

DataEng S24: PubSub

A. [MUST] PubSub Tutorial

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

Created a new VM instance in my GCP:

The screenshot shows the Google Cloud Console interface. The left sidebar displays the 'Compute Engine' menu with 'VM instances' selected. The main content area shows the 'VM instances' page with a table listing one instance: 'instance-20240418-214014' in the 'us-west1-b' zone. Below the table, there are several 'Related actions' cards such as 'Explore Backup and DR', 'View billing report', 'Monitor VMs', 'Explore VM logs', 'Set up firewall rules', 'Patch management', and 'Load balance between VMs'.

2. Complete this PubSub tutorial: [link](#) 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:

The screenshot shows the Google Cloud Console interface for Pub/Sub. The left sidebar displays the 'Pub/Sub' menu with 'Topics' selected. The main content area shows the 'Topics' page with a table listing one topic: 'MyTopic' with a 'Google-managed' encryption key and a topic name of 'projects/dataeng-activity/to...'. On the right side, there is a 'Select a topic' panel with tabs for 'PERMISSIONS', 'LABELS', and 'STORAGE POLICY'. A message at the bottom of the panel says 'Please select at least one resource.'

Created subscriptions:

The screenshot shows the Google Cloud console interface for the 'DataEng-Activity' project. The left sidebar contains navigation options like 'Pub/Sub', 'Topics', 'Snapshots', 'Schemas', 'Pub/Sub Lite', 'Lite Reservations', 'Lite Topics', 'Lite Subscriptions', 'Manage Resources', and 'Release Notes'. The main content area is titled 'Subscriptions' and includes a 'CREATE SUBSCRIPTION' button and a 'DELETE' button. Below this is a table of subscriptions with columns: State, Subscription ID, Delivery type, Topic name, Ack deadline, and Retention. Two subscriptions are listed: 'MySub' and 'MyTopic-sub', both in a 'Pull' state. The right sidebar shows the 'Select a subscription' panel with tabs for 'PERMISSIONS' and 'LABELS'.

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 :

```
GNU nano 5.4
from google.cloud import pubsub_v1
import json

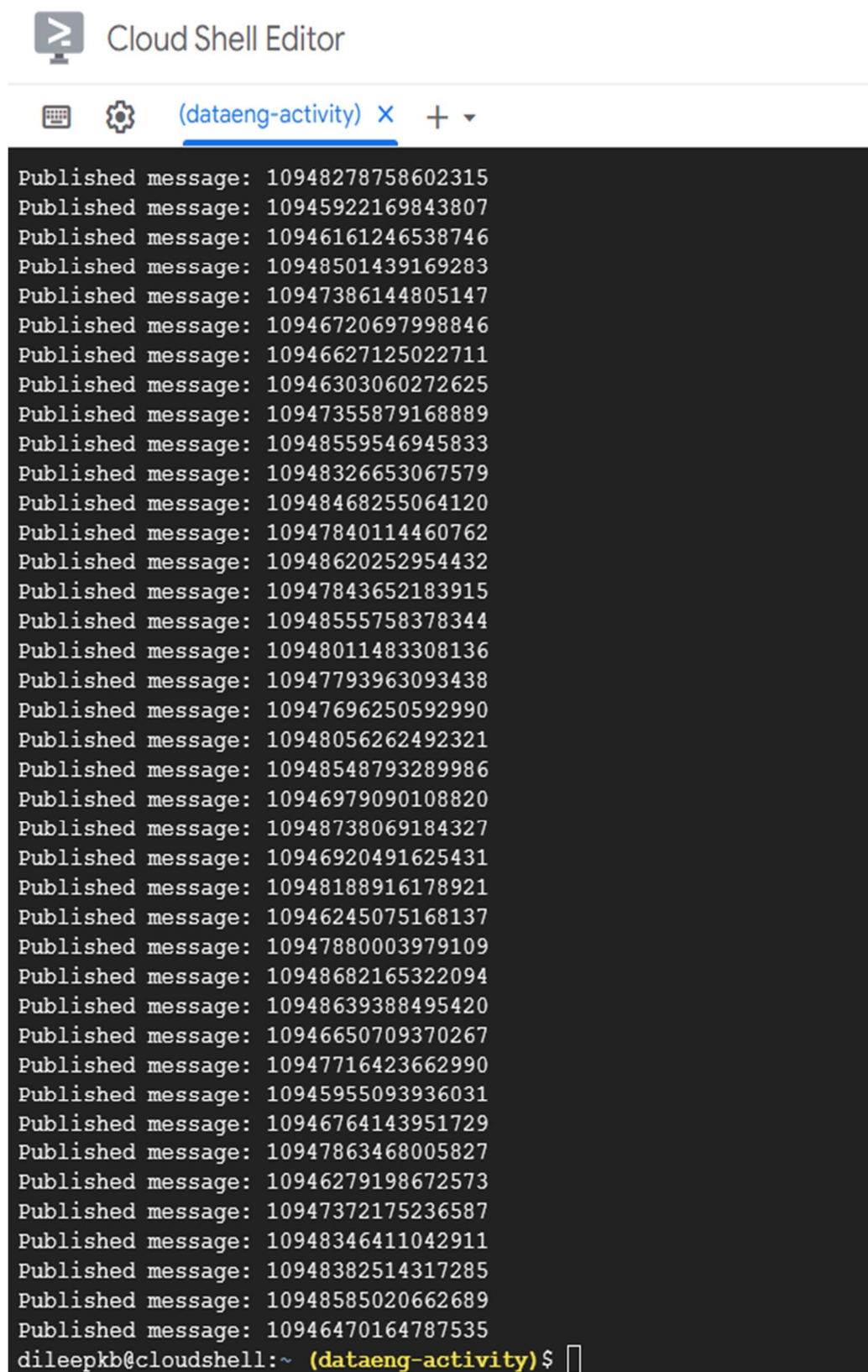
project_id = 'dataeng-activity'
topic_name = 'MyTopic'

publisher = pubsub_v1.PublisherClient()
topic_path = publisher.topic_path(project_id, topic_name)

# Function to publish messages from JSON file
def publish_messages_from_file(filename):
    with open(filename, 'r') as f:
        data = json.load(f)
        for record in data:
            message_data = json.dumps(record).encode('utf-8')
            future = publisher.publish(topic_path, data=message_data)
            print(f"Published message: {future.result()}")

# Publish messages from each JSON file
publish_messages_from_file('3951_vehicle_data.json')
publish_messages_from_file('3235_vehicle_data.json')
```

I received message published like this:



The image shows a screenshot of the Cloud Shell Editor interface. At the top, there is a header bar with the Cloud Shell Editor logo and name. Below the header, there is a tab bar with a single tab labeled "(dataeng-activity)". The main area is a terminal window with a dark background and white text. The terminal displays a list of 35 "Published message:" entries, each followed by a long alphanumeric string. The strings are: 10948278758602315, 10945922169843807, 10946161246538746, 10948501439169283, 10947386144805147, 10946720697998846, 10946627125022711, 10946303060272625, 10947355879168889, 10948559546945833, 10948326653067579, 10948468255064120, 10947840114460762, 10948620252954432, 10947843652183915, 10948555758378344, 10948011483308136, 10947793963093438, 10947696250592990, 10948056262492321, 10948548793289986, 10946979090108820, 10948738069184327, 10946920491625431, 10948188916178921, 10946245075168137, 10947880003979109, 10948682165322094, 10948639388495420, 10946650709370267, 10947716423662990, 10945955093936031, 10946764143951729, 10947863468005827, 10946279198672573, 10947372175236587, 10948346411042911, 10948382514317285, 10948585020662689, 10946470164787535. The terminal ends with a prompt "dileepkb@cloudshell:~ (dataeng-activity)\$" and a cursor.

```
Published message: 10948278758602315
Published message: 10945922169843807
Published message: 10946161246538746
Published message: 10948501439169283
Published message: 10947386144805147
Published message: 10946720697998846
Published message: 10946627125022711
Published message: 10946303060272625
Published message: 10947355879168889
Published message: 10948559546945833
Published message: 10948326653067579
Published message: 10948468255064120
Published message: 10947840114460762
Published message: 10948620252954432
Published message: 10947843652183915
Published message: 10948555758378344
Published message: 10948011483308136
Published message: 10947793963093438
Published message: 10947696250592990
Published message: 10948056262492321
Published message: 10948548793289986
Published message: 10946979090108820
Published message: 10948738069184327
Published message: 10946920491625431
Published message: 10948188916178921
Published message: 10946245075168137
Published message: 10947880003979109
Published message: 10948682165322094
Published message: 10948639388495420
Published message: 10946650709370267
Published message: 10947716423662990
Published message: 10945955093936031
Published message: 10946764143951729
Published message: 10947863468005827
Published message: 10946279198672573
Published message: 10947372175236587
Published message: 10948346411042911
Published message: 10948382514317285
Published message: 10948585020662689
Published message: 10946470164787535
dileepkb@cloudshell:~ (dataeng-activity)$
```

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

Receiver code:

```
GNU nano 5.4
from google.cloud import pubsub_v1
#g
project_id = "dataeng-activity"
subscription_id = "MySub"

def receive_messages(project_id, subscription_id):
    subscriber = pubsub_v1.SubscriberClient()
    subscription_path = subscriber.subscription_path(project_id, subscription_id)

    def callback(message):
        print(f"Received message: {message.data.decode('utf-8')}")
        message.ack()

    streaming_pull_future = subscriber.subscribe(subscription_path, callback=callback)

    print(f"Listening for messages on {subscription_path}..")
    # Keep the receiver script running to continuously consume messages
    try:
        streaming_pull_future.result() # Blocking call
    except Exception as e:
        print(f"Error occurred: {e}")

if __name__ == "__main__":
    receive_messages(project_id, subscription_id)
```

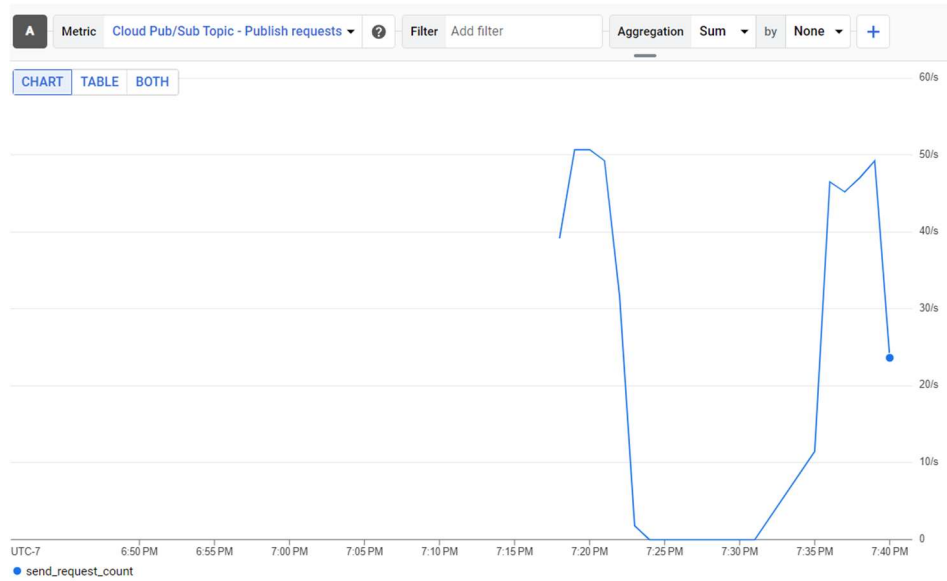
Received messages:

```
(dataeng-activity) X + X
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305493, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 180928, "ACT_TIME": 49682, "GPS_LONGITUDE": -122.68118,
"GPS_LATITUDE": 45.608288, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.6}
Received message: {"EVENT_NO_TRIP": 222305038, "EVENT_NO_STOP": 222305086, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 58919, "ACT_TIME": 23089, "GPS_LONGITUDE": -122.663567,
"GPS_LATITUDE": 45.59033, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.7}
Received message: {"EVENT_NO_TRIP": 222305038, "EVENT_NO_STOP": 222305085, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 58163, "ACT_TIME": 23020, "GPS_LONGITUDE": -122.664018,
"GPS_LATITUDE": 45.586452, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305550, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197547, "ACT_TIME": 52402, "GPS_LONGITUDE": -122.680145,
"GPS_LATITUDE": 45.515063, "GPS_SATELLITES": 10.0, "GPS_HDOP": 0.9}
Received message: {"EVENT_NO_TRIP": 222305889, "EVENT_NO_STOP": 222305897, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 298669, "ACT_TIME": 76777, "GPS_LONGITUDE": -122.677048,
"GPS_LATITUDE": 45.602018, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.7}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305546, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197122, "ACT_TIME": 52247, "GPS_LONGITUDE": -122.675915,
"GPS_LATITUDE": 45.515398, "GPS_SATELLITES": 12.0, "GPS_HDOP": 1.0}
Received message: {"EVENT_NO_TRIP": 222305038, "EVENT_NO_STOP": 222305085, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 58704, "ACT_TIME": 23055, "GPS_LONGITUDE": -122.666162,
"GPS_LATITUDE": 45.591012, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222306137, "EVENT_NO_STOP": 222306144, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 382278, "ACT_TIME": 89565, "GPS_LONGITUDE": -122.707482,
"GPS_LATITUDE": 45.510318, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305551, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197818, "ACT_TIME": 52477, "GPS_LONGITUDE": -122.683447,
"GPS_LATITUDE": 45.515877, "GPS_SATELLITES": 11.0, "GPS_HDOP": 1.0}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305533, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 192825, "ACT_TIME": 51483, "GPS_LONGITUDE": -122.661678,
"GPS_LATITUDE": 45.539292, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305546, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 196353, "ACT_TIME": 52147, "GPS_LONGITUDE": -122.667205,
"GPS_LATITUDE": 45.512548, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222305038, "EVENT_NO_STOP": 222305040, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 47733, "ACT_TIME": 21193, "GPS_LONGITUDE": -122.687138,
"GPS_LATITUDE": 45.516003, "GPS_SATELLITES": 11.0, "GPS_HDOP": 0.9}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305549, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197449, "ACT_TIME": 52362, "GPS_LONGITUDE": -122.678963,
"GPS_LATITUDE": 45.514757, "GPS_SATELLITES": 10.0, "GPS_HDOP": 0.9}
Received message: {"EVENT_NO_TRIP": 222305287, "EVENT_NO_STOP": 222305312, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 128926, "ACT_TIME": 38069, "GPS_LONGITUDE": -122.661548,
"GPS_LATITUDE": 45.541803, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.7}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305551, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197655, "ACT_TIME": 52452, "GPS_LONGITUDE": -122.681462,
"GPS_LATITUDE": 45.515438, "GPS_SATELLITES": 9.0, "GPS_HDOP": 1.1}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305551, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197788, "ACT_TIME": 52472, "GPS_LONGITUDE": -122.68307,
"GPS_LATITUDE": 45.51578, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222305038, "EVENT_NO_STOP": 222305090, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 59632, "ACT_TIME": 23159, "GPS_LONGITUDE": -122.6668, "
GPS_LATITUDE": 45.594598, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.8}
Received message: {"EVENT_NO_TRIP": 222306137, "EVENT_NO_STOP": 222306143, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 380183, "ACT_TIME": 89465, "GPS_LONGITUDE": -122.686962,
"GPS_LATITUDE": 45.51879, "GPS_SATELLITES": 10.0, "GPS_HDOP": 0.9}
Received message: {"EVENT_NO_TRIP": 222305287, "EVENT_NO_STOP": 222305299, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 126389, "ACT_TIME": 37429, "GPS_LONGITUDE": -122.660715,
"GPS_LATITUDE": 45.518988, "GPS_SATELLITES": 12.0, "GPS_HDOP": 0.7}
Received message: {"EVENT_NO_TRIP": 222305490, "EVENT_NO_STOP": 222305546, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 197003, "ACT_TIME": 52207, "GPS_LONGITUDE": -122.674703,
"GPS_LATITUDE": 45.514833, "GPS_SATELLITES": 11.0, "GPS_HDOP": 2.1}
Received message: {"EVENT_NO_TRIP": 222305950, "EVENT_NO_STOP": 222305952, "OPD_DATE": "19DEC2022:00:00:00", "VEHICLE_ID": 3235, "METERS": 312686, "ACT_TIME": 79425, "GPS_LONGITUDE": -122.687082
```


C. [MUST] PubSub Monitoring

1. Review the PubSub Monitoring tutorial: [link](#) 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.

2. 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.

3. 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.

4. 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:

```
from google.cloud import pubsub_v1

project_id = "dataeng-activity"
subscription_id = "MySub"

subscriber = pubsub_v1.SubscriberClient()
subscription_path = subscriber.subscription_path(project_id, subscription_id)

def callback(message: pubsub_v1.subscriber.message.Message) -> None:
    print(f"Received and discarded message: {message.data.decode('utf-8')}")
    message.ack()

print(f"Listening for messages on {subscription_path}...")

streaming_pull_future = subscriber.subscribe(subscription_path, callback=callback)

with subscriber:
    try:
        streaming_pull_future.result()
    except KeyboardInterrupt:
        streaming_pull_future.cancel()
        print("Subscription canceled by user.")
    except TimeoutError:
        streaming_pull_future.cancel()
        print("Timed out waiting for messages.")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
        streaming_pull_future.cancel()

print("Finished listening for messages.")
```

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.