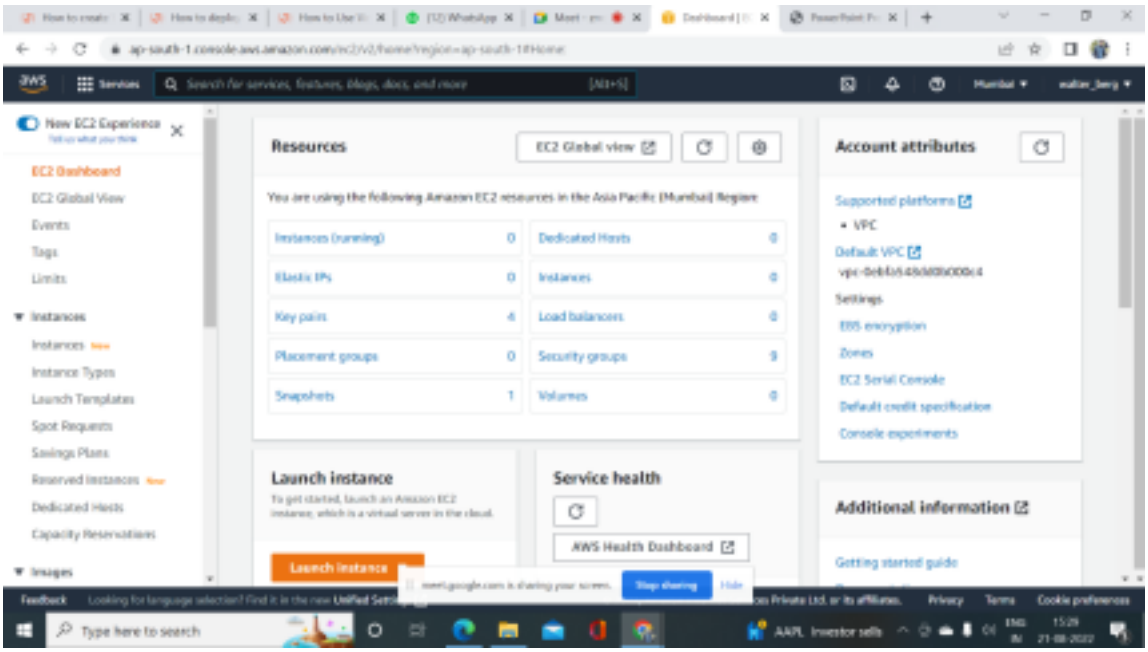
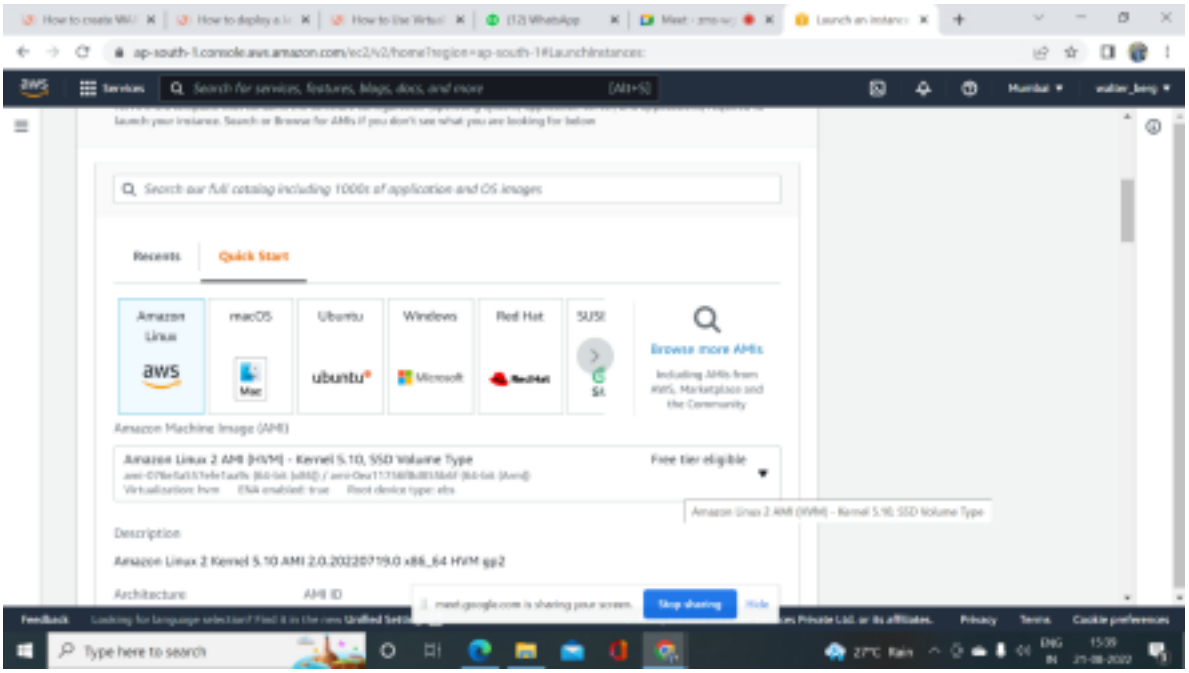
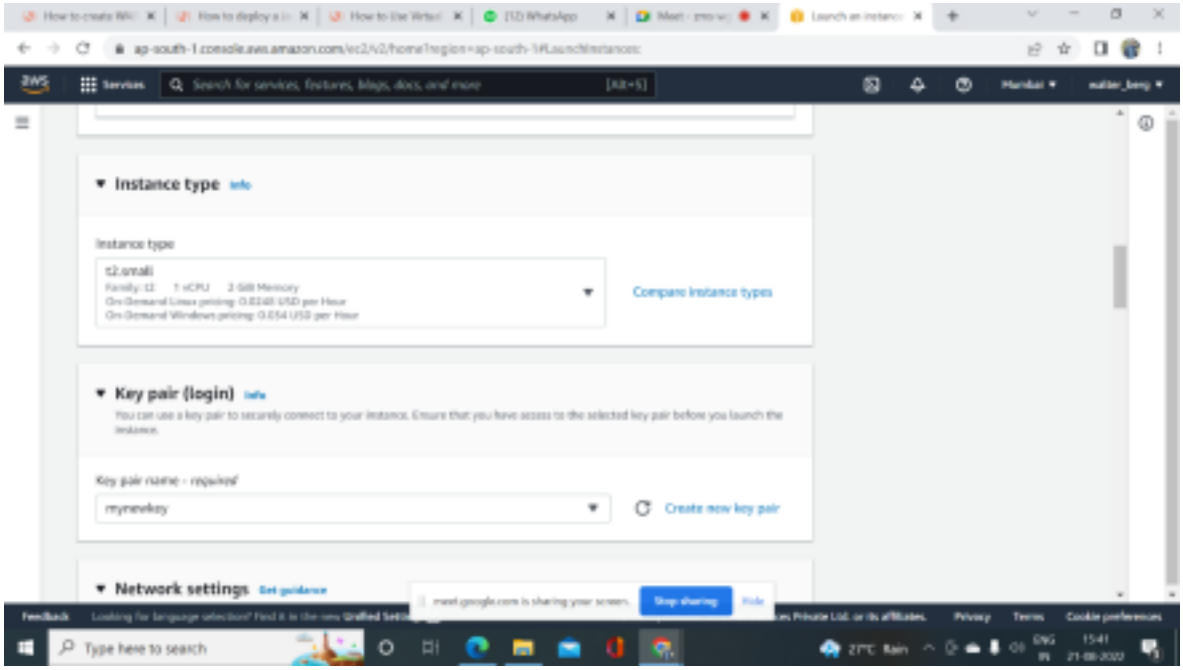
Create Selenium Grid (Using Docker Compose) and Execute the test cases in parallel with different browsers at single instance

Step 1: Login to Aws Console and Launch an ec2 dashboard

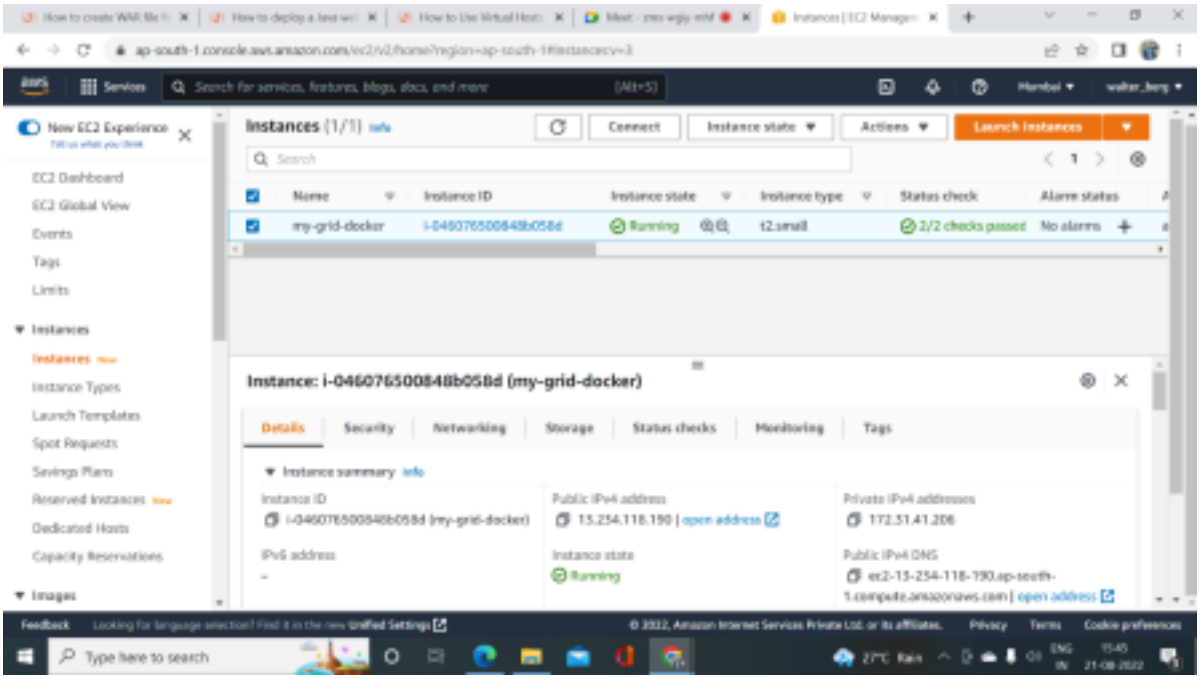


Step 2: Launch an Amazon Linux EC2 Instance

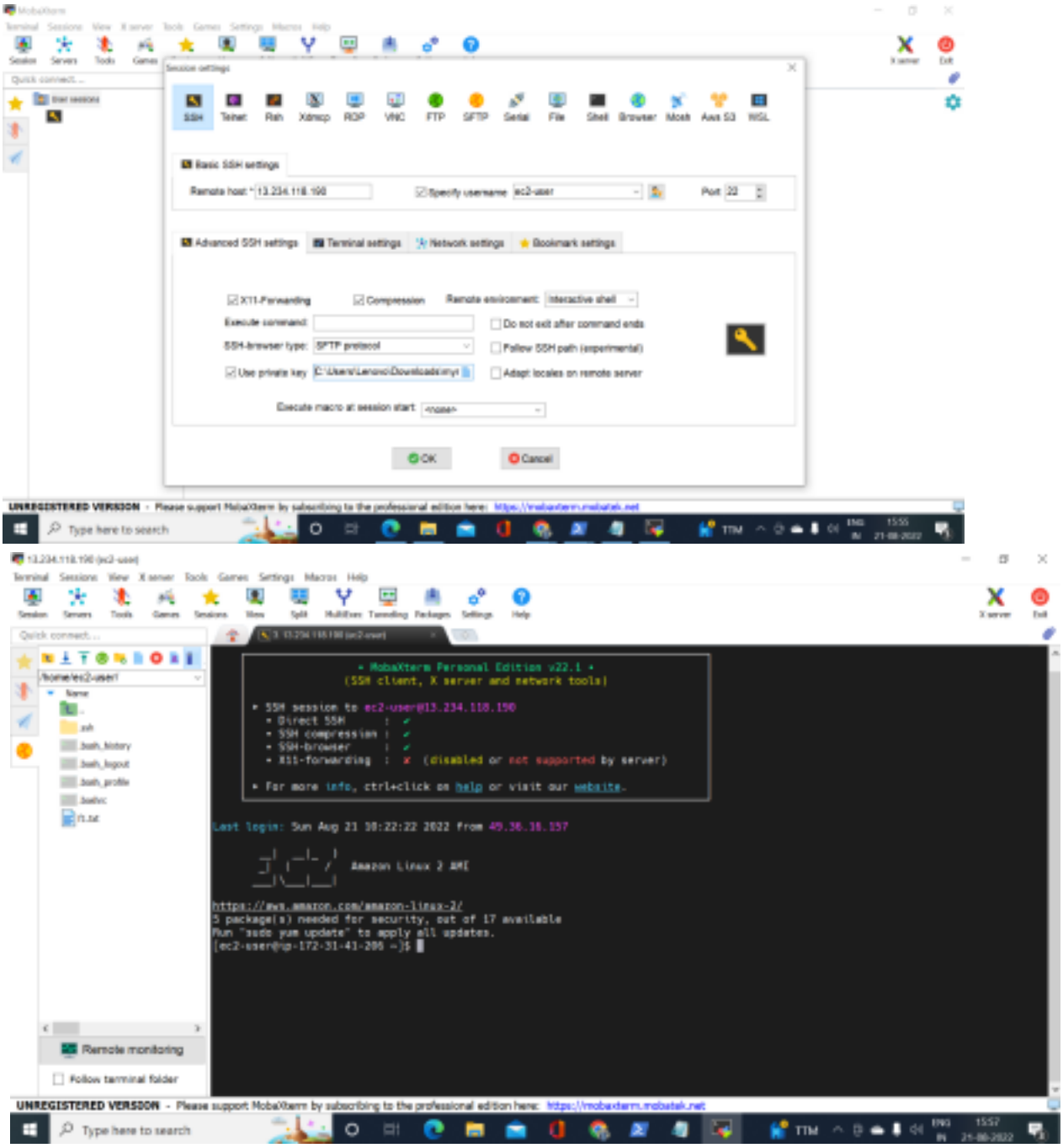




Instance is running



Step 3: Access the server from Mobaxterm

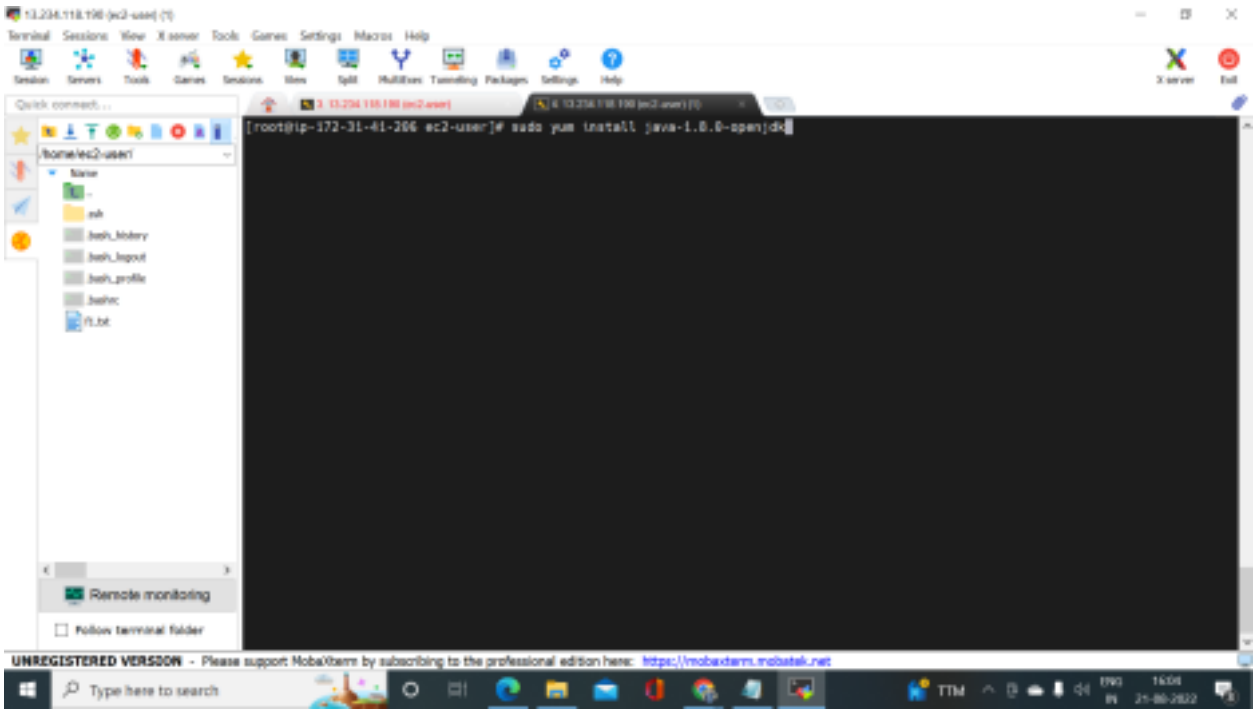


Instance is running and up

Step 4: we need to install java

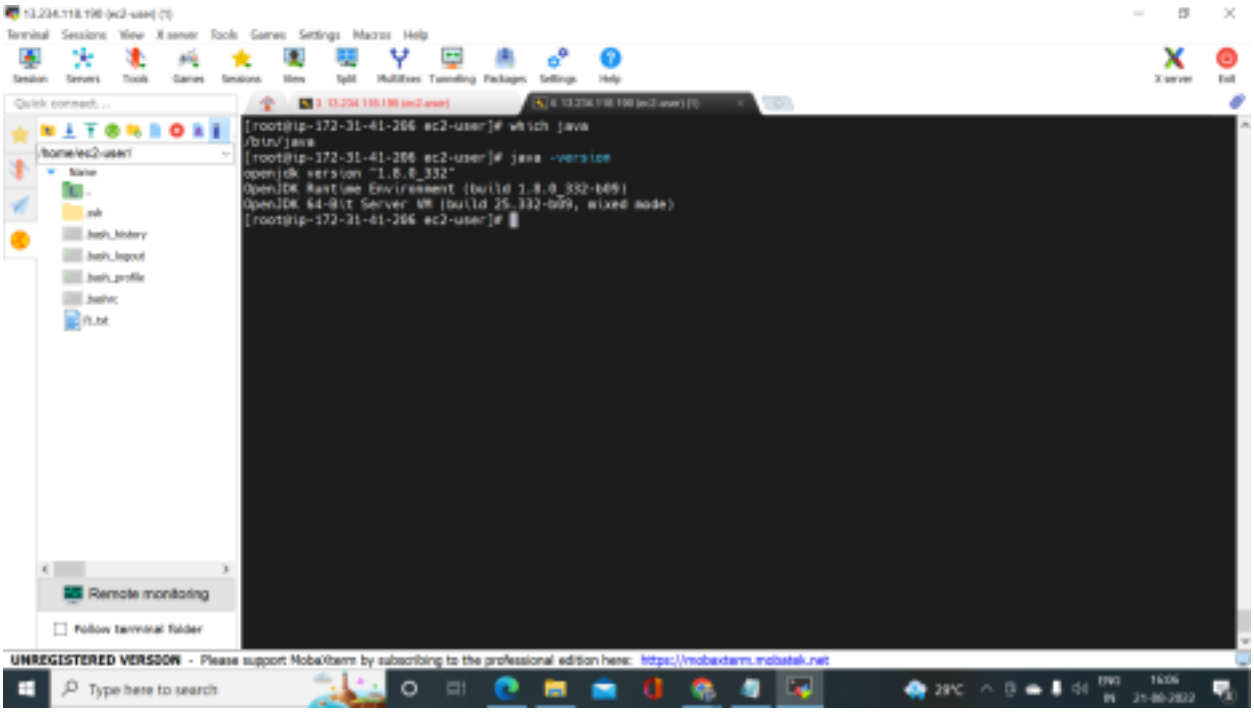
Command to install java:

sudo yum install java-1.8.0-openjdk



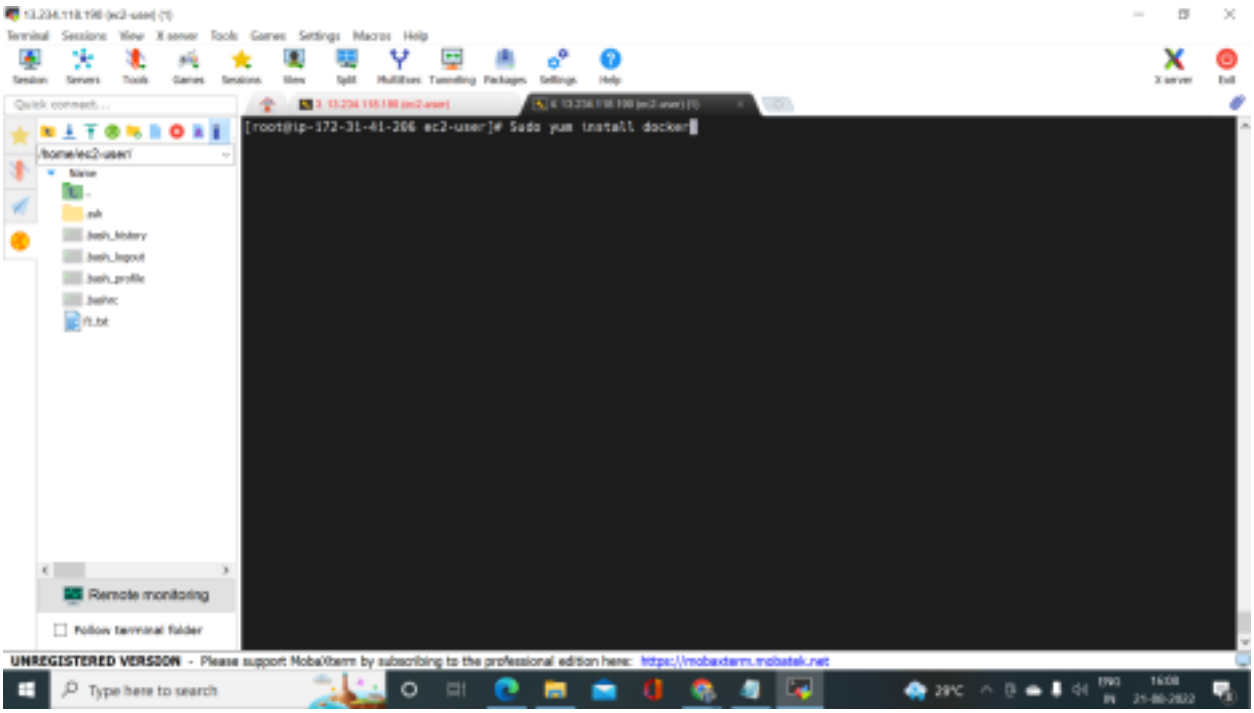
Now check whether java is installed or not with command:

Java -version



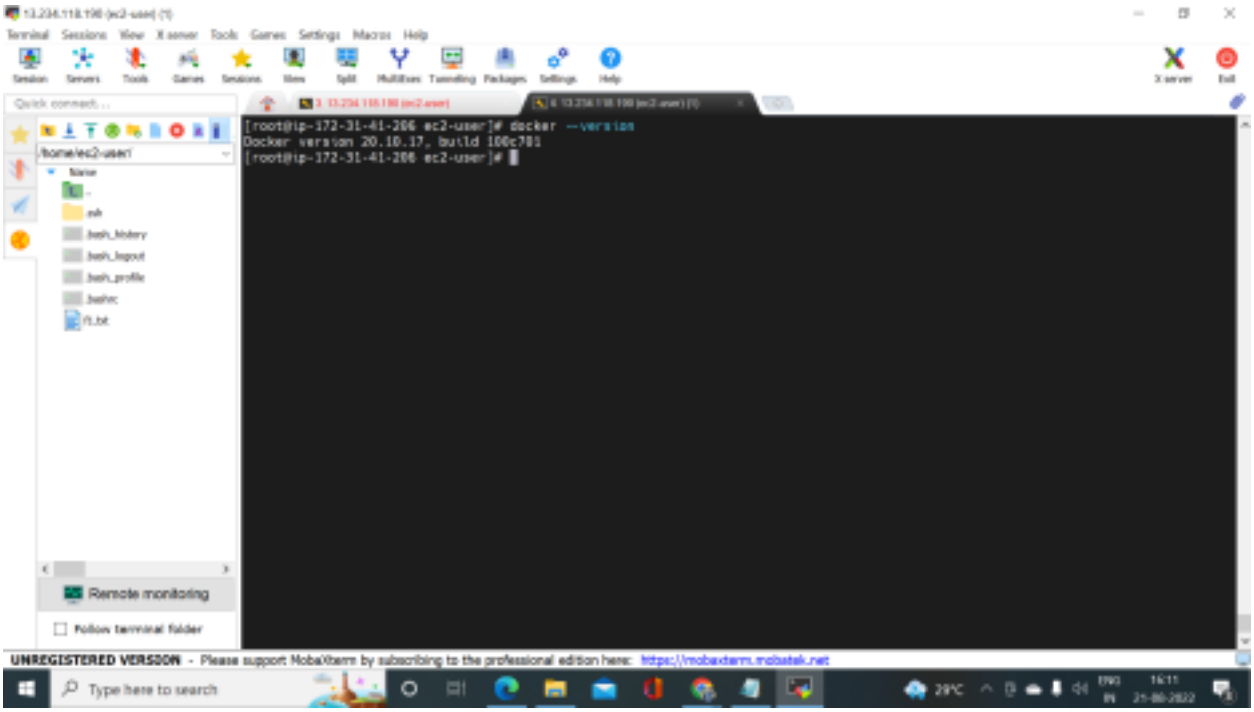
Step 5: we need to install docker with command:

sudo yum install docker



Check docker version with command:

docker --version



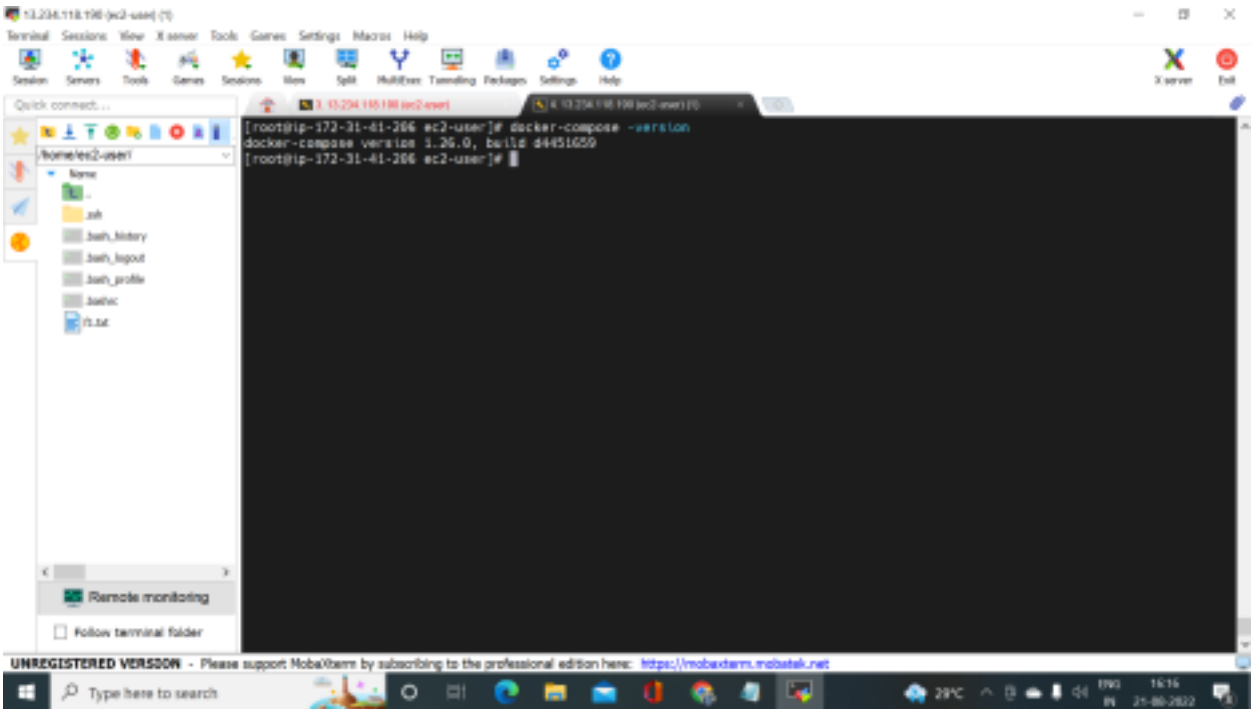
Step 6: after that we need to install docker-compose with commands:

sudo curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

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

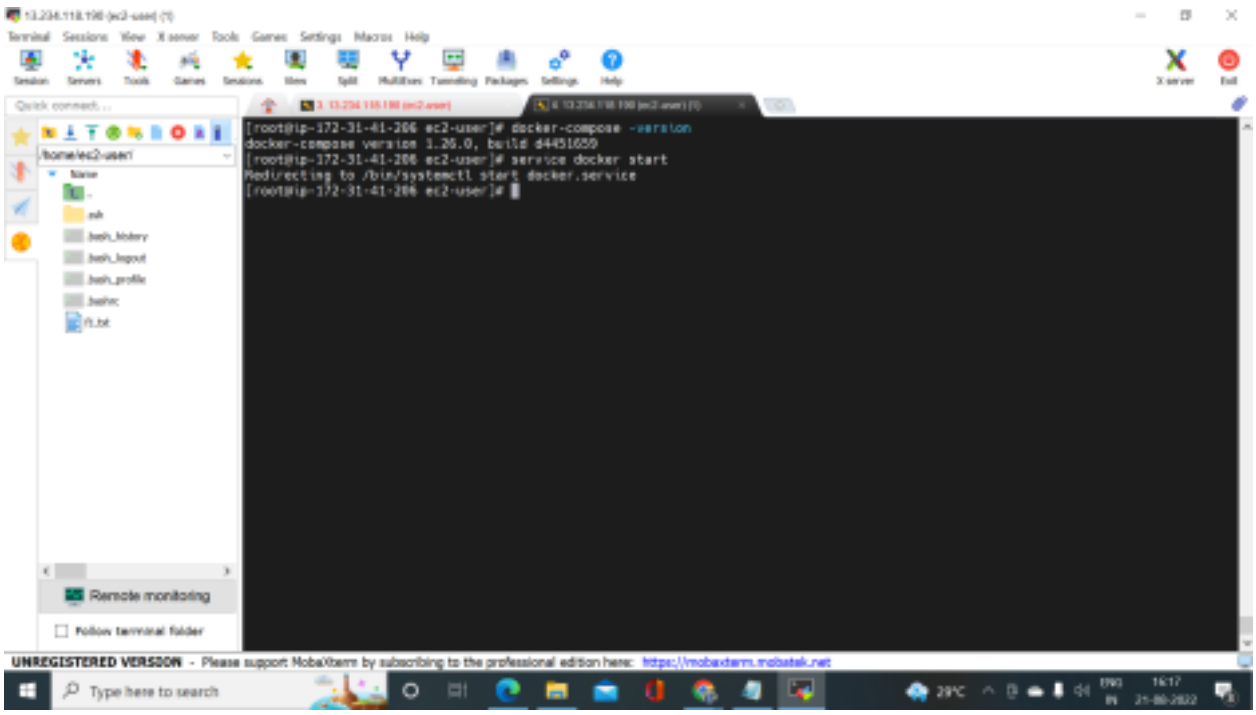
Now we need to check whether docker-compose is installed or not with command:

docker-compose -version



Step 7: we need to start docker service with command:

service docker start



 # To execute this docker-compose yml file use `docker-compose -f docker-compose-v3.yml up`

 # Add the `-d` flag at the end for detached execution

 # To stop the execution, hit Ctrl+C, and then `docker-compose -f docker-compose-v3.yml down`

version: "3"

```
services:
 chrome:
  image: selenium/node-chrome:4.4.0-20220812
  shm_size: 2gb
  depends_on:
    - selenium-hub
  environment:
    - SE_EVENT_BUS_HOST=selenium-hub
    - SE_EVENT_BUS_PUBLISH_PORT=4442
    - SE_EVENT_BUS_SUBSCRIBE_PORT=4443

 edge:
  image: selenium/node-edge:4.4.0-20220812
  shm_size: 2gb
  depends_on:
    - selenium-hub
  environment:
    - SE_EVENT_BUS_HOST=selenium-hub
    - SE_EVENT_BUS_PUBLISH_PORT=4442
    - SE_EVENT_BUS_SUBSCRIBE_PORT=4443

 firefox:
  image: selenium/node-firefox:4.4.0-20220812
  shm_size: 2gb
  depends_on:
    - selenium-hub
  environment:
```

- SE_EVENT_BUS_HOST=selenium-hub

- SE_EVENT_BUS_PUBLISH_PORT=4442

- SE_EVENT_BUS_SUBSCRIBE_PORT=4443

selenium-hub:

image: selenium/hub:4.4.0-20220812

container_name: selenium-hub

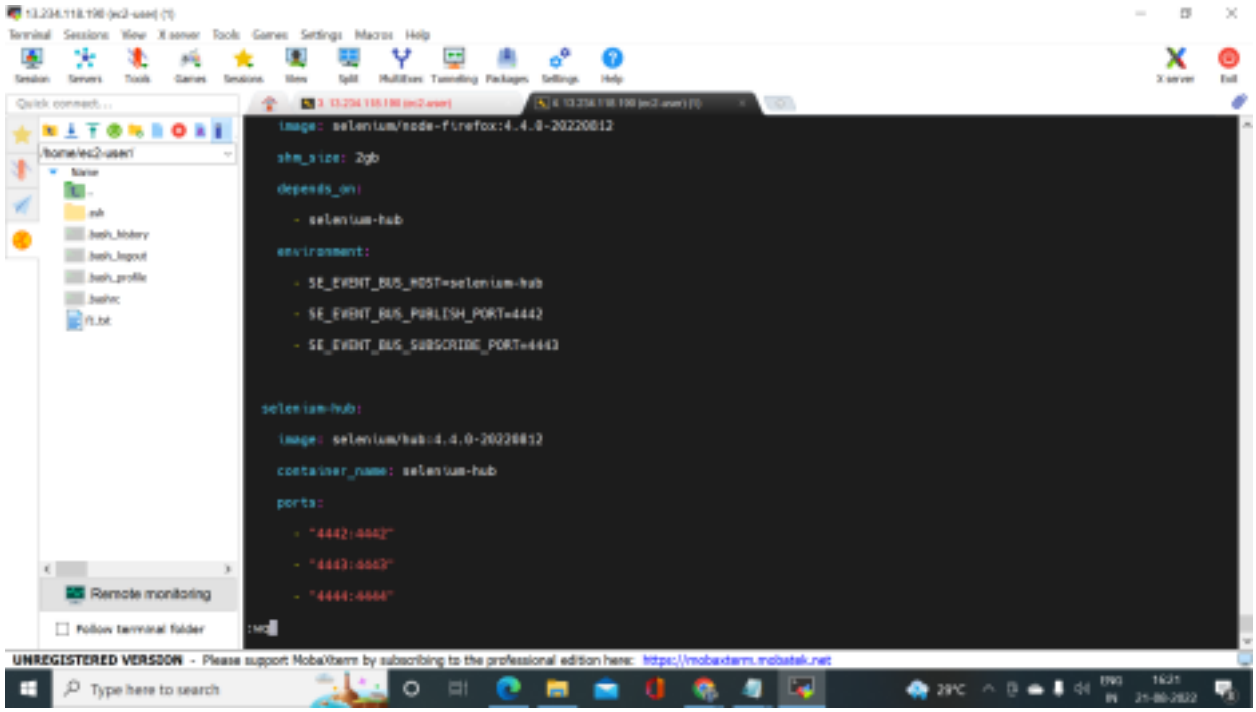
ports:

- "4442:4442"

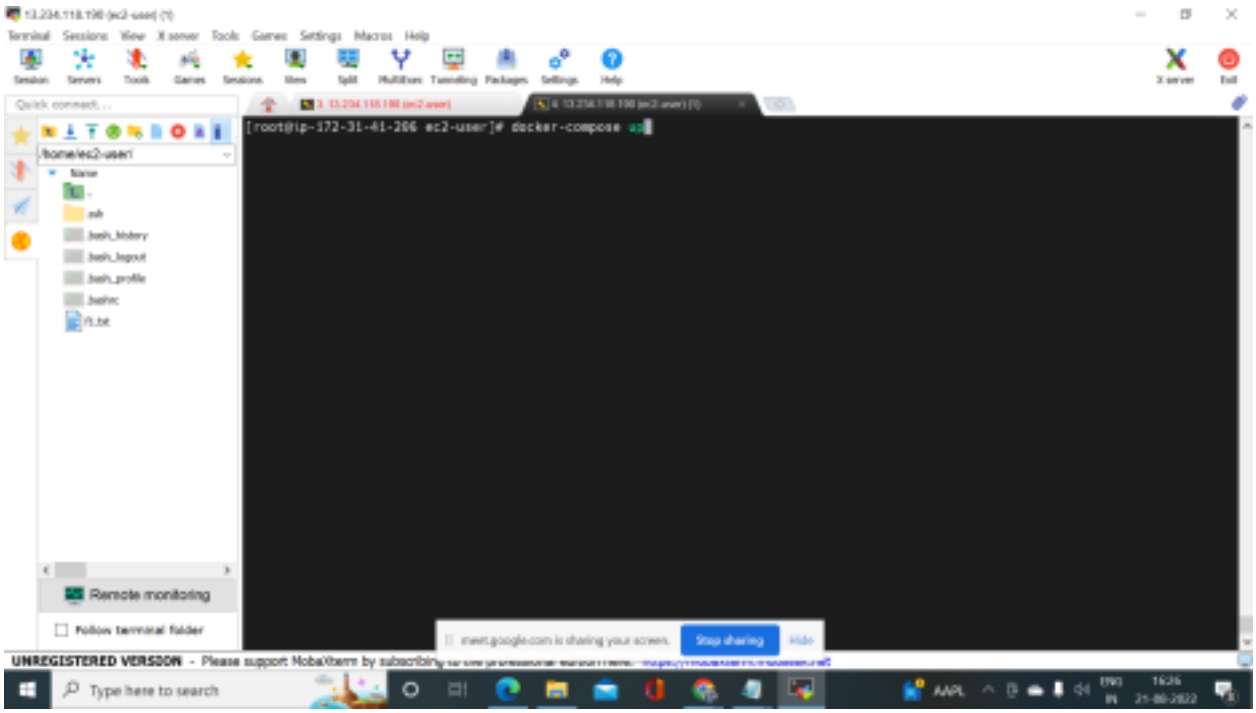
- "4443:4443"

- "4444:4444"

Step 8: After that we need to create file docker-compose.yml in order to run more than one containers in a one go.

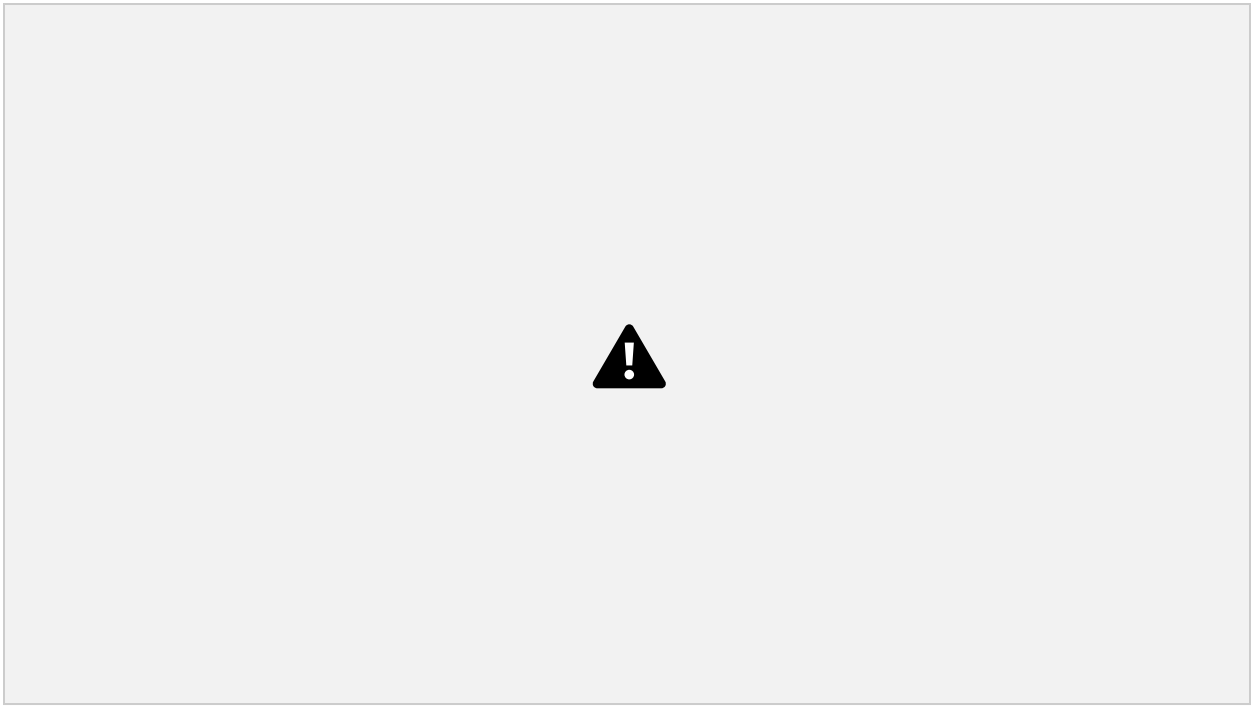


Step 9 : Now we need to run our docker-compose file to setup the selenium hub and nodes: **docker-compose up**



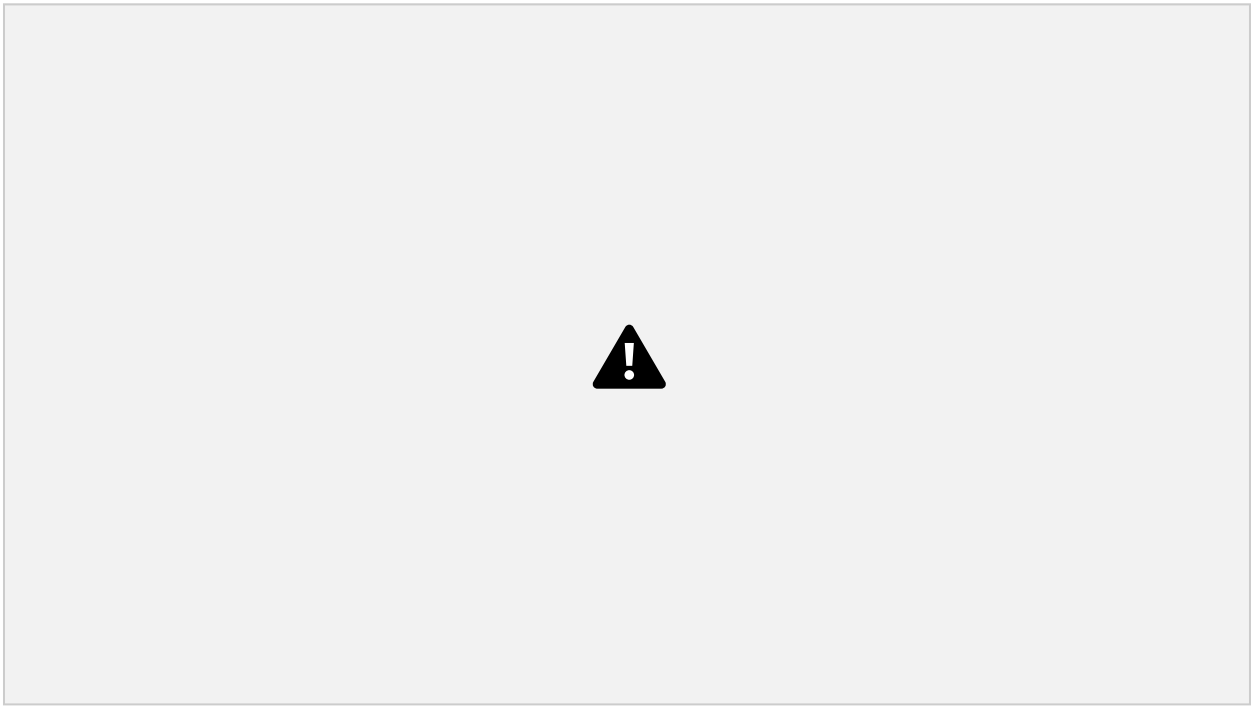
We can check docker images with command

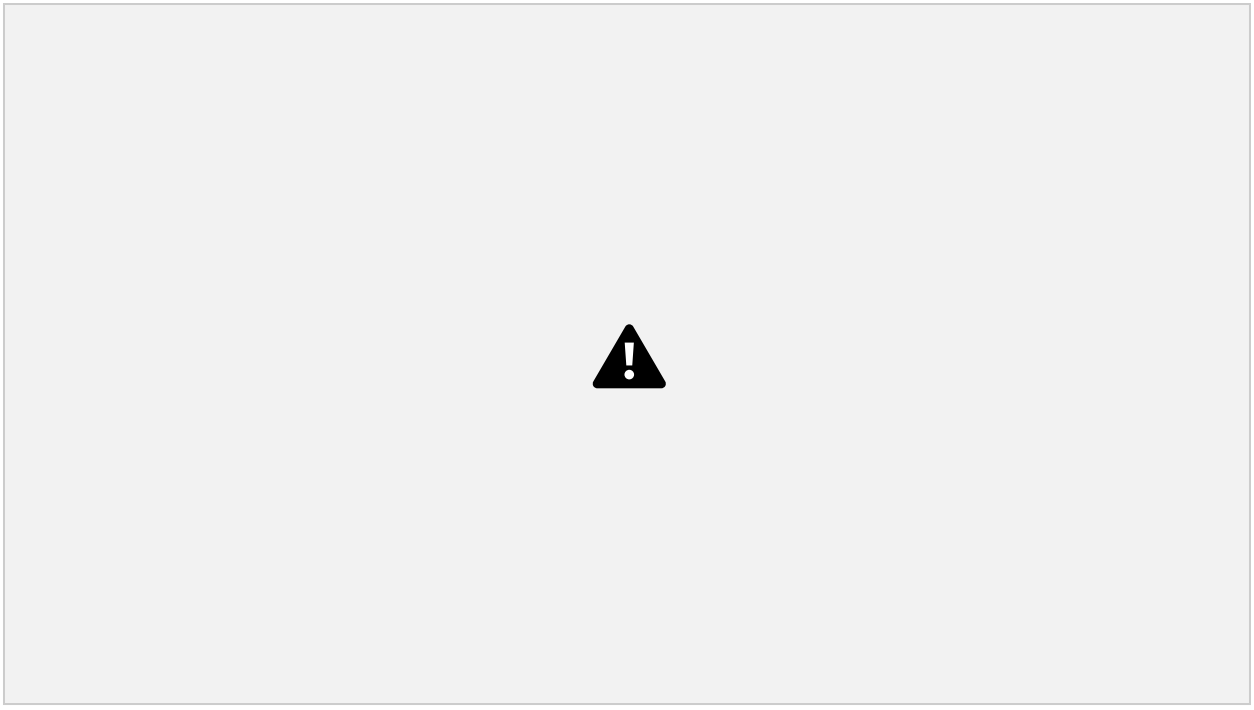
docker images



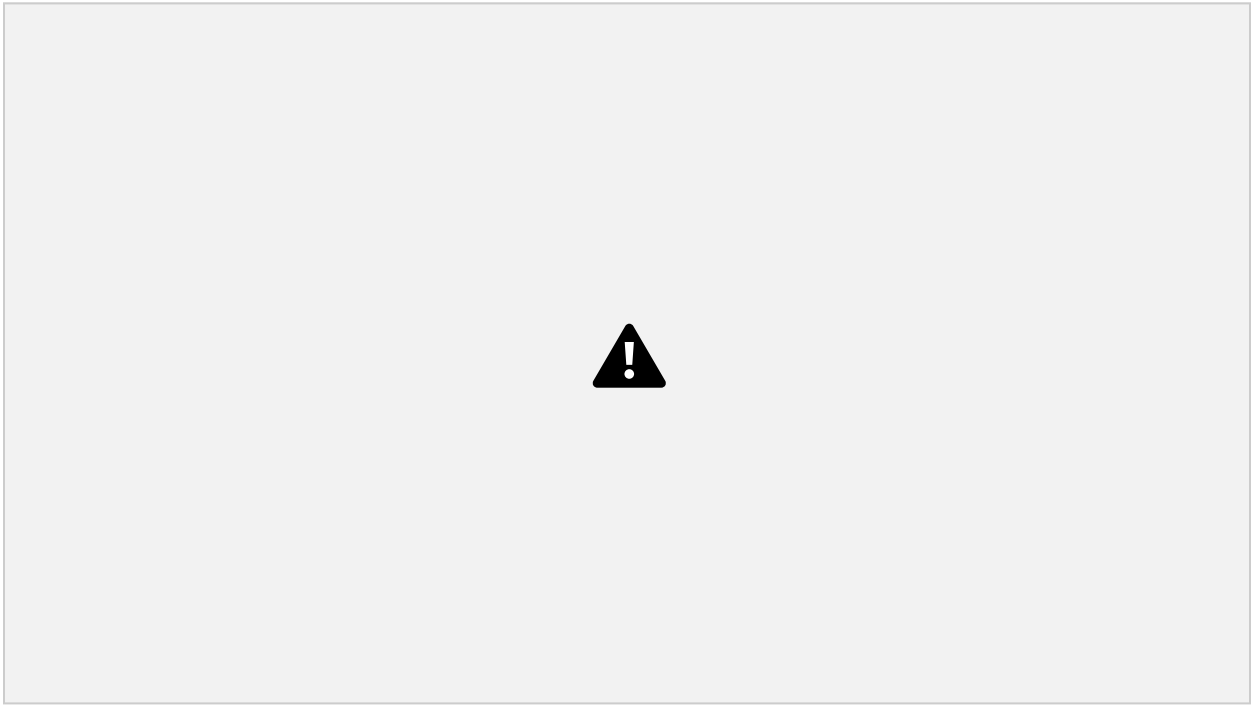
In order to check running Containers we use command

docker ps

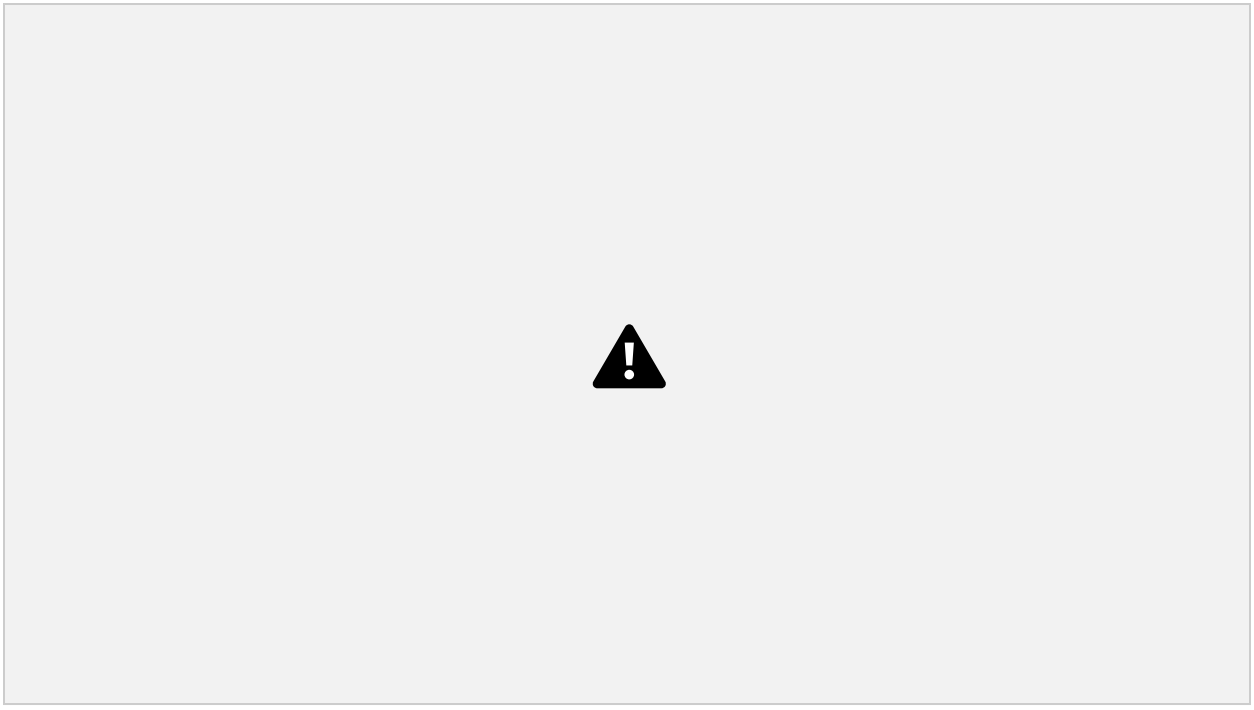




Step 10: After that our selenium hub and nodes are ready and up and we can check that by accessing them with public ip of our amazon ec2 instance



After that we can run our selenium testcases on eclipse by providing url of our hub



All test cases are running on grid paralelly on different browser nodes.