End Lab 00:23:33

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.

Open Google Console

GCP Username

student-02-20b774a60c91@

GCP Password

AMGH0XLHVrDc

Creating a Streaming Data Pipeline for a Real-Time Dashboard with Dataflow

1 hour Free ★★★★

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.

Set up your environments

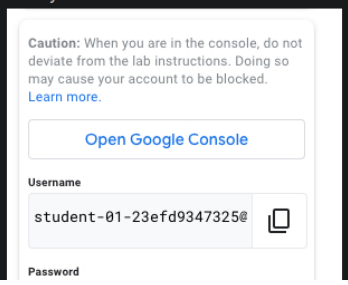
Qwiklabs setup

For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

1. Make sure you signed into Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, 02:00:00 and make sure you can finish in that time block.

> There is no pause feature. You can restart if needed, but you have to start at the beginning.

3. When ready, click START LAB.
4. Note your lab credentials. You will use them to sign in to Cloud Platform Console.



5. Click **Open Google Console**.

6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

> If you use other credentials, you'll get errors or **incur charges**.

7. Accept the terms and skip the recovery resource page.

> Do not click **End Lab** unless you are finished with the lab or want to restart it. This clears your work and removes the project.

Task 1. Create a Pub/Sub topic and BigQuery dataset

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. Pub/Sub delivers low-latency, durable messaging.

In 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.

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: The command-line tool

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 that you will stream into later).

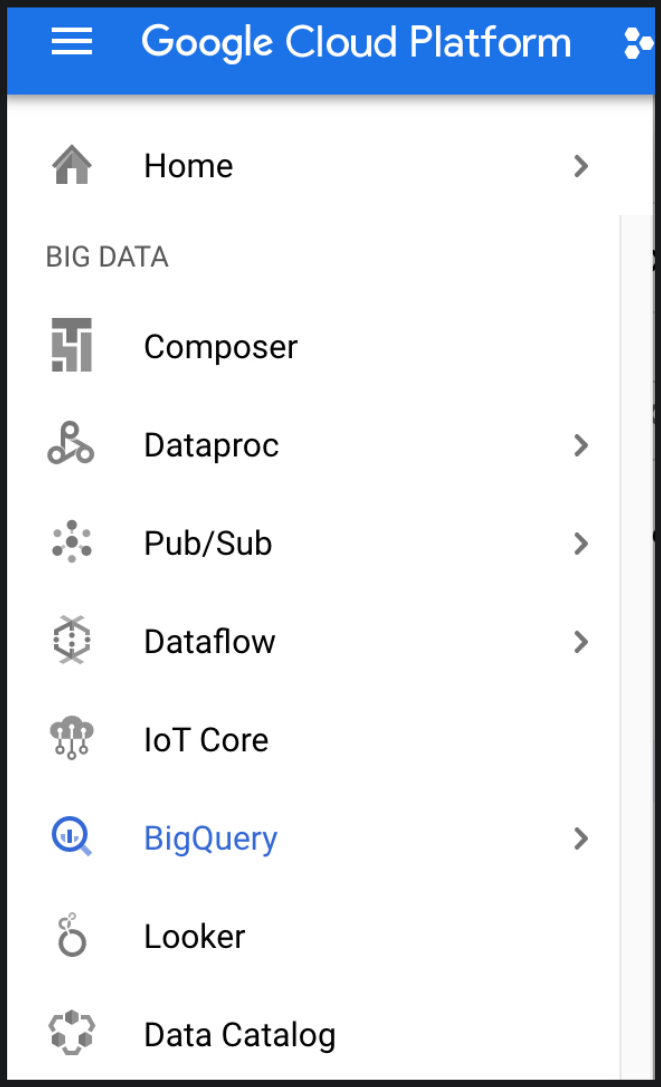
```
bq mk \
--time_partitioning_field timestamp \
--schema
ride_id:string,point_idx:integer,latitude:float,longitude:float,\
timestamp:timestamp,meter_reading:float,meter_increment:float,ride_
passenger_count:integer -t taxirides.realtime
```

Option 2: The BigQuery Console UI

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

Open BigQuery Console

In the Google Cloud Console, select **Navigation menu** > **Big Data** > **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**.

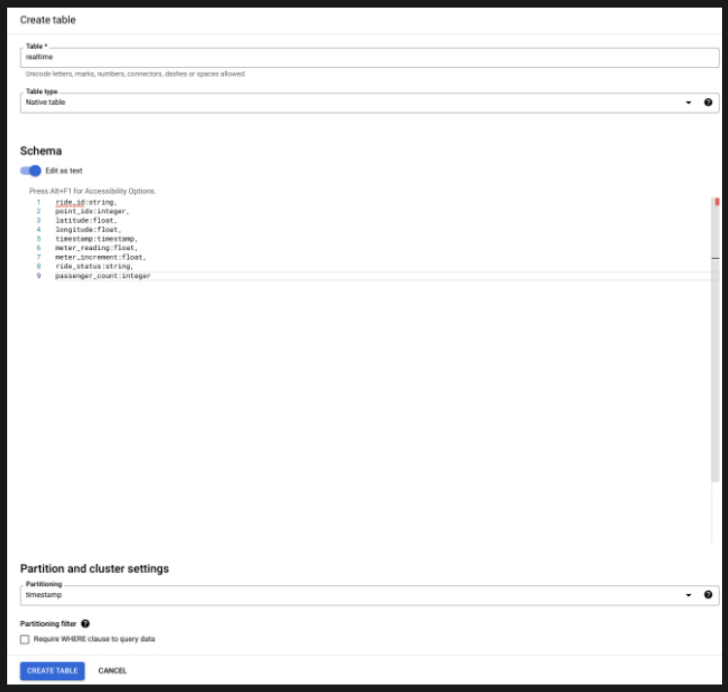
1. Click on the **View actions** icon next to your Project ID and click **Create dataset**.
2. Set the **Dataset ID** as **taxirides**, leave all the other fields the way they are, and click **CREATE DATASET**.
3. If you look at the left-hand resources menu, you should see your newly created dataset.
4. Click on the **View actions** icon next to the **taxirides** dataset and click **Open**.
5. Click **CREATE TABLE**.
6. Name the table **realtime**
7. For the 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
```

8. Under **Partition and cluster settings**, select the **timestamp** option for the Partitioning field.

9. Confirm against the below screenshot:



10. Click the **CREATE TABLE** button.

Task 2. 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, you use Cloud Storage to provide working space for your Dataflow pipeline.

1. In the Cloud Console, go to **Navigation menu** > **Cloud Storage**.

2. Click **CREATE BUCKET**.

3. For **Name**, paste in your **GCP Project ID** and then click **Continue**.

4. For **Location type**, click **Multi-region** if it is not already selected.

5. Click **CREATE**.

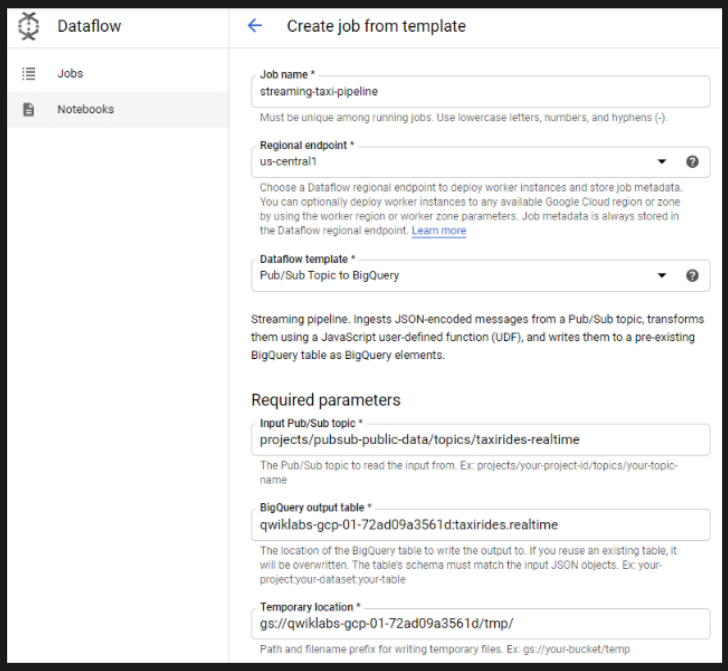
Task 3. Set up a Dataflow Pipeline

Dataflow is a serverless way to carry out data analysis. In this lab, you 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 Cloud 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 Dataflow job.
4. Under **Dataflow template**, select the **Pub/Sub Topic to BigQuery** template.
5. Under **Input Pub/Sub 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.

> **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.

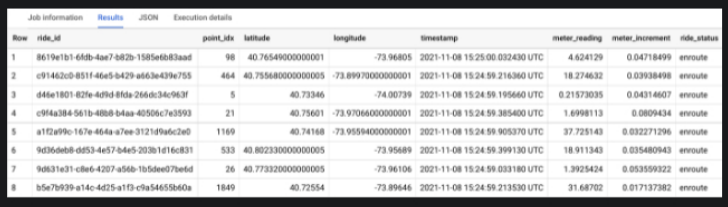
Task 4. Analyze the taxi data using BigQuery

To analyze the data as it is streaming:

1. In the Cloud Console, select **Navigation menu** > **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:



Job information | Results | JSON | Execution details

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

1. Copy and paste the below query and click **RUN**.

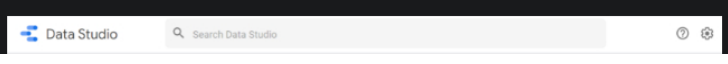
```
WITH streaming_data AS (
SELECT
  timestamp,
  TIMESTAMP_TRUNC(timestamp, HOUR, 'UTC') AS hour,
  TIMESTAMP_TRUNC(timestamp, MINUTE, 'UTC') AS minute,
  TIMESTAMP_TRUNC(timestamp, SECOND, 'UTC') AS second,
  ride_id,
  latitude,
  longitude,
  meter_reading,
  ride_status,
  passenger_count
FROM
  taxirides.realtime
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
```

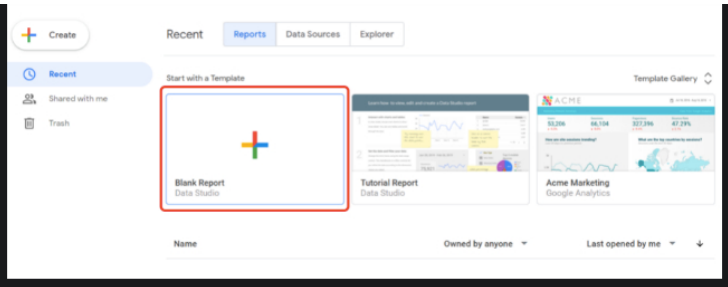
The result shows key metrics by the minute for every taxi drop-off.

Task 6. Create a real-time dashboard

1. Open this Google Data Studio link in a new incognito browser tab.

2. On the **Reports** page, in the **Start with a Template** section, click the **[+] Blank Report** template.





3. If prompted with the **Welcome to Google Studio** window, click **Get started**.

4. Check the checkbox to acknowledge the Google Data Studio Additional Terms, and click **Continue**.

5. Select **No** to all the questions, then click **Continue**.

6. Switch back to the **BigQuery** Console.

7. Click **EXPLORE DATA > Explore with Data Studio** in BigQuery page.

8. Click **GET STARTED**, then click **AUTHORIZE**.

9. Specify the below settings:

- **Chart type:** `Combo chart`
- **Date range Dimension:** `dashboard_sort`
- **Dimension:** `dashboard_sort`
- **Drill Down:** `dashboard_sort` (Make sure that Drill down option is turned `ON`)
- **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.

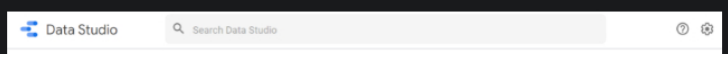
10. When you're happy with your dashboard, click **Save** to save this data source.

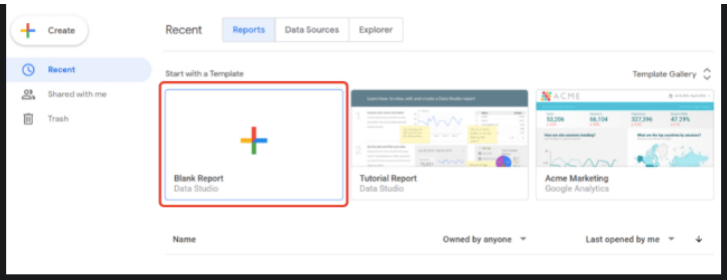
11. Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking on the Refresh button near the Save button.

Task 7. Create a time series dashboard

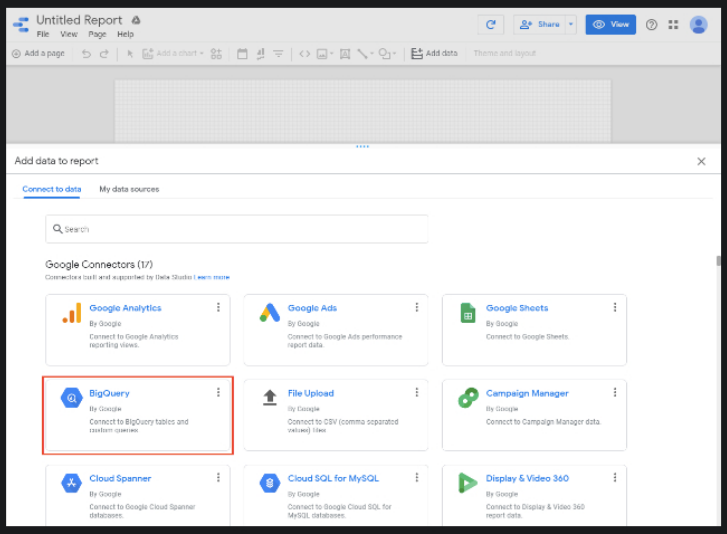
1. Click this Google Data Studio link to open Data Studio in a new browser tab.

2. On the **Reports** page, in the **Start with a Template** section, click the **[+] Blank Report** template.





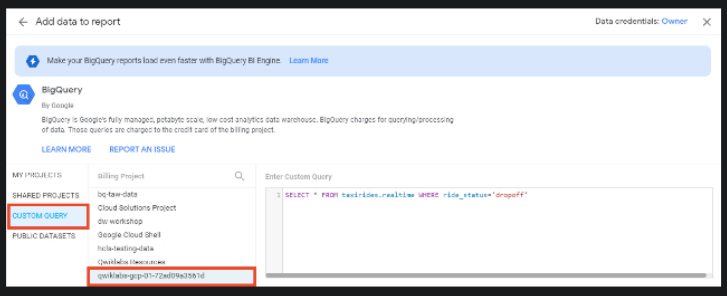
3. A new, empty report opens with **Add data to report**.



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

5. Under **CUSTOM QUERY**, click **qwiklabs-gcp-xxxxxxx** > **Enter Custom Query**, add the following query.

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



6. Click **Add > ADD TO REPORT**.

Create a time series chart

1. In the **Data** panel, scroll down to the bottom right and click **ADD A FIELD**. Click **All Fields** on the left corner.

2. Change the field **timestamp** type to **Date & Time > Date Hour Minute (YYYYMMDDhhmm)**.

3. Click **Continue** and then click **Done**.

4. Click **Add a chart**.

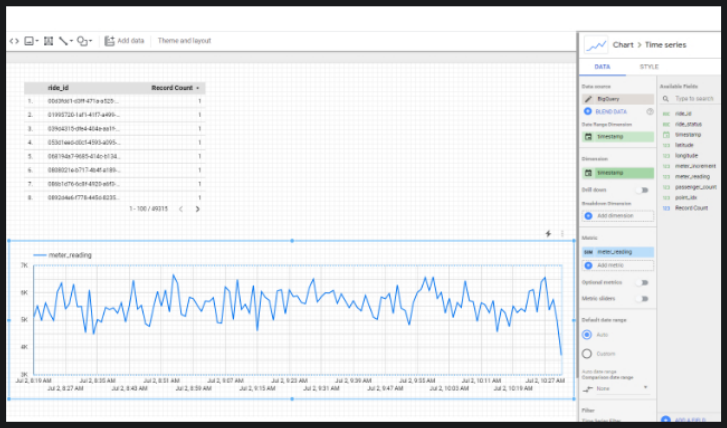
5. Choose **Time series chart**.

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

7. In the **Data** panel on the right, change the following:

- **Dimension:** `timestamp`
- **Metric:** `meter_reading(SUM)`

Your time series chart should look similar to this:



If Dimension is `timestamp(Date)`, then click on calendar icon next to `timestamp(Date)`, and select type to **Date & Time > Date Hour Minute**.

Task 8. Stop the Dataflow job

1. Navigate back to **Dataflow**.

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

3. Click **STOP** and select **Cancel** > **STOP JOB**.

This will free up resources for your project.

Congratulations!

In this lab you used 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.