

GCP Fundamentals: Getting Started with BigQuery

In this lab, you load a web server log into a BigQuery table. After loading the data, you query it using the BigQuery web user interface and the BigQuery CLI.

BigQuery helps you perform interactive analysis of petabyte-scale databases, and it enables near-real time analysis of massive datasets. It offers a familiar SQL 2011 query language and functions.

Data stored in BigQuery is highly durable. Google stores your data in a replicated manner by default and at no additional charge for replicas. With BigQuery, you pay only for the resources you use. Data storage in BigQuery is inexpensive. Queries incur charges based on the amount of data they process: when you submit a query, you pay for the compute nodes only for the duration of that query. You don't have to pay to keep a compute cluster up and running.

Using BigQuery involves interacting with a number of Google Cloud Platform resources, including projects (covered elsewhere in this course), datasets, tables, and jobs. This lab introduces you to some of these resources, and this brief introduction summarizes their role in interacting with BigQuery.

Datasets: A dataset is a grouping mechanism that holds zero or more tables. A dataset is the lowest level unit of access control. Datasets are owned by GCP projects. Each dataset can be shared with individual users.

Tables: A table is a row-column structure that contains actual data. Each table has a schema that describes strongly typed columns of values. Each table belongs to a dataset.

Objectives

In this lab, you learn how to perform the following tasks:

- Load data from Cloud Storage into BigQuery.
- Perform a query on the data in BigQuery.

Task 1: Sign in to the Google Cloud Platform (GCP) Console

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click Start Lab, shows how long 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 the Google Cloud Platform 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 GCP account or project, do not use it for this lab.
- Make a note of whether your assigned region is closer to the United States or to Europe.

Task 2: Load data from Cloud Storage into BigQuery



1. In the Console, on the **Navigation menu** () click **BigQuery** then click **Done**.
2. Create a new dataset within your project by selecting your project in the Resources section, then clicking on **CREATE DATASET** on the right.
3. In the **Create Dataset** dialog, for **Dataset ID**, type **logdata**.
4. For **Data location**, select the continent closest to the region your project was created in. click **Create dataset**.
5. Create a new table in the **logdata** to store the data from the CSV file.
6. Click on **Create Table**. On the **Create Table** page, in the **Source** section:
 - For **Create table from**, choose select **Google Cloud Storage**, and in the field, type `gs://cloud-training/gcpfci/access_log.csv`.
 - Verify **File format** is set to **CSV**.

Note: When you have created a table previously, the Create from Previous Job option allows you to quickly use your settings to create similar tables.
7. In the **Destination** section:
 - For **Dataset name**, leave **logdata** selected.
 - For **Table name**, type **accesslog**.
 - For **Table type**, **Native table** should be selected and unchangeable.
8. Under **Schema** section, for **Auto detect** check the **Schema and input Parameters**.
9. Accept the remaining default values and click **Create Table**.
BigQuery creates a load job to create the table and upload data into the table (this may take a few seconds).
10. (Optional) To track job progress, click **Job History**.
11. When the load job is complete, click **logdata > accesslog**.
12. On the **Table Details** page, click **Details** to view the table properties, and then click **Preview** to view the table data.
Each row in this table logs a hit on a web server. The first field, **string_field_0**, is the IP address of the client. The fourth through ninth fields log the day, month, year, hour, minute, and second at which the hit occurred. In this activity, you will learn about the daily pattern of load on this web server.

Click *Check my progress* to verify the objective.

Load data from Cloud Storage into BigQuery

Check my progress

Task 3: Perform a query on the data using the BigQuery web UI

In this section of the lab, you use the BigQuery web UI to query the **accesslog** table you created previously.

1. In the **Query editor** window, type (or copy-and-paste) the following query:
2. Because you told BigQuery to automatically discover the schema when you load the data, the hour of the day during which each web hit arrived is in a field called **int_field_6**.

```
3. select int64_field_6 as hour, count(*) as hitcount from logdata.accesslog
4. group by hour
   order by hour
```

Notice that the Query Validator tells you that the query syntax is valid (indicated by the green check mark) and indicates how much data the query will process. The amount of data processed allows you to determine the price of the query using the [Cloud Platform Pricing Calculator](#).

5. Click **Run** and examine the results. At what time of day is the website busiest? When is it least busy?

Task 4: Perform a query on the data using the bq command

In this section of the lab, you use the bq command in Cloud Shell to query the **accesslog** table you created previously.



1. On the **Google Cloud Platform** menu, click **Activate Cloud Shell**. If a dialog box appears, click **Start Cloud Shell**.
2. At the Cloud Shell prompt, enter this command:

```
3. bq query "select string_field_10 as request, count(*) as requestcount  
from logdata.accesslog group by request order by requestcount desc"
```

The first time you use the `bq` command, it caches your Google Cloud Platform credentials, and then asks you to choose your default project. Choose the project that Qwiklabs assigned you to. Its name will look like `qwiklabs-gcp-` followed by a hexadecimal number.

The `bq` command then performs the action requested on its command line. What URL offered by this web server was most popular? Which was least popular?