**Creating an HBase table using Java API on CDH EC2 instance**

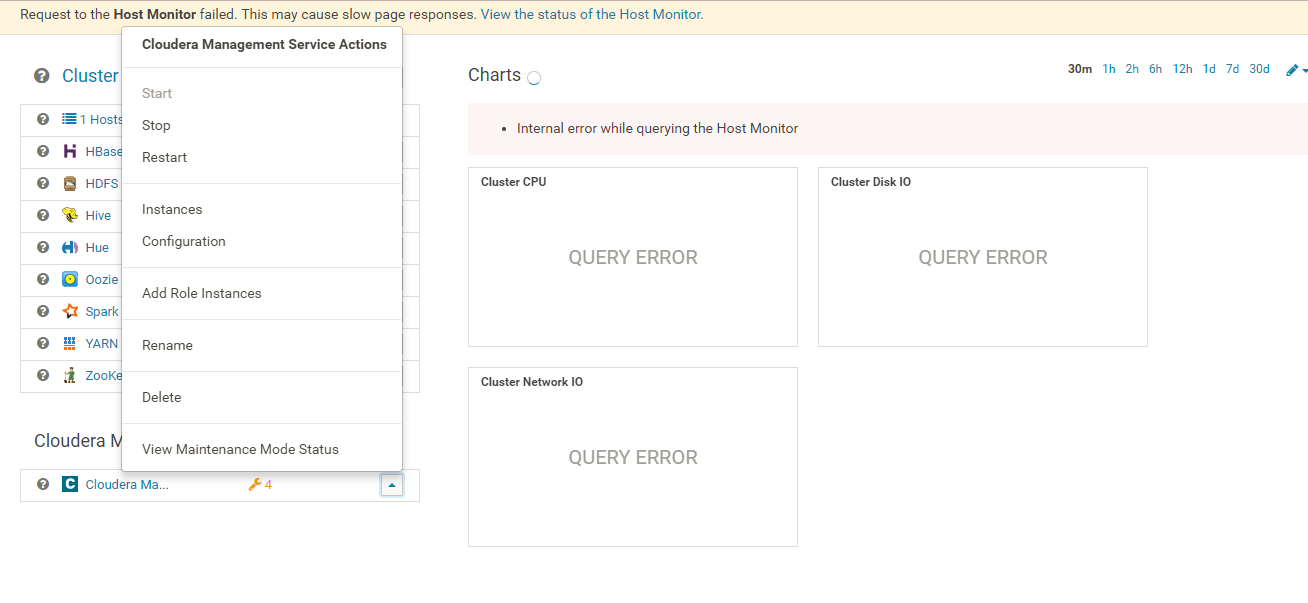
1. After starting the CDH instance from EC2 dashboard, login to Cloudera manager from your browser and restart the Cloudera Management Services.

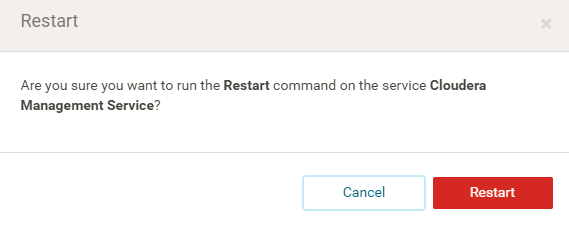
Use the url: [http://**<Your instance’s public ip>**:7180](#)

Username:**admin**

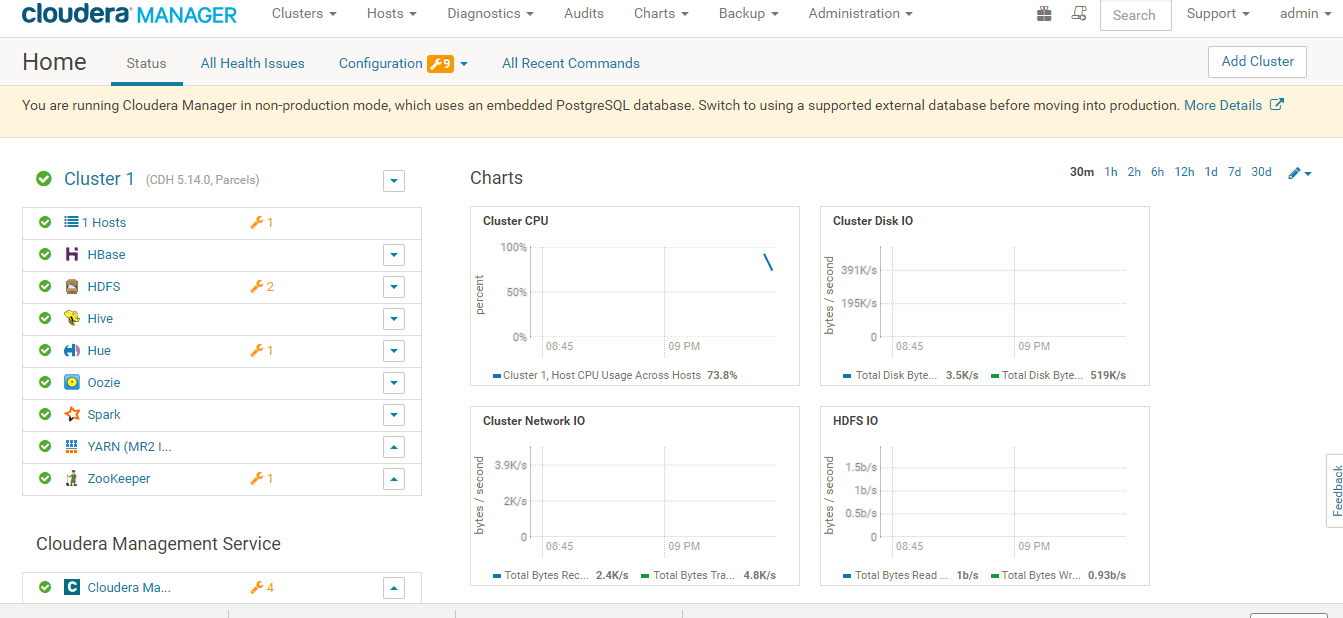
Password:**admin**

1. Click on **‘Restart’** to restart the cloudera management services.





1. Wait for few minutes and click on **close** and wait until services is green.



1. Connect to the EC2 instance using PuTTY. Once connected, we log in as an ec2- user. We now need to switch to the root user using the **sudo -i** command

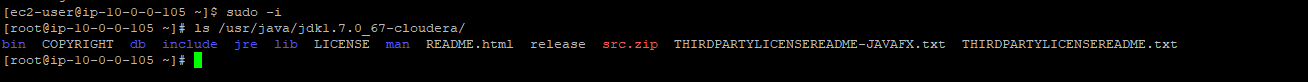
[ec2-user@ip-10-0-0-105 ~]$ **sudo -i**



1. To check whether Java is available or not, do the following:

* run the following command

[root@ip-10-0-0-105 ~]# **ls /usr/java/jdk1.7.0\_67-cloudera/**



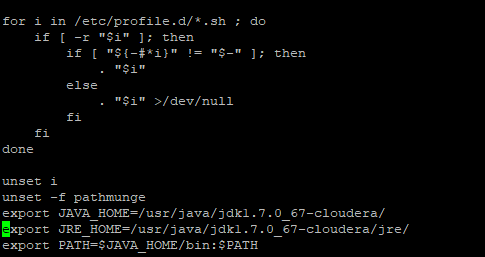
* Set the JAVA\_HOME and JRE\_HOME in the /etc/profile location as follows:

1. Open the file using the command **vi /etc/profile**
2. Add the following at the end of the file as shown below. Please change to the insert mode by pressing i​ before pasting the following lines.

**export JAVA\_HOME=/usr/java/jdk1.7.0\_67-cloudera/**

**export JRE\_HOME=/usr/java/jdk1.7.0\_67-cloudera/jre/**

**export PATH=$JAVA\_HOME/bin:$PATH**



1. Now, save and exit the file. It is important to exit from the insert mode and enter the following in the command mode while using the vi editor:

**:wq!**



* Now run the following commands as shown below:

1. [root@ip-10-0-0-105 ~]# **source /etc/profile**



1. [root@ip-10-0-0-105 ~]# **echo $JAVA\_HOME**

*/usr/java/jdk1.7.0\_67-cloudera/*



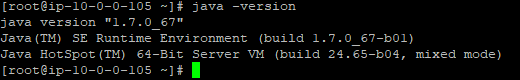
1. [root@ip-10-0-0-105 ~]# **java -version**

*java version "1.7.0\_67"*

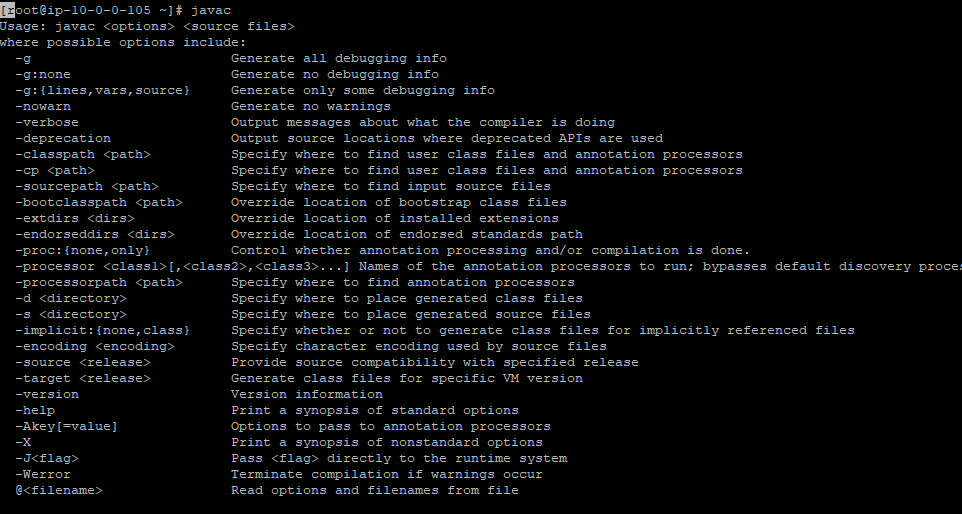
*Java(TM) SE Runtime Environment (build 1.7.0\_67-b01)*

*Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)*

*[root@ip-10-0-0-105 ~]#*



1. [root@ip-10-0-0-105 ~]# **javac**



1. [root@ip-10-0-0-105 ~]# **javac -version**

*javac 1.7.0\_67*



1. Create one directory name as HBase using mkdir command and enter to Hbase directory.

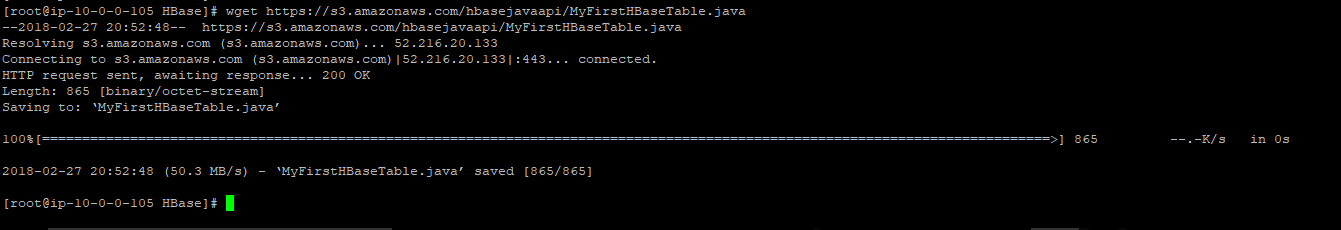
[root@ip-10-0-0-105 ~]# **mkdir HBase**

[root@ip-10-0-0-105 ~]# **cd HBase/**



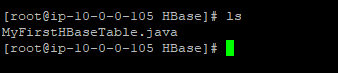
1. Copy “MyFirstHBaseTable.java” program from the provided s3 link to the CDH Instance on AWS ec2 using command wget inside the HBase directory.

[root@ip-10-0-0-105 HBase]# **wget** [**https://s3.amazonaws.com/hbasejavaapi/MyFirstHBaseTable.java**](https://s3.amazonaws.com/hbasejavaapi/MyFirstHBaseTable.java)



1. Verify with ls command.

[root@ip-10-0-0-105 HBase]# **ls**



1. Compile and execute the above program as shown below-

* Set the environment Variables.

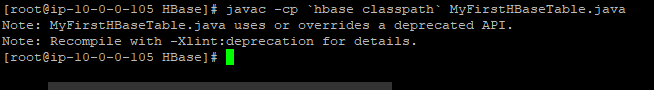
[root@ip-10-0-0-105 HBase]# **export HADOOP\_CLASSPATH=$HADOOP\_PATH:`hbase classpath`**



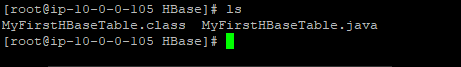
* [root@ip-10-0-0-105 HBase]# **javac -cp `hbase classpath` MyFirstHBaseTable.java**

*Note: MyFirstHBaseTable.java uses or overrides a deprecated API.*

*Note: Recompile with -Xlint:deprecation for details.*



* Verify if the class file is created or not by entering the **ls** command.



* Now execute:-

[root@ip-10-0-0-105 HBase]# **java -cp `hbase classpath` MyFirstHBaseTable**



* The following should be the output:

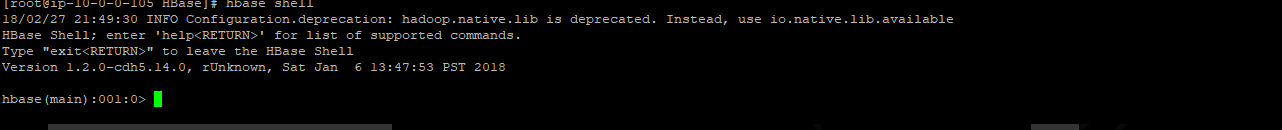


*Table created*

**Checking with HBase Shell**

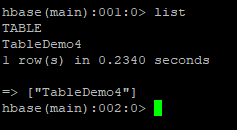
We can check our hbase table using following HBase shell commands:

1. [root@ip-10-0-0-105 HBase]# **hbase shell**



1. Run list to see the table.

hbase(main):001:0> **list**



1. hbase(main):002:0> **exit**

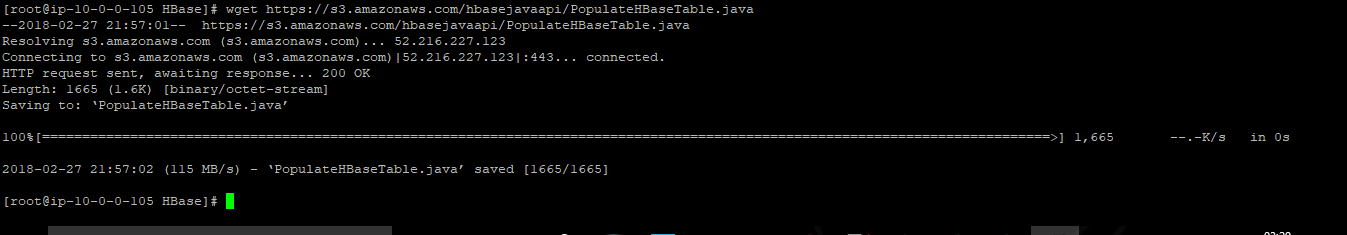


**Inserting Data in the table Using Java API**

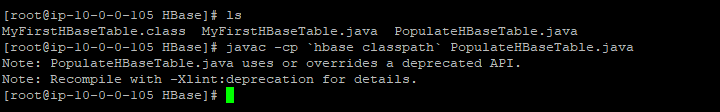
Now we created TableDemo4 table in Hbase with columns ‘Id’ and ‘Name’. You can insert data into Hbase using the add() method of the Put class. You can save it using the put() method of the HTable class. The following code inserts data into the TableDemo4 table:

1. Download “PopulateHBaseTable.java “ from the s3 link using the wget command and place it inside the Hbase directory.

[root@ip-10-0-0-105 HBase]# **wget** [**https://s3.amazonaws.com/hbasejavaapi/PopulateHBaseTable.java**](https://s3.amazonaws.com/hbasejavaapi/PopulateHBaseTable.java)



1. [root@ip-10-0-0-105 HBase]# **javac -cp `hbase classpath` PopulateHBaseTable.java**



1. [root@ip-10-0-0-105 HBase]# **ls**

*MyFirstHBaseTable.class MyFirstHBaseTable.java PopulateHBaseTable.class PopulateHBaseTable.java*



1. [root@ip-10-0-0-105 HBase]# **java -cp `hbase classpath` PopulateHBaseTable**



1. Output:



1. Verify-login hbase shell

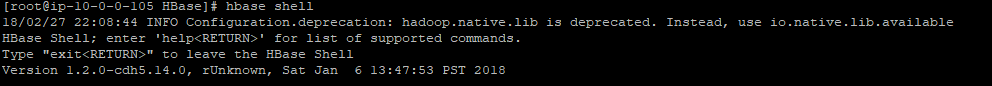
* [root@ip-10-0-0-105 HBase]# **hbase shell**

*18/02/27 22:08:44 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available*

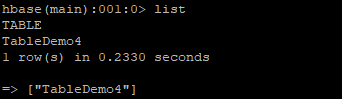
*HBase Shell; enter 'help<RETURN>' for list of supported commands.*

*Type "exit<RETURN>" to leave the HBase Shell*

*Version 1.2.0-cdh5.14.0, rUnknown, Sat Jan 6 13:47:53 PST 2018*



* hbase(main):001:0> **list**



* hbase(main):002:0> **scan "TableDemo4"**

*ROW COLUMN+CELL*

*row1 column=Id:col1, timestamp=1519769114329, value=AAA*

*row1 column=Name:col2, timestamp=1519769114329, value=BBB*

*1 row(s) in 0.1770 seconds*

