> > > ***PROFESSIONAL DATA ENGINEER** - Google Cloud Platform*

> *TITLE: "Introduction to Google Cloud - Cloud BigTable - HBase shell"*
>
> *Author:*
>
> - *Name: "Vignesh Sekar S"*
> - *Designation: "Multi Cloud Architect"*
> - *Tags: [Google Cloud, DataEngineer, Python, PySpark, SQL, BigData]*

Create an instance and write data with the HBase shell

This lab explains how to use the HBase shell to connect to a Cloud Bigtable instance, perform basic administrative tasks, and read and write data in a table.

Task 1. Create a Bigtable instance

- Open the Create Instance page in the Google Cloud console.
- For Instance name, enter demo-hbaseshell-instance.
- For Instance ID, enter demo-hbaseshell-instance.
- For Storage type, select SSD.
- For Cluster ID, enter demo-hbaseshell-instance-c1.
- For Region, select us-east1.
- For Zone, select us-east1-c.
- Click Create to create the instance.

Task 2. Connect to your instance

- Open a terminal window in Cloud Shell.
  - git clone https://github.com/GoogleCloudPlatform/cloud-bigtable-examples.git (https://github.com/GoogleCloudPlatform/cloud-bigtable-examples.git)
  - cd cloud-bigtable-examples/quickstart
- Scroll down to line 125 and comment out the **gcloud -q components install beta bigtable** command. That line should now resemble the following:
  - sudo apt-get install google-cloud-sdk google-cloud-sdk-bigtable-emulator
- To use the HBase shell with the Cloud Bigtable HBase client for Java, you must install a Java 8 runtime environment. Other versions of Java are not supported. Install and set up the Java 8 environment by running the

following commands:
  - sudo apt-get update
  - sudo apt-get install openjdk-8-jdk-headless
  - export JAVA_HOME=$(update-alternatives --list java | tail -1 | sed -E 's//bin/java//')
- Refer the code quickstart.sh
  - ./quickstart.sh

Task 3. Read and write data

- Bigtable stores data in tables, which contain rows. Each row is identified by a row key.
- Data in a row is organized into column families, which are groups of columns. A column qualifier identifies a single column within a column family.
- There can be multiple time-stamped cells at the intersection of a row and column.
- Create a table named my-table, with one column family named cf1:
  - create 'my-table', 'cf1'
- List your tables:
  - list
- Put the values test-value1 and test-value2 in the row r1, using the column family cf1 and the column qualifier c1:
  - put 'my-table', 'r1', 'cf1:c1', 'test-value1'
  - put 'my-table', 'r1', 'cf1:c1', 'test-value2'
- Use the scan command to scan the table and read the latest two versions of the data you added:
  - scan 'my-table', {VERSIONS => 2}
- Delete the table my-table:
  - disable 'my-table'
  - drop 'my-table'
- Type exit and press Enter to exit the HBase shell. You will see a series of log messages after you exit, which is normal.

Task 4. Clean up

- Open the list of Bigtable instances in the Google Cloud console.
- Click Quickstart instance.
- Click Delete instance.

- Type **instance-name**, then click Delete to delete the instance