Lab: Streaming Data Processing: Publish Streaming Data into Pub/Sub

Preparation

✓ Open the SSH terminal and connect to the training VM

Compute Engine > VM instances > Connect

✓ **In SSH terminal Verify initialization is complete**

ls /training

✓ **In SSH terminal Download Code Repository**

git clone https://github.com/GoogleCloudPlatform/training-data-analyst

✓ **In SSH terminal Identify a project**

export DEVSHELL_PROJECT_ID=$(gcloud config get-value project)

Create Pub/Sub topic and subscription

1.On the **training-vm** SSH terminal, navigate to the directory for this lab.

cd ~/training-data-analyst/courses/streaming/publish

Verify that the Pub/Sub service is accessible and working using the gcloud command.

2.Create your topic and publish a simple message.

gcloud pubsub topics create sandiego

3.Publish a simple message.

gcloud pubsub topics publish sandiego --message "hello“

4.Create a subscription for the topic.

gcloud pubsub subscriptions create --topic sandiego mySub1

5.Pull the first message that was published to your topic.

gcloud pubsub subscriptions pull --auto-ack mySub1

Do you see any result? If not, why?

6. Try to publish another message and then pull it using the subscription.

gcloud pubsub topics publish sandiego --message "hello again“

gcloud pubsub subscriptions pull --auto-ack mySub1

Create Pub/Sub topic and subscription

Did you get any response this time?
Output:

DATA	MESSAGE_ID	ATTRIBUTES
hello again	38138015771622	

7.In the **training-vm** SSH terminal, cancel your subscription.
gcloud pubsub subscriptions delete mySub1

Simulate traffic sensor data into Pub/Sub

1.Explore the python script to simulate San Diego traffic sensor data. **Do not make any changes to the code.**

```
cd ~/training-data-analyst/courses/streaming/publish
nano send_sensor_data.py
```

Look at the simulate function. This one lets the script behave as if traffic sensors were sending in data in real time to Pub/Sub. The speedFactor parameter determines how fast the simulation will go. Exit the file by pressing **Ctrl+X**.

2. Download the traffic simulation dataset.

```
./download_data.sh
```

Simulate streaming sensor data

3.Run the **send_sensor_data.py**.

```
./send_sensor_data.py --speedFactor=60 --project $DEVSHELL_PROJECT_ID
```

This command simulates sensor data by sending recorded sensor data via Pub/Sub messages. The script extracts the original time of the sensor data and pauses between sending each message to simulate realistic timing of the sensor data. The value **speedFactor** changes the time between messages proportionally. So a **speedFactor** of 60 means "60 times faster" than the recorded timing. It will send about an hour of data every 60 seconds.
Leave this terminal open and the simulator running.

Verify that messages are received

- ✓ **Open a second SSH terminal and connect to the training VM**
- ✓ **cd ~/training-data-analyst/courses/streaming/publish**
- ✓ Create a subscription for the topic and do a pull to confirm that messages are coming in (note: you may need to issue the 'pull' command more than once to start seeing messages):
  gcloud pubsub subscriptions create --topic sandiego mySub2
  gcloud pubsub subscriptions pull --auto-ack mySub2
  Confirm that you see a message with traffic sensor information.

```
student-00-b60ed2d43003@training-vm:~/training-data-analyst/courses/streaming/publish$ gcloud pubsub subscriptions
pull --auto-ack mySub2
```

DATA	MESSAGE_ID	ATTRIBUTES
2008-11-01 00:45:00,32.780922,-117.089026,8,W,1,78.9	1194028531598790	

```
student-00-b60ed2d43003@training-vm:~/training-data-analyst/courses/streaming/publish$
```

- ✓ Cancel this subscription. **gcloud pubsub subscriptions delete mySub2**
- ✓ Close the second terminal.
- ✓ In first terminal **Ctrl+C** and exit it