Apache HBase 1.2.4 Configuration Guide for Ubuntu 14.04 / 16.04 Single Node Setup

Data Science Lab, The Department of Computer Science, KSKV Kachchh University. Web: http://cs.kutchuni.edu.in

1. wget http://mirror.fibergrid.in/apache/hbase/1.2.5/hbase-1.2.5-bin.tar.gz
2. tar -xvzf hbase-1.2.5-bin.tar.gz
3. sudo mv hbase-1.2.5 /usr/local/hbase
4. Edit hbase-env.sh
    4.1 sudo gedit /usr/local/hbase/conf/hbase-env.sh
    4.2 insert / replace below changes

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HBASE_MANAGES_ZK=true
```

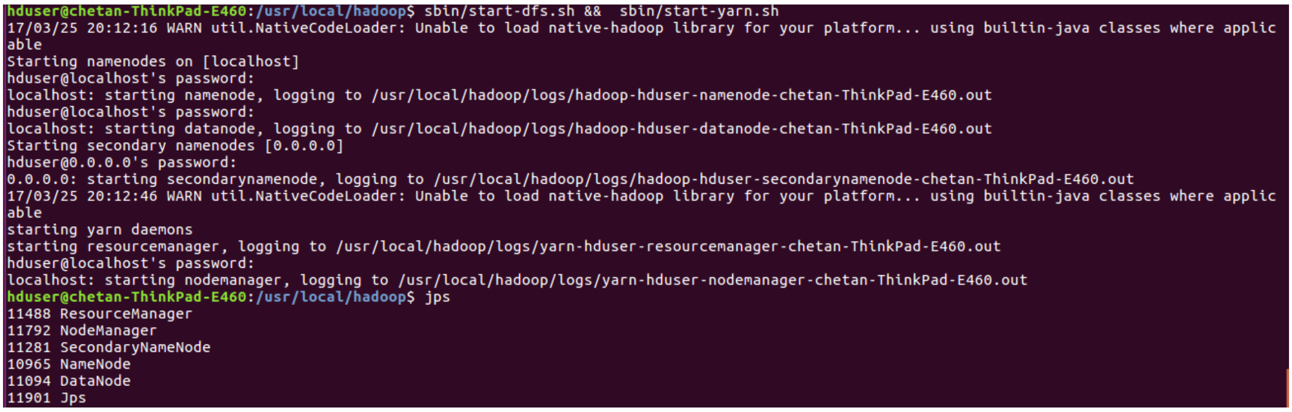
5. Edit hbase-site.xml
    5.1 Create Directory – sudo mkdir /usr/local/zookeeper-data
    5.2 sudo gedit /usr/local/hbase/conf/hbase-site.xml

do changes as below

```
<configuration>
 <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
    <name>hbase.zookeeper.property.clientPort</name>
    <value>2181</value>
  </property>
<property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/usr/local/zookeeper-data</value>
</property>
</configuration>
```

6. Start Hadoop Cluster if not yet started.

```
     $ cd /usr/local/hadoop
     > /usr/local/hadoop$ sbin/start-dfs.sh &&  sbin/start-yarn.sh
```

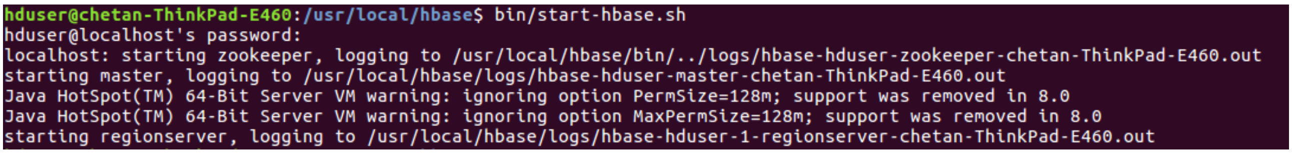


```
hduser@chetan-ThinkPad-E460:/usr/local/hadoop$ sbin/start-dfs.sh &&  sbin/start-yarn.sh
17/03/25 20:12:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-chetan-ThinkPad-E460.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-chetan-ThinkPad-E460.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-chetan-ThinkPad-E460.out
17/03/25 20:12:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applic
able
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-chetan-ThinkPad-E460.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-chetan-ThinkPad-E460.out
hduser@chetan-ThinkPad-E460:/usr/local/hadoop$ jps
11488 ResourceManager
11792 NodeManager
11281 SecondaryNameNode
10965 NameNode
11094 DataNode
11901 Jps
```

[Figure 1]: Starting Apache Hadoop deamon services

7. Start Zookeeper and HBase region-server deamon services.

```
    > /usr/local/hbase$ bin/start-hbase.sh
```



```
hduser@chetan-ThinkPad-E460:/usr/local/hbase$ bin/start-hbase.sh
hduser@localhost's password:
localhost: starting zookeeper, logging to /usr/local/hbase/bin/../logs/hbase-hduser-zookeeper-chetan-ThinkPad-E460.out
starting master, logging to /usr/local/hbase/logs/hbase-hduser-master-chetan-ThinkPad-E460.out
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Java HotSpot(TM) 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-regionserver-chetan-ThinkPad-E460.out
```

[Figure 2]: Starting Zookeeper and HBase Region Server deamon services

8. Start HBase Shell

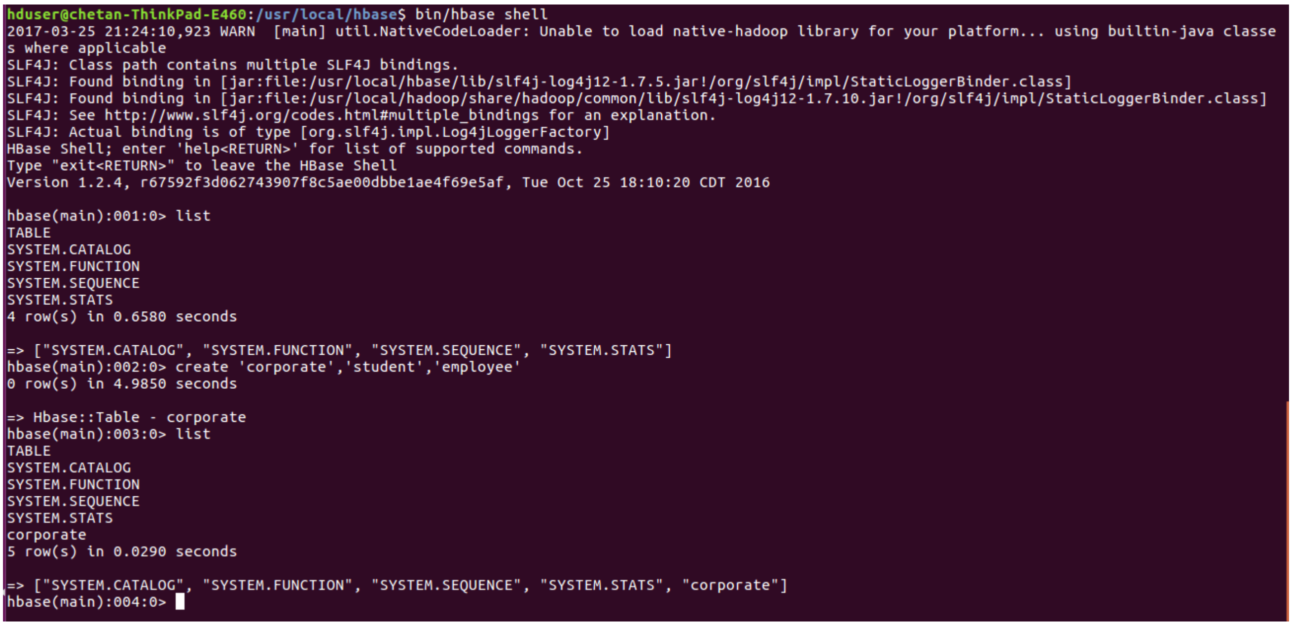
```
    > /usr/local/hbase$ bin/hbase shell
```

9. You can create table with Column family

```
    > create ‘tablename’,’columnfamily1’....’columnfamilyN’
```

example: `> create ‘corporate’,’student’,’employee’`

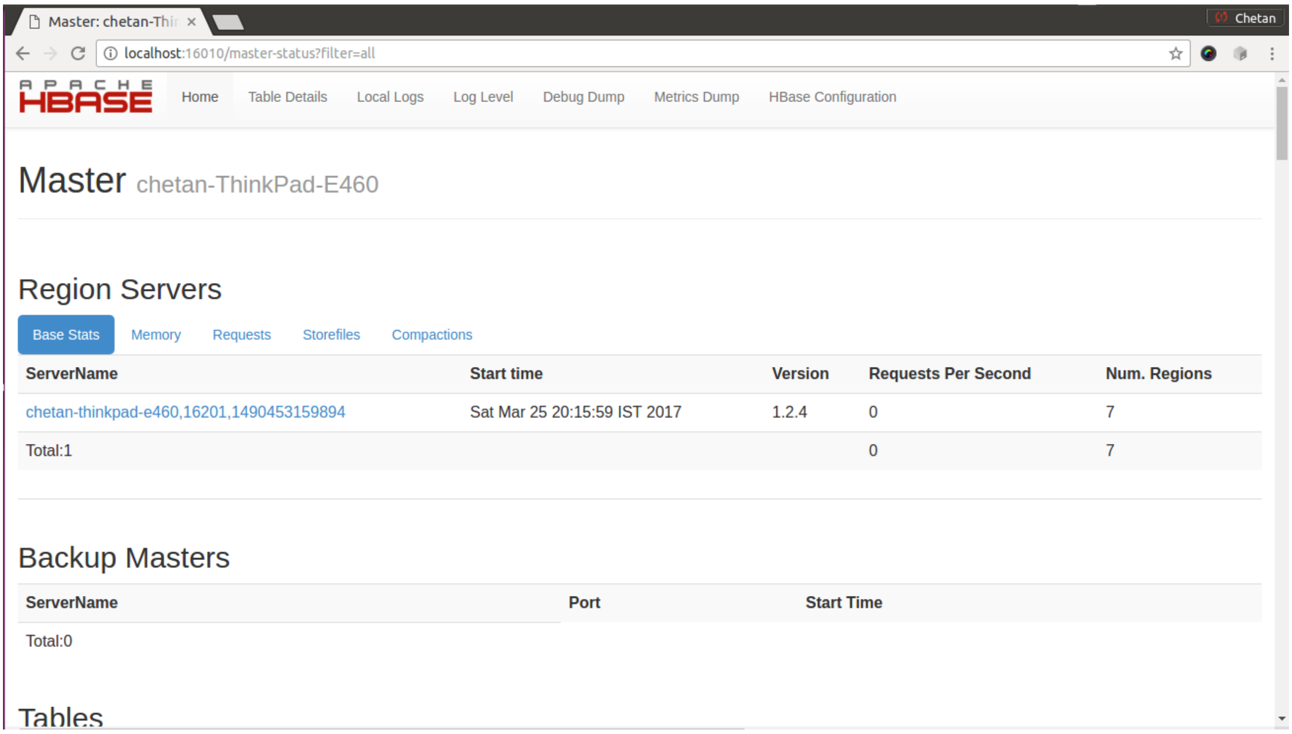
you can list out all the tables by `> list`



```
hduser@chetan-ThinkPad-E460:/usr/local/hbase$ bin/hbase shell
2017-03-25 21:24:10,923 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
s where applicable
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-log4j12-1.7.10.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 1.2.4, r67592f3d062743907f8c5ae00dbbe1ae4f69e5af, Tue Oct 25 18:10:20 CDT 2016

hbase(main):001:0> list
TABLE
SYSTEM.CATALOG
SYSTEM.FUNCTION
SYSTEM.SEQUENCE
SYSTEM.STATS
4 row(s) in 0.6580 seconds

=> ["SYSTEM.CATALOG", "SYSTEM.FUNCTION", "SYSTEM.SEQUENCE", "SYSTEM.STATS"]
hbase(main):002:0> create 'corporate','student','employee'
0 row(s) in 4.9850 seconds

=> Hbase::Table - corporate
hbase(main):003:0> list
TABLE
SYSTEM.CATALOG
SYSTEM.FUNCTION
SYSTEM.SEQUENCE
SYSTEM.STATS
corporate
5 row(s) in 0.0290 seconds

=> ["SYSTEM.CATALOG", "SYSTEM.FUNCTION", "SYSTEM.SEQUENCE", "SYSTEM.STATS", "corporate"]
hbase(main):004:0>
```

[Figure 3]: Starting Zookeeper and HBase Region Server deamon services

10. You can check HBase-UI at http://localhost:16010



Master chetan-ThinkPad-E460

Region Servers

ServerName	Start time	Version	Requests Per Second	Num. Regions
chetan-thinkpad-e460,16201,1490453159894	Sat Mar 25 20:15:59 IST 2017	1.2.4	0	7
Total:1			0	7

Backup Masters

ServerName	Port	Start Time
Total:0		

Tables

[Figure 4]: Apache HBase UI

Apache HBase terminologies explanation:

- Each HMaster uses three ports by default – 16010, 16020, 16030
- to take backup of HMaster run – **local-master-backup.sh**
- ulimit for HBase, By default 1024. you can check at **ulimit -n** command. Because HBase stores data in files knows as HFiles at HDFS and HBase helps to get random access to HDFS, where OS kernel has by default limit on , 1. Number of open files, 2. Number of open TCP connections etc.
  You can change it to 10,000 or most likely 10,240
- Each ColumnFamily has atleast one storefile and possibly more than six storefiles if the region is under load.
- Number of Open files required = Number of column family * Regions per regionserver
- You can check number of process allowed to run on Operating System by command **ulimit -u** if it is too low can cause 'OutOfMemory' exceptions. **nproc** command which controls the number of CPUs are available to a given user.
- You must not consider Pseudo Distribution setup we did for Performance evolution.
- According to Aaron Kimball's Configuration parameters -
  - /etc/security/limits.conf
  - hadoop – nofile 32768
  - hadoop – nproc 32000
  - Threading
  - dfs.datanode.max.transfer.threads – 4096
- You can also set the Heap Memory Size

HBase Data Model:

- Multi-Dimensional Map, Rows are seperated by row keys where design of row keys are important.
- Column Name = Column Family prefix + qualifier
- example, content:html where content is column family and html is qualifier.
- Each column family has a set of properties:
  - Should be cached in memory
  - How data is compressed / keys are encoded.
  - Each row in a table has the same column families, through a given row might not store anything in a given collumn family.
  - Column qualifier adds column family for indexing.

Namespace:

- Namespace is a logical grouping of table analogous to a database in relation database systems.
  - Quota Management
  - Namespace security administration
  - Region Server groups

<table namespace>:<table qualifier>

```
create_namespace ‘pocschema’
```

create my_table in my_ns namespace

```
create ‘my_ns:my_table’, ‘family’
```

Drop namespace

```
drop_namespace ‘my_ns’
```

Alter namespace

```
alter_namespace ‘my_ns’, {METHOD => ‘set’, PROPERTY_NAME’ =>
‘PROPERTY_VALUE’}
```

Predefined Namespace

hbase – System namespace for hbase internal.
default – tables with no explicit specified namespace will automatically fall into this namespace.

Row

Row keys are uninterpreted bytes. Because HBase stores everything in Bytes.

Column Family

- Columns in Apache HBase are grouped into column families, all column members of a column family have the same prefix.
- For example, the columns courses: history and courses:math are both members of the courses column family.
- **:** is delimiter for column family and column qualifier.

Cells

- A {row,column,version} tuple exactly specifies cell in HBase

Sort Order

- All data model operations HBase returns are data in sorted order, first by row then by columnfamily, followed by column qualifier and timestamps sorted in reverse so newest records are returned first.

Operations:

Get:

- Get returns attributes for a specified row
- Get are executed via Table.get

Put:

- Put - either adds new rows to a table (if the key is new) or can update existing rows if the key already exists.

Scans:

- Scan allows iterations over multiple rows for specified attributes.

Delete:

- Delete removes a row from a table.
- Delete – for a specific version of a column
- Delete column – for all versions of a column
- delete family – for all columns of a particular columnfamily

Note: When deleting an entire row, HBase will internally create a tombstone for each columnfamily (not each individual column)

Joins:

- HBase doesn’t supports joins as you do with RDBMS (e.g with equi-joins or outer-joins in SQL).
- Reading Data model in HBase are **get** & **scan** commands

Note:

- Denormalizing the Data Model upon writing to HBase or have loopup tables and do joins in your code using MapReduce.
- Tables must be disabled when making columnfamily modifications.
- You can use HBase shell / HBase Client API to implement operations.

SQL Queries on HBase:

You can use Apache Drill on HBase files to queries on HBase tables.
- Apache Drill - https://drill.apache.org/
- Apache Presto - https://prestodb.io/
- Apache Impala - https://impala.incubator.apache.org/

Read more:

[1] MapR: GUIDELINES FOR HBASE SCHEMA DESIGN
Online: https://mapr.com/blog/guidelines-hbase-schema-design/

[2] Apache HBase Book, Chapter 6 Schema Design
Online: http://hbase.apache.org/0.94/book/schema.html

[3] O’reilly Conference talk - HBase Schema Design - Things you need to know
Online: https://www.youtube.com/watch?v=_HLoH_PgrLk

[4] HBASE SCHEMA DESIGN and Cluster Sizing Notes ApacheCon Europe, November 2012 Lars George Director EMEA Services.
Online: http://archive.apachecon.com/eu2012/presentations/06-Tuesday/L2L-Big_Data/aceu-2012-HBase-sizing-and-schema-design.pdf

References:

[1]. Apache HBase Documentation
Online : https://hbase.apache.org/

[2]. Bigtable: A Distributed Storage System for Structured Data
Online : https://research.google.com/archive/bigtable-osdi06.pdf