**GCS to BigQuery using Bash – Complete Guide**

**Introduction**

I will walk you through how to load data from Google Cloud Storage (GCS) to BigQuery using Bash. This process helps integrate GCS and BigQuery, making it easier to perform data analytics and streamline ETL (Extract, Transform, Load) pipelines. By the end of this guide, you will have a clear understanding of how to use the command line to automate the process of moving data from GCS to BigQuery.

**Step-by-Step Process**

**1. Set Up Your Google Cloud Environment**

To begin, I need to ensure that I have the necessary tools installed:

**1.1 Install Google Cloud SDK**: I followed the installation guide.

**1.2 Authenticated with Google Cloud**: Once the SDK is installed, I authenticated using my service account by running the following command in my terminal:

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Google Cloud Storage (GCS):**

In the below screenshot, Here are the details:

bucket-name: covid19data202024

blob-name: WHO\_COVID19\_globaldata.csv

**A screenshot of a computer

AI-generated content may be incorrect.  
Create a BigQuery Dataset**

Before I can load data into BigQuery, I need to set up a dataset. If the dataset already exists, I can skip this step.

I use the following command to create a new dataset in BigQuery:

bq mk --dataset --description "Data from GCS to BigQuery" <project-id>:<dataset-name>



We can see in the below Screenshot that “Staging” Dataset has been created under my project ID:   
caramel-duality-452616-e6

A screenshot of a computer

AI-generated content may be incorrect.

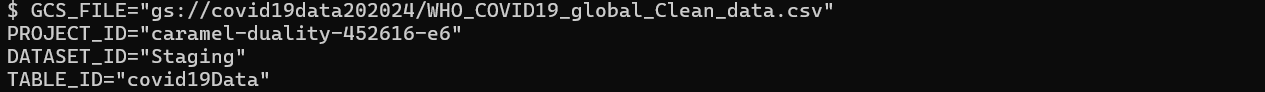
A screenshot of a computer

AI-generated content may be incorrect.

**3. Define the GCS File and BigQuery Table**

Next, I specify the source file in Google Cloud Storage (GCS) and the destination table in BigQuery.

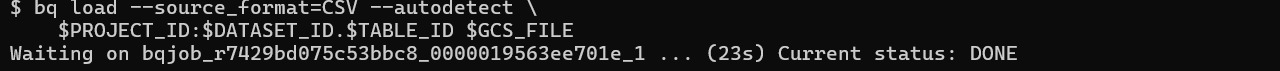
* Here are the variables I set:
* GCS\_FILE: This is the GCS file path I want to load into BigQuery.
* ROJECT\_ID: My Google Cloud project ID.
* DATASET\_ID: The dataset in BigQuery where I want to store my data.
* TABLE\_ID: The name of the table in BigQuery where the data will be loaded.



**4. Load Data from GCS to BigQuery**

Now, I can load the data into BigQuery using the bq load command. The following command loads the CSV data from GCS into BigQuery:

bq load --source\_format=CSV --autodetect \

$PROJECT\_ID:$DATASET\_ID.$TABLE\_ID $GCS\_FILE  
  


**Explanation of Parameters**:

* --source format=CSV: I specify that the source file is in CSV format.
* --autodetect: BigQuery will automatically detect the schema of the CSV file.
* $PROJECT\_ID:$DATASET\_ID.$TABLE\_ID: Specifies where in BigQuery the data will go.
* $GCS\_FILE: Specifies the path to the file in Google Cloud Storage.

View the BigQuery in the following screenshot and can see covid19Data under Staging and can see the data below:

A screenshot of a computer

AI-generated content may be incorrect.

Once I run the command, the data begins loading into BigQuery. The job status will be displayed in the terminal.

bq query --use\_legacy\_sql=false "SELECT \* FROM \`$PROJECT\_ID.$DATASET\_ID.$TABLE\_ID\` LIMIT 10"

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Why GCS to BigQuery in Real-Time?**

Moving data from Google Cloud Storage (GCS) to BigQuery has several real-time use cases:

**1. Data Analytics**: I can store raw data in GCS and then load it into BigQuery for complex analyses using SQL queries. This is particularly useful for large datasets that require powerful analytics tools.

**2. Automated ETL Pipelines**: By automating the process of moving data from GCS to BigQuery, I can build efficient ETL pipelines that run at scheduled intervals, ensuring my data is always up to date.

**3. Business Intelligence**: BigQuery is ideal for running business intelligence queries on large datasets. I can use the data loaded from GCS to generate reports, dashboards, and other insights.

**4. Machine Learning**: Once the data is in BigQuery, I can use it for feature extraction and even run machine learning models directly within BigQuery using BigQuery ML.

**5. Real-time Decision Making**: By continuously loading data from GCS into BigQuery, I can enable real-time analytics, which helps with faster decision-making for businesses.

**Conclusion**

In this guide, I’ve demonstrated how to load data from Google Cloud Storage (GCS) into BigQuery using Git Bash. By following these steps, I can easily move large datasets into BigQuery for advanced data analysis, reporting, and business intelligence. This process is highly beneficial for automating ETL pipelines and leveraging BigQuery's powerful analytics capabilities for real-time insights.