**Name**: Urvee Chaudhari

**GCP Assessment- Stalkdriver, Cloud Function, Pub/Sub**

**Ques1:** Steps to export all the logs related to firewall rules to BigQuery for further analysis. Use console. (Only export to BigQuery, analysis not required). (1m)

(Steps are for Stalkdriver Classic)

1. In Stackdriver Logging, at the top of Log Exports page, select on Create export

2. Click on Edit Export

3. Enter the Sink name, service and destination

In Sink Service, we select a destination service, here we click on BigQuery

In Sink Destination, select or create the particular dataset to receive the exported logs.

4. Click on Update Sink to create a sink

To view the logs in BigQuery-

1. GO to BigQuery UI

2. Select the dataset used as the sink's destination

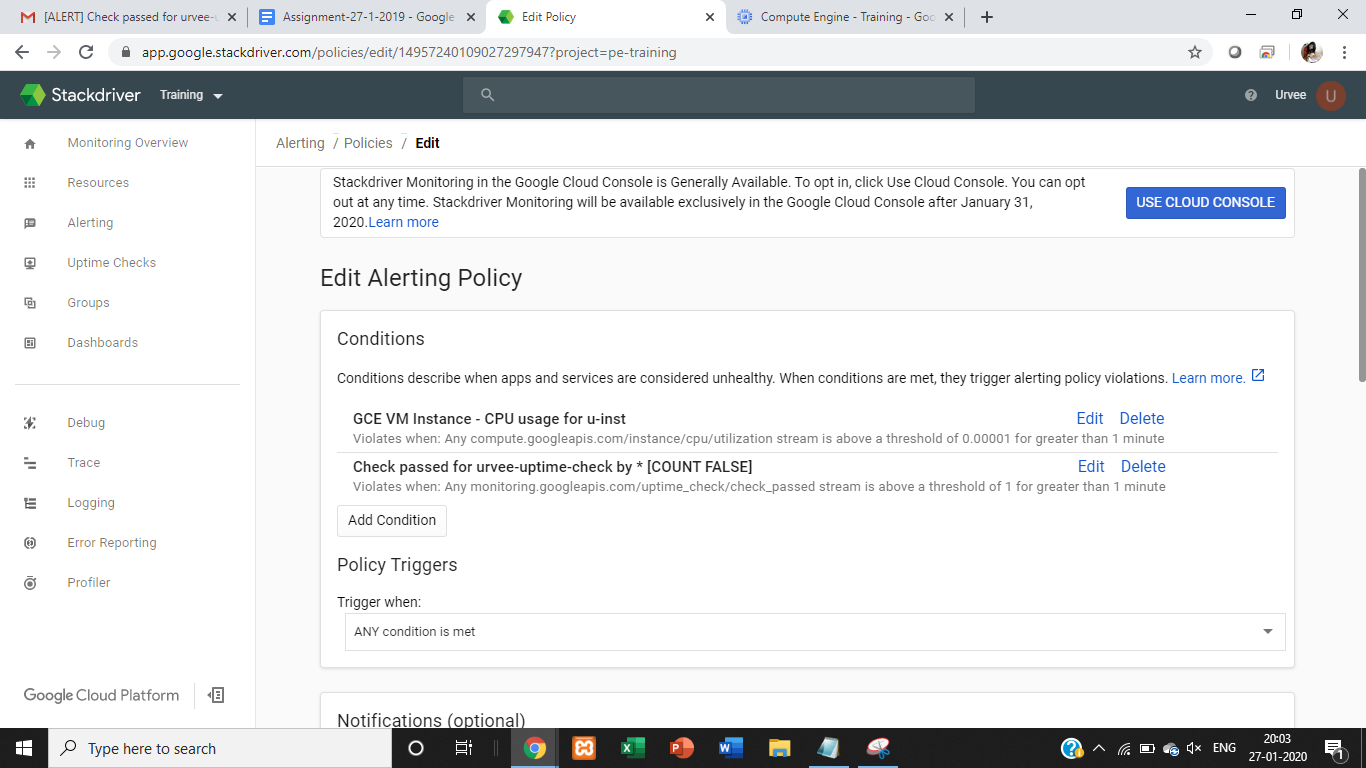
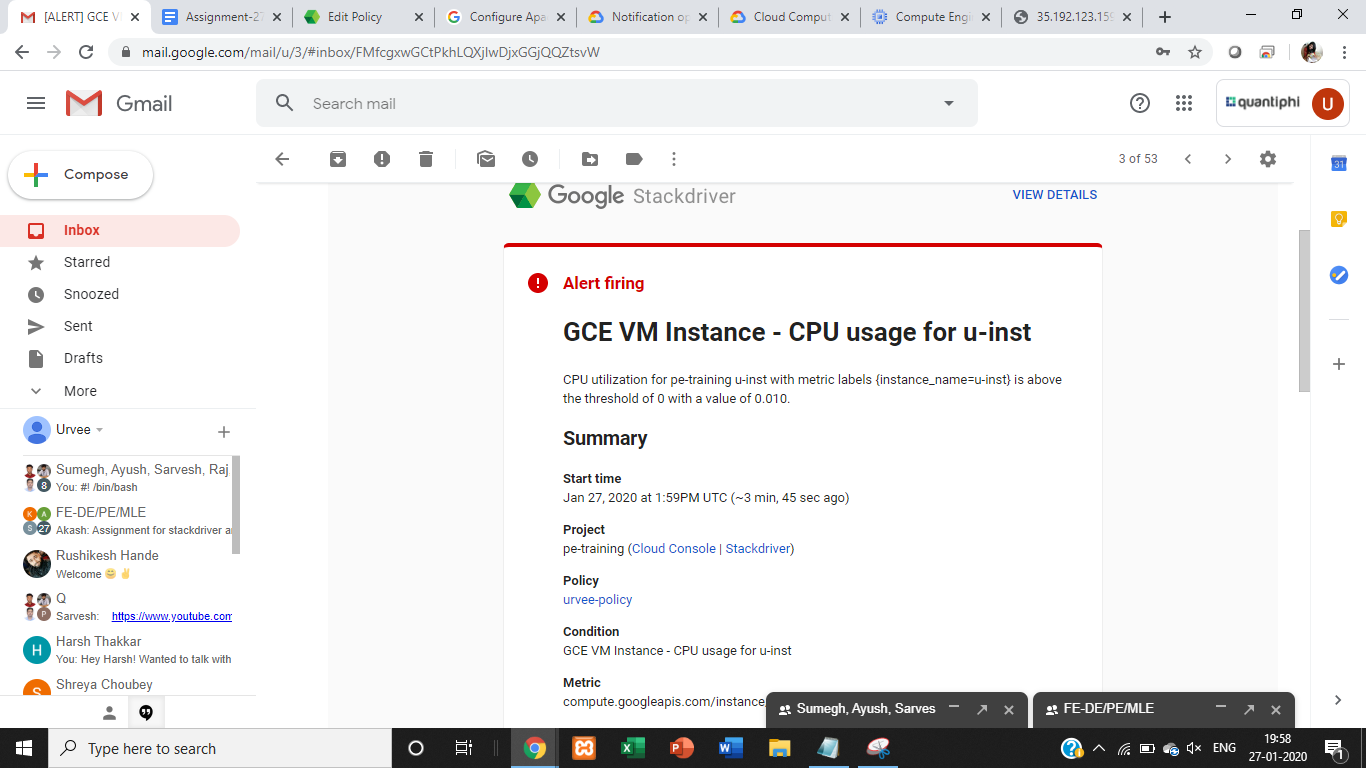
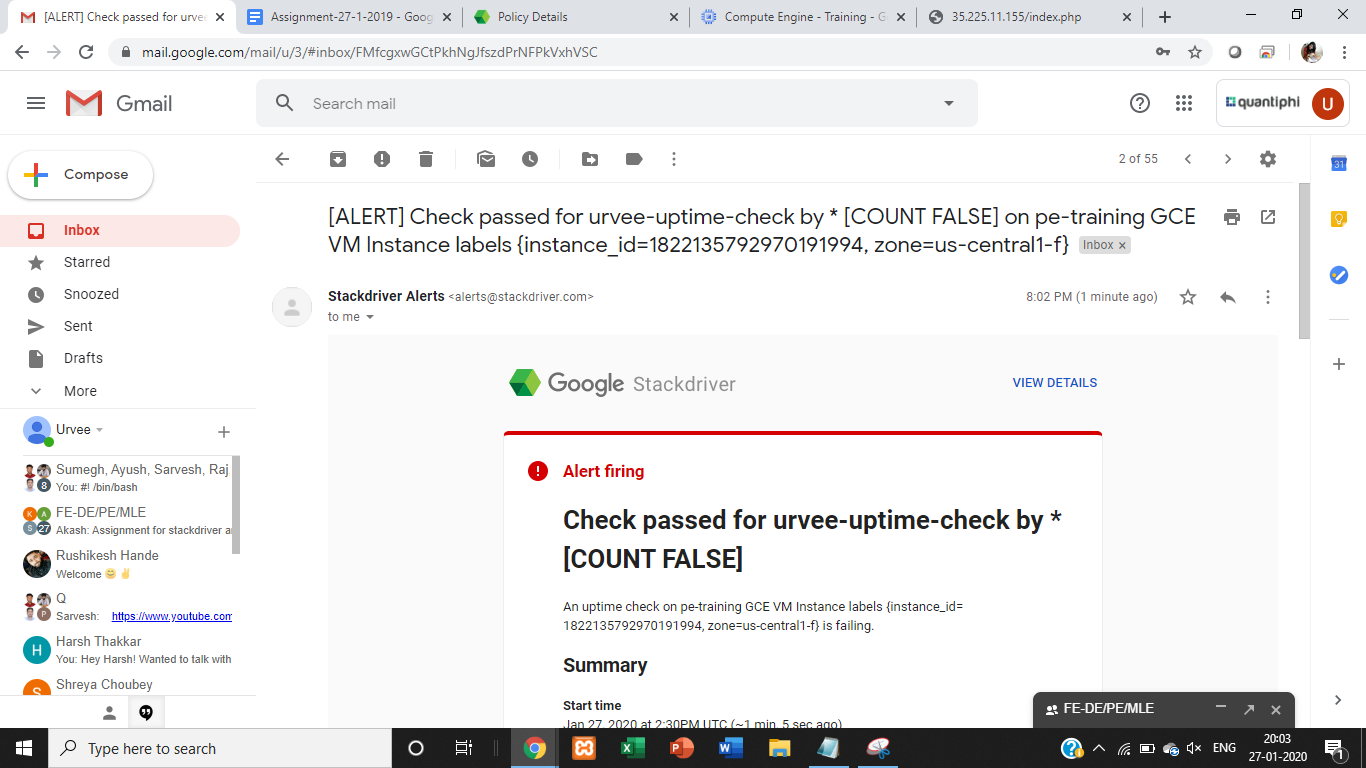
3.Select one of the dataset's tables.

The log entries are visible on the Details tab.

**Ques2**: Configure Apache2 HTTP server on a GCE VM instance and setup an email alert notification which triggers when the health check of the instance fails. Use console.

**Screenshots**:

1. Output mail of first policy condition
2. Output mail of second policy condition
3. The 2 policies created



**Ques3**: Create a Cloud Function to convert the pub/sub message to json file and store it in GCS bucket

Eg:

If message published is:  
{

"name":"test-file",

"content":'{"source": "pub/sub", "destination": "gcs"}'

}

Then there should be a file `test-file.json` in the destination bucket with the content value.

**Screenshots:**

1.Logs

2. Bucket (u-buck)

**Code**:

  import base64  
from gcloud import storage  
import os  
import json  
   
def hello\_pubsub(event, context):  
    pubsub\_message = base64.b64decode(event['data']).decode('utf-8')  
    print(type(pubsub\_message))  
    print(pubsub\_message)  
     
     
    list1 = pubsub\_message.split(" ")  
    print(list1)  
     
    text = {  
    "name":list1[0],  
    "content":  
{  
"source":list1[1],  
"destination":list1[2]  
}  
    }  
   
    print(text)  
    filename = "/tmp/"+str(list1[0])+".json"  
     
    with open(filename, "w") as write\_file:  
                json.dump(text, write\_file)  
   
    client = storage.Client(project='Training')  
    bucket = client.get\_bucket('u-buck')  
    blob = bucket.blob('test\_file.json')  
    blob.upload\_from\_filename(filename)  
    print("Done")

