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
- Subscribe to a Pub/Sub topic
- 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

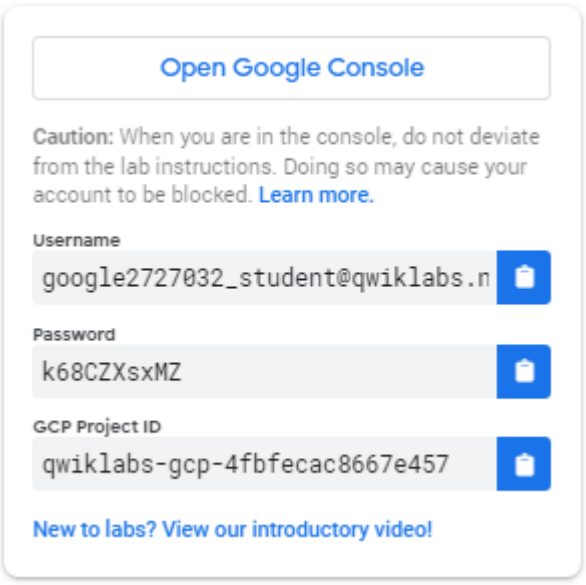
To complete this lab, you need:

- 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

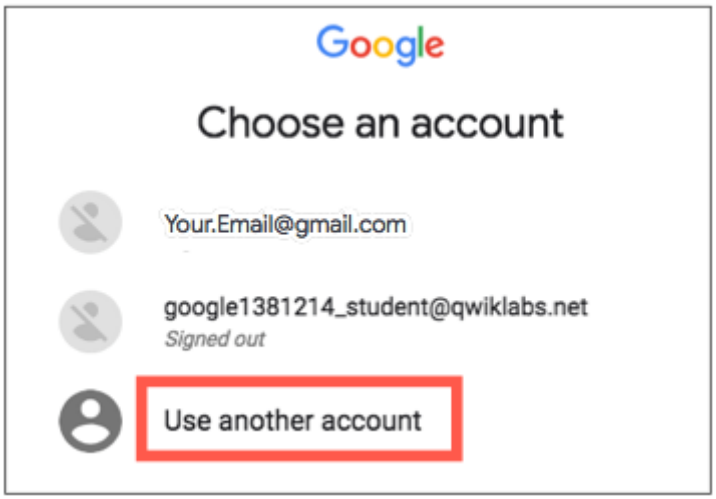
1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



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.



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

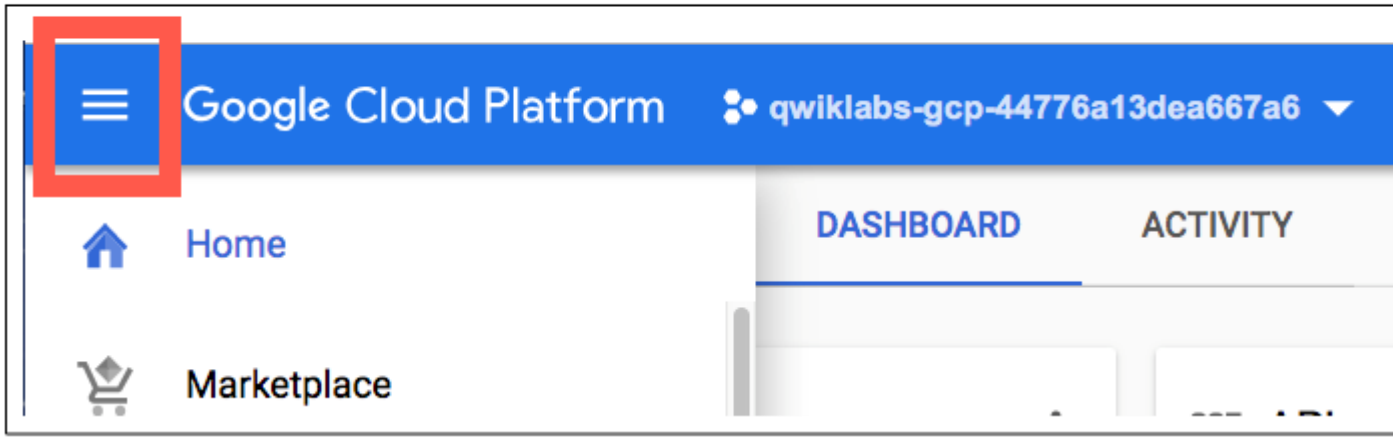
Note: You must use the credentials from the Connection Details panel. Do not use your Google Cloud Skills Boost credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

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.

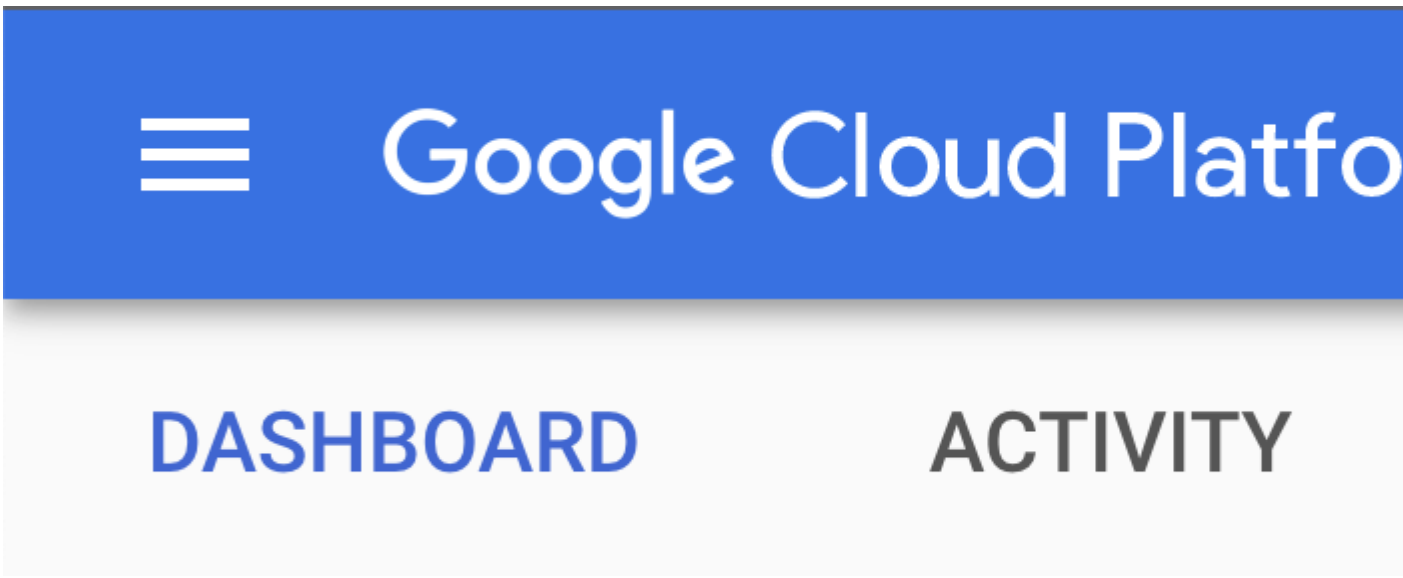


Activate Google Cloud Shell

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.

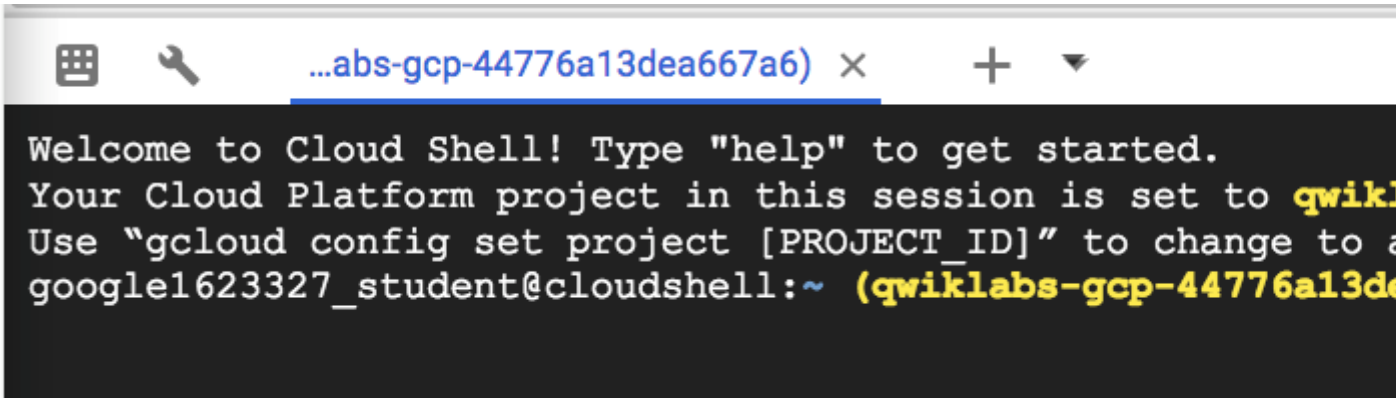
Google Cloud Shell provides command-line access to your Google Cloud resources.

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



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:



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

Copied!

content_copy

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

Copied!

content_copy

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. Source a pre-created Pub/Sub topic and 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.

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 an extract of the NYC Taxi & Limousine Commission's open dataset. The Pub/Sub topic has already been created and populated in your project environment.

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.

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=us-west1 mk taxirides

Copied!

content_copy

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

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

Copied!

content_copy

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, click **us-west1 (Oregon)**, and 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

Copied!

content_copy

10. In **Partition and cluster settings**, select **timestamp**.
11. Click **Create Table**.

Task 2. Create a Cloud Storage bucket

In this task, you create a Cloud Storage bucket to provide working space for your Dataflow pipeline.

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.

1. In the Cloud console, in the **Navigation menu** ( ), click **Cloud Storage > Buckets**.
2. Click **Create Bucket**.
3. For **Name**, copy and paste in your **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

In this task, 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.

Dataflow is a serverless way to carry out data analysis.

Restart the connection to the Dataflow API.

1. In the Cloud console, in the top search bar, type **Dataflow API**, and then press ENTER.
2. In the search results window, click **Dataflow API**.
3. Click **Manage**.
4. Click **Disable API**.
5. In the Disable dialog, click **Disable**.
6. Click **Enable**.

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 **us-central1 (Iowa)**.
5. In **Dataflow template**, select the **Pub/Sub Topic to BigQuery** template.
6. In **Input Pub/Sub topic**, select the topic that already exists in your project from the dropdown list . It will appear like the following:

projects/<myprojectid>/topics/taxirides-realtime

7. In **BigQuery output table**, type <myprojectid>:taxirides.realtime

Note: You must replace **myprojectid** with your Project ID.**Note**: There is a colon : between the project and dataset name and a dot . between the dataset and table name.

8. In Temporary location, click **Browse**.
9. Click **view child resources**(›).
10. Click **Create new folder**(📁), and then type the name **tmp**.
11. Click **Create**, and then click **Select**.
12. Click **Show Optional Parameters**.
13. In **Max workers**, type **2**
14. In **Number of workers**, type **1**
15. 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.

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
```

Copied!

content_copy

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:

Job information | Results | JSON | Execution details

Row	ride_id	point_idx	latitude	longitude
1	8619e1b1-6fdb-4ae7-b82b-1585e6b83aad	98	40.76549000000001	
2	c91462c0-851f-46e5-b429-a663e439e755	464	40.755680000000005	-73.8997
3	d46e1801-82fe-4d9d-8fda-266dc34c963f	5	40.73346	
4	c9f4a384-561b-48b8-b4aa-40506c7e3593	21	40.75601	-73.9706
5	a1f2a99c-167e-464a-a7ee-3121d9a6c2e0	1169	40.74168	-73.9559
6	9d36deb8-dd53-4e57-b4e5-203b1d16c831	533	40.802330000000005	
7	9d631e31-c8e6-4207-a56b-1b5dee07be6d	26	40.773320000000005	
8	b5e7b939-a14c-4d25-a1f3-c9a54655b60a	1849	40.72554	

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
```

Copied!

content_copy

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.

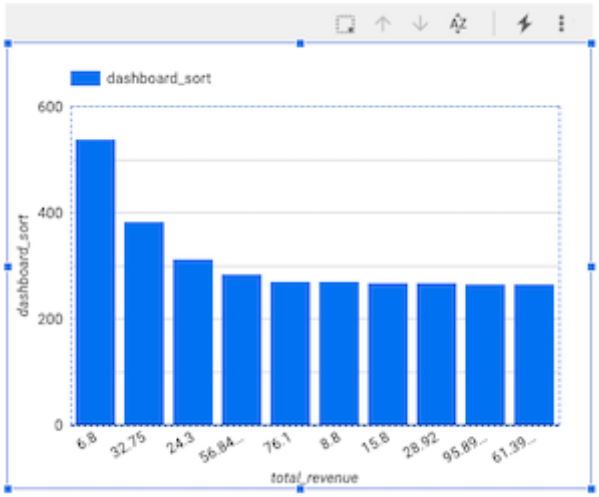
1. In the Cloud console, in the **Navigation menu** (☰), click **BigQuery**.
2. In the Explorer Pane, expand your **Project ID**.
3. Expand **Saved queries**, and then click **My Saved Query**.

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.

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



(

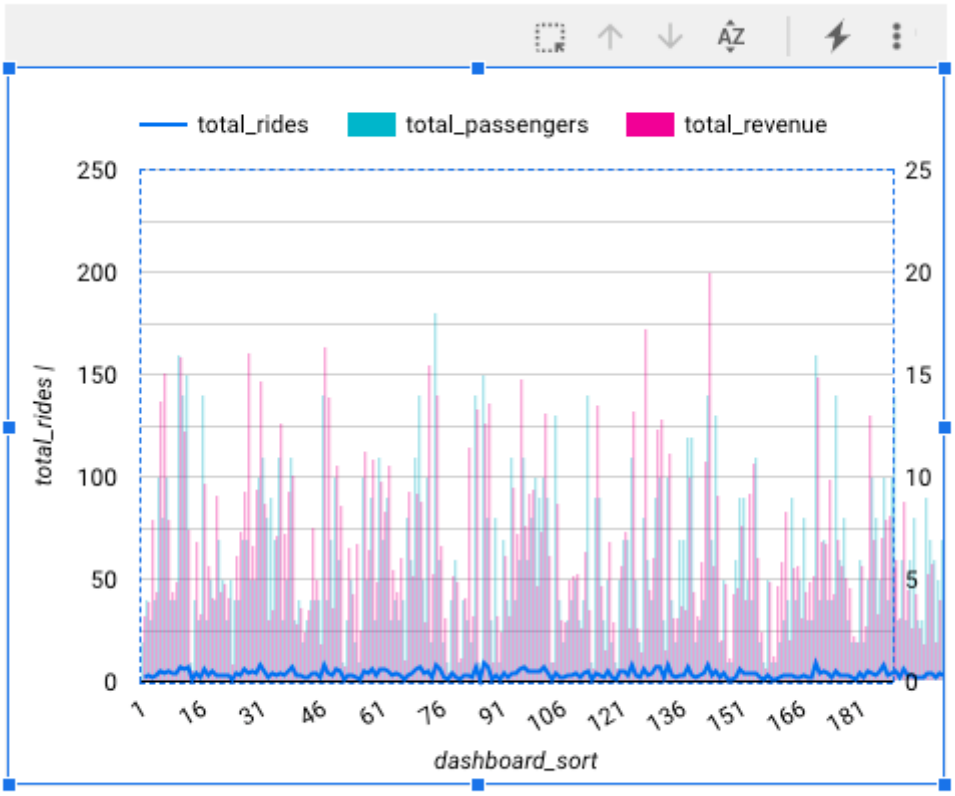
The Chart pane appears.

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 **Add metric**, and then select **total_passengers**.
13. In **Setup > Metric**, click **Record Count**, 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 for email preferences, answer **no** to all, then click **Continue**.
19. If prompted with the **Review data access before saving** window, click **Acknowledge and save**.
20. Click **Add to report**.

21. 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='dropoff'
```

Copied!

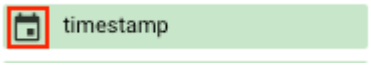
content_copy

6. Click **Add > Add To Report**.

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

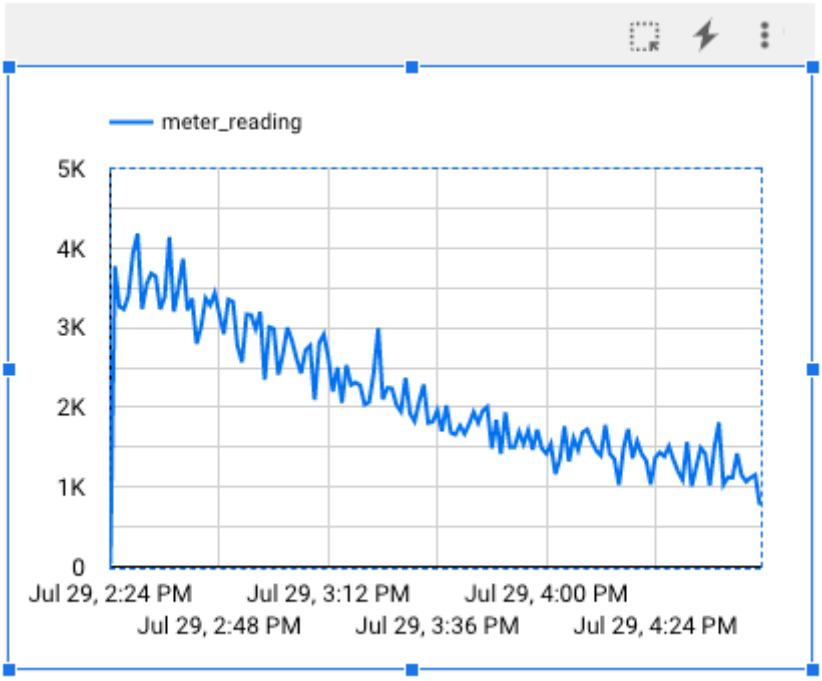
Create a time series chart

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**. 

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**.

Your time series chart should look similar to this but will vary based upon the volume and rate of data that was loaded from Pub/Sub:



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**. 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**.

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 2022 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.