**Overview**

Google Cloud Pub/Sub is a fully-managed real-time messaging service that allows you to send and receive messages between independent applications. Use Cloud Pub/Sub to publish and subscribe to data from multiple sources, then use Google Cloud Dataflow to understand your data, all in real time.

In this lab, you will use simulate your traffic sensor data into a Pub/Sub topic for later to be processed by Dataflow pipeline before finally ending up in a BigQuery table for further analysis.

At the time of this writing, streaming pipelines are not available in the DataFlow Python SDK. So the streaming labs are written in Java.

**Objectives**

In this lab, you will perform the following tasks:

* Create a Pub/Sub topic and subscription
* Simulate your traffic sensor data into Pub/Sub

## Task 1: Preparation

You will be running a sensor simulator from the training VM. There are several files and some setup of the environment required.

### **Open the SSH terminal and connect to the training VM**

1. In the Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. Locate the line with the instance called **training-vm**.
3. On the far right, under **Connect**, click on **SSH** to open a terminal window.
4. In this lab, you will enter CLI commands on the **training-vm**.

### **Verify initialization is complete**

1. The **training-vm** is installing software in the background. Verify that setup is complete by checking that the following directory exists. If it does not exist, wait a few minutes and try again.

ls /training

Wait until setup is complete before proceeding. You can verify the installation of python with **pip --version**.

### **Copy files**

1. A repository has been downloaded to the VM. Copy the repository to your home directory.

cp -r /training/training-data-analyst/ .

### **Identify a project**

One environment variable that you will set is **$DEVSHELL\_PROJECT\_ID** that contains the Google Cloud project ID required to access billable resources.

1. In the Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **Home**. In the panel with Project Info, the **Project ID** is listed. You can also find this information in the Qwiklabs tab under Connection Details, where it is labeled **GCP Project ID**.
2. On the **training-vm** SSH terminal, set the DEVSHELL\_PROJECT\_ID environment variable and export it so it will be available to other shells.

export DEVSHELL\_PROJECT\_ID=<project-id>

## Task 2: 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.**

1. Create your topic and publish a simple message.

gcloud pubsub topics create sandiego

1. Publish a simple message.

gcloud pubsub topics publish sandiego --message "hello"

1. Create a subscription for the topic.

gcloud pubsub subscriptions create --topic sandiego mySub1

1. 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?

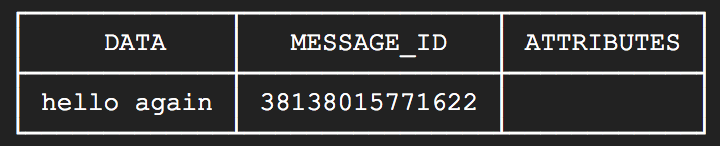
1. Try to publish another message and then pull it using the subscription.

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

gcloud pubsub subscriptions pull --auto-ack mySub1

Did you get any response this time?

Output:



1. Return to the Console tab. On the **Navigation menu** ( 7a91d354499ac9f1.png), click **Pub/Sub > Topics.**
2. You should see a line with the **Topic Name** ending in sandiego and the number of **Subscriptions** set to 1.
3. In the **training-vm** SSH terminal, cancel your subscription.

gcloud pubsub subscriptions delete mySub1

1. Return to the Console tab. Refresh the browser and you should see the **Subscriptions** drop to 0.

## Task 3: 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**.

1. Download the traffic simulation dataset.

./download\_data.sh

### **Install API support**

1. Install the Python PIP program required to install the API.

sudo apt-get install -y python-pip

1. Use PIP to install the Google Cloud Pub/Sub API.

sudo pip install -U google-cloud-pubsub

### **Simulate streaming sensor data**

1. 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.

## Task 4: Verify that messages are received

### **Open a second SSH terminal and connect to the training VM**

1. In the Console, on the **Navigation menu** ( 7a91d354499ac9f1.png), click **Compute Engine** > **VM instances**.
2. Locate the line with the instance called **training-vm**.
3. On the far right, under **Connect**, click on **SSH** to open a second terminal window.
4. Change into the directory you were working in:

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

1. Create a subscription for the topic and do a pull to confirm that messages are coming in:

gcloud pubsub subscriptions create --topic sandiego mySub2

gcloud pubsub subscriptions pull --auto-ack mySub2

Confirm that you see a message with traffic sensor information.

1. Cancel this subscription.

gcloud pubsub subscriptions delete mySub2

1. Close the second terminal.

exit

### **Stop the sensor simulator**

1. Return to the first terminal.
2. Interrupt the publisher by typing **Ctrl+C** to stop it.
3. Close the first terminal.

exit