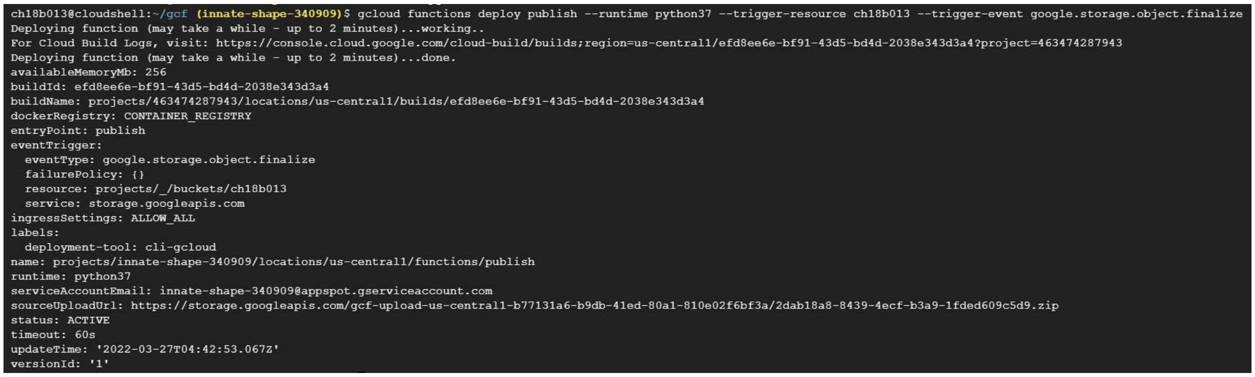
CS4830 - BIG DATA LABORATORY
LAB - 6
VISHAL RISHI MK - CH18B013

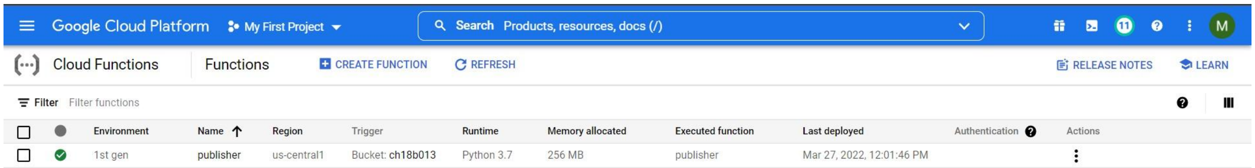
Q1. In this assignment, you will count the number of lines in a file uploaded to the GCS bucket in real-time by using Google Cloud Functions and Pub/Sub.

a) The file *‘sample1.txt’* is downloaded and saved in the terminal.
b) The Google Cloud Function ‘*publisher’* gets triggered whenever a file is added to a bucket and publishes the file name to a topic in Pub/Sub. It is given in the ‘*main.py’* file.
c) ***‘****subscriber.py’* acts as a subscriber to this topic and prints out the number of lines in the file in real-time.



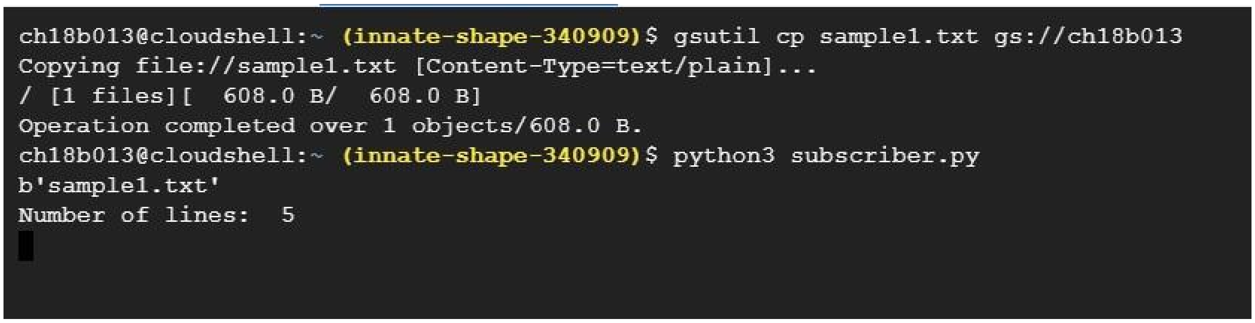
```
ch18b013@cloudshell:~/gcf (innate-shape-340909)$ gcloud functions deploy publish --runtime python37 --trigger-resource ch18b013 --trigger-event google.storage.object.finalize
Deploying function (may take a while - up to 2 minutes)...working..
For Cloud Build Logs, visit: https://console.cloud.google.com/cloud-build/builds;region=us-central1/efd8ee6e-bf91-43d5-bd4d-2038e343d3a4?project=463474287943
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: efd8ee6e-bf91-43d5-bd4d-2038e343d3a4
buildName: projects/463474287943/locations/us-central1/builds/efd8ee6e-bf91-43d5-bd4d-2038e343d3a4
dockerRegistry: CONTAINER_REGISTRY
entryPoint: publish
eventTrigger:
  eventType: google.storage.object.finalize
  failurePolicy: {}
  resource: projects/_/buckets/ch18b013
  service: storage.googleapis.com
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
name: projects/innate-shape-340909/locations/us-central1/functions/publish
runtime: python37
serviceAccountEmail: innate-shape-340909@appspot.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/gcf-upload-us-central1-b77131a6-b9db-41ed-80a1-810e02f6bf3a/2dab18a8-8439-4ecf-b3a9-1fded609c5d9.zip
status: ACTIVE
timeout: 60s
updateTime: '2022-03-27T04:42:53.067Z'
versionId: '1'
```

Figure 1: Deploying the GCF



□	●	Environment	Name ↑	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
□	✓	1st gen	publisher	us-central1	Bucket: ch18b013	Python 3.7	256 MB	publisher	Mar 27, 2022, 12:01:46 PM		⋮

Figure 2: Deploying the GCF



```
ch18b013@cloudshell:~ (innate-shape-340909)$ gsutil cp sample1.txt gs://ch18b013
Copying file://sample1.txt [Content-Type=text/plain]...
/ [1 files][  608.0 B/  608.0 B]
Operation completed over 1 objects/608.0 B.
ch18b013@cloudshell:~ (innate-shape-340909)$ python3 subscriber.py
b'sample1.txt'
Number of lines:  5
```

Figure 3: Output of the subscriber. We count the number of lines in real-time in the sample1.txt file using this code. Hence, we have created the subscriber to the topic and we have calculated the number of lines in the sample.txt document.

Q2. There are two kinds of subscribers - pull and push subscribers. What are the differences between the two and when would you prefer one over the other?

Subscriptions are named resources representing the stream of messages from a single, specific topic, to be delivered to the subscribing application. A publisher application creates and sends messages to a topic. Subscriber applications create a subscription to a topic to receive messages from it. There are two types of subscriptions that one can choose from: pull subscriptions and push subscriptions.

Push Subscription:
With a push subscription, the Publisher propagates changes to a Subscriber without a request from the Subscriber. Changes can be pushed to Subscribers on demand, continuously, or on a scheduled basis. Push Subscription is used when:

- Data needs to be synchronized continuously or on a frequently recurring schedule.
- Publications require near real-time movement of data.
- Most often used with snapshot and transactional replication.

Pull Subscription:
With a pull subscription, the Subscriber requests changes made at the Publisher. Pull subscriptions allow the user at the Subscriber to determine when the data changes are synchronized. Pull Subscription is used when:

- Data needs to be synchronized on-demand or on a schedule rather than continuously.
- The publication has a large number of Subscribers, and/or it would be too resource-intensive to run all the agents at the Distributor.
- Subscribers are autonomous, disconnected, and/or mobile. Subscribers will determine when they will connect and synchronize changes.
- Most often used with merge replication.