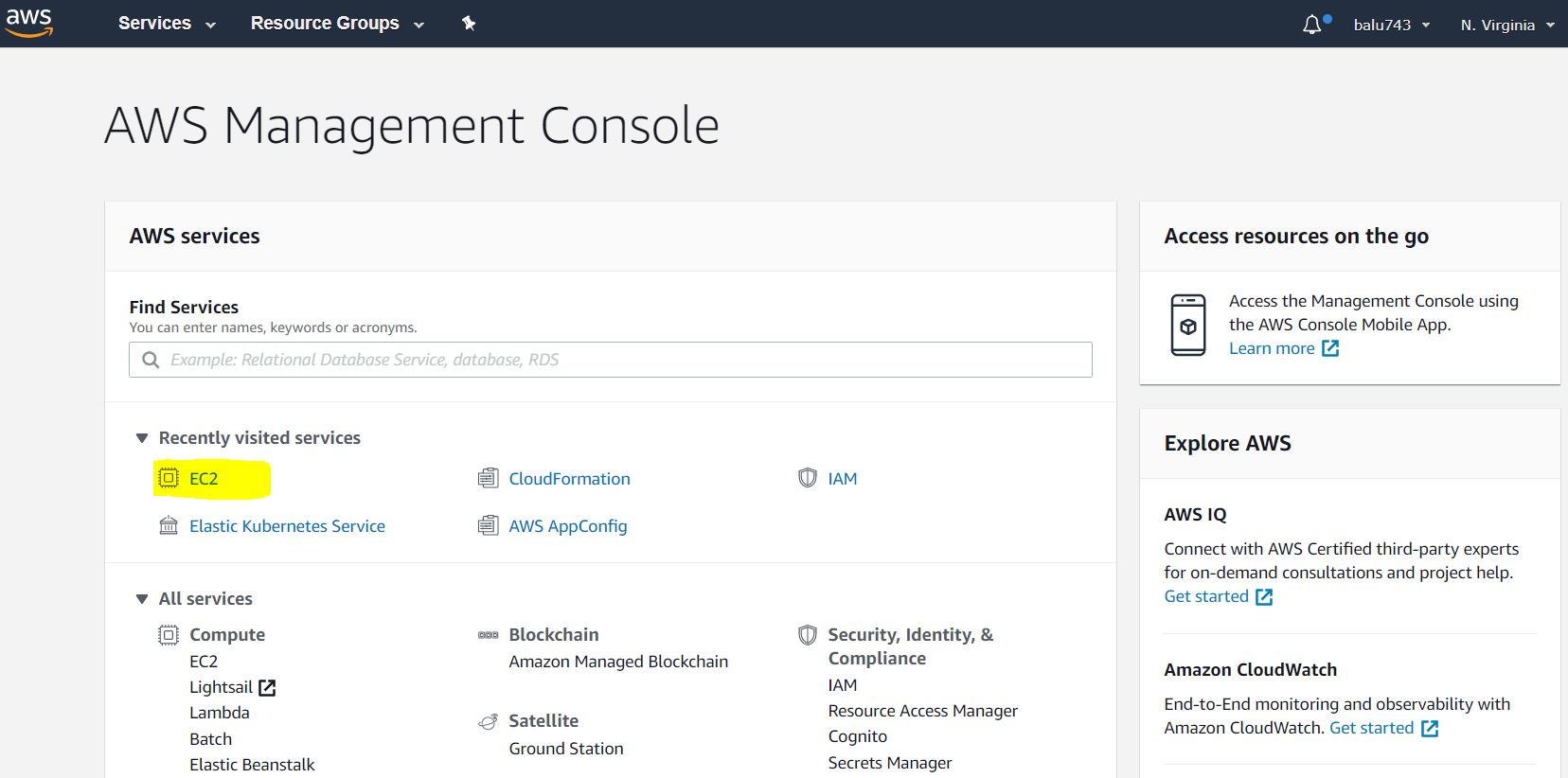
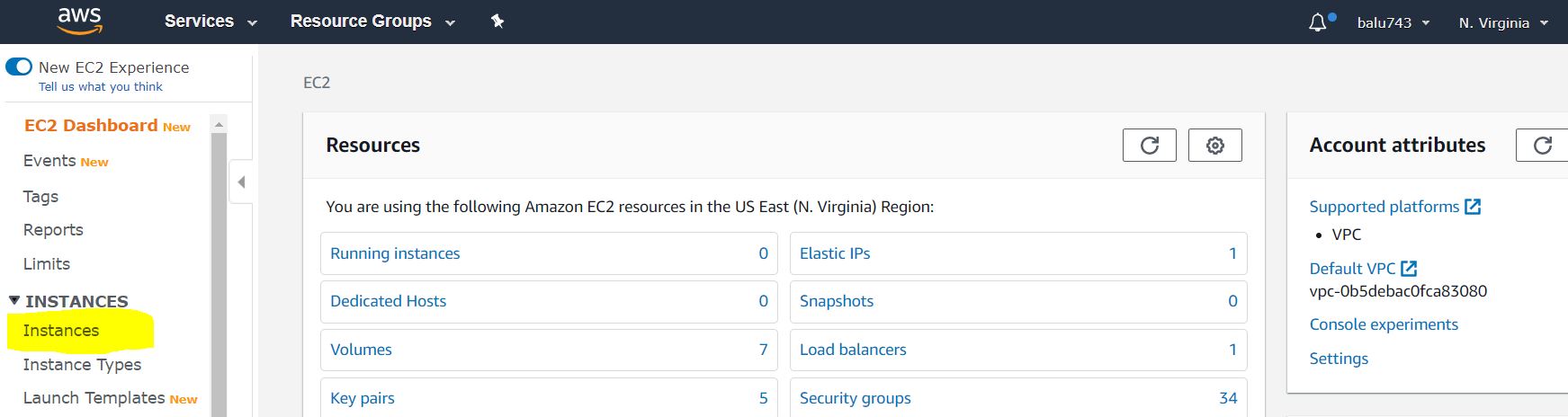
**How to Install Software’s in Amazon Linux-2**

**Create EC2 Instance:**

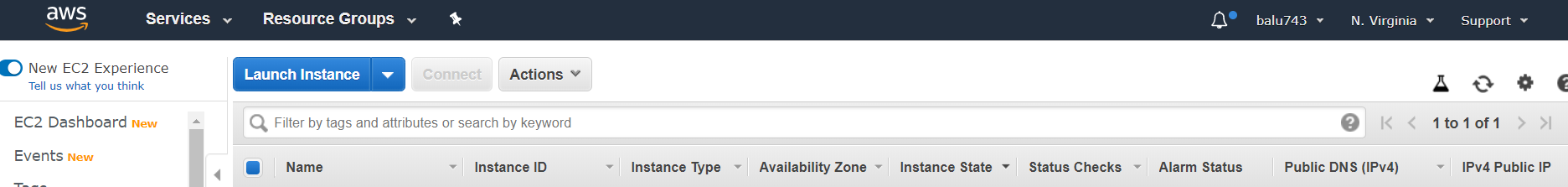
Login to AWS Management Console we will get below page



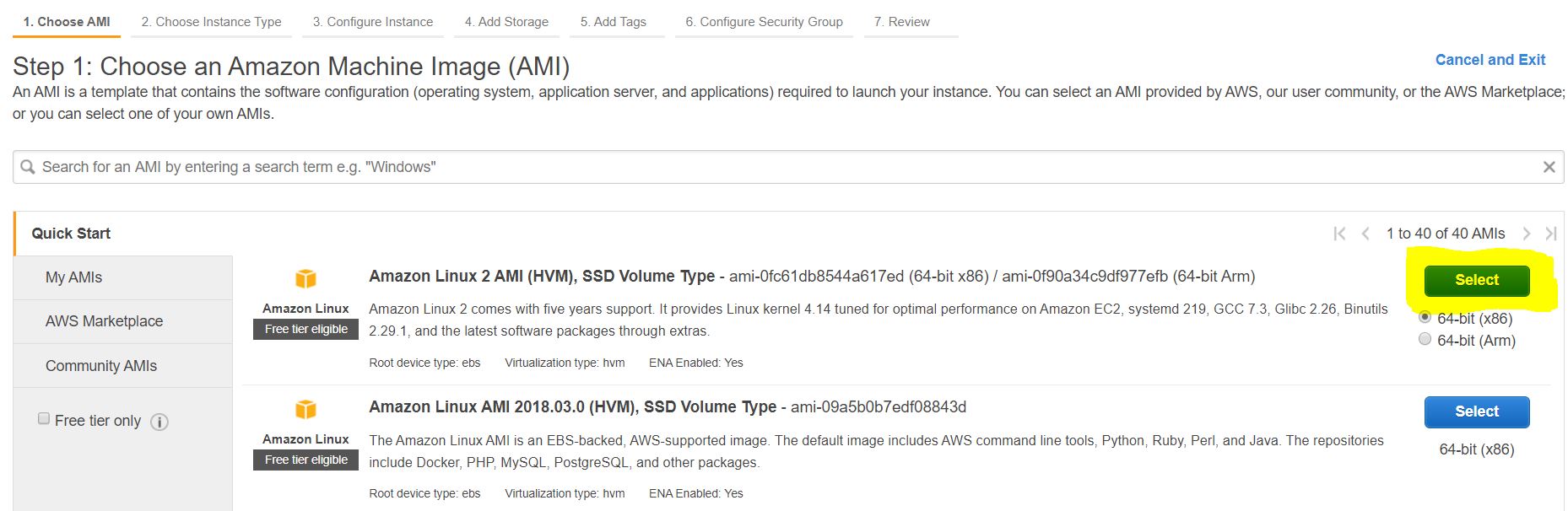
Click on **EC2**



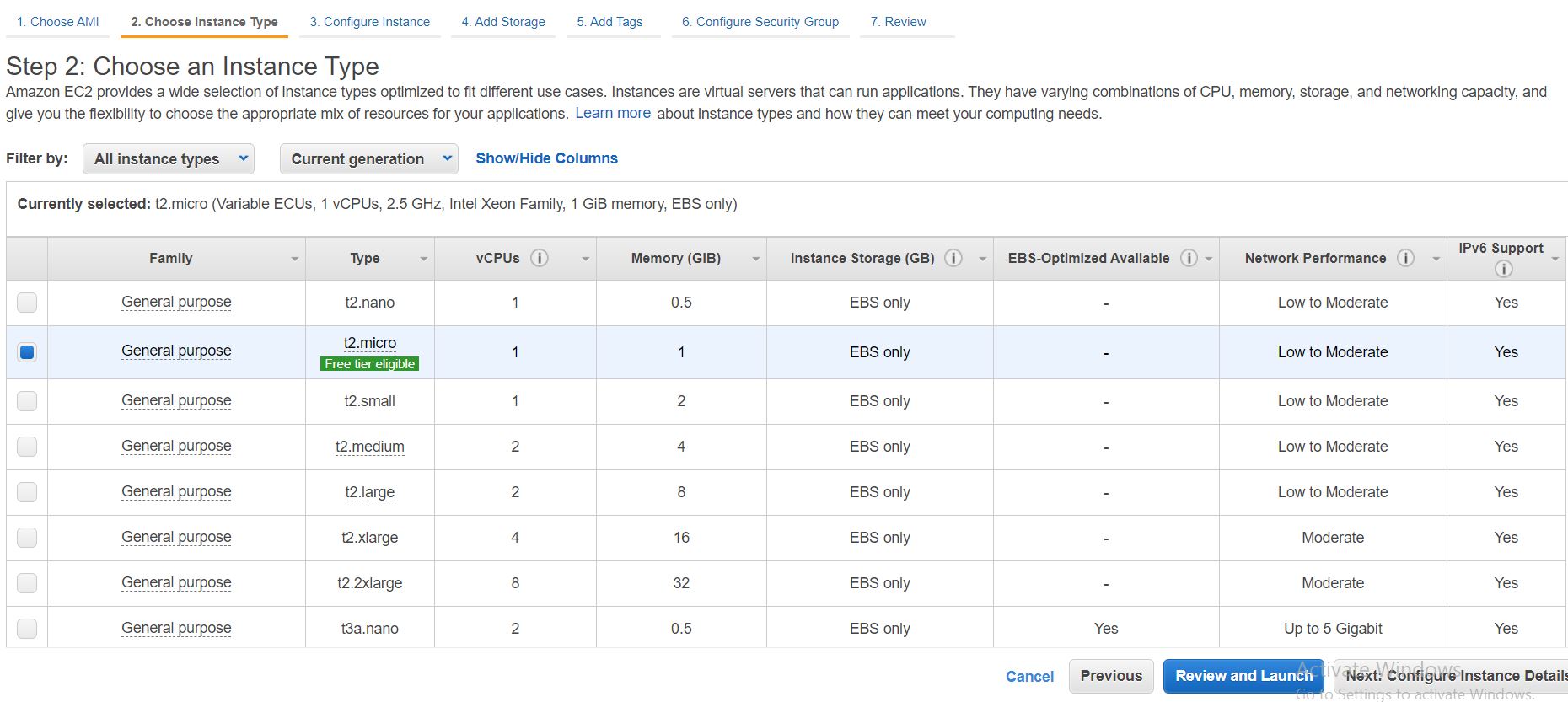
Click Instances



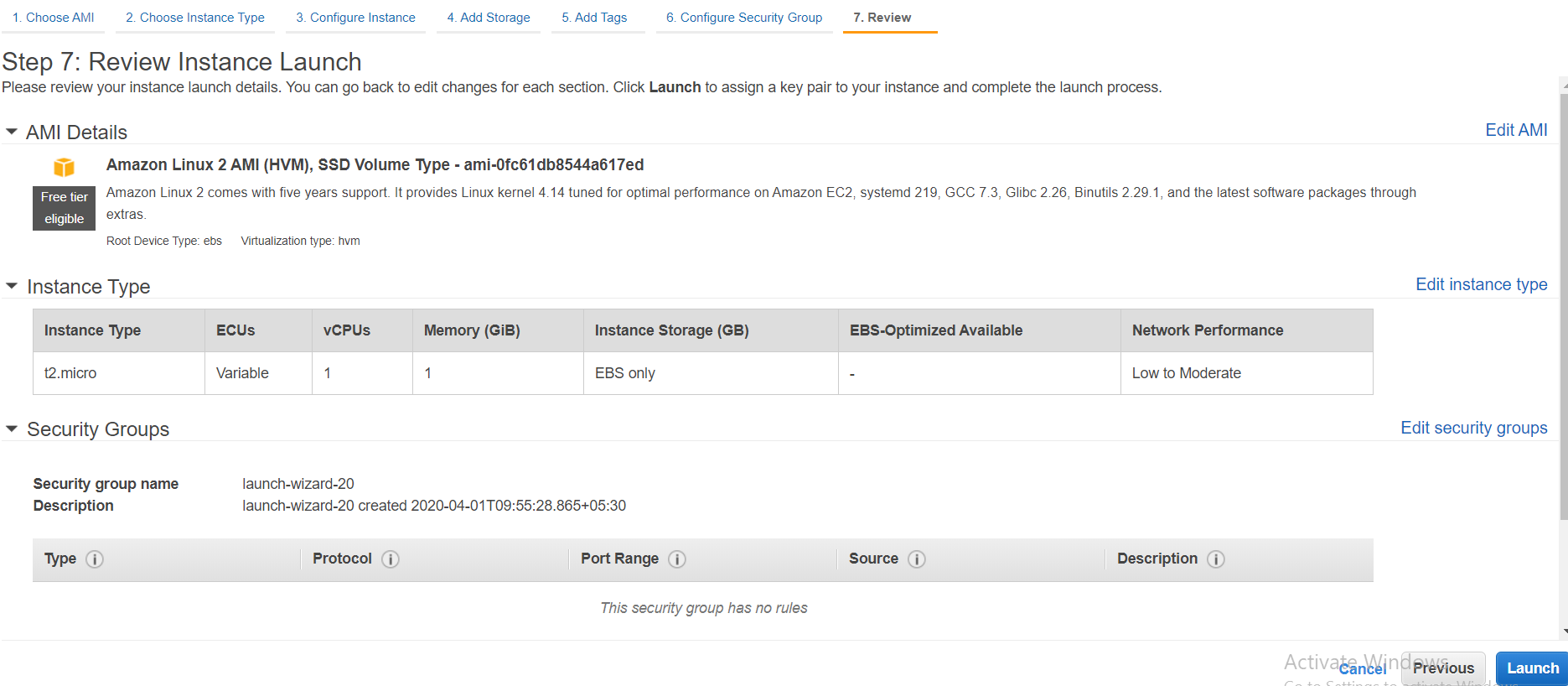
Click on **Launch Instance**



Click on **Select**



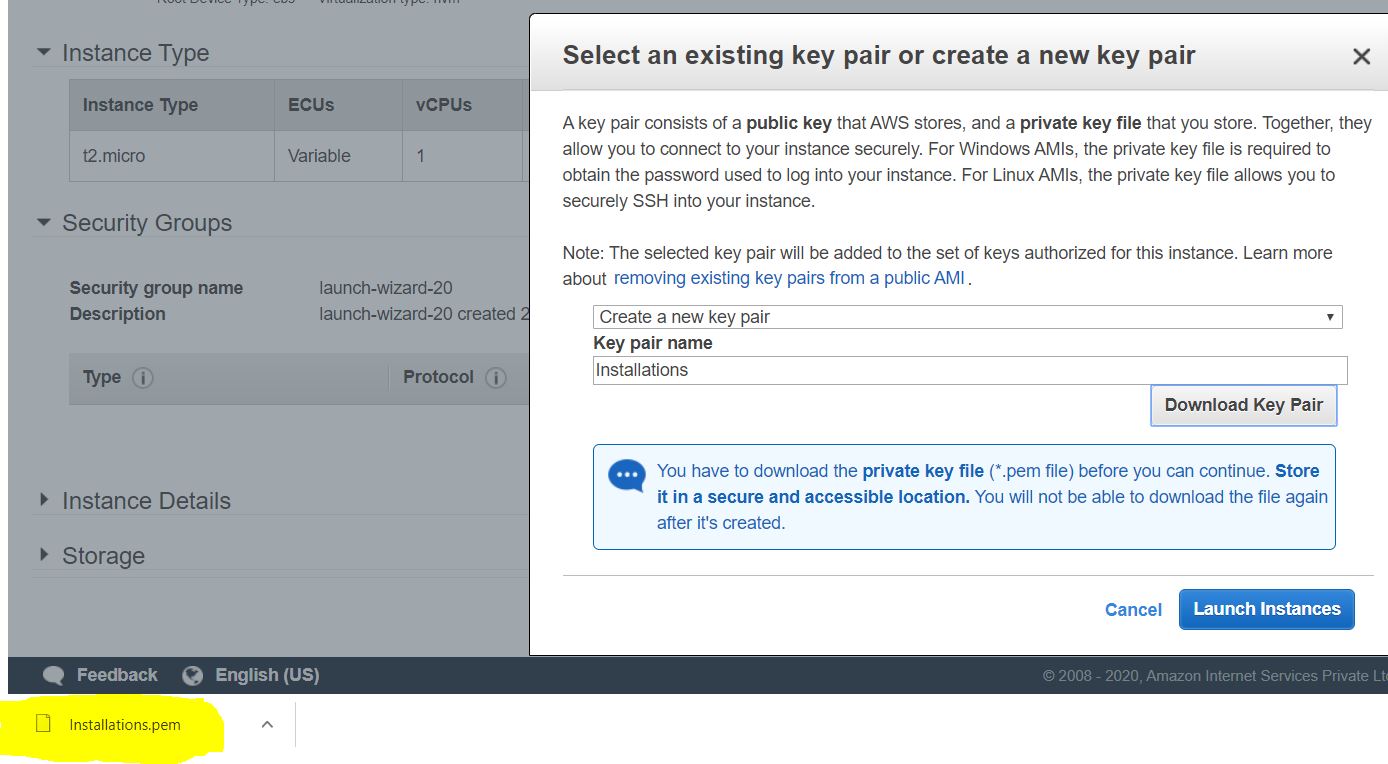
Select **t2.micro** and Click on **Review and Launch**



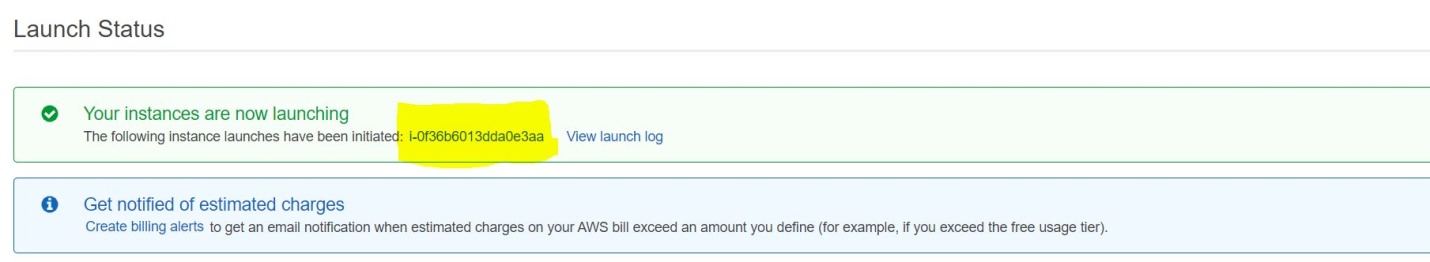
Click on **Launch**



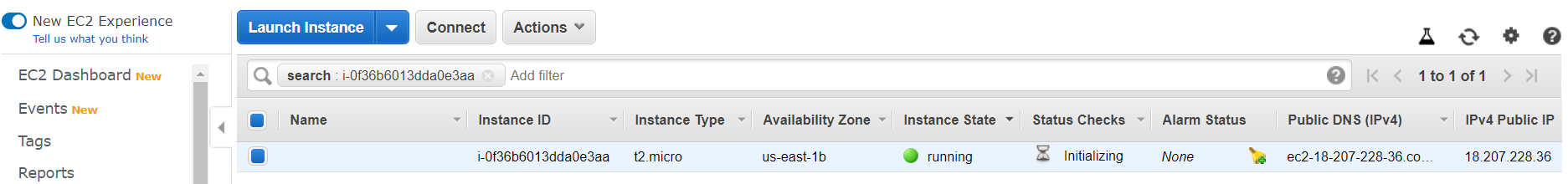
Select **Create a new key pair** and Give some name for **key pair name**



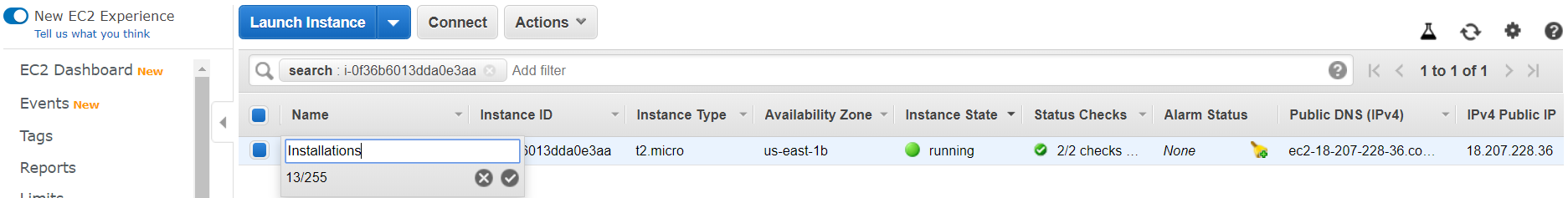
Here we can see downloaded file and then click on **Launch Instances**



Click on Launching instance



Keep a name for Instance

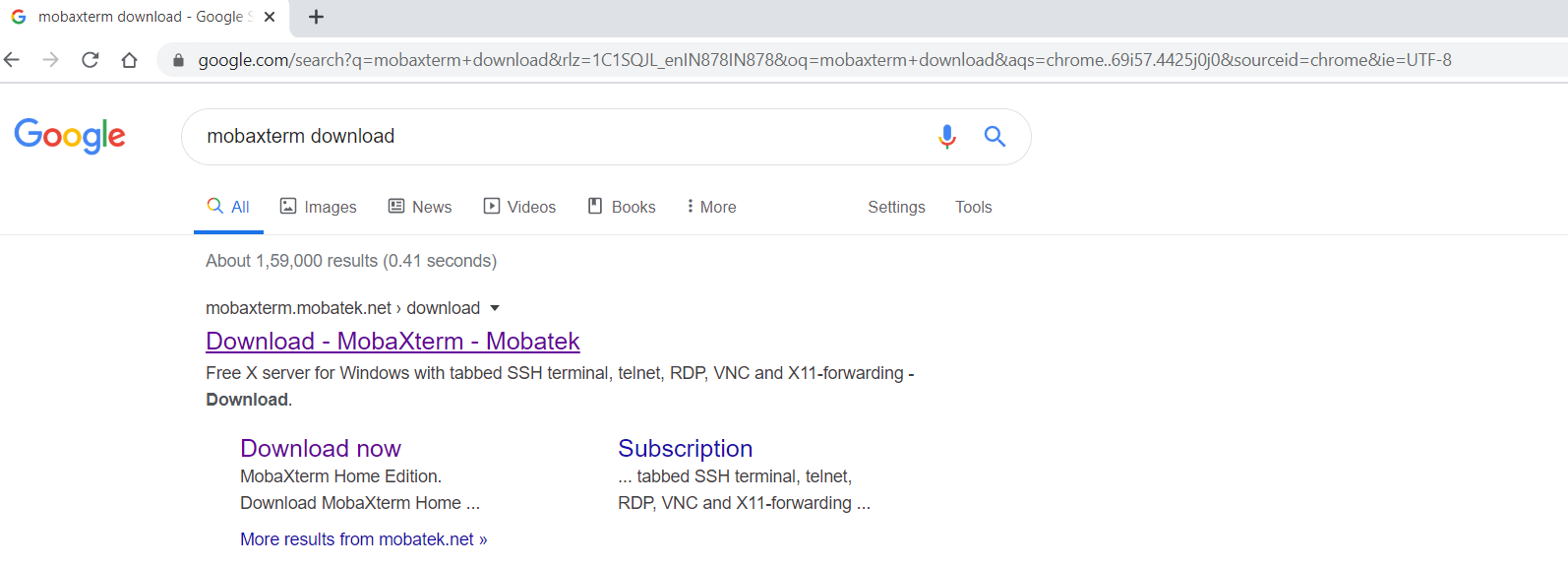


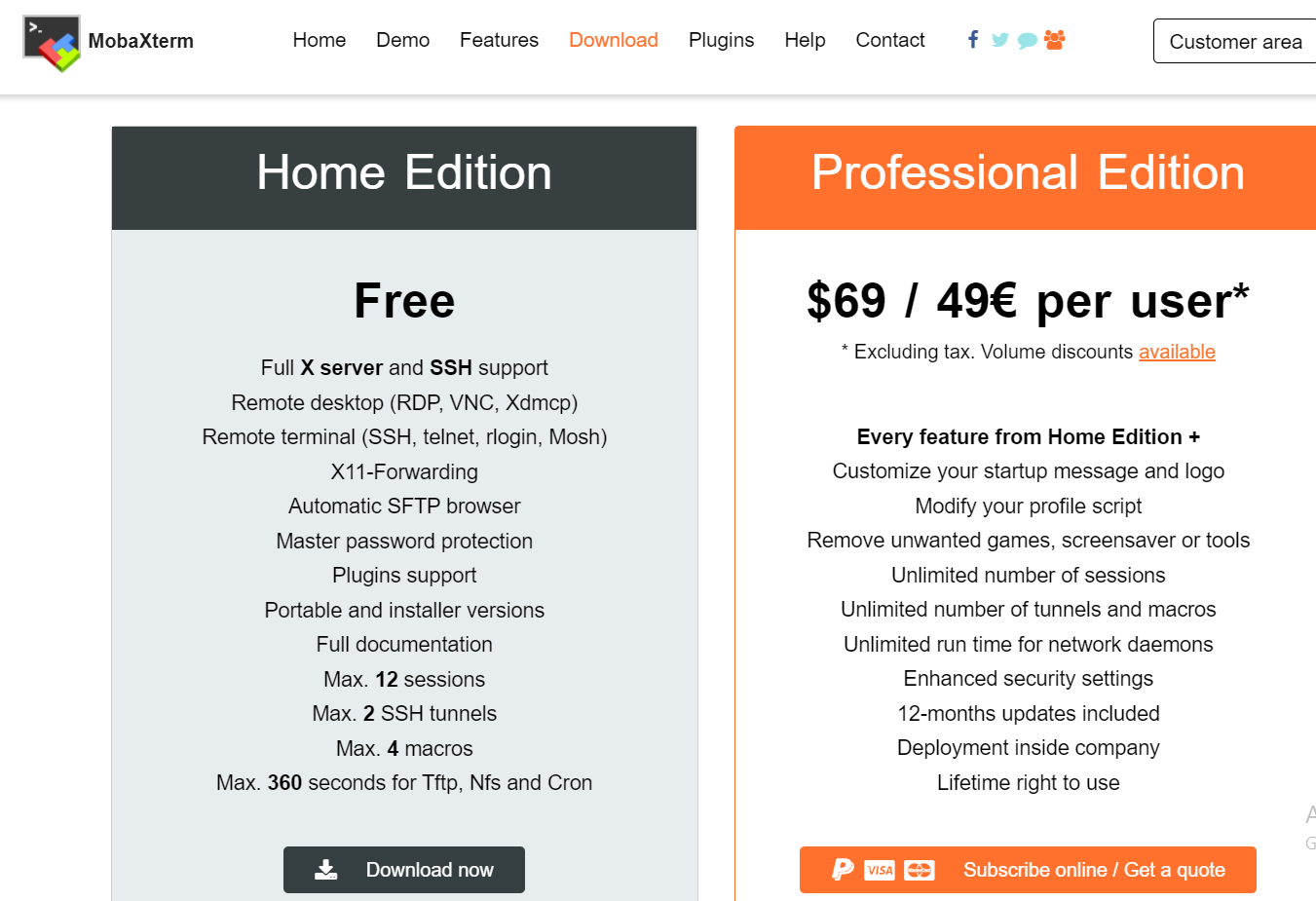
Click on this symbol... 



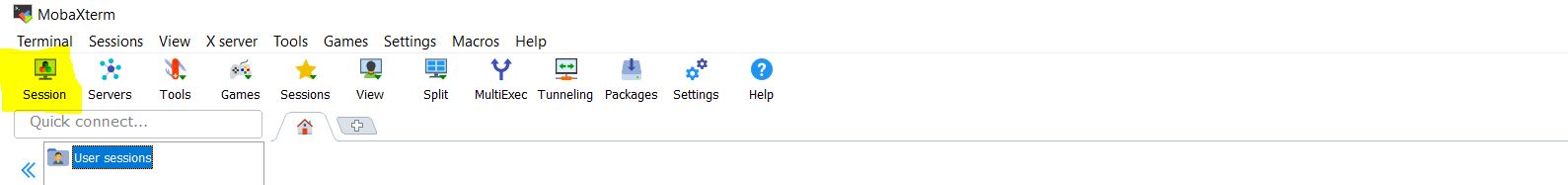
Copy **IPv4 Public IP** to connect to our server using **MobaXterm**

**Download MobaXterm:**

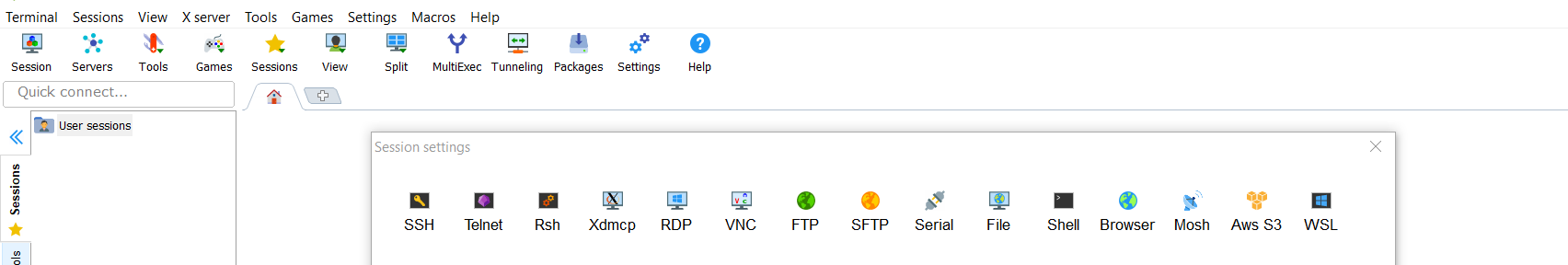


Click on Download – MobaXterm – MobatekDownload Home Edition

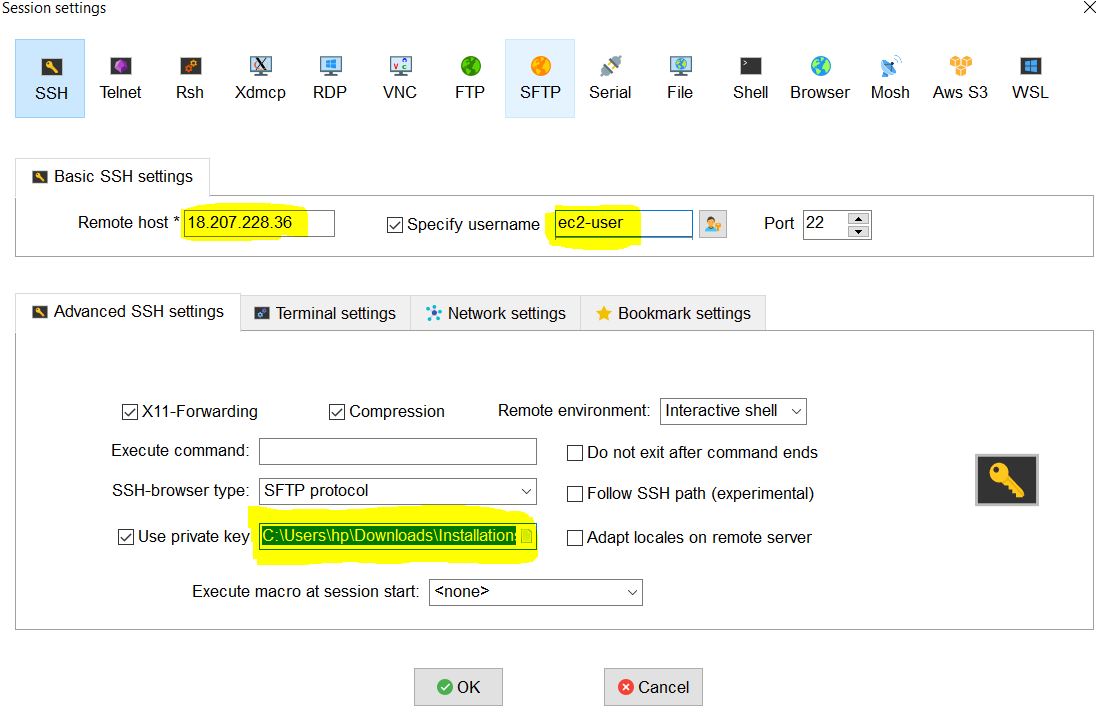
Open MobaXterm:



Click on **Session**



Click on **SSH**



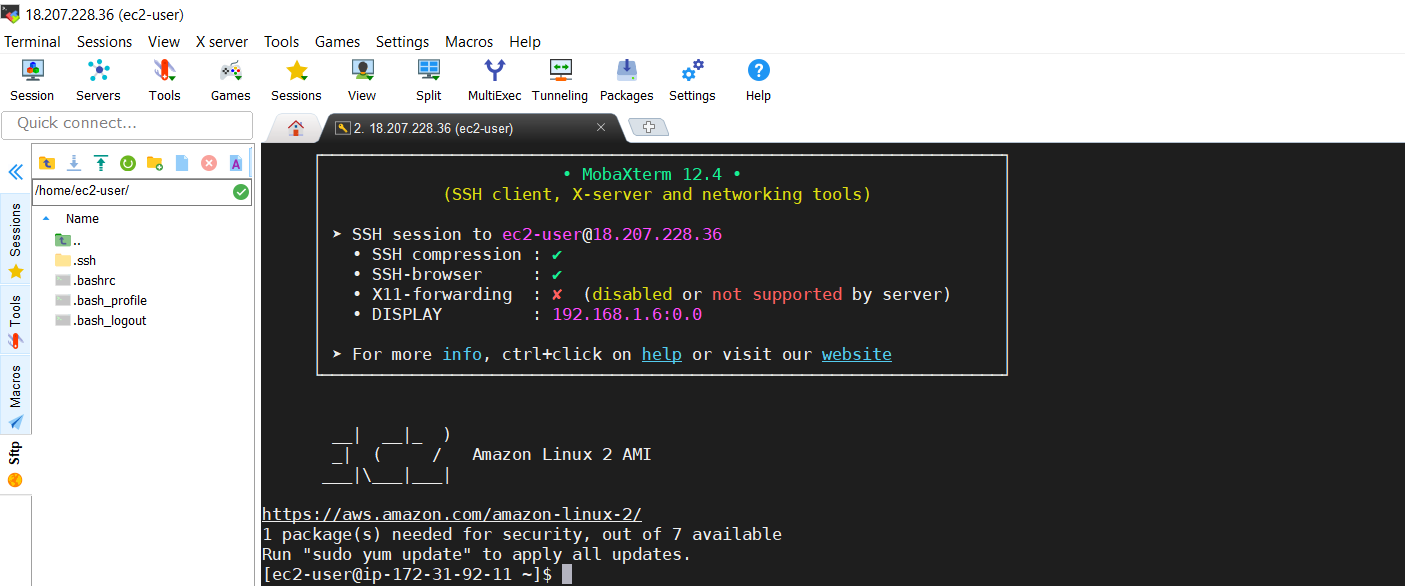
Give Details:

Remote host : IPv4 Public IP

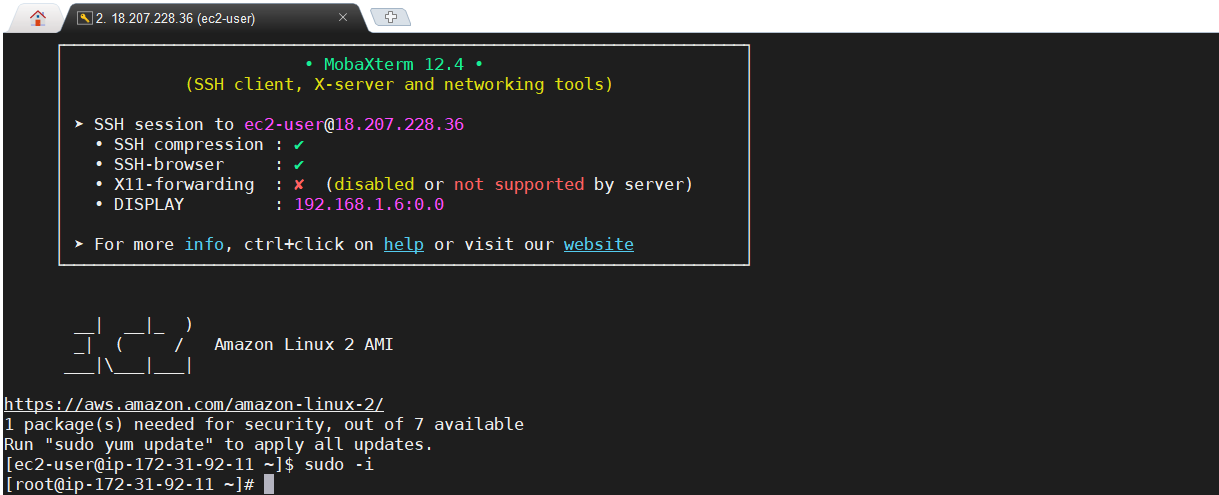
UserName : ec2-user

Private key : upload keypair

Click on **OK**

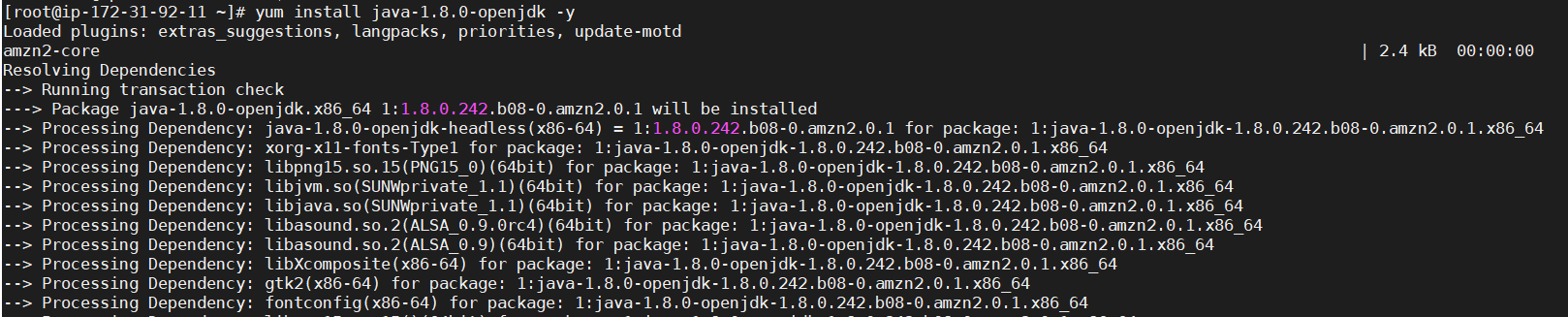


Now we are going to connect **root** user



**Install Java:**

yum install java-1.8.0-openjdk -y



Check Java Installed or not



**Install Git:**

yum install git -y

Check Git installed or not:

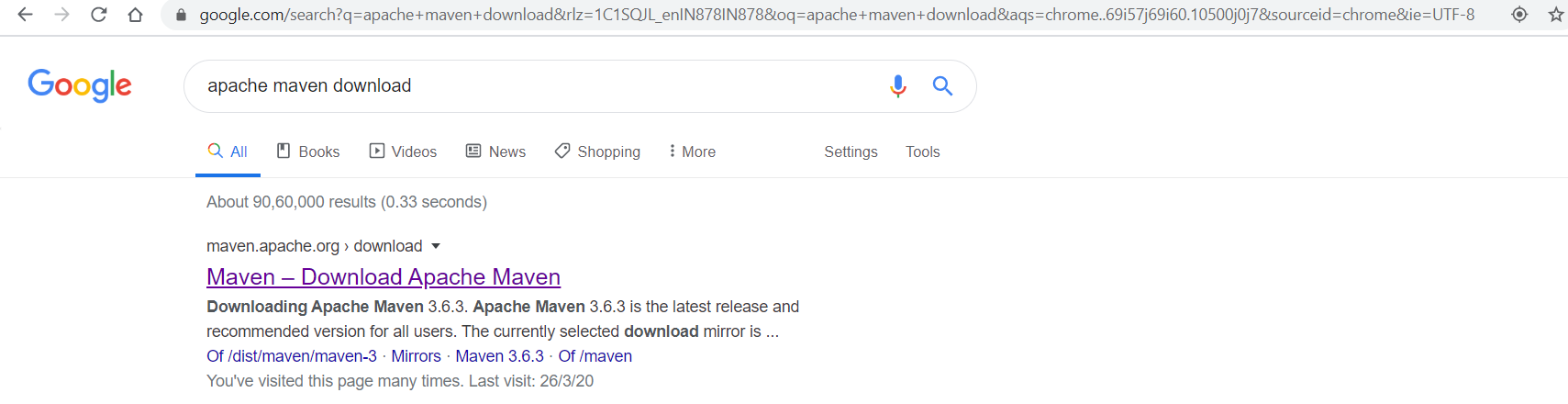
git --version



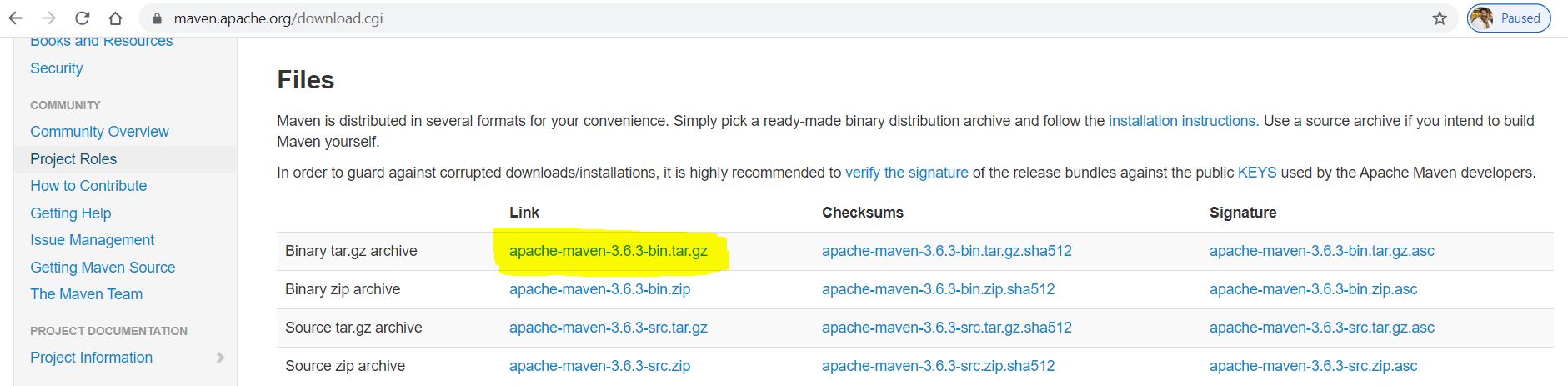
**Install Maven:**

Third party software’s we need to install in **/opt** directory

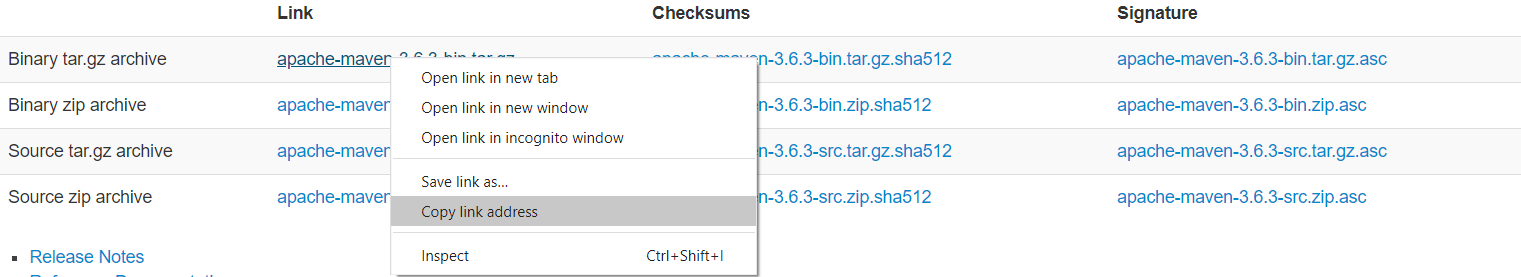
Download Maven for UI



Click on Maven – Download Apache Maven



Right Click on **apache-maven-3.6.3-bin.tar.gz**

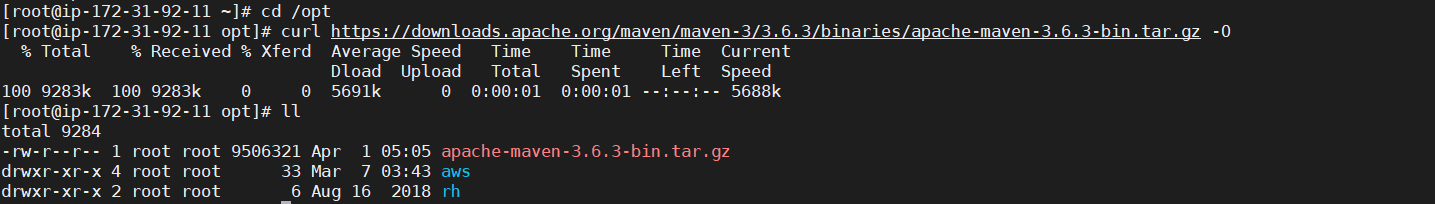


Copy link address

Goto MobaXterm

cd /opt

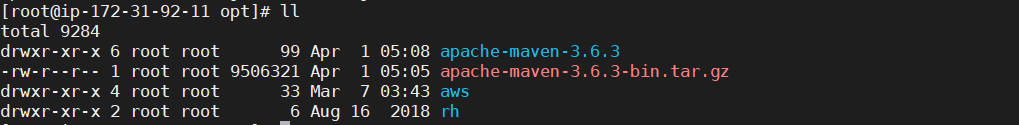
curl https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz -O



Extract the file:

tar -xvf apache-maven-3.6.3-bin.tar.gz

ll



Check extracted file here

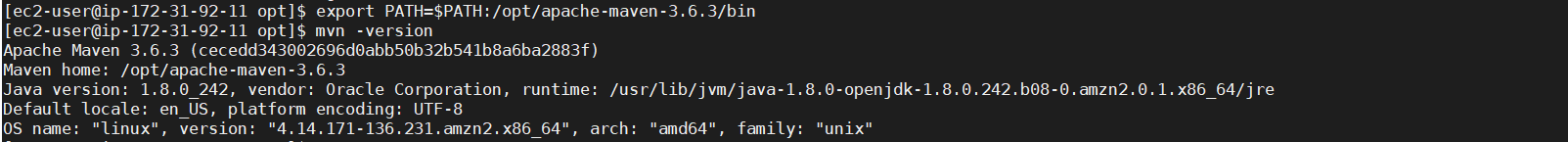
**Set Path for Maven:**

We need to set **/opt/apache-maven-3.6.3/bin** as a **MAVEN PATH**

export PATH=$PATH:/opt/apache-maven-3.6.3/bin

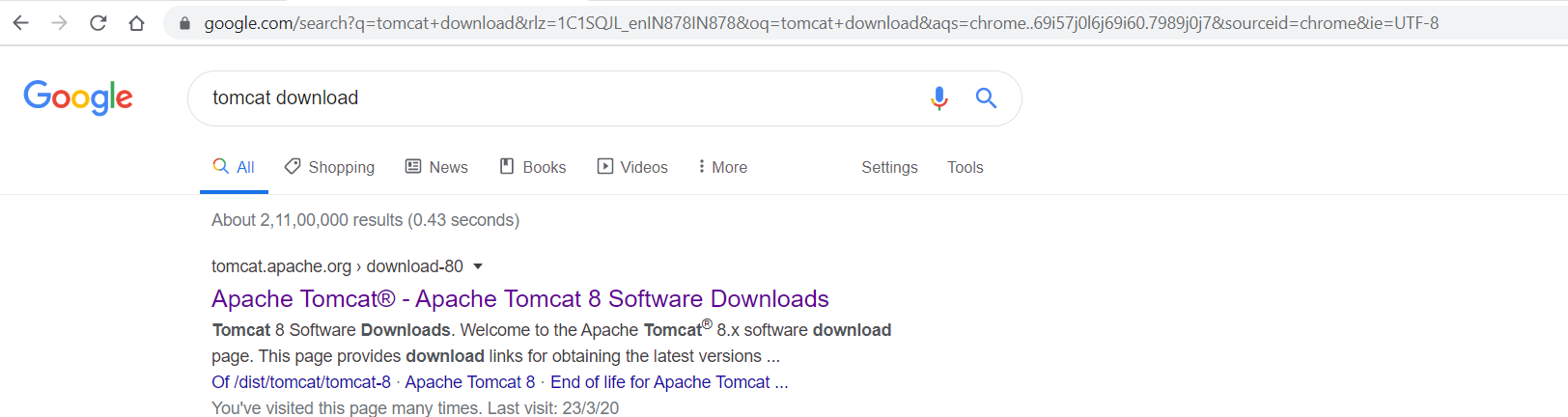
Check Maven Installed or not:

mvn -version

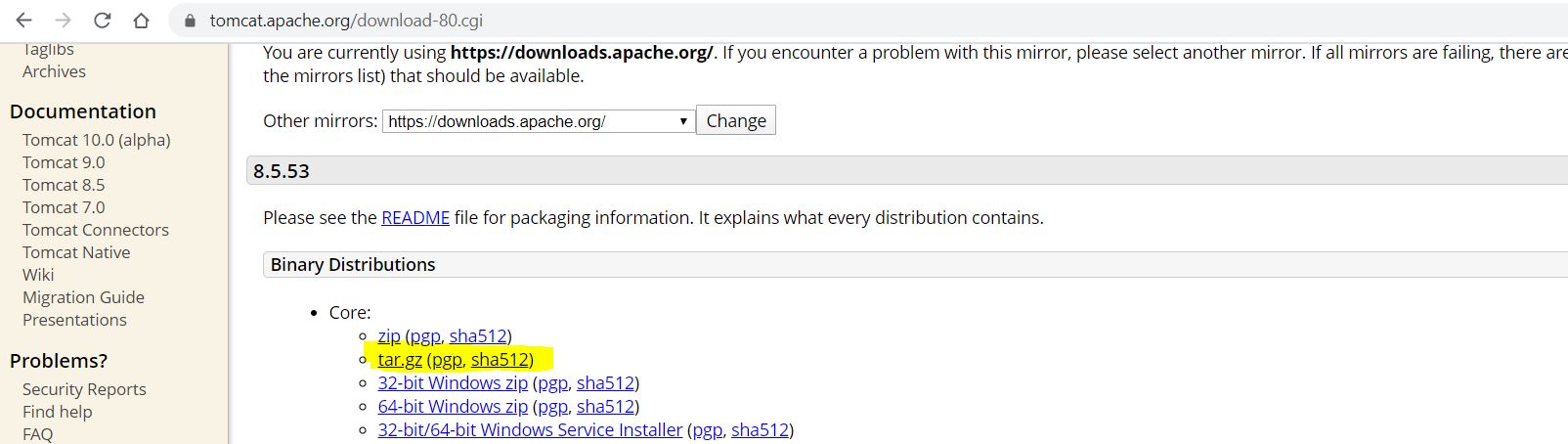


**Tomcat Installation:**

Download Apache tomcat from UI:



Click on **Apache Tomcat® - Apache Tomcat 8 Software Downloads**

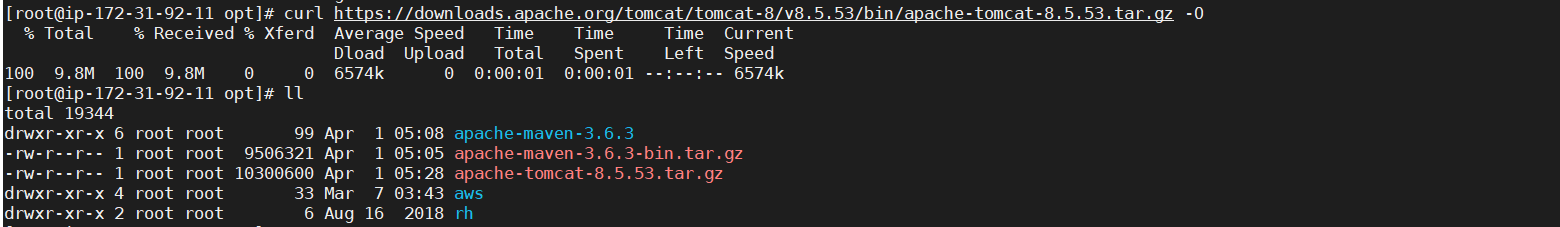


Right Click and copy link of **tar.gz (pgp, sha512)**

Goto MobaXterm and Download Apache tomcat:

curl https://downloads.apache.org/tomcat/tomcat-8/v8.5.53/bin/apache-tomcat-8.5.53.tar.gz -O

ll



Extract tomcat:

tar -xvf apache-tomcat-8.5.53.tar.gz

ll

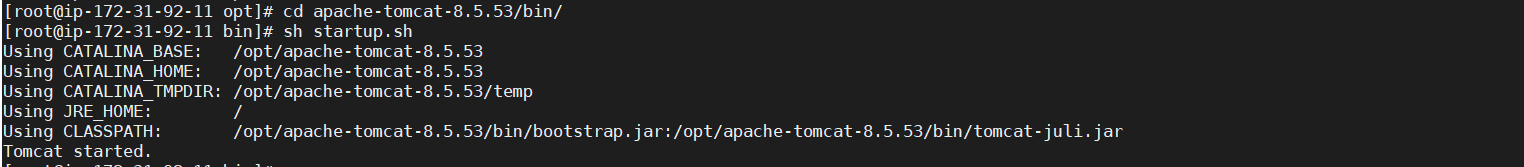


Now start Tomcat:

Goto bin directory and run **startup.sh** file

cd apache-tomcat-8.5.53/bin/

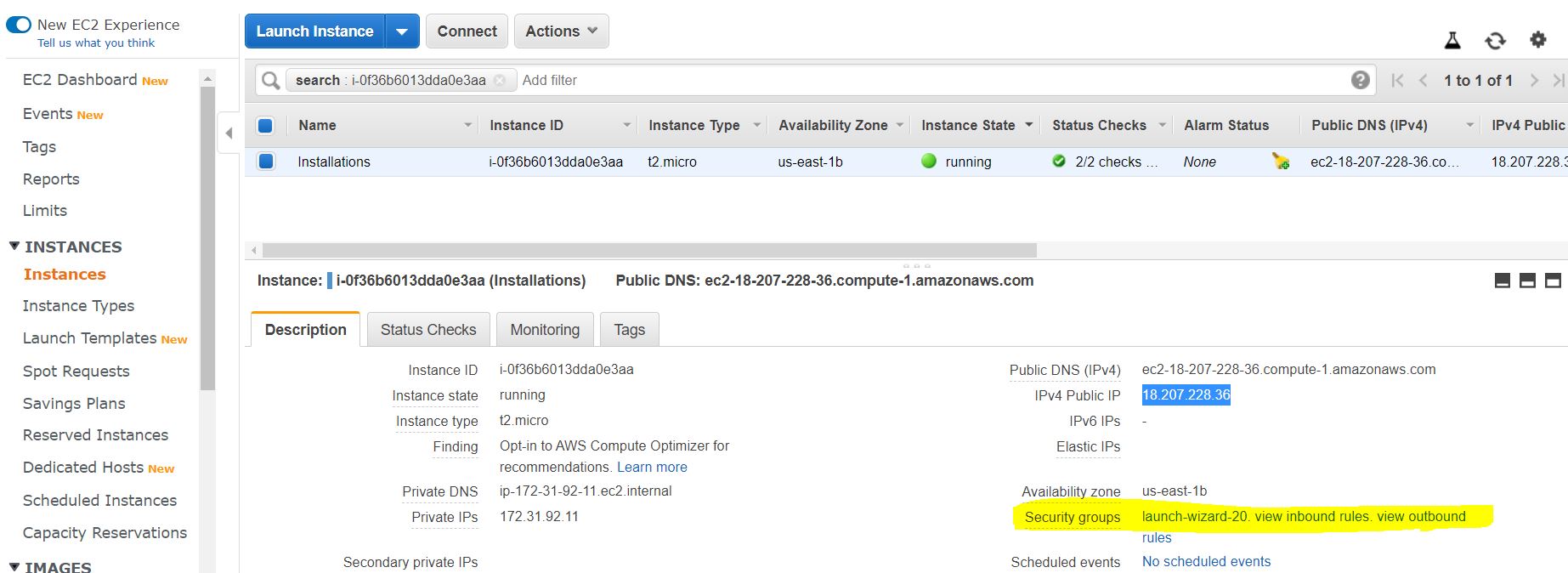
sh startup.sh



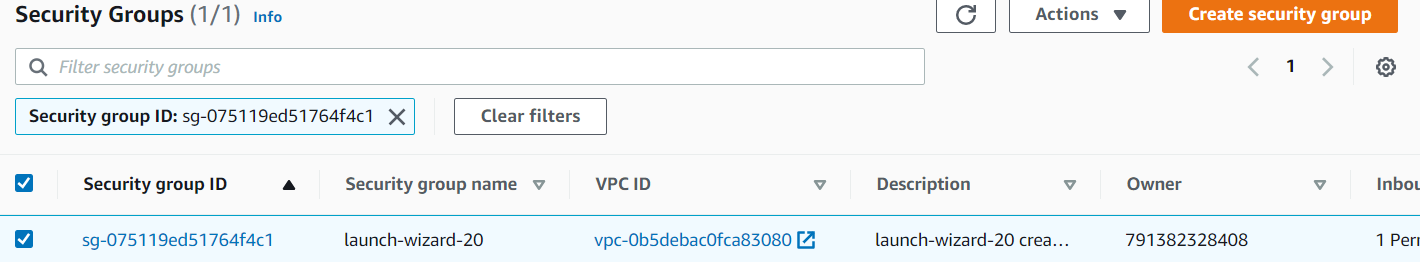
Now we can check tomcat page in Web UI:

First we need to allow **8080** port number in **security group** for this server

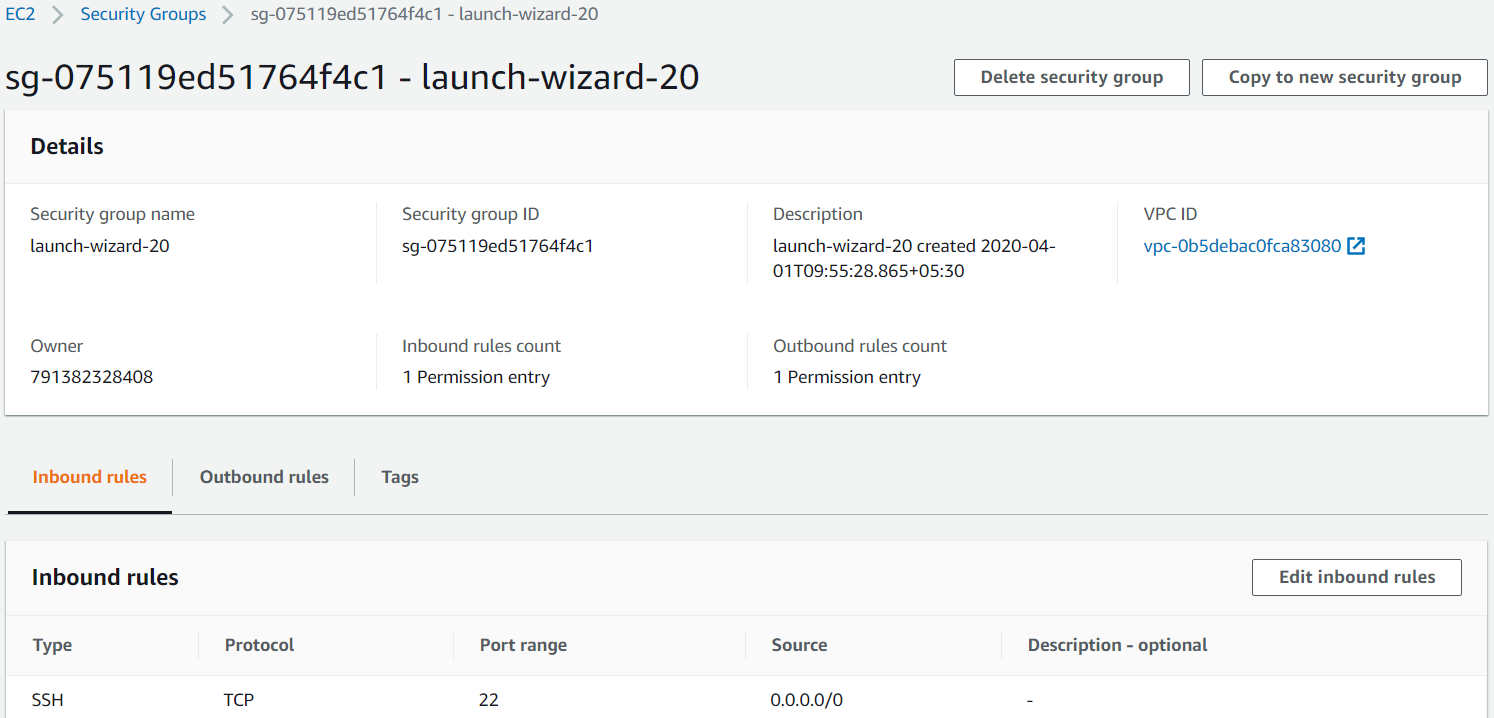
Goto EC2 Service



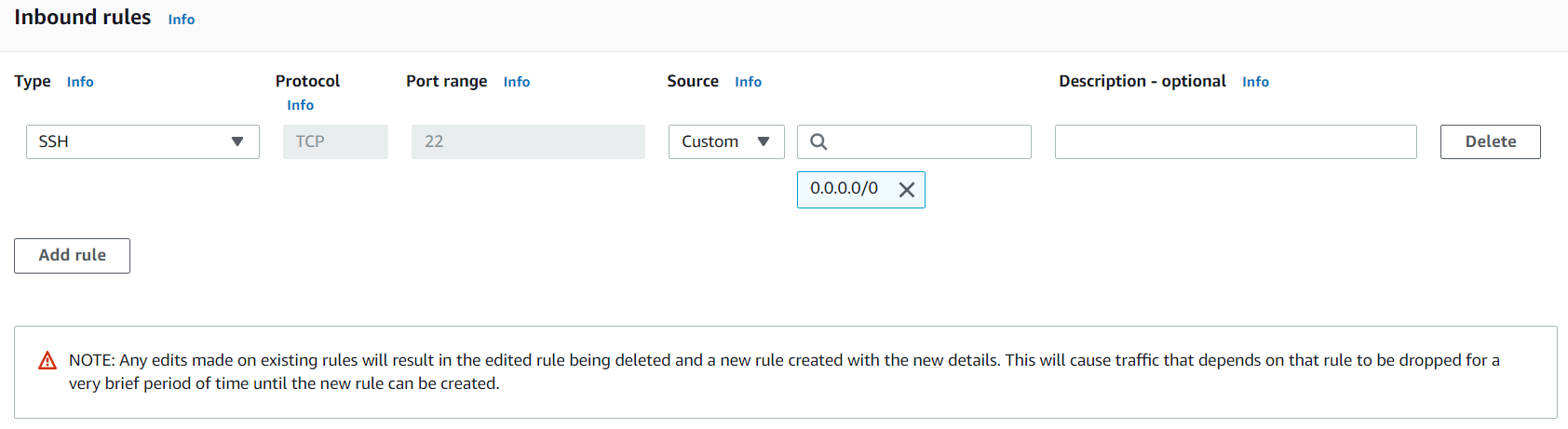
Click on Security groups: launch-wizard



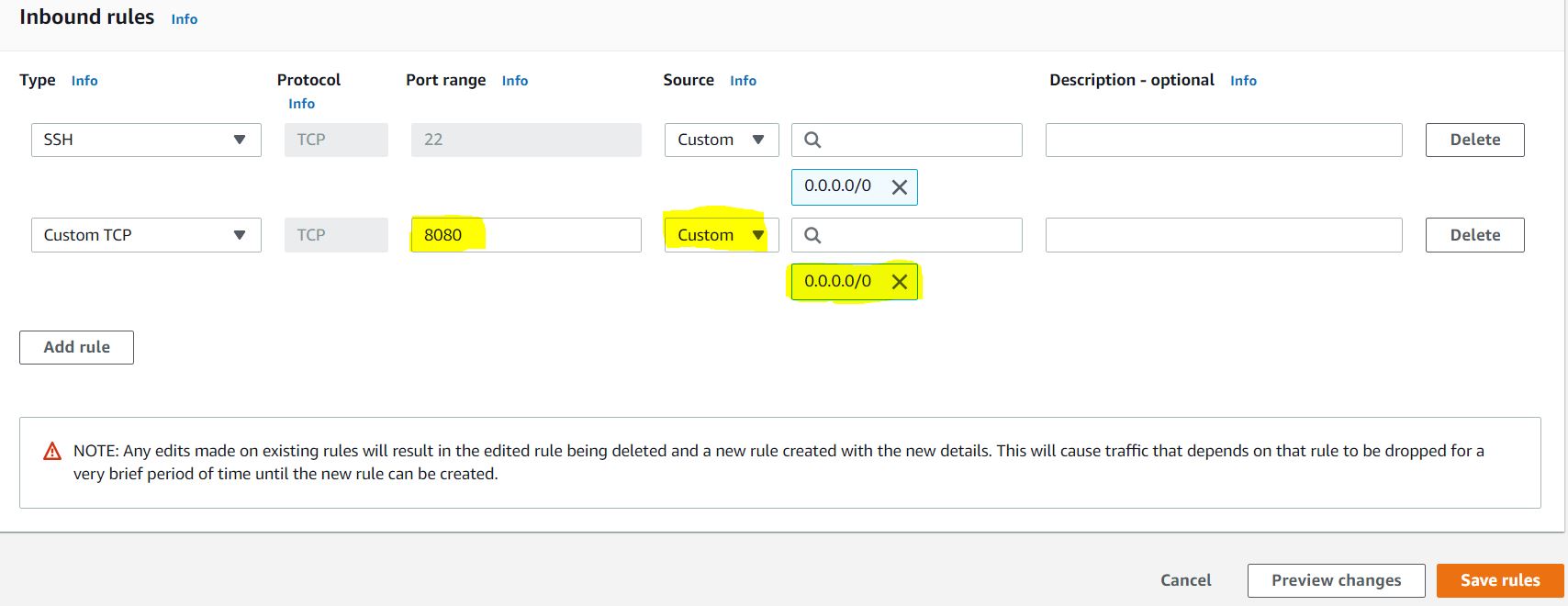
Click on Security group ID: **sg-075119ed51764f4c1**



Click on **Edit inbound rules**

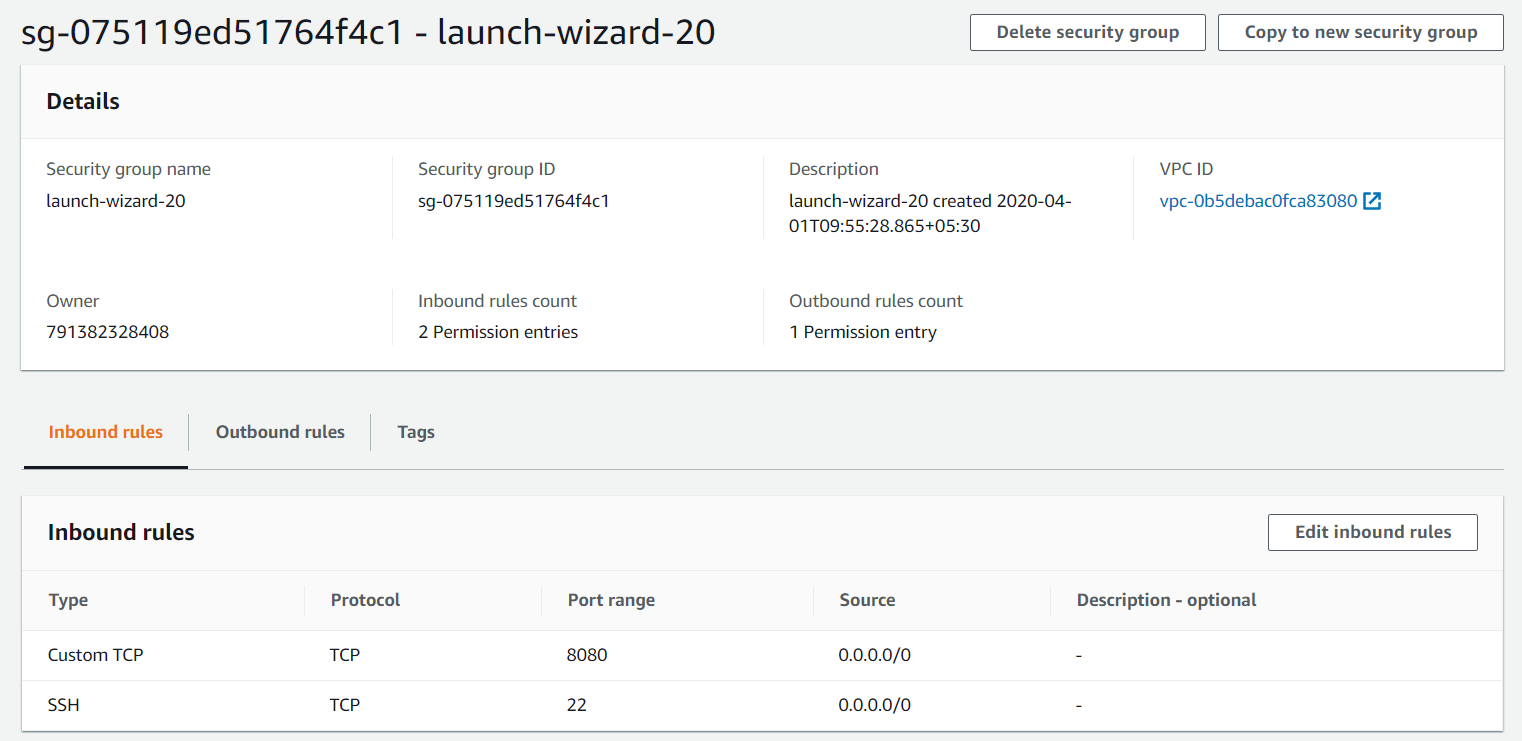


Click on **Add rule**



Give details as above

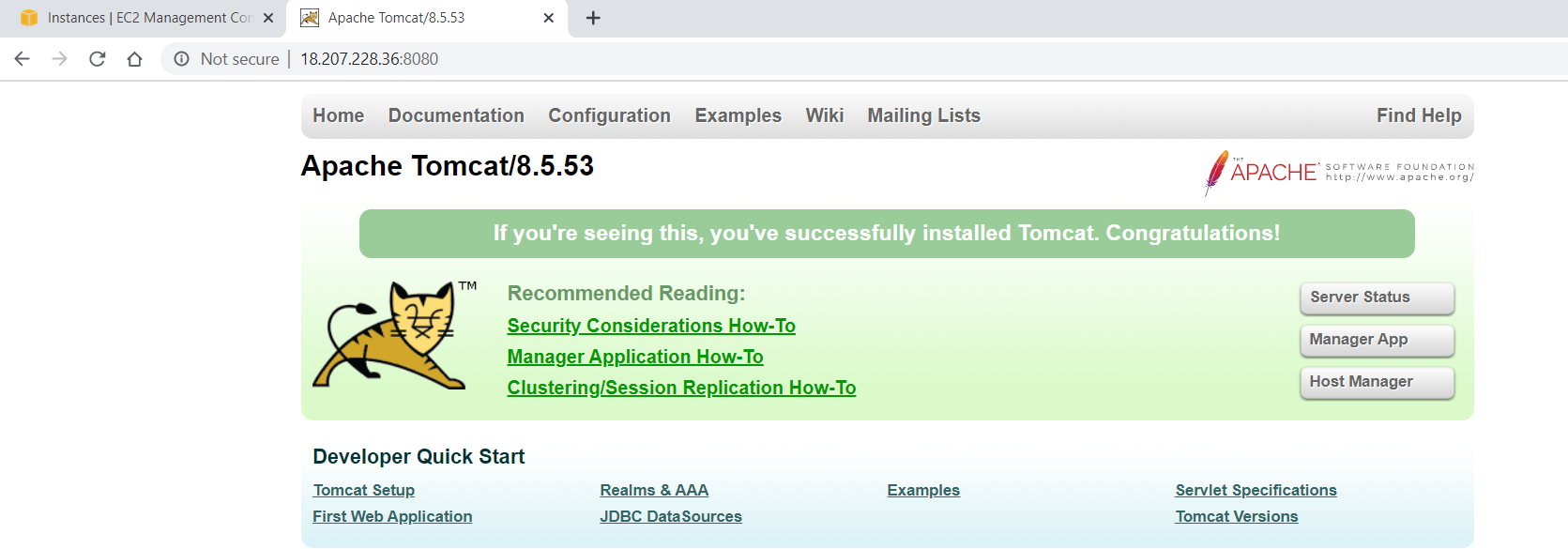
Click on **Save rules**



Now check in Web UI:

Copy IPv4 Private IP for the server and give in web UI with port number

<http://18.207.228.36:8080/>



**Install Jenkins:**

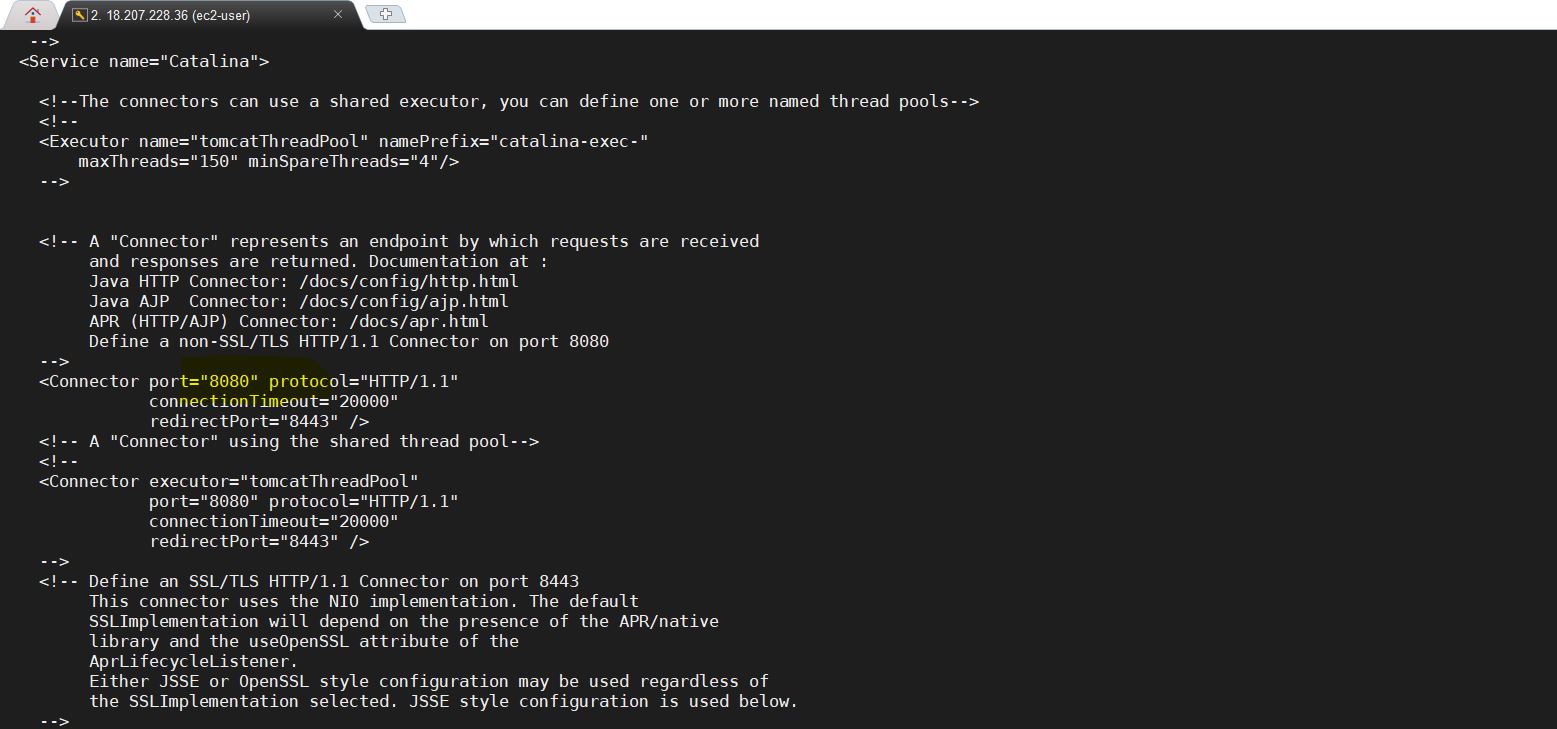
If you want to install Jenkins and Apache Tomcat with in the same server, we need to change port number either for Jenkins or Tomcat. Because Jenkins and tomcat having same port number:8080

Here I am going to Change port number for tomcat.

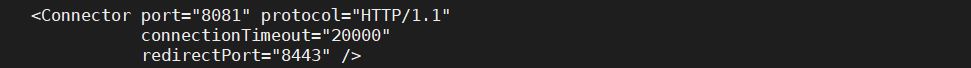
Goto MobaXterm:

Goto this PATH: **/opt/apache-tomcat-8.5.53/conf**

vi server.xml



Change this Port number 8080 to 8081 or any other Port number



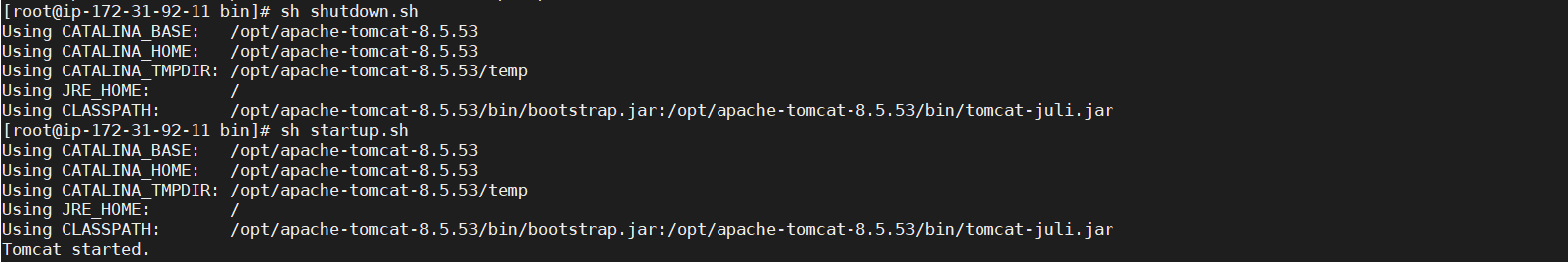
Save this file and Restart tomcat

Goto bin directory:

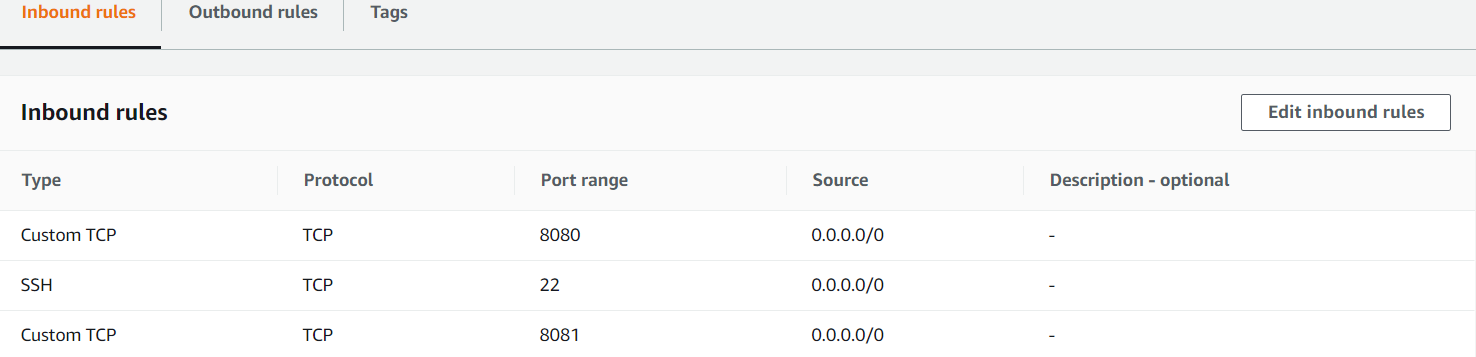
cd /opt/apache-tomcat-8.5.53/bin/

sh shutdown.sh

sh startup.sh

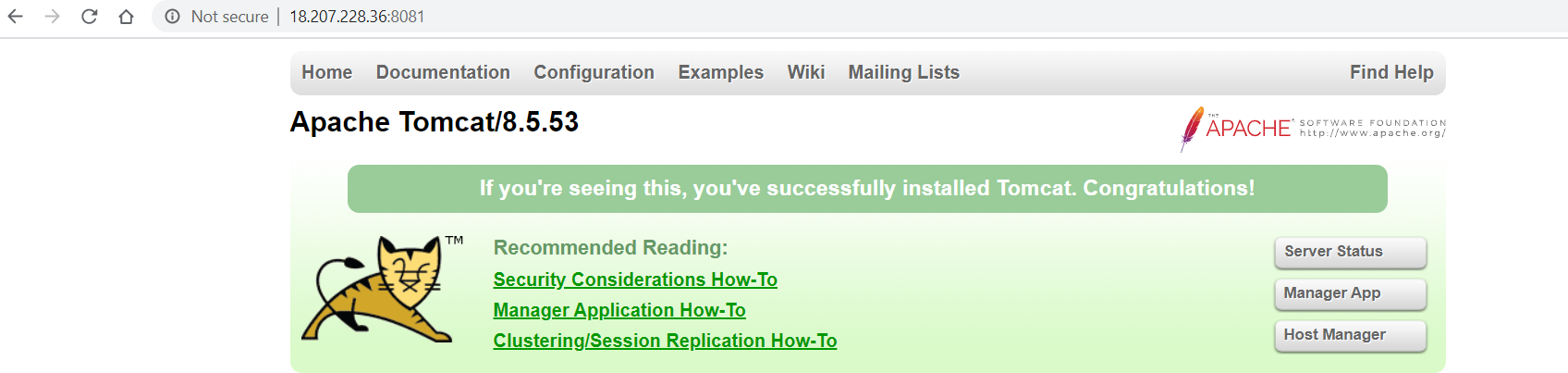


Now allow Port Number: 8081 in security group of EC2 Instance:



Check Apache Tomcat page in Web UI:

<http://18.207.228.36:8081/>



Now Install Jenkins:

sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo

sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key

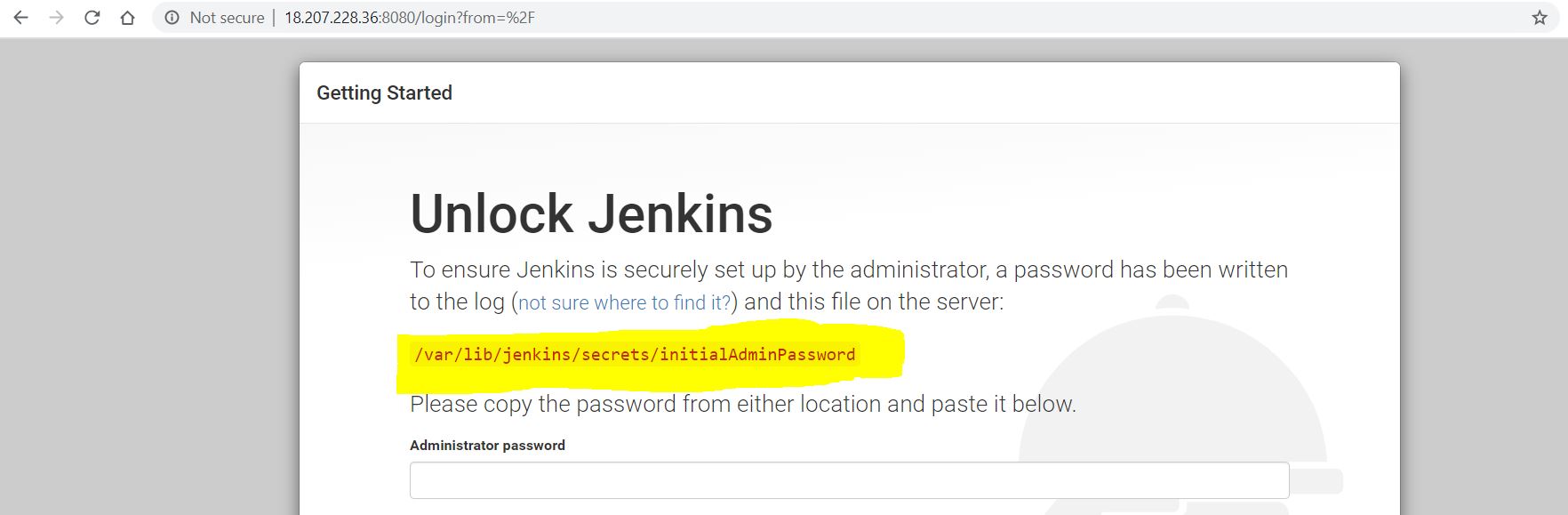
sudo yum install jenkins -y

service jenkins start



Here Jenkins port number: 8080 already allowed in security group. So we can given IP with port number in web UI:

<http://18.207.228.36:8080/>

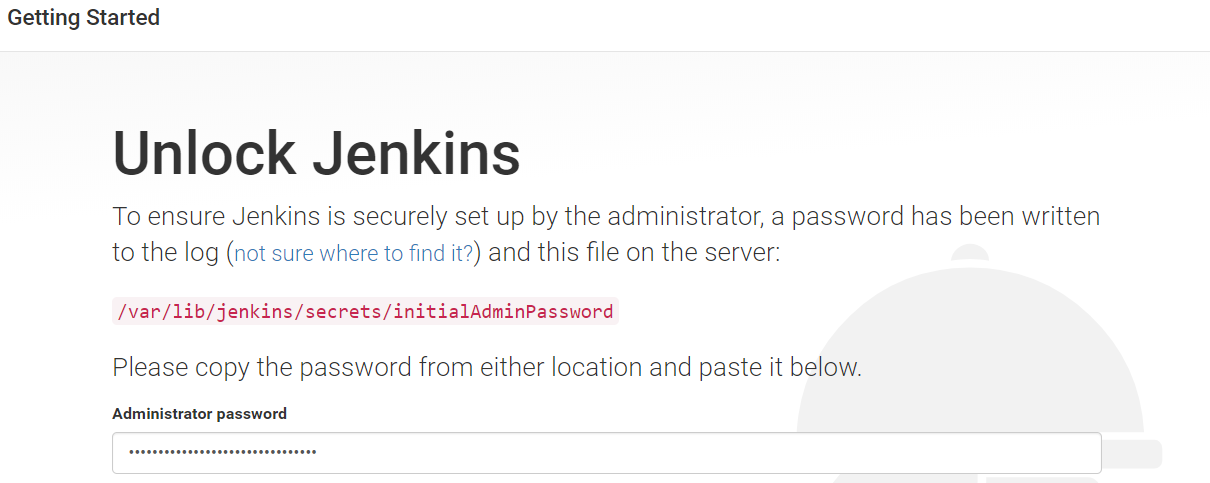


For Administrator password Copy this path and give at MobaXterm

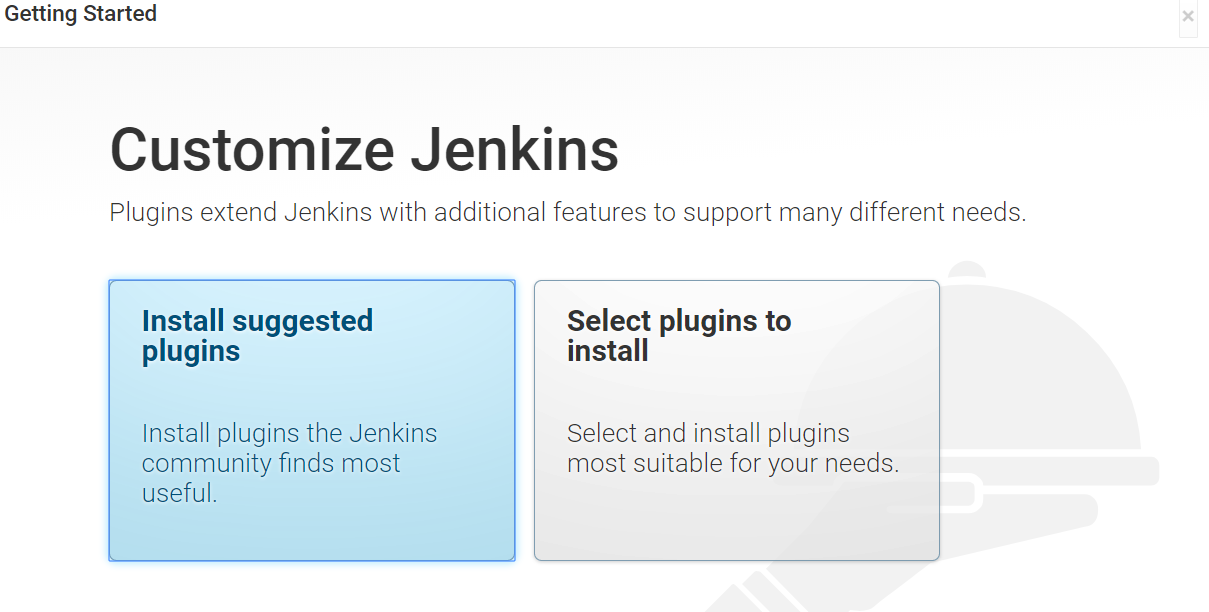
cat /var/lib/jenkins/secrets/initialAdminPassword



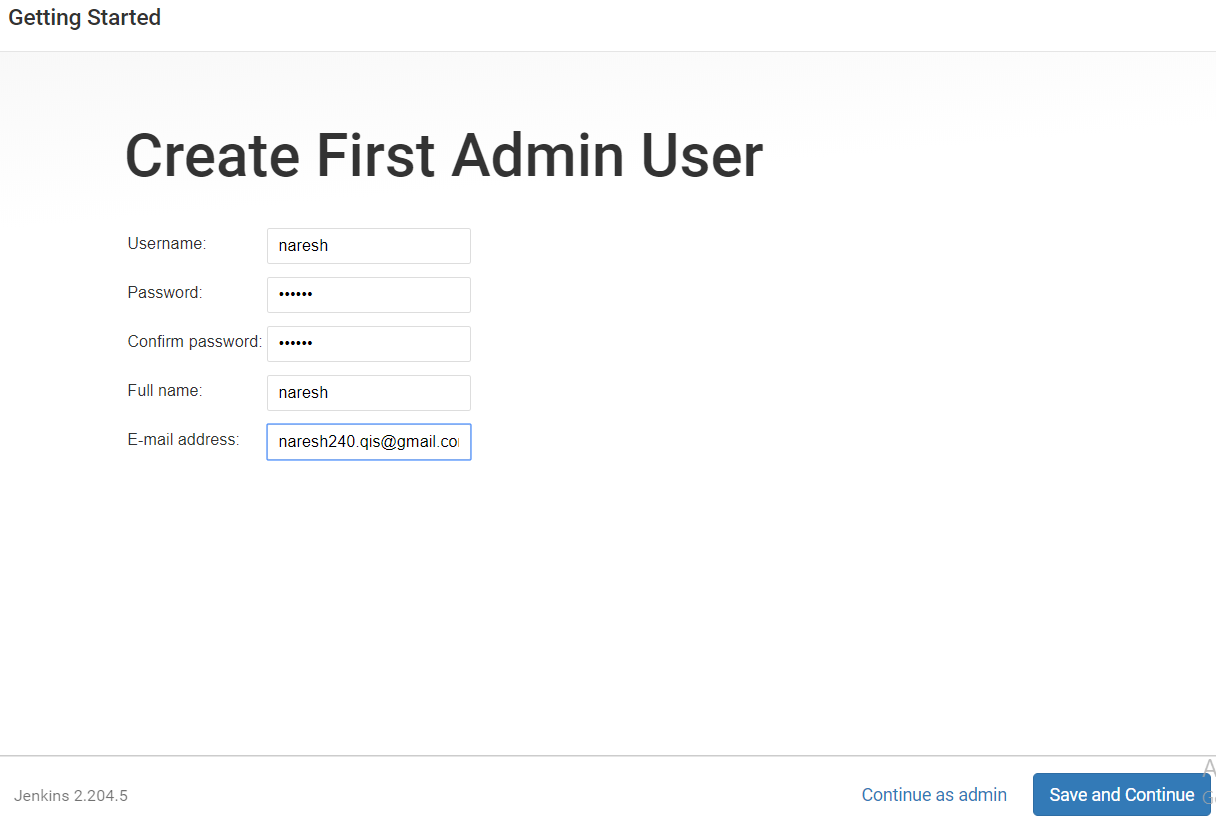
Copy password and keep inside the box



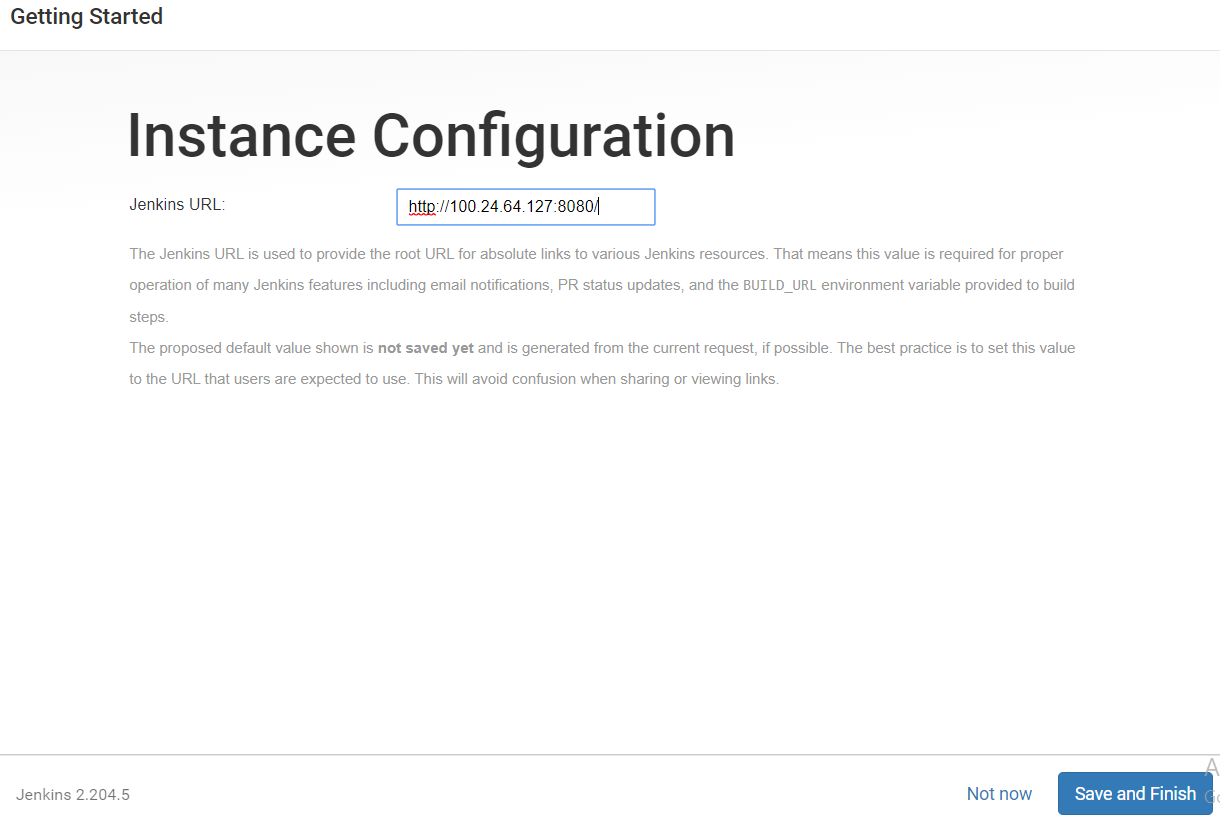
Click on continue



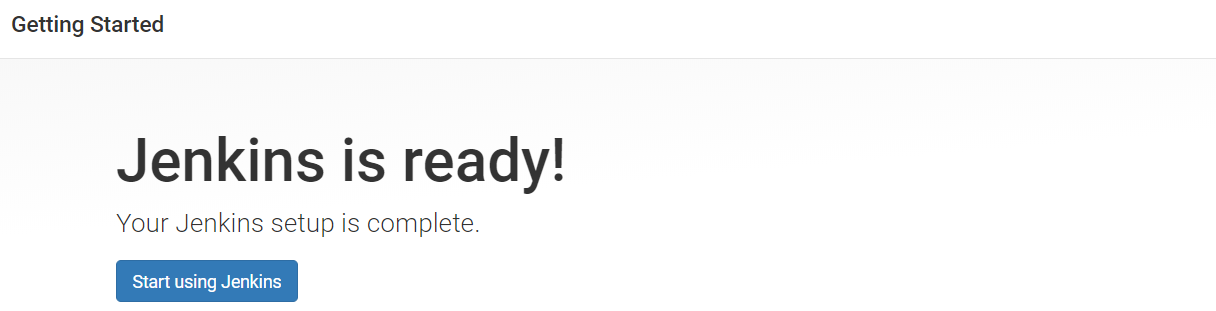
Click on **Install suggested plugins**



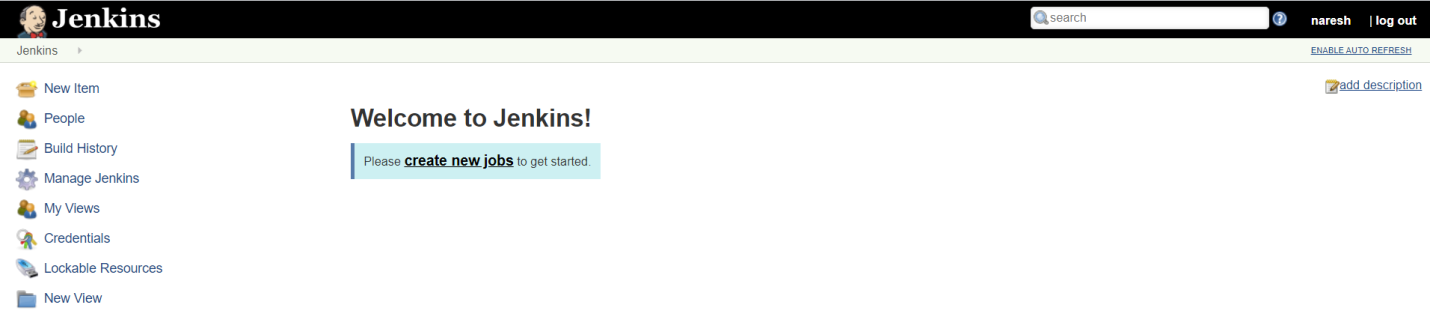
Provide details as above and click on **Save and Continue**



Click on **Save and Finish**



Click on **Start using Jenkins**



**Install MYSQL:**

wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

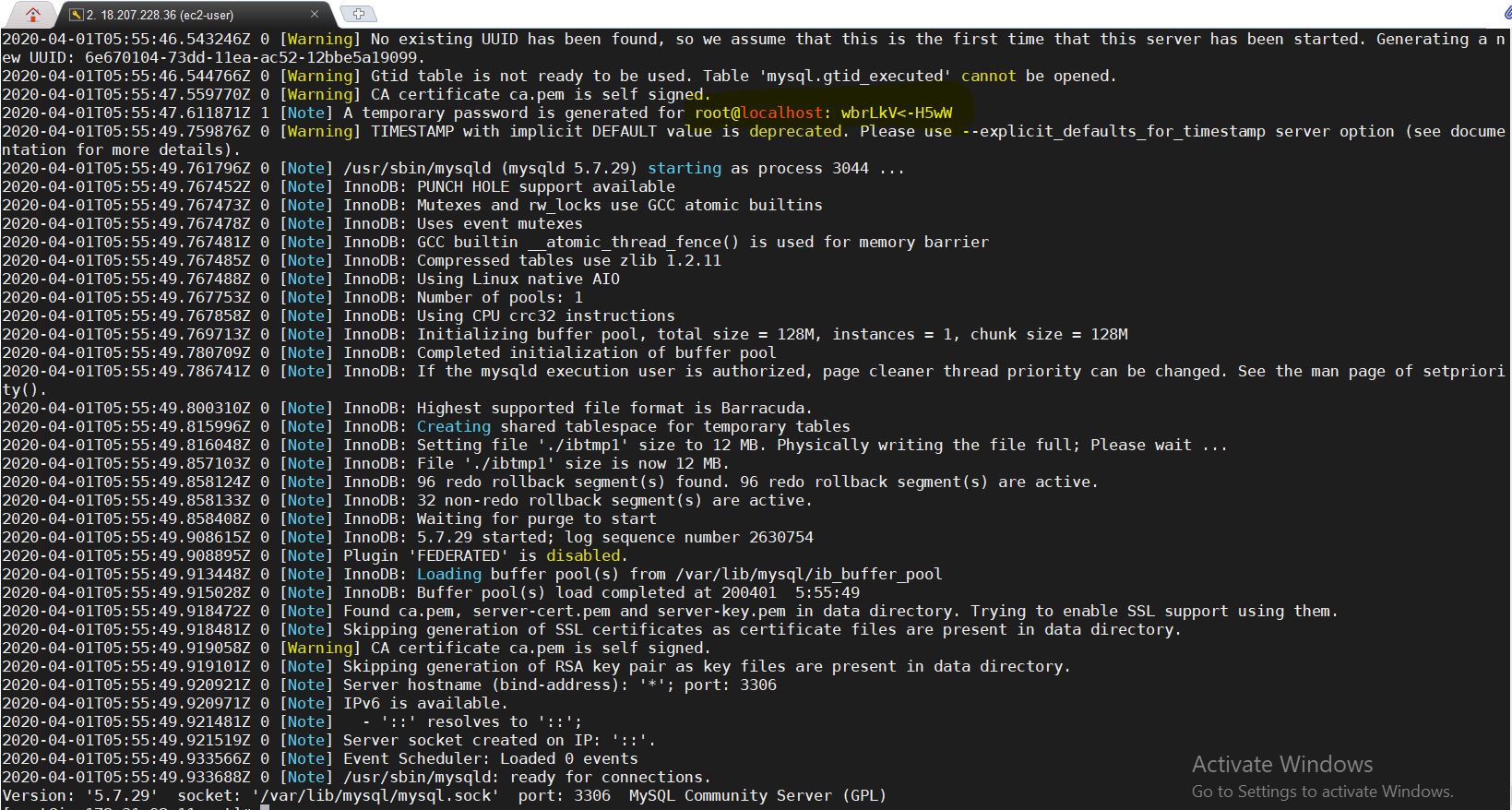
yum localinstall mysql57-community-release-el7-11.noarch.rpm -y

yum install mysql-community-server -y

systemctl start mysqld.service

# Here we get pwd

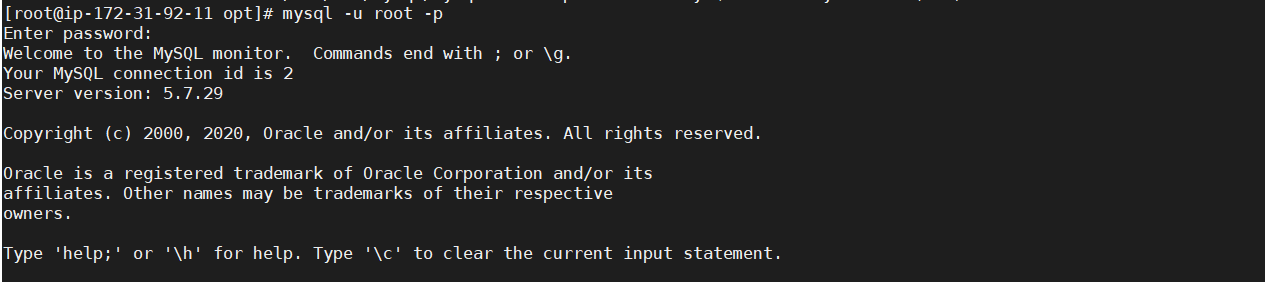
cat /var/log/mysqld.log



Copy **Password**

# Mysql Login

mysql -u root -p



Change password for root User:

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Naresh#240';

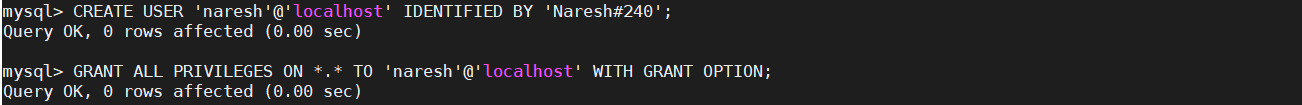


Create new USER:

CREATE USER 'naresh'@'localhost' IDENTIFIED BY 'Naresh#240';

GRANT ALL PRIVILEGES for the USER:

GRANT ALL PRIVILEGES ON \*.\* TO 'naresh'@'localhost' WITH GRANT OPTION;



When we grant some privileges for a user, running the command flush privileges will reloads the grant tables in the mysql database enabling the changes to take effect without reloading or restarting mysql service.

FLUSH PRIVILEGES;



**Install Docker:**

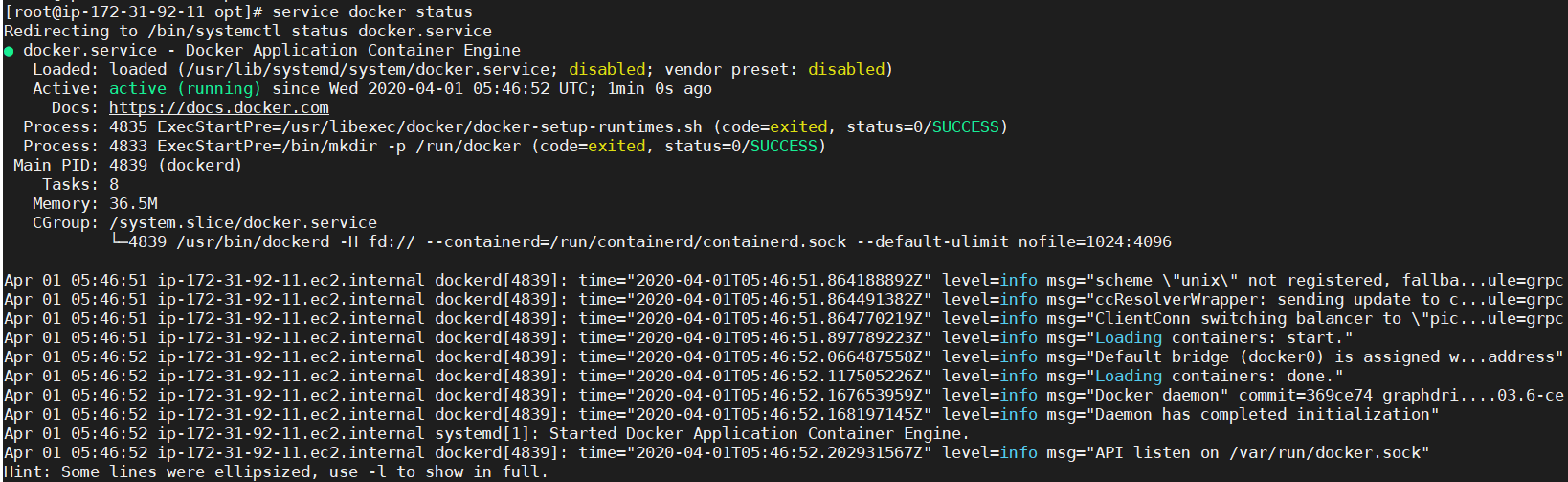
amazon-linux-extras install docker -y

service docker start

usermod -a -G docker ec2-user

Check docker installed or not:

service docker status



**Install Docker-Compose:**

sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

docker-compose -version

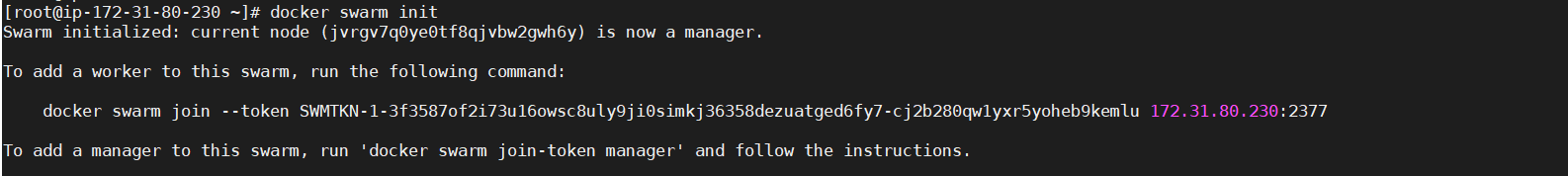
**Docker Swarm:**

For Docker Swarm we need two servers

Take one as Master and other as node

Goto master server and give below command:

docker swarm init



Here we get **token.** This token we need to give in node servers

Before going to add node to Master please allow port number: 2377 in both the servers

Goto node server and give token which we get in master:

docker swarm join --token SWMTKN-1-3f3587of2i73u16owsc8uly9ji0simkj36358dezuatged6fy7-cj2b280qw1yxr5yoheb9kemlu 172.31.80.230:2377



Check nodes in Master server:

docker node ls

