```
####################################################################
## Basic Initialization script To Install Hadoop as Distributed Mode
## Hadoop 2.6.5
## CP5261 - Big Data Analytics Lab - Semester 2 - 2017-18
####################################################################


JAVA_HOME
/usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java
/usr/lib/jvm/java-7-openjdk-amd64/bin/javac

ssh-keygen
/home/hduser/.ssh/id_rsa

--------------------


Steps to install Hadoop.


sudo apt-get update
sudo apt-get install openjdk-7-jre
sudo apt-get install openjdk-7-jdk
sudo update-alternatives --config java
sudo update-alternatives --config javac
java -version
sudo addgroup hadoop
sudo adduser --ingroup hadoop hduser
su - hduser

$hduser ssh-keygen -t rsa -P ""
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

cat /proc/sys/net/ipv6/conf/all/disable_ipv6

wget https://archive.apache.org/dist/hadoop/core/hadoop-1.0.3/hadoop-1.0.3.tar.gz

clear
ls
sudo cp hadoop-1.0.3.tar.gz /usr/local/
ls
cd /usr/local
sudo tar xzf hadoop-1.0.3.tar.gz
ls
sudo mv hadoop-1.0.3 hadoop
sudo chown -R hduser:hadoop hadoop
$HOME
sudo nano /home/hduser/.bashrc
-----------
# Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop

# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/usr/lib/jvm/java-6-sun

# Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"

# If you have LZO compression enabled in your Hadoop cluster and
# compress job outputs with LZOP (not covered in this tutorial):
# Conveniently inspect an LZOP compressed file from the command
# line; run via:
#
```

```
# $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
#
# Requires installed 'lzop' command.
#
lzohead () {
    hadoop fs -cat $1 | lzop -dc | head -1000 | less
}

# Add Hadoop bin/ directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin
--------------
sudo nano hadoop/conf/hadoop-env.sh
-----------
# The java implementation to use.  Required.
export JAVA_HOME=/usr/lib/jvm/java-6-sun
-----------

sudo mkdir -p /app/hadoop/tmp
sudo chown hduser:hadoop /app/hadoop/tmp
sudo chmod 750 /app/hadoop/tmp
sudo nano hadoop/conf/core-site.xml
----------
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>

<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>
----------
sudo nano hadoop/conf/mapred-site.xml
----------
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>

----------
sudo nano hadoop/conf/hdfs-site.xml
----------
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>

----------
clear

hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode -format
```

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh

hduser@ubuntu:/usr/local/hadoop$ jps

hduser@ubuntu:~$ /usr/local/hadoop/bin/stop-all.sh


-----------------
namenode [master] -  10.240.14.10       / 130.211.243.46
datanode1 [slave1] - 10.240.11.153  / 107.167.181.185


-------------ssh public key - namenode
In google computer install dashboard copy the /home/hduser/id_rsa.pub <namenode>
to datanode instance <ssh>.

--------------------------Multiple Nodes

After copy ssh to datanode.

<in master node>
hduser@master:~$ ssh master
hduser@master:~$ ssh slave

hduser@master:~$ bin/hadoop-daemon.sh start [namenode | secondarynamenode |
datanode | jobtracker | tasktracker]
or
update conf/masters
master

update conf/slaves
master
slaves

<in all machines>
update conf/core-site.xml
---------
<property>
  <name>fs.default.name</name>
  <value>hdfs://master:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>

----------
update conf/mapred-site.xml
----------
<property>
  <name>mapred.job.tracker</name>
  <value>master:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>
----------
update conf/hdfs-site.xml
----------
<property>
  <name>dfs.replication</name>
  <value>2</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is created.
  The default is used if replication is not specified in create time.
  </description>
</property>
----------

Additional Settings done in conf/mapred-site.xml
----------
“mapred.local.dir“
    Determines where temporary MapReduce data is written. It also may be a list of
directories.
“mapred.map.tasks“
    As a rule of thumb, use 10x the number of slaves (i.e., number of
TaskTrackers).
“mapred.reduce.tasks“
    As a rule of thumb, use num_tasktrackers * num_reduce_slots_per_tasktracker *
0.99. If num_tasktrackers is small (as in the case of this tutorial), use
(num_tasktrackers - 1) * num_reduce_slots_per_tasktracker.

<in master node>
hduser@master:/usr/local/hadoop$ bin/hadoop namenode -format

hduser@master:/usr/local/hadoop$ bin/start-dfs.sh
hduser@master:/usr/local/hadoop$ jps

<in slave node>
hduser@slave:/usr/local/hadoop$ jps


<in master node>
hduser@master:/usr/local/hadoop$ bin/start-mapred.sh

hduser@master:/usr/local/hadoop$ jps

<in slave node>
hduser@slave:/usr/local/hadoop$ jps
```

Result

Hadoop Multi Node cluster is installed successfully.