

Hadoop + Hbase + Phoenix Deployment

Versions:

Ubuntu 16.04

JDK1.8

Hadoop: 2.7.6

Hbase: 1.3.X

Phoenix: 4.14

Currently most stable and compatible combination

Java deployment:

Use apt-get to install Java:

```
sudo apt-get install update
```

```
sudo apt-get update
```

```
sudo apt-get install default-jdk
```

```
java -version
```

 see whether Java has been successfully installed

```
sudo nano /etc /profile
```

 deploy the Java environment variable

put below command in the end of the file:

```
export JAVA_HOME= /usr
```

```
source etc/profile
```

ssh deployment:

To install ssh secret key for users in the system to avoid typing password every time initiate Hadoop and other processes

Example below use Harvey as the user name:

```
sudo apt-get install ssh
```

```
sudo systemctl start ssh
```

```
ssh-keygen -t rsa -P ""
```

 generate secret key files

```
cat /home/Harvey/.ssh/id_rsa.pub >> /home/Harvey/.ssh/authorized_keys
```

```
cd .ssh/
```

```
chmod 600 ./authorized_keys
```

```
ssh-copy-id -i /home/harvey/.ssh/id_rsa.pub localhost
```

```
ssh localhost
```

Hadoop 2.7.6 deployment and configuration:

`sudo wget 'link of downloading Hadoop 2.7.6'`

`sudo tar -xvf 'downloaded Hadoop file'`

`sudo mv hadoop-2.7.6 /usr/local/hadoop` move the unzipped Hadoop file to specified file directory

`sudo chown -R harvey /usr/local` empower to the user Harvey

`nano ~/.bashrc`

configure the environment variable, put below command in the file:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 export
HADOOP_HOME=/usr/local/hadoop export
PATH=$PATH:$HADOOP_HOME/bin export
PATH=$PATH:$HADOOP_HOME/sbin export
HADOOP_MAPRED_HOME=$HADOOP_HOME export
HADOOP_COMMON_HOME=$HADOOP_HOME export
HADOOP_HDFS_HOME=$HADOOP_HOME export
YARN_HOME=$HADOOP_HOME export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native export
HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

`cd /usr/local/Hadoop/etc/hadoop`

`nano hadoop-env.sh`

`export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64`

`nano core-site.xml` paste below command to the <configuration>:

```
<property> <name>fs.default.name</name>
<value>hdfs://localhost:9000</value> </property>
```

`nano hdfs-site.xml` paste below command to the <configuration>:

```
<property> <name>dfs.replication</name> <value>1</value> </property>
<property> <name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value> </property>
<property> <name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value> </property>
```

nano yarn-site.xml paste below command to the <configuration>:

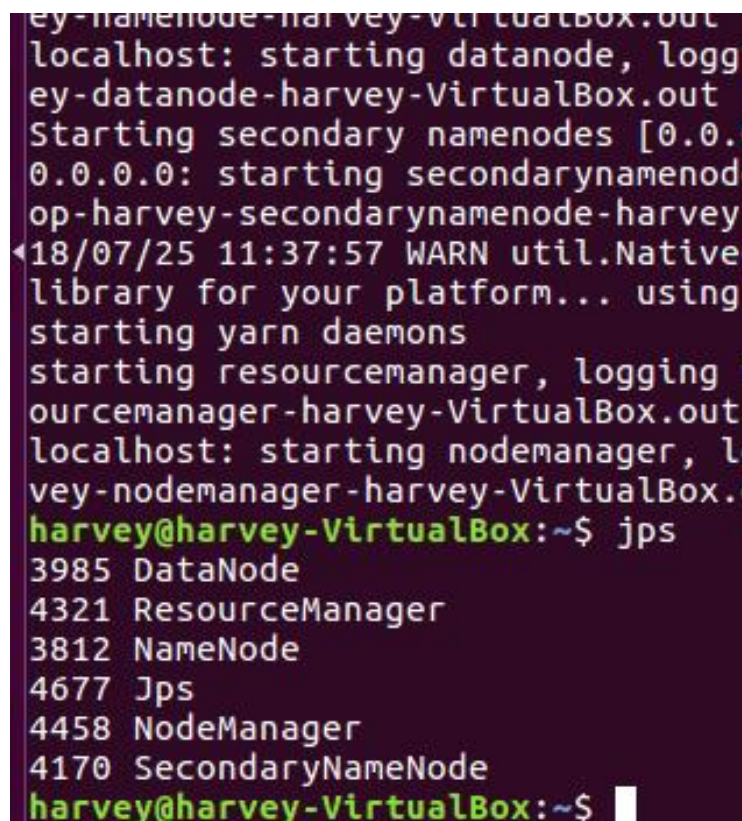
```
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value> </property> <property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value> </property>
```

nano mapred-site.xml.template saved as mapred-site.xml

```
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
```

hdfs namenode -format Hadoop all files configuration has been done, create namenode

_start-all.sh and type **jps**:



```
harvey-namenode-harvey-VirtualBox.out
localhost: starting datanode, logging to harvey-datanode-harvey-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode
op-harvey-secondarynamenode-harvey-VirtualBox.out
18/07/25 11:37:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using generic fallback
starting yarn daemons
starting resourcemanager, logging to harvey-resourcemanager-harvey-VirtualBox.out
localhost: starting nodemanager, logging to harvey-nodemanager-harvey-VirtualBox.out
harvey@harvey-VirtualBox:~$ jps
3985 DataNode
4321 ResourceManager
3812 NameNode
4677 Jps
4458 NodeManager
4170 SecondaryNameNode
harvey@harvey-VirtualBox:~$
```

Hadoop has been deployed successfully

Hbase deployment and configuration

`sudo wget 'link of downloading Hbase 1.3'`

`sudo tar -xvf 'downloaded Hbase file'`

`sudo mv hbase-1.3 /usr/local/hbase` move the unzipped Hadoop file to specified file directory

`nano ~/.bashrc`

configure the environment variable, put below command in the file:

`export HBASE_HOME=/usr/local/hbase`

`export PATH=$PATH:/usr/local/hbase/bin`

`source ~/.bashrc` save the file

`cd /usr/local/hbase`

`mkdir hbasestorage` generate Hbase storage file

`cd conf`

`nano hbase-site.xml` put below command in the file:

```
<property>
<name>hbase.rootdir</name>
<value>/usr/local/hbase/hbasestorage</value>
</property>
<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>
<property>
<name>hbase.zookeeper.quorum</name>
<value>localhost</value>
</property>
<property>
<name>zookeeper.property.dataDir</name>
<value>/usr/local/hbase/hbasestorage/zookeeper</value>
</property>
<property>
<name>hbase.zookeeper.property.clientPort</name>
<value>2181</value>
</property>
</property>
```

```
<name>hbase.regionserver.wal.codec</name>
<value>org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec</value>
</property>
```

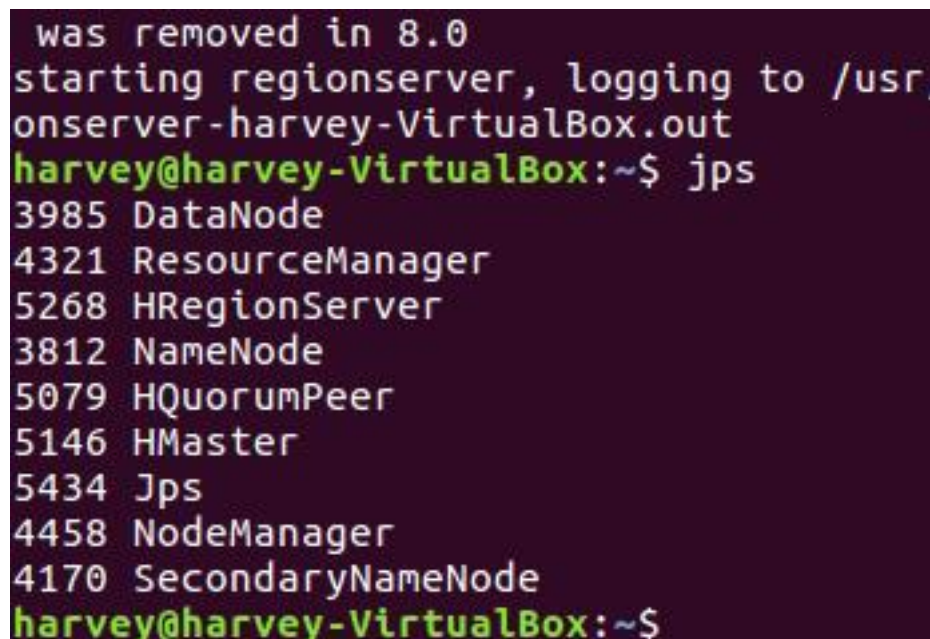
nano hbase-env.sh configurate under # configure PermSize:

export JAVA_HOME=/usr/lib/ jvm/java-8-openjdk-amd64

export HBASE_MANAGES_ZK= true

all files configuration has been done

start-hbase.sh, type jps:

A terminal window with a dark purple background. The text shows the process of starting Hbase services. It starts with "was removed in 8.0", followed by "starting regionserver, logging to /usr, onserver-harvey-VirtualBox.out". Then, the user runs "jps" and the output lists several Hbase processes: DataNode (3985), ResourceManager (4321), HRegionServer (5268), NameNode (3812), HQuorumPeer (5079), HMaster (5146), Jps (5434), NodeManager (4458), and SecondaryNameNode (4170). The prompt "harvey@harvey-VirtualBox:~\$" is visible at the bottom.

```
was removed in 8.0
starting regionserver, logging to /usr,
onserver-harvey-VirtualBox.out
harvey@harvey-VirtualBox:~$ jps
3985 DataNode
4321 ResourceManager
5268 HRegionServer
3812 NameNode
5079 HQuorumPeer
5146 HMaster
5434 Jps
4458 NodeManager
4170 SecondaryNameNode
harvey@harvey-VirtualBox:~$
```

Hbase has been deployed successfully

Phoenix deployment and configuration

sudo wget 'link of downloading Phoenix'

sudo tar -xvf 'downloaded Phoenix file'

sudo mv phoenix4.14 /usr/local/phoenix move the unzipped Phoenix file to specified file directory

Open Pheonix files, find phoenix-hbase-client.jar, phoenix-hbase-server.jar under 'lib' directory, copy and paste to 'lib' files under Hbase

nano ~/.bashrc configurate the environment variable, put below command in the file:

export PHOENIX_HOME=/usr/local/phoenix/

export PHOENIX_CLASSPATH=\$PHOENIX_HOME

`export PATH=$PATH: $PHOENIX_HOME/bin`

`source ./bashrc` save the configuration and restart hadoop and Hbase

`stop-all.sh`

`stop-hbase.sh`

`start-all.sh`

`start-hbase.sh`

`cd /usr/local/phoenix/bin`

`sqlie.py` open phoenix sql interface

```
Setting property: [incremental, false]
Setting property: [isolation, TRANSACTION_READ_COMMITTED]
issuing: !connect jdbc:phoenix: none none org.apache.phoenix.jdbc.PhoenixDriver
Connecting to jdbc:phoenix:
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/phoenix/phoenix-4.14.0-HBase-1.3-client.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.
18/07/25 12:01:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Connected to: Phoenix (version 4.14)
Driver: PhoenixEmbeddedDriver (version 4.14)
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED
Building list of tables and columns for tab-completion (set fastconnect to true to skip)...
144/144 (100%) Done
Done
sqlie version 1.2.0
0: jdbc:phoenix:> █
```

Phoenix has been deployed successfully

Create table and build index:

In the phoenix sql interface, input below command to create table:

```
CREATE TABLE test1 ( times TIME not null, id varchar not null, lat float, lot float, direction float, speed float, vacant BOOLEAN CONSTRAINT pk PRIMARY KEY (times, id));
```

Back to `/usr/local/phoenix/bin` directory, type:

`psql.py -t TEST1 localhost /home/harvey/pcshare/merge_r.csv`

`merge_r.csv` is the target dataset, use bulk load import it to phoenix and saved it in Hbase

After importing the data, enter phoenix sql interface again and type:

create index IDX1 on TEST1 (lat,lot) ASYNC

create index for latitude and longitude, name of the index is IDX1

quit phoenix sql interface, type:

```
${HBASE_HOME}/bin/hbase org.apache.phoenix.mapreduce.index.IndexTool -  
dt TEST1 -it IDX1 -op /phoenix/ ASYNC_IDX_HFILES
```