**Create a t2.micro instance**

connect thru putty using .ppk key

(or ssh -i Downloads/file.pem [ubuntu@public\_dns\_address](mailto:ubuntu@public_dns_address))

wget https://s3.amazonaws.com/shaan/My+Self/alam.pem

**# Copy public key on to the DataCenter main server**

sudo apt-get update && sudo apt-get dist-upgrade -y

cp alam.pem ~/.ssh (make sure keyname)

sudo -i

passwd

exit

**# Create a Hadoop user for accessing HDFS (**no need to create same user**)**

sudo addgroup ubuntu

sudo adduser --ingroup ubuntu ubuntu

sudo adduser ubuntu sudo

sudo su ubuntu

**# Install Java 8**

sudo apt-get install -y python-software-properties debconf-utils  
sudo add-apt-repository -y ppa:webupd8team/java  
sudo apt-get update  
echo "oracle-java8-installer shared/accepted-oracle-license-v1-1 select true" | sudo debconf-set-selections  
sudo apt-get install -y oracle-java8-installer

**# Install Hadoop**

wget http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz

sudo tar xzvf hadoop-2.7.2.tar.gz

sudo mv hadoop-2.7.2 /usr/local/hadoop

**# Set Enviornment Variable**

readlink -f $(which java)

cat >>$HOME/.bashrc <<EOL

# -- HADOOP ENVIRONMENT VARIABLES START -- #

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

export HADOOP\_HOME=/usr/local/hadoop

export PATH=\$PATH:\$HADOOP\_HOME/bin

export PATH=\$PATH:\$HADOOP\_HOME/sbin

export PATH=\$PATH:/usr/local/hadoop/bin/

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\_CONF\_DIR=/usr/local/hadoop/etc/hadoop  
export PDSH\_RCMD\_TYPE=ssh

# -- HADOOP ENVIRONMENT VARIABLES END -- #

EOL

exec bash

sudo chown -R ubuntu:ubuntu /usr/local/hadoop

**#Update hadoop-env.sh**

sudo su -c 'echo export JAVA\_HOME=/usr/lib/jvm/java-8-oracle >> /usr/local/hadoop/etc/hadoop/hadoop-env.sh'

sudo su -c 'echo export HADOOP\_LOG\_DIR=/var/log/hadoop/ >> /usr/local/hadoop/etc/hadoop/hadoop-env.sh'

sudo mkdir /var/log/hadoop/

sudo chown ubuntu:ubuntu -R /var/log/hadoop

**#Disable IPV6**

cat /proc/sys/net/ipv6/conf/all/disable\_ipv6

sudo sysctl -p

sudo su -c 'cat >>/etc/sysctl.conf <<EOL

net.ipv6.conf.all.disable\_ipv6 =1

net.ipv6.conf.default.disable\_ipv6 =1

net.ipv6.conf.lo.disable\_ipv6 =1

EOL'

**#Disable FireWall iptables**

sudo iptables -L -n

sudo ufw status

sudo ufw disable

**#Disabling Transparent Hugepage Compaction**

#Red Hat/CentOS: /sys/kernel/mm/redhat\_transparent\_hugepage/defrag  
#Ubuntu/Debian, OEL, SLES: /sys/kernel/mm/transparent\_hugepage/defrag  
  
cat /sys/kernel/mm/transparent\_hugepage/defrag

sudo sed -i '/exit 0/d' /etc/rc.local

sudo su -c 'cat >>/etc/rc.local <<EOL

if test -f /sys/kernel/mm/transparent\_hugepage/enabled; then

echo never > /s ys/kernel/mm/transparent\_hugepage/enabled

fi

if test -f /sys/kernel/mm/transparent\_hugepage/defrag; then

echo never > /sys/kernel/mm/transparent\_hugepage/defrag

fi

exit 0

EOL'

sudo -i

source /etc/rc.local

**# Set Swappiness**

sudo sysctl -a | grep vm.swappiness

sudo su -c 'cat >>/etc/sysctl.conf <<EOL

sudo sysctl -w vm.swappiness=0

EOL'

sudo sysctl –p (no use)

**# Configure NTP**

timedatectl status  
timedatectl list-timezones  
sudo timedatectl set-timezone Asia/Kolkata  
sudo apt-get install ntp

sudo ntpq -p  
sudo nano /etc/ntp.conf (no use)

**# Root Reserved Space**

mkfs.ext4 -m 0 /dev/xvda1 ( filesystem is not suppose to be mounted)  
lsblk  
sudo tune2fs -m 0 /dev/xvda1

Most frequently asked Question whether a JBOD configuration, RAID configuration, or LVM configuration is required. The entire Hadoop ecosystem was created with a JBOD configuration in mind. HDFS is an immutable filesystem that was designed for large file sizes with long sequential reads. This goal plays well with stand-alone SATA drives, as they get the best performance with sequential reads.  
In summary, whereas RAID is typically used to add redundancy to an existing system, HDFS already has that built in.  
In fact, using a RAID system with Hadoop can negatively affect performance.  
For the same reasons, configuring your Hadoop drives under LVM is neither necessary nor recommended.

**##Configure SSH Password less logins**

sudo su -c touch /home/ubuntu/.ssh/config; echo "Host \*\n StrictHostKeyChecking no\n UserKnownHostsFile=/dev/null" > /home/ubuntu/.ssh/config

echo -e 'y\n'| ssh-keygen -t rsa -P "" -f $HOME/.ssh/id\_rsa

cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

sudo service ssh restart

**#Create SnapShot at this point.**

Go to Launch Instance->My AMIs->Select->and take 2 t2.micro instance for dn & rm

**sudo nano /etc/hosts and include these lines:FQDN**

127.0.0.1 localhost

172.31.30.102 ip-172-31-30-102.ap-south-1.compute.internal nn

172.31.23.4 ip-172-31-23-4.ap-south-1.compute.internal rm  
172.31.23.3 ip-172-31-23-3.ap-south-1.compute.internal dn

**#Configure SSH Password less logins**

sudo chown ubuntu:ubuntu -R /var/log/hadoop

*touch ~/.ssh/config*

*nn]$ssh dn*

*sudo nano /etc/hosts (make all 3 host entry)*

*IdentityFile ~/.ssh/alam.pem*

*nn]$ssh rm*

*sudo nano /etc/hosts (make all 3 host entry)*

*IdentityFile ~/.ssh/alam.pem*

*ssh nn (already there no need to move)*

*scp ~/.ssh/config nn:~/.ssh*

*scp ~/.ssh/config dn:~/.ssh*

*scp ~/.ssh/config rm:~/.ssh*

*!ssh nn 'cat >> ~/.ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub  
!ssh rm 'cat >> ~/.ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub  
!ssh dn 'cat >> ~/.ssh/authorized\_keys' < ~/.ssh/id\_rsa.pub*

**# Configure pdsh**

sudo apt-get install pdsh -y

sudo nano /etc/genders

export PDSH\_RCMD\_TYPE=ssh

(nn,dn,rm)

pdsh -a uptime

**#Setting Up Secondary Name Node**

create the masters file in HADOOP\_CONF\_DIR

nano /usr/local/hadoop/etc/hadoop/masters

hostname -f

nano /usr/local/hadoop/etc/hadoop/slaves

**#Update core-site.xml on nn**

sudo sed -i '/<configuration>/,/<\/configuration>/d' /usr/local/hadoop/etc/hadoop/core-site.xml

sudo su -c 'cat >> /usr/local/hadoop/etc/hadoop/core-site.xml <<EOL

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://private dns:9000</value>

</property>

</configuration>

EOL'

**#Update hdfs-site.xml on name node**

mkdir -p /usr/local/hadoop/data/hdfs/namenode

sudo sed -i '/<configuration>/,/<\/configuration>/d' /usr/local/hadoop/etc/hadoop/hdfs-site.xml

sudo su -c 'cat >>/usr/local/hadoop/etc/hadoop/hdfs-site.xml <<EOL

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///usr/local/hadoop/data/hdfs/namenode</value>

</property>

</configuration>

EOL'

**#Update hdfs-site.xml on datanode**

sudo sed -i '/<configuration>/,/<\/configuration>/d' /usr/local/hadoop/etc/hadoop/hdfs-site.xml

sudo su -c 'cat >>/usr/local/hadoop/etc/hadoop/hdfs-site.xml <<EOL

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.datanode.name.dir</name>

<value>file:///usr/local/hadoop/data/hdfs/datanode</value>

</property>

</configuration>

EOL'

mkdir -p /usr/local/hadoop/data/hdfs/datanode

**#Update yarn-site.xml (on nn)**

sudo sed -i '/<configuration>/,/<\/configuration>/d' /usr/local/hadoop/etc/hadoop/yarn-site.xml

sudo su -c 'cat >>/usr/local/hadoop/etc/hadoop/yarn-site.xml <<EOL

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>private dns of rm</value>

</property>

</configuration>

EOL'

**#Update mapred-site.xml**

cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml

sudo sed -i '/<configuration>/,/<\/configuration>/d' /usr/local/hadoop/etc/hadoop/mapred-site.xml

sudo su -c 'cat >>/usr/local/hadoop/etc/hadoop/mapred-site.xml <<EOL

<configuration>

<property>

<name>mapreduce.jobtracker.address</name>

<value>rm private dns:54311</value>

</property>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

EOL'

sudo chown -R ubuntu:ubuntu $HADOOP\_HOME

**#SCP all the files**

cd /usr/local/hadoop/etc/hadoop && scp core-site.xml mapred-site.xml yarn-site.xml hadoop-env.sh slaves dn:/usr/local/hadoop/etc/hadoop

cd /usr/local/hadoop/etc/hadoop && scp core-site.xml mapred-site.xml yarn-site.xml hdfs-site.xml hadoop-env.sh slaves rm:/usr/local/hadoop/etc/hadoop

**#Format Namenode**

hdfs namenode -format

start-dfs.sh

start-yarn.sh

**pdsh -a jps**

hdfs dfs -mkdir /user

hdfs dfs -mkdir /user/ubuntu

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar teragen 500000 random-data

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar terasort random-data sorted-data

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-\*-tests.jar TestDFSIO -write -nrFiles 10 -fileSize 5MB

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-\*-tests.jar TestDFSIO -read -nrFiles 10 -fileSize 5MB

extra

nano /usr/local/hadoop/etc/hadoop/core-site.xml

scp /home/ubuntu/.bashrc dn:~

scp /home/ubuntu/.bashrc rm:~