**LAB:** **Create a Streaming Data Pipeline for a Real-Time Dashboard with Cloud Dataflow**

**Overview**

In this lab, you own a fleet of New York City taxi cabs and are looking to monitor how well your business is doing in real-time. You will build a streaming data pipeline to capture taxi revenue, passenger count, ride status, and much more and visualize the results in a management dashboard.

1. **Open BigQuery Console**

In the Google Cloud Console, select **Navigation menu** > **BigQuery**:

1. **Confirm that Needed APIs Are Enabled**

Make a note of the name of your GCP project. This value is shown in the top bar of the GCP Console.

* In the GCP Console, in the **Navigation menu**, click **Home.**
* In the **Project Info** section, copy and save your Project ID value for later use. Your project ID will resemble qwiklabs-gcp-d2e509fed105b3ed.
* In the GCP Console, in the Navigation menu, click **APIs & services.**
* Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:

**Cloud Pub/Sub API**

**Dataflow API**

* If one or more API is not enabled, click the **ENABLE APIS AND SERVICES** button at the top. Search for the APIs by name and enable each API for your current project.

1. **Create a Cloud Pub/Sub Topic**

Cloud Pub/Sub is an asynchronous global messaging service. By decoupling senders and receivers, it allows for secure and highly available communication between independently written applications. Cloud Pub/Sub delivers low-latency, durable messaging.

With the command line giving the topic the name taxirides-realtime

**gcloud pubsub topics create *taxirides-realtime***

Google maintains a few public Pub/Sub streaming data topics for labs like this one. We'll be using the NYC Taxi & Limousine Commission’s open dataset. <https://opendata.cityofnewyork.us/>

1. **Create a BigQuery dataset**

BigQuery is a serverless data warehouse. Tables in BigQuery are organized into datasets. In this lab, messages published into Pub/Sub will be aggregated and stored in BigQuery.

To create a new BigQuery dataset:

Command Line

1. Open **Cloud Shell** and run the below command to create the taxirides dataset

*bq mk taxirides*

1. Run this command to create the taxirides.realtime table (empty schema we will stream into later)

**bq mk --time\_partitioning\_field timestamp --schema ride\_id:string,point\_idx:integer,latitude:float,longitude:float,timestamp:timestamp,meter\_reading:float,meter\_increment:float,ride\_status:string, passenger\_count:integer -t taxirides.realtime**

1. **Create a Cloud Storage Bucket**

In this lab we will use Cloud Storage to provide working space for our Cloud Dataflow pipeline.

To create a new bucket, use the command

**gsutil mb gs://<projectId>**

Use your projectID to make your bucket name unique.

1. **Set up a Cloud Dataflow Pipeline**

Cloud Dataflow is a serverless way to carry out data analysis. In this lab, you will set up a streaming data pipeline to read sensor data from Pub/Sub, compute the maximum temperature within a time window, and write this out to BigQuery.

To Set up for a cloud dataflow pipeline use the command

**gcloud dataflow jobs run streaming-taxi-pipeline --gcs-location gs://dataflow-templates-us-central1/latest/PubSub\_to\_BigQuery --region us-central1 --staging-location gs://qwiklabs-gcp-02-0a872f7591a4/tmp --parameters inputTopic=projects/pubsub-public-data/topics/taxirides-realtime,outputTableSpec=qwiklabs-gcp-02-0a872f7591a4:taxirides.realtime**

1. **Analyze the Taxi Data Using BigQuery**

To analyze the data as it is streaming:

1. In the GCP Console, open the Navigation menu and select **BigQuery**.
2. Enter the following query in the Query editor and click **RUN**:

**SELECT \* FROM taxirides.realtime LIMIT 10**

1. If no records are returned, wait another minute and re-run the above query (Dataflow takes 3-5 minutes to setup the stream). You will receive a similar output.
2. **Perform aggregations on the stream for reporting**
3. Copy and paste the below query and run

**WITH streaming\_data AS (**

**SELECT**

**timestamp,**

**TIMESTAMP\_TRUNC(timestamp, HOUR, 'UTC') AS hour,**

**TIMESTAMP\_TRUNC(timestamp, MINUTE, 'UTC') AS minute,**

**TIMESTAMP\_TRUNC(timestamp, SECOND, 'UTC') AS second,**

**ride\_id,**

**latitude,**

**longitude,**

**meter\_reading,**

**ride\_status,**

**passenger\_count**

**FROM**

**taxirides.realtime**

**WHERE ride\_status = 'dropoff'**

**ORDER BY timestamp DESC**

**LIMIT 100000**

**)**

**# calculate aggregations on stream for reporting:**

**SELECT**

**ROW\_NUMBER() OVER() AS dashboard\_sort,**

**minute,**

**COUNT(DISTINCT ride\_id) AS total\_rides,**

**SUM(meter\_reading) AS total\_revenue,**

**SUM(passenger\_count) AS total\_passengers**

**FROM streaming\_data**

**GROUP BY minute, timestamp**

The result shows key metrics by the minute for every taxi drop-off

1. **Create a Real-Time Dashboard**
2. Click **Explore Data** and then, select **Explore with Data Studio**.
3. On the **Welcome** page, click on **GET STARTED**.
4. On the Next page, click **Authorize**.

**Note:** If you are getting the prompt *Oops… Not able to connect to your data* then click **Back**. Click **Save** in save data studio explorer.

1. Click on **GET STARTED** and acknowledge the Terms of Service. Click **Accept**.
2. Select **No, thanks** for all in preferences and click **Done**.
3. Refresh the tab to load the data.
4. Specify the below settings:

* Chart type: column chart
* Date range dimension: dashboard\_sort
* Drill down: dashboard\_sort (Make sure that **Drill down** option is turned **ON**.)
* Dimension: dashboard\_sort, minute
* Metric: SUM() total\_rides, SUM() total\_passengers, SUM() total\_revenue (If **Record Count** is present then, mouse over **Record Count** and click the (x) to remove it.)
* Sort: dashboard\_sort Ascending (latest rides first)

Note: Visualizing data at a minute-level granularity is currently not supported in Data Studio as a timestamp. This is why we created our own dashboard\_sort dimension.

1. When you're happy with your dashboard, click Save to save this data source
2. If prompted then select following,

* On the Welcome page, click on **GET STARTED**.
* On the **Terms** page, click on the checkbox to acknowledge the terms and click **ACCEPT**.
* On the **Preferences** page, select **No, thanks** for each option to receive email notifications, and click **DONE**.

1. Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking on the Refresh button near the Save button.
2. **Stop the Cloud Dataflow job**

Using the command line to stop(Drain) the pipeline I ran

I used the job\_id that was given to it

**gcloud dataflow jobs drain** ***[JOB\_ID]* –us-central1**