HBASE INSTALLATION & WORKOUTS

1) Go to the below path.

cd /home/hduser/install/

2) Extract the tarball,

HBase installation
---
tar xvzf hbase-0.98.4-hadoop2-bin.tar.gz
sudo mv hbase-0.98.4-hadoop2 /usr/local/hbase
sudo chown -R hduser:hadoop /usr/local/hbase

Zookeeper installation
---
tar xvzf zookeeper-3.4.6.tar.gz
sudo mv zookeeper-3.4.6 /usr/local/zookeeper
sudo chown -R hduser:hadoop /usr/local/zookeeper

3) Zookeeper config

cd /usr/local/zookeeper/conf
mv zoo_sample.cfg zoo.cfg

vi zoo.cfg
dataDir=/usr/local/zookeeper/data

Create the below dir for zookeeper

mkdir /usr/local/zookeeper/data

4) Start Zookeeper by running below command:

cd /usr/local/zookeeper/bin
./zkServer.sh start

5) Edit hbase environment script

cd /usr/local/hbase/conf
echo 'export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_71' >> hbase-env.sh
echo 'export HBASE_MANAGES_ZK=false' >> hbase-env.sh

6) Edit the hbase-site.xml to set hbase distribution, hbase root data dir, zookeeper quorum and zk port.

vi /usr/local/hbase/conf/hbase-site.xml
<configuration>
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:54310/user/hduser/hbase</value>
</property>
<property>
<name>hbase.zookeeper.quorum</name>
<value>localhost</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>

This property is to add at the time of configuring phoenix.
<property>
<name>hbase.regionserver.wal.codec</name>
<value>org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec</value>
</property>

7) Once completed the above steps start the hbase daemon

start-hbase.sh

Type jps and see if zookeeper and hbase is running

8) To get into the hbase interactive shell type the below command

hbase shell

9) Type 'list' to see if hbase is working properly

##################################HBASE WORKOUTS##################################

####### Creating a table "Patient" with the column Families (Personal and Medical) #######
create 'Patient1','Personal','Medical'

####### Inserting a record into the table#######
put 'Patient1','001','Personal:pname','Ramesh'
put 'Patient1','002','Personal:pname','saravanan'

```
put 'Patient1','002','Personal:filenum','100'
put 'Patient1','003','Personal:pname','gowtham'
put 'Patient1','004','Personal:pname','amudhan'
put 'Patient1','005','Personal:pname','alex'
put 'Patient1','002','Personal:age','24'
put 'Patient1','105','Personal:pname','alex'
put 'Patient1','202','Personal:age','24'
put 'Patient1','202','Personal:filenum','101'
put 'Patient1','202','Personal:addr','3 first ave,NJ'
put 'Patient1','001','Medical:history','Anemic'
put 'Patient1','105','Medical:history','General check'
put 'Patient1','102','Medical:history','Arthritis'
put 'Patient1','102','Medical:oldhistory','Ostophenia'

#######Check whether the below put works with the column family used as Medical1 instead of
Medical#######

put 'Patient1','102','Medical1:oldhistory','Ostophenia'

####### Scan/select all the data from the table#######
scan 'Patient1'

####### Scan/select with rowkey#######
get 'Patient1','002'

####### Retrieve more versions#######

alter 'Patient1',{NAME=>'Personal',VERSIONS=>1}
put 'Patient1','001','Personal:pname','Ramesh kumar'
scan 'Patient1',{VERSIONS => 3}
alter 'Patient1',{NAME=>'Personal',VERSIONS=>3}
put 'Patient1','001','Personal:pname','Ramesh k'
put 'Patient1','001','Personal:pname','k Ramesh'
scan 'Patient1',{VERSIONS => 3}

#######List only the latest version#######

get 'Patient1','001'
put 'Patient1','001','Personal:pname','RK'
scan 'Patient1',{VERSIONS => 6}

####### delete a specific column from rowkey#######

delete 'Patient1','002','Personal:pname'

####### delete entire rowkey details#######

deleteall 'Patient1','001'

####### Describe the table#######

describe 'Patient1'
```

```
####### Add column family #######
alter 'Patient1',{NAME=>'Medical1'}

####### drop the column family from the table#######

alter 'Patient1',{NAME=>'Medical',METHOD=>'delete'}

####### Execute the hbase commands from file#######

hbase shell /home/hduser/install/hbase_commands.hbase

####### Example of keyonlyfilter
####### This filter does not take any arguments. It returns only the key component of each key-value.#######

scan 'Patient1',{ FILTER => "KeyOnlyFilter()"}

####### FirstKeyOnlyFilter#######
####### This filter does not take any arguments. It returns only the first key-value from each row.#######

scan 'Patient1',{ FILTER => "FirstKeyOnlyFilter()"}

####### prefixfilter: #######
####### This filter takes one argument a prefix of a row key. It returns only those key-values present in
a row that starts with the specified row prefix#######

scan 'Patient1', {FILTER => "(PrefixFilter ('002'))"}

####### ColumnPrefixFilter - This filter takes one argument a column prefix. It returns only those keyvalues
present in a column that starts with the specified column prefix. The column prefix must be of the form
qualifier#######

scan 'Patient1', {FILTER => "(PrefixFilter ('002')) AND ColumnPrefixFilter('a')"}

####### MultipleColumnPrefixFilter - This filter takes a list of column prefixes. It returns key-values that are
present in a column that starts with any of the specified column prefixes. Each of the column prefixes must be of
the form qualifier#######

scan 'Patient1',{FILTER => "MultipleColumnPrefixFilter('p','a')"}

####### InclusiveStopFilter - This filter takes one argument a row key on which to stop scanning. It
returns all key-values present in rows up to and including the specified row. #######

scan 'Patient1',{FILTER => "InclusiveStopFilter('003')"}

####### Selecting columns and introducing limit#######

scan 'Patient1', { COLUMNS => 'Personal:pname', LIMIT => 2}

####### Disable table#######

disable 'Patient1'
```

####### Enable table#######

enable 'Patient1'

####### drop the table. Table should be disabled to drop. #######

drop 'Patient1'

#######HBase Pig integeration#######
Create the below table in Hbase

create 'PigDataHTable','UserColFamily'

#######Sample data to upload from Pig#######

user_01|Arun|30|Chennai
user_02|Bala|40|Madurai
user_03|Devi|50|Trichy
user_04|Faizal|60|Chennai

#######Login to pig in local mode#######

pig -x local

userdata = load '/home/hduser/pigdata/hbdata' using PigStorage('|') as (id:chararray, name:chararray, age:chararray, place:chararray);
store userdata into 'hbase://PigDataHTable' using org.apache.pig.backend.hadoop.hbase.HBaseStorage ('UserColFamily:name,UserColFamily:age,UserColFamily:place');