Cloud pub/sub:

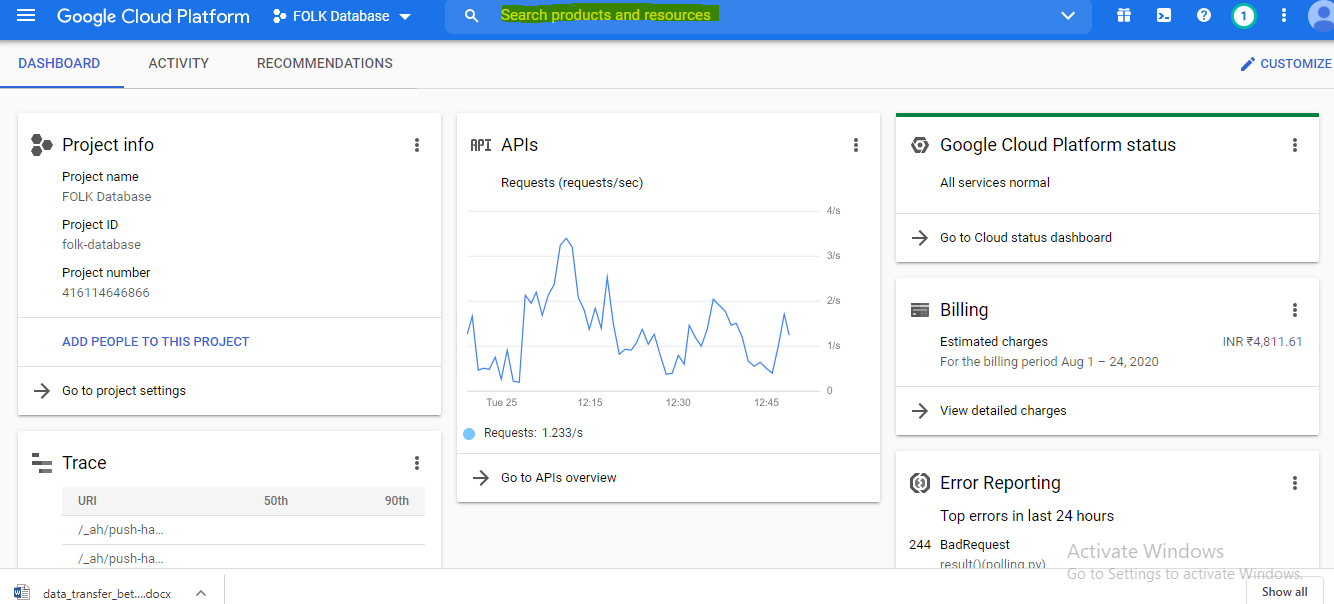
* Before going to cloud pub/sub please see all videos

<https://www.youtube.com/watch?v=cvu53CnZmGI&list=PLIivdWyY5sqKwVLe4BLJ-vlh9r9zCdOse>

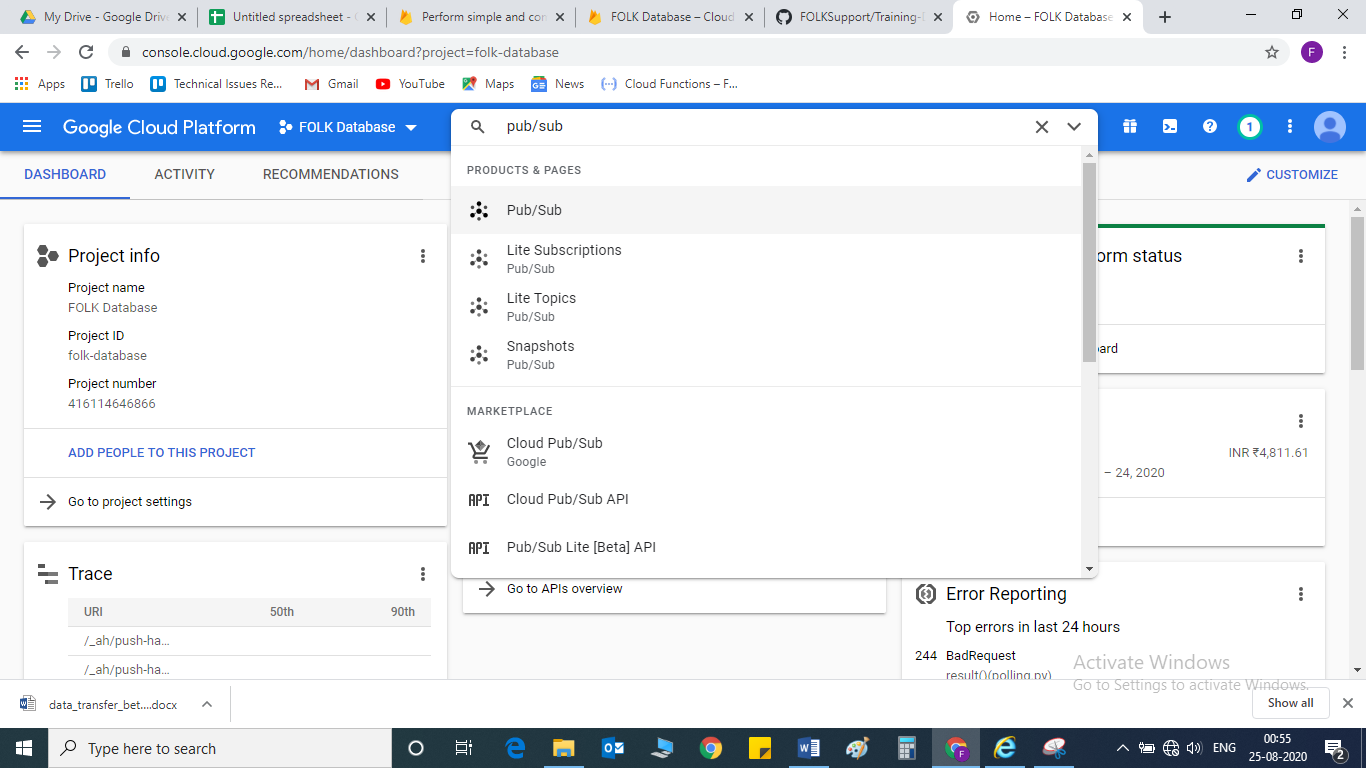
espectially video 9 important

* Now you got overview of cloud pub/sub
* Note: The main use of pub/sub for our FOLK is notification functionality, if we talk technically the use pub/sub is when you call from one cloud function to other cloud function the first cloud function does not bother about response , first cloud function just calls the pub/sub remaining things pub sub will take care
* Lets see an example for pub/sub
* **Here we see how to create pub/sub**

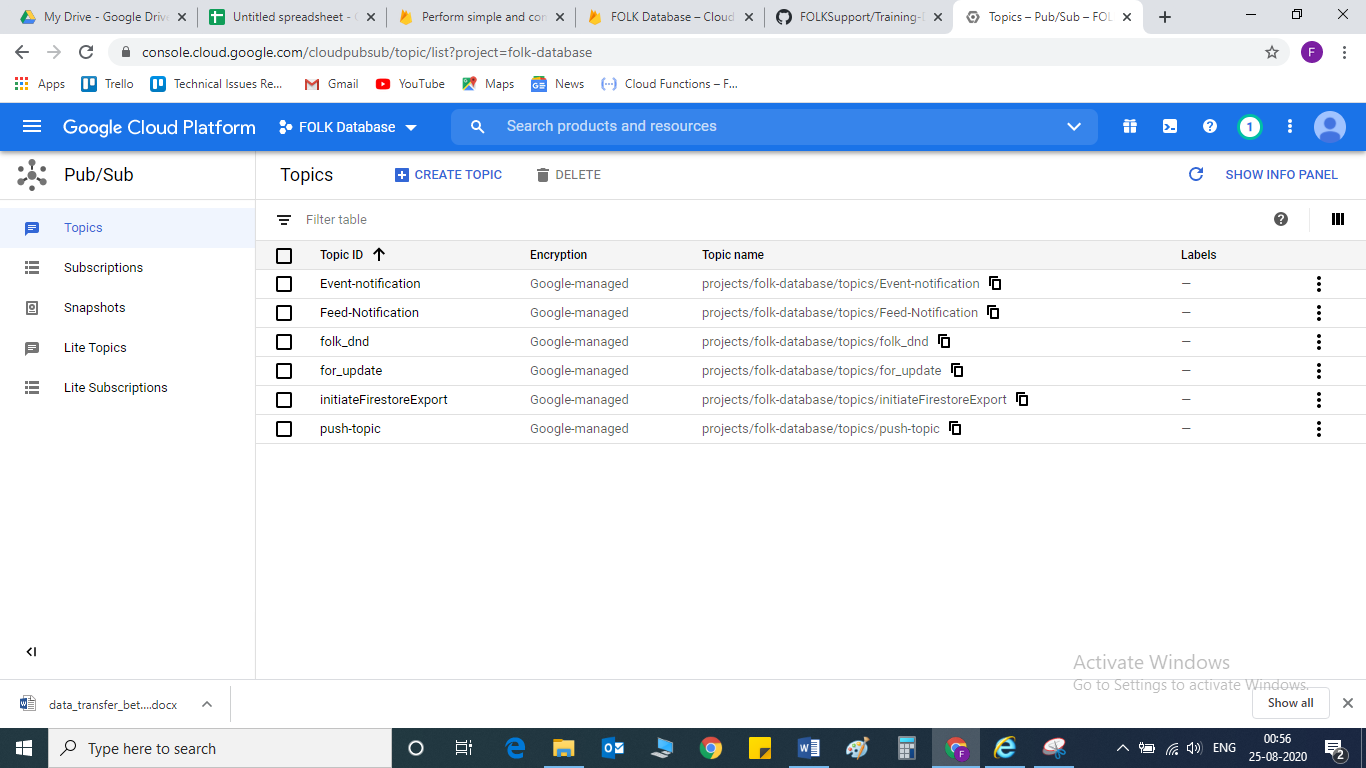
1. Go to google cloud console by clicking below link



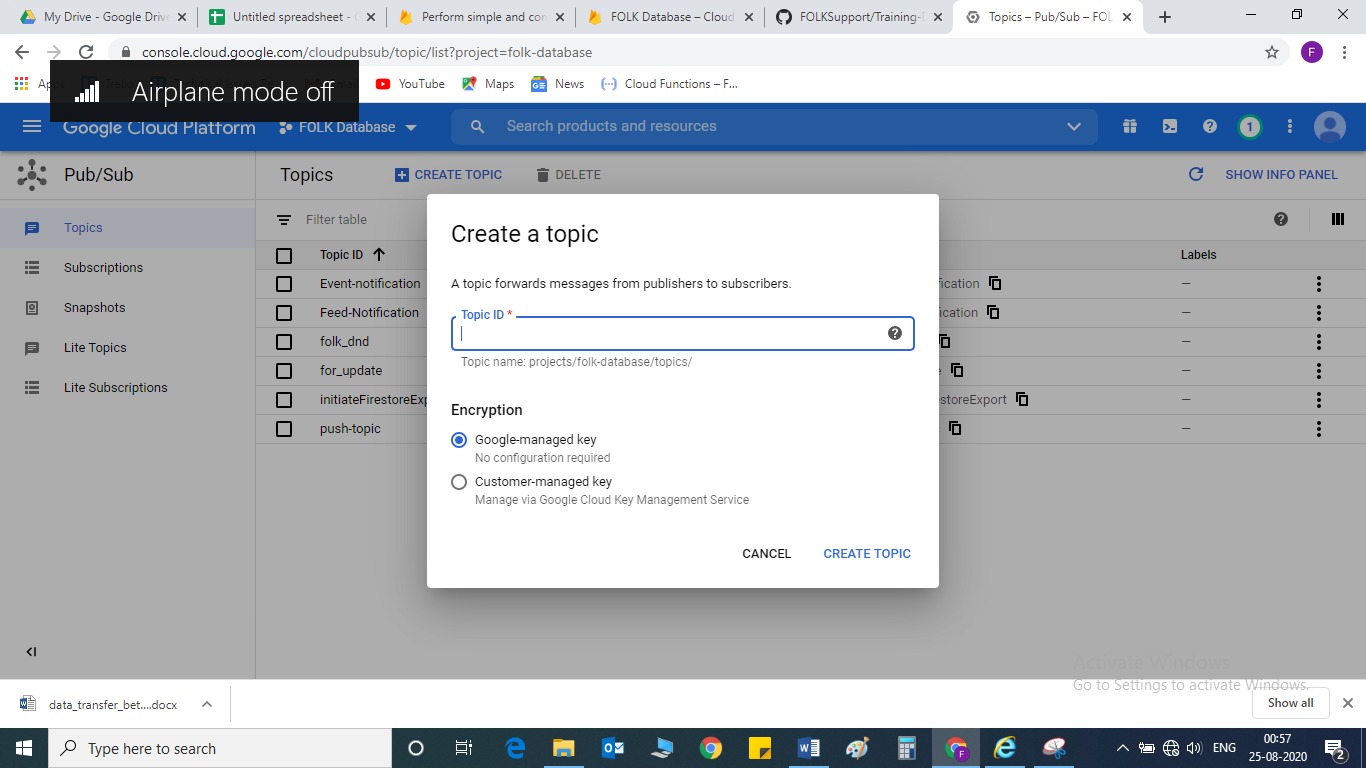
1. In the search box , type pub/sub



1. Select pub/sub



1. In that click on topics->create topic-> click on that



1. Give some topic name and select “google managed key” and then click on create topic
2. After that you can create cloud function inside the pub/sub by clicking trigger cloud function( you can implement your logic based on requirement)

* **now we will see how to call pub/sub:**

**create a cloud function normally in google cloud console, below is the code for main.py**

### (this function name is “Event-Notification-Sending-Data-To-Publisher)

**publisher = pubsub\_v1.PublisherClient()**

**topic\_path = publisher.topic\_path('folk-database', 'Event-notification')**

#**Event-notification** is topic name( which you created in pub sub)

Iam giving here basic structure of calling pub/sub function in main.py

from google.cloud import pubsub\_v1

publisher = pubsub\_v1.PublisherClient()

topic\_path = publisher.topic\_path('folk-database', “topic-name”)

futures = dict()

def get\_callback(f, data):

    def callback(f):

        try:

            print(f.result())

            futures.pop(data)

        except:  # noqa

            print("Please handle {} for {}.".format(f.exception(), data))

    return callback

def ourfunction (request):

    recdata = flask.request.json

data = recdata[‘temp’]

data = str(data)

futures.update({data: None})

# When you publish a message, the client returns a future.

future = publisher.publish(

topic\_path, data=data.encode("utf-8")  # data must be a bytestring.

)

futures[data] = future

# Publish failures shall be handled in the callback function.

future.add\_done\_callback(get\_callback(future, data))

# Wait for all the publish futures to resolve before exiting.

while futures:

time.sleep(5)

print("Published message with error handler.")

Please implement it practically with dummy data and try to complete all videos and read google documents first apply in pycharm after try to implement in google cloud console