**Cloud Functions and Pub/Sub - Assessment**

**Name: Sarvesh Virkud**

**EID: 20791**

**1. 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)**

**Ans:**

1. Using 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.

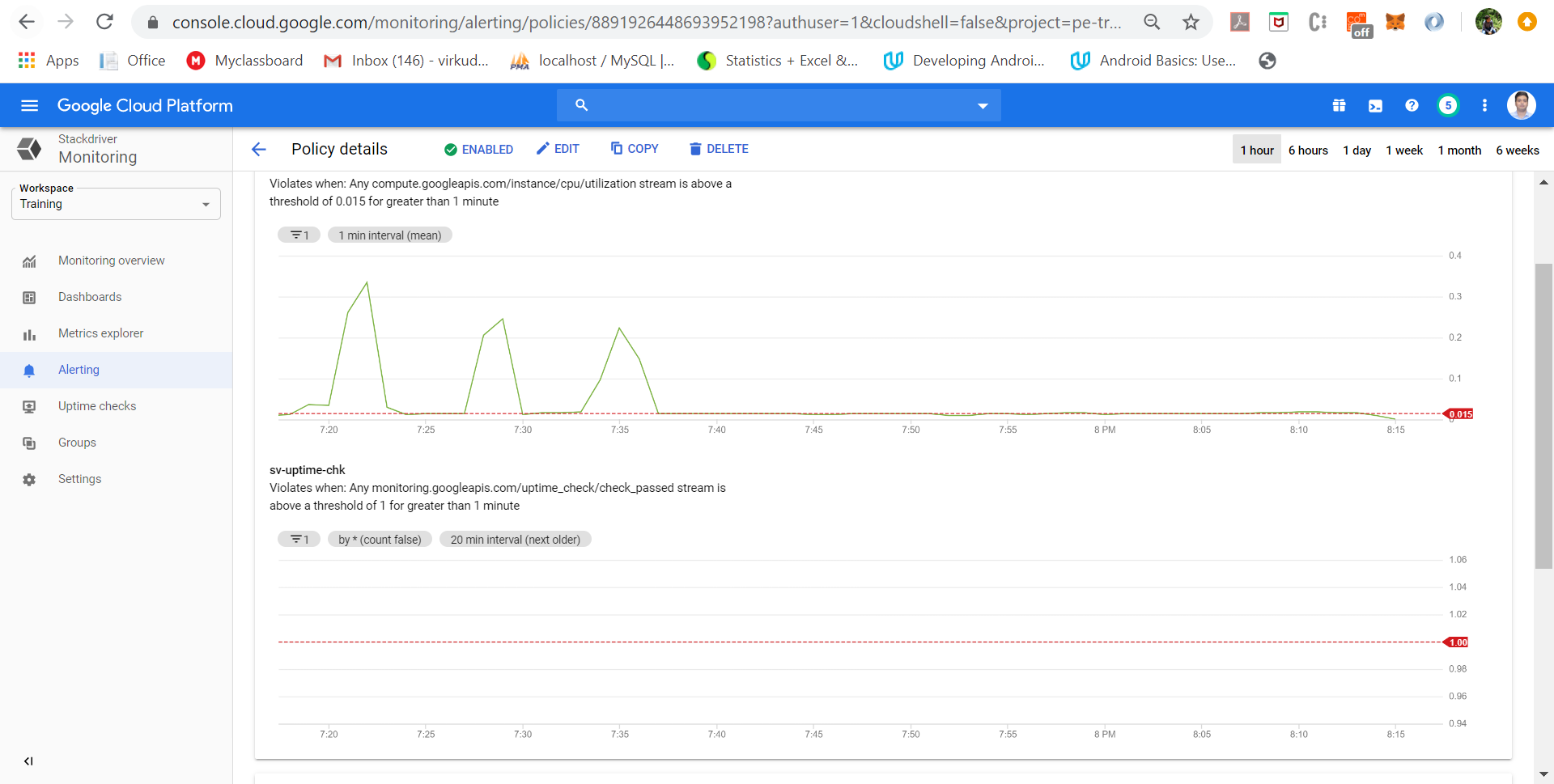
2. Select the dataset used as the sink's destination.

3. Select one of the dataset's tables.

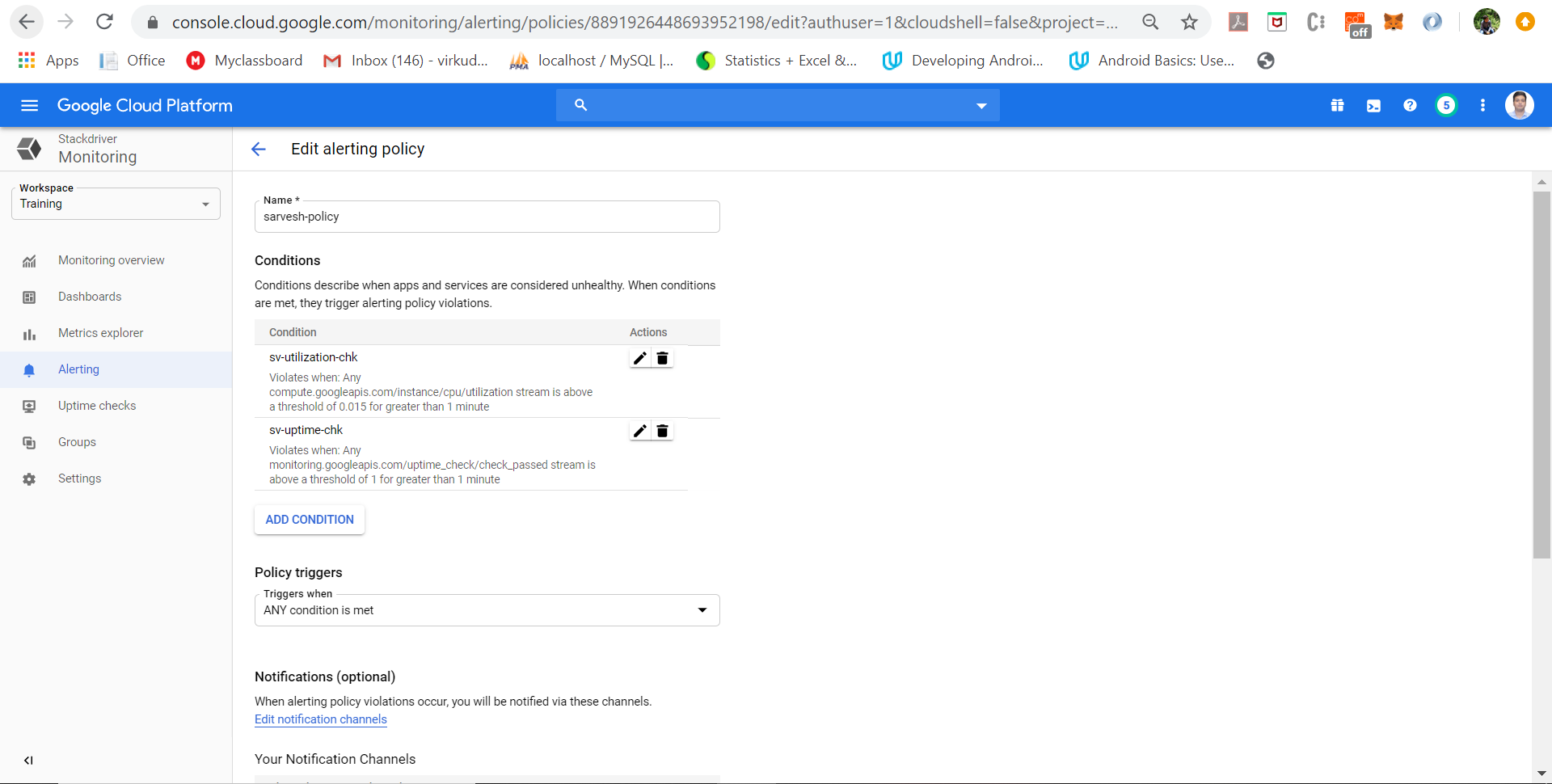
The log entries are visible on the Details tab.

**2. 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. (1.5m)**

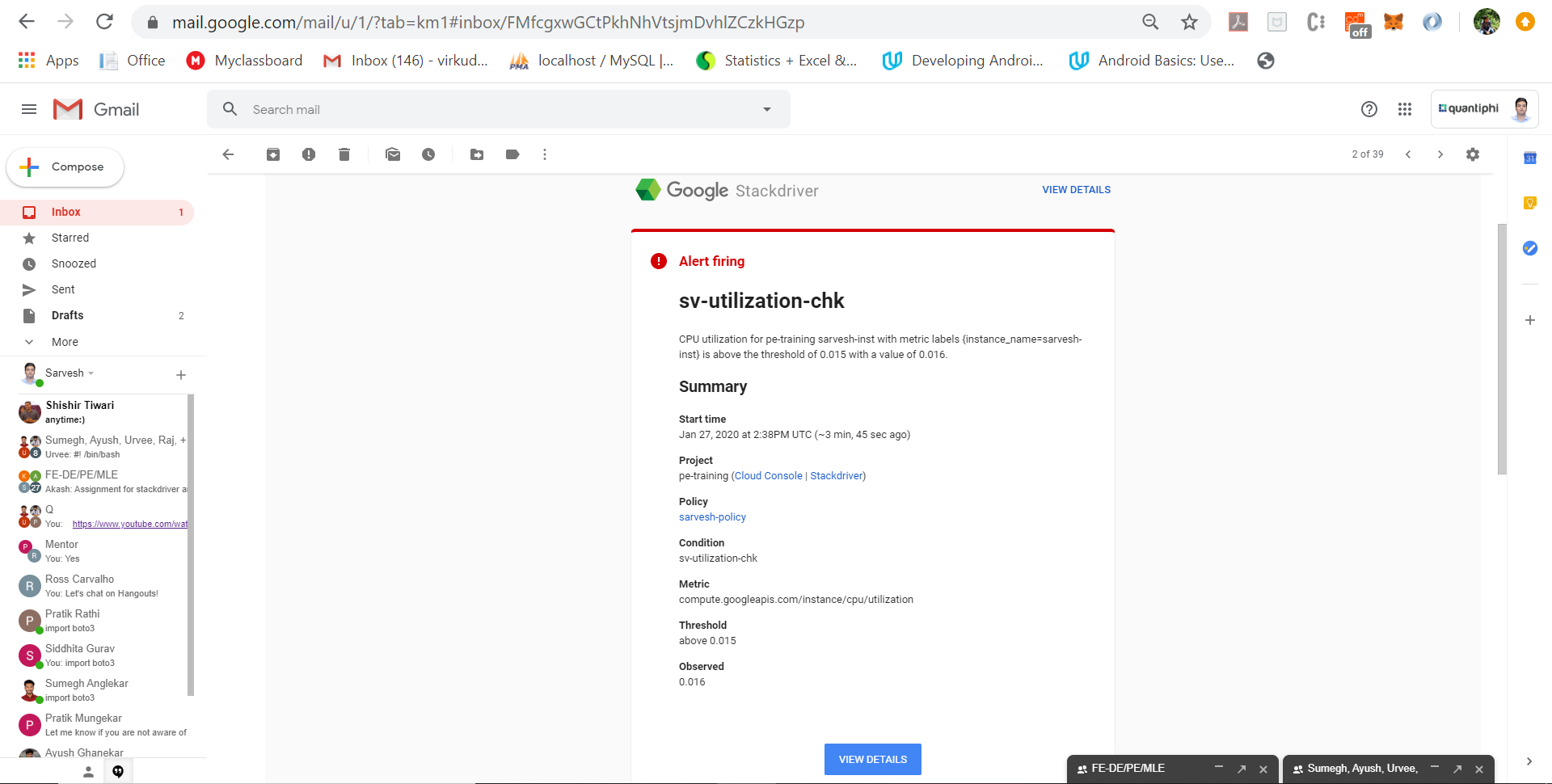
**Policy:**



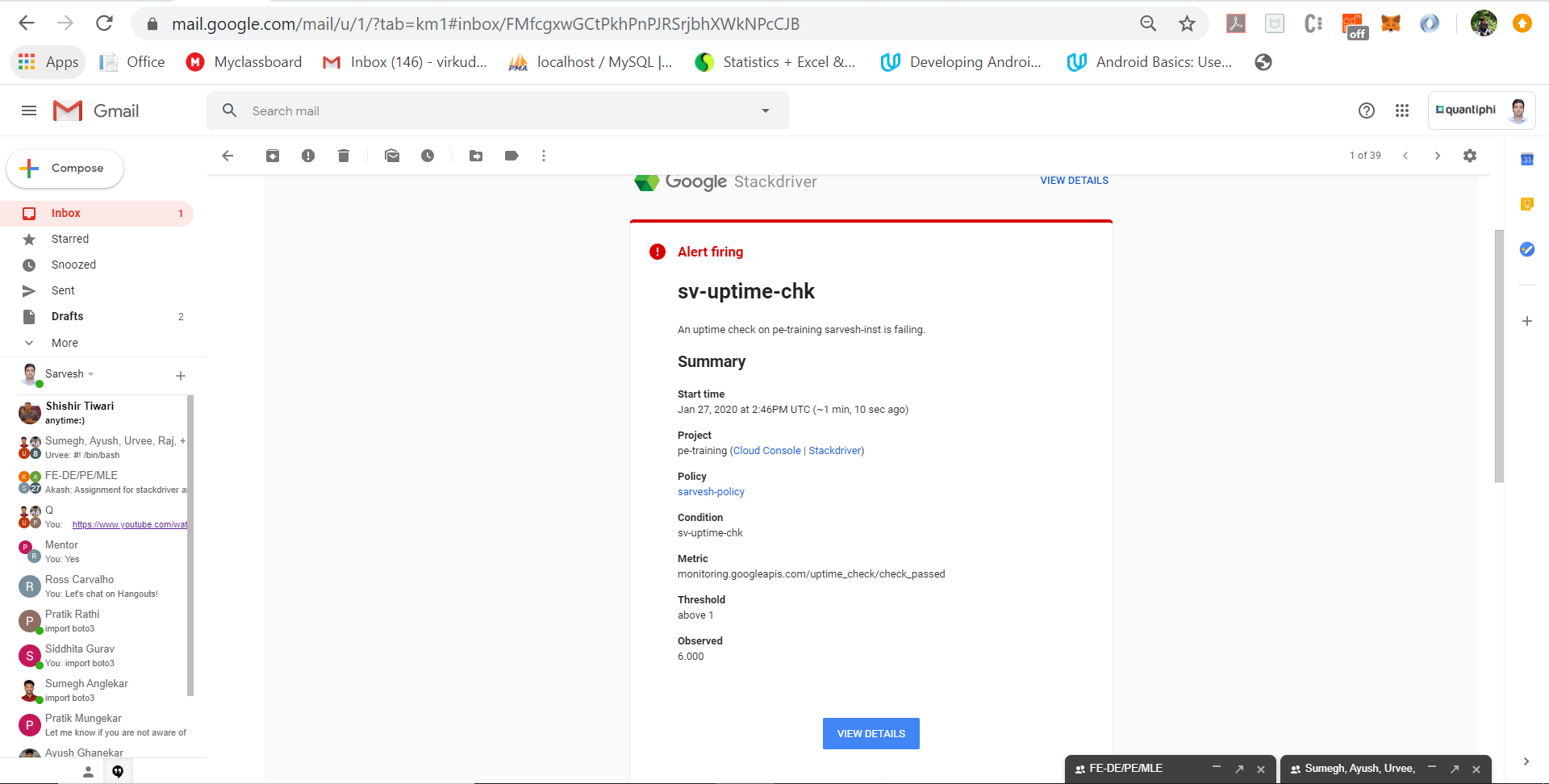
**2 policies created:**



**Alert regarding Utilization Check:**



**Alert regarding uptime-check:**



**3. Create a Cloud Function to convert the pub/sub message to json file and store it in GCS bucket  (2.5m)**

**Eg:**

**If message published is:  
{**

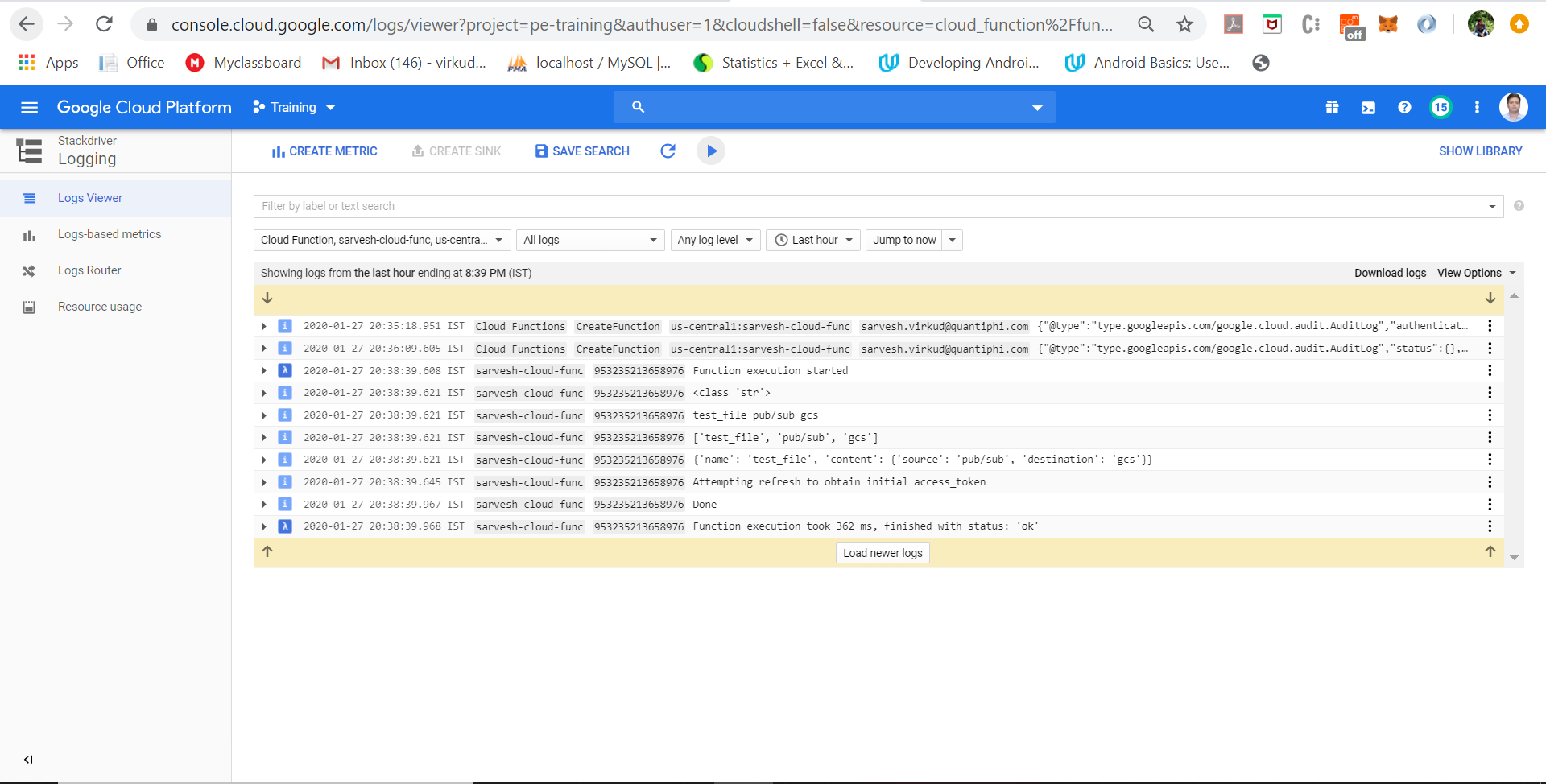
**"name":"test-file",**

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

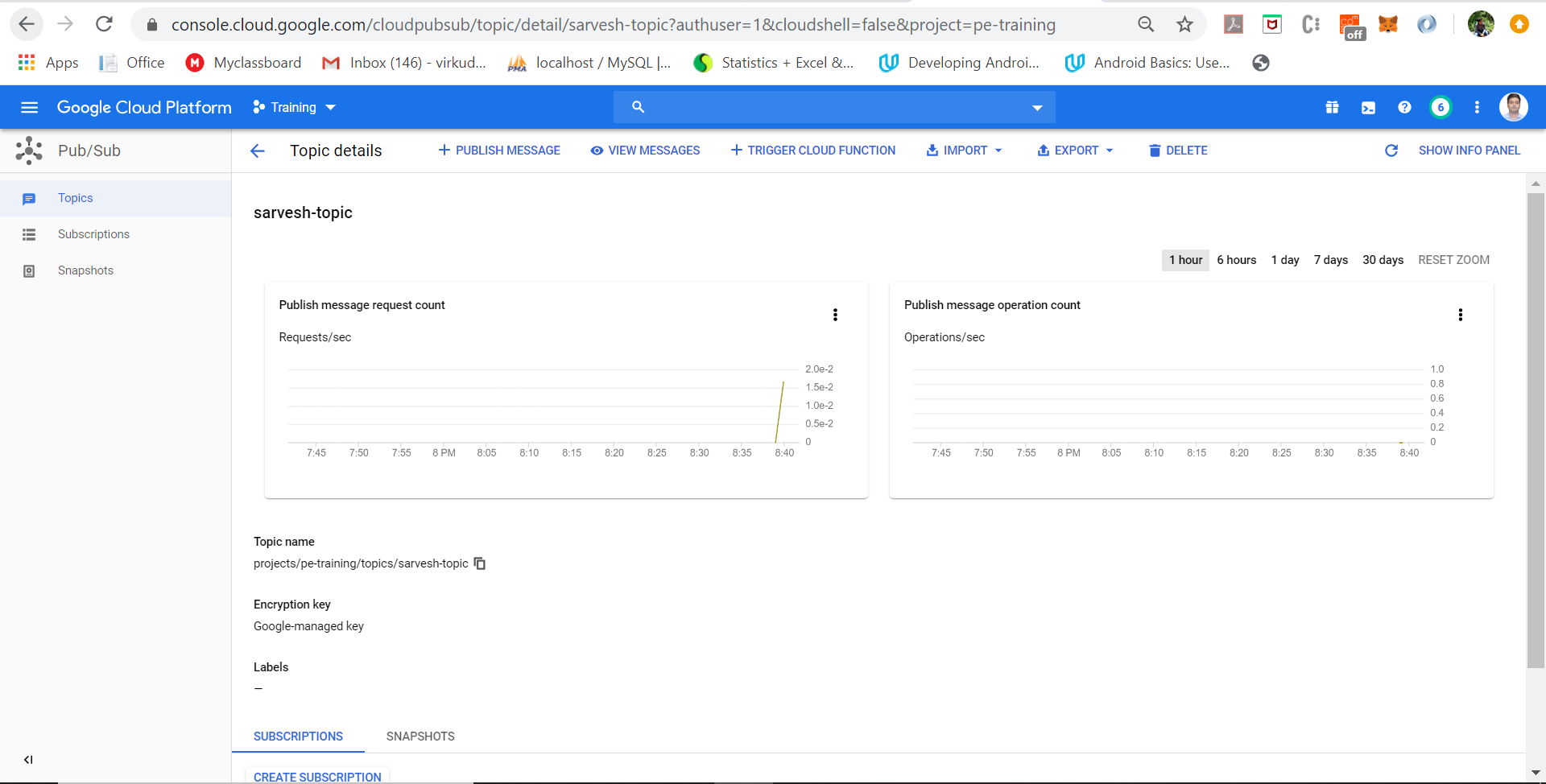
**}**

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

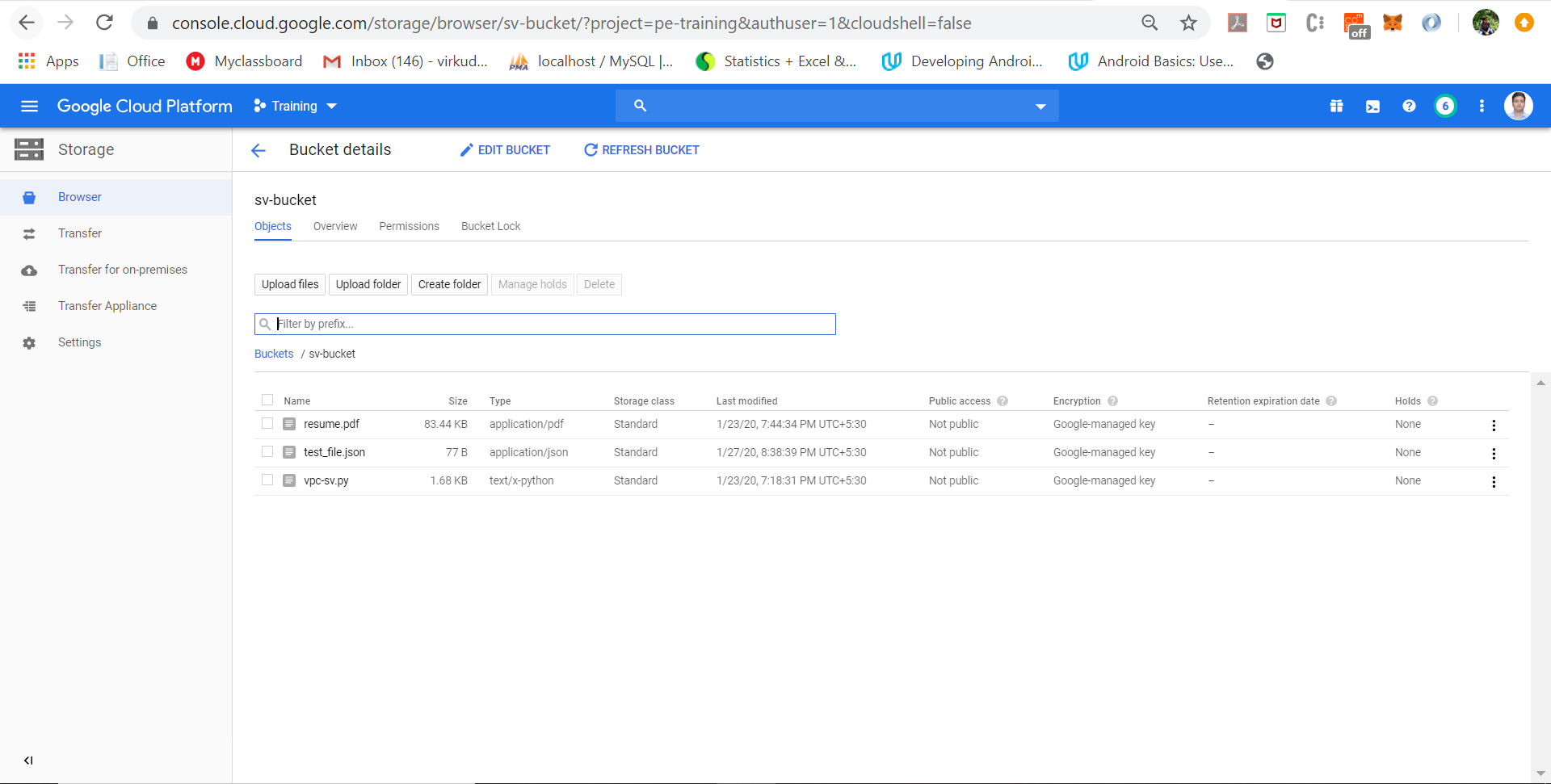
**Logs:**



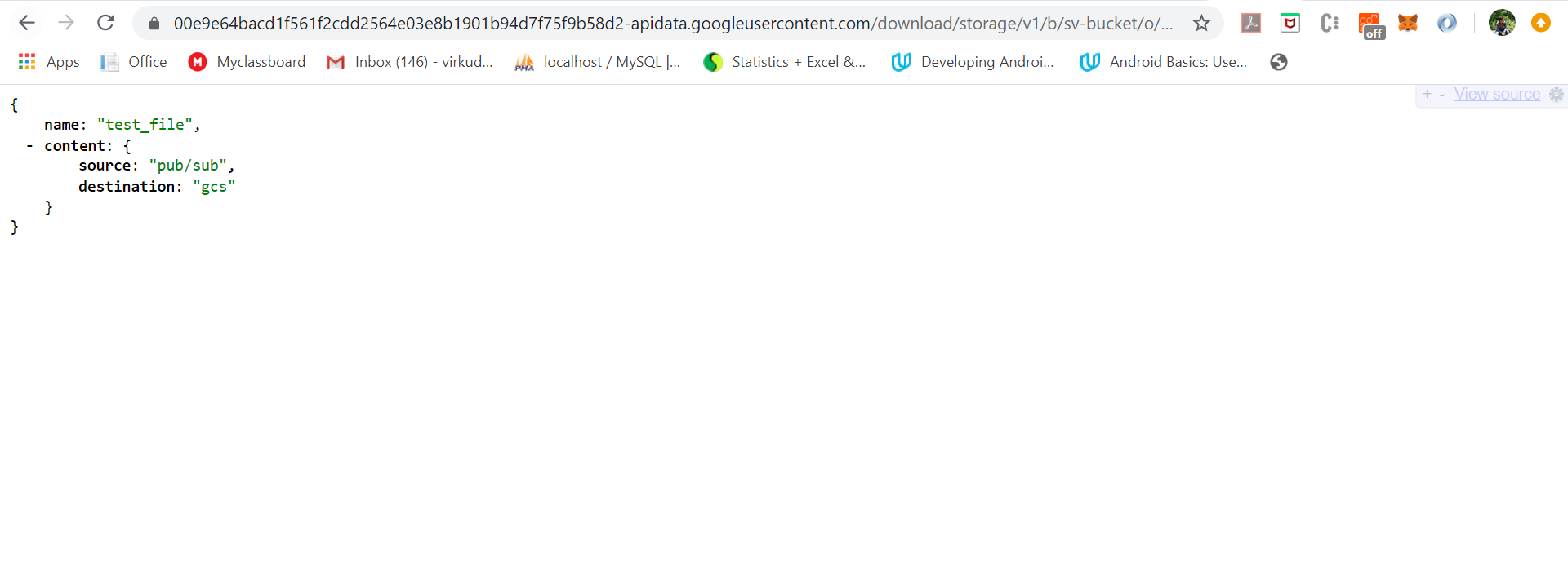
**Topic Created:**



**File created in the bucket:**



**JSON File:**

****

**Source 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('sv-bucket')

    blob = bucket.blob('test\_file.json')

    blob.upload\_from\_filename(filename)

    print("Done")

**Code for taking the entire JSON file as text and storing it in the bucket.**

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)

json\_data = eval(pubsub\_message)

print(json\_data)

print(type(json\_data))

filename = "/tmp/"+str(json\_data['name'])+".json"

print(filename)

with open(filename, "w") as write\_file:

json.dump(json\_data, write\_file)

client = storage.Client(project='Training')

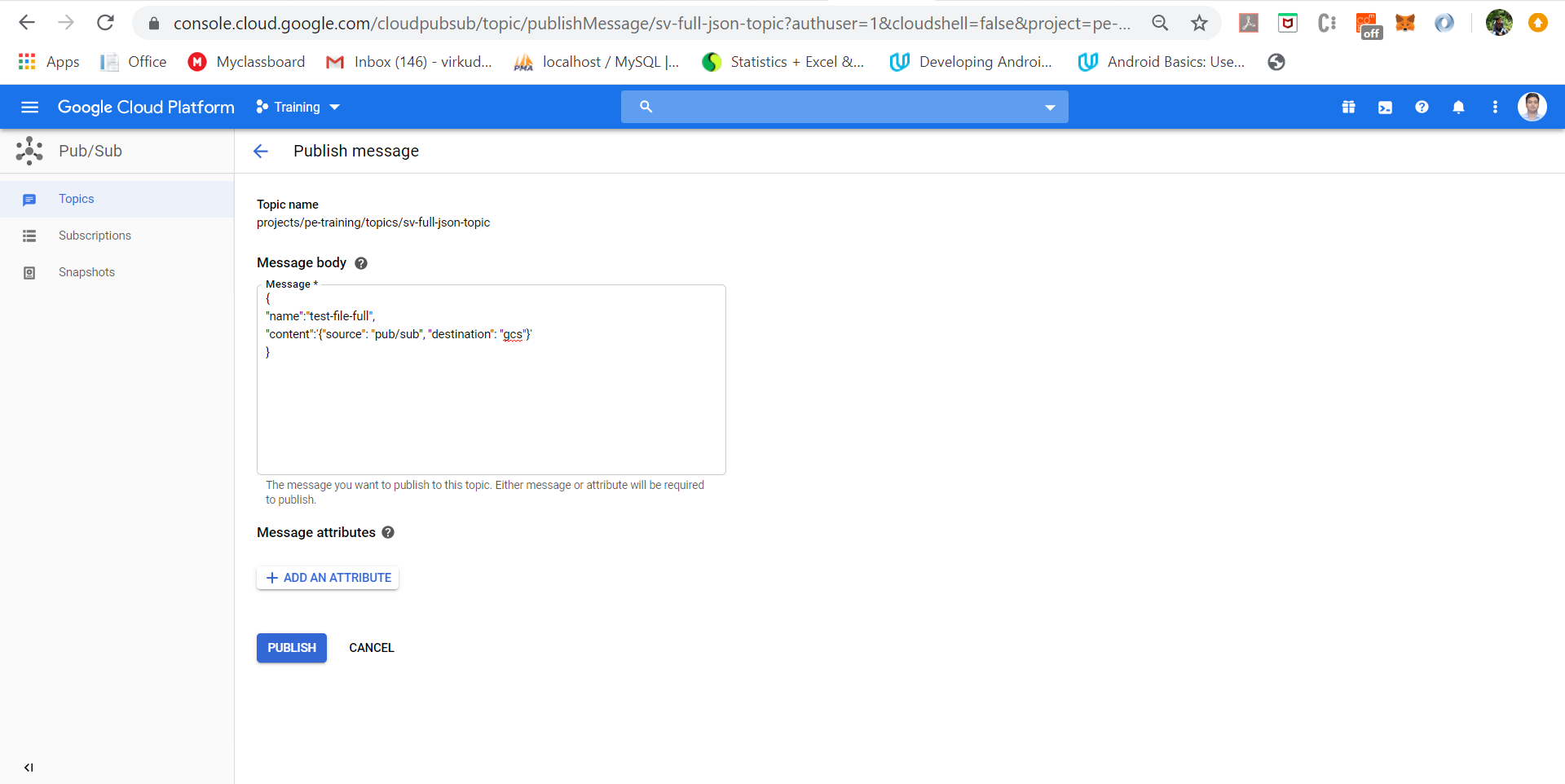
bucket = client.get\_bucket('sv-bucket')

blob = bucket.blob(filename)

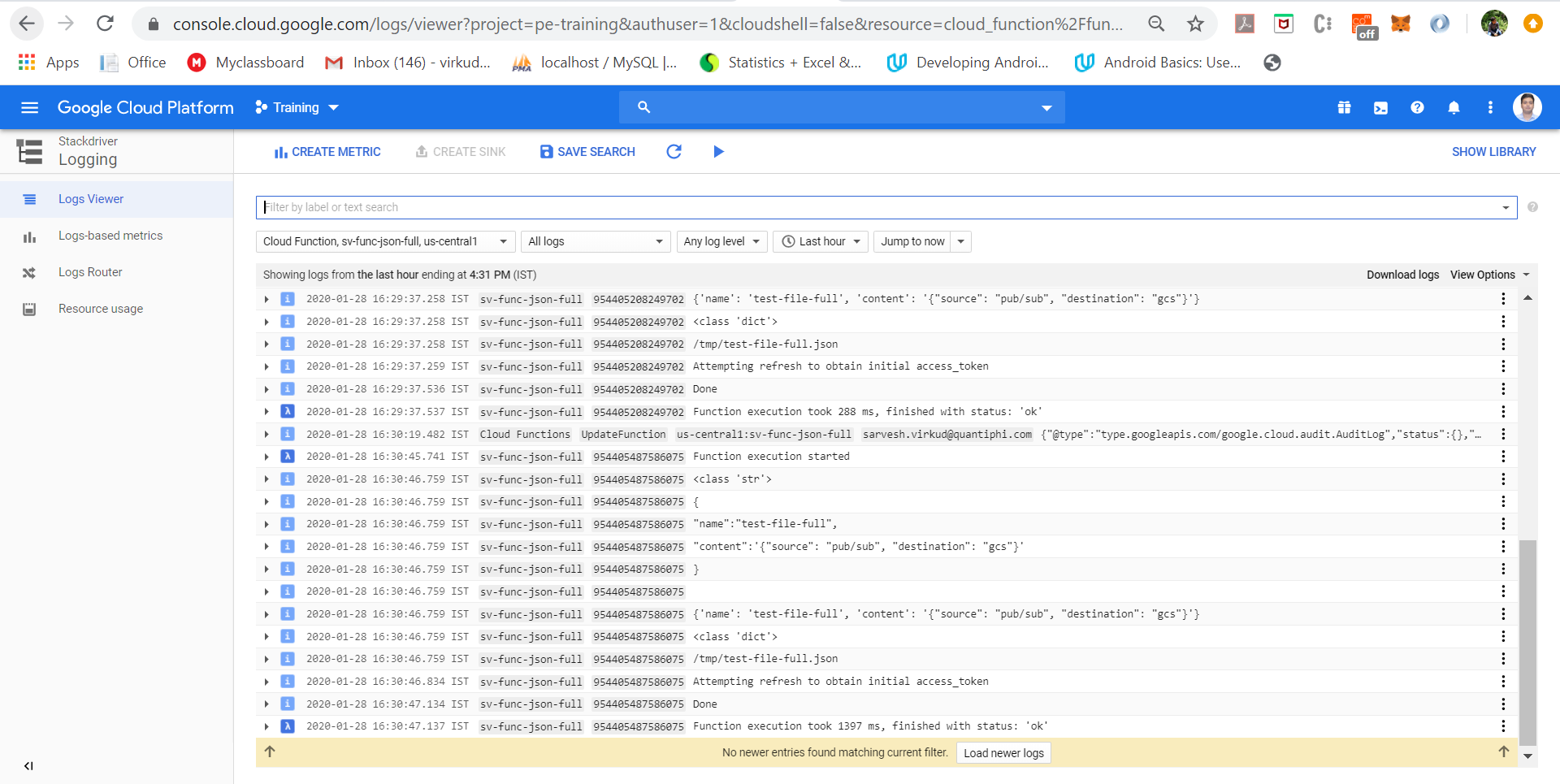
blob.upload\_from\_filename(filename)

print("Done")

**Publishing the message:**

****

**Logs:**

****