**LAB: Google Cloud Fundamentals: Getting Started with BigQuery**

**Overview:**

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.

**STEPS:**

1. **Load Data from Cloud Storage into BigQuery**

gcloud compute bq create dataset --dataset-id “logdata” --data-location “us-central1-b”

* Create a new table in the **logdata** to store the data from the CSV file:

gcloud compute bq logdata create table --create-table-form “google-cloud-storage” --field “gs://cloud-training/gcpfci/access\_log.csv” --destination “--dataset-name “logdata” --table-name “accesslog” --table-type “native-table”” --schema “ --auto-detect “schema-and-input-parameter””

 To view the table properties:

gcloud compute bq logdata accesslog details preview

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.

1. **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.

* + In the **Query editor** window, type (or copy-and-paste) the following query:

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**.

select int64\_field\_6 as hour, count(\*) as hitcount from logdata.accesslog 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](https://cloud.google.com/products/calculator/).

1. **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.

* + 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.