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

Tuesday, October 27, 2020 9:28 AM

Clipped from: https://googlecourses.qwiklabs.com/course_sessions/39079/labs/48299

## 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

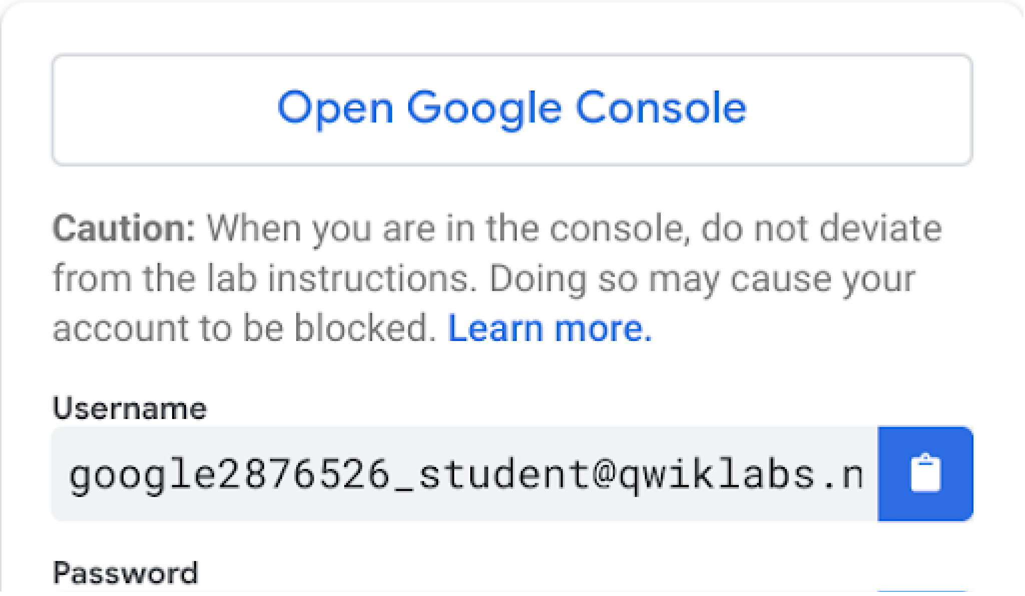
For each lab, you get a new GCP project and set of resources for a fixed time at no cost.

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.
3. When ready, click

.
4. Note your lab credentials. You will use them to sign in to Cloud Platform Console.

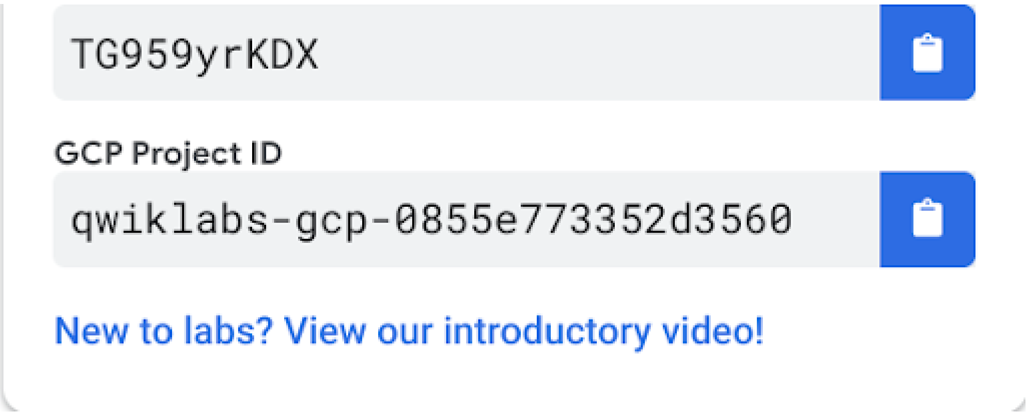


**Caution:** When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. Learn more.

Username

google2876526_student@qwiklabs.n

Password



TG959yrKDX

GCP Project ID

qwiklabs-gcp-0855e773352d3560

New to labs? View our introductory video!

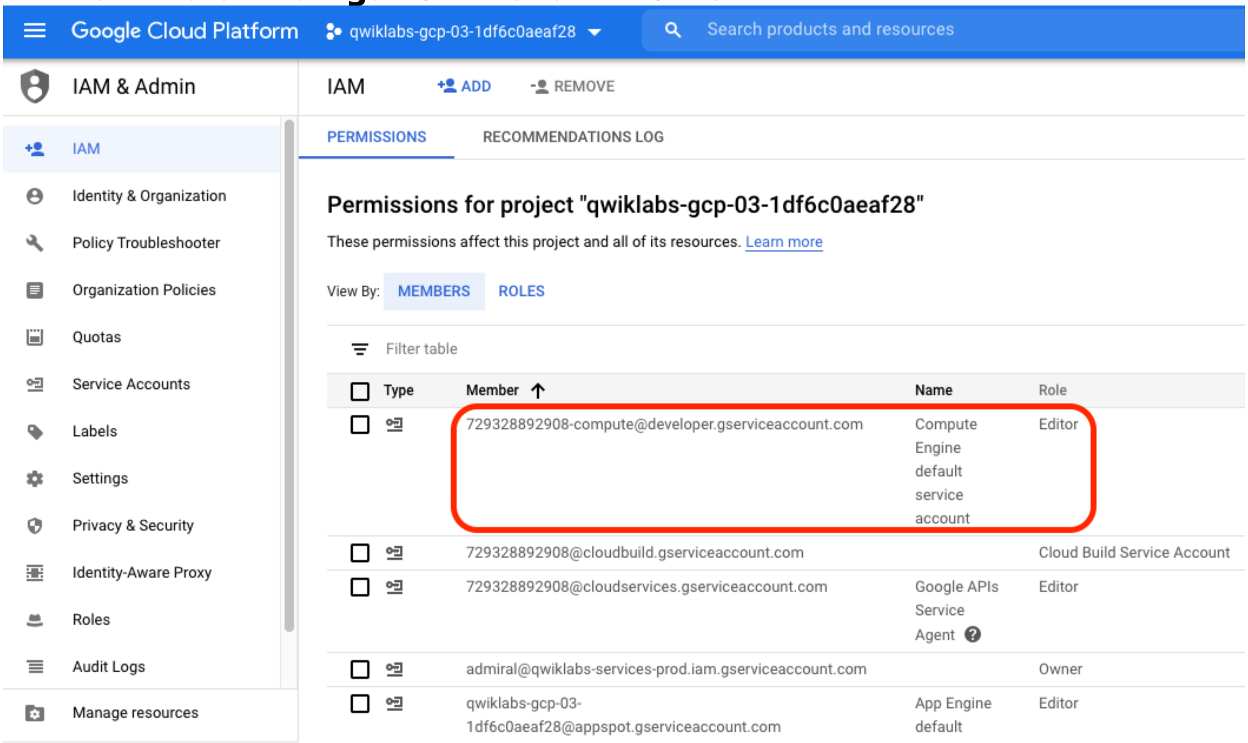
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.

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

### Check project permissions

Before you begin your work on Google Cloud, you need to ensure that your project has the correct permissions within Identity and Access Management (IAM).

1. In the Google Cloud console, on the **Navigation menu** ( ), click **IAM & Admin** > **IAM**.
2. Confirm that the default compute Service Account {project-number}-compute@developer.gserviceaccount.com is present and has the `editor` role assigned. The account prefix is the project number, which you can find on **Navigation menu** > **Home**.



Permissions for project "qwiklabs-gcp-03-1df6c0aeaf28"

These permissions affect this project and all of its resources. Learn more

View By: MEMBERS ROLES

Filter table

| ☐ Type | Member ↑ | Name | Role |
|---|---|---|---|
| ☐ | 729328892908-compute@developer.gserviceaccount.com | Compute Engine default service account | Editor |
| ☐ | 729328892908@cloudbuild.gserviceaccount.com | | Cloud Build Service Account |
| ☐ | 729328892908@cloudservices.gserviceaccount.com | Google APIs Service Agent | Editor |
| ☐ | admiral@qwiklabs-services-prod.iam.gserviceaccount.com | | Owner |
| ☐ | qwiklabs-gcp-03-1df6c0aeaf28@appspot.gserviceaccount.com | App Engine default | Editor |

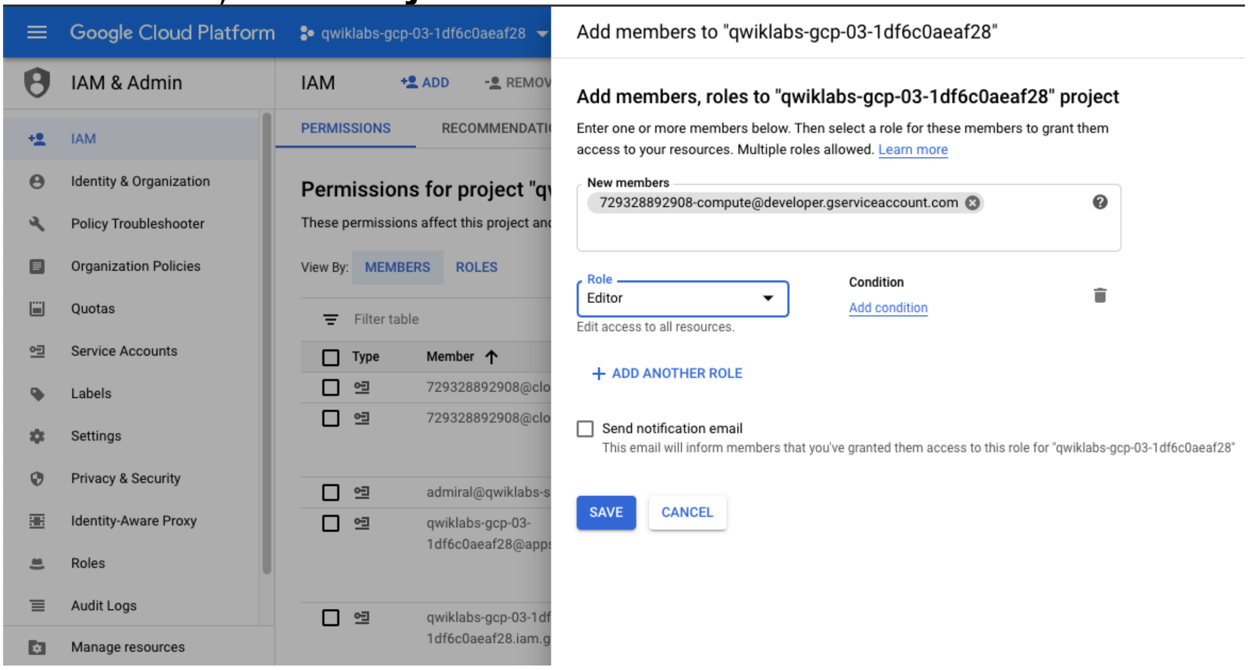
If the account is not present in IAM or does not have the `editor` role, follow the steps below to assign the required role.

- In the Google Cloud console, on the **Navigation menu**, click **Home**.
- Copy the project number (e.g. `729328892908`).
- On the **Navigation menu**, click **IAM & Admin** > **IAM**.
- At the top of the **IAM** page, click **Add**.
- For **New members**, type:

`{project-number}-compute@developer.gserviceaccount.com`

Replace `{project-number}` with your project number.

- For **Role**, select **Project** > **Editor**. Click **Save**.



Add members to "qwiklabs-gcp-03-1df6c0aeaf28"

Add members, roles to "qwiklabs-gcp-03-1df6c0aeaf28" project

Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. Learn more

New members: 729328892908-compute@developer.gserviceaccount.com

Role: Editor

Edit access to all resources.

Condition: Add condition

+ ADD ANOTHER ROLE

□ Send notification email

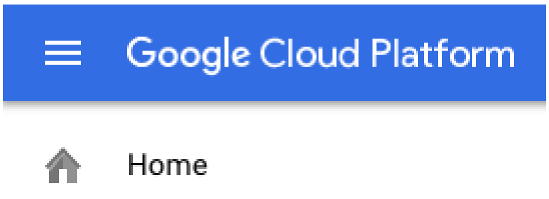
This email will inform members that you've granted them access to this role for "qwiklabs-gcp-03-1df6c0aeaf28"

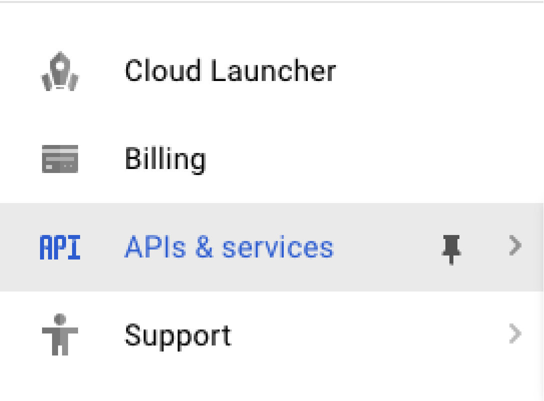
SAVE CANCEL

## Note your project name; confirm that needed APIs are enabled

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

1. In the Cloud 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 Cloud Console, in the Navigation menu, click **APIs & services**.





1. Scroll down in the list of enabled APIs, and confirm that these APIs are enabled:

   - **Cloud Pub/Sub API**
   - **Dataflow API**

2. 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.

## 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
```

1. 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_in
crement:float,ride_status:string,\
passenger_count:integer -t taxirides.realtime
```

### Option 2: The BigQuery Console UI

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

1. In the Cloud Console, go to **Navigation menu** > **BigQuery**.
2. Once there, click on your Project ID from the left-hand menu.
3. Now on the right-hand side of the Cloud 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 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
```

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

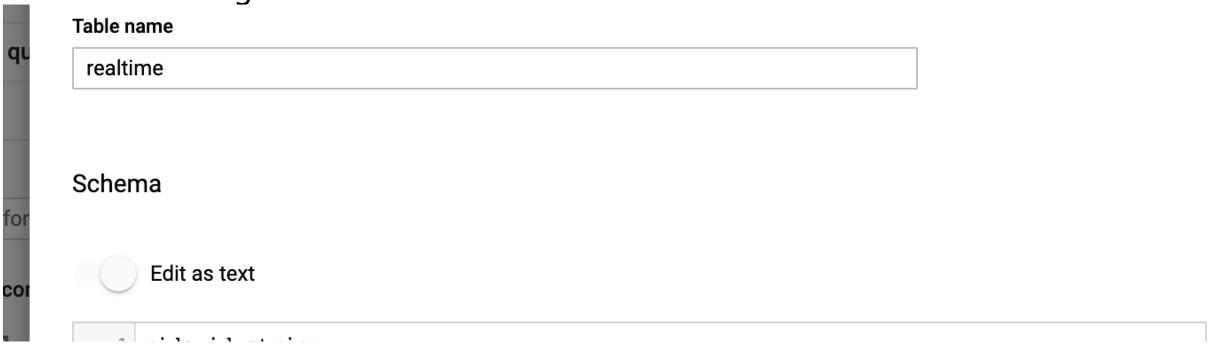
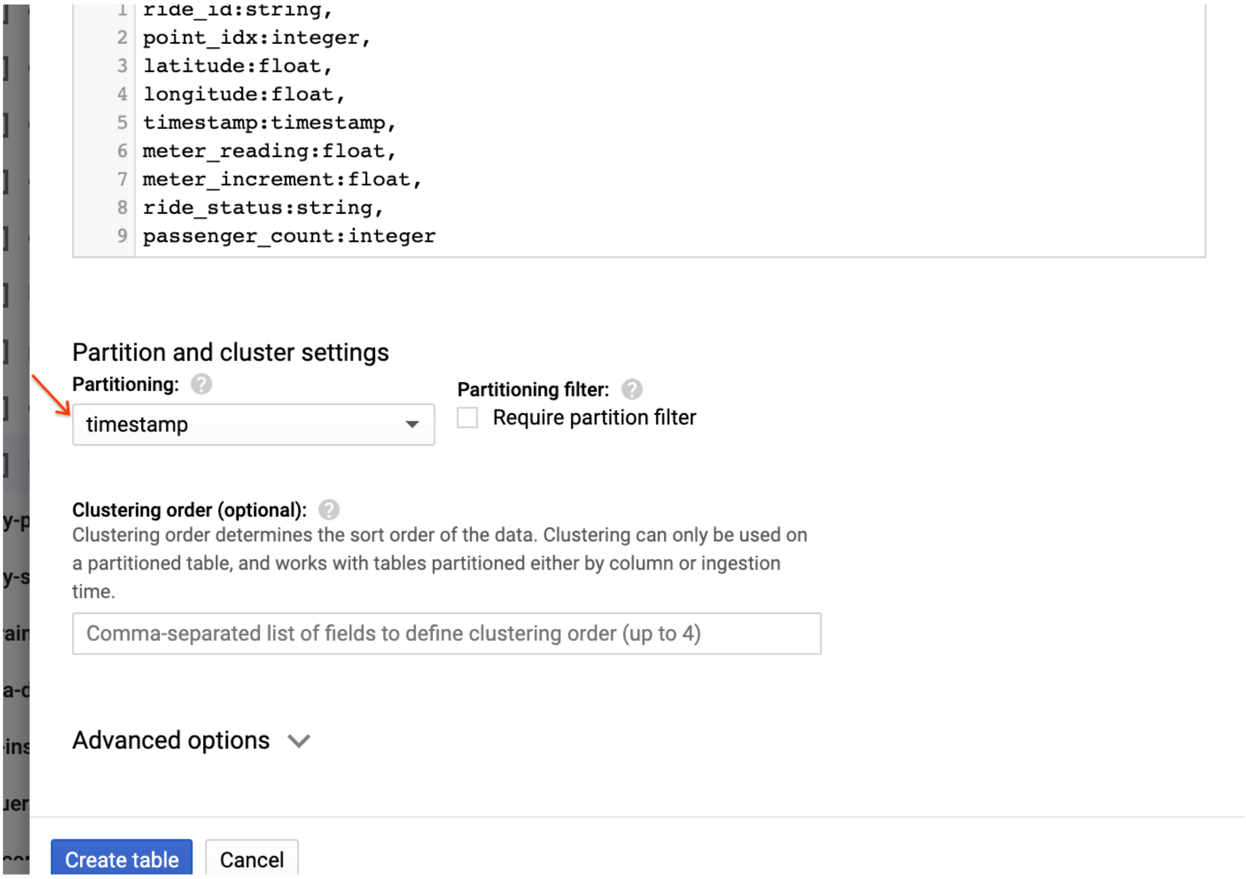


Table name

realtime

Schema

Edit as text



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

Partition and cluster settings

Partitioning: timestamp

Partitioning filter: □ Require partition filter

Clustering order (optional): Clustering order determines the sort order of the data. Clustering can only be used on a partitioned table, and works with tables partitioned either by column or ingestion time.

Comma-separated list of fields to define clustering order (up to 4)

1. Click the **Create table** button.

## Task 2. Create a Cloud Storage bucket

*Skip this step if you already have a bucket created.*

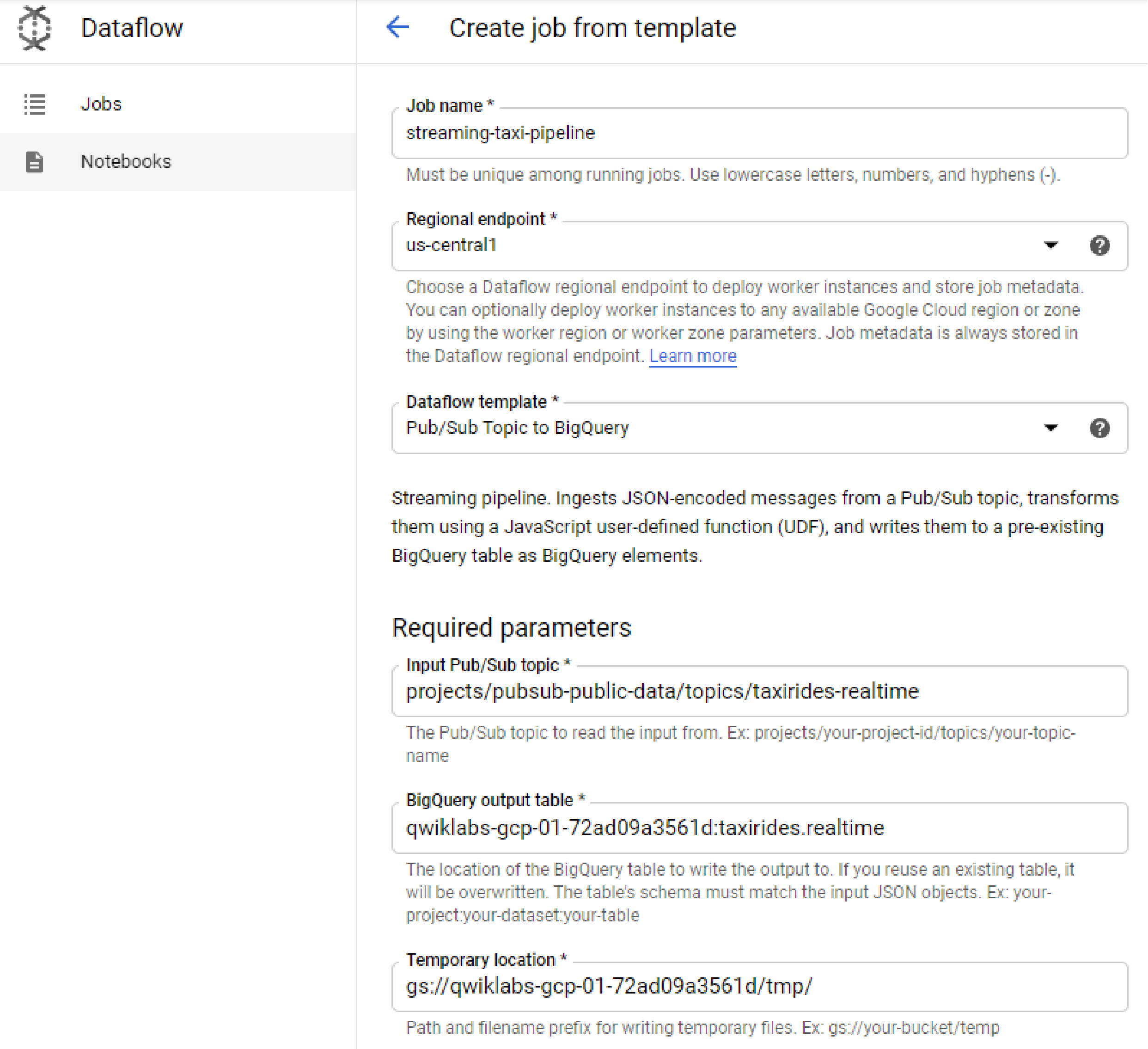
Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download. In this lab,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

1. In the Cloud Console, go to **Navigation menu** > **Storage**.
2. Click **Create bucket**.
3. For **Name**, paste in your **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**.

## 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/`



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

Regional endpoint *
us-central1
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 *
Pub/Sub Topic to BigQuery

Streaming pipeline. Ingests JSON-encoded messages from a Pub/Sub topic, transforms them using a JavaScript user-defined function (UDF), and writes them to a pre-existing BigQuery table as BigQuery elements.

Required parameters

Input Pub/Sub topic *
projects/pubsub-public-data/topics/taxirides-realtime
The Pub/Sub topic to read the input from. Ex: projects/your-project-id/topics/your-topic-name

BigQuery output table *
qwiklabs-gcp-01-72ad09a3561d:taxirides.realtime
The location of the BigQuery table to write the output to. If you reuse an existing table, it will be overwritten. The table's schema must match the input JSON objects. Ex: your-project:your-dataset:your-table

Temporary location *
gs://qwiklabs-gcp-01-72ad09a3561d/tmp/
Path and filename prefix for writing temporary files. Ex: gs://your-bucket/temp

1. Click the **Run Job** button.

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

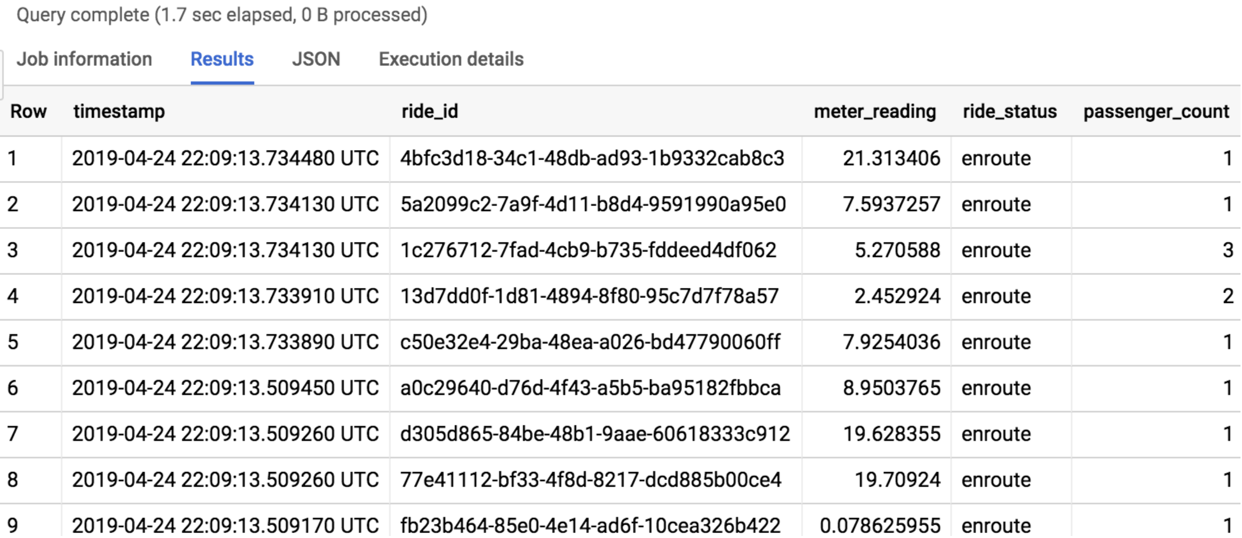
## Task 4. Analyze the taxi data using BigQuery

To analyze the data as it is streaming:

1. In the Cloud 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:



Query complete (1.7 sec elapsed, 0 B processed)

| 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 |

## Task 5. Perform aggregations on the stream for reporting

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

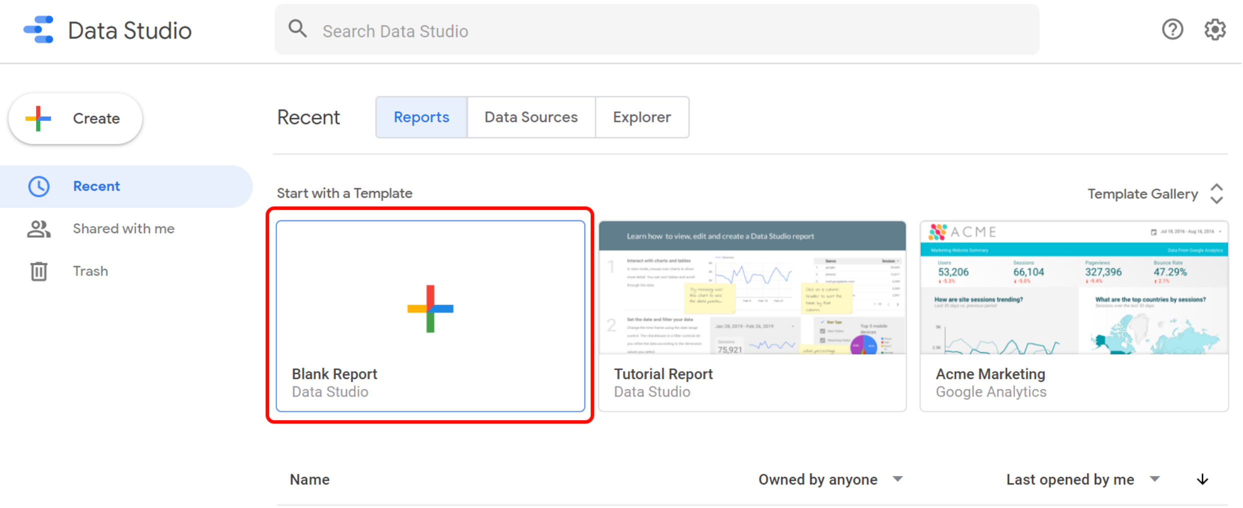
```
WITH streaming_data AS (

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

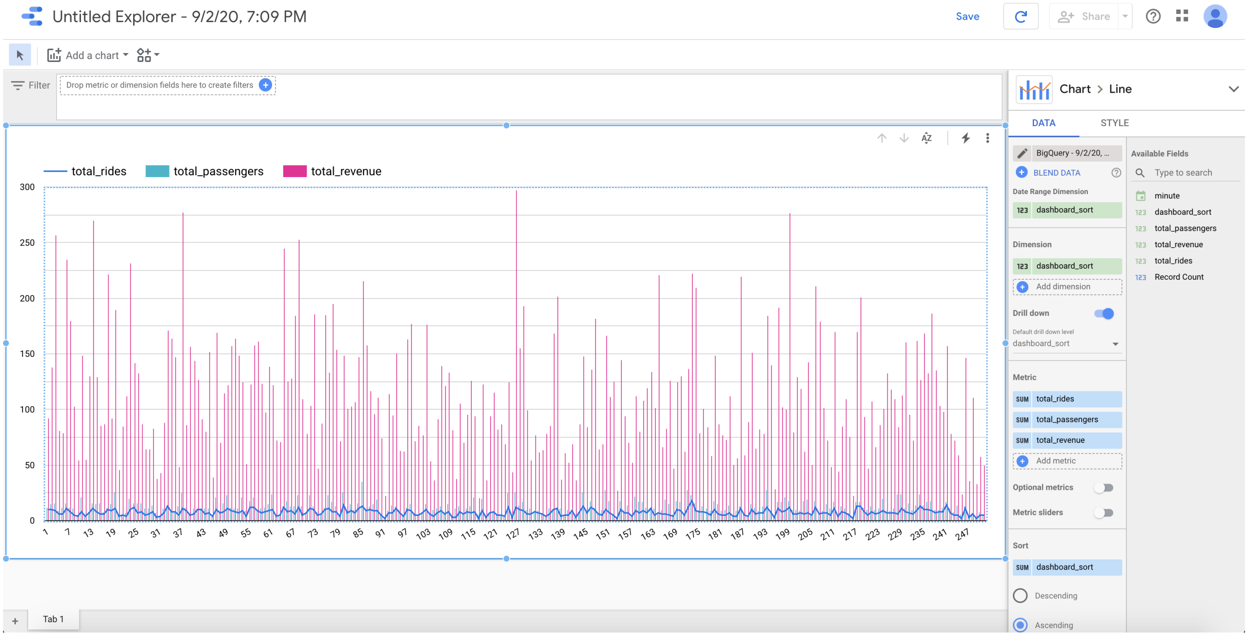
1. Open this Google Data Studio link in a new incognito browser tab.
2. On the **Reports** page, in the **Start with a Template** section, click the **[+] Blank Report** template.



1. If prompted with the **Welcome to Google Studio** window, click **Get started**.
2. Check the checkbox to acknowledge the Google Data Studio Additional Terms, and click **Accept**.
3. Select **No thanks** to all 4 questions, then click **Done**.
4. Switch back to the **BigQuery** Console.
5. Click **Explore Data > Explore with Data Studio** in BigQuery page.
6. Click **Get Started**, then click **Authorize**.
7. 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)



Untitled Explorer - 9/2/20, 7:09 PM

Chart > Line

DATA | STYLE

BigQuery - 9/2/20, ...

Date Range Dimension: 123 dashboard_sort

Dimension: 123 dashboard_sort

Drill down: on

Default drill down level: dashboard_sort

Metric:
- SUM total_rides
- SUM total_passengers
- SUM total_revenue

Optional metrics: off

Metric sliders: off

Sort: SUM dashboard_sort

○ Descending

● Ascending

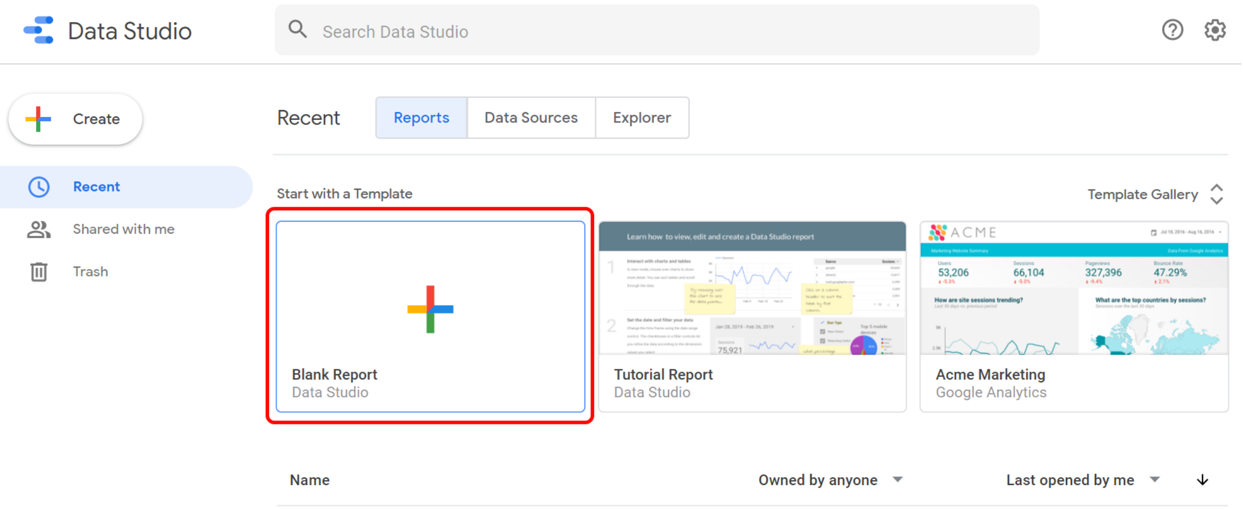
Available Fields: minute, dashboard_sort, total_passengers, total_revenue, total_rides, Record Count

**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.

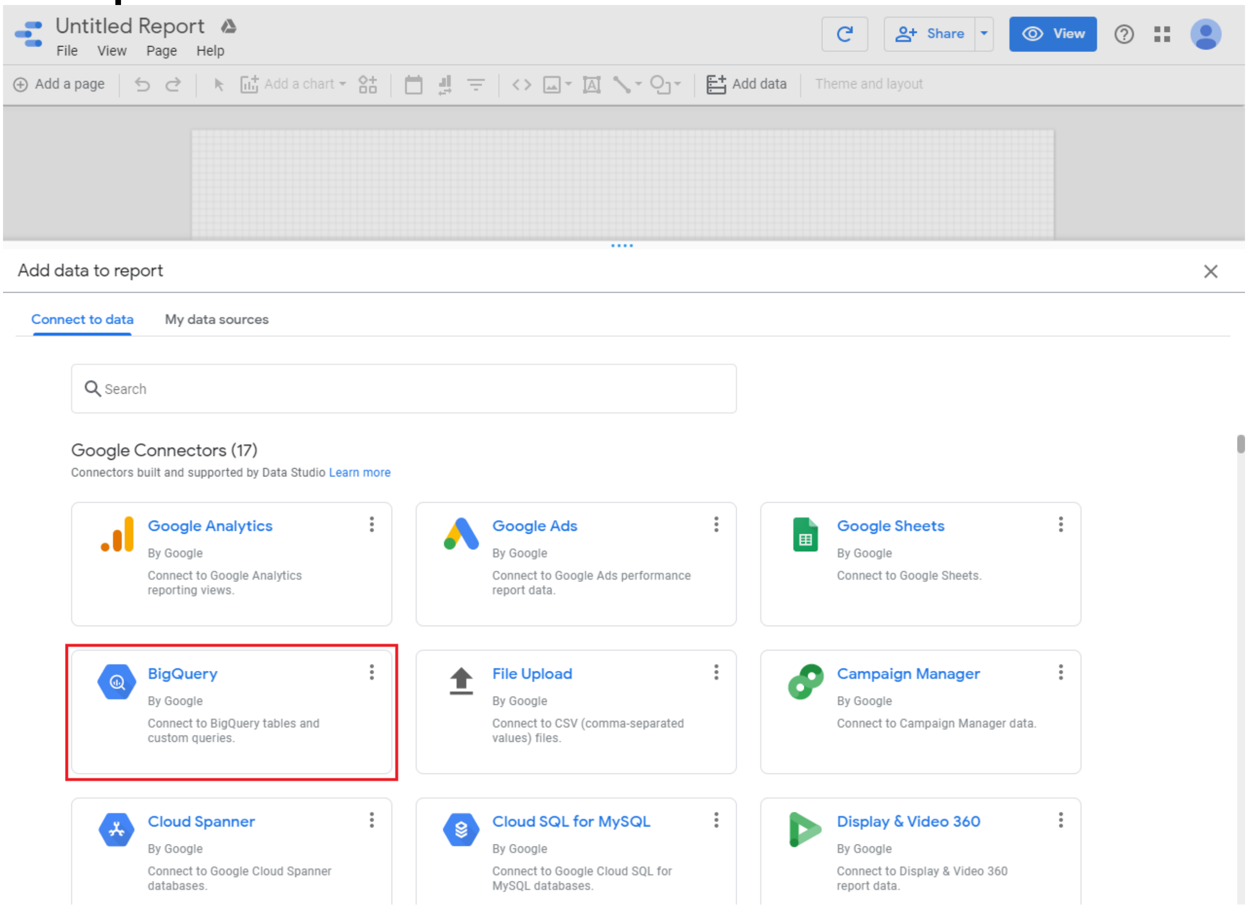
1. When you're happy with your dashboard, click **Save** to save this data source.
2. Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking on the Refresh button near the Save button.

## 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.



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



Google Connectors (17)

Connectors built and supported by Data Studio Learn more

Google Analytics — By Google — Connect to Google Analytics reporting views.

Google Ads — By Google — Connect to Google Ads performance report data.

Google Sheets — By Google — Connect to Google Sheets.

BigQuery — By Google — Connect to BigQuery tables and custom queries.

File Upload — By Google — Connect to CSV (comma-separated values) files.

Campaign Manager — By Google — Connect to Campaign Manager data.

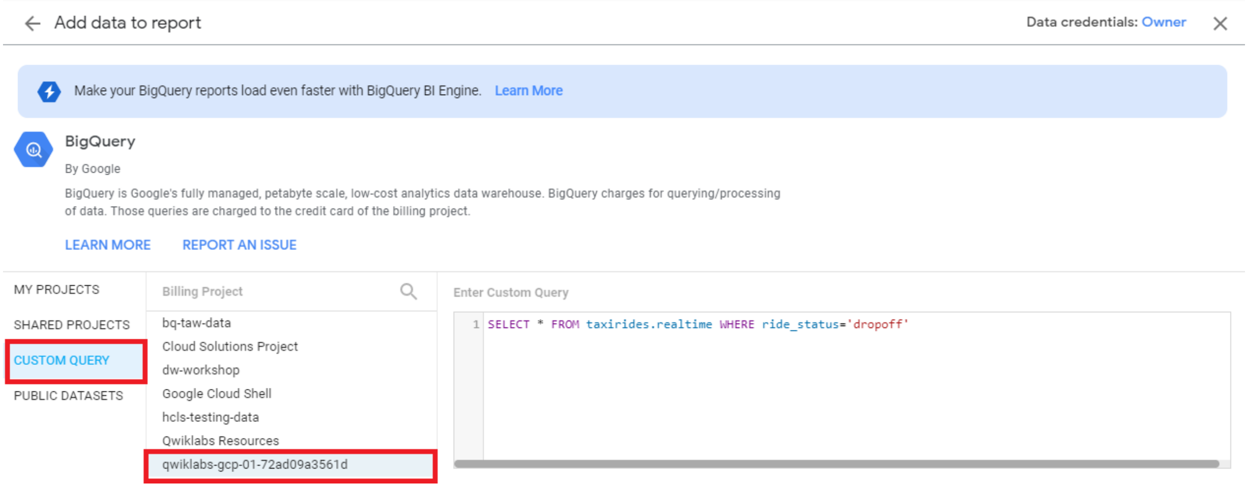
Cloud Spanner — By Google — Connect to Google Cloud Spanner databases.

Cloud SQL for MySQL — By Google — Connect to Google Cloud SQL for MySQL databases.

Display & Video 360 — By Google — Connect to Display & Video 360 report data.

1. From the list of **Google Connectors**, select the **BigQuery** tile.
2. Under **Custom query**, click **qwiklabs-gcp-xxxxxxx** > **Enter Custom Query**, add the following query.

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



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

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.

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

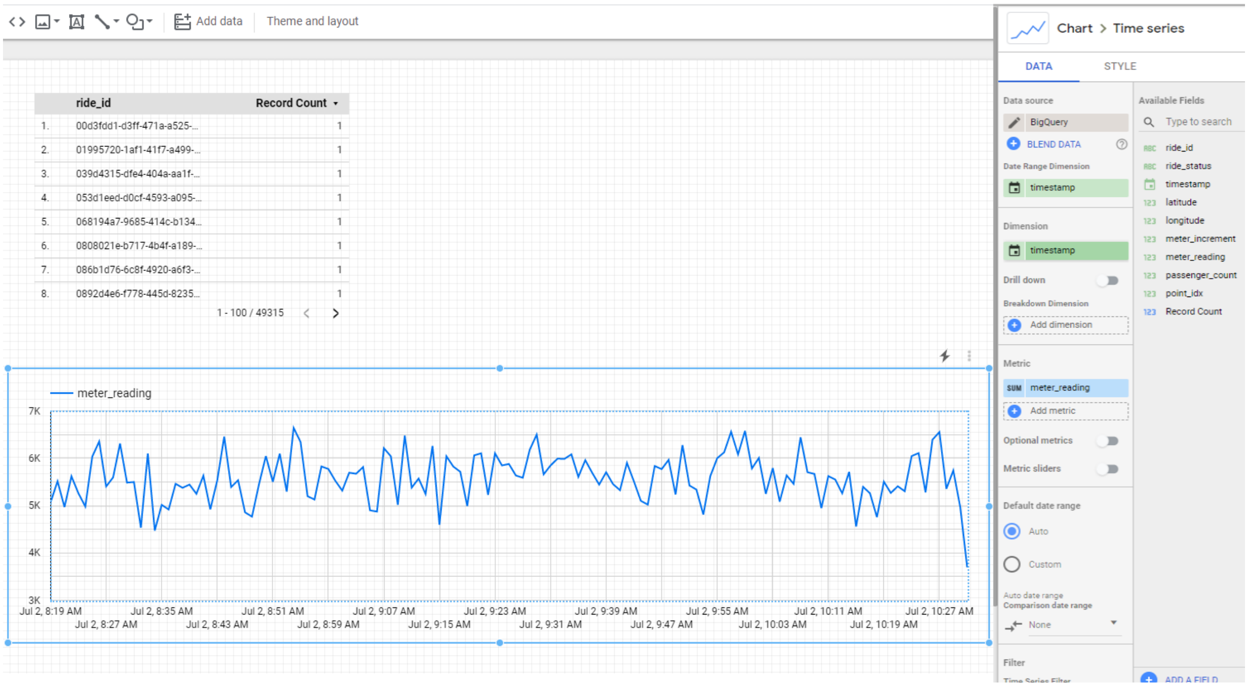
1. Click **Add > Add to Report**.

## Create a time series chart

1. In the **Data panel**, scroll down to the bottom right and click **Add a Field** section. Click **All Fields** on the left corner.
2. Change the field **timestamp** type to **Date & Time > Date Hour Minute (YYYYMMDDhhmm)**.
3. 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:



|  | ride_id | Record Count ▾ |
|---|---|---|
| 1. | 00d3fdd1-d3ff-471a-a525-... | 1 |
| 2. | 01995720-1af1-41f7-a499-... | 1 |
| 3. | 039d4315-dfe4-404a-aa1f-... | 1 |
| 4. | 053d1eed-d0cf-4593-a095-... | 1 |
| 5. | 068194a7-9685-414c-b134... | 1 |
| 6. | 0808021e-b717-4b4f-a189-... | 1 |
| 7. | 086b1d76-6c8f-4920-a6f3-... | 1 |
| 8. | 0892d4e6-f778-445d-8235... | 1 |

1 - 100 / 49315

Data source: BigQuery
BLEND DATA
Date Range Dimension: timestamp
Dimension: timestamp
Drill down
Breakdown Dimension: Add dimension
Metric: SUM meter_reading
Add metric
Optional metrics
Metric sliders
Default date range
● Auto
○ Custom
Auto date range
Comparison date range: None
Filter
Time Series Filter

Available Fields
Type to search
ride_id
ride_status
timestamp
latitude
longitude
meter_increment
meter_reading
passenger_count
point_idx
Record Count
ADD A FIELD

## Task 8. Stop the Dataflow job

1. Navigate back to **Dataflow**.
2. Click the **streaming-taxi-pipeline**.
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.