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

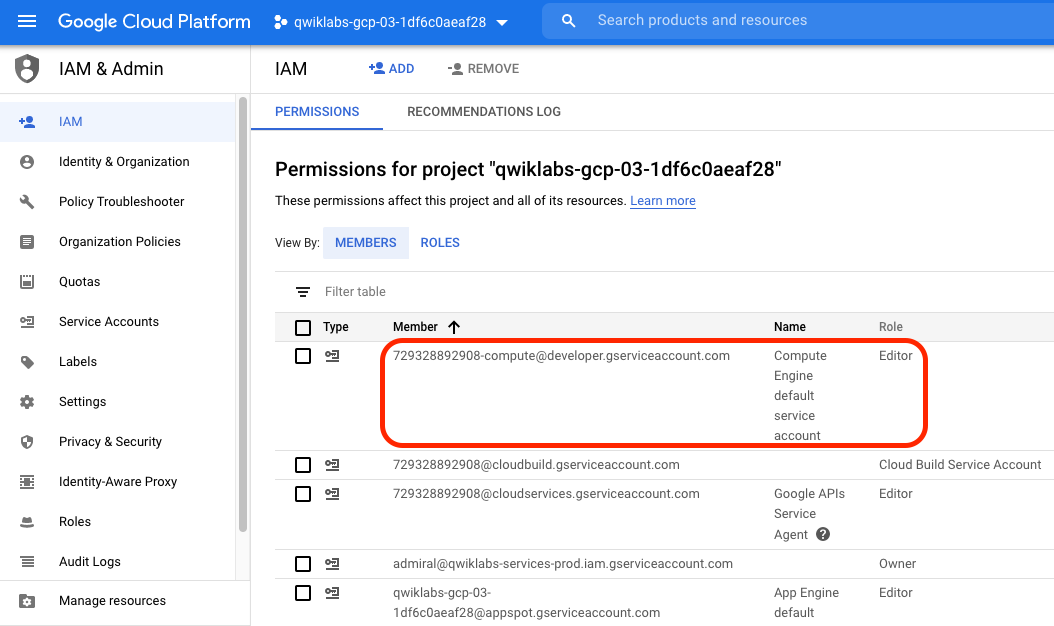
# Overview

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 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.

### **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**.



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

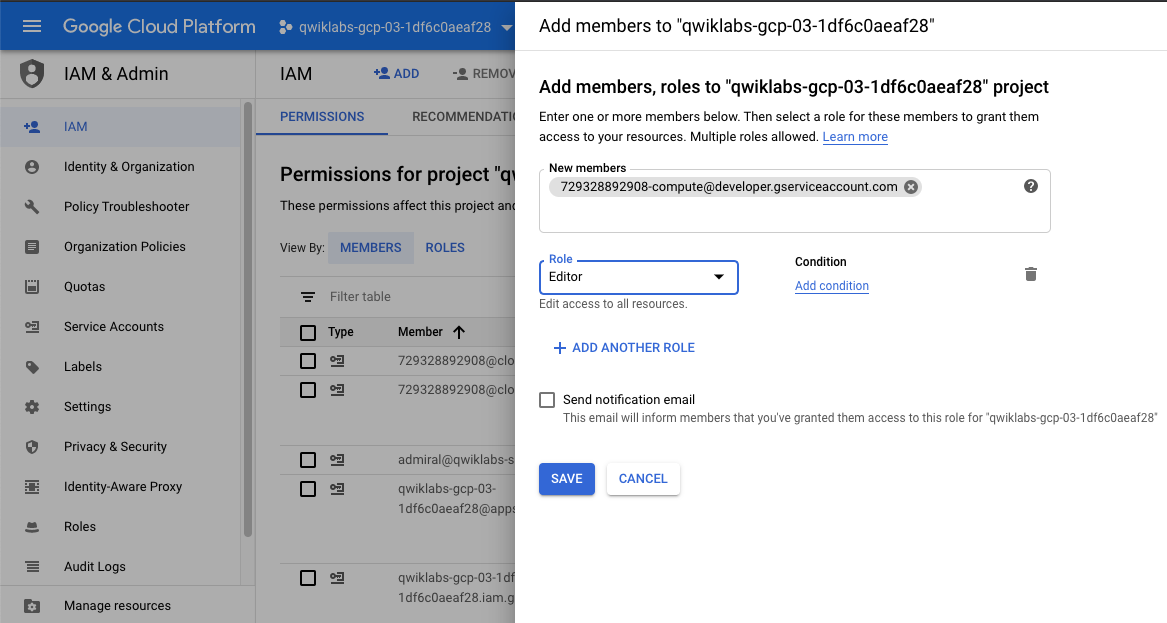
* In the Google Cloud console, on the **Navigation menu**, click **Home**.
* Copy the project number (e.g. 729328892908).
* On the **Navigation menu**, click **IAM & Admin** > **IAM**.
* At the top of the **IAM** page, click **Add**.
* For **New members**, type:

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

content\_copy

Replace {project-number} with your project number.

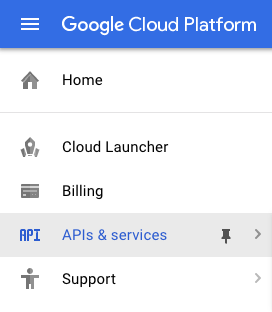
* For **Role**, select **Project** > **Editor**. Click **Save**.



## 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](https://cloud.google.com/pubsub/) 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](https://data.cityofnewyork.us/).

[BigQuery](https://cloud.google.com/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 taxiridescontent\_copy

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\_increment:float,ride\_status:string,\

passenger\_count:integer -t taxirides.realtimecontent\_copy

### **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,

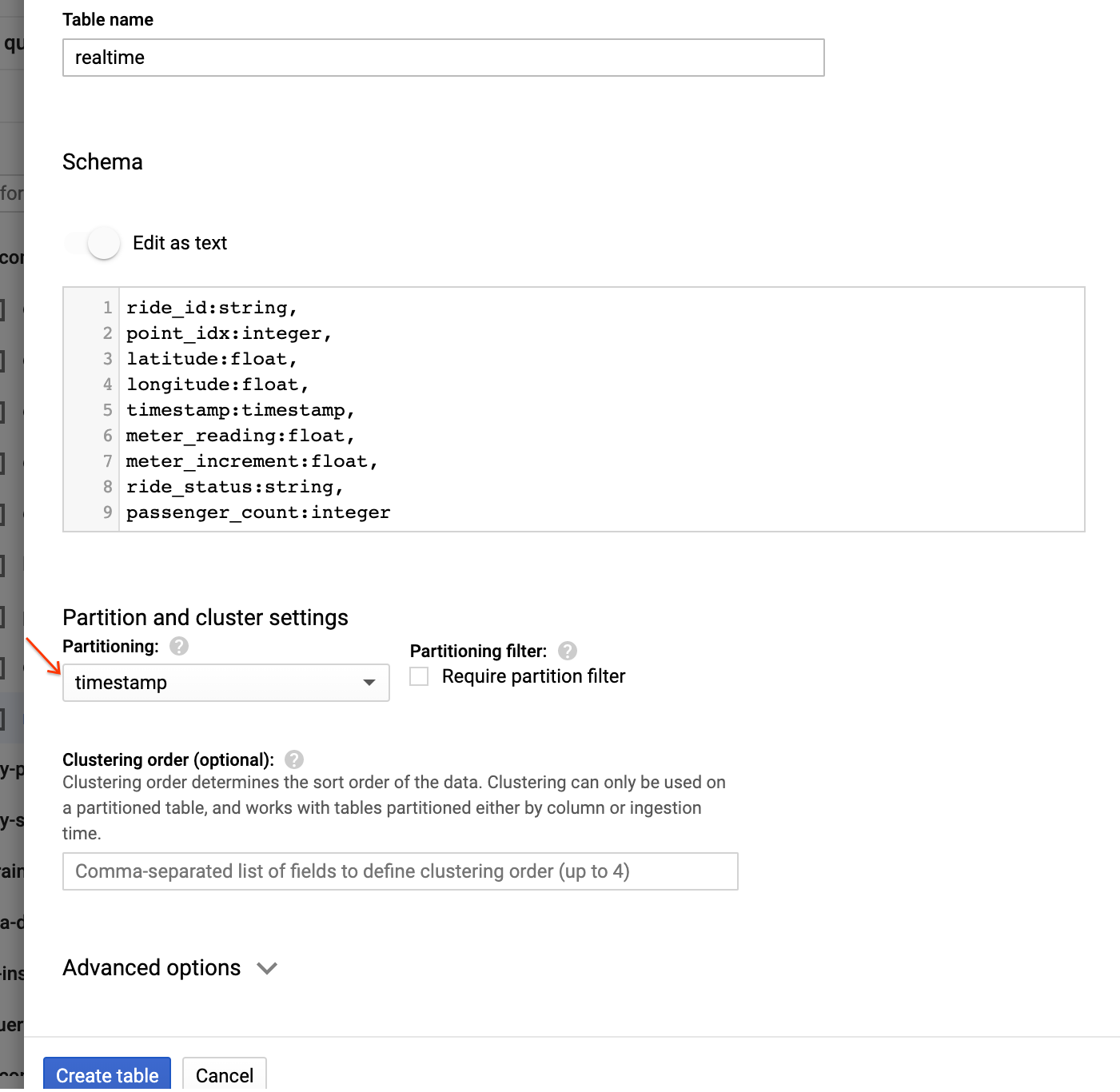
meter\_reading:float,

meter\_increment:float,

ride\_status:string,

passenger\_count:integercontent\_copy

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



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](https://cloud.google.com/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**.

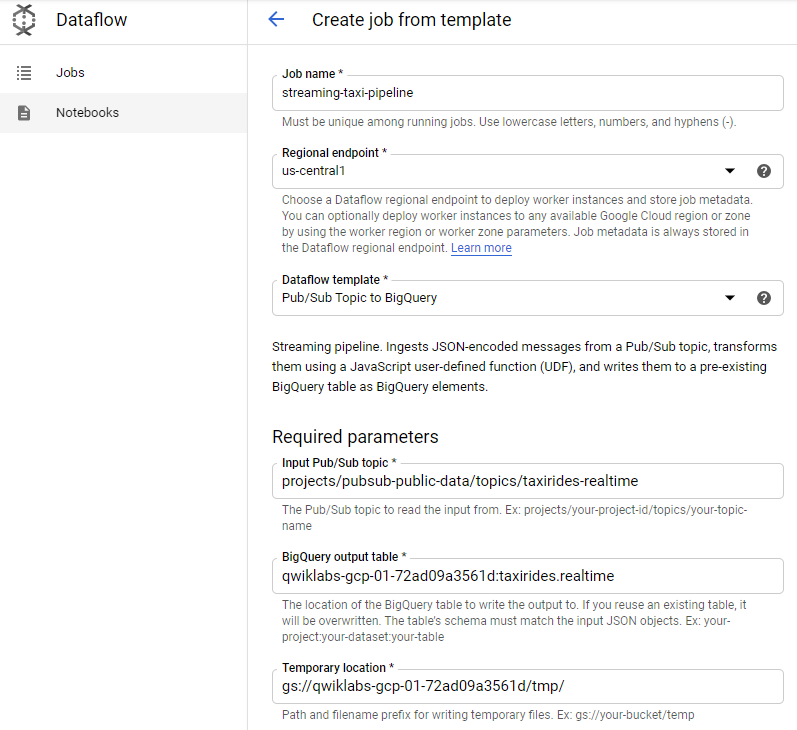
## Task 3. Set up a Dataflow Pipeline

[Dataflow](https://cloud.google.com/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

1. Under **Temporary location**, enter gs://<mybucket>/tmp/



1. Click the **Run Job** button.

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

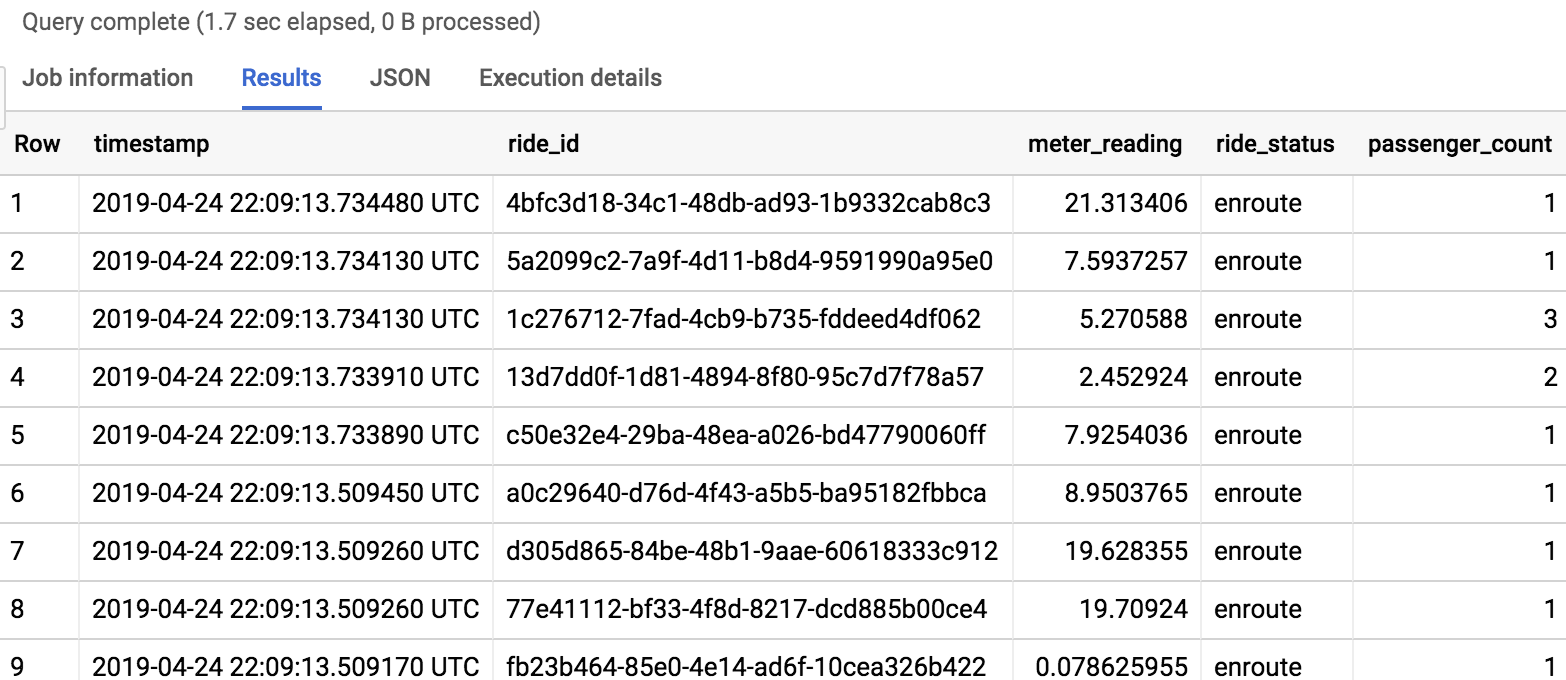
## Task 4. Analyze the taxi data using BigQuery

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 10content\_copy

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:



## 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

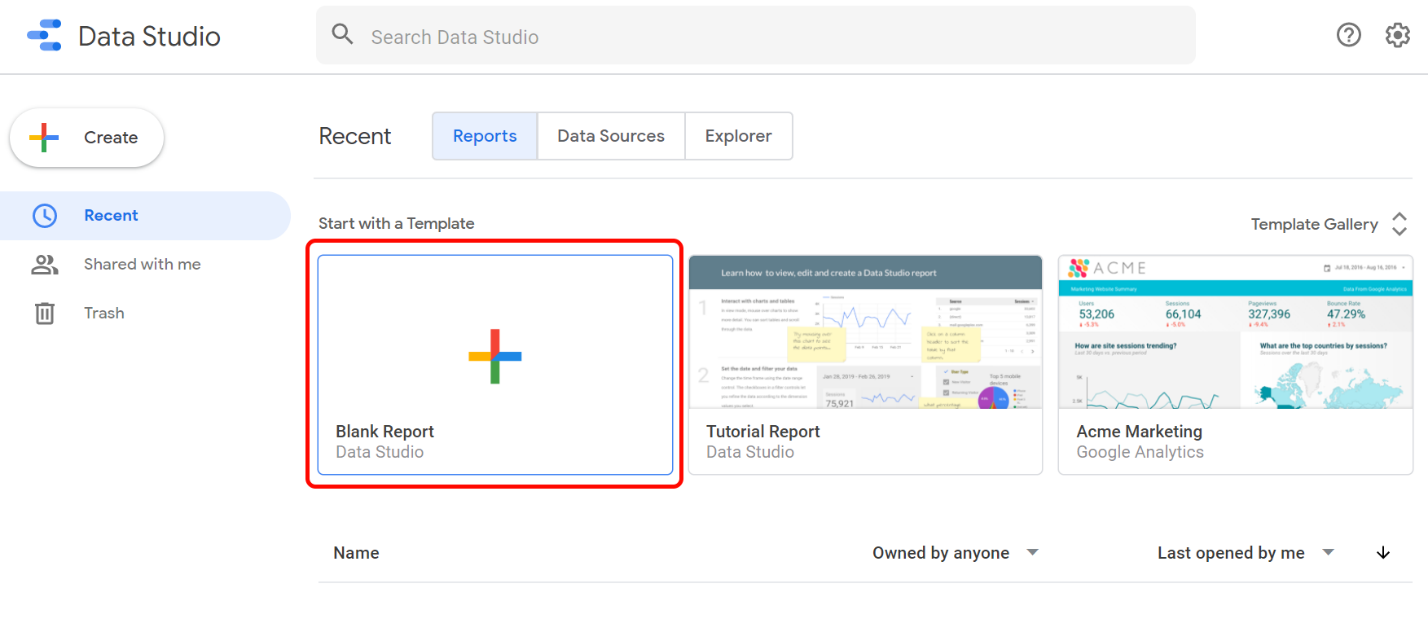
FROM streaming\_data

GROUP BY minute, timestampcontent\_copy

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](https://datastudio.google.com/) 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)

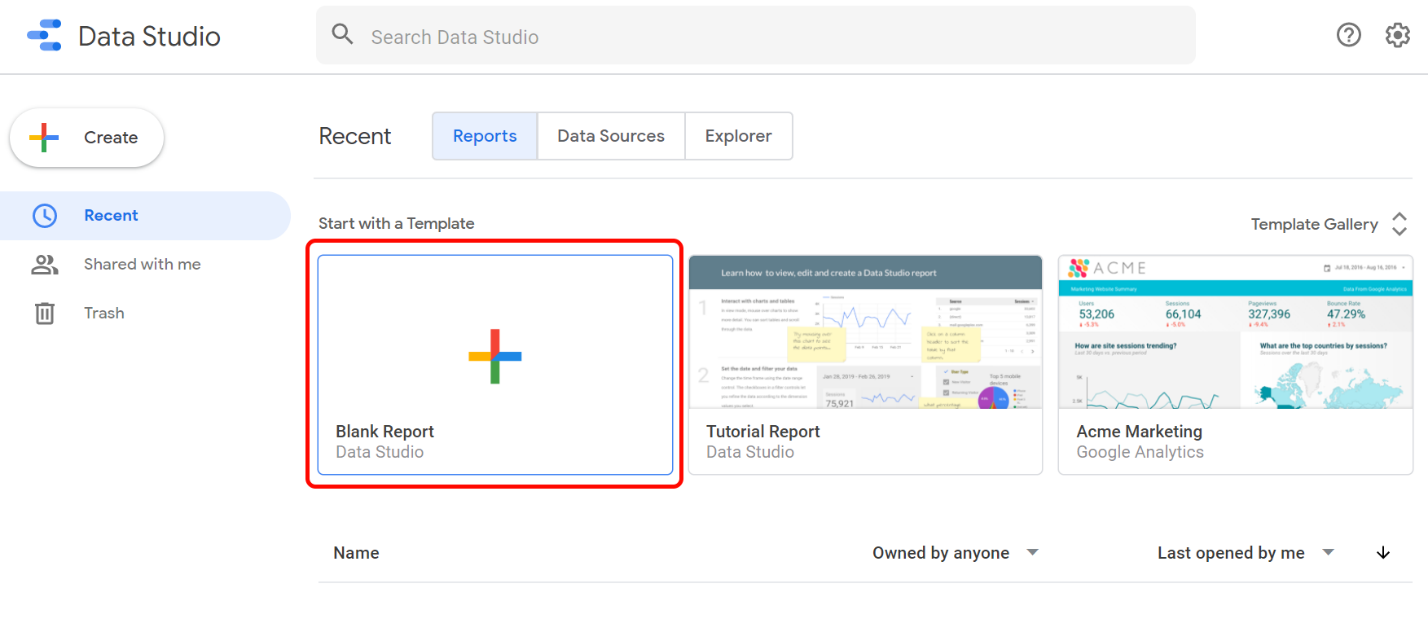


**Note:** Visualizing data at a minute-level granularity is currently not supported in Data Studio as a timestamp. This is why we created our own dashboard\_sort dimension.

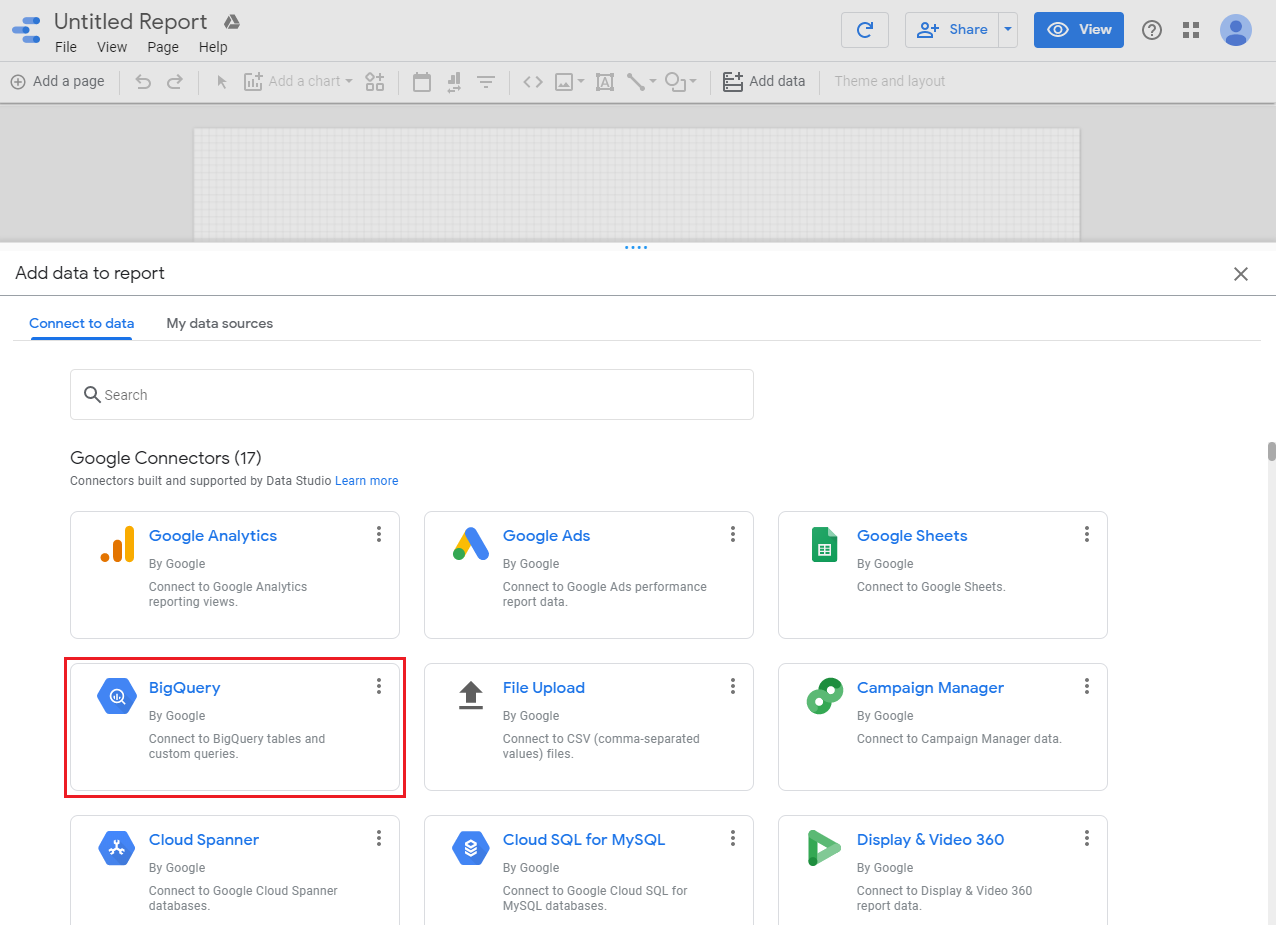
1. When you're happy with your dashboard, click **Save** to save this data source.
2. Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking on the Refresh button near the Save button.

Task 7. Create a time series dashboard

1. Click this [Google Data Studio link](https://datastudio.google.com/) 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**.



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

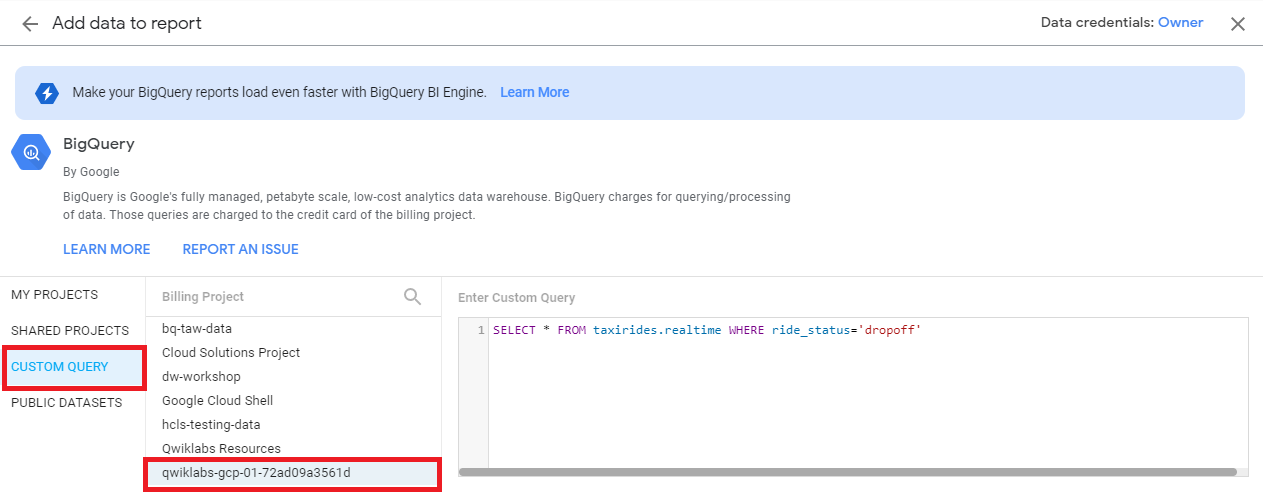
\*

FROM

taxirides.realtime

WHERE

ride\_status='dropoff'content\_copy



1. Click **Add > Add to Report**.
2. **Create a time series chart**
3. In the **Data panel**, scroll down to the bottom right and click **Add a Field** section. Click **All Fields** on the left corner.
4. Change the field **timestamp** type to **Date & Time > Date Hour Minute (YYYYMMDDhhmm)**.
5. Click **Done**.
6. Click **Add a chart**.
7. Choose **Time series chart**.
8. Position the chart in the bottom left corner - in the blank space.
9. 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:

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.