Creating a Streaming Data Pipeline for a Real-Time Dashboard with 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 build a streaming data pipeline to capture taxi revenue, passenger count, ride status, and much more, and then visualize the results in a management dashboard.

Objectives

In this lab you learn how to:

- Create a Dataflow job from a template
- Stream a Dataflow pipeline into BigQuery
- Monitor a Dataflow pipeline in BigQuery
- Analyze results with SQL
- Visualize key metrics in Looker Studio

Set up and requirements

Before you click the Start Lab button

Note: Read these instructions.

Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google 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 Google Cloud for the duration of the lab.

What you need

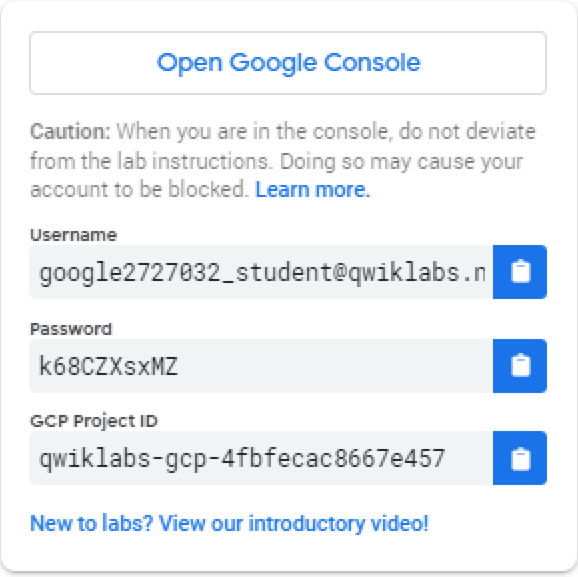
To complete this lab, you need:

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

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.**Note:** If you are using a Pixelbook, open an Incognito window to run this lab.

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.



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.

Username
google2727032_student@qwiklabs.n

Password
k68CZXsxMZ

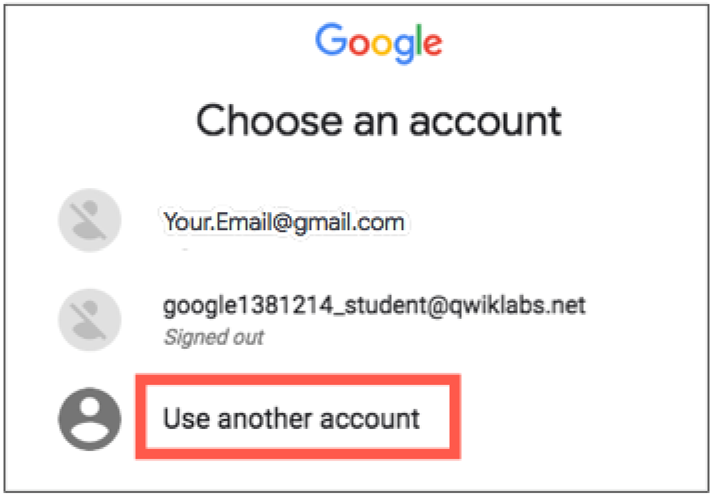
GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

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.

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

3. On the Choose an account page, click **Use Another Account**. The Sign in page opens.



Choose an account

Your.Email@gmail.com

google1381214_student@qwiklabs.net
Signed out

4. Paste the username that you copied from the Connection Details panel. Then copy and paste the password.

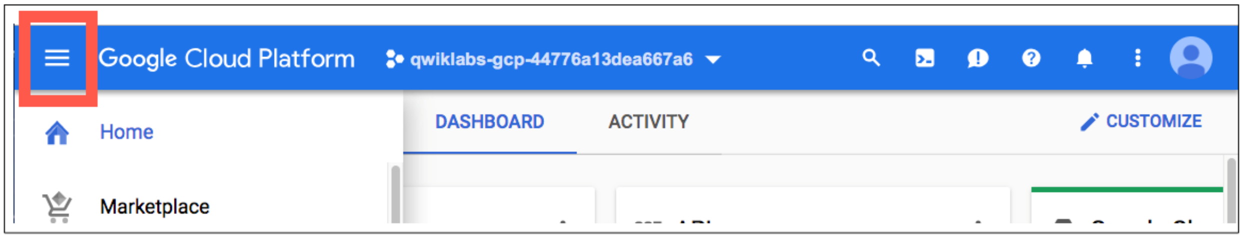
Note: You must use the credentials from the Connection Details panel. Do not use your Google Cloud Skills Boost credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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 Cloud console opens in this tab.

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

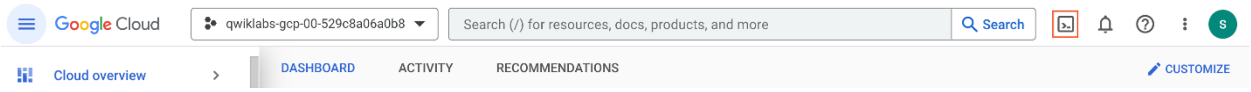


Activate Google Cloud Shell

Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud.

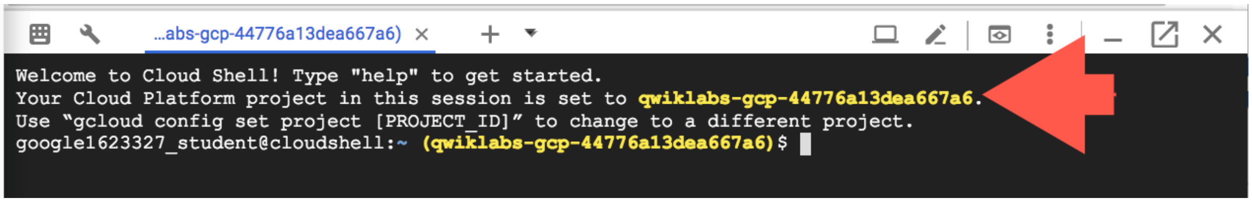
Google Cloud Shell provides command-line access to your Google Cloud resources.

1. In Cloud console, on the top right toolbar, click the Open Cloud Shell button.


Google Cloud
qwiklabs-gcp-00-529c8a06a0b8
Search (/) for resources, docs, products, and more
Search
S
Cloud overview
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE

2. Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

- You can list the active account name with this command:

```
gcloud auth list
```

Output:

```
Credentialed accounts:
 - @.com (active)
```

Example output:

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

- You can list the project ID with this command:

```
gcloud config list project
```

Output:

```
[core]
project =
```

Example output:

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

Task 1. Create a BigQuery dataset

In this task, you create the `taxirides` dataset. You have two different options which you can use to create this, using the Google Cloud Shell or the Google Cloud Console.

In this lab you will be using an extract of the NYC Taxi & Limousine Commission's open dataset. A small, comma-separated, datafile will be used to simulate periodic updates of taix data.
BigQuery is a serverless data warehouse. Tables in BigQuery are organized into datasets. In this lab, taxi data will flow from the standalone file via Dataflow to be stored in BigQuery. With this configuration, any new datafile deposited into the source Cloud Storage bucket would automatically be processed for loading.
Use one of the following options to create a new BigQuery dataset:

Option 1: The command-line tool

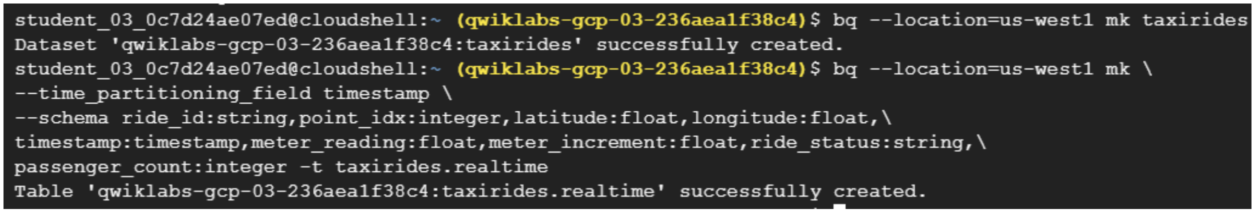
1. In **Cloud Shell** (>_), run the following command to create the `taxirides` dataset.

```
bq --location=Lab GCP Region mk taxirides
```

2. Run this command to create the `taxirides.realtime` table (empty schema that you will stream into later).

```
bq --location=Lab GCP Region 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 taxiridesrealtime
```

output:



```
student_03_0c7d24ae07ed@cloudshell:~ (qwiklabs-gcp-03-236aea1f38c4)$ bq --location=us-west1 mk taxirides
Dataset 'qwiklabs-gcp-03-236aea1f38c4:taxirides' successfully created.
student_03_0c7d24ae07ed@cloudshell:~ (qwiklabs-gcp-03-236aea1f38c4)$ bq --location=us-west1 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
Table 'qwiklabs-gcp-03-236aea1f38c4:taxirides.realtime' successfully created.
```

Option 2: The BigQuery Console UI

Note: Skip these steps if you created the tables using the command line.

1. In the Google Cloud console, in the **Navigation menu**(☰), click **BigQuery**.

2. If you see the Welcome dialog, click **Done.**

3. Click on **View actions** (⋮) next to your Project ID, and then click **Create dataset**.

4. In Dataset ID, type **taxirides**.

5. In Data location, select:

```
Lab GCP Region
```
Eg: us-west1

then click **Create Dataset**.

6. In the Explorer pane, click **expand node** (▸) to reveal the new taxirides dataset.

7. Click on **View actions** (⋮) next to the **taxirides** dataset, and then click **Open**.

8. Click **Create Table**.

9. In Table, type **realtime**

10. For the schema, click **Edit as text** and paste in the following:

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

11. In **Partition and cluster settings**, select **timestamp**.

12. Click **Create Table**.

Task 2. Copy required lab artifacts

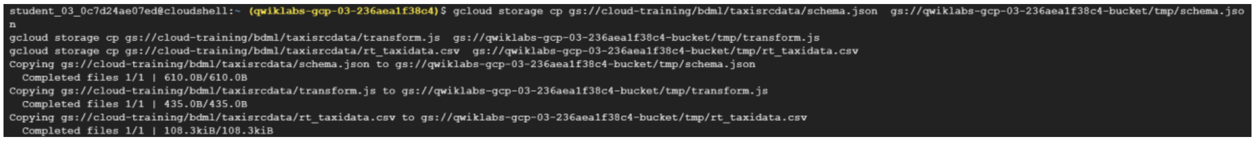
In this task, you move the required files to your Project.

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.
A Cloud Storage bucket was created for you during lab start up.

1. In **Cloud Shell** (>_), run the following commands to move files needed for the Dataflow job.

```
gcloud storage cp gs://cloud-training/bdml/taxisrcdata/schema.json
gs://Project_ID-bucket/tmp/schema.json
gcloud storage cp gs://cloud-training/bdml/taxisrcdata/transform.js
gs://Project_ID-bucket/tmp/transform.js
gcloud storage cp gs://cloud-training/bdml/taxisrcdata/rt_taxidata.csv
gs://Project_ID-bucket/tmp/rt_taxidata.csv
```

output:



```
student_03_0c7d24ae07ed@cloudshell:~ (qwiklabs-gcp-03-236aea1f38c4)$ gcloud storage cp gs://cloud-training/bdml/taxisrcdata/schema.json  gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/schema.jso
n
gcloud storage cp gs://cloud-training/bdml/taxisrcdata/transform.js  gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/transform.js
gcloud storage cp gs://cloud-training/bdml/taxisrcdata/rt_taxidata.csv  gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/rt_taxidata.csv
Copying gs://cloud-training/bdml/taxisrcdata/schema.json to gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/schema.json
  Completed files 1/1 | 610.0B/610.0B
Copying gs://cloud-training/bdml/taxisrcdata/transform.js to gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/transform.js
  Completed files 1/1 | 435.0B/435.0B
Copying gs://cloud-training/bdml/taxisrcdata/rt_taxidata.csv to gs://qwiklabs-gcp-03-236aea1f38c4-bucket/tmp/rt_taxidata.csv
  Completed files 1/1 | 108.3kiB/108.3kiB
```

Task 3. Set up a Dataflow Pipeline

In this task, you set up a streaming data pipeline to read files from your Cloud Storage bucket and write data to BigQuery.

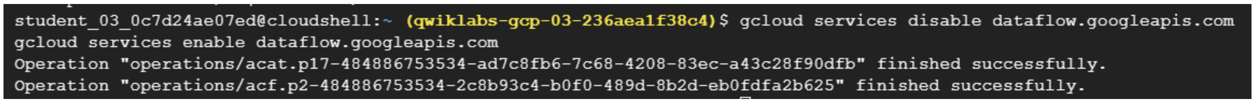
Dataflow is a serverless way to carry out data analysis.

Restart the connection to the Dataflow API.

1. In the Cloud Shell, run the following commands to ensure that the Dataflow API is enabled cleanly in your project.

```
gcloud services disable dataflow.googleapis.com
gcloud services enable dataflow.googleapis.com
```

output:



```
student_03_0c7d24ae07ed@cloudshell:~ (qwiklabs-gcp-03-236aea1f38c4)$ gcloud services disable dataflow.googleapis.com
gcloud services enable dataflow.googleapis.com
Operation "operations/acat.p17-484886753534-ad7c8fb6-7c68-4208-83ec-a43c28f90dfb" finished successfully.
Operation "operations/acf.p2-484886753534-2c8b93c4-b0f0-489d-8b2d-eb0fdfa2b625" finished successfully.
```

Create a new streaming pipeline:

1. In the Cloud console, in the **Navigation menu** (☰), click **Dataflow**.

2. In the top menu bar, click **Create Job From Template**.

3. Type **streaming-taxi-pipeline** as the Job name for your Dataflow job.

4. In **Regional endpoint**, select

```
Lab GCP Region
```

5. In **Dataflow template**, select the **Text Files on Cloud Storage to BigQuery** template.

6. In **Cloud Storage location of your JavaScript UDF**, paste or type:

```
Project_ID-bucket/tmp/transform.js
```

7. In **Cloud Storage location of your BigQuery schema file, described as a JSON**, paste or type:

```
Project_ID-bucket/tmp/schema.json
```

8. In **The name of the JavaScript function you wish to call as your UDF**, paste or type:

```
Transform
```

9. In **The fully qualified BigQuery table**, paste or type:

```
Project_ID:taxirides.realtime
```

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

10. In **The Cloud Storage location of the text you'd like to process**, paste or type:

```
gs://Project_ID-bucket/tmp/rt_taxidata.csv
```

11. In **Temporary directory for BigQuery loading process**, paste or type:

```
Project_ID-bucket/tmp
```

12. In **Temporary location**, paste or type:

```
Project_ID-bucket/tmp
```

13. Click **Optional Parameters**.

14. In **Max workers**, type **2**

15. In **Number of workers**, type **1**

16. Uncheck **Use default machine type**.

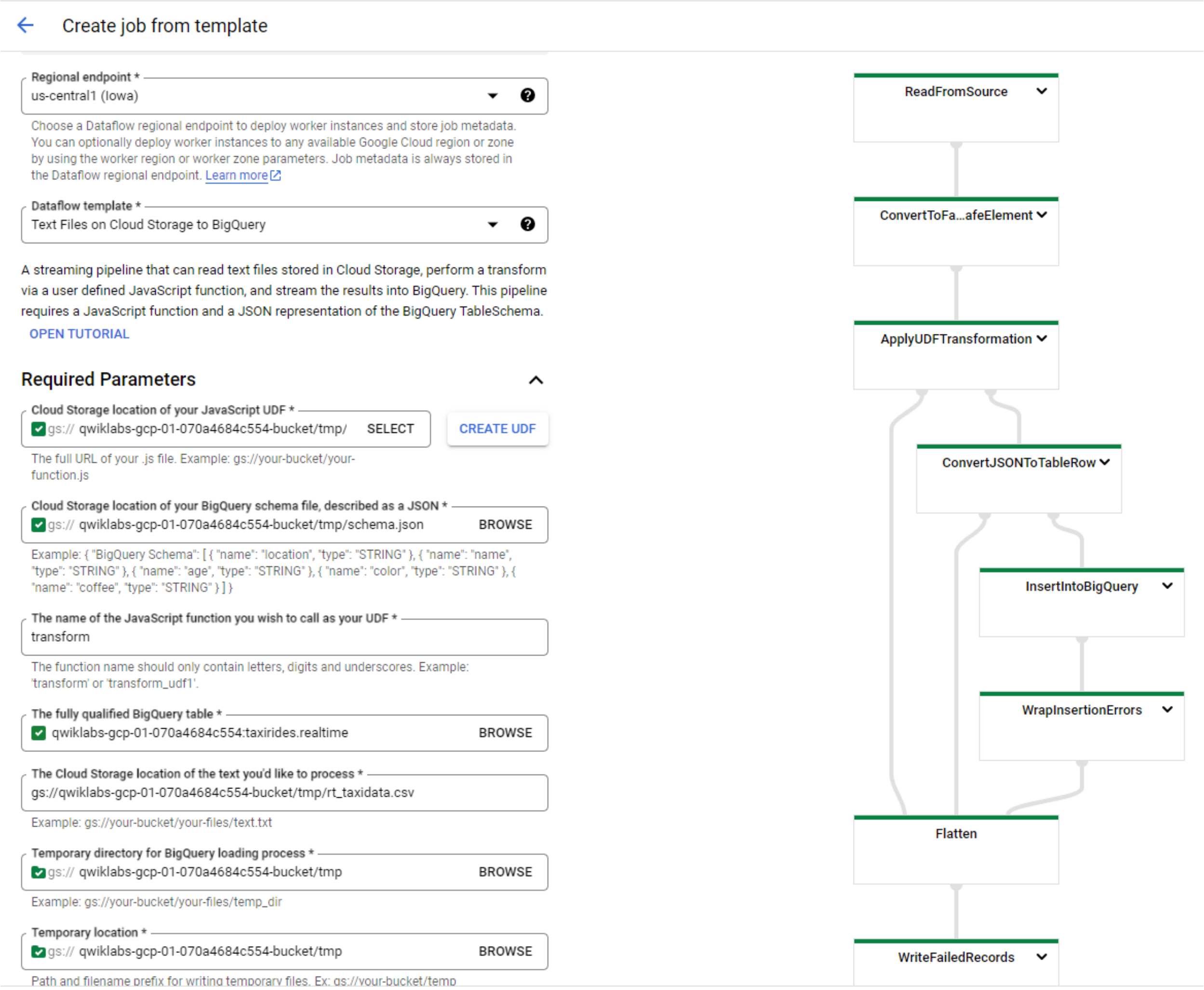
17. Under **General purpose**, choose the following:

Series: **E2**
Machine type: **e2-medium (2 vCPU, 4 GB memory)**

18. Click **Run Job**.

A new streaming job has started! You can now see a visual representation of the data pipeline. It will take 3 to 5 minutes for data to begin moving into BigQuery.

Note: If the dataflow job fails for the first time then re-create a new job template with new job name and run the job.



Create job from template

Regional endpoint *
us-central1 (Iowa)

Choose a Dataflow regional endpoint to deploy worker instances and store job metadata. You can optionally deploy worker instances to any available Google Cloud region or zone by using the worker region or worker zone parameters. Job metadata is always stored in the Dataflow regional endpoint. Learn more

Dataflow template *
Text Files on Cloud Storage to BigQuery

A streaming pipeline that can read text files stored in Cloud Storage, perform a transform via a user defined JavaScript function, and stream the results into BigQuery. This pipeline requires a JavaScript function and a JSON representation of the BigQuery TableSchema.

OPEN TUTORIAL

Required Parameters

Cloud Storage location of your JavaScript UDF *
☑ gs:// qwiklabs-gcp-01-070a4684c554-bucket/tmp/ SELECT CREATE UDF

The full URL of your .js file. Example: gs://your-bucket/your-function.js

Cloud Storage location of your BigQuery schema file, described as a JSON *
☑ gs:// qwiklabs-gcp-01-070a4684c554-bucket/tmp/schema.json BROWSE

Example: { "BigQuery Schema": [{ "name": "location", "type": "STRING" }, { "name": "name", "type": "STRING" }, { "name": "age", "type": "STRING" }, { "name": "color", "type": "STRING" }, { "name": "coffee", "type": "STRING" }] }

The name of the JavaScript function you wish to call as your UDF *
transform

The function name should only contain letters, digits and underscores. Example: 'transform' or 'transform_udf1'.

The fully qualified BigQuery table *
☑ qwiklabs-gcp-01-070a4684c554:taxirides.realtime BROWSE

The Cloud Storage location of the text you'd like to process *
gs://qwiklabs-gcp-01-070a4684c554-bucket/tmp/rt_taxidata.csv

Example: gs://your-bucket/your-files/text.txt

Temporary directory for BigQuery loading process *
☑ gs:// qwiklabs-gcp-01-070a4684c554-bucket/tmp BROWSE

Example: gs://your-bucket/your-files/temp_dir

Temporary location *
☑ gs:// qwiklabs-gcp-01-070a4684c554-bucket/tmp BROWSE

Path and filename prefix for writing temporary files. Ex: gs://your-bucket/temp

Task 4. Analyze the taxi data using BigQuery

In this task, you analyze the data as it is streaming.

1. In the Cloud console, in the **Navigation menu** (☰), click **BigQuery**.

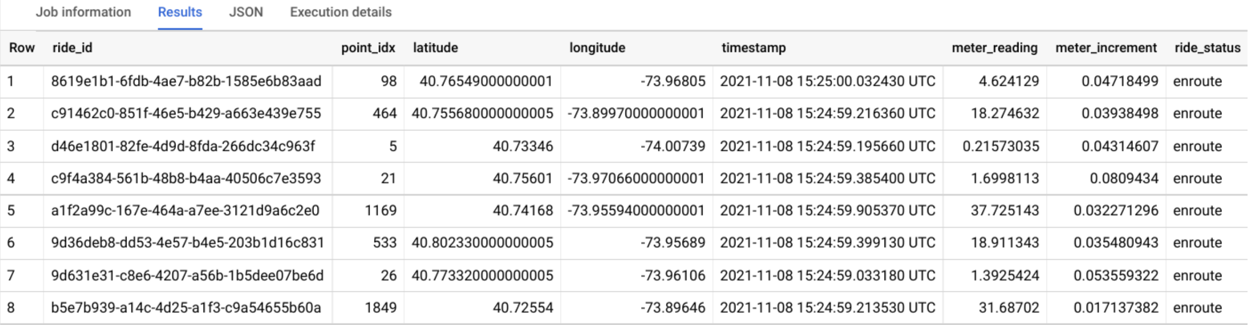
2. If the Welcome dialog appears, click **Done**.

3. In the Query Editor, type the following, and then click **Run**:

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

Note: If no records are returned, wait another minute and re-run the above query (Dataflow takes 3-5 minutes to setup the stream).

Your output will look similar to the following:



Row	ride_id	point_idx	latitude	longitude	timestamp	meter_reading	meter_increment	ride_status
1	8619e1b1-6fdb-4ae7-b82b-1585e6b83aad	98	40.76549000000001	-73.96805	2021-11-08 15:25:00.032430 UTC	4.624129	0.04718499	enroute
2	c91462c0-851f-46e5-b429-a663e439e755	464	40.755680000000005	-73.89970000000001	2021-11-08 15:24:59.216360 UTC	18.274632	0.03938498	enroute
3	d46e1801-82fe-4d9d-8fda-266dc34c963f	5	40.73346	-74.00739	2021-11-08 15:24:59.195660 UTC	0.21573035	0.04314607	enroute
4	c9f4a384-561b-48b8-b4aa-40506c7e3593	21	40.75601	-73.97066000000001	2021-11-08 15:24:59.385400 UTC	1.6998113	0.0809434	enroute
5	a1f2a99c-167e-464a-a7ee-3121d9a6c2e0	1169	40.74168	-73.95594000000001	2021-11-08 15:24:59.905370 UTC	37.725143	0.032271296	enroute
6	9d36deb8-dd53-4e57-b4e5-203b1d16c831	533	40.802330000000005	-73.95689	2021-11-08 15:24:59.399130 UTC	18.911343	0.035480943	enroute
7	9d631e31-c8e6-4207-a56b-1b5dee07be6d	26	40.773320000000005	-73.96106	2021-11-08 15:24:59.033180 UTC	1.3925424	0.053559322	enroute
8	b5e7b939-a14c-4d25-a1f3-c9a54655b60a	1849	40.72554	-73.89646	2021-11-08 15:24:59.213530 UTC	31.68702	0.017137382	enroute

Task 5. Perform aggregations on the stream for reporting

In this task, you calculate aggregations on the stream for reporting.

1. In the **Query Editor**, clear the current query.

2. Copy and paste the following query, and then click **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
ORDER BY timestamp DESC
LIMIT 1000

)

# 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
```

Note: Ensure Dataflow is registering data in BigQuery before proceeding to the next task.

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

3. Click **Save > Save query**.

4. In the Save query dialog, in the **Name** field, type **My Saved Query**.

5. Click **Save**.

Task 6. Stop the Dataflow Job

In this task, you stop the Dataflow job to free up resources for your project.

1. In the Cloud console, in the **Navigation menu** (☰), click **Dataflow**.

2. Click the **streaming-taxi-pipeline**, or the new job name.

3. Click **Stop**, and then select **Cancel > Stop Job**.

Task 7. Create a real-time dashboard

In this task, you create a real-time dashboard to visualize the data.

1. In the Cloud console, in the **Navigation menu** (☰), click **BigQuery**.

2. In the Explorer Pane, expand your **Project ID**.

3. Expand **Saved queries**, and then click **My Saved Query**.

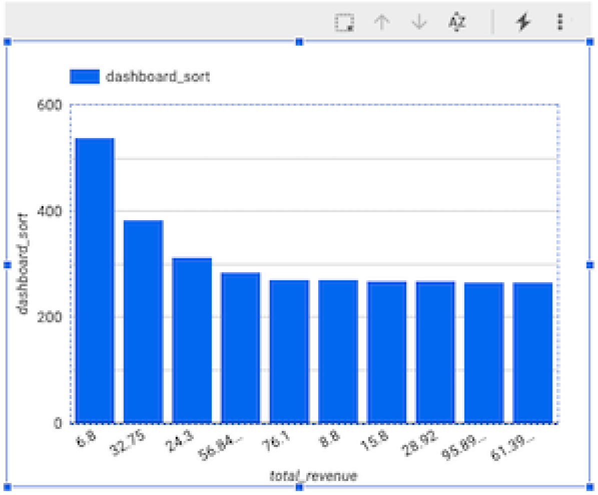
Your query is loaded in to the query editor.

4. Click **Run.**

5. In BigQuery, click **Explore Data > Explore with Looker Studio**.

   Looker Studio Opens. Click **Get started**.

6. In the Looker Studio window, click your bar chart.



(

The Chart pane appears.

7. Click **Add a chart**, and then select **Combo chart**.

8. In the Setup pane, in Data Range Dimension, hover over **minute (Date)** and click **X** to remove it.

9. In the Data pane, click **dashboard_sort** and drag it to **Setup > Data Range Dimension > Add dimension.**

10. In **Setup > Dimension**, click **minute**, and then select **dashboard_sort**.

11. In **Setup > Metric**, click **dashboard_sort**, and then select **total_rides.**

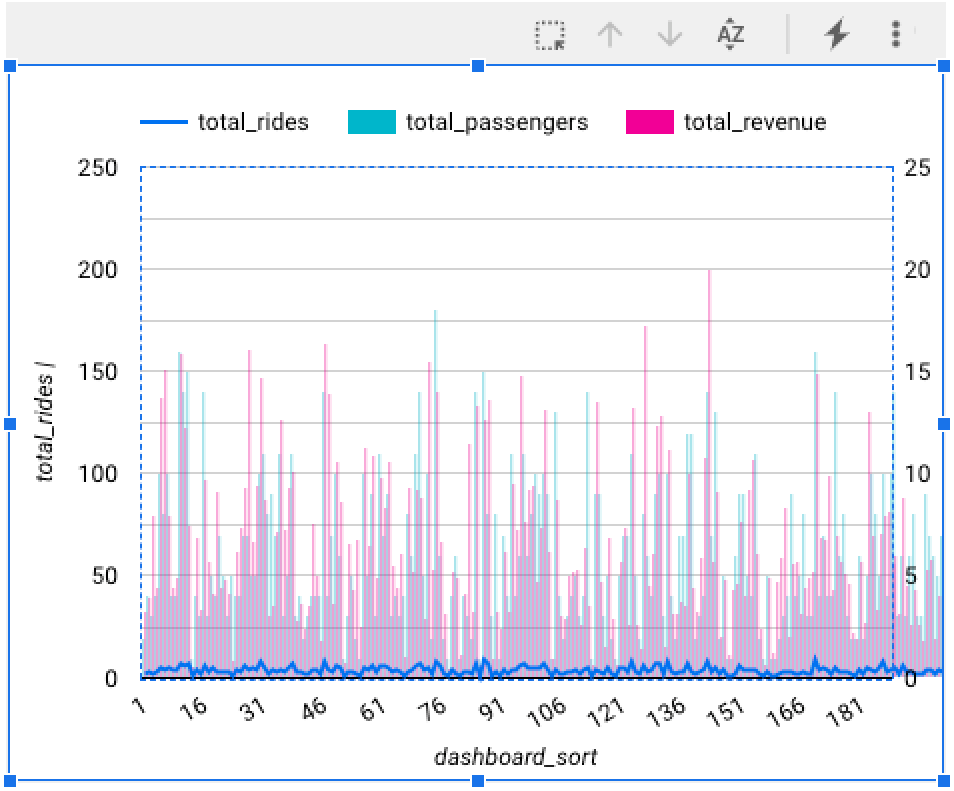
12. In **Setup > Metric**, click **Record Count**, and then select **total_passengers**.

13. In **Setup > Metric**, click **Add metric**, and then select **total_revenue**.

14. In **Setup > Sort**, click **total_rides**, and then select **dashboard_sort**.

15. In **Setup > Sort**, click **Ascending**.

Your chart should look similar to this:



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

16. When you're happy with your dashboard, click **Save and share** to save this data source.

17. If prompted to complete your account setup, agree to the terms and conditions, and then click **Continue**.

18. If prompted which updates you want to receive, answer **no** to all, then click **Continue**.

19. If prompted with the **Review data access before saving** window, click **Acknowledge and save**.

20. If prompted to choose an account select your **Student Account**.

21. Click **Add to report**.

22. Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking **More options** (⋮), and then **Refresh data**.

Task 8. Create a time series dashboard

In this task, you create a time series chart.

1. Click this [Looker Studio link]() to open Looker Studio in a new browser tab.
2. On the **Reports** page, in the **Start with a Template** section, click the **[+] Blank Report** template.

3. A new, empty report opens with the **Add data to report** window.

4. From the list of **Google Connectors**, select the **BigQuery** tile.

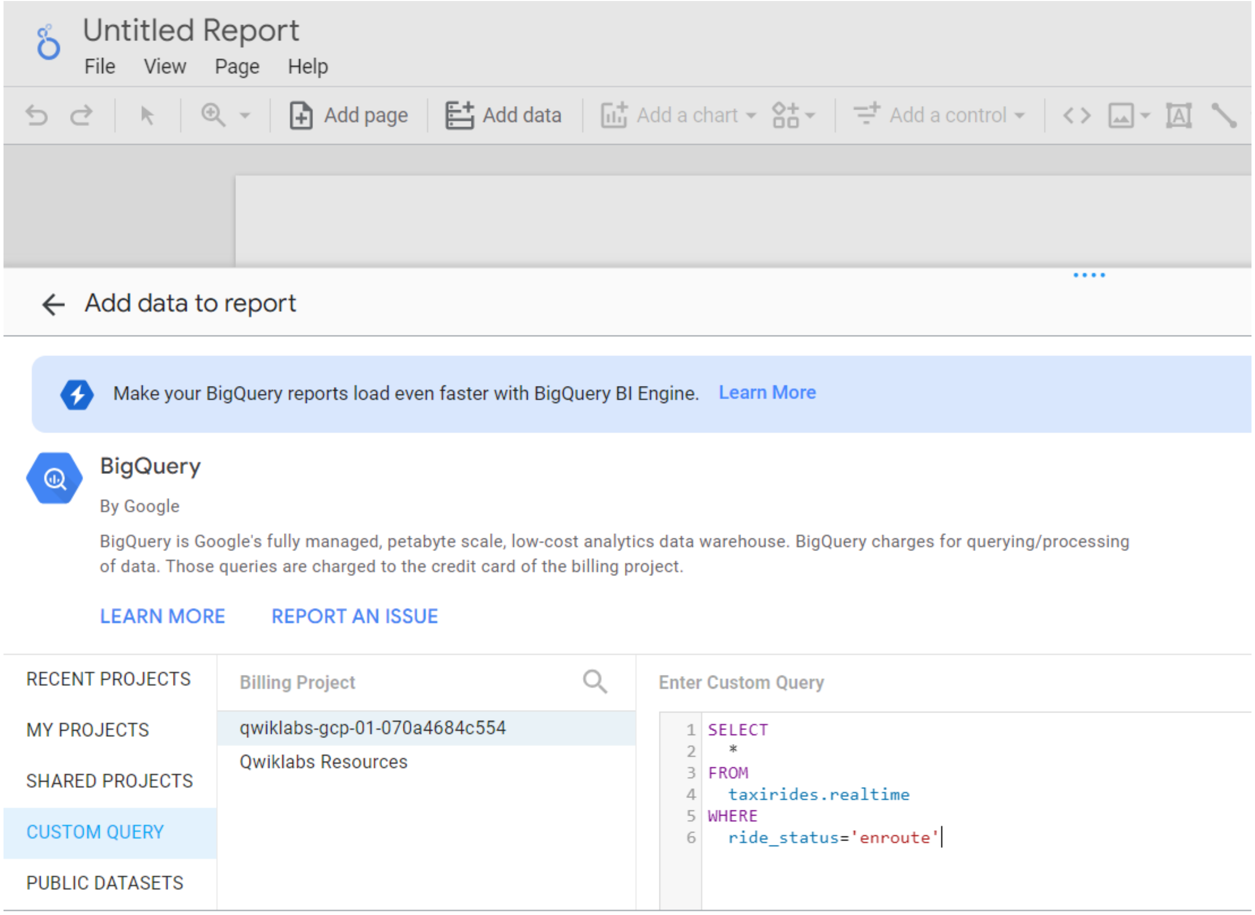
5. Click **Custom Query**, and then select your ProjectID. This should appear in the following format, **qwiklabs-gcp-xxxxxxx**.

6. In Enter Custom Query, paste the following query:

```
SELECT
  *
FROM
  taxirides.realtime
WHERE
  ride_status='enroute'
```

7. Click **Add > Add To Report**.

   A new untitled report appears. It may take up to a minute for the screen to finish refreshing.



Make your BigQuery reports load even faster with BigQuery BI Engine.

BigQuery

By Google

BigQuery is Google's fully managed, petabyte scale, low-cost analytics data warehouse. BigQuery charges for querying/processing of data. Those queries are charged to the credit card of the billing project.

qwiklabs-gcp-01-070a4684c554

Qwiklabs Resources

```
SELECT
  *
FROM
  taxirides.realtime
WHERE
  ride_status='enroute'
```

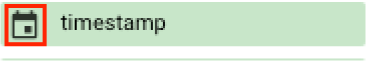
Create a time series chart

1. In the **Data** pane, click **Add a Field**.
2. Click **All Fields** on the left corner.
3. Change the **timestamp** field type to **Date & Time > Date Hour Minute (YYYYMMDDhhmm)**.
4. In the change timestamp dialog, click **Continue**, and then click **Done**.
5. In the top menu, click **Add a chart**.

6. Choose **Time series chart.**

7. Position the chart in the bottom left corner - in the blank space.

8. In **Setup > Dimension**, click **timestamp (Date)**, and then select **timestamp**.

9. In **Setup > Dimension**, click **timestamp**, and then select **calendar.** 

   timestamp

10. In **Type**, select **Date & Time** > **Date Hour Minute**.

11. Click outside the dialog to close it. You do not need to add a name.

12. In **Setup > Metric**, click **Record Count**, and then select **meter reading**.

Congratulations!

In this lab, you used Dataflow to stream data through a pipeline into BigQuery.

End your lab

When you have completed your lab, click **End Lab**. Google Cloud Skills Boost 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**.