Bigtable introduction - Command Line

Cloud Bigtable is Google's NoSQL Big Data database service. It's the same database that powers many core Google services, including Search, Analytics, Maps, and Gmail. Bigtable is designed to handle massive workloads at consistent low latency and high throughput, so it's a great choice for both operational and analytical applications, including IoT, user analytics, and financial data analysis.

In this lab you'll learn how to use the `cbt` command line to connect to a Cloud Bigtable instance, perform basic administrative tasks, and read and write data in a table.

Task 1. Create a Cloud Bigtable instance

1. In the Cloud Console, go to **Navigation menu** menu, click on **Bigtable** in the Databases section, then click **Create instance**.
2. Fill in the fields for your instance as follows:

Field	Value
Instance name:	quickstart-instance
Instance ID:	quickstart-instance
Storage type:	SSD
Cluster ID:	quickstart-instance-c1
Region:	us-east1
Zone:	us-east1-c

Create an instance

A Cloud Bigtable instance is a container for your clusters. Learn more

1 Name your instance

Instance name *

quickstart-instance

For display purposes only

Instance ID *

quickstart-instance

ID is permanent

CONTINUE

A Cloud Bigtable instance is a container for your clusters. Learn more

Name your instance

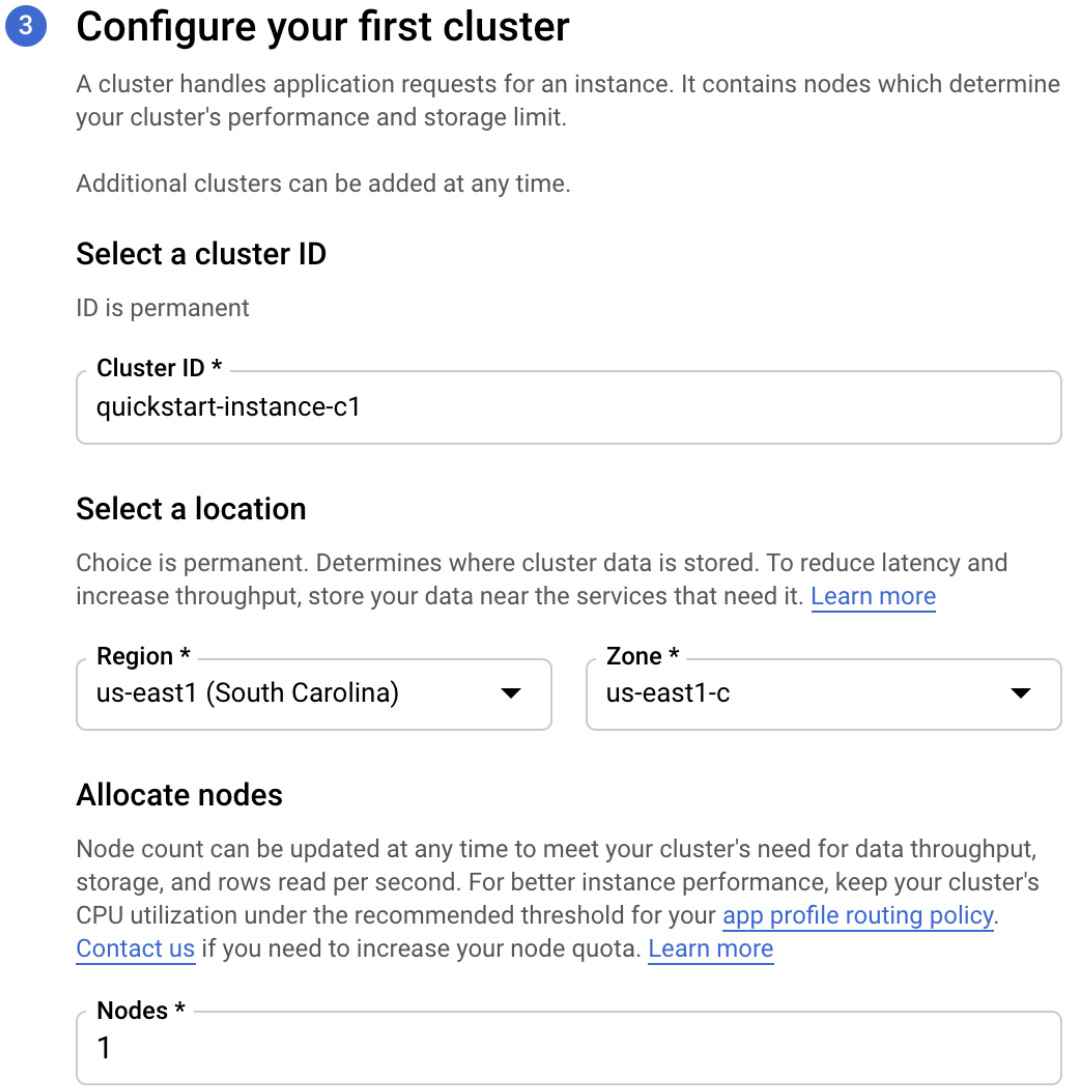
2 Select your storage type

Choice is permanent. Applies to all clusters. Affects cost.

- ○ HDD
  Higher latency for random reads. Good performance on scans and typically used for batch analytics, such as machine learning or data mining.

- ◉ SSD
  Lower latency and more rows read per second. Typically used for real-time serving use cases, such as ad serving and mobile app recommendations.

CONTINUE



3. Click **Create** to create the instance.

Task 2. Connect to your instance

1. In Cloud Shell, configure `cbt` to use your project and instance by modifying the `.cbtrc` file:

```
echo project = `gcloud config get-value project` > ~/.cbtrc
echo instance = quickstart-instance >> ~/.cbtrc
```

Now you're ready to use the `cbt` command.

Task 3. Read and Write Data

Cloud 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*, or groups of columns. A *column qualifier* identifies a single column within a column family.

A *cell* is the intersection of a row and a column. Each cell can contain multiple *versions* of a value.

1. Create a table named `my-table`.

```
cbt createtable my-table
```

2. List your tables:

```
cbt ls
```

The command displays output similar to the following:

```
my-table
```

3. Add one column family named `cf1`:

```
cbt createfamily my-table cf1
```

4. List your column families:

```
cbt ls my-table
```

5. Put the value `test-value` in the row `r1`, using the column family `cf1` and the column qualifier `c1`:

```
cbt set my-table r1 cf1:c1=test-value
```

6. Use the `cbt read` command to read the data you added to the table:

```
cbt read my-table
```

7. Delete the table my-table:

```
cbt deletetable my-table
```

You have now used the `cbt` command line to access Bigtable.