**Spark on Hadoop Multinode Cluster**

Here I’m recording critical steps on achieving movie recommendation on spark on multiple AWS EC2 instances. Totally there are 3 primary steps as following:

**1.Set up multi node hadoop cluster on 4 ubuntu instances:**

* Namenode (master)
* SecondaryNamenode (back up)
* Datanode (slave 1)
* Datanode (slave 2)

**2. Installing spark and run pyspark in both ‘local’ and ‘yarn’ mode**

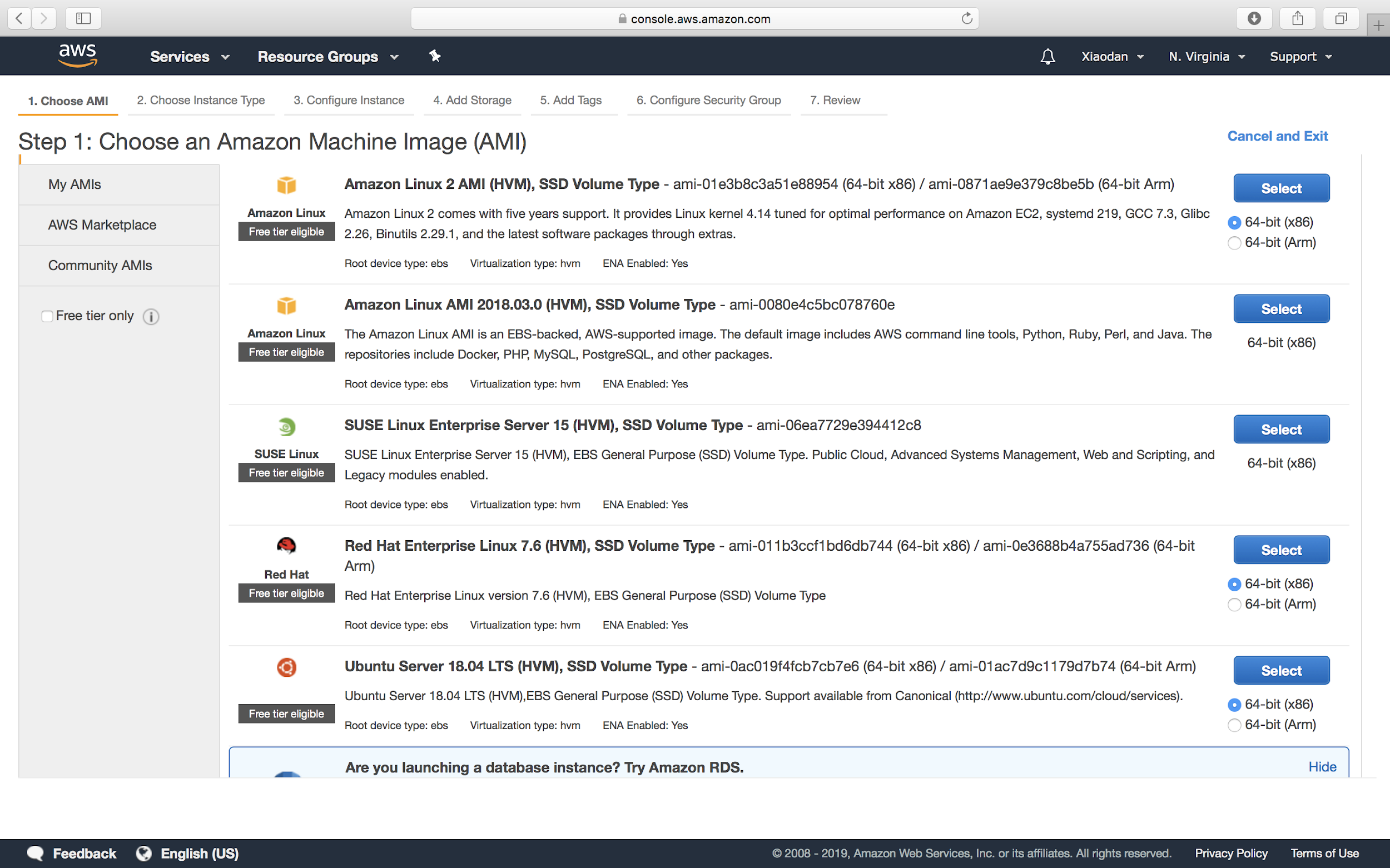
**3. Achieving movie recommendation with Spark Mllib ALS algorithm**

Here we go.

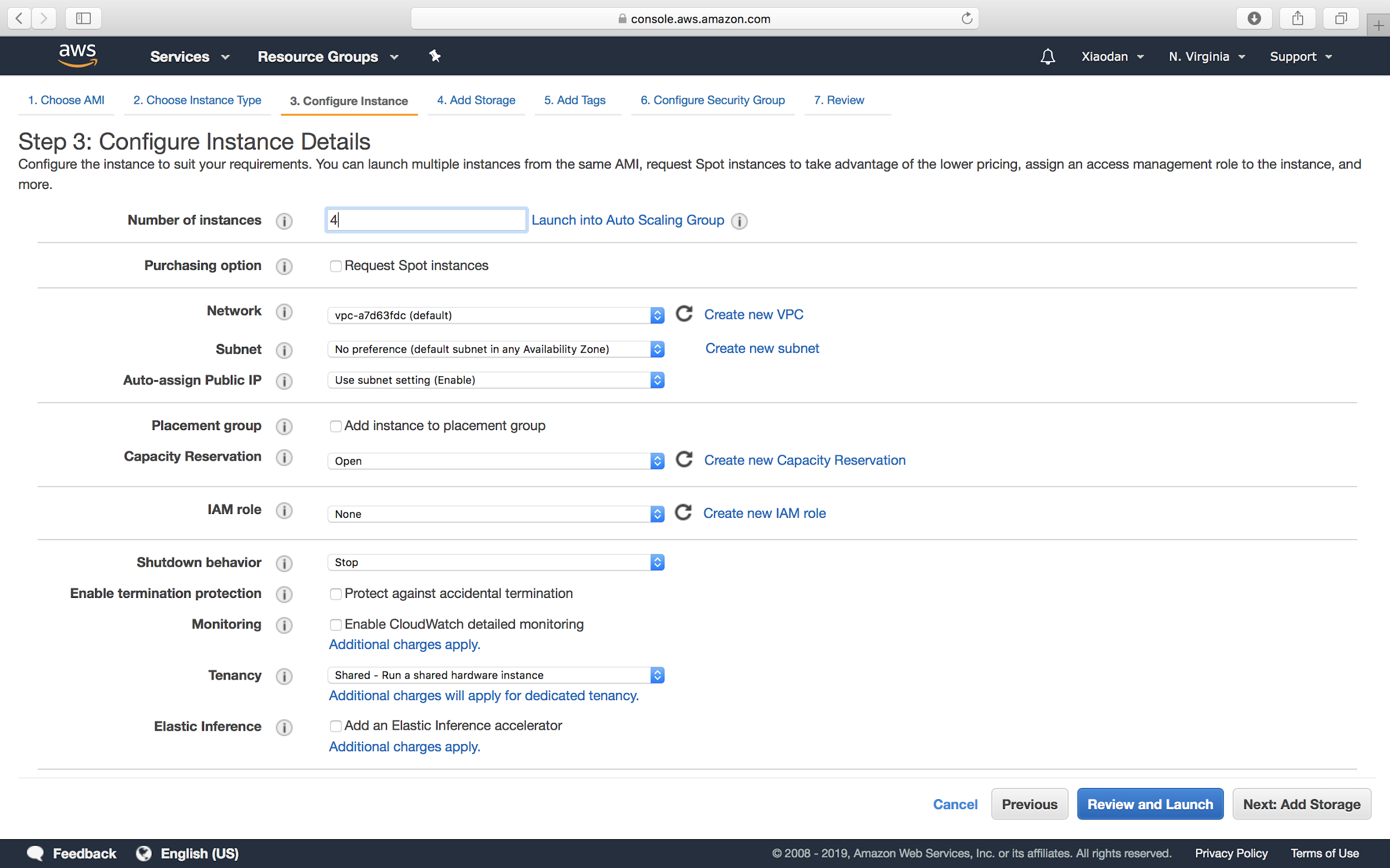
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.Set up multi node hadoop cluster on 4 ubuntu instances:**

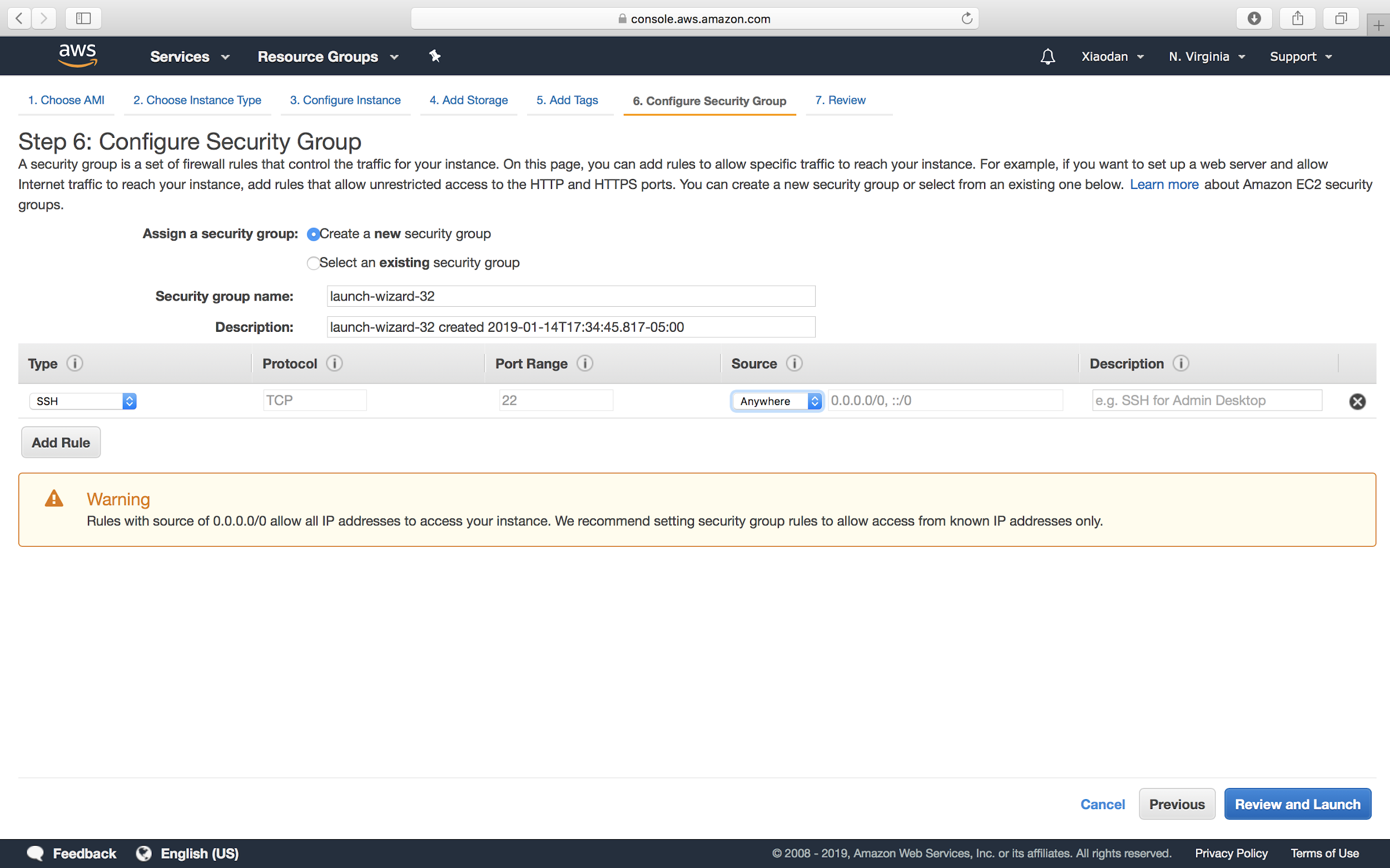
1.1 Choosing AWS EC2 Ubuntu Server 18.04



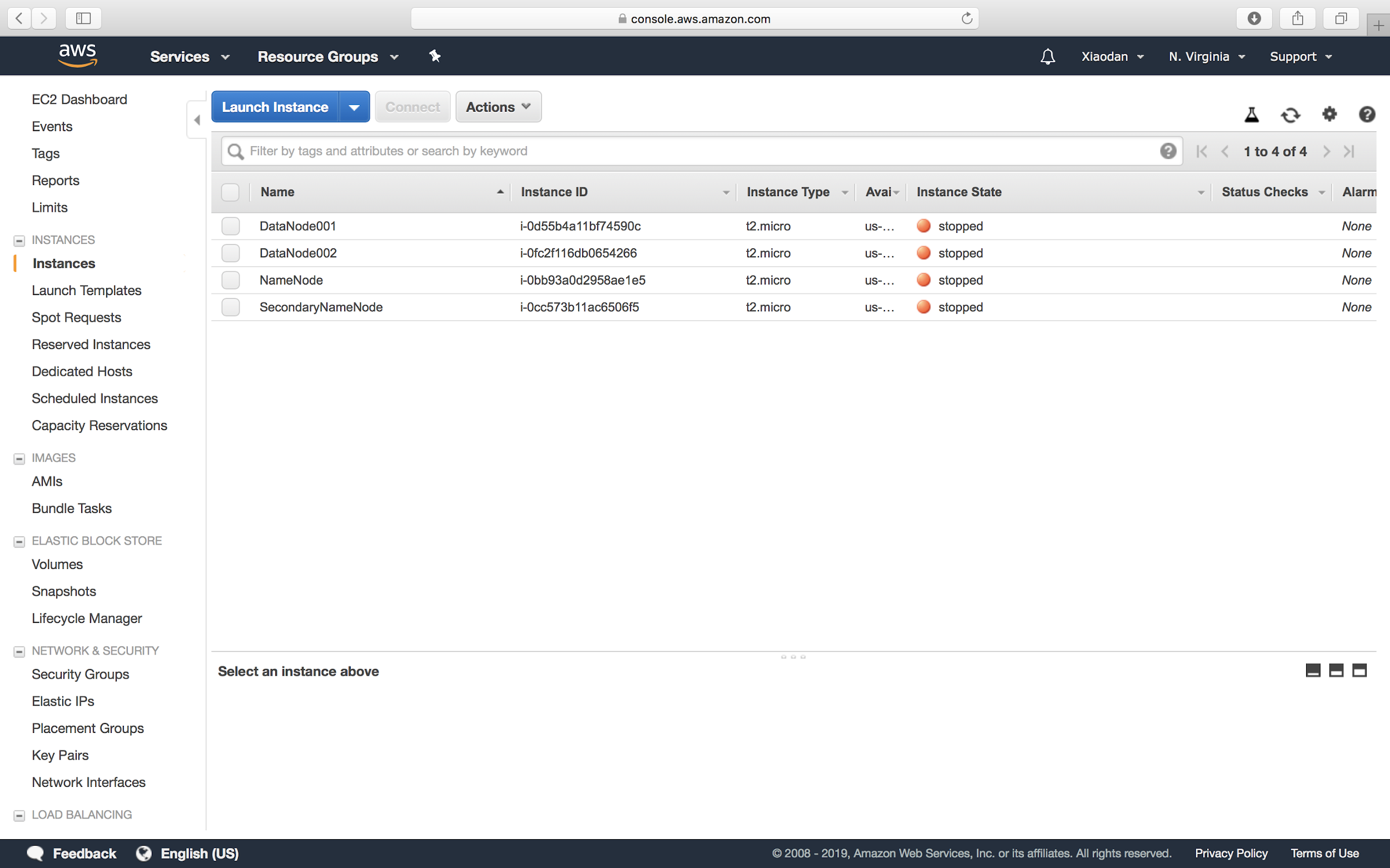
1.2 Configure 4 instances



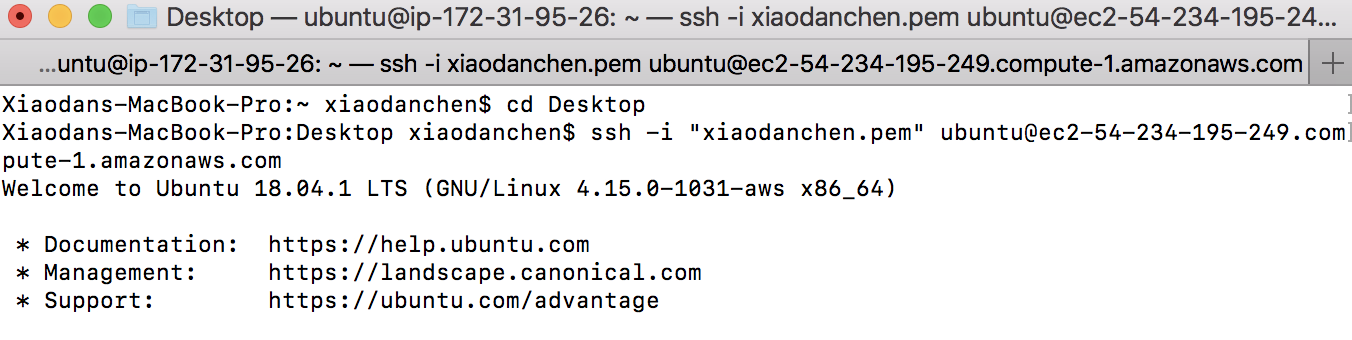
1.3 Configure security group



1.4 Rename 4 instances and get it started later on



1.5 SSH to these 4 instances



1.6 Pre-installations for the cluster

$ sudo apt-get update

# install java

$ sudo add-apt-repository -y ppa:webupd8team/java

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

# install hadoop

$ wget {hadoop from Apache download page}

$ tar -xzvf hadoop-1.2.1.tar.gz

$ mv hadoop-1.2.1 hadoop

# set environment variable

$ sudo vim ~/.bashrc

$ export PATH=$PATH:{}

$ source ~/.bashrc

# ssh

$ ssh-keygen -t dsa -P ‘’ -f ~/.ssh/id\_dsa

$ ll ~/.ssh

$ cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

$ chmod 644 authorized\_keys

# remote ssh

$ eval `ssh-agent -s`

$ ssh-add ~/xiaodanchen.pem

# hadoop cluster configuration

$ cd ~/hadoop/etc/hadoop

# configure these files:

* Hadoop-env.sh
* Core-site.xml
* Mapred-site.xml
* Hdfs-site.xml

$ sudo vim masters # add two namenodes ip to masters

$ sudo vim slaves # add two datanode ip to slaves

# copy these files to SecondaryNamenode

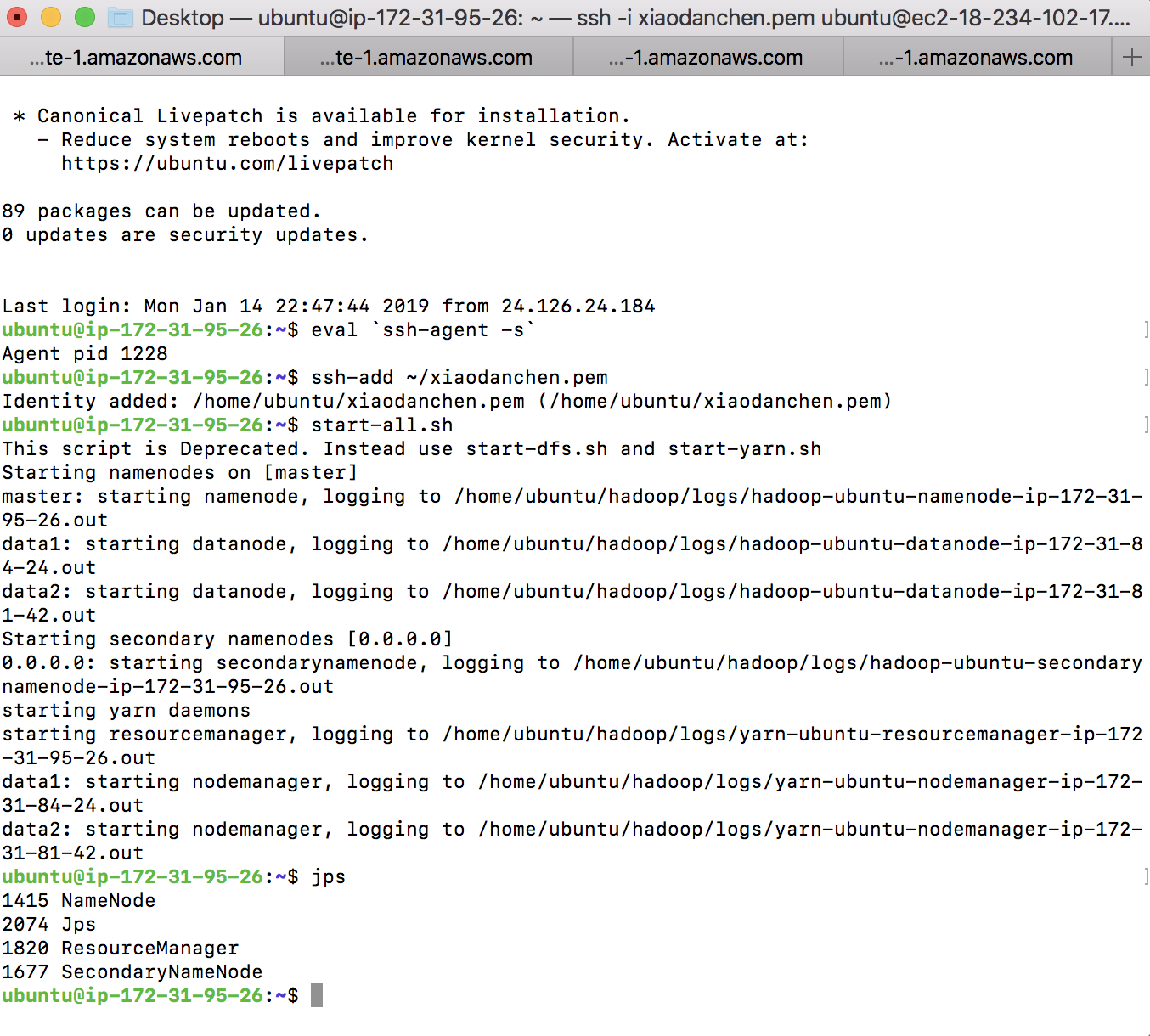
# for slave 1: leave masters blank, only copy datanode 1 ip tp slaves

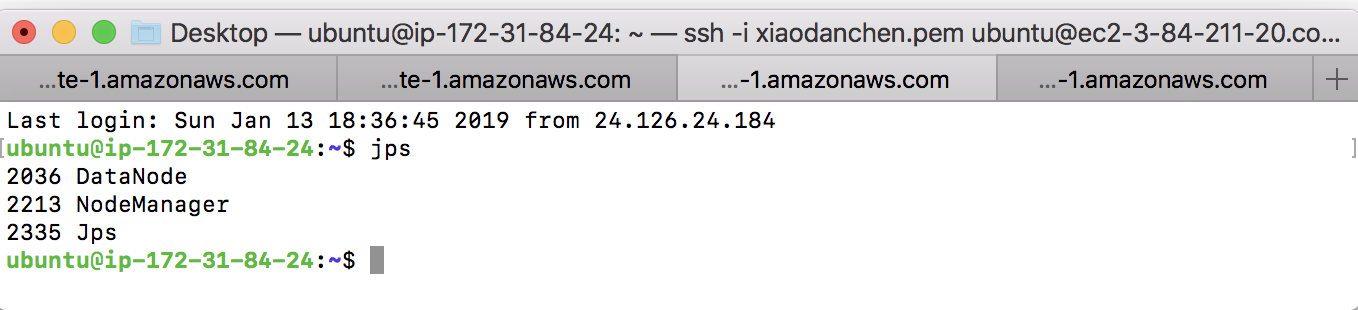
# for slave 2: leave masters blank, only copy datanode 2 ip tp slaves

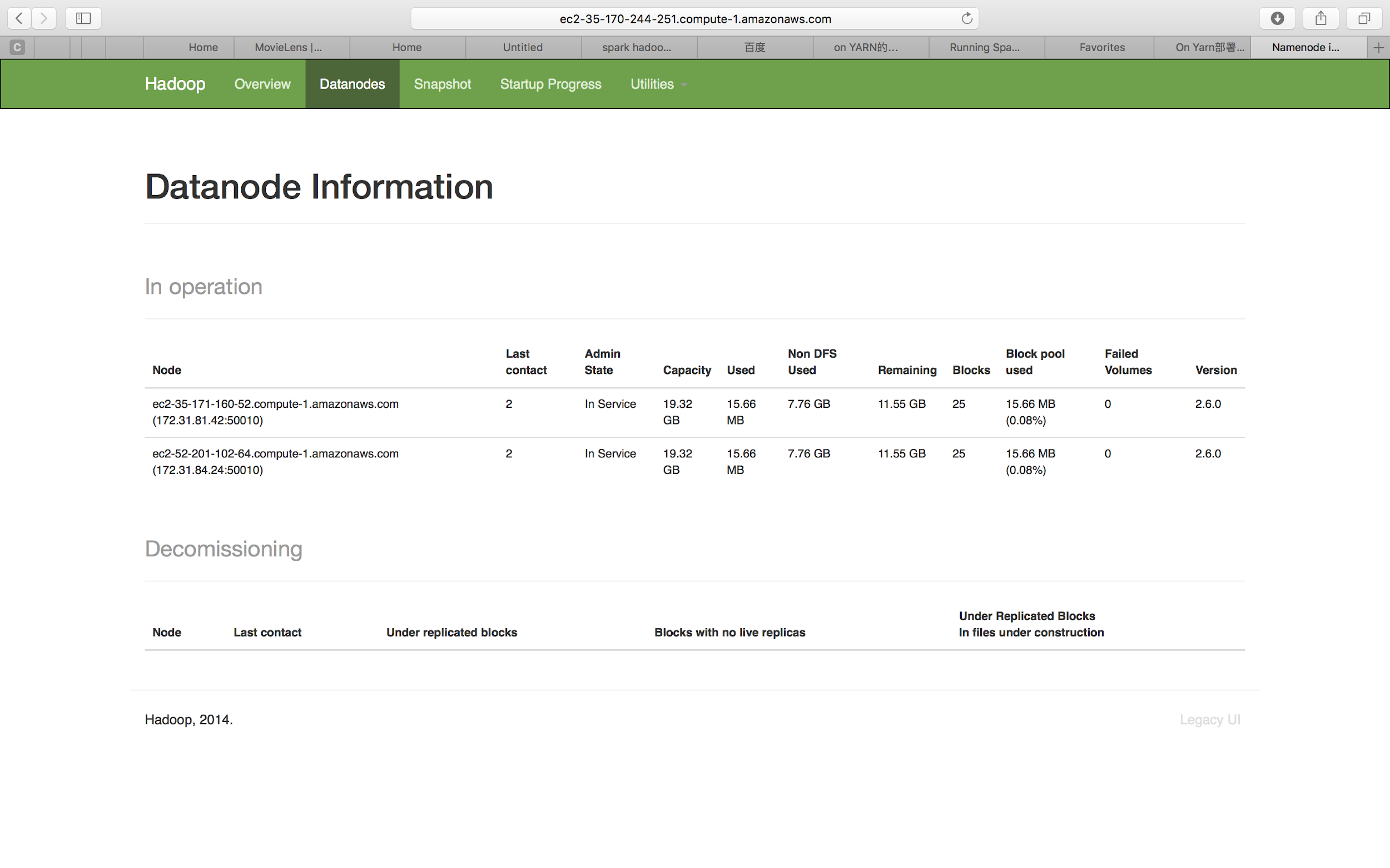
# hadoop daemon startup

$ hadoop namenode -format

$ start-all.sh # I’ve already set environment variable







**2. Installing spark and run pyspark**

# install scala

$ wget <http://www.scala-lang.org/files/archive/scala-2.11.6.tgz>

$ tar xvf scala-2.11.6.tgz

$ sudo mv scala-2.11.6 ~/scala

# set environment variable

$ sudo ~/.bashrc

$ export SCALA\_HOME=/home/ubuntu/scala

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

$ source ~/.bashrc

# install spark

$ wget {Spark download page}

$ tar zxf spark-xxx.tgz

$ sudo mv spark-xxx.tgz spark

# set environment variable

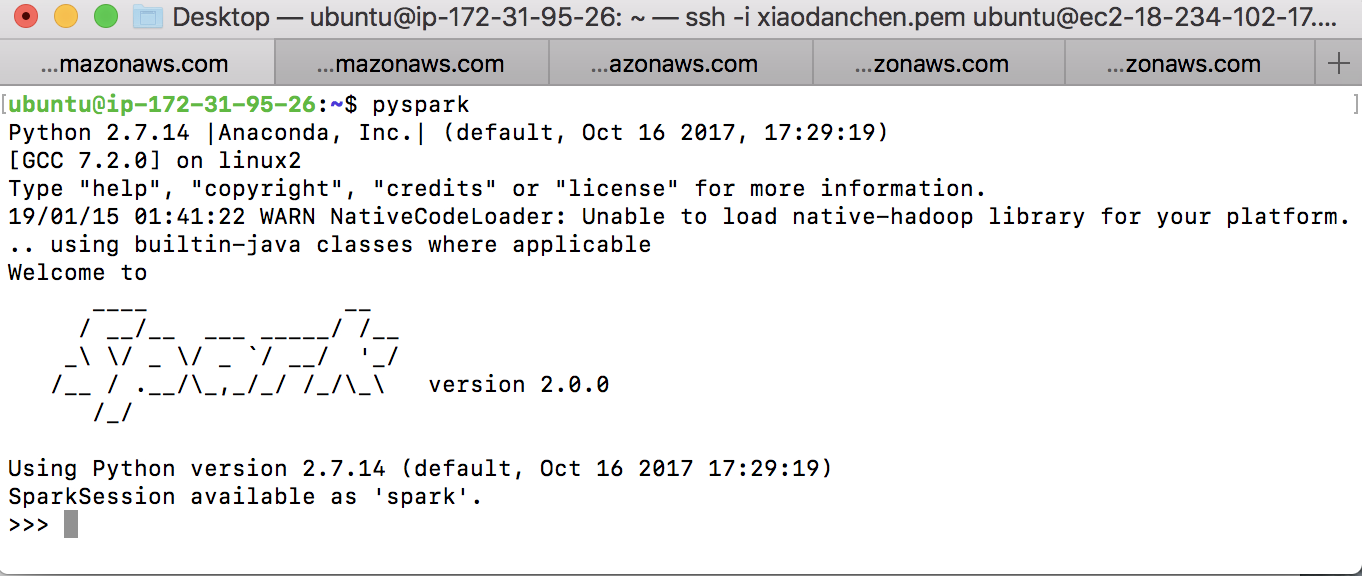
$ sudo ~/.bashrc

$ export SPARK\_HOME=/home/ubuntu/scala

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

$ source ~/.bashrc

# start pyspark



# open jupyter notebook

# install python, pip, jupyter firstly

$ jupyter notebook --no-browser --port=8888 --ip=0.0.0.0

**3. Achieving movie recommendation with Spark Mllib ALS algorithm**

Here I’d download data from MovieLens, and see some data stats.

* The first column would be the ‘user\_id’, as we can see, there are 6040 unique users
* The second column would be the ‘movie\_id’ for like unique 3952 movies
* The third column would be ‘rating\_score’, there are like 1 million ratings here
* Other columns are ignored

For the recommendation process, I will put the code in the code file

