Published in AWS Tip

Dineshvarma Guduru Follow

Dec 27, 2021 · 7 min read · Listen

Setting up Multi-Node Apache Hadoop Cluster on AWS EC2 from scratch

This article will be a demonstration of installing and running Apache Hadoop on AWS EC2 cluster.

If you are looking for video resource, Checkout this link.

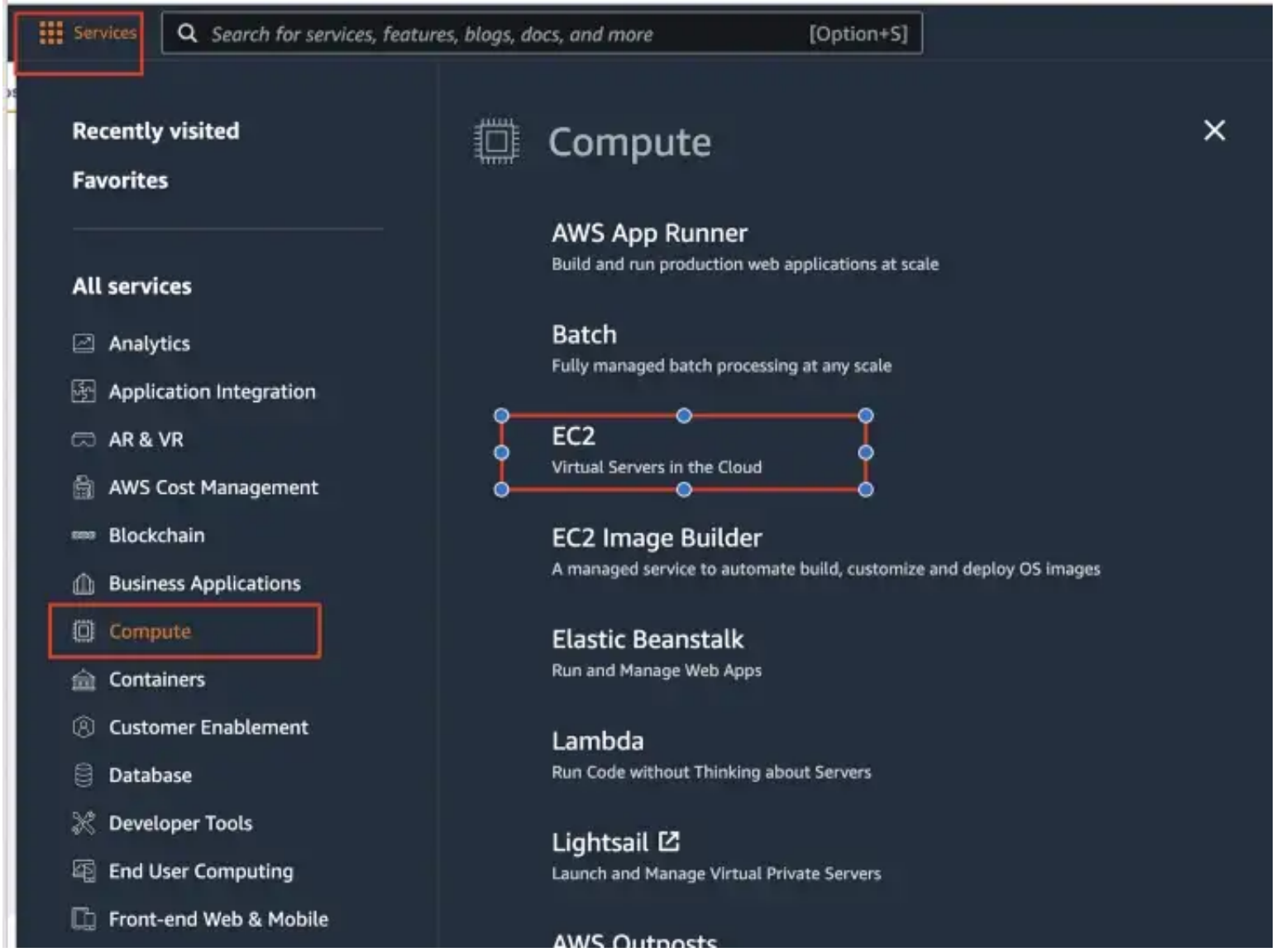
Pre-Requisites

AWS Account is required for following this blog. If you don't have an account already Sign Up for one from here or check out this video for full steps.

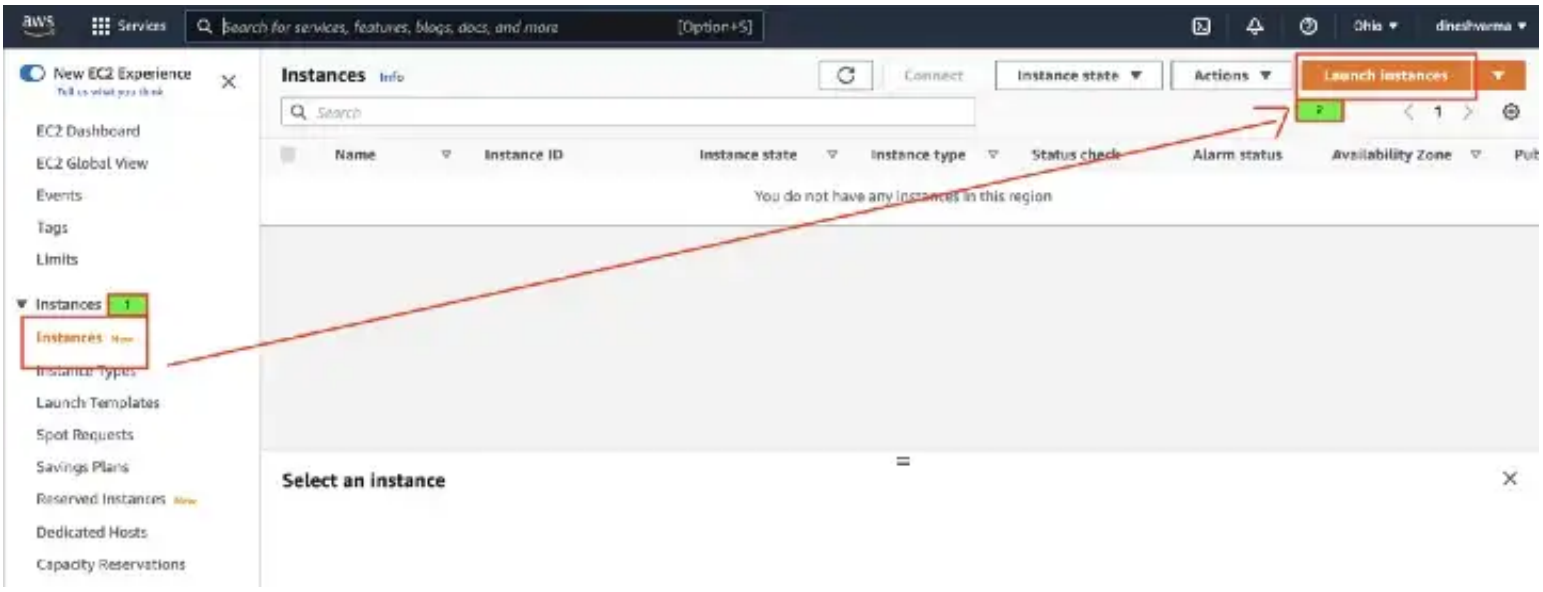
This set up involves Namenode, Secondary Namenode and 2 Datanodes.

Create AWS EC2 Instances

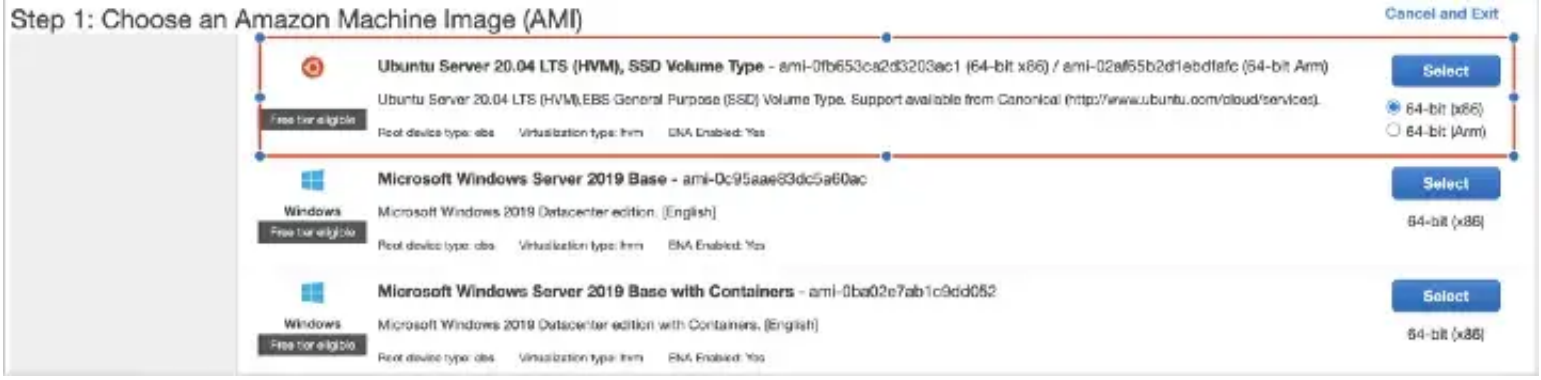
Open AWS Management Console and select **Services -> Computing -> EC2** from the on top left corner.



Now select **Instances** and **Launch instances.**

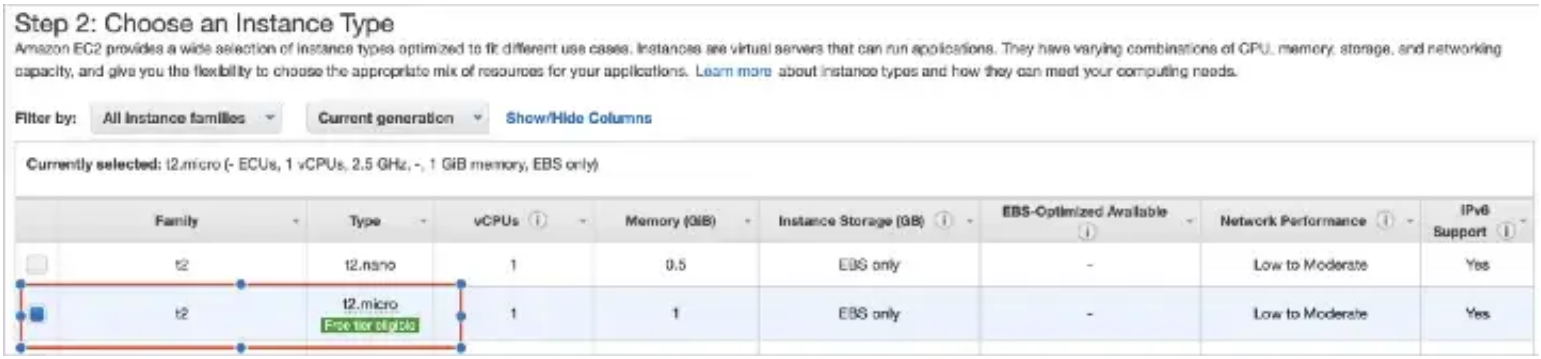


Select any Amazon Machine Image (AMI) which has free tier eligible tag on it. For Hadoop installation I prefer Ubuntu Server. Select **Ubuntu Server 20.04 LTS (HVM).**



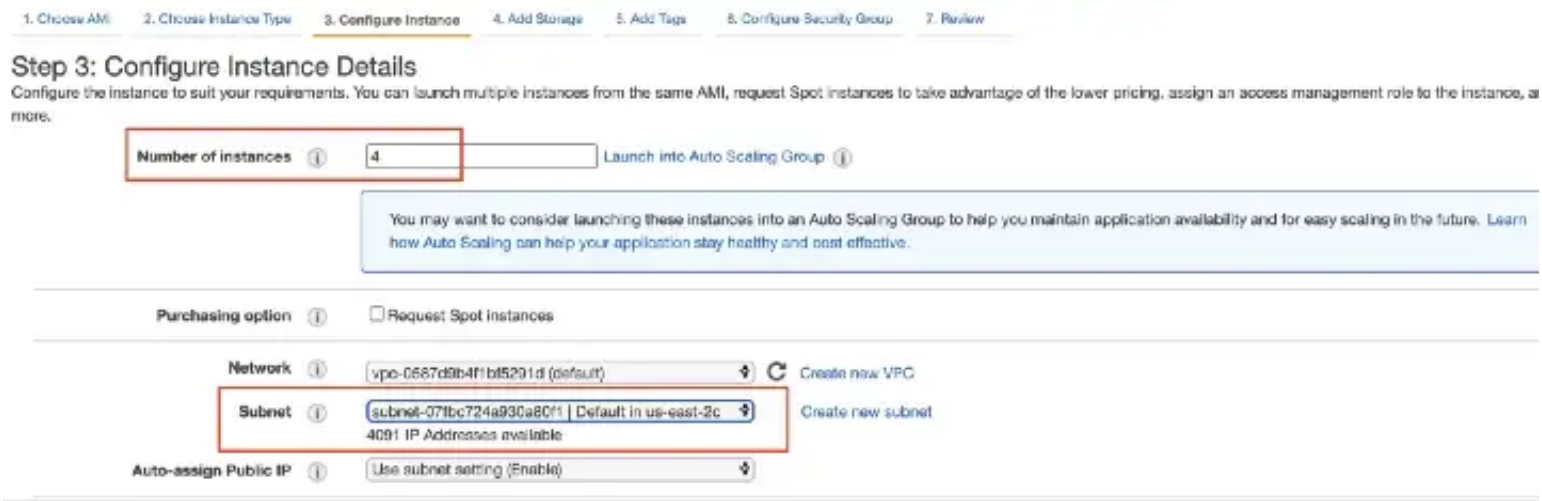
Next: Choose an Instance Type

Choose the instance that has Free tier eligible on it. We only have **t2.micro** with only 1 CPU and 1 GB Memory. If you want better configuration, you might get charged for using the other instance types.



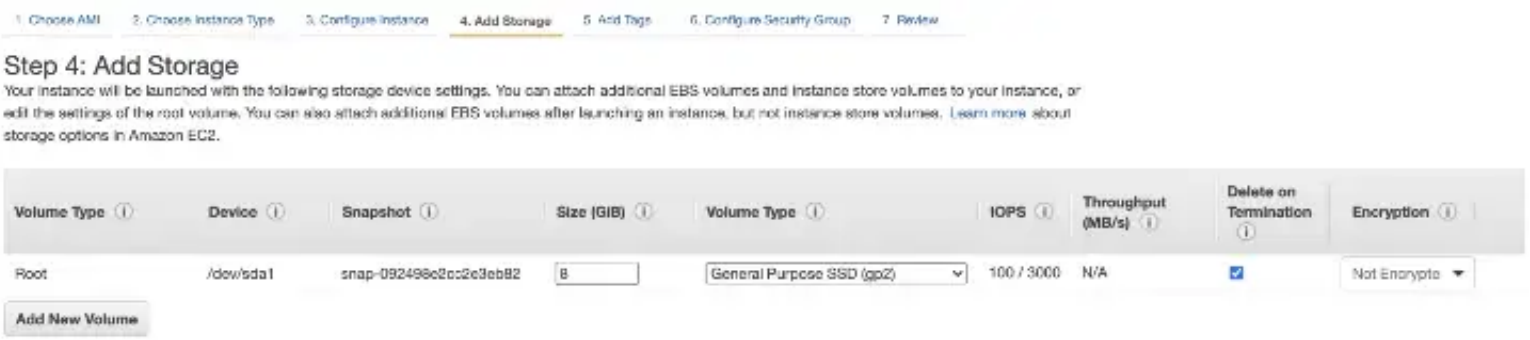
Next: Configure Instance Details

Provide the number of instances as 4 and select one one of the available subnet options so that all the 4 instances will be in the same region and leave the rest of the options as same



Next: Add Storage

By default 8 GB of storage will be allotted. If you need more storage you may Add New Volume or change the size based on the requirements. For Hadoop installation and learning purpose 8 GB will be sufficient.



Next: Add Tags

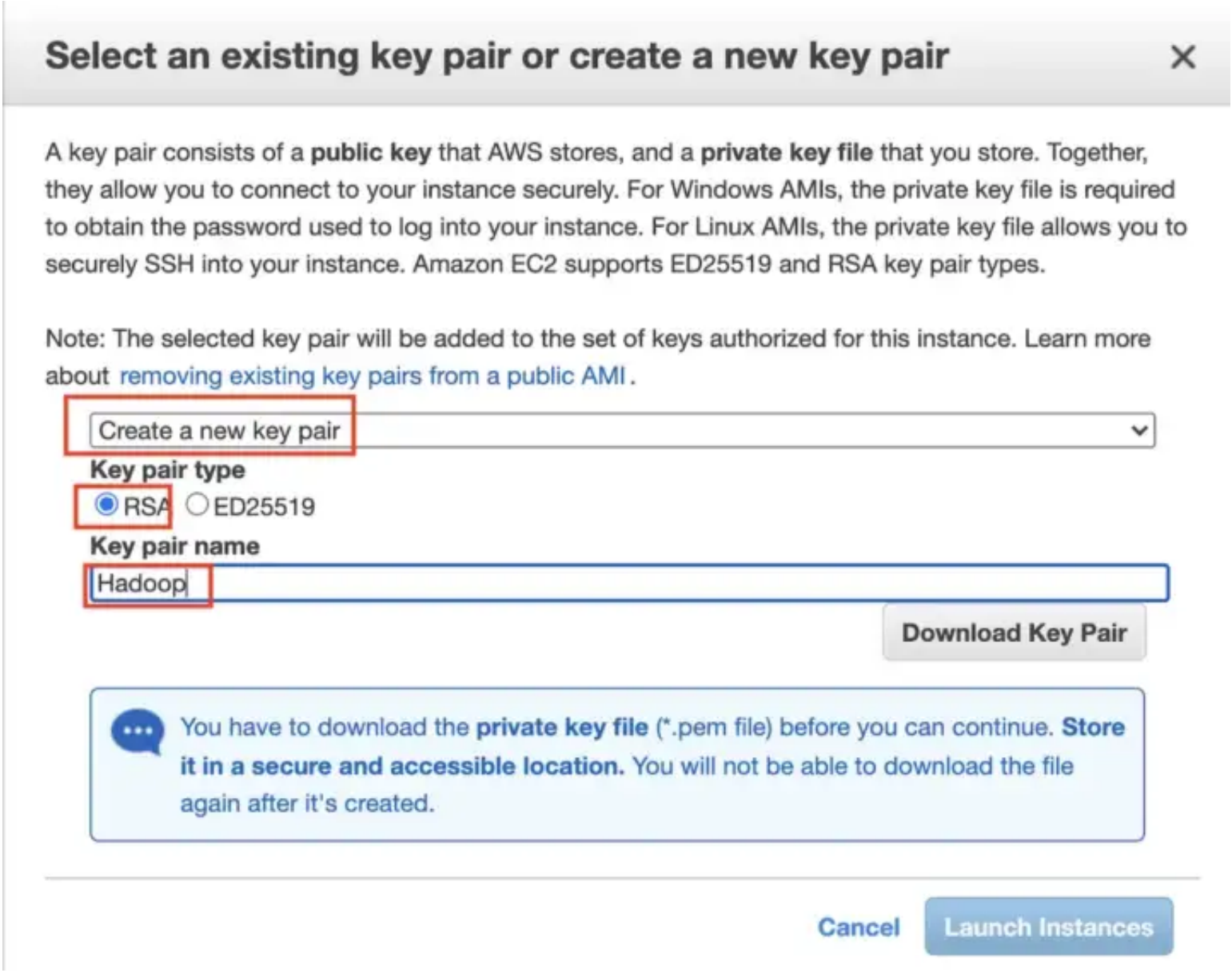
Give some tag to the instance to identify the instance with the name we choose, We can later change the Name “Hadoop” to **Namenode, SNN and Datanodes** once instances are launched

Next: Configure Security Group

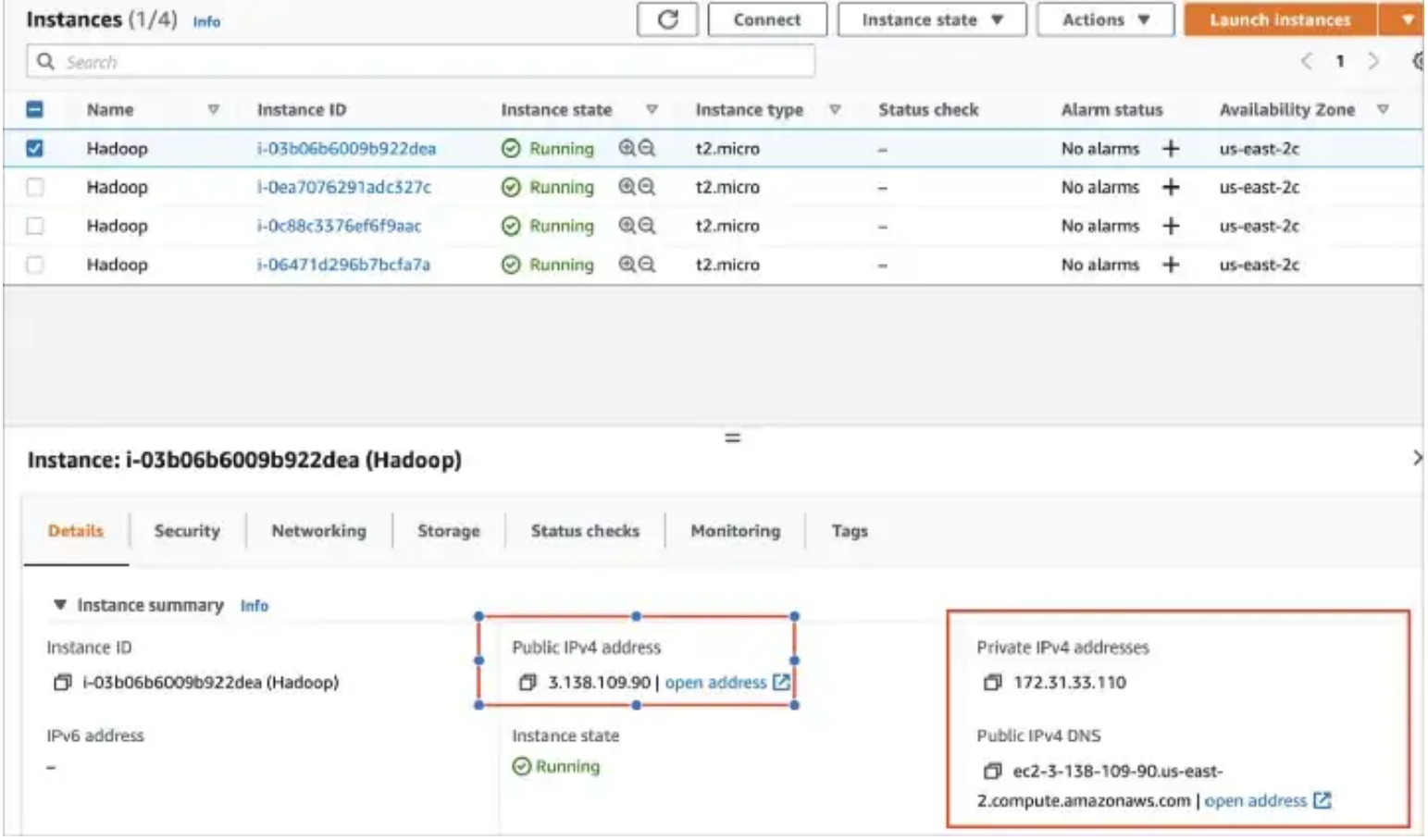
Select Create a new security group and type as **All traffic** so the security group will be open for testing.

Review and Launch : Review all the information selected again and proceed with **Launch** then you will be prompted with the below screen.

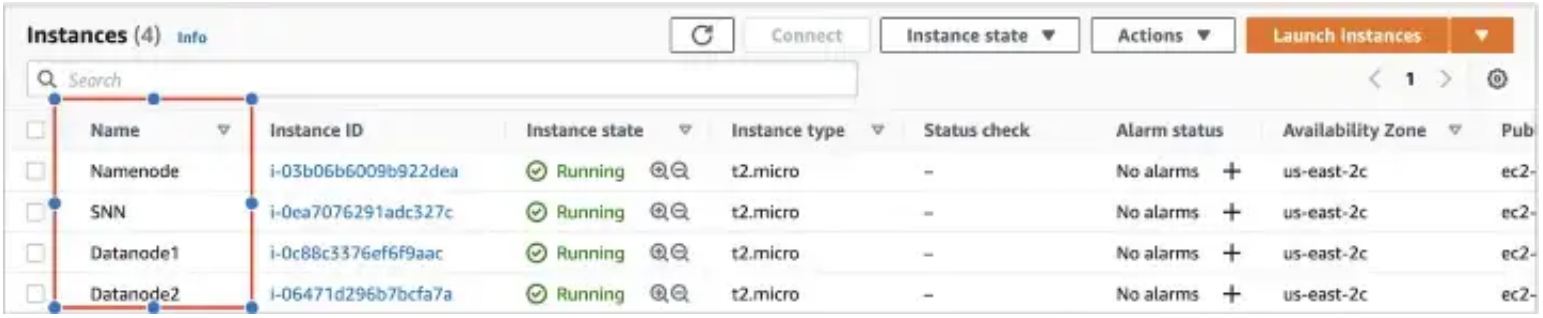
Select **Create new key pair** → Type as **RSA** → Give a name to the key pair and click on download the key-pair. It will download .pem file which is used to access the EC2 instances then **Launch Instances**



Click on **View instances** and select instance to see the properties through which we can see the ip address and dns of our instance.



Click on Edit option near to the Name and rename the instances as **Namenode, SNN, Datanode1 and Datanode2.**



Now that the instances are up and running, We need to setup Hadoop in these instances. Before setting up Hadoop in this instances make sure you are able to connect to these EC2 instances with SSH.

If you are not aware of how to SSH, follow any of the below article/video

Connecting to EC2 instance using SSH from Mac or Linux

Video on Creating AWS Account, launch EC2 instance and connect using SSH from Mac/ Linux

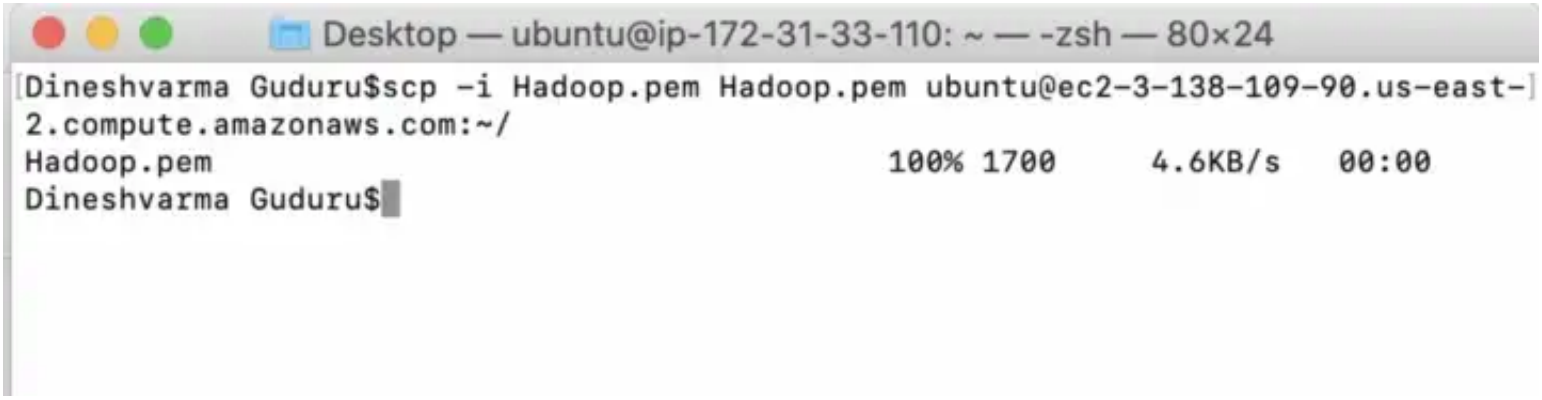
Video on establishing SSH to EC2 instance from Windows

Copy Keypair (xxxx.pem) from Local to Namenode

We need the Keypair (xxxx.pem) in the Namenode for the namenode to access the Datanodes

scp -i <path_to_keypair> <path_to_keypair> user@<ec2_namenode_dns>:<path_at_ec2>

```
scp -i Hadoop.pem Hadoop.pem ubuntu@ec2-3-138-109-90.us-east-
2.compute.amazonaws.com:~/
```



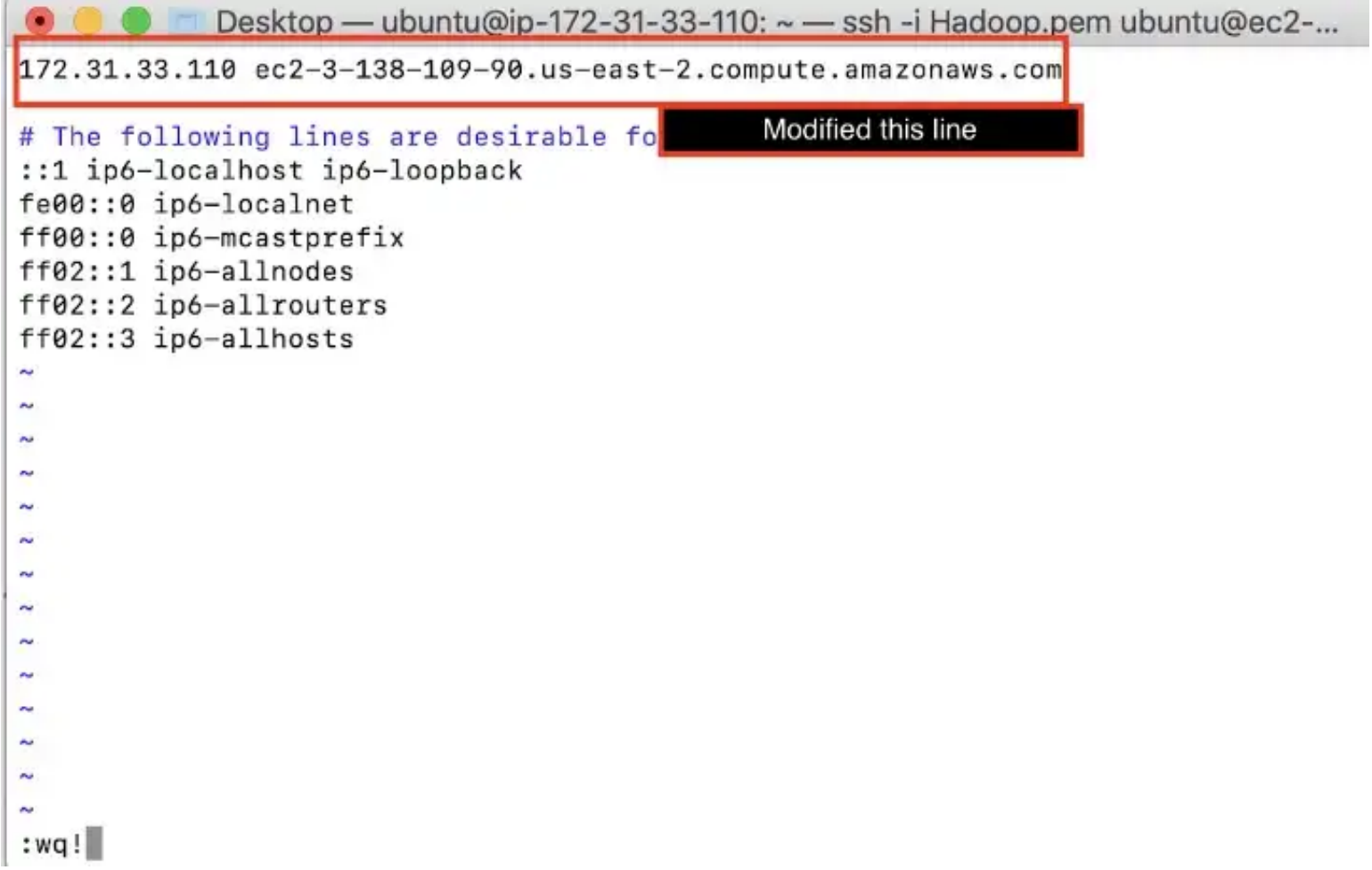
All Nodes:

The below activities to be performed in all the nodes one by one.

Edit the hosts File: Replace the first line with

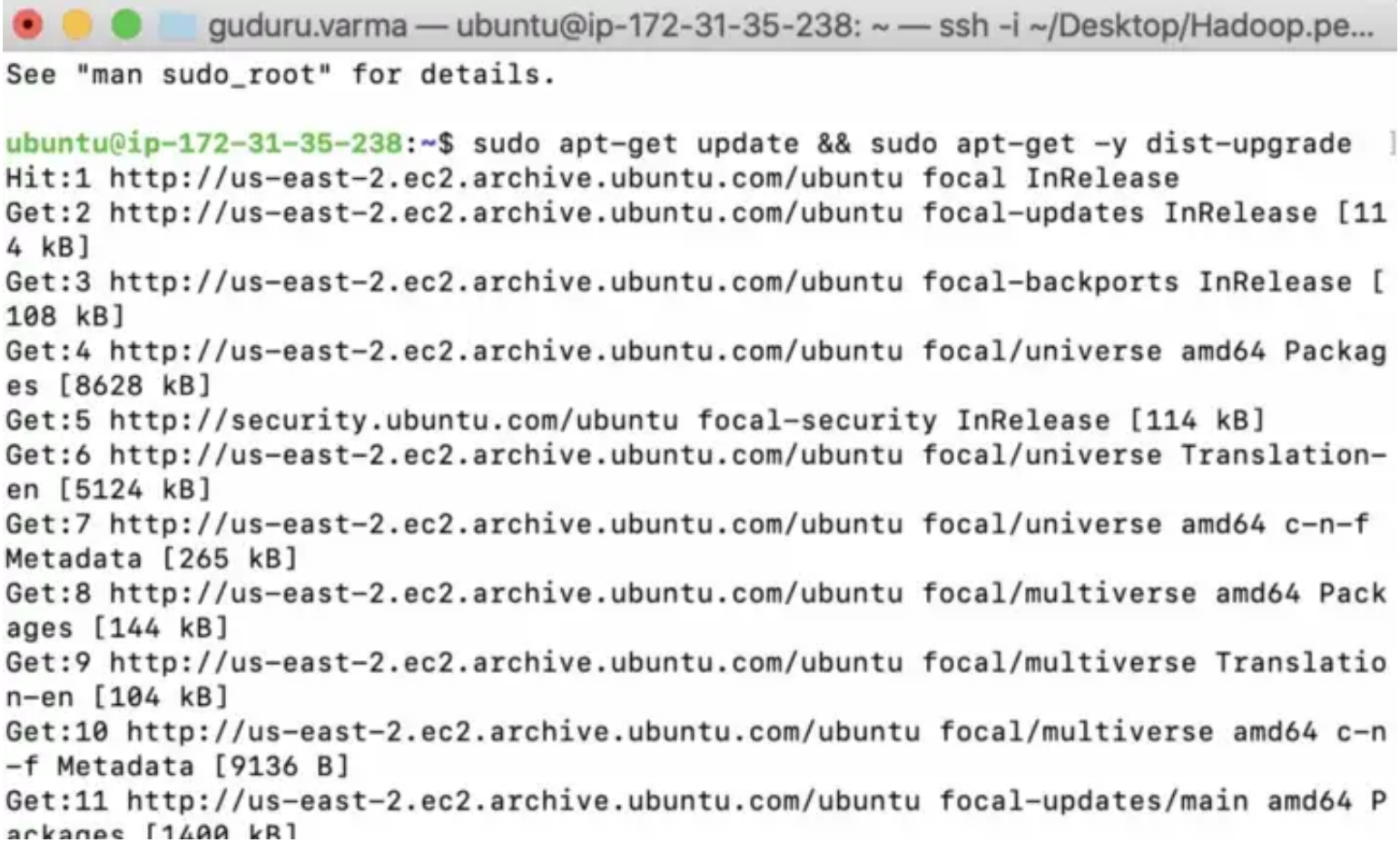
<ec2_private_ipv4> <ec2_dns>

```
ubuntu@ip-172-31-33-110:~$ sudo vi /etc/hosts
172.31.33.110 ec2-3-138-109-90.us-east-2.compute.amazonaws.com
```



Update OS with latest available patches in all nodes

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



Install Java in all nodes

```
sudo apt-get -y install openjdk-8-jdk-headless
```

```
guduru.varma — ubuntu@ip-172-31-35-238: ~ — ssh -i ~/Desktop/Hadoop.pe...
ubuntu@ip-172-31-35-238:~$ sudo apt-get -y install openjdk-8-jdk-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1 libxi6
  libxrender1 libxtst6 openjdk-8-jre-headless x11-common
Suggested packages:
  default-jre cups-common liblcms2-utils pcscd openjdk-8-demo openjdk-8-source
  libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common
  libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1 libxi6
  libxrender1 libxtst6 openjdk-8-jdk-headless openjdk-8-jre-headless
  x11-common
0 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 38.4 MB of archives.
After this operation, 150 MB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 java-commo
n all 0.72 [6816 B]
```

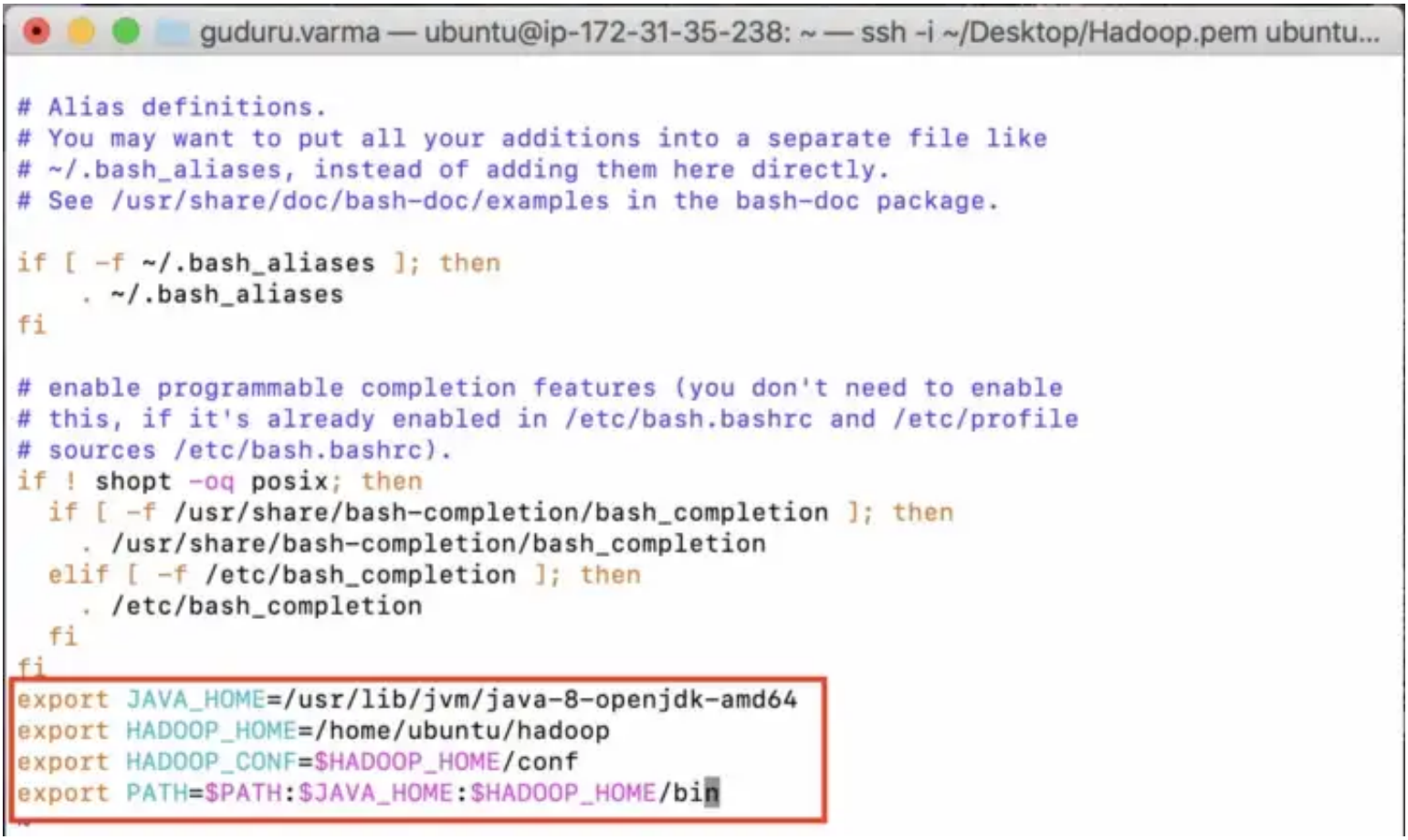
Download and install Hadoop in all nodes

step1: wget https://downloads.apache.org/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz
step2: tar xvzf hadoop-3.2.2.tar.gz
step3: mv hadoop-3.2.2 hadoop

```
ubuntu@ip-172-31-35-238:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.2.2/
hadoop-3.2.2.tar.gz
--2021-12-28 02:08:24--  https://downloads.apache.org/hadoop/common/hadoop-3.2.2/hadoop-
3.2.2.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.95.219,
2a01:4f8:10a:201a::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... connec
ted.
HTTP request sent, awaiting response... 200 OK
Length: 395448622 (377M) [application/x-gzip]
Saving to: 'hadoop-3.2.2.tar.gz'

hadoop-3.2.2.tar.gz   100%[=========================>] 377.13M  11.2MB/s    in 32s

2021-12-28 02:08:57 (11.7 MB/s) - 'hadoop-3.2.2.tar.gz' saved [395448622/395448622]
```

Modify **~/.bashrc** in all nodes and add below lines at the end

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/ubuntu/hadoop
export HADOOP_CONF=$HADOOP_HOME/conf
export PATH=$PATH:$JAVA_HOME:$HADOOP_HOME/bin
```



Now run ***source ~/.bashrc*** *in all nodes*

Run **hadoop** command to check if hadoop is installed correctly in all nodes

You should now see something like below, which means hadoop is installed correctly.

```
ubuntu@ip-172-31-33-110:~$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
 or    hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
  where CLASSNAME is a user-provided Java class

  OPTIONS is none or any of:

--config dir                     Hadoop config directory
--debug                          turn on shell script debug mode
--help                           usage information
buildpaths                       attempt to add class files from build tree
hostnames list[,of,host,names]   hosts to use in slave mode
hosts filename                   list of hosts to use in slave mode
loglevel level                   set the log4j level for this command
workers                          turn on worker mode
```

Setup Password-less SSH on servers

On **Namenode** run **ssh-keygen** and use the default location for the key and hit enter twice for an empty passphrase. The rsa_id will now be available on Namenode as */home/ubuntu/.ssh/id_rsa.pub*

```
ubuntu@ip-172-31-35-238:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kUW2j4I3YKWK2m57GseqhnJ4pBv43pnGdO5KEQk+KgA ubuntu@ip-172-31-35-238
The key's randomart image is:
+---[RSA 3072]----+
|E .     ..+      |
|.. . . o + .     |
|. o o + o .      |
|.. o + o . o     |
|o . o . S . .    |
|oo..... o        |
|=+oo+o           |
|==+*+o.          |
|=BB==o.          |
+----[SHA256]-----+
```

Copy the above file to all the Nodes. Command to copy is shown below which needs to be executed in Namenode thrice, once for each node.

```
scp -i Hadoop.pem /home/ubuntu/.ssh/id_rsa.pub ubuntu@ec2-3-139-95-172.us-
east-2.compute.amazonaws.com:/home/ubuntu/.ssh/id_rsa.pub
```

On **Datanodes,** Move the rsa_id to the authorized keys

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

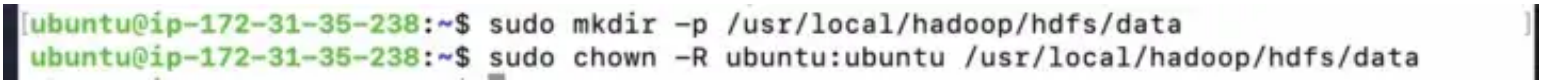
Setup SSH config in Namenode: Add below lines in ~/.ssh/config

vi ~/.ssh/config

```
Host nnode
  HostName <nnode dns>
  User ubuntu
  IdentityFile ~/.ssh/id_rsa

Host snnode
  HostName <snn dns>
  User ubuntu
  IdentityFile ~/.ssh/id_rsa

Host dnode2
  HostName <dnode1 dns>
  User ubuntu
  IdentityFile ~/.ssh/id_rsa

Host dnode3
  HostName <dnode2 dns>
  User ubuntu
  IdentityFile ~/.ssh/id_rsa
```

Create data directories in all nodes

```
sudo mkdir -p /usr/local/hadoop/hdfs/data
sudo chown -R ubuntu:ubuntu /usr/local/hadoop/hdfs/data
```



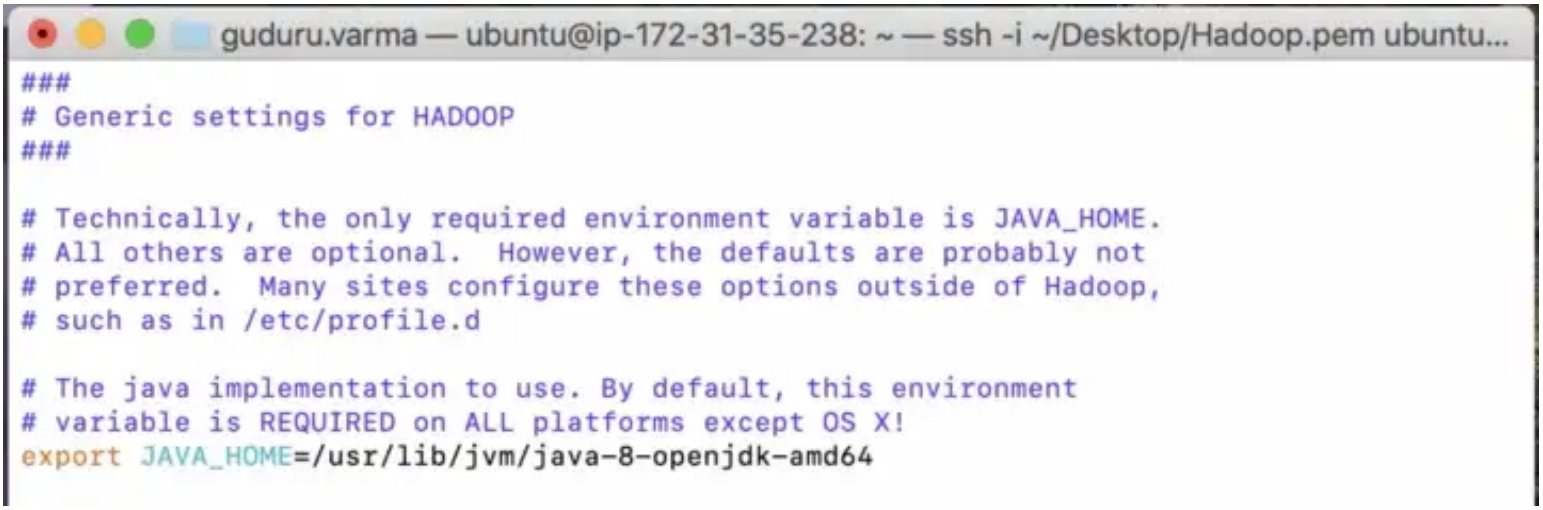
Set up Hadoop cluster

Hadoop cluster needs below 7 files to be configured

```
$HADOOP_HOME/etc/hadoop/hadoop-env.sh
$HADOOP_HOME/etc/hadoop/core-site.xml
$HADOOP_HOME/etc/hadoop/hdfs-site.xml
$HADOOP_HOME/etc/hadoop/mapred-site.xml
$HADOOP_HOME/etc/hadoop/yarn-site.xml
$HADOOP_HOME/etc/hadoop/masters
$HADOOP_HOME/etc/hadoop/slaves
```

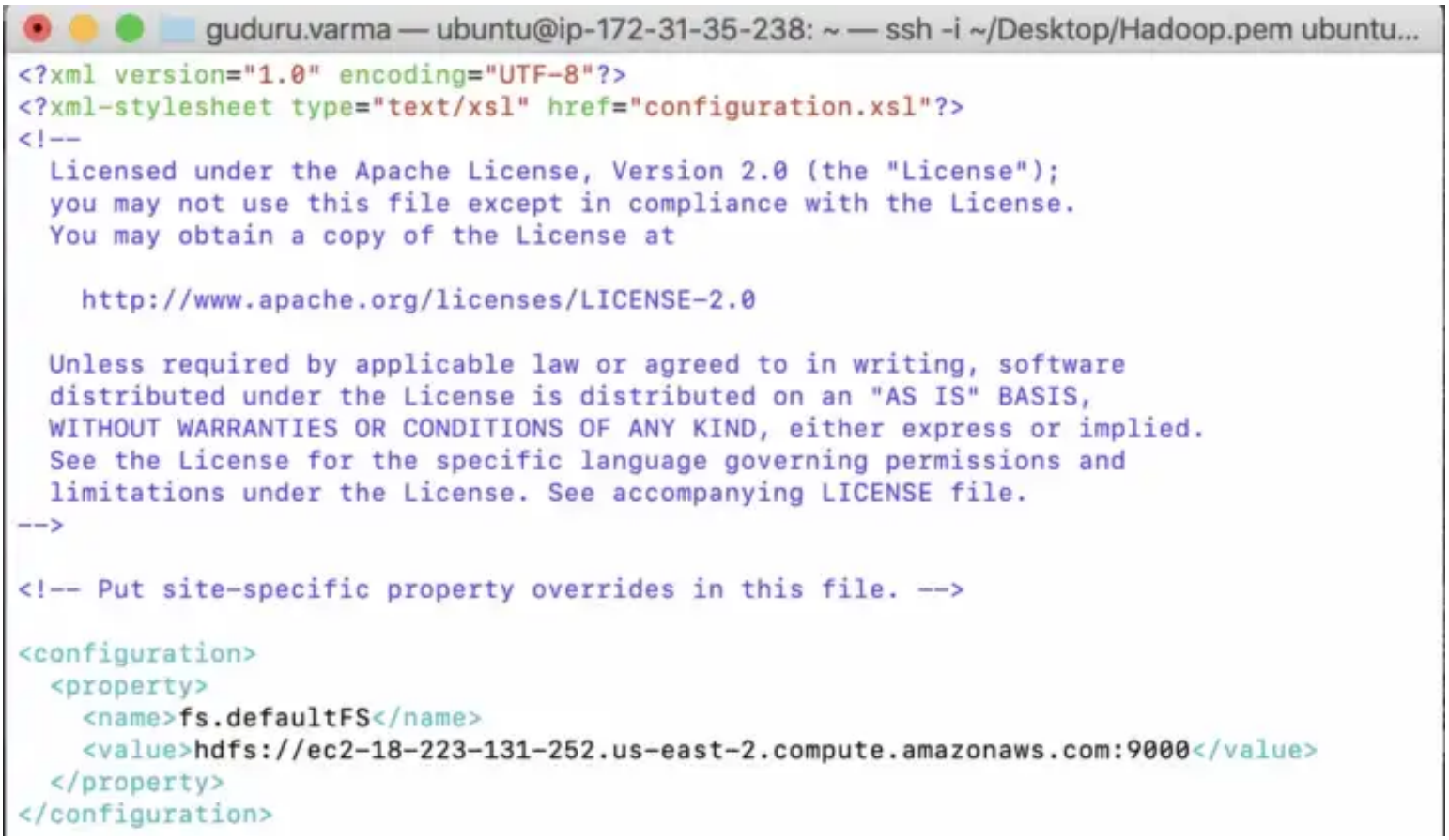
Add below line in **$HADOOP_HOME/etc/hadoop/hadoop-env.sh**

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```



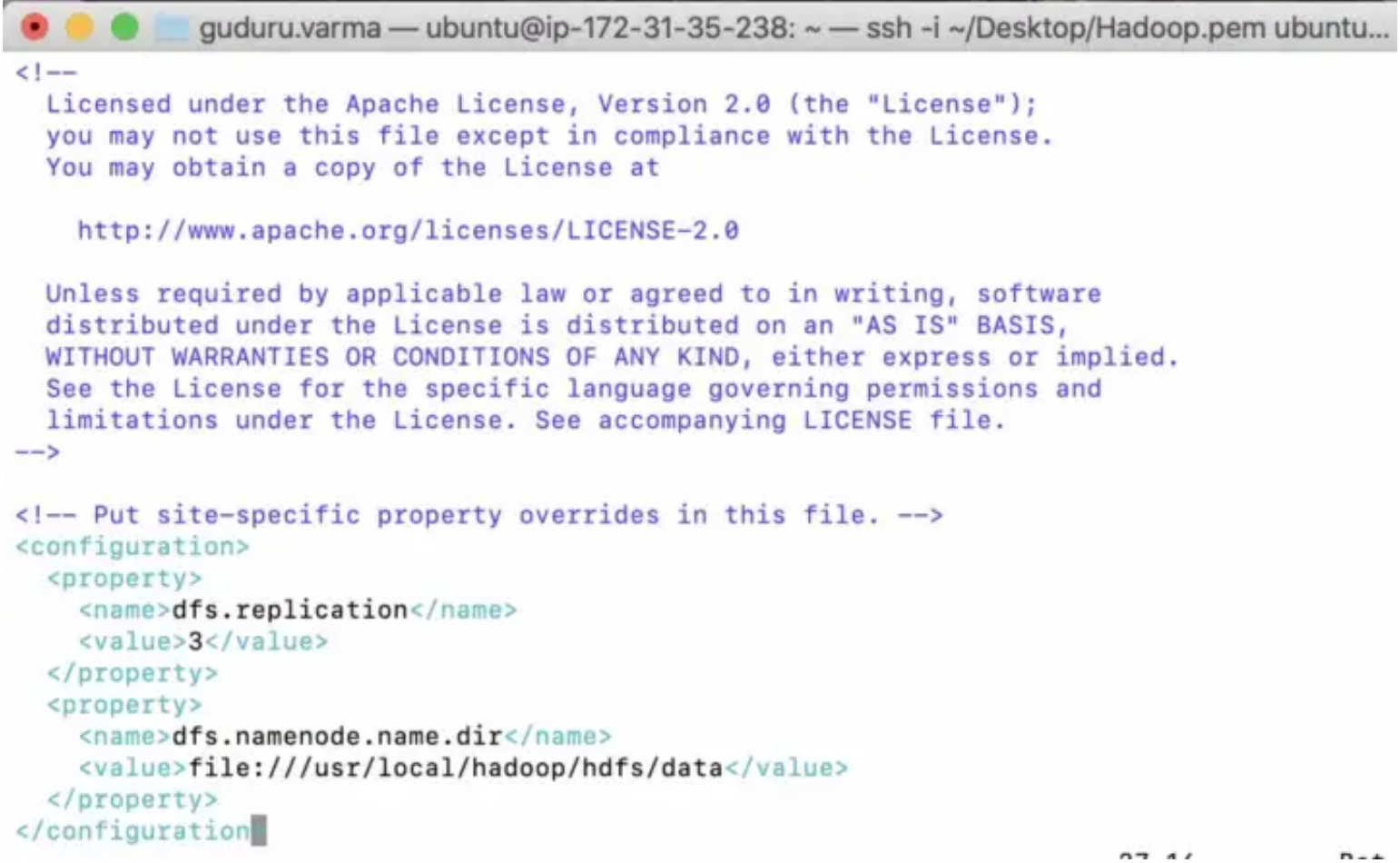
Add below lines in **$HADOOP_HOME/etc/hadoop/core-site.xml**

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://<nnode>:9000</value>
  </property>
</configuration>
```



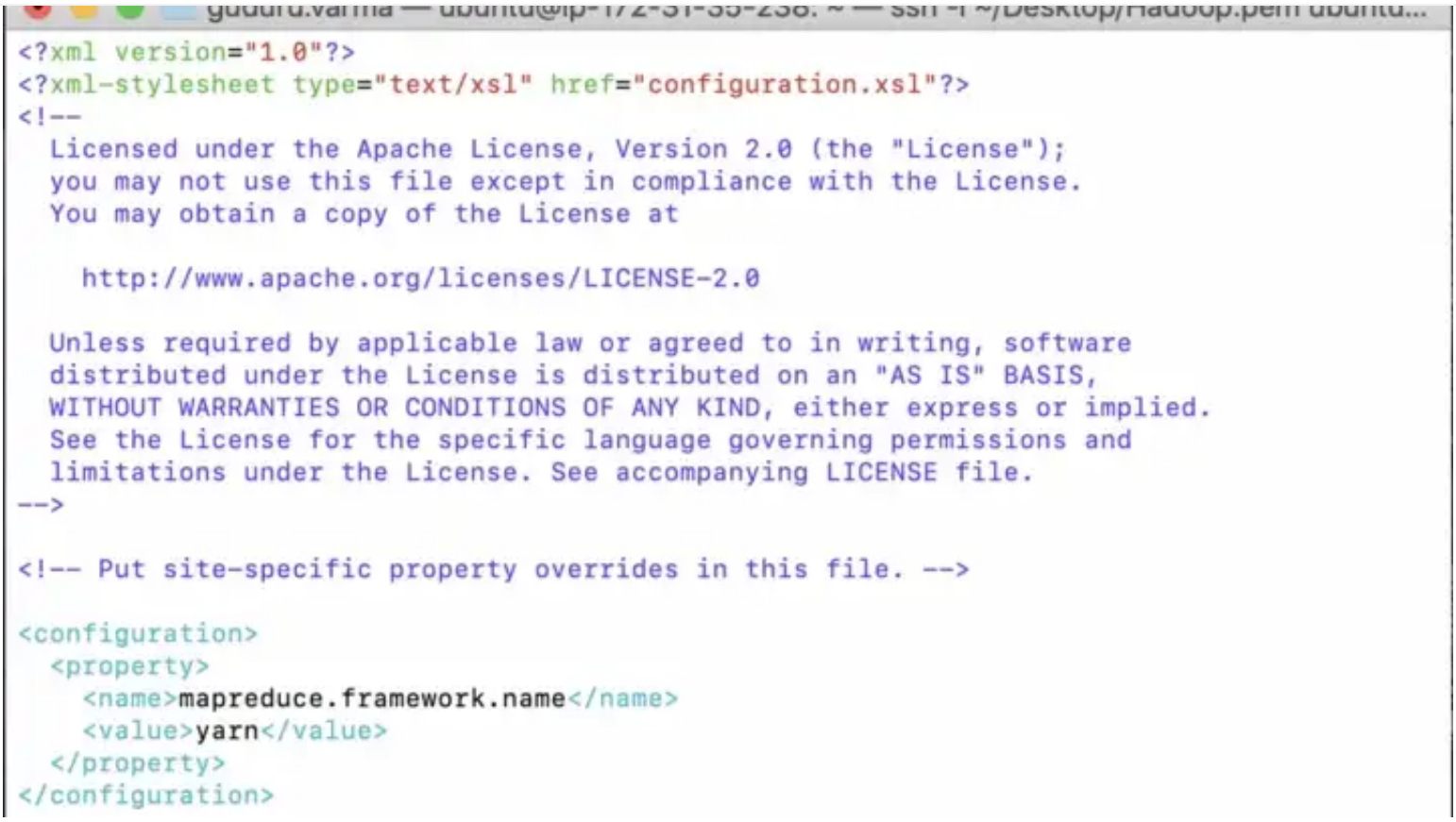
Add below lines in **$HADOOP_HOME/etc/hadoop/hdfs-site.xml**

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///usr/local/hadoop/hdfs/data</value>
  </property>
</configuration>
```



Add below lines in **$HADOOP_HOME/etc/hadoop/mapred-site.xml**

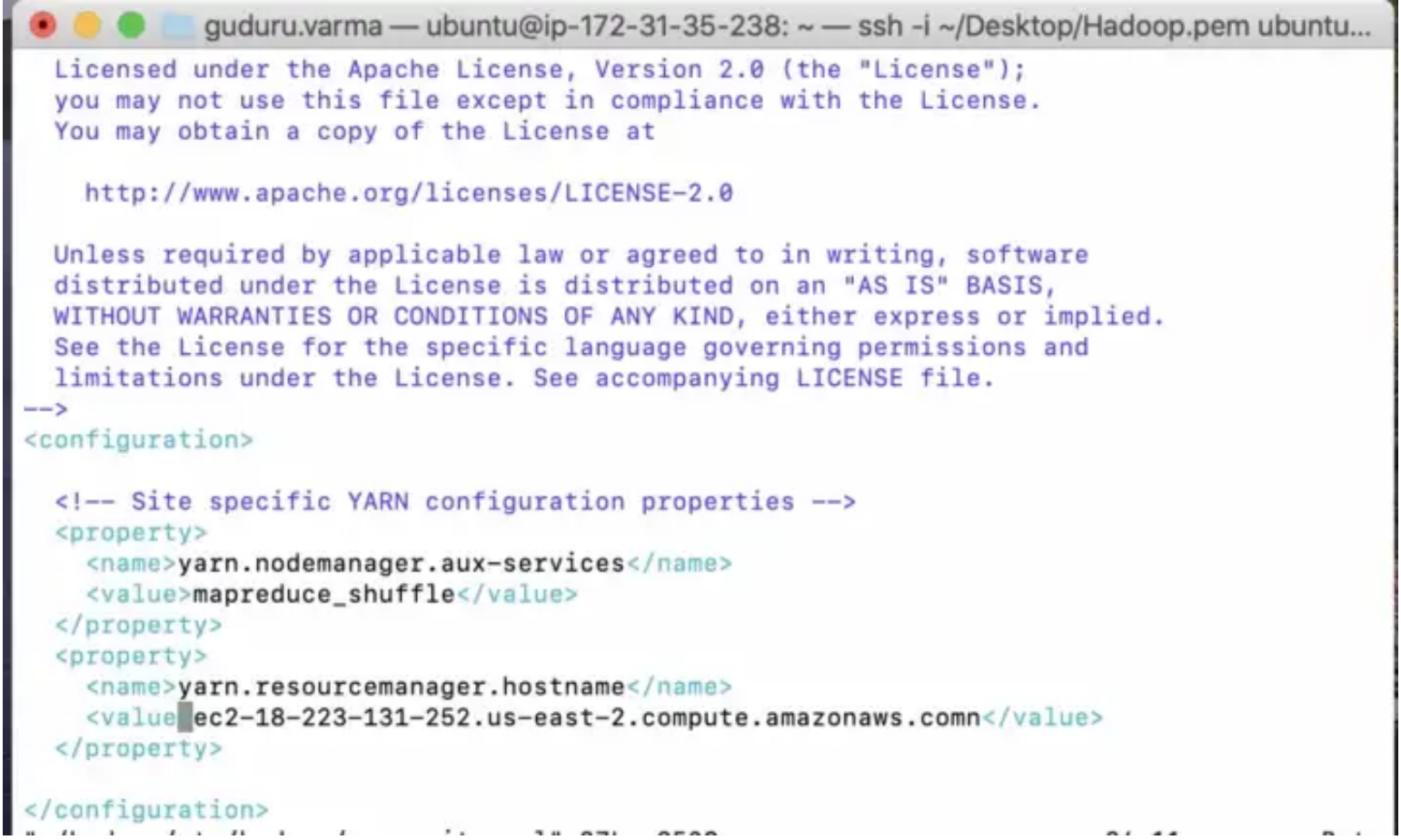
```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

Add below lines in **$HADOOP_HOME/etc/hadoop/yarn-site.xml**

```
  <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><nnode></value>
    </property>

  </configuration>
```



Move configuration files to Slaves

scp hadoop-env.sh core-site.xml hdfs-site.xml mapred-site.xml yarn-site.xml <user>@<dnode>:/home/ubuntu/hadoop/conf

```
scp hadoop-env.sh core-site.xml hdfs-site.xml mapred-site.xml yarn-site.xml
ubuntu@ec2-54-209-221-47.compute-1.amazonaws.com:/home/ubuntu/hadoop/conf
```

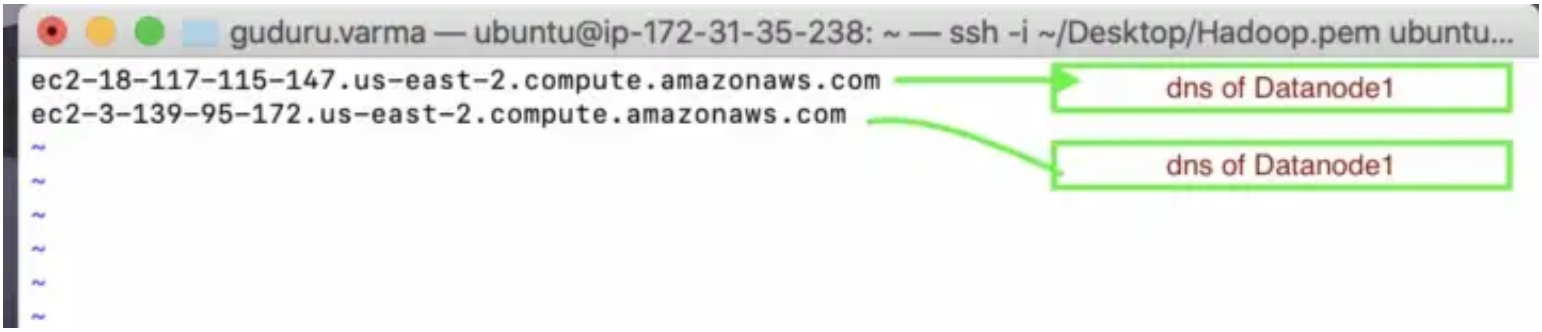
Now on **Namenode and Secondary Namenode** edit below 2 files

Add below lines in **$HADOOP_HOME/etc/hadoop/masters**

```
<nnode>
<snnode>
```

Add below lines in **$HADOOP_HOME/etc/hadoop/slaves**

```
<dnode1>
<dnode2>
```



Datanode

$HADOOP_HOME/etc/hadoop/masters will be empty in Datanodes

Add below lines in **$HADOOP_HOME/etc/hadoop/slaves**

```
<dnode_dns>
```

Each Datanode will have only it's own dns in **slaves** file

Hadoop Daemon Startup

```
Namenode: hdfs namenode -format
Namenode: $HADOOP_HOME/sbin/start-dfs.sh
Namenode: $HADOOP_HOME/sbin/start-yarn.sh
Namenode: $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
```

```
ubuntu@ip-172-31-35-238:~$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [ec2-18-223-131-252.us-east-2.compute.amazonaws.com]
ec2-18-223-131-252.us-east-2.compute.amazonaws.com: Warning: Permanently added 'ec2-18-2
23-131-252.us-east-2.compute.amazonaws.com' (ECDSA) to the list of known hosts.
Starting datanodes
Starting secondary namenodes [ip-172-31-35-238]
ubuntu@ip-172-31-35-238:~$ $HADOOP_HOME/sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
ubuntu@ip-172-31-35-238:~$ jps
2432 Jps
2313 NodeManager
1802 DataNode
1998 SecondaryNameNode
```

Now that we have started Hadoop in Cluster mode and will be able to check the services running in all the nodes by using jps command as shown above

Once the cluster is running, we can check the web UI for the status of the data nodes. Go to <nnode>:50070 for the Web UI and verify that the datanodes added are online

Hadoop services can be stopped using **$HADOOP_HOME/sbin/stop-all.sh** command.

```
ubuntu@ip-172-31-35-238:~$ $HADOOP_HOME/sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as ubuntu in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [ec2-18-223-131-252.us-east-2.compute.amazonaws.com]
Stopping datanodes
Stopping secondary namenodes [ip-172-31-35-238]
Stopping nodemanagers
localhost: WARNING: nodemanager did not stop gracefully after 5 seconds: Trying to kill
with kill -9
Stopping resourcemanager
```

More detailed video demonstration is available in this link.

Multinode Cluster Hadoop Cluster Hadoop Cluster On Ec 2 Hadoop On Aws Aws Ec 2 Hadoop