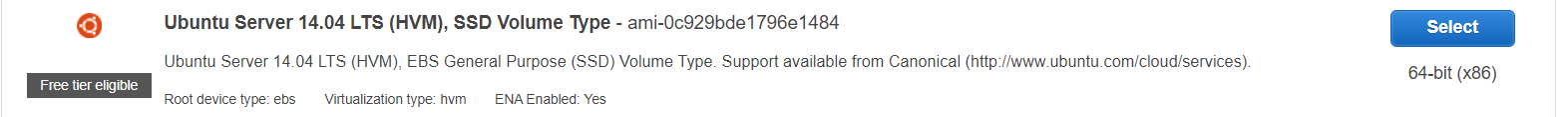
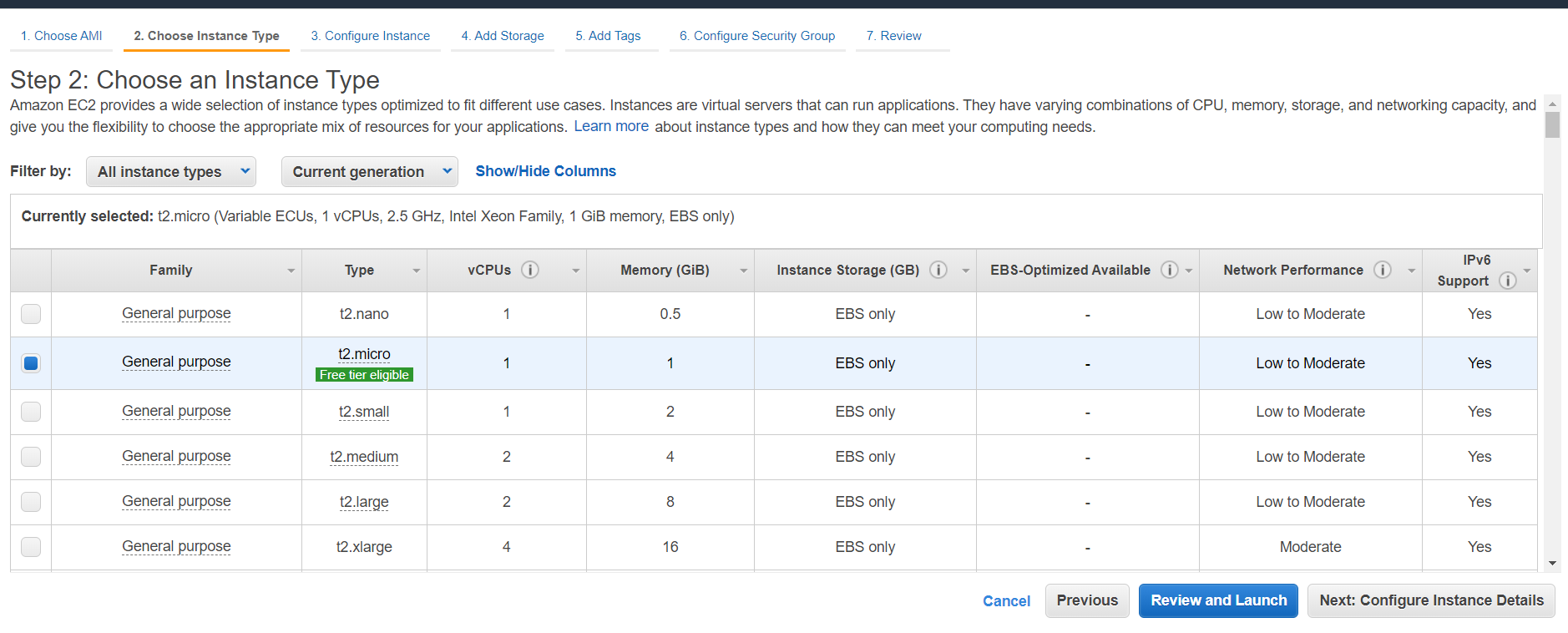
A step by step description of how I built my AMI

\*\* Due to an issue was unable to use last name as my instance name.

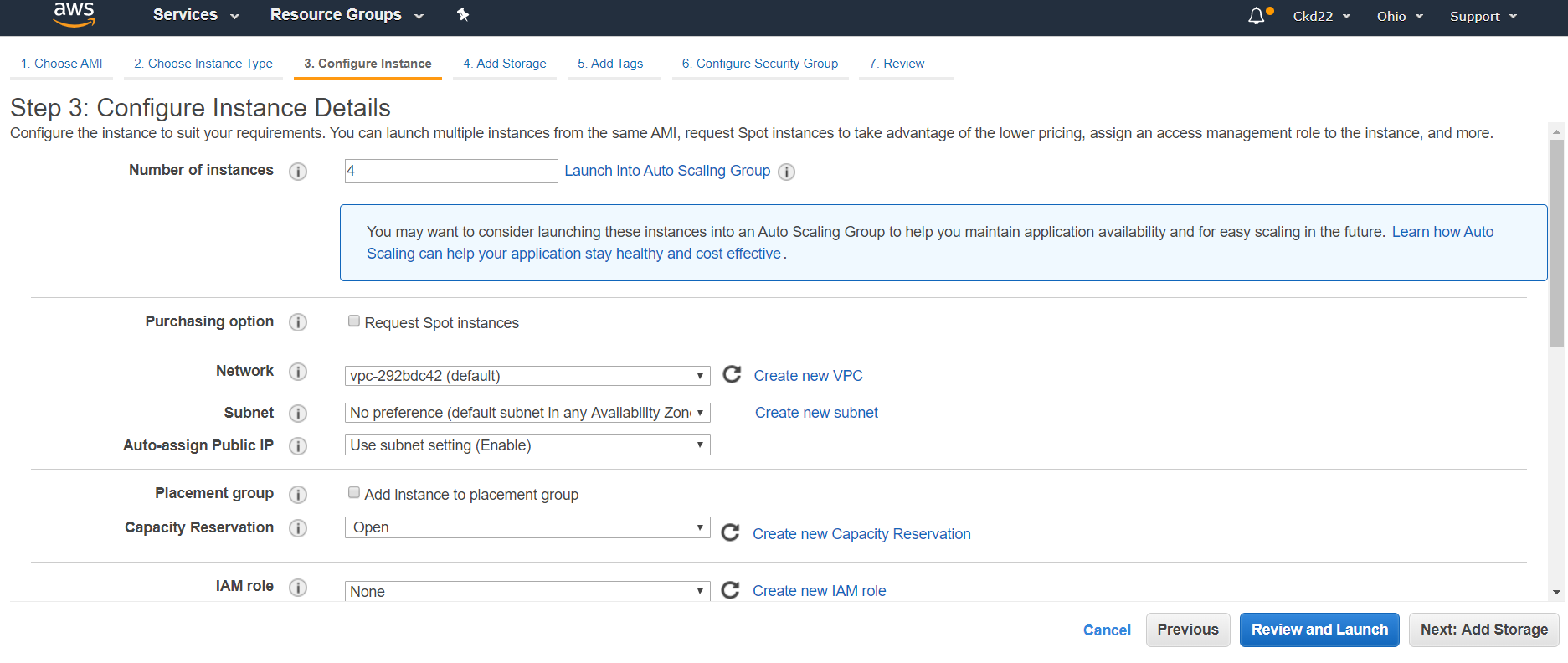
1. Launch instance button
2. We choose and amazon machine image (AMI), We will here set 4 node system i.e. 1 main node and 3 data node. We chose the free tier.



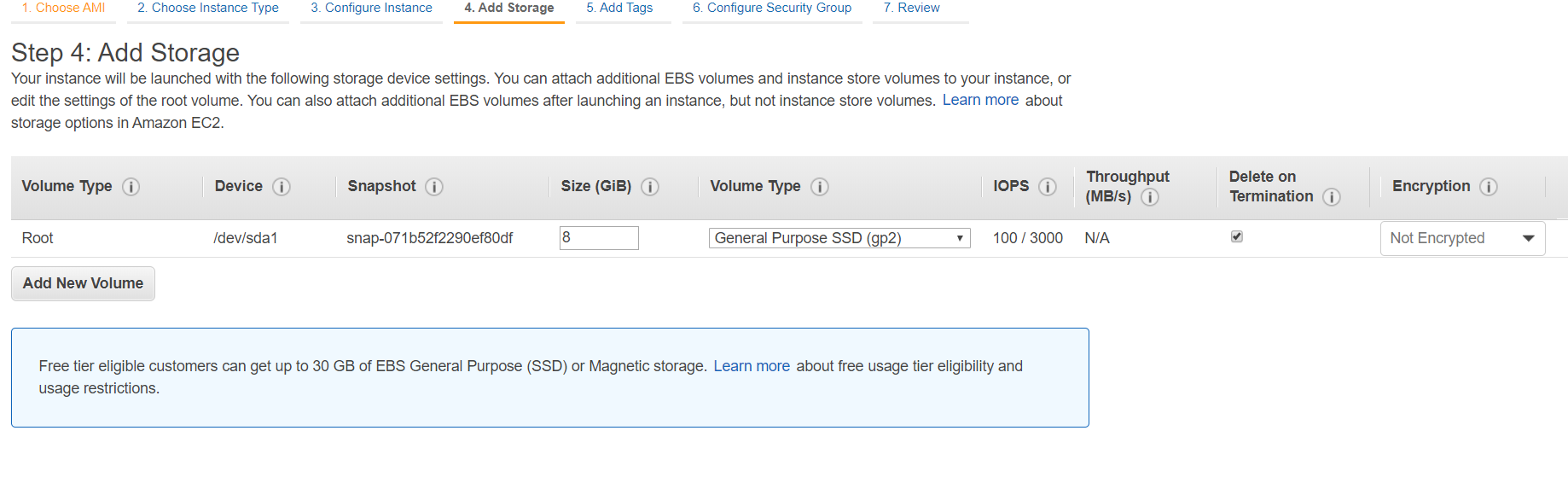
1. I used **Ubuntu Server 14.04 LTS (HVM), SSD Volume Type** - ami-0c929bde1796e1484.
2. Then we choose an instance type General purpose t2.micro



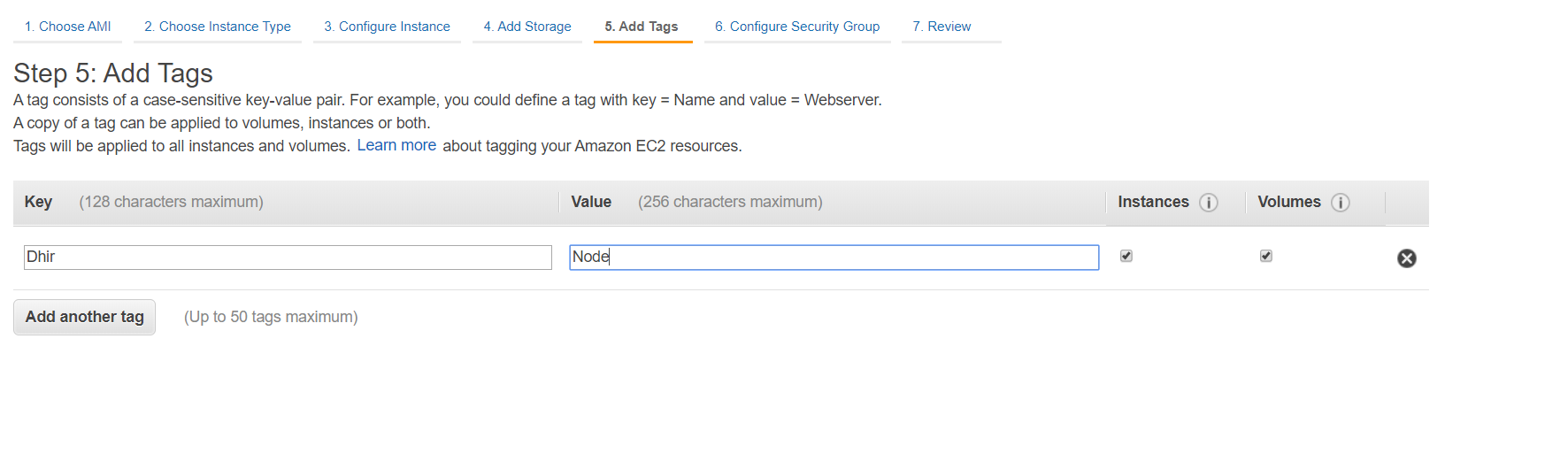
1. Click on configure instance detail
2. Change number of instances into 4



1. Click on ADD storage
2. We will not change anything 8 GB is enough



1. Click on Add tags
2. Key = Dhir and Value = Node



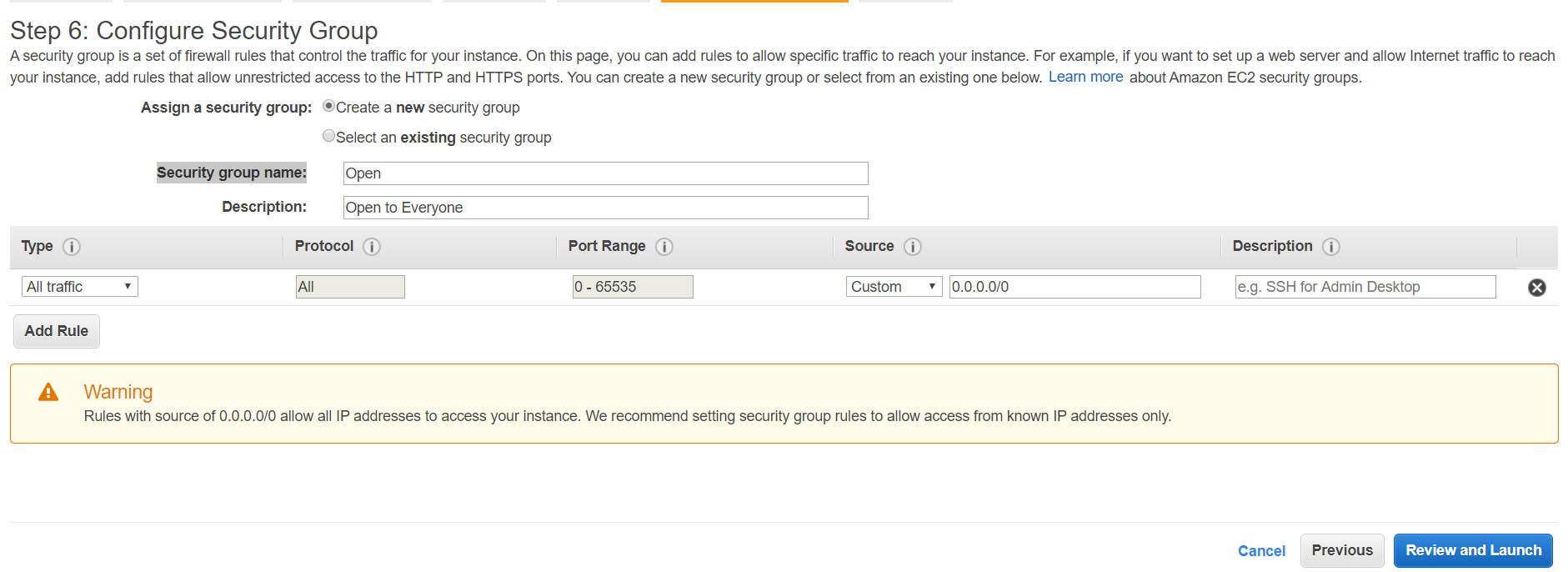
1. Click on configure security group

Changes made are :

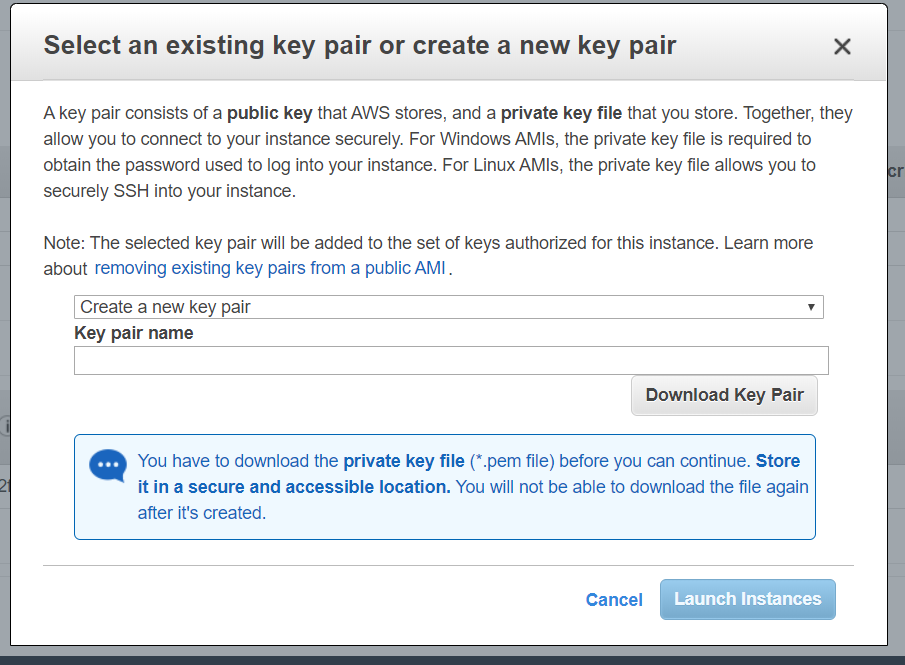
**Security group name: Open**

**Description: Open to Everyone**

**Type: All Traffic**

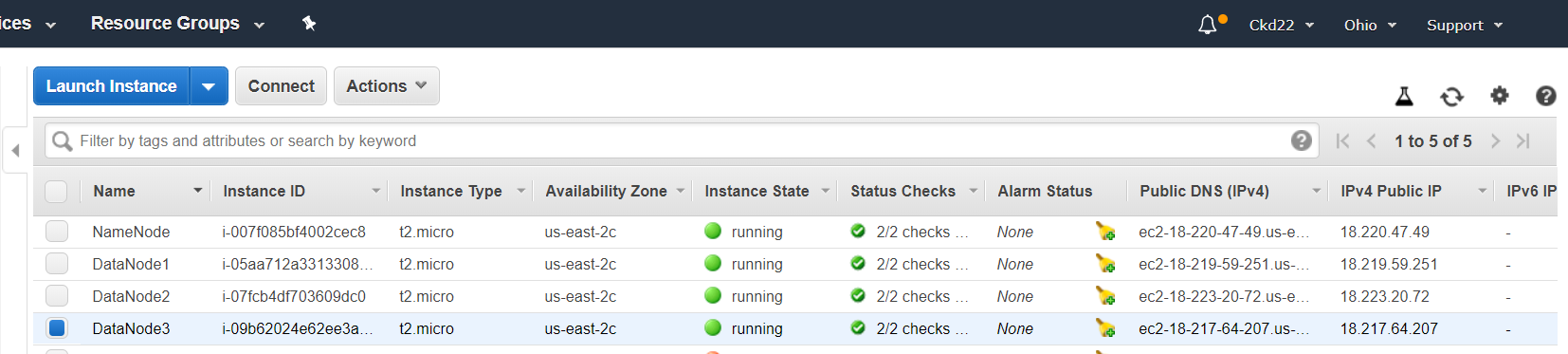


1. Review and Launch
2. Click on launch
3. Create a key pair name

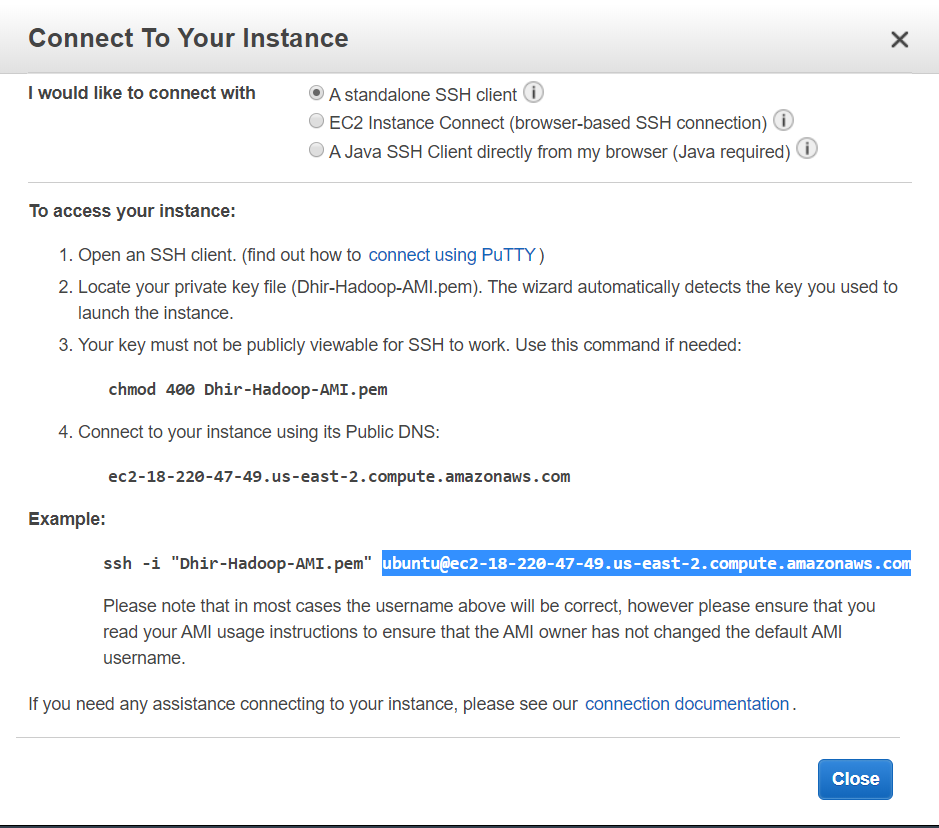


My key pair name is : Dhir-Hadoop-AMI

1. Click on launch instances
2. Click on view instances
3. Now Name the nodes. Screenshot shows the name of my 4 instances .



1. Now we will set up putty
2. From Name node –

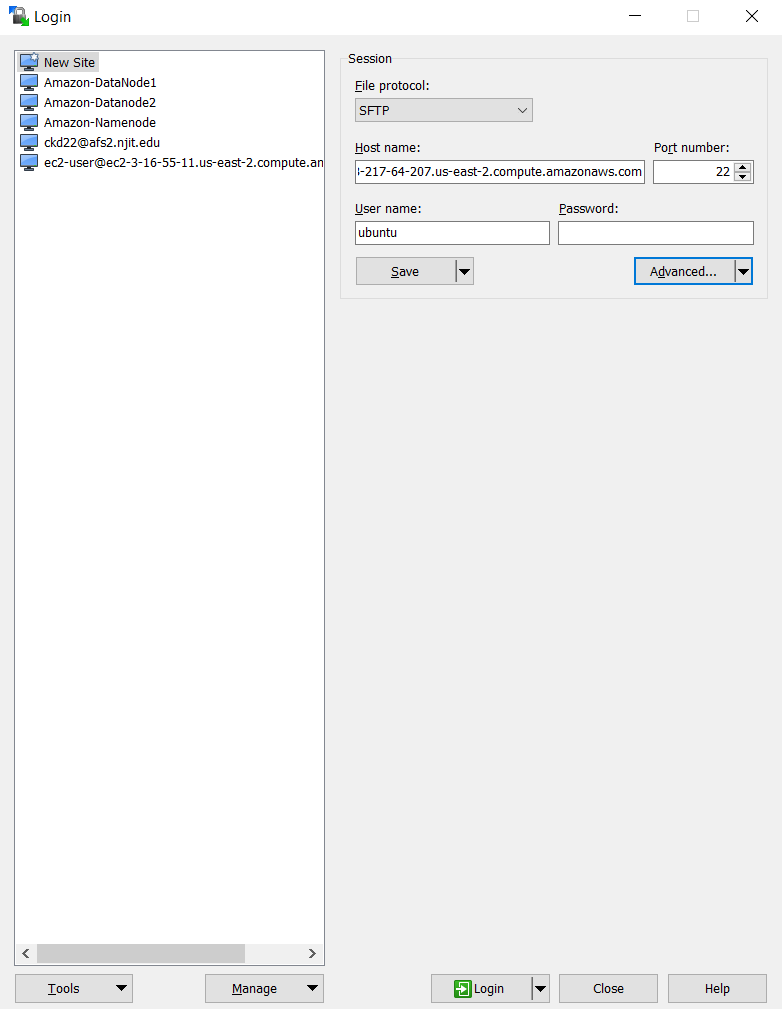


We will use the highlighted portion as host name and from menu in left we will go to SSH and then to auth and then browse the ppk file

( first we need to convert the ppm file for that we need to open putty gen and then select the ppk file by loading private key and set it to RSA generate then save private key)

Similarly we have to do it for all the three data nodes.

1. Now we will setup WINSCP to set files
2. Select Newsite from left tab -> in host name write Public dns name for all your 4 nodes -> user name : ubuntu -> advanced tab -> advanced option -> ssh (left side)-> authentication -> select ppk file -> ok -> save (do the same for all four nodes )



1. Configurations

Host NameNode

Hostname ec2-18-220-47-49.us-east-2.compute.amazonaws.com

User ubuntu

Identity file ~/.ssh/Dhir-Hadoop-AMI.pem

Host DataNode1

Hostname ec2-18-219-59-251.us-east-2.compute.amazonaws.com

User ubuntu

Identity file ~/.ssh/Dhir-Hadoop-AMI.pem

Host DataNode2

Hostname ec2-18-223-20-72.us-east-2.compute.amazonaws.com

User ubuntu

Identity file ~/.ssh/Dhir-Hadoop-AMI.pem

Host DataNode3

Hostname ec2-18-217-64-207.us-east-2.compute.amazonaws.com

User ubuntu

Identity file ~/.ssh/Dhir-Hadoop-AMI.pem

Now in winscp this config file and pem file is uploaded to namenode and then security level has to be set on pem file by opening session on putty .

Then we type

sudo chmod 600 ~/Dhir-Hadoop-AMI.pem

we do this for all the other data nodes from winscp.

1. Now we will write the command

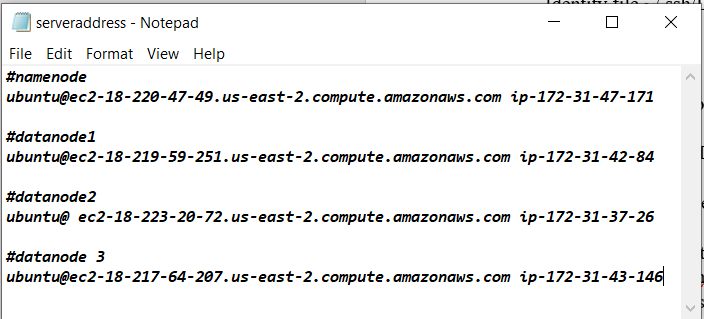
ssh-keygen -f ~/.ssh/sshkey\_rsa -t rsa -P ""

cat ~/.ssh/sshkey\_rsa.pub >> ~/.ssh/authorized\_keys

now we will do it for all data nodes

1. And then on namenode do ssh datanode1, ssh datanode2 and ssh datanode3
2. Now install Hadoop

I wrote the server details of all namenodes and datanodes in one file



Now commands used are (commands are used in all four instances)

* Sudo apt-get update
* Sudo apt-get install openjdk-7-jdk
* Java -version
* wget <http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz> -P ~/Downloads

sudo tar zxvf ~/Downloads/hadoop-\* -C/usr/local

sudo mv /usr/local/Hadoop-\* /usr/local/Hadoop

1. now in home/ubuntu

open .profile which is hidden and end of file paste this in all instances

**export JAVA\_HOME=/usr**

**export PATH=$PATH:$JAVA\_HOME/bin**

**export HADOOP\_HOME=/usr/local/Hadoop**

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

**export HADOOP\_CONF\_DIR=/usr/local/hadoop/etc/hadoop**

#now load variables

. ~/.profile

1. #Hadoop Configuration file on all nodes

#$HADOOP\_CONF\_DIR/hadoop-env.sh change JAVA\_HOME

Export JAVA\_HOME=/usr

#$HADOOP\_CONF\_DIR/core-site.xml change configuration element

#change the namenode\_public\_dns to your namenode public dns

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://namenode\_public\_dns:9000</value>

</property>

</configuration>

On namenode on /etc at hosts file

sudo chown ubuntu /etc/hosts

now add this to beginning of file

172.31.47.171 ec2-18-220-47-49.us-east-2.compute.amazonaws.com

172.31.42.84 ec2-18-219-59-251.us-east-2.compute.amazonaws.com

172.31.37.26 ec2-18-223-20-72.us-east-2.compute.amazonaws.com

172.31.43.146 ec2-18-217-64-207.us-east-2.compute.amazonaws.com

And change the permission back : sudo chown root /etc/hosts

1. # now -> /usr/local/hadoop/etc/Hadoop -> hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

<value>file:///usr/local/hadoop/hadoop\_data/hdfs/namenode</value>

</property>

</configuration>

30)

# create the Hadoop data directory in the $hadoop\_home directory

sudo mkdir -p $HADOOP\_HOME/Hadoop\_data/hdfs/namenode

# create filename “masters” in the $HADOOP\_CONF\_DIR directory

echo “namenode” | cat >>/usr/local/hadoop/etc/hadoop /masters

echo "datanode1" | cat >> $HADOOP\_CONF\_DIR/slaves

echo "datanode2" | cat >> /usr/local/hadoop/etc/hadoop/slaves

echo "datanode3" | cat >> /usr/local/hadoop/etc/hadoop/slaves

changing ownership to ubuntu

sudo chown -R ubuntu $HADOOP\_HOME

1. Now datanode configurations:

Hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

<value>file:///usr/local/hadoop/hadoop\_data/hdfs/datanode</value>

</property>

</configuration>

1. yarnconfiguration

<configuration>

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

<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>

<property>

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

<value>ec2-18-220-47-49.us-east-2.compute.amazonaws.com</value>

</property>

</configuration>

--mapreduceconfiguration

<configuration>

<property>

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

<value>ec2-18-220-47-49.us-east-2.compute.amazonaws.com:54311</value>

</property>

<property>

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

<value>yarn</value>

</property>

</configuration>

1. scp $HADOOP\_CONF\_DIR/hdfs-site.xml datanode2:$HADOOP\_CONF\_DIR

scp $HADOOP\_CONF\_DIR/hdfs-site.xml datanode3:$HADOOP\_CONF\_DIR

1. #create a data directory on each datanode

Sudo mkdir -p $HADOOP\_HOME/hadoop\_data/hdfs/datanode

Location for all the hdfs for namenodes, datanodes are:

/usr/local/hadoop/hadoop\_data/hdfs/namenode

/usr/local/hadoop/hadoop\_data/hdfs/datanode

1. for namenode only

#starting up the Hadoop cluster

#format the HDFS

hdfs namenode -format //formatting hardrive

1. #start the DFS service

$HADOOP\_HOME/sbin/start-dfs.sh

1. ##BROWSE the hdfs in your web browser

# change namenode\_PUBLIC\_DNS to ur namenode public dns

ec2-18-220-47-49.us-east-2.compute.amazonaws.com:50070

1. #start yarn on NameNode

$HADOOP\_HOME/sbin/start-yarn.sh

$HADOOP\_HOME/sbin/mr-jobhistory-daemon.sh start historyserver

39)#run JPS on namenode

jps

40)#testing the hdfs

#create text file

echo “Hello! This is my file ” | cat >> my\_file.txt

#list the hdfs files

hdfs dfs -ls

#make directory name user

hdfs dfs -mkdir /user

#list the hdfs files

hdfs dfs -ls /

#copied created file few times

hdfs dfs -copyFromLocal ~/my\_file.txt /user

hdfs dfs -copyFromLocal ~/my\_file.txt /user/my\_file2.txt

hdfs dfs -copyFromLocal ~/my\_file.txt /user/my\_file2.txt

# list the files in new directory

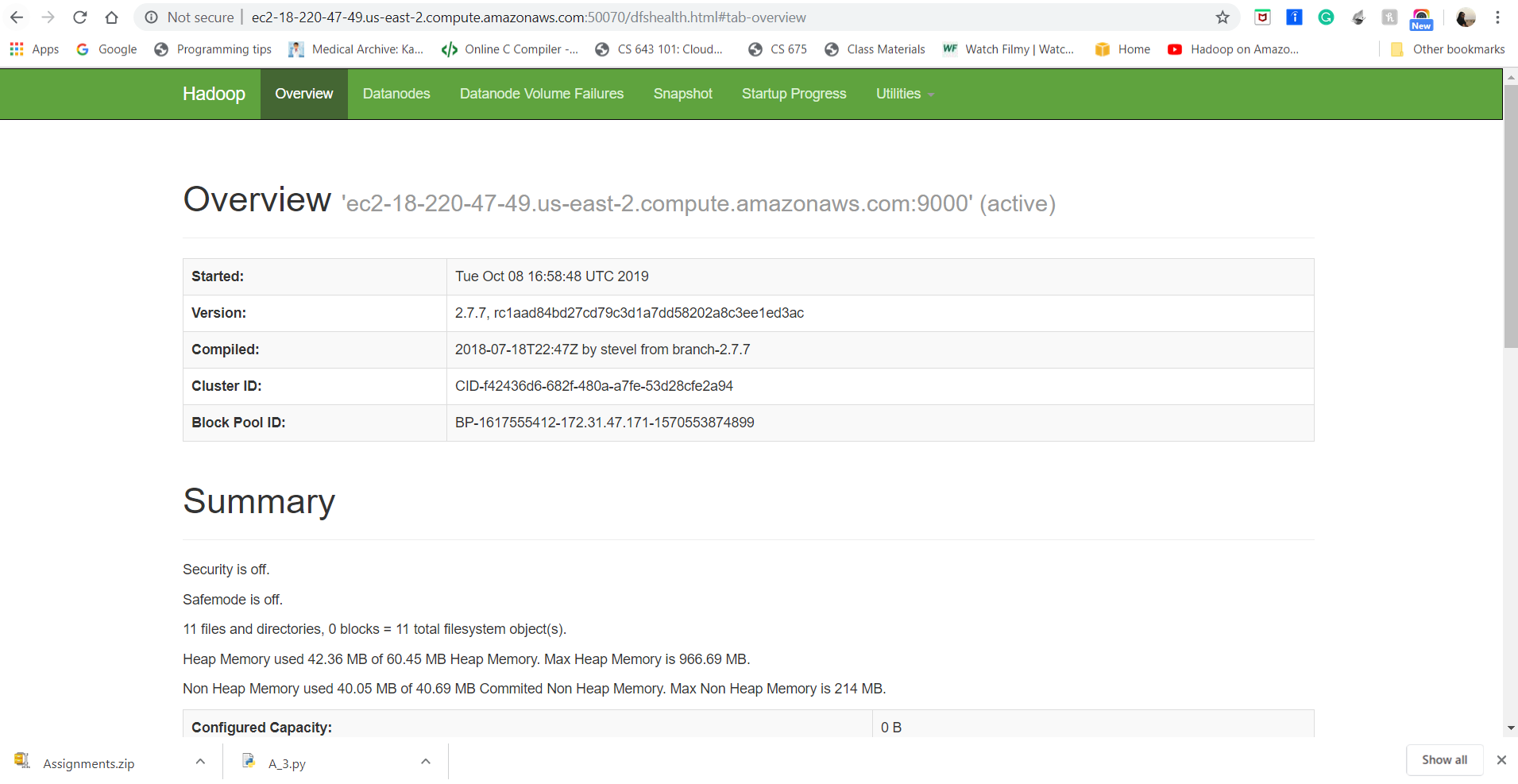
hdfs dfs -ls /user

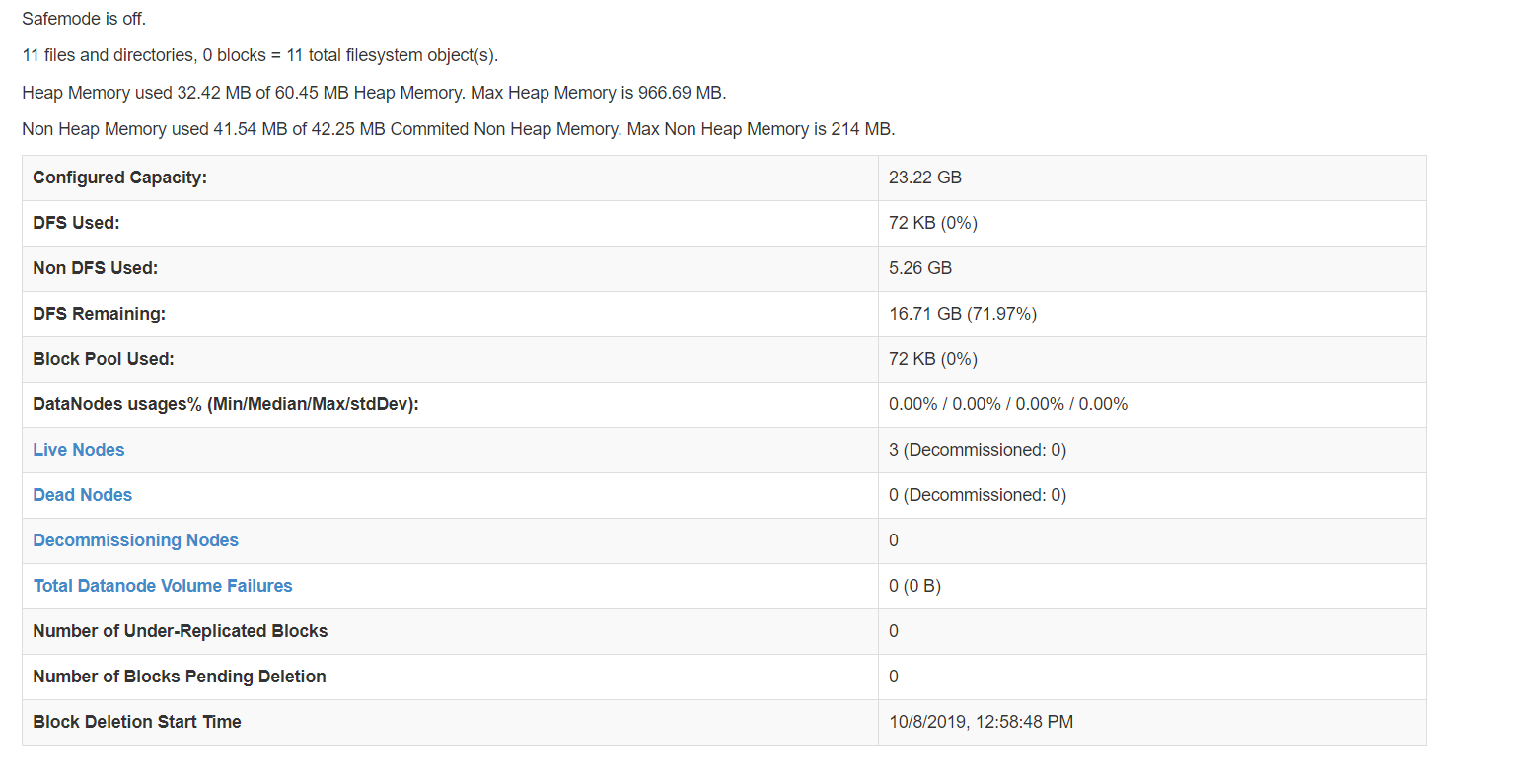
#remove the files with name starting woth my\_file

hdfs dfs -rm /usr/my\_file\*

#remove new directory

hdfs dfs -rmdir /usr





hdfs dfs -copyFromLocal /home/ubuntu/Downloads/Assignment/\* /Assignment1/