Part 5

Let’s use our new model. This section will show you how to use GCS and Cloud Functions to get annotations about images.

1. Open Google Cloud Console: https://console.cloud.google.com
2. Open Cloud Shell.
3. Create bucket using:

```
gsutil mb -l us-central1 gs://$DEVSHELL_PROJECT_ID-vision/
```

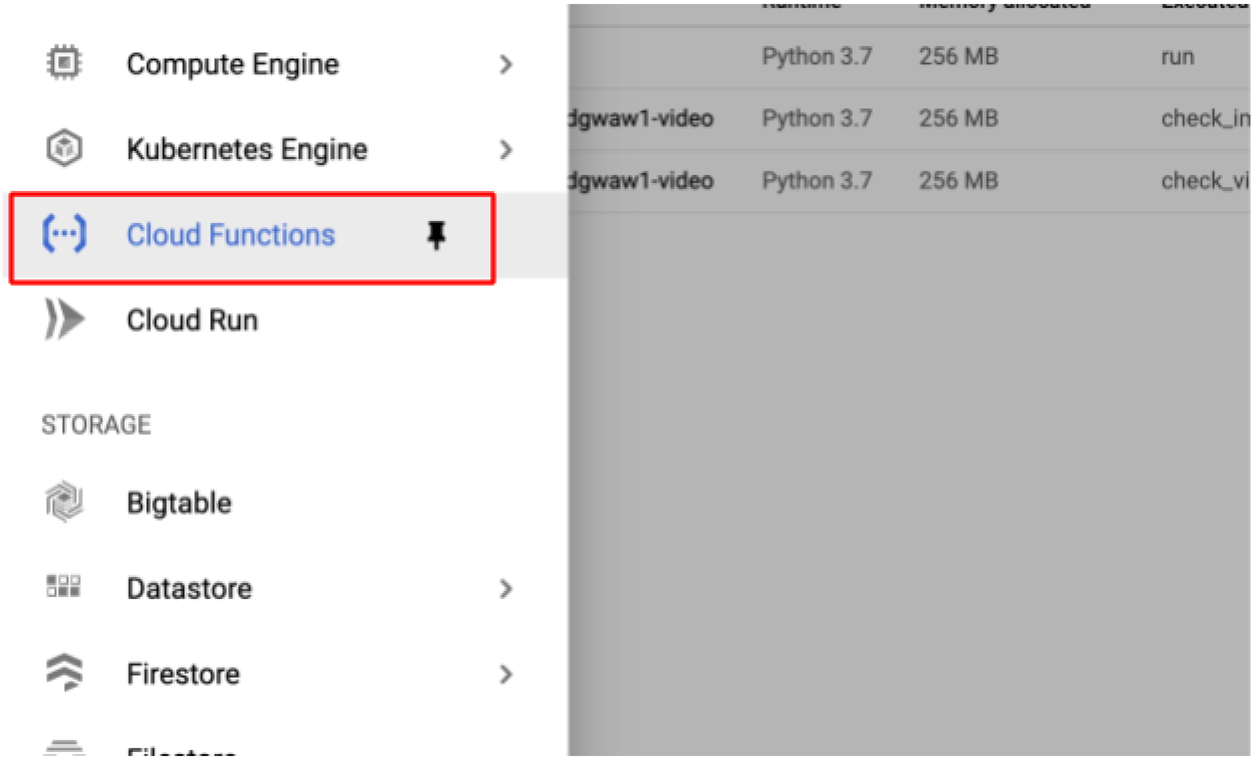
4. Go to folder vision:

```
cd ~/gdgwarsaw2019/vision
```

5. Run visionapi.py:

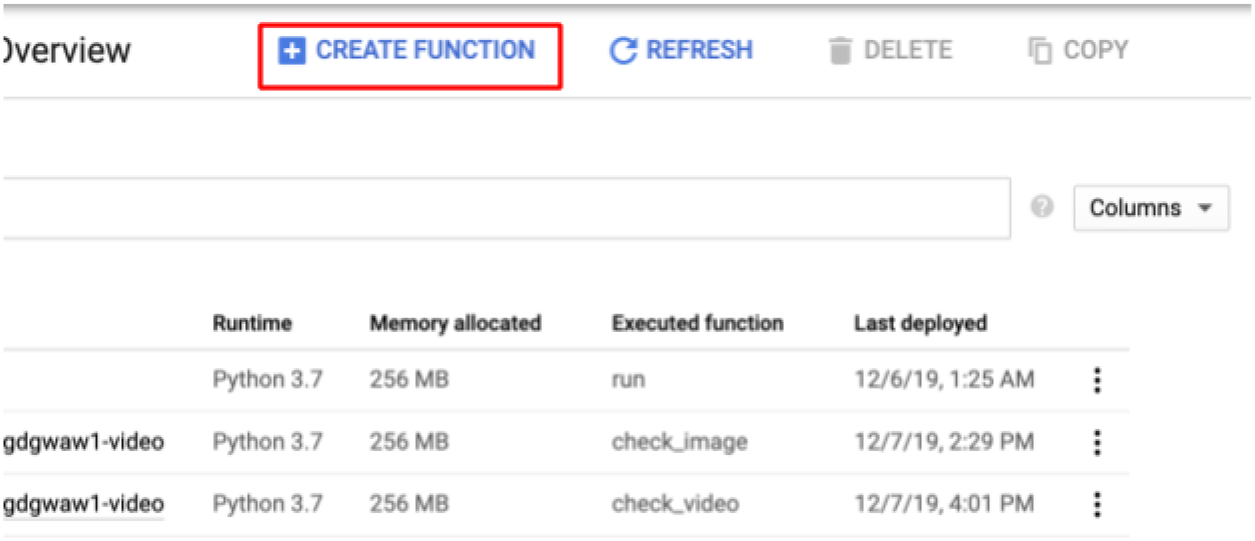
```
python visionapi.py
```

6. Go to Cloud Functions:



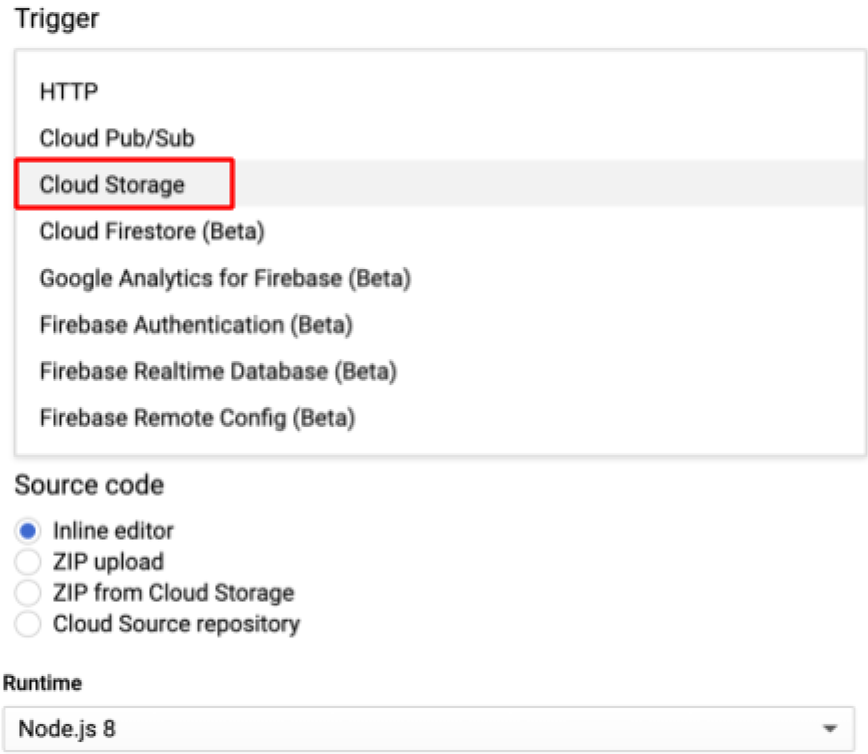
	Python 3.7	256 MB	run
dgwaw1-video	Python 3.7	256 MB	check_in
dgwaw1-video	Python 3.7	256 MB	check_vi

7. Click CREATE FUNCTION:



	Runtime	Memory allocated	Executed function	Last deployed
	Python 3.7	256 MB	run	12/6/19, 1:25 AM
gdgwaw1-video	Python 3.7	256 MB	check_image	12/7/19, 2:29 PM
gdgwaw1-video	Python 3.7	256 MB	check_video	12/7/19, 4:01 PM

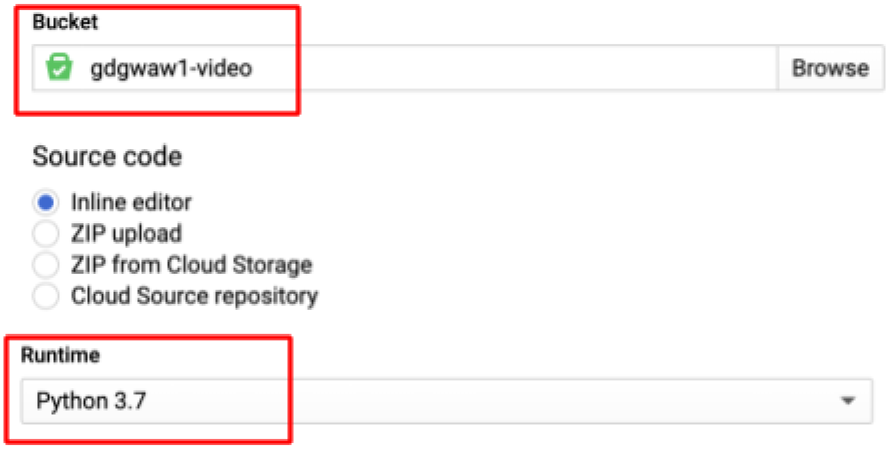
8. Choose Trigger - **Cloud Storage**:


Trigger
HTTP
Cloud Pub/Sub
Cloud Storage
Cloud Firestore (Beta)
Google Analytics for Firebase (Beta)
Firebase Authentication (Beta)
Firebase Realtime Database (Beta)
Firebase Remote Config (Beta)
Source code
ZIP upload
ZIP from Cloud Storage
Cloud Source repository

Source code: ● Inline editor

Runtime: Node.js 8

9. Choose bucket - **PROJECT_ID-video** and runtime **Python 3.7**:

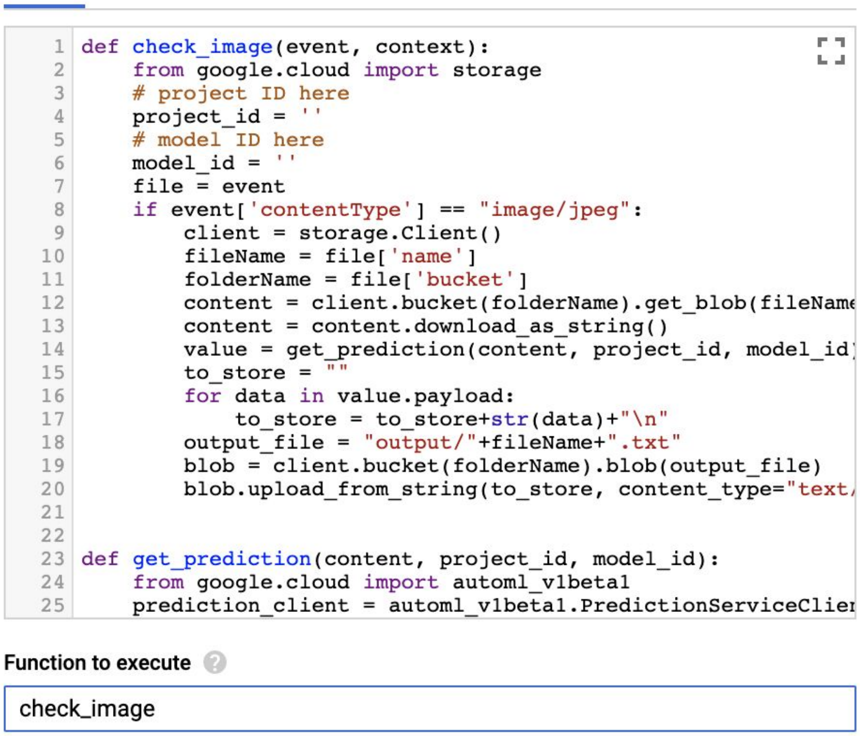


Bucket: gdgwaw1-video

Source code: ● Inline editor

Runtime: Python 3.7

10. Open ~/gdgwarsaw2019/vision/automl.py copy all and paste to main.py in Cloud Functions. Change field *Function to execute to* **check_image**:



```
def check_image(event, context):
    from google.cloud import storage
    # project ID here
    project_id = ''
    # model ID here
    model_id = ''
    file = event
    if event['contentType'] == "image/jpeg":
        client = storage.Client()
        fileName = file['name']
        folderName = file['bucket']
        content = client.bucket(folderName).get_blob(fileName
        content = content.download_as_string()
        value = get_prediction(content, project_id, model_id)
        to_store = ""
        for data in value.payload:
            to_store = to_store+str(data)+"\n"
        output_file = "output/"+fileName+".txt"
        blob = client.bucket(folderName).blob(output_file)
        blob.upload_from_string(to_store, content_type="text/


def get_prediction(content, project_id, model_id):
    from google.cloud import automl_v1beta1
    prediction_client = automl_v1beta1.PredictionServiceClien
```

Function to execute

check_image

11. Fill **project_id** and **model_id** variables
12. Add to requirements.txt:

google-cloud-storage >= 1.14.0
google-cloud-automl >= 0.9.0

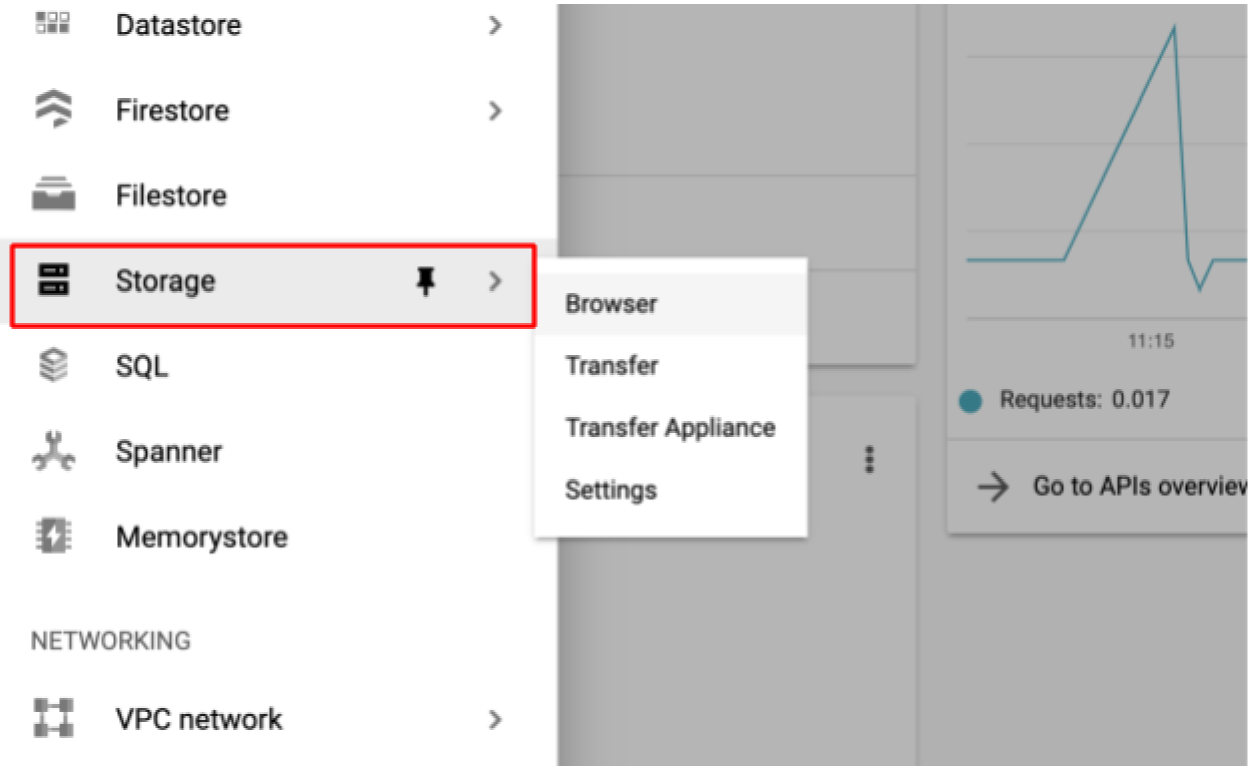
13. Click **Create**
14. Wait for Deployment.
15. Open Cloud shell and go to ~/gdgwarsaw2019/vision:

```
cd ~/gdgwarsaw2019/vision
```

16. Copy jpg files to storage bucket:

```
gsutil cp *.jpg gs://$DEVSHELL_PROJECT_ID-vision/
```

17. Go to Storage:



18. Open PROJECT_ID-vision bucket
19. Open folder output and check output files.