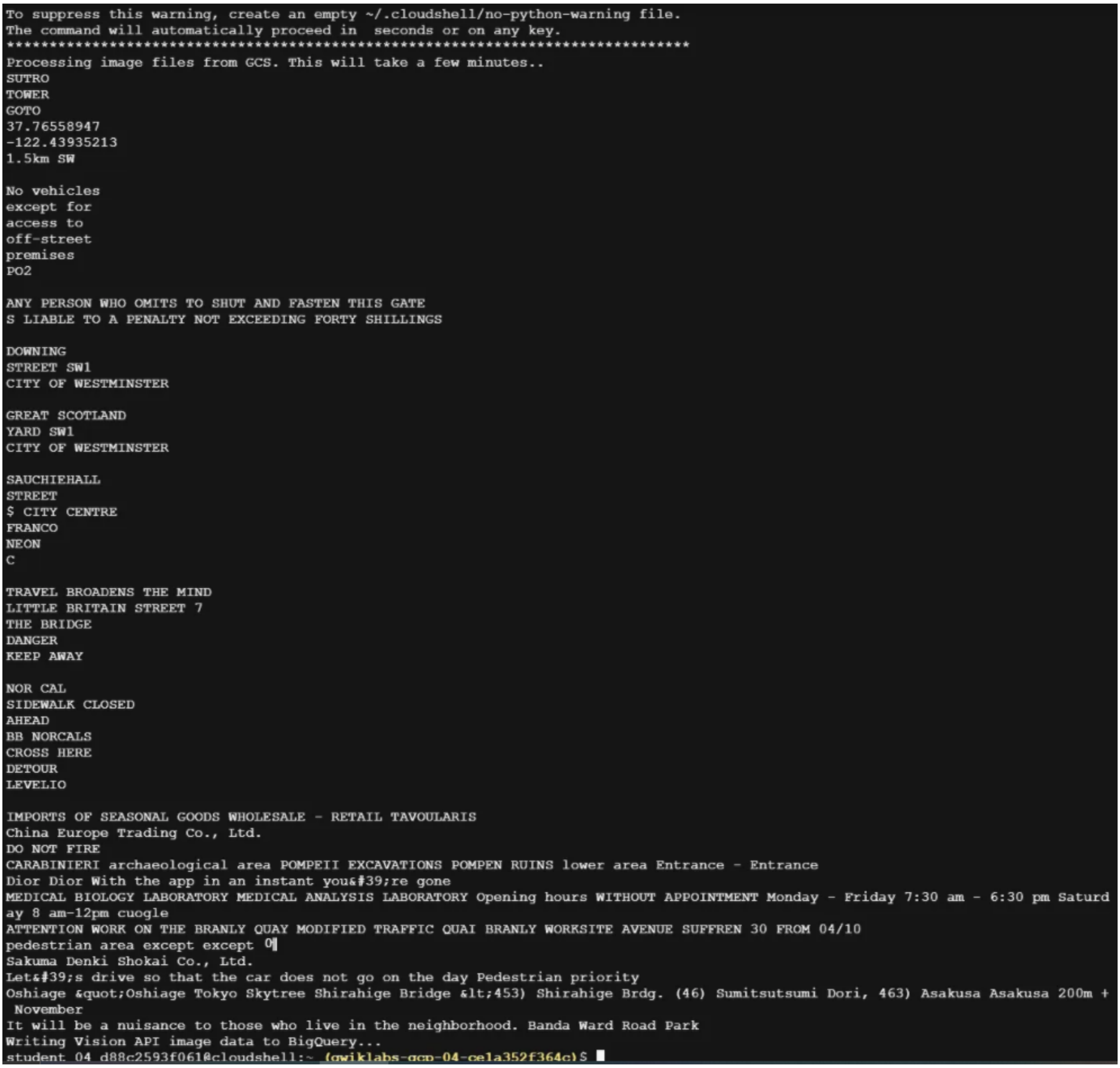
```
print('Writing Vision API image data to BigQuery...')
# Write original text, locale and translated text to BQ
# TBD: When the script is working uncomment the next line to upload results to BigQuery
# errors = bq_client.insert_rows(table, rows_for_bq)
assert errors == []



```

Process the image files using the updated Python

Save the changes and then run the modified script file in the Cloud Shell:

```
export BUCKET=$PROJECT

python analyze-images.py $PROJECT $BUCKET
```

You should see an output like that:

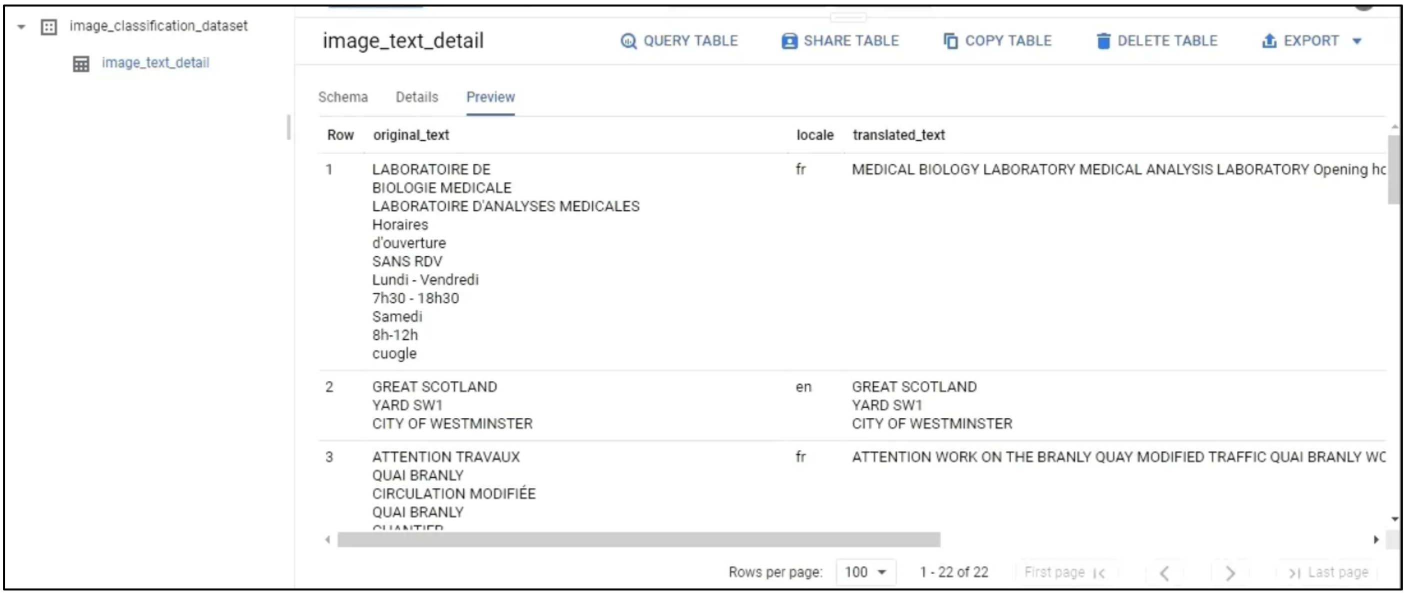


```
To suppress this warning, create an empty ~/.cloudshell/no-python-warning file.
The command will automatically proceed in  seconds or on any key.
********************************************************************************
Processing image files from GCS. This will take a few minutes..
SUTRO
TOWER
GOTO
37.76558947
-122.43935213
1.5km SW

No vehicles
except for
access to
off-street
premises
PO2

ANY PERSON WHO OMITS TO SHUT AND FASTEN THIS GATE
S LIABLE TO A PENALTY NOT EXCEEDING FORTY SHILLINGS

DOWNING
STREET SW1
CITY OF WESTMINSTER

GREAT SCOTLAND
YARD SW1
CITY OF WESTMINSTER

SAUCHIEHALL
STREET
$ CITY CENTRE
FRANCO
NEON
C

TRAVEL BROADENS THE MIND
LITTLE BRITAIN STREET 7
THE BRIDGE
DANGER
KEEP AWAY

NOR CAL
SIDEWALK CLOSED
AHEAD
BB NORCALS
CROSS HERE
DETOUR
LEVELIO

IMPORTS OF SEASONAL GOODS WHOLESALE - RETAIL TAVOULARIS
China Europe Trading Co., Ltd.
DO NOT FIRE
CARABINIERI archaeological area POMPEII EXCAVATIONS POMPEN RUINS lower area Entrance - Entrance
Dior Dior With the app in an instant you&#39;re gone
MEDICAL BIOLOGY LABORATORY MEDICAL ANALYSIS LABORATORY Opening hours WITHOUT APPOINTMENT Monday - Friday 7:30 am - 6:30 pm Saturd
ay 8 am-12pm cuogle
ATTENTION WORK ON THE BRANLY QUAY MODIFIED TRAFFIC QUAI BRANLY WORKSITE AVENUE SUFFREN 30 FROM 04/10
pedestrian area except except 0
Sakuma Denki Shokai Co., Ltd.
Let&#39;s drive so that the car does not go on the day Pedestrian priority
Oshiage &quot;Oshiage Tokyo Skytree Shirahige Bridge &lt;453) Shirahige Brdg. (46) Sumitsutsumi Dori, 463) Asakusa Asakusa 200m +
 November
It will be a nuisance to those who live in the neighborhood. Banda Ward Road Park
Writing Vision API image data to BigQuery...
student_04_d88c2593f061@cloudshell:~ (qwiklabs-gcp-04-ce1a352f364c)$
```

Confirm that image data has been successfully uploaded to BigQuery

Go back to the Cloud Console, navigate to **BigQuery**.

Preview the table `image_text_detail` in the dataset called `image_classification_dataset` in your project.



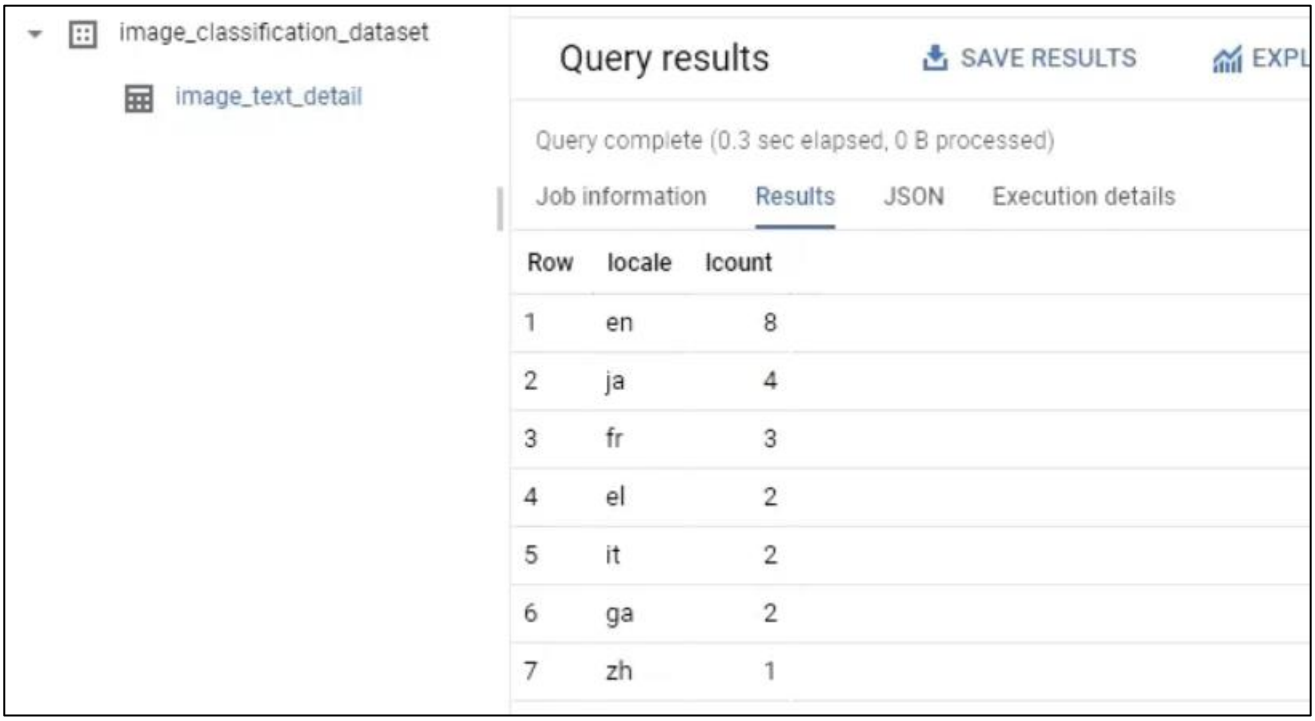
Row	original_text	locale	translated_text
1	LABORATOIRE DE BIOLOGIE MEDICALE LABORATOIRE D'ANALYSES MEDICALES Horaires d'ouverture SANS RDV Lundi - Vendredi 7h30 - 18h30 Samedi 8h-12h cuogle	fr	MEDICAL BIOLOGY LABORATORY MEDICAL ANALYSIS LABORATORY Opening hc
2	GREAT SCOTLAND YARD SW1 CITY OF WESTMINSTER	en	GREAT SCOTLAND YARD SW1 CITY OF WESTMINSTER
3	ATTENTION TRAVAUX QUAI BRANLY CIRCULATION MODIFIÉE QUAI BRANLY CHANTIER	fr	ATTENTION WORK ON THE BRANLY QUAY MODIFIED TRAFFIC QUAI BRANLY WC

Rows per page: 100 1 - 22 of 22 First page Last page

Confirm that image data has been successfully processed by running the following Query in BigQuery:

```
SELECT locale,COUNT(locale) as lcount FROM image_classification_dataset.image_text_detail GROUP
BY locale ORDER BY lcount DESC
```

The query results should look like:



Query complete (0.3 sec elapsed, 0 B processed)

Row	locale	lcount
1	en	8
2	ja	4
3	fr	3
4	el	2
5	it	2
6	ga	2
7	zh	1

Congratulations! You completed this challenge lab.