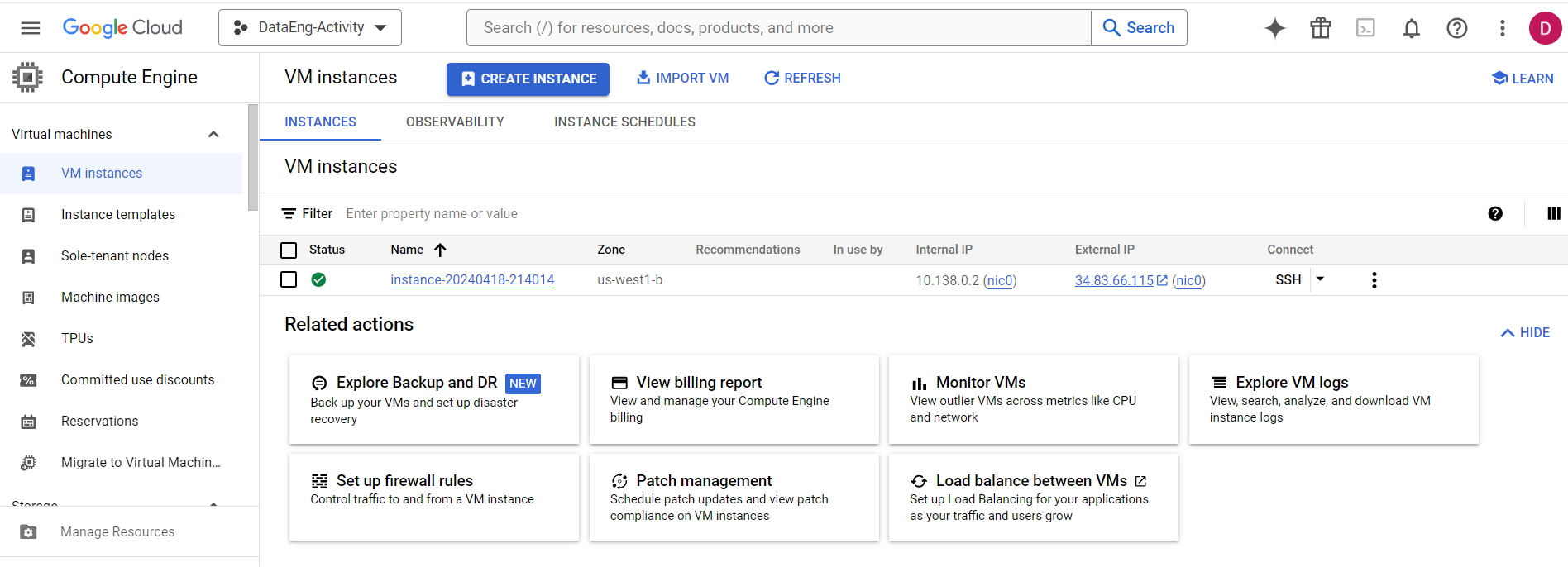
DataEng S24: PubSub

## A. [MUST] PubSub Tutorial

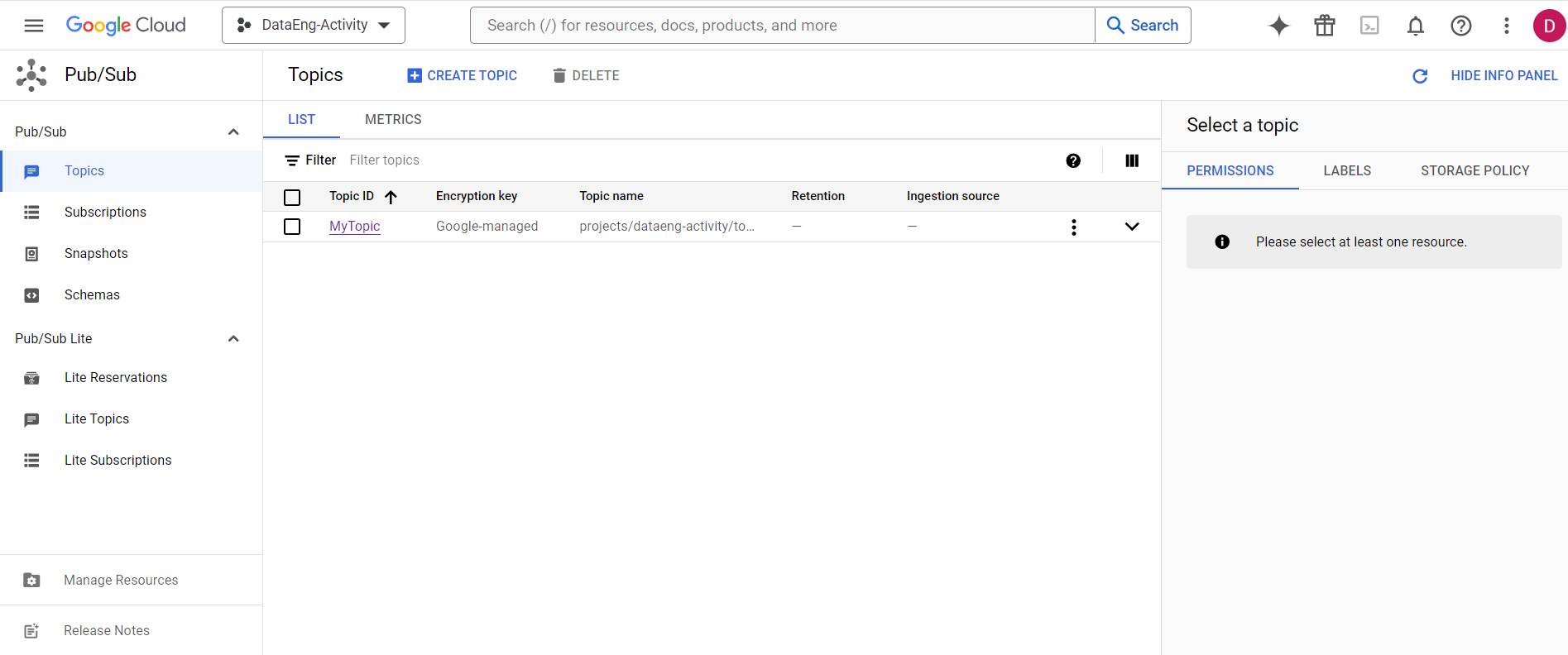
1. Get your cloud.google.com account up and running
   1. Redeem your GCP coupon
   2. Login to your GCP console
   3. Create a new, separate VM instance

**Created a new VM instance in my GCP:**

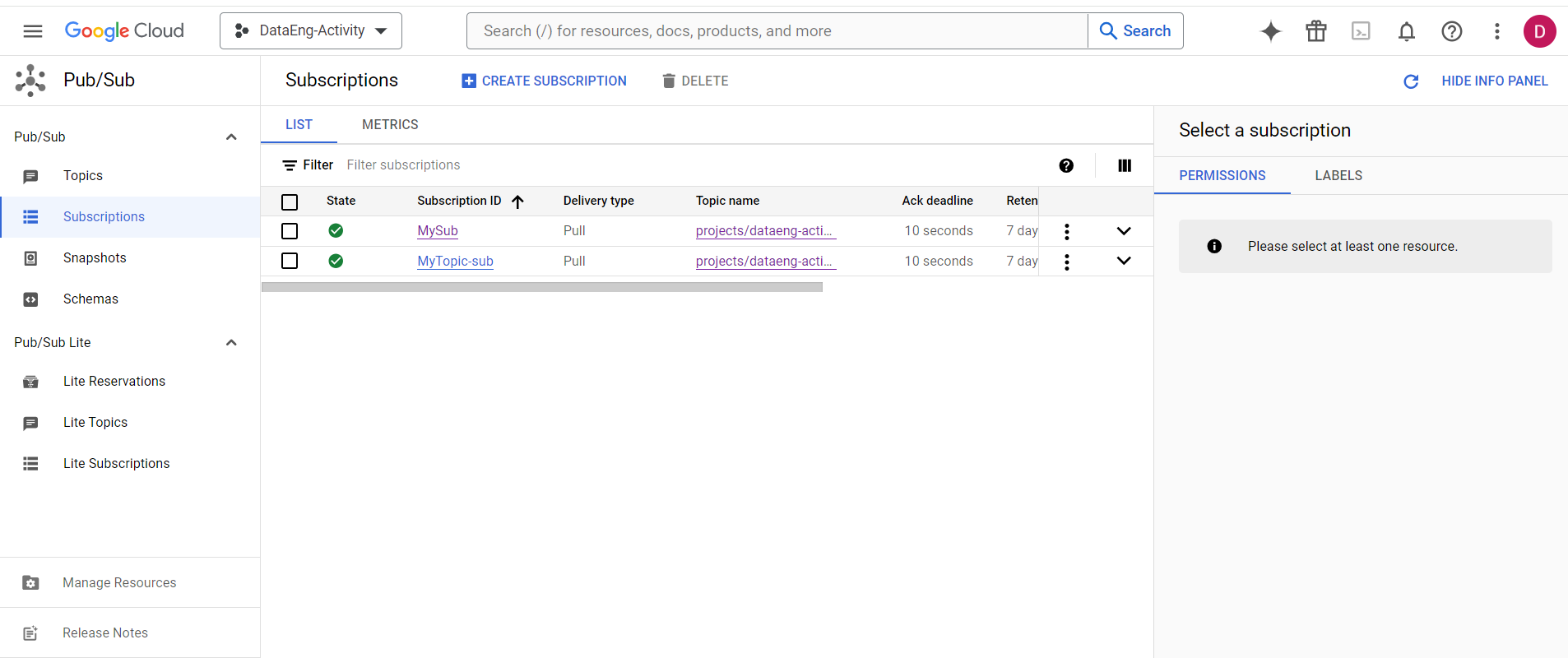


1. Complete this PubSub tutorial: [link](https://cloud.google.com/pubsub/docs/publish-receive-messages-client-library) Note that the tutorial instructs you to destroy your PubSub topic, but you should not destroy your topic just yet. Destroy the topic after you finish the following parts of this in-class assignment.

**Created topics with MyTopic name:**



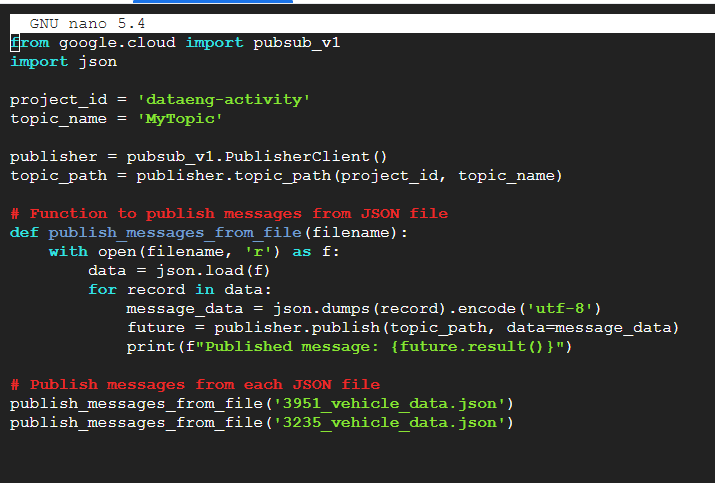
**Created subscriptions:**



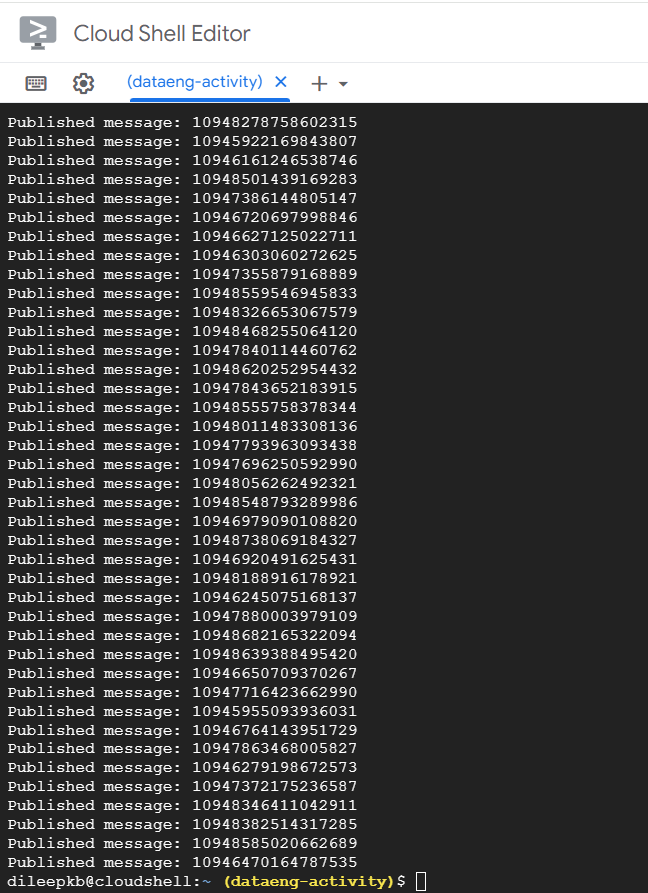
## B. [MUST] Create Sample Data

1. Get data from <https://busdata.cs.pdx.edu/api/getBreadCrumbs> for two Vehicle IDs from among those that have been assigned to you for the class project.
2. Save this data in a sample file (named bcsample.json)
3. Update the publisher python program that you created in the PubSub tutorial to read and parse your bcsample.json file and send its contents, one record at a time, to the my-topic PubSub topic that you created for the tutorial.

**Publisher Code :**

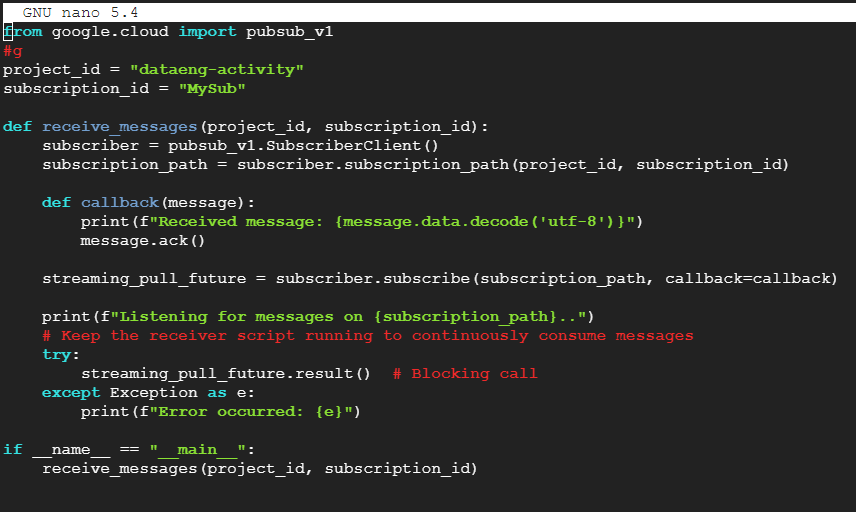
****

**I received message published like this:**

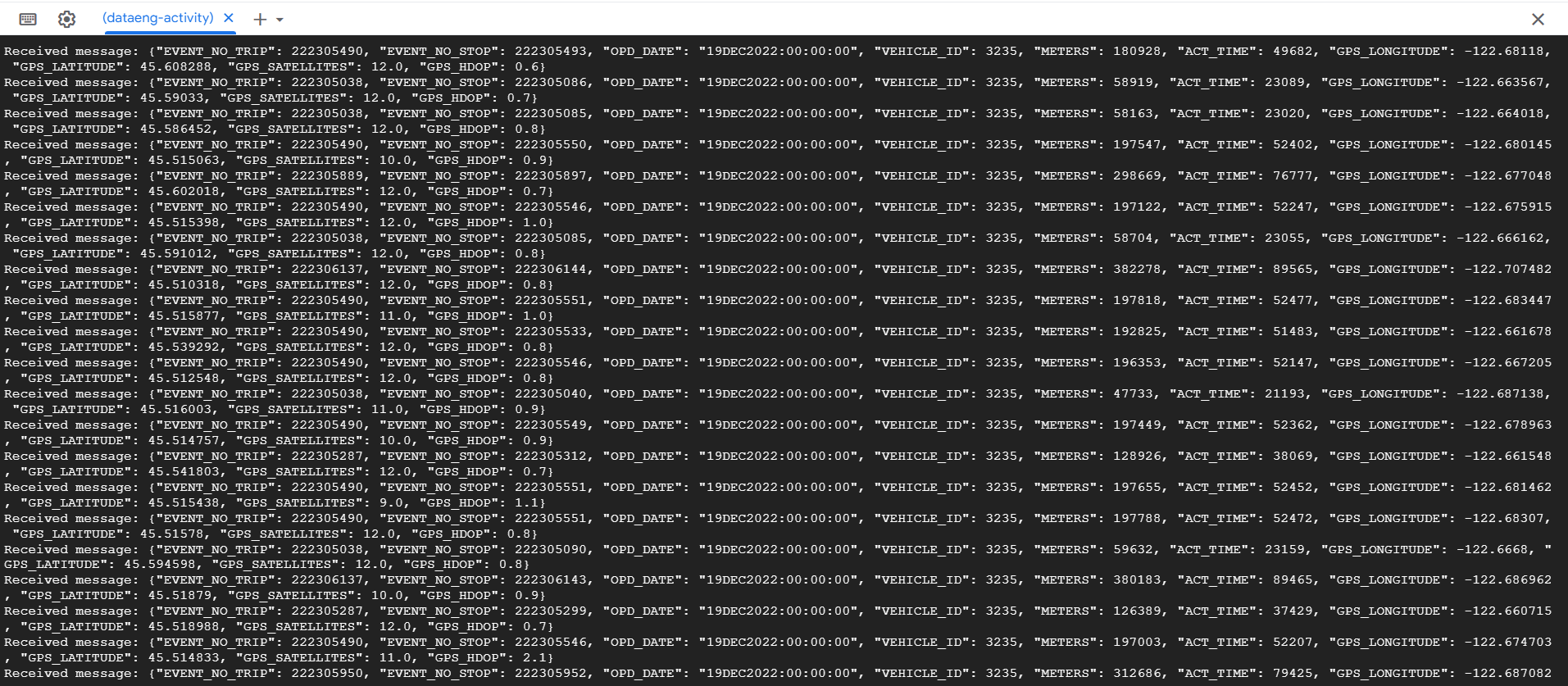


1. Use your receiver python program (from the tutorial) to consume your records.

**Receiver code:**

****

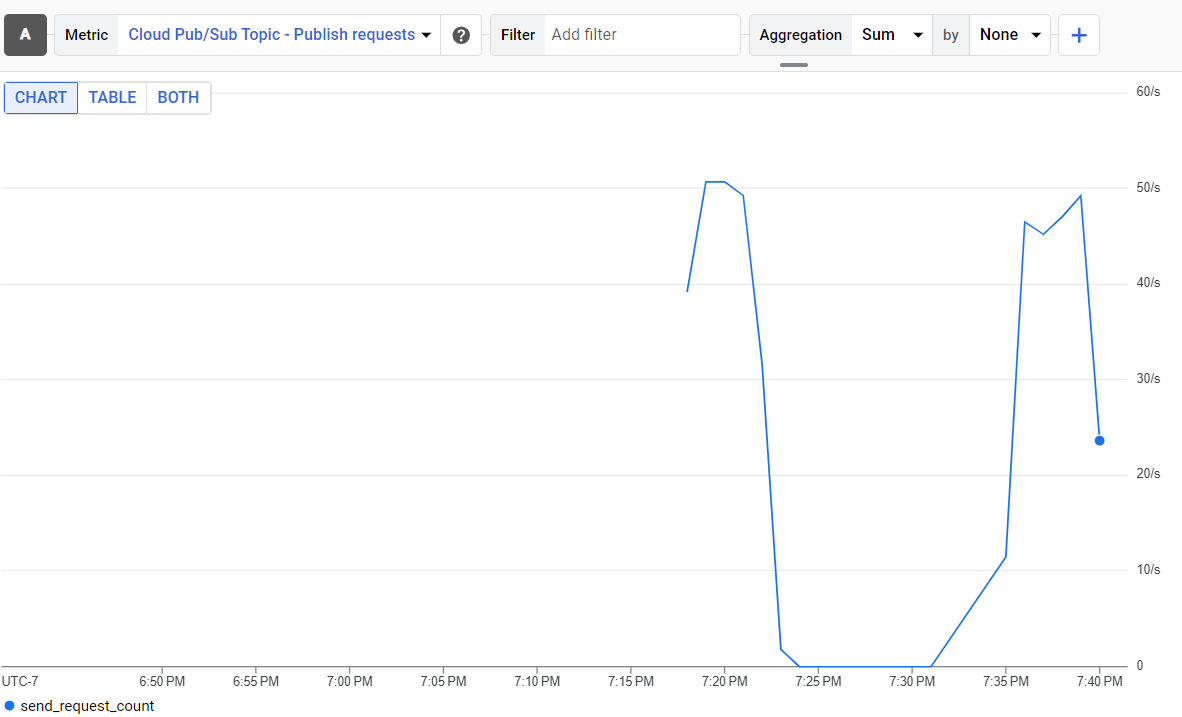
**Received messages:**

****

## C. [MUST] PubSub Monitoring

1. Review the PubSub Monitoring tutorial: [link](https://cloud.google.com/pubsub/docs/monitoring) and work through the steps listed there. You might need to rerun your publisher and receiver programs multiple times to trigger enough activity to monitor your my-topic effectively.

**Cloud PubSub Topic-publish requests:**

****

## D. [MUST] PubSub Storage

1. What happens if you run your receiver multiple times while only running the publisher once?

Ans:  
The publisher reads messages from json and publishes them to the my-topic. The subscriber listens for messages on my-sub. Upon the first run, the subscriber processes all messages published by the single run of the publisher. Subsequent runs of the subscriber do not receive any new messages unless additional messages are published by the publisher.

1. Before the consumer runs, where might the data go, where might it be stored?

Ans:

Before the consumer runs, the data sent by the publisher might be stored temporarily within Pub/Sub. Pub/Sub is a fully managed messaging service provided by Google Cloud, designed to handle large-scale, real-time message ingestion and delivery. When a publisher sends a message to a Pub/Sub topic, the message is stored temporarily in the topic until it is acknowledged by a subscriber. If no subscriber is actively consuming messages from the topic, the messages will remain in the topic for a configurable retention period.

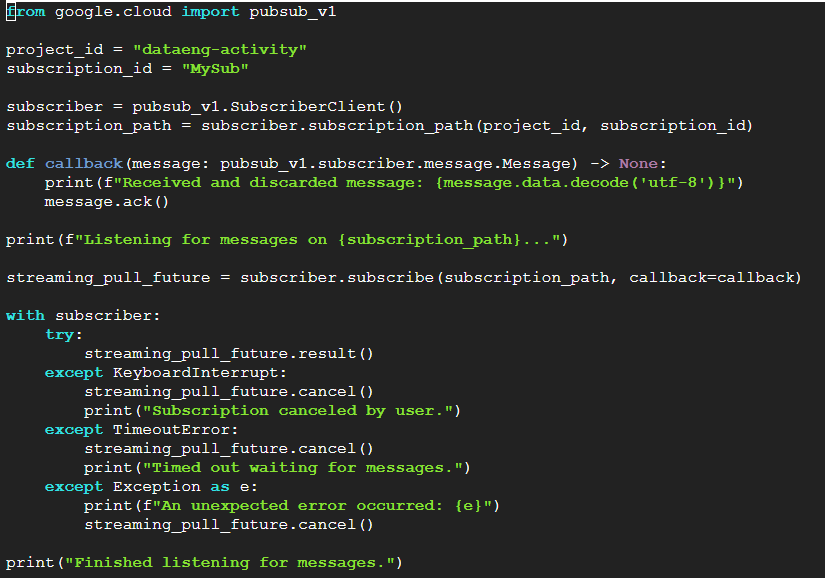
1. Is there a way to determine how much data PubSub is storing for your topic? Do the PubSub monitoring tools help with this?

Ans:

To determine how much data Pub/Sub is storing for your topic, you can use Pub/Sub monitoring tools such as Cloud Monitoring. To track the performance and usage of your subscriptions and topics Pub/Sub exposes various metrics that you can monitor. Metrics include the message backlog size, the number of unacknowledged messages, and the message throughput rate. By these metrics we can gain insights into the amount of data stored in your topic and the overall health of your Pub/Sub system.

1. Create a “topic\_clean.py” receiver program that reads and discards all records for a given topic. This type of program can be very useful for debugging your project code.

**Ans:**

****

## This program creates a subscription to the specified topic and defines a callback function to discard each received message immediately. It then starts consuming messages indefinitely until terminated manually.