

Implement an application that stores big data in HBase using Hadoop.

Expt No: 8

May 27,2019

Author: Subalakshmi Shanthosi S (186001008)

Aim

Implementation of storage in big data analytics using columnar database - HBase using Hadoop.

Description

1. What is HBase?
 - Apache HBase is a column-oriented key/value data store built to run on top of the Hadoop Distributed File System (HDFS).
 - A non-relational (NoSQL) database that runs on top of HDFS.
 - Provides real-time read/write access to those large datasets.
 - Provides random, real time access to your data in Hadoop.
 - Great choice to store multi-structured or sparse data.
 - Low latency storage .
 - Versioned Database.
 - Installation of HBase in standalone mode.
2. Advantages and Disadvantages of clustering methodologies:
 - Advantages:
 - Great for analytics in association with Hadoop MapReduce.
 - It can handle very large volumes of data.
 - Supports scaling out in coordination with Hadoop file system even on commodity hardware.
 - Fault tolerance.
 - License free.
 - Very flexible on schema design/no fixed schema.
 - Auto Sharding.
 - Row-level atomicity, that is, the PUT operation will either write or fail.
 - Disadvantages:
 - Single point of failure (when only one HMaster is used).
 - No transaction support.
 - JOINS are handled in MapReduce layer rather than the database itself.
 - Indexed and sorted only on key, but RDBMS can be indexed on some arbitrary field.
 - No built-in authentication or permissions.

| Column-oriented Database | Row oriented Database |
|---|---|
| When the situation comes to process and analytics we use this approach. Such as Online Analytical Processing and it's applications. | Online Transactional process such as banking and finance domains use this approach. |
| The amount of data that can able to store in this model is very huge like in terms of petabytes | It is designed for a small number of rows and columns. |

Table 1: Column-oriented vs Row-oriented storages

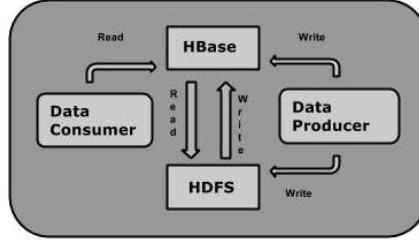


Figure 1: Hadoop Random Access Databases.

Procedure

1. Prepare a Virtual Machine Environment
2. Install Java
3. Install Hadoop
4. Install HBase

HBase Commands

Write about HBase Commands

1. **Create:** create a table in HBase with the specified name given according to the dictionary or specifications as per column family. In addition to this we can also pass some table-scope attributes

create *{tablename}*, *{columnfamilyname}* (1)

2. **Put:** It will put a cell's value at defined or specified table or row or column. It will optionally coordinate time stamp.

put *<' tablename' >*, *<' rowname' >*, *<' columnvalue' >*, *<' value' >* (2)

3. **Scan:** We can pass several optional specifications to this scan command to get more information about the tables present in the system. Scanner specifications may include one or more of the following attributes. These are TIMERANGE, FILTER, TIMESTAMP, LIMIT, MAXLENGTH, COLUMNS, CACHE, STARTROW and STOPROW.

scan *<' tablename' >*, *Optionalparameters* (3)

4. **Get:** You will get a row or cell contents present in the table. In addition to that you can also add additional parameters to it like TIMESTAMP, TIMERANGE, VERSIONS, FILTERS, etc. to get a particular row or cell content.

get *<' tablename' >*, *<' rowname' >*, *< Additionalparameters >* (4)

5. **Disable:** This command will start disabling the named table. If table needs to be deleted or dropped, it has to be disabled first.

disable *< tablename >* (5)

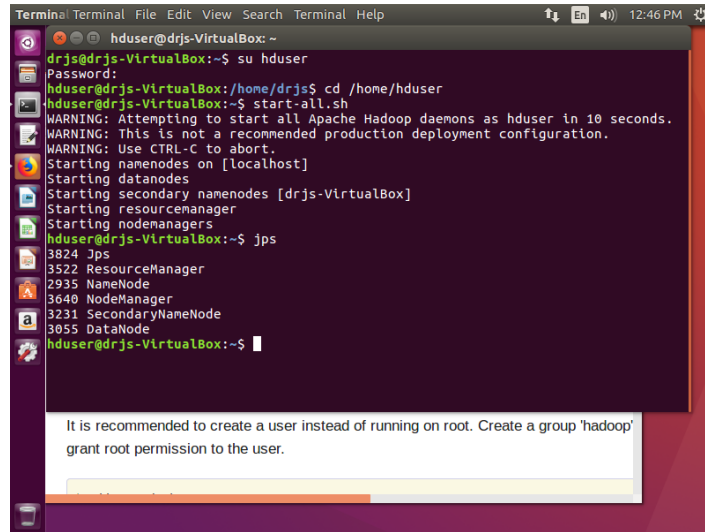
This command will disable all the tables matching the given regex. The implementation is same as delete command (Except adding regex for matching). Once the table gets disabled the user can be able to delete the table from HBase. Before delete or dropping table, it should be disabled first.

disableall *< "matchingregex" >* (6)

6. **Drop:** To delete the table present in HBase, first we have to disable it. To drop the table present in HBase, first we have to disable it. So either table to drop or delete first the table should be disable using disable command. Here in above screenshot we are dropping table "education". Before execution of this command, it is necessary that you disable table "education".

drop < tablename > (7)

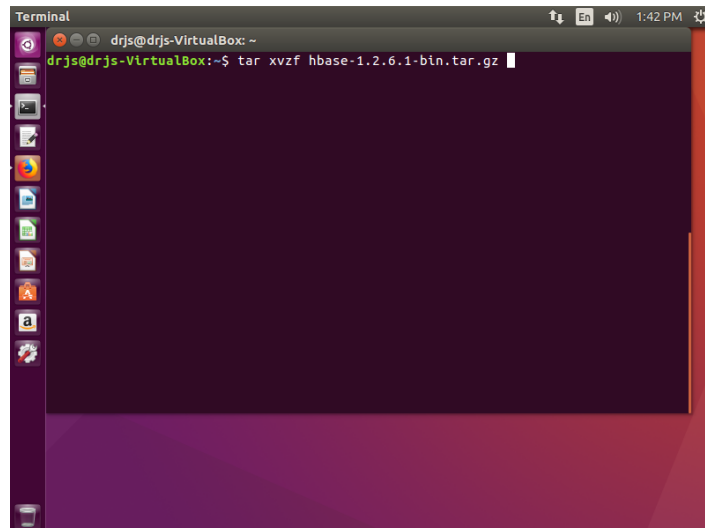
Output



```
Terminal Terminal File Edit View Search Terminal Help
hduser@drjs-VirtualBox: ~
drjs@drjs-VirtualBox:~$ su hduser
Password:
hduser@drjs-VirtualBox:/home/drjs$ cd /home/hduser
hduser@drjs-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [drjs-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hduser@drjs-VirtualBox:~$ jps
3824 Jps
3522 ResourceManager
2935 NameNode
3640 NodeManager
3231 SecondaryNameNode
3055 DataNode
hduser@drjs-VirtualBox:~$
```

It is recommended to create a user instead of running on root. Create a group 'hadoop' grant root permission to the user.

Figure 2: Starting of Hadoop.



```
Terminal
drjs@drjs-VirtualBox: ~
drjs@drjs-VirtualBox:~$ tar xvfz hbase-1.2.6.1-bin.tar.gz
```

Figure 3: Unzipping the hadoop file using tar command.

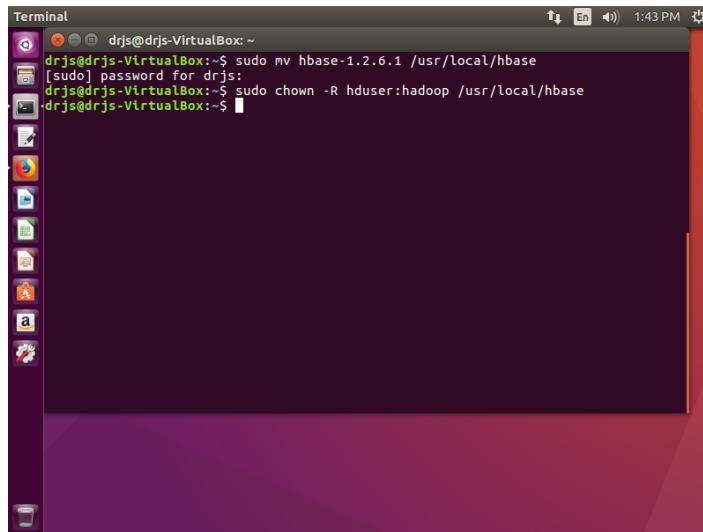


Figure 4: Moving and changing the owner to /usr/local/hadoop.

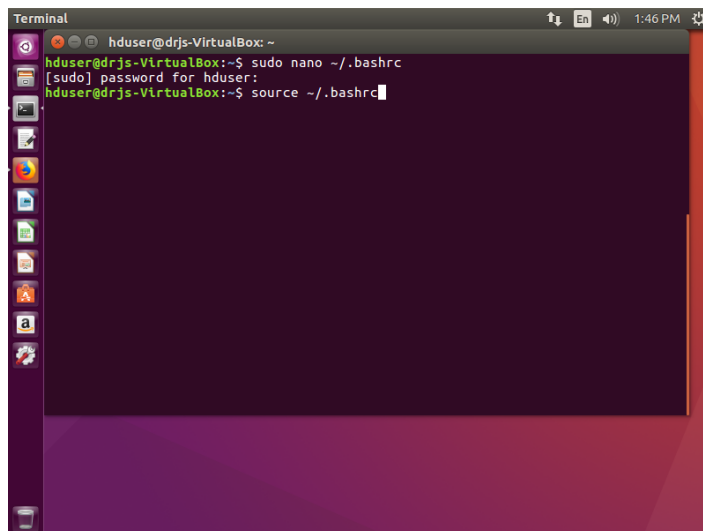


Figure 5: Configuring bashrc.

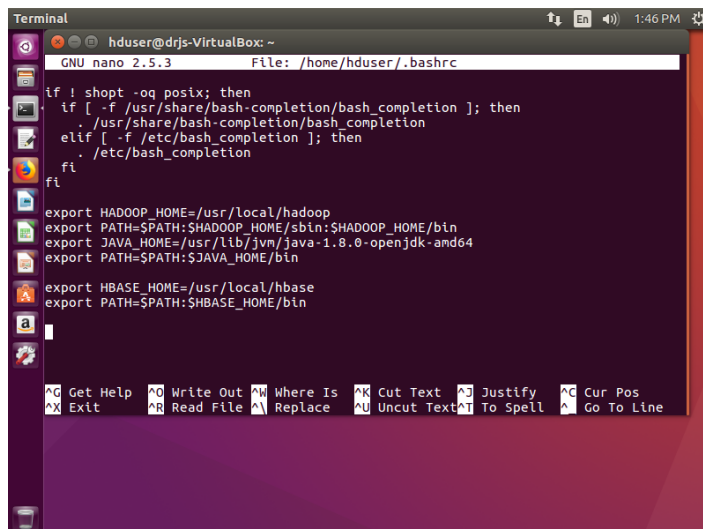


Figure 6: Setting the hbase home and path.

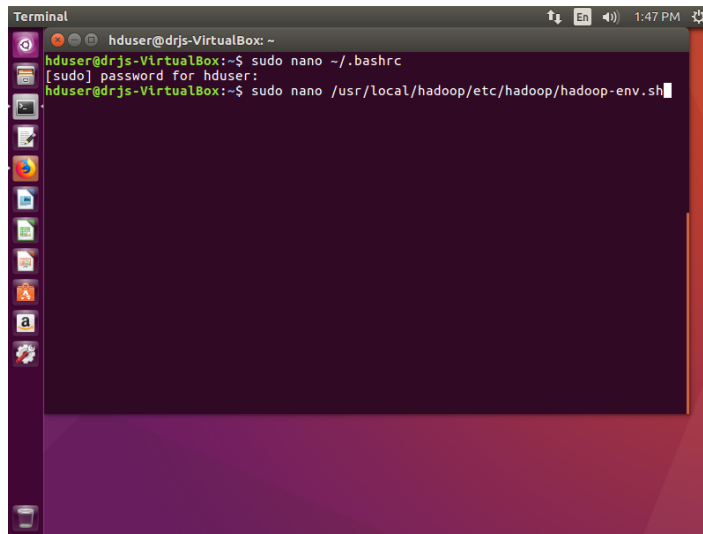


Figure 7: Configuring hadoop-env.sh file.

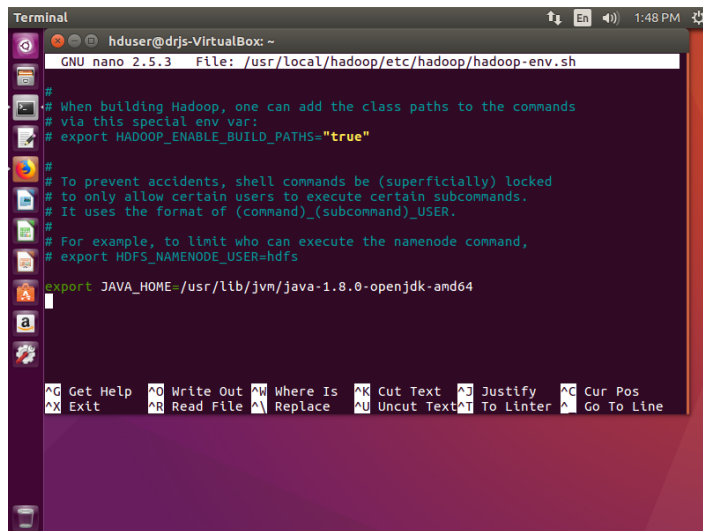


Figure 8: Setting java path.

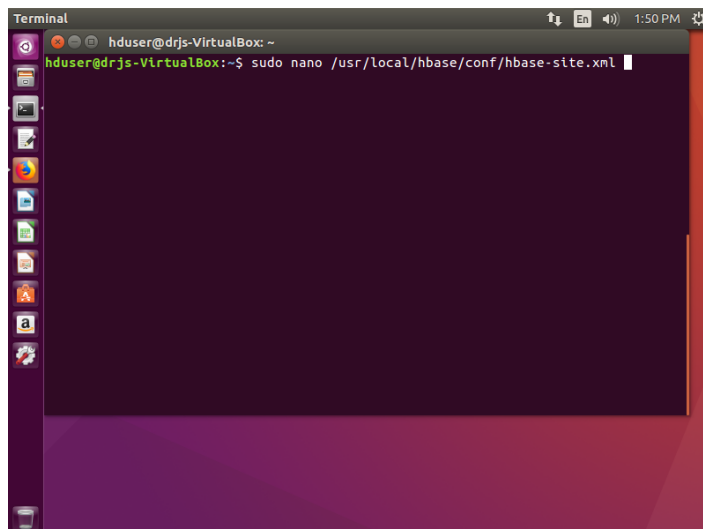


Figure 9: Configuring hbase-site.xml file.

```

Terminal
hduser@drjs-VirtualBox: ~
GNU nano 2.5.3 File: /usr/local/hbase/conf/hbase-site.xml Modified

*/
-->

<configuration>
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:9000/hbase</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/home/hduser/zookeeper</value>
</property>
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
</configuration>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Figure 10: hbase-site.xml.

```

Terminal
hduser@drjs-VirtualBox: ~
hduser@drjs-VirtualBox:~$ start-hbase.sh
localhost: zookeeper running as process 3445. Stop it first.
starting master, logging to /usr/local/hbase/logs/hbase-hduser-master-drjs-Virtu
alBox.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was rem
oved in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was
removed in 8.0
starting regionserver, logging to /usr/local/hbase/logs/hbase-hduser-1-regionser
ver-drjs-VirtualBox.out
hduser@drjs-VirtualBox:~$ jps
2352 SecondaryNameNode
2772 NodeManager
3445 HQuorumPeer
2053 NameNode
7158 HRegionServer
7000 HMaster
2648 ResourceManager
2171 DataNode
7551 Jps
hduser@drjs-VirtualBox:~$

```

Figure 11: Starting HBase.

Master: drjs-VirtualBox - Mozilla Firefox

localhost:16010/master-status 50%

Home Table Details Local Logs Log Level Debug Dump Metrics Dump HBase Configuration

Region Servers

Stop Stop Memory Requests Statistics Comparisons

| ServerName | Start time | Version | Requests Per Second | Num. Regions |
|------------------------------------|------------------------------|---------|---------------------|--------------|
| drjs-virtualbox-16010-155030040408 | Tue May 28 13:31:04 EDT 2019 | 1.2.0.1 | 0 | 0 |
| Total: | | | 0 | 0 |

Backup Masters

| ServerName | Port | Start Time |
|------------|------|------------|
| Total: | | |

Tables

Load Tables System Tables Snapshots

Tasks

Show All Monitored Tasks Show Non-RPC Tasks Show All RPC Handler Tasks Show Active RPC Calls Show Client Operations View as JSON

| Start Time | Description | State | Status |
|------------------------------|---|----------------------------|---|
| Tue May 28 13:31:21 EDT 2019 | Closing region hbase-metall, 1.156230740 | COMPLETE (since 17sec ago) | Closed (since 17sec ago) |
| Tue May 28 13:31:20 EDT 2019 | Initializing region hbase-metall, 1.156230740 | COMPLETE (since 17sec ago) | Region opened successfully (since 17sec ago) |
| Tue May 28 13:31:12 EDT 2019 | Master startup | RUNNING (since 17sec ago) | Assigning hbase-metall region (since 17sec ago) |

Figure 12: Master status.

```
Terminal
hduser@drjs-VirtualBox: ~
hduser@drjs-VirtualBox:~$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6.1, rUnknown, Sun Jun  3 23:19:26 CDT 2018

hbase(main):001:0> list
TABLE
table1
1 row(s) in 0.5110 seconds
=> ["table1"]
hbase(main):002:0> 
```

Figure 13: hbase shell.

```
Terminal
hduser@drjs-VirtualBox: ~
j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6.1, rUnknown, Sun Jun  3 23:19:26 CDT 2018

hbase(main):001:0> list
TABLE
table1
1 row(s) in 0.5110 seconds
=> ["table1"]
hbase(main):002:0> clea
NameError: undefined local variable or method `clea' for #<Object:0x750a04ec>
hbase(main):003:0> clear
NameError: undefined local variable or method `clear' for #<Object:0x750a04ec>
hbase(main):004:0> create 'table2' , 'per' , 'exp'
0 row(s) in 3.0020 seconds
=> Hbase::Table - table2
hbase(main):005:0> 
```

Figure 14: Creation of table.

```
Terminal
hduser@drjs-VirtualBox: ~
hbase(main):003:0> clear
NameError: undefined local variable or method `clear' for #<Object:0x750a04ec>
hbase(main):004:0> create 'table2' , 'per' , 'exp'
0 row(s) in 3.0020 seconds
=> Hbase::Table - table2
hbase(main):005:0> put 'table2' , 'row1' , 'per:name' , 'Dharini'
0 row(s) in 0.3350 seconds
hbase(main):006:0> put 'table2' , 'row1' , 'per:location' , 'Chennai'
0 row(s) in 0.0300 seconds
hbase(main):007:0> put 'table2' , 'row1' , 'per:dept' , 'CSE'
\0 row(s) in 0.0270 seconds
hbase(main):008:0> put 'table2' , 'row1' , 'exp:post' , 'AP'
0 row(s) in 0.0400 seconds
hbase(main):009:0> put 'table2' , 'row1' , 'exp:year' , '2019'
0 row(s) in 0.0300 seconds
hbase(main):010:0> 
```

Figure 15: Inserting rows using "put".

```

hduser@drjs-VirtualBox: ~
hbase(main):030:0> scan 'table2'
COLUMN+CELL
row1      column=exp:post, timestamp=1559032535637, value=AP
row1      column=exp:year, timestamp=1559032556486, value=2019
row1      column=per:dept, timestamp=1559032415701, value=CSE
row1      column=per:location, timestamp=1559032387305, value=Chennai
row1      column=per:name, timestamp=1559032361670, value=Dharini
row2      column=exp:post, timestamp=1559032633630, value=AP
row2      column=exp:year, timestamp=1559032646381, value=2019
row2      column=per:dept, timestamp=1559032619968, value=CSE
row2      column=per:location, timestamp=1559032609123, value=Chennai
row2      column=per:name, timestamp=1559032594245, value=Sarada
row3      column=exp:post, timestamp=1559032713883, value=AP
row3      column=exp:year, timestamp=1559032731407, value=2019
row3      column=per:dept, timestamp=1559032700580, value=CSE
row3      column=per:location, timestamp=1559032691702, value=Chennai
row3      column=per:name, timestamp=1559032677926, value=Salini
row4      column=exp:post, timestamp=1559032806831, value=P
row4      column=exp:year, timestamp=1559032825732, value=2019
row4      column=per:dept, timestamp=1559032793967, value=CSE
row4      column=per:location, timestamp=1559032784113, value=Chennai
row4      column=per:name, timestamp=1559032774890, value=Jafrin
row5      column=exp:post, timestamp=1559032896660, value=P
row5      column=exp:year, timestamp=1559032911127, value=2019
row5      column=per:dept, timestamp=1559032887746, value=CSE
row5      column=per:location, timestamp=1559032877773, value=Chennai
row5      column=per:name, timestamp=1559032869656, value=Preethi

```

Figure 16: Displaying the rows in table using "scan".

```

Terminal
hduser@drjs-VirtualBox: ~
row3      column=per:location, timestamp=1559032691702, value=Chennai
row3      column=per:name, timestamp=1559032677926, value=Salini
row4      column=exp:post, timestamp=1559032806831, value=P
row4      column=exp:year, timestamp=1559032825732, value=2019
row4      column=per:dept, timestamp=1559032793967, value=CSE
row4      column=per:location, timestamp=1559032784113, value=Chennai
row4      column=per:name, timestamp=1559032774890, value=Jafrin
row5      column=exp:post, timestamp=1559032896660, value=P
row5      column=exp:year, timestamp=1559032911127, value=2019
row5      column=per:dept, timestamp=1559032887746, value=CSE
row5      column=per:location, timestamp=1559032877773, value=Chennai
row5      column=per:name, timestamp=1559032869656, value=Preethi
5 row(s) in 0.3270 seconds
hbase(main):031:0> disable 'table2'
0 row(s) in 2.4420 seconds
hbase(main):032:0> drop 'table2'
0 row(s) in 1.3280 seconds
hbase(main):033:0> scan 'table2'

```

Figure 17: Disabling and Dropping the table 2.

1 Result

Thus the implementation of Hbase using hadoop is executed successfully.