

[Cloud Bigtable](https://cloud.google.com/bigtable/) (<https://cloud.google.com/bigtable/>) > [Documentation](https://cloud.google.com/bigtable/docs/) (<https://cloud.google.com/bigtable/docs/>)

Installing the HBase Shell for Cloud Bigtable

The HBase shell is a command-line tool that performs administrative tasks, such as creating and deleting tables. The [Cloud Bigtable](https://cloud.google.com/bigtable/) (<https://cloud.google.com/bigtable/>) HBase client for Java makes it possible to use the HBase shell to connect to Cloud Bigtable.

This page describes how to install the HBase shell, along with the Cloud Bigtable HBase client for Java, on a Compute Engine instance or on your own machine. You do not need to install any other HBase dependencies, such as Apache ZooKeeper. The Cloud Bigtable HBase client for Java does not require these additional dependencies.

Note: If you want to set up a cluster of Compute Engine instances that can run Hadoop map/reduce jobs, see "[Creating a Hadoop Cluster for Cloud Bigtable](https://cloud.google.com/bigtable/docs/creating-hadoop-cluster)" (<https://cloud.google.com/bigtable/docs/creating-hadoop-cluster>)" for instructions.

Before you start

Before you start, you'll need to complete the following tasks:

- [Create a Cloud Bigtable instance](https://cloud.google.com/bigtable/docs/creating-instance) (<https://cloud.google.com/bigtable/docs/creating-instance>). Be sure to note the project ID and instance ID.
- Install the `gcloud` command-line tool. See the [Cloud SDK setup instructions](https://cloud.google.com/bigtable/docs/installing-cloud-sdk) (<https://cloud.google.com/bigtable/docs/installing-cloud-sdk>) for details.
- If you're installing the HBase shell on a Compute Engine instance, [create an instance](https://cloud.google.com/bigtable/docs/creating-compute-instance) (<https://cloud.google.com/bigtable/docs/creating-compute-instance>) that has the correct scopes for Cloud Bigtable.

Installing Java

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.

If you're installing the HBase shell on a Debian or Ubuntu system, you can install the Java 8 JRE by running the following commands:

```
sudo apt-get update
sudo apt-get install openjdk-8-jre-headless
```

For other operating systems, you can [download and install the Java JRE](http://www.oracle.com/technetwork/java/javase/downloads/) (<http://www.oracle.com/technetwork/java/javase/downloads/>).

Obtaining credentials

When you install the Cloud Bigtable HBase client for Java on your own machine, you need to obtain user access credentials for your Google Cloud Platform resources.

Note: If you're installing the client on a Compute Engine instance, you don't need to obtain credentials.

To obtain credentials, run the following command:

```
gcloud auth application-default login
```

The command opens your web browser. Choose your Google account if prompted, then click **Allow** to authorize access to your account.

Setting up the HBase shell

Downloading required files

1. From the command line, download Apache HBase to any directory:

```
curl -f -O http://storage.googleapis.com/cloud-bigtable/hbase-dist/hbase-1.2.1/hbase-1.2.1-bin.tar.gz
tar xvf hbase-1.2.1-bin.tar.gz
```

2. Create a directory for the Cloud Bigtable HBase client for Java, then download the Cloud Bigtable JAR file to the directory:

```
mkdir -p hbase-1.2.1/lib/bigtable
curl http://repo1.maven.org/maven2/com/google/cloud/bigtable/bigtable-hbase-1.2/0.9.5.1/bigtable-hbase-1.2-0.9.5.1.jar \
-f -o hbase-1.2.1/lib/bigtable/bigtable-hbase-1.2-0.9.5.1.jar
```

★ **Note:** These instructions assume that you are using HBase 1.2.x.

- If you are using HBase 1.0.x, download the JAR file from: <http://repo1.maven.org/maven2/com/google/cloud/bigtable/bigtable-hbase-1.0/0.9.5.1/bigtable-hbase-1.0-0.9.5.1.jar>
- If you are using HBase 1.1.x, download the JAR file from: <http://repo1.maven.org/maven2/com/google/cloud/bigtable/bigtable-hbase-1.1/0.9.5.1/bigtable-hbase-1.1-0.9.5.1.jar>

3. Download the dependencies for the Cloud Bigtable HBase client for Java:

```
curl http://repo1.maven.org/maven2/io/netty/netty-tcnative-boringssl-static/1.1.33.Fork19/netty-tcnative-boringssl-static-1.1.33.Fork19.jar \
-f -o hbase-1.2.1/lib/netty-tcnative-boringssl-static-1.1.33.Fork19.jar
```

Editing configuration files

1. Open the file `hbase-1.2.1/conf/hbase-site.xml` in a text editor. Add properties to the `<configuration>` element as shown below, replacing [VALUES_IN_BRACKETS] with the correct values for your cluster:

```
<configuration>
  <property>
    <name>hbase.client.connection.impl</name>
    <value>com.google.cloud.bigtable.hbase1_2.BigtableConnection</value>
  </property>
  <property>
    <name>google.bigtable.instance.id</name>
    <value>[YOUR-INSTANCE-ID]</value>
  </property>
  <property>
    <name>google.bigtable.project.id</name>
    <value>[YOUR-PROJECT-ID]</value>
  </property>
</configuration>
```

★ **Note:** These instructions assume that you are using HBase 1.2.x.

- If you are using HBase 1.0.x, set `hbase.client.connection.impl` to `com.google.cloud.bigtable.hbase1_0.BigtableConnection`.
- If you are using HBase 1.1.x, set `hbase.client.connection.impl` to `com.google.cloud.bigtable.hbase1_1.BigtableConnection`.

2. Open the file `hbase-1.2.1/conf/hbase-env.sh` in a text editor. Add the following lines to the end of the file:

```
export HBASE_CLASSPATH="$(pwd)/lib/bigtable/bigtable-hbase-1.2-0.9.5.1.jar:\
$(pwd)/lib/netty-tcnative-boringssl-static-1.1.33.Fork19.jar"
export HBASE_OPTS="{HBASE_OPTS} -Xms1024m -Xmx2048m"
```

Setting the JAVA_HOME environment variable

Your shell's `JAVA_HOME` environment variable should be set to the location where your Java runtime environment is located.

To find the current value of `JAVA_HOME`:

```
echo ${JAVA_HOME}
```

If this command does not print any output, the variable is not set.

To set the value on Debian or Ubuntu:

```
export JAVA_HOME=$(update-alternatives --list java | tail -1 | sed -E 's/\/bin\/java//')
```

A typical value on Debian or Ubuntu is `/usr/lib/jvm/java-8-openjdk-amd64/jre`.

To set the value on OS X (requires Xcode's command-line tools):

```
export JAVA_HOME=$(/usr/libexec/java_home)
```

A typical value on OS X is `/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home`.

After you set the value of `JAVA_HOME`, run the following command to ensure that the value refers to the appropriate Java version:

```
${JAVA_HOME}/bin/java -version
```

Testing your configuration settings

The Cloud Bigtable client includes a utility that tests your configuration settings and displays an error if they are incorrect. Running this utility makes it much easier to detect issues with your settings.

To test your configuration, change to the `hbase-1.2.1` directory, then run the utility:

```
cd hbase-1.2.1
./bin/hbase com.google.cloud.bigtable.hbase.CheckConfig
```

The utility attempts to connect to your Cloud Bigtable instance. If successful, the utility displays a list of tables in the instance's cluster. If the utility cannot connect, it displays a stack trace that identifies the cause of the error. Use the information from the utility to edit your `hbase-site.xml` file as necessary.

Now that you've verified your configuration settings, you're ready to use the HBase shell to connect to your Cloud Bigtable cluster.

Connecting to a Cloud Bigtable instance

You can use the HBase shell to connect to your Cloud Bigtable instance, get information about your existing tables, and create new tables. See "[Managing Tables](https://cloud.google.com/bigtable/docs/managing-tables)" (<https://cloud.google.com/bigtable/docs/managing-tables>) for more information about commonly used commands.

To start the HBase shell, run the following command from the `hbase-1.2.1` directory:

```
./bin/hbase shell
```

The HBase shell displays a warning, which is normal. The shell prompt then appears:

```
hbase(main):001:0>
```

If you get an error message that begins with `NativeException`, and the shell exits immediately, double-check all of the values in your `hbase-site.xml` and `hbase-env.sh` files.

Now that you've started the HBase shell, try running the following commands:

- To list your tables, type `list` and press Enter. If you haven't created any tables yet, the HBase shell displays a message similar to the following:

```
TABLE
0 row(s) in 1.3580 seconds

=> []
```

- To create a table, use the `create` command. For example, to create a table named `my-table`, with one column family named `cf1`:

```
create 'my-table', 'cf1'
```

The shell displays a message similar to the following:

```
0 row(s) in 1.5210 seconds
```

```
=> Hbase::Table - my-table
```

- To add data to a table, use the `put` command. For example, to put the value `test-value` in the row `r1`, using the column family `cf1` and the column qualifier `c1`:

```
put 'my-table', 'r1', 'cf1:c1', 'test-value'
```

You can then use the `scan` command to scan the table and read the data you added:

```
scan 'my-table'
```

The shell displays output similar to the following:

```
ROW      COLUMN+CELL
 r1      column=cf1:c1, timestamp=1430687836046, value=test-value
1 row(s) in 0.6260 seconds
```

- To drop a table, use the `disable` command to disable the table, followed by the `drop` command. For example, to drop the table `my-table`:

```
disable 'my-table'
drop 'my-table'
```

- To exit the HBase shell, type `exit` and press Enter.

While the HBase shell is running, you can type `help` to get usage instructions. See the [HBase documentation](http://hbase.apache.org/book.html#shell) (<http://hbase.apache.org/book.html#shell>) for additional information.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated February 14, 2017.