Deploying Hadoop on Single Node Cluster

(In distributed mode)

White paper

By

Anjali Kalra(417660)

Hadoop

Hadoop is an infrastructure for large scale distributed processing of batch. But it can be used on a single machine as well. In Actual Hadoop was built to process "web-scale" data on the order of hundreds of gigabytes to terabytes or petabytes. At this scale, it is likely that the input data set will not even fit on a single computer's hard drive, which is supposed to be much less in memory. So Hadoop includes a distributed file system which breaks up input data and sends different fractions of the original data to several machines in single/multiple cluster to hold. This results in the problem to process the data in parallel from all of the machines in the cluster and to compute output results as efficiently as possible.

Data type that can be handled using Hadoop
Hadoop handles any data type,in any quantity

a) Structured, unstructured

b) Schema, no schema

c) High volume, low volume

d) All kinds of analytic applications

Running Hadoop

Hadoop runs on Unix and on Windows. Linux is the only supported production platform, but other flavors of Unix (including Mac OS X) can be used to run Hadoop for development. Windows is only supported as a development platform.

Hadoop can be used as a single node system or multi node system. Single Node Hadoop cluster is also called as Hadoop Pseudo-Distributed Mode.

How to Install Hadoop on Ubuntu/Other Unix Flavors:

Step 1: Install Latest Version of Java.

As hadoop is developed in JAVA so for Hadoop JAVA is prerequisite.

Step 1A : Install python Properties

--First we need to install forcely Python software properties to add JAVA repositories.

```
$sudo apt-get install python-software-properties
```

Step 1B: Add Repository

--We need to configure repository properties from where ubuntu will install packages automatically.

```
$sudo add-apt-repository ppa:webupd9team/java
```

With above command Ubuntu will install JAVA automatically.

Step 1C: Update the source list

```
$sudo apt-get update
```

--With above command Ubuntu will update its source list.

Step 1D: Install Java

--We can install default(latest) version of JAVA.

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

else we can specify version of JAVA with below mentioned command.

```
$sudo apt-get install oracle-java8-installer
```

--To check for JAVA whether correct version is installed or not run following command-

```
$java -version
```

First it will download the package from the web location of Oracle JAVA download page(Its because we have already installed python software properties hence it will not ask for the location from where we need to download the packages of JAVA). And then it will automatically install JAVA(its because of apt-get install command).

Step 2- Configure SSH

SSH is secured shell. Its used for remote login. We need to intall password less SSH. i.e. without any password we can login into any machine. Pssword less SSH is required for remote script invocation. Because only from master machine we will be starting the daemons, calling the scripts so that on all the slaves required daemons will automatically start.

We know there are two ways of authentication one is password and second is key.

As the process cant send the password interactively hence we will configure keys. There are two types of keys- public and private key.

If we want to authenticate anyone to login in our system. We will provide our public key to that person. Then while loging into the system pubic and private key combination will be checked if it maches then only system permit to login into the system.

This can be achived using Open SSH server and Open SSH Client.

Step 2A-Install Open SSH Server-Client

```
$sudo apt-get install openssh-server openssh-client
```

Step 2B-Generate Key Pairs

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

it will generate the key pairs and ask for file name. We dont need to enter any file name so that it can generate in the default path.

Step 2C-Configure password-less SSH

```
$cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Password less SSH is required for localhost.

To login into localhost secure shell command will be(ssh localhost)

Step 3-Install Hadoop

First we need to download hadoop setup.(we can download any hadoop package provided by any vendor).
For example for cloudera hadoop

go to link http://archive.cloudera.com/cdh5/cdh/5/

download hadoop package/any utility for installation

Step -3A Untar Tar ball

```
$tar xzf  hadoop-2.6.0-cdh5.4.7.tar.gz
```

Move the downloaded and unzippped directory to the local path-

```
sudo mv hadoop-2.6.0-cdh5.4.7 /usr/local/hadoop
```

Note: All the required jars, scripts, configuration files, etc. are available in HADOOP_HOME directory (hadoop-2.6.0-cdh5.4.7).

Step4- Setup Configuration:

Edit .bashrc:
Edit .bashrc file located in user’s home directory and add following parameters:

```
sudo gedit $HOME/.bashrc
```

in the end of file

```
export HADOOP_PREFIX="/home/hdadmin/hadoop-2.6.0-cdh5.4.7"
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}
```

Note: After above step restart the terminal, so that all the environment variables will come into effect

Edit hadoop-env.sh:

Edit configuration file hadoop-env.sh (located in HADOOP_HOME/etc/hadoop) and set JAVA_HOME:

```
sudo gedit hadoop-env.sh
```

change line export JAVA_HOME with-- “export JAVA_HOME=<path-to-the-root-of-your-Java-installation>” (eg: /usr/lib/jvm/java-8-oracle/)
then add

```
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

save
exit

Edit core-site.xml:

Edit configuration file core-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<configuration>
 <property>
 <name>fs.defaultFS</name>
 <value>hdfs://localhost:9000</value>
 </property>
 <property>
 <name>hadoop.tmp.dir</name>
<value>/home/hdadmin/hdata</value>
 </property>
</configuration>
```

Save and exit

Note: /home/hdadmin/hdata is a sample location; please specify a location where you have Read Write privileges

Editing System Configuration

```
sudo gedit /etc/sysctl.conf
```

(type three lines)

```
net.ipv6.conf.all.disable_ipv6=1

net.ipv6.conf.default.disable_ipv6=1

net.ipv6.conf.io.disable_ipv6=1
```

save and exit

Edit hdfs-site.xml:

Edit configuration file hdfs-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<configuration>
<property>
 <name>dfs.replication</name>
 <value>1</value>
 </property>
</configuration>
```

Note-if the value is set to '1' it means the cluster includes single node. We can mention no of nodes which are to be created in cluster in the value field in properties of the configuration tab in hdfs-site.xml

Edit mapred-site.xml:

Edit configuration file mapred-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<configuration>

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

Note-There will be no mapred-site.xml
so cp mapred-site-template.xml mapres-site.xml
Edit configuration file mapred-site.xml(located in HADOOP_HOME/etc/hadoop) and add above entries.

Edit yarn-site.xml:

Edit configuration file yarn-site.xml (located in HADOOP_HOME/etc/hadoop) and add following entries:

```
<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>
</configuration>
```

Create Directories for Name Node and Data Node

We need to create separate directories for each datanode and name node. In which all information related to the nodes will be stored

For creating directories run following commands-

```
sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/namenode

sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/datanode
```

Edit hdfs-site.xml for entering values of name node and data node

```
sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
 <property>
 <name>dfs.replication</name>
<value>1</value>
 </property>

<property>
        <name>dfs.namenode.name.dir/name>
<value>file:/usr/local/hadoop/hadoop_data/hdfs/namenode</value?
<property>
<property>
        <name>dfs.namenode.name.dir/name>
<value>file:/usr/local/hadoop/hadoop_data/hdfs/datanode</value?
```

```
<property>
</configuration>
```

Save and Exit

```
sudo chown 'username':hadoop -R /usr/local/hadoop
```

Start the cluster:
Format the name node:

```
$bin/hdfs namenode -format
```

NOTE: This activity should be done once when you install hadoop, else It will delete all your data from HDFS

Start HDFS Services:

```
$sbin/start-dfs.sh
```

Start YARN Services:

```
$sbin/start-yarn.sh
```

Check whether services have been started

```
$jps
NameNode
DataNode
ResourceManager
NodeManager
```

If all services are started it means we can proceed with the programming on Hadoop Node and cluster.