Lab 3

Install Hbase

Step 1: Download and Install HBase

Check [latest release](http://apache.mirror.gtcomm.net/hbase/) or [Stable release version](http://apache.mirror.gtcomm.net/hbase/stable/)before you download. For production use, I recommend you go with Stabke release.

VER="1.4.10"

wget http://apache.mirror.gtcomm.net/hbase/stable/hbase-$VER-bin.tar.gz

Extract Hbase archive downloaded.

tar xvf hbase-$VER-bin.tar.gz  
sudo mv hbase-$VER/ /usr/local/HBase/

Update your $PATH values.

cat <<EOF | sudo tee /etc/profile.d/hadoop\_java.sh  
export JAVA\_HOME=$(dirname $(dirname $(readlink $(readlink $(which javac)))))  
export HADOOP\_HOME=/usr/local/hadoop  
export HADOOP\_HDFS\_HOME=\$HADOOP\_HOME  
export HADOOP\_MAPRED\_HOME=\$HADOOP\_HOME  
export YARN\_HOME=\$HADOOP\_HOME  
export HADOOP\_COMMON\_HOME=\$HADOOP\_HOME  
export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=\$HADOOP\_HOME/lib/native  
export HBASE\_HOME=/usr/local/HBase  
export PATH=\$PATH:\$JAVA\_HOME/bin:\$HADOOP\_HOME/bin:\$HADOOP\_HOME/sbin:\$HBASE\_HOME/bin  
EOF

Update your shell environment values.

$ **source /etc/profile.d/hadoop\_java.sh**  
$ **echo $HBASE\_HOME**  
/usr/local/HBase

Edit JAVA\_HOME in shell script hbase-env.sh:

$ **sudo vim /usr/local/HBase/conf/hbase-env.sh**  
# Set JAVA\_HOME - Line 27  
export JAVA\_HOME=$(dirname $(dirname $(readlink $(readlink $(which javac)))))

Step 2: Configure HBase

We will do configurations like we did for Hadoop. All configuration files for HBase are located on /usr/local/HBase/conf/ directory.

Option 1: Install HBase in Standalone Mode (Not recommended)

In standalone mode all daemons (HMaster, HRegionServer, and ZooKeeper) ran in one jvm process/instance

Create HBase root directory.

sudo mkdir -p /hadoop/HBase/HFiles  
sudo mkdir -p /hadoop/zookeeper  
sudo chown -R hadoop:hadoop /hadoop/

Open the file for editing.

sudo vim /usr/local/HBase/conf/hbase-site.xml

Now add the following configurations between the *<configuration>* and *</configuration>*tags to look like below.

<configuration>

<property>

<name>hbase.rootdir</name>

<value>file:/hadoop/HBase/HFiles</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/hadoop/zookeeper</value>

</property>

</configuration>

By default, unless you configure the hbase.rootdir property, your data is still stored in */tmp/*.

Now start HBase by using **start-hbase.sh** script in HBase bin directory.

$ **sudo su - hadoop**  
$ **start-hbase.sh**   
running master, logging to /usr/local/HBase/logs/hbase-hadoop-master-hbase.out

Option 2: Install HBase in Pseudo-Distributed Mode (Recommended)

Our value of hbase.rootdir set earlier will start in Standalone Mode. Pseudo-distributed mode means that HBase still runs completely on a single host, but each HBase daemon (HMaster, HRegionServer, and ZooKeeper) runs as a separate process.

To install HBase in Pseudo-Distributed Mode, set its values to:

<configuration>

<property>

<name>hbase.rootdir</name>

<value>hdfs://localhost:8030/hbase</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/hadoop/zookeeper</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

</configuration>

In this setup, Data is stored your data in HDFS instead.

Ensure Zookeeper directory is created.

sudo mkdir -p /hadoop/zookeeper  
sudo chown -R hadoop:hadoop /hadoop/

Now start HBase by using **start-hbase.sh** script in HBase bin directory.

$ **sudo su - hadoop**

$ **start-hbase.sh**

localhost: running zookeeper, logging to /usr/local/HBase/logs/hbase-hadoop-zookeeper-hbase.out

running master, logging to /usr/local/HBase/logs/hbase-hadoop-master-hbase.out

: running regionserver, logging to /usr/local/HBase/logs/hbase-hadoop-regionserver-hbase.out

Check the HBase Directory in HDFS:

$ **hadoop fs -ls /hbase**  
Found 9 items  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:19 /hbase/.tmp  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:19 /hbase/MasterProcWALs  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:19 /hbase/WALs  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:17 /hbase/corrupt  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:16 /hbase/data  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:16 /hbase/hbase  
-rw-r--r-- 1 hadoop supergroup 42 2019-04-07 09:16 /hbase/hbase.id  
-rw-r--r-- 1 hadoop supergroup 7 2019-04-07 09:16 /hbase/hbase.version  
drwxr-xr-x - hadoop supergroup 0 2019-04-07 09:17 /hbase/oldWALs

Step 3: Managing HMaster & HRegionServer

The HMaster server controls the HBase cluster. You can start up to 9 backup HMaster servers, which makes 10 total HMasters, counting the primary.

The HRegionServer manages the data in its StoreFiles as directed by the HMaster. Generally, one HRegionServer runs per node in the cluster. Running multiple HRegionServers on the same system can be useful for testing in pseudo-distributed mode.

Master and Region Servers can be started and stopped using the scripts local-master-backup.sh and local-regionservers.sh respectively.

$ local-master-backup.sh start 2 # Start backup HMaster  
$ local-regionservers.sh start 3 # Start multiple RegionServers

* Each HMaster uses two ports (16000 and 16010 by default). The port offset is added to these ports, so using an offset of **2**, the backup HMaster would use ports *16002* and *16012*

The following command starts 3 backup servers using ports 16002/16012, 16003/16013, and 16005/16015.

$ local-master-backup.sh start 2 3 5

* Each RegionServer requires two ports, and the default ports are *16020* and *16030*

The following command starts four additional RegionServers, running on sequential ports starting at 16022/16032 (base ports 16020/16030 plus 2).

$ local-regionservers.sh start 2 3 4 5

To stop, replace start parameter with stop for each command followed by the offset of the server to stop. Example.

$ local-regionservers.sh stop 5

Starting HBase Shell

Hadoop and Hbase should be running before you can use HBase shell. Here the correct order of starting services.

$ start-all.sh

$ start-hbase.sh

Then use HBase shell.

hadoop@hbase:~$ **hbase shell**  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/usr/local/HBase/lib/slf4j-log4j12-1.7.10.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  
Use "help" to get list of supported commands.  
Use "exit" to quit this interactive shell.  
Version 1.4.9, rd625b212e46d01cb17db9ac2e9e927fdb201afa1, Wed Dec 5 11:54:10 PST 2018  
**hbase(main):001:0>**

Stopping HBase.

stop-hbase.sh