

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

1 hour 10 minutes

1 Credit



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.

Setup and requirements

Qwiklabs setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click Start Lab, shows how long Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access the Google Cloud Platform for the duration of the lab.

What you need

To complete this lab, you need:

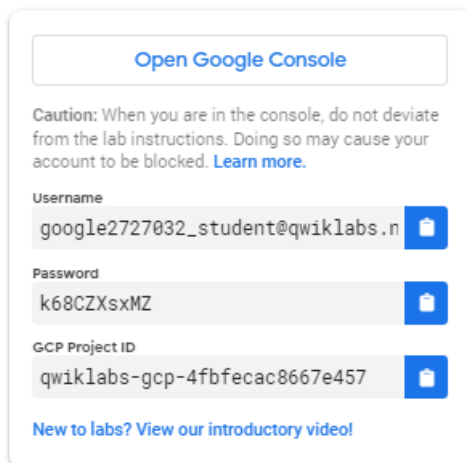
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal GCP account or project, do not use it for this lab.

Google Cloud Platform Console

How to start your lab and sign in to the Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.

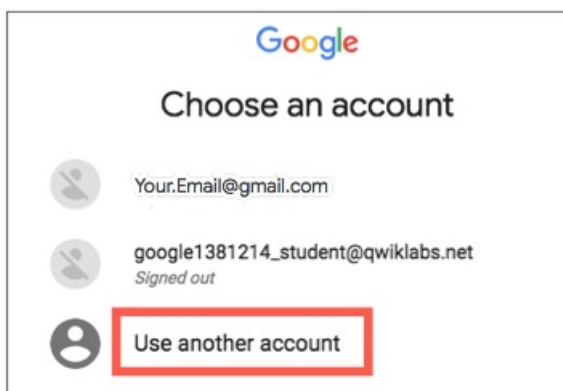


A screenshot of a web panel titled "Open Google Console". At the top is a button labeled "Open Google Console". Below it is a caution message: "Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)". The panel contains three input fields, each with a blue copy icon to its right: "Username" with the value "google2727032_student@qwiklabs.n", "Password" with the value "k68CZxsxMZ", and "GCP Project ID" with the value "qwiklabs-gcp-4fbfecac8667e457". At the bottom is a link: "New to labs? View our introductory video!"

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Choose an account** page.

Tip: Open the tabs in separate windows, side-by-side.

3. On the Choose an account page, click **Use Another Account**.



4. The Sign in page opens. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.

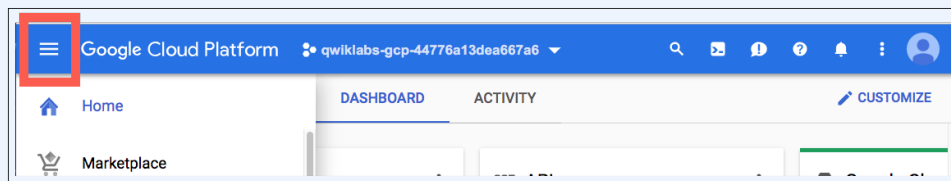
Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own GCP account, do not use it for this lab (avoids incurring charges).

5. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

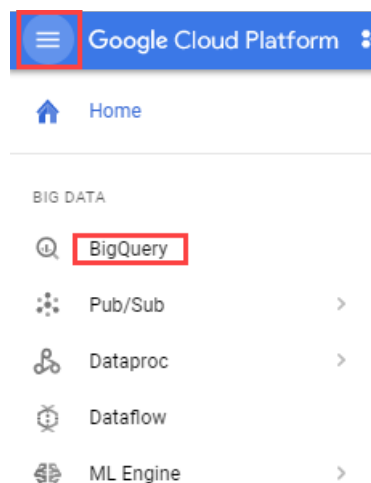
After a few moments, the GCP console opens in this tab.

Note: You can view the menu with a list of GCP Products and Services by clicking the **Navigation menu** at the top-left, next to "Google Cloud Platform".



Open BigQuery Console

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



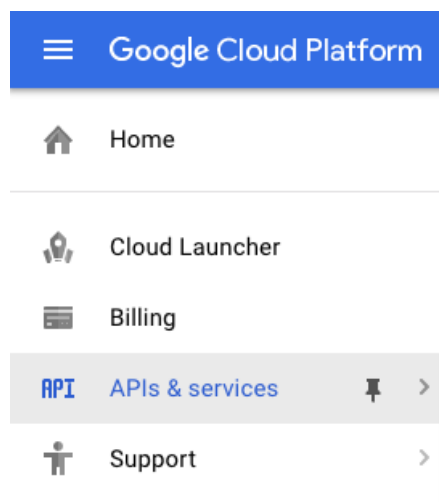
The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

Click **Done**.

Note your Project Name; 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.

1. In the GCP Console, in the **Navigation menu**, click **Home**.
2. In the **Project Info** section, copy and save your Project ID value for later use. Your project ID will resemble `qwiklabs-gcp-d2e509fed105b3ed`.
3. In the GCP Console, in the Navigation menu, click **APIs & services**.



4. Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:
 - **Google Cloud Pub/Sub API**
 - **Dataflow API**
5. 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.

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.

In Cloud Pub/Sub, publisher applications and subscriber applications connect with one another through the use of a shared string called a **topic**. A publisher application creates and sends messages to a topic. Subscriber applications create a subscription to a topic to receive messages from it.

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](#)

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:

Option 1: Command Line

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

```
bq mk taxirides
```

2. 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:st  
passenger_count:integer -t taxirides.realtime
```

Option 2: BigQuery Console UI

Skip these steps if you created the tables using the command line

1. In the GCP Console, go to **Navigation menu > BigQuery**.
2. Once there, click on your GCP Project ID from the left-hand menu

3. Now on the right-hand side of the console underneath the query editor click **CREATE DATASET**.
4. Give the new dataset the name **taxirides**, leave all the other fields the way they are, and click **Create dataset**.
5. If you look at the left-hand resources menu, you should see your newly created dataset
6. Click on the **taxirides** dataset
7. Click **create table**
8. Name the table **realtime**
9. For schema, click **edit as text** and paste in the below

```
ride_id:string,  
point_idx:integer,  
latitude:float,  
longitude:float,  
timestamp:timestamp,  
meter_reading:float,  
meter_increment:float,  
ride_status:string,  
passenger_count:integer
```

10. Under **Partition and cluster settings**, select the **timestamp** option for the Partitioning field.
11. Confirm against the below screenshot:

Table name

realtime

Schema

☐ Edit as text

```

1 ride_id:string,
2 point_idx:integer,
3 latitude:float,
4 longitude:float,
5 timestamp:timestamp,
6 meter_reading:float,
7 meter_increment:float,
8 ride_status:string,
9 passenger_count:integer

```

Partition and cluster settings

Partitioning: Partitioning filter: ☐ Require partition filter

Clustering order (optional):

Advanced options ☐

12. Click the **Create Table** button.

Create a Cloud Storage Bucket

Skip this step if you already have a bucket created

[Cloud Storage](#) allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download. In this lab we will use Cloud Storage to provide working space for our Cloud Dataflow pipeline.

1. In the GCP Console, go to **Navigation menu > Storage**.
2. Click **CREATE BUCKET**.
3. For **Name**, paste in your GCP project ID.
4. For **Default storage class**, click **Multi-regional** if it is not already selected.
5. For **Location**, choose the selection closest to you.

6. Click **Create**.

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.

1. In the GCP Console, go to **Navigation menu > Dataflow**.
2. In the top menu bar, click **CREATE JOB FROM TEMPLATE**.
3. Enter **streaming-taxi-pipeline** as the Job name for your Cloud Dataflow job.
4. Under **Cloud Dataflow template**, select the Cloud Pub/Sub Topic to BigQuery template.
5. Under **Cloud Pub/Sub input topic**, enter `projects/pubsub-public-data/topics/taxirides-realtime`
6. Under **BigQuery output table**, enter `<myprojectid>:taxirides.realtime`

Note: there is a colon `:` between the project and dataset name and a dot `.` between the dataset and table name


7. Under **Temporary Location**, enter `gs://<mybucket>/tmp/`
8. Click the **Run job** button.


A new streaming job has started! You can now see a visual representation of the data pipeline.

 Dataflow

[←](#) Create job from template

Job name
Must be unique among running jobs. Use lowercase letters, numbers, and hyphens (-).

Regional endpoint 
Choose where to deploy Cloud Dataflow workers and store metadata for the job.

Cloud Dataflow template 
A pipeline that ingests a Cloud Pub/Sub stream of JSON-encoded messages, performs a transform via a user defined javascript function, and writes to a pre-existing BigQuery table.

Parameters
Cloud Pub/Sub input topic
Cloud Pub/Sub topic to read the input from, in the format of 'projects/<project>/topics/<topic>'

GCS location of your Javascript UDF (Optional)
The full URL of your .js file. Example: gs://my_bucket/my_function.js

The name of the javascript function you wish to call as your UDF (Optional)
The function name should only contain letters, digits and underscores. Example: 'transform' or 'transform_udf1'.

BigQuery output table
BigQuery table location (<project>:<dataset>.<table_name>) to write the output to. The table's schema must match the input JSON objects.

Temporary Location
Path and filename prefix for writing temporary files. ex: gs://MyBucket/tmp

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
```

3. 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:

Query complete (1.7 sec elapsed, 0 B processed)

| Job information | | Results | JSON | Execution details | | |
|-----------------|--------------------------------|--------------------------------------|------|-------------------|-------------|-----------------|
| Row | timestamp | ride_id | | meter_reading | ride_status | passenger_count |
| 1 | 2019-04-24 22:09:13.734480 UTC | 4bfc3d18-34c1-48db-ad93-1b9332cab8c3 | | 21.313406 | enroute | 1 |
| 2 | 2019-04-24 22:09:13.734130 UTC | 5a2099c2-7a9f-4d11-b8d4-9591990a95e0 | | 7.5937257 | enroute | 1 |
| 3 | 2019-04-24 22:09:13.734130 UTC | 1c276712-7fad-4cb9-b735-fddeed4df062 | | 5.270588 | enroute | 3 |
| 4 | 2019-04-24 22:09:13.733910 UTC | 13d7dd0f-1d81-4894-8f80-95c7d7f78a57 | | 2.452924 | enroute | 2 |
| 5 | 2019-04-24 22:09:13.733890 UTC | c50e32e4-29ba-48ea-a026-bd47790060ff | | 7.9254036 | enroute | 1 |
| 6 | 2019-04-24 22:09:13.509450 UTC | a0c29640-d76d-4f43-a5b5-ba95182fbbca | | 8.9503765 | enroute | 1 |
| 7 | 2019-04-24 22:09:13.509260 UTC | d305d865-84be-48b1-9aae-60618333c912 | | 19.628355 | enroute | 1 |
| 8 | 2019-04-24 22:09:13.509260 UTC | 77e41112-bf33-4f8d-8217-dcd885b00ce4 | | 19.70924 | enroute | 1 |
| 9 | 2019-04-24 22:09:13.509170 UTC | fb23b464-85e0-4e14-ad6f-10cea326b422 | | 0.078625955 | enroute | 1 |

Perform aggregations on the stream for reporting

1. 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
```

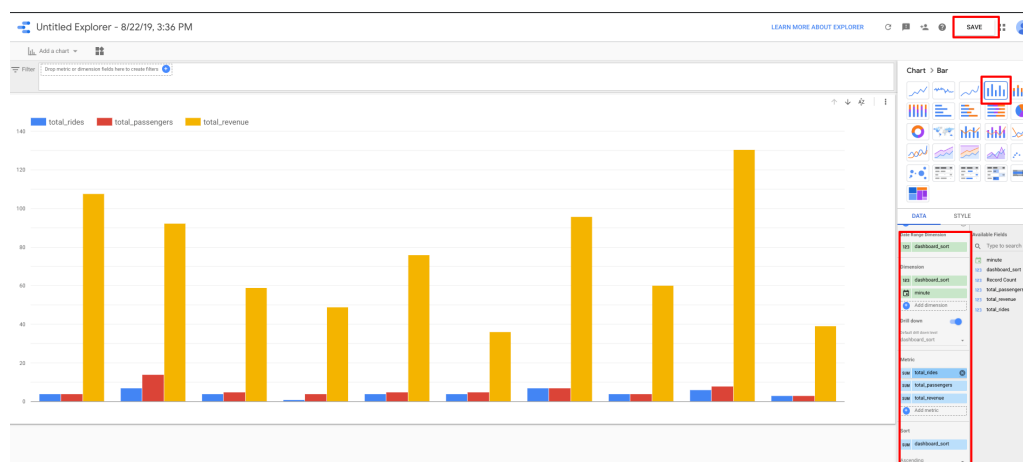
The result shows key metrics by the minute for every taxi drop-off

Create a Real-Time Dashboard

1. Click **Explore with Data Studio**

2. Specify the below settings:

- Chart type: column chart
- Date range dimension: dashboard_sort
- Dimension: dashboard_sort, minute
- Drill Down: dashboard_sort
- Metric: SUM() total_rides, SUM() total_passengers, SUM() total_revenue
- 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.

3. When you're happy with your dashboard, click Save to save this data source

4. 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

Stop the Cloud Dataflow job

1. Navigate back to Cloud Dataflow

2. Click the **streaming-taxi-pipeline**

3. Click **Stop Job** and **Cancel** pipeline

This will free up resources for your project

Congratulations!

In this lab you Pub/Sub to collect streaming data messages from Taxis and feed it through your Dataflow pipeline into BigQuery.

End your lab

When you have completed your lab, click **End Lab**. Qwiklabs removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.

Copyright 2019 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.