**INDUSTRY GRADE PROJECT 1**

**SUMMARY:**

I choose project-1 as my submission in this course. Working on real time operation to enhance the business of ABC technologies.

**AIM:**

Implement CI/CD such that ABC Company is able to be

* + highly available
  + highly scalable
  + easily built and maintained
  + developed and deployed quickly

**GIT REPO LINK:** https://github.com/Maneeshnirmal/Project\_submit

**SOFTWARE IS INSTALLED IN THE WORKING MACHINE**

1. JAVA

2. MAVEN

3. GIT

4. JENKINS

5. DOCKER

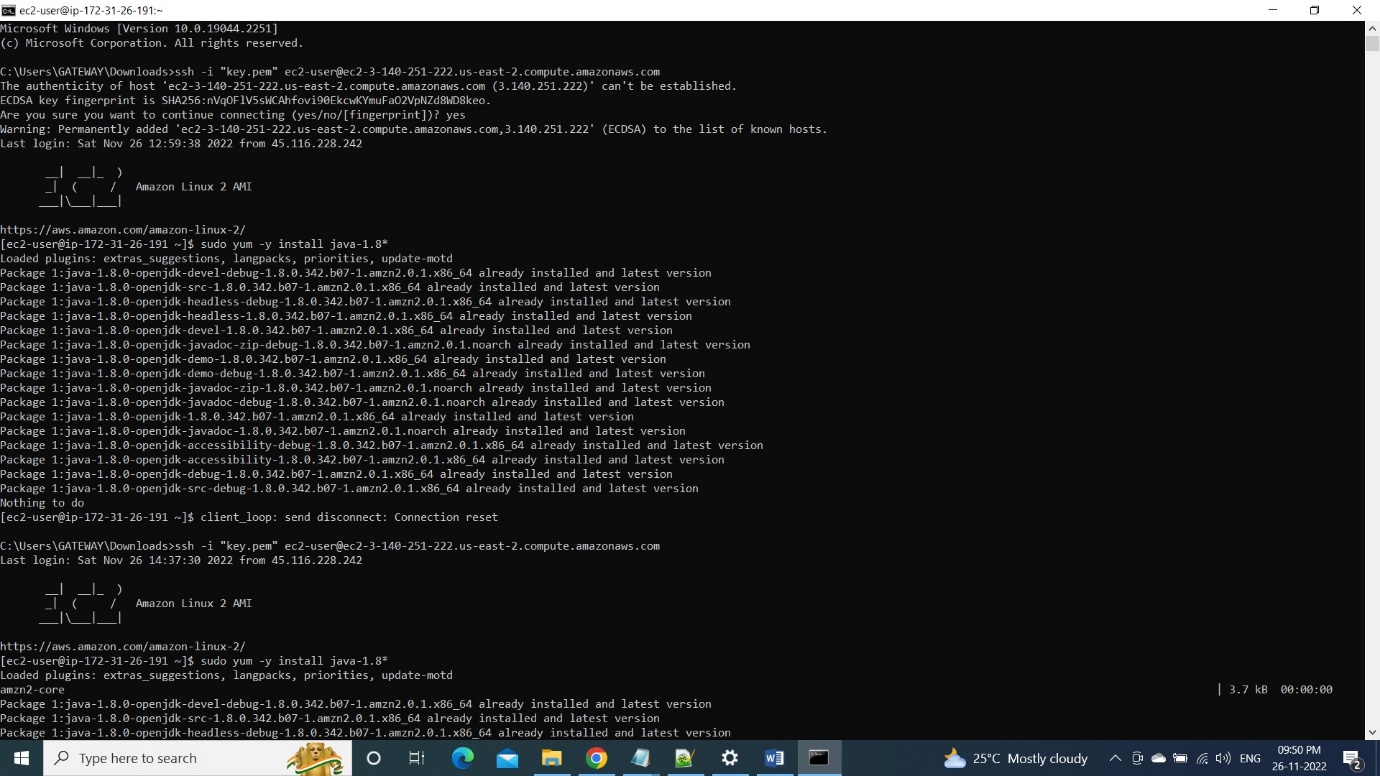
6. ANSIBLE

7. KUBERNETES

8. GRAFANA

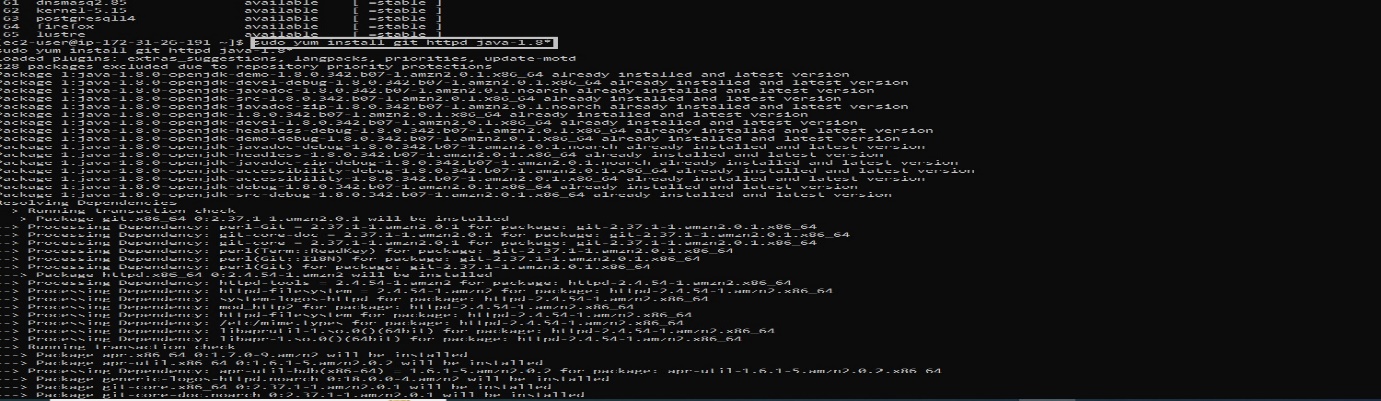
9. PROMETHEUS

**PROCEDURE FOLLOWED STEP BY STEP**

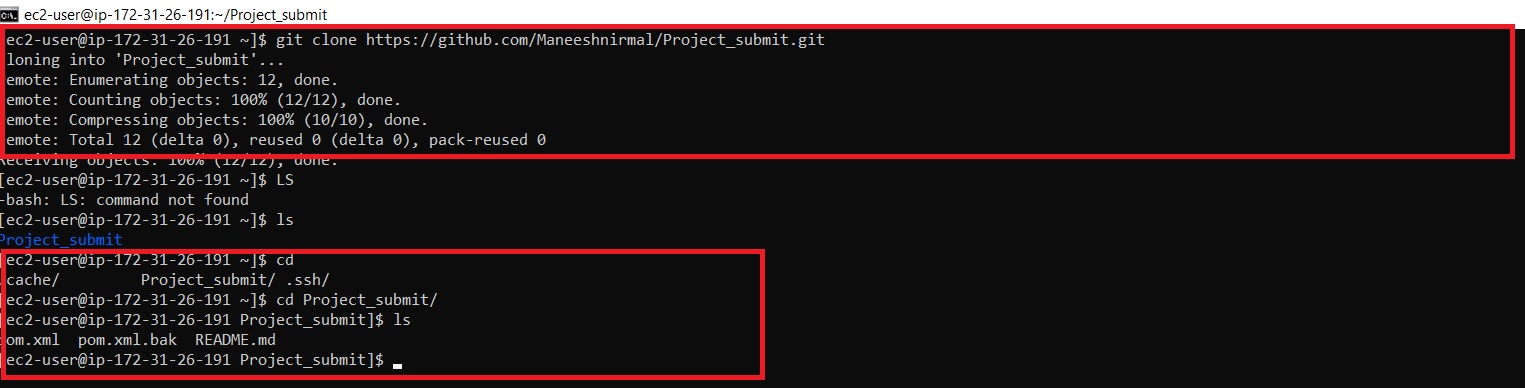
**1.AWS INSTANCE CONFIGRUATION**

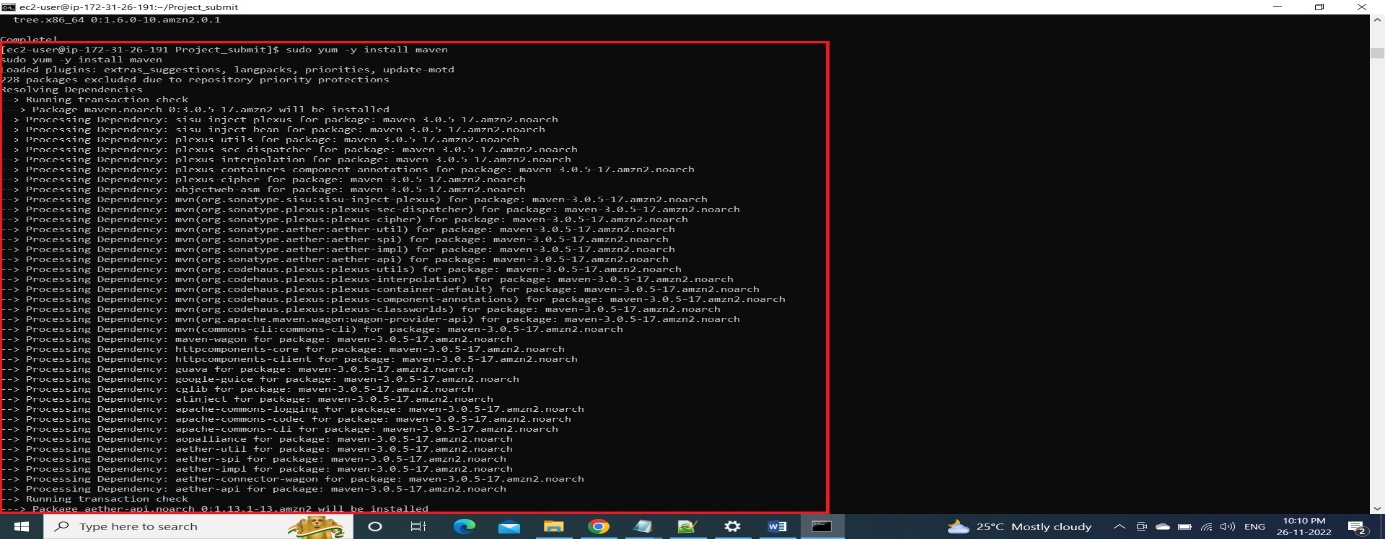
**2.GIT config**

**2A.INSTALLATION OF GIT**

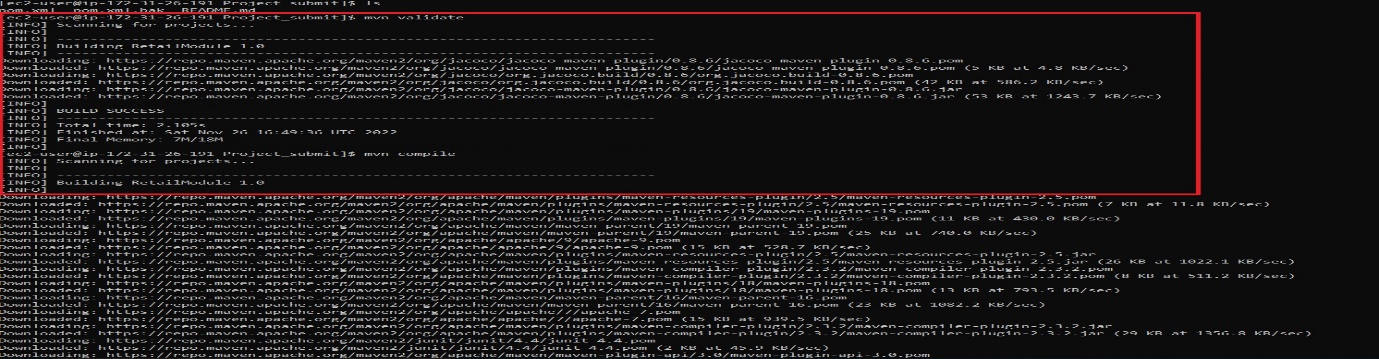
****

**2B.CLONING OF REPO**

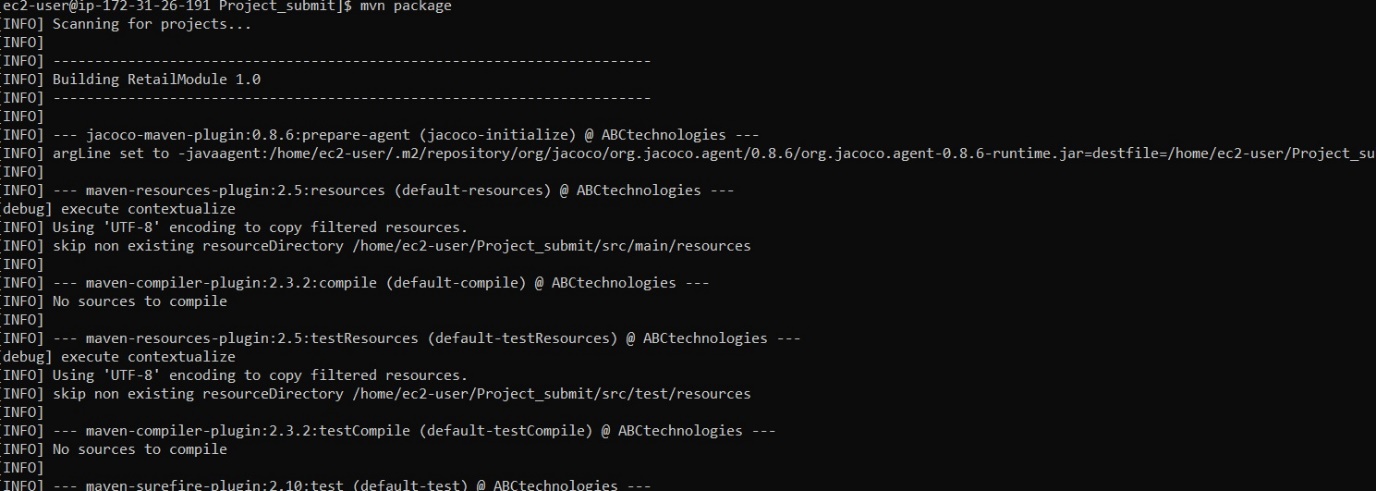
****

**3. MAVEN installation & working structure**

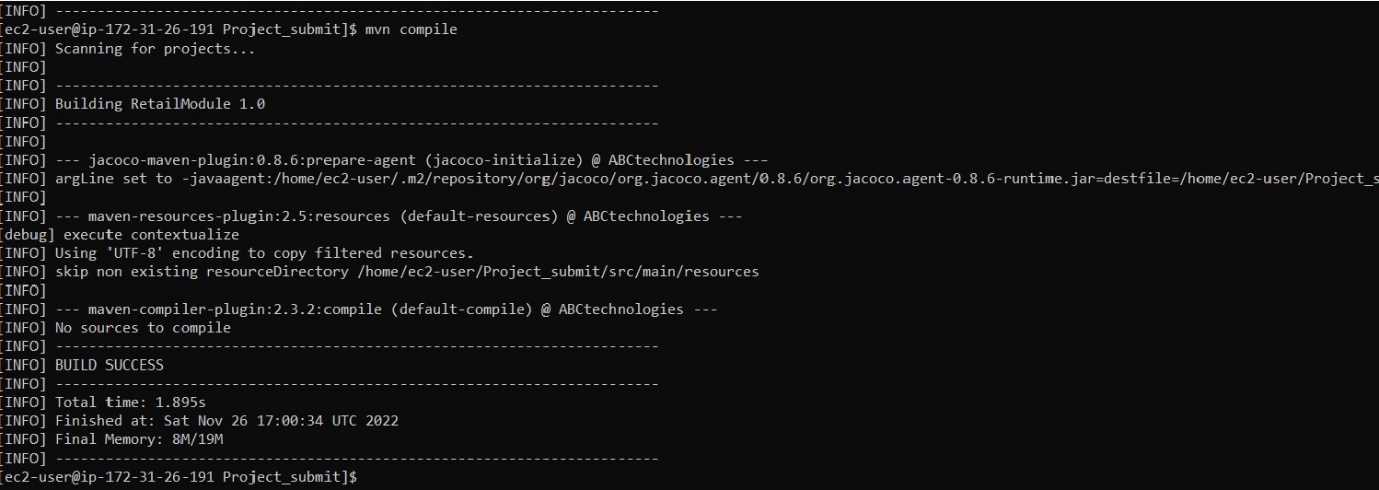
**3a.validate**

****

**3b.package**

****

**3c.compile**

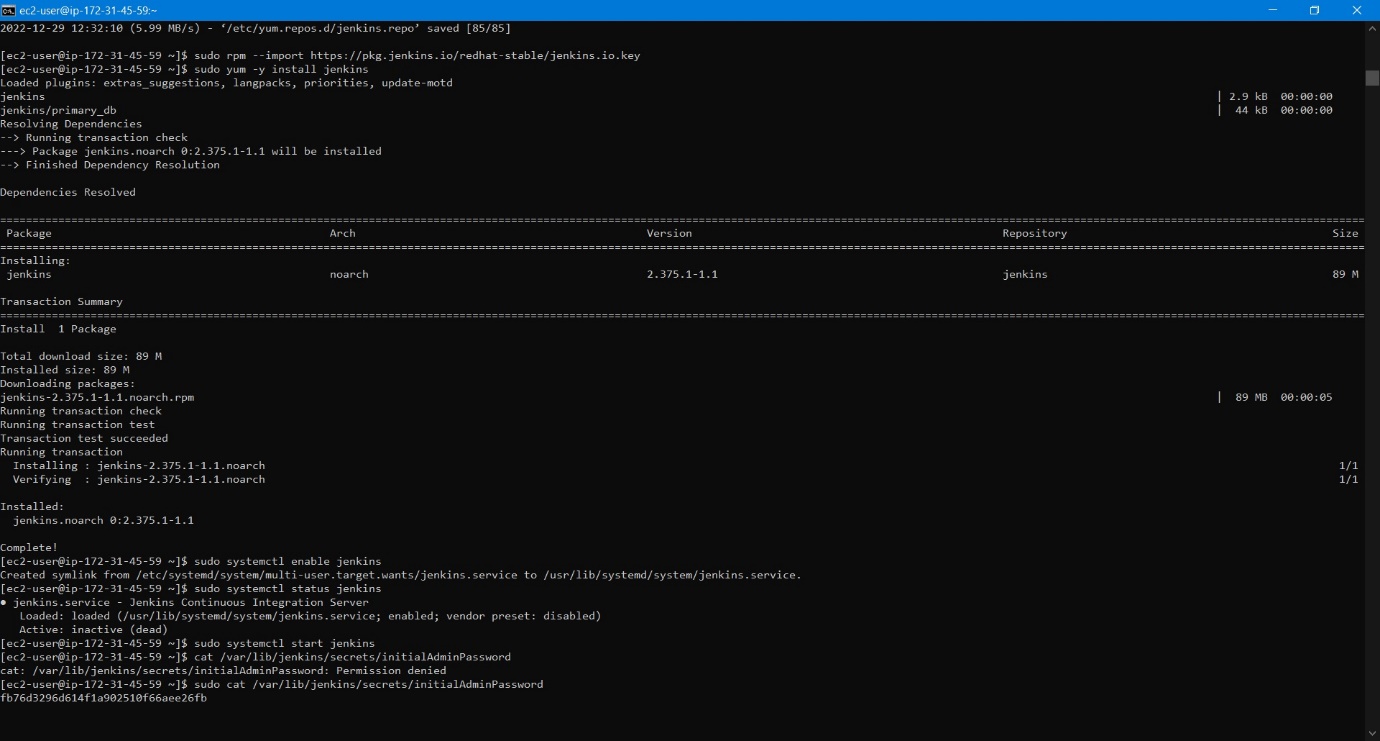
****

**4.JENKINS INSTALLATION**

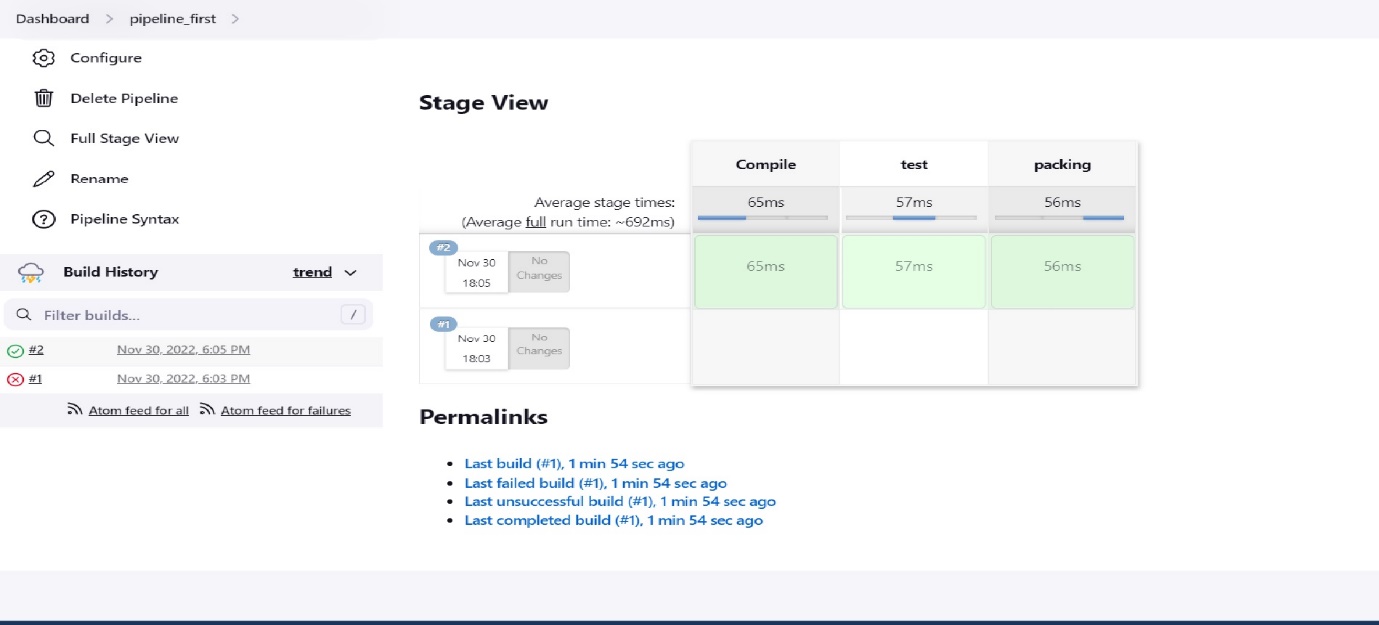
**Procedure followed to setup Jenkins**1. Setup Jenkins

2. Set up custom port: 8080

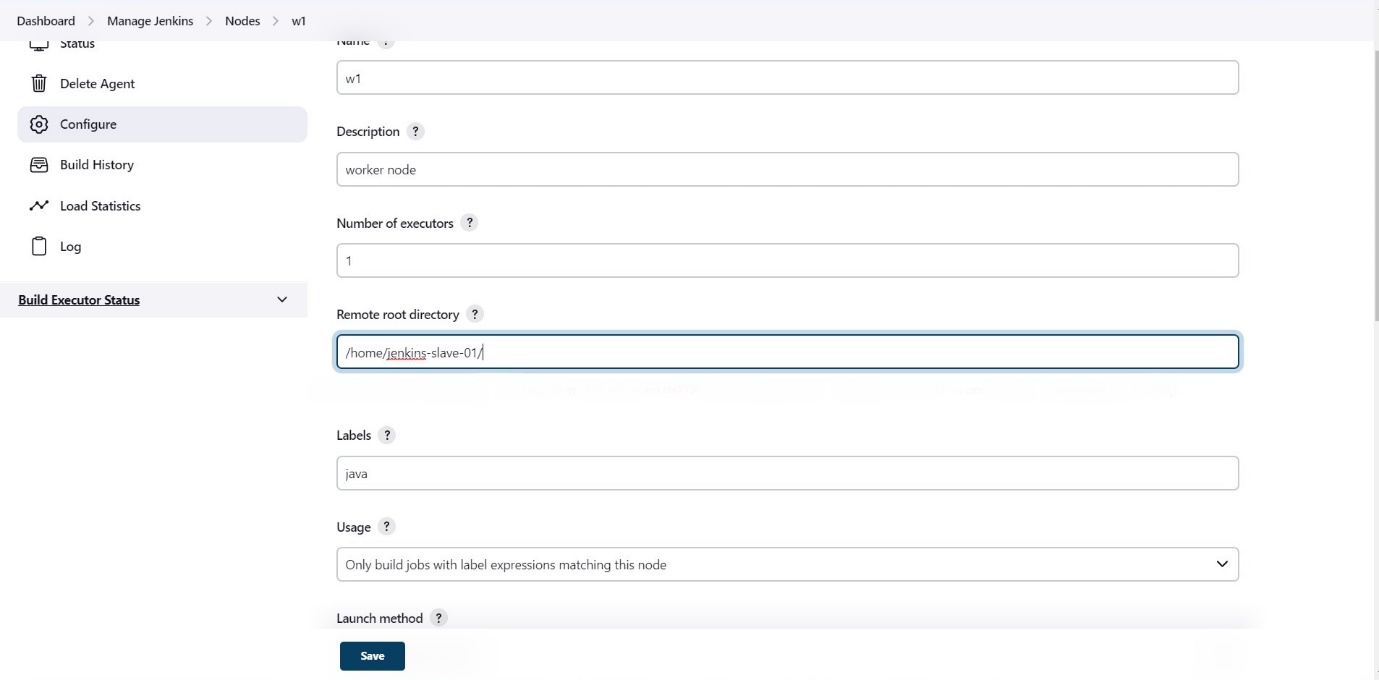
3. Set ssh port range: 22

****

**4a.CI/CD PIPELINE (CODE ADDED IN GIT REPO)**

****

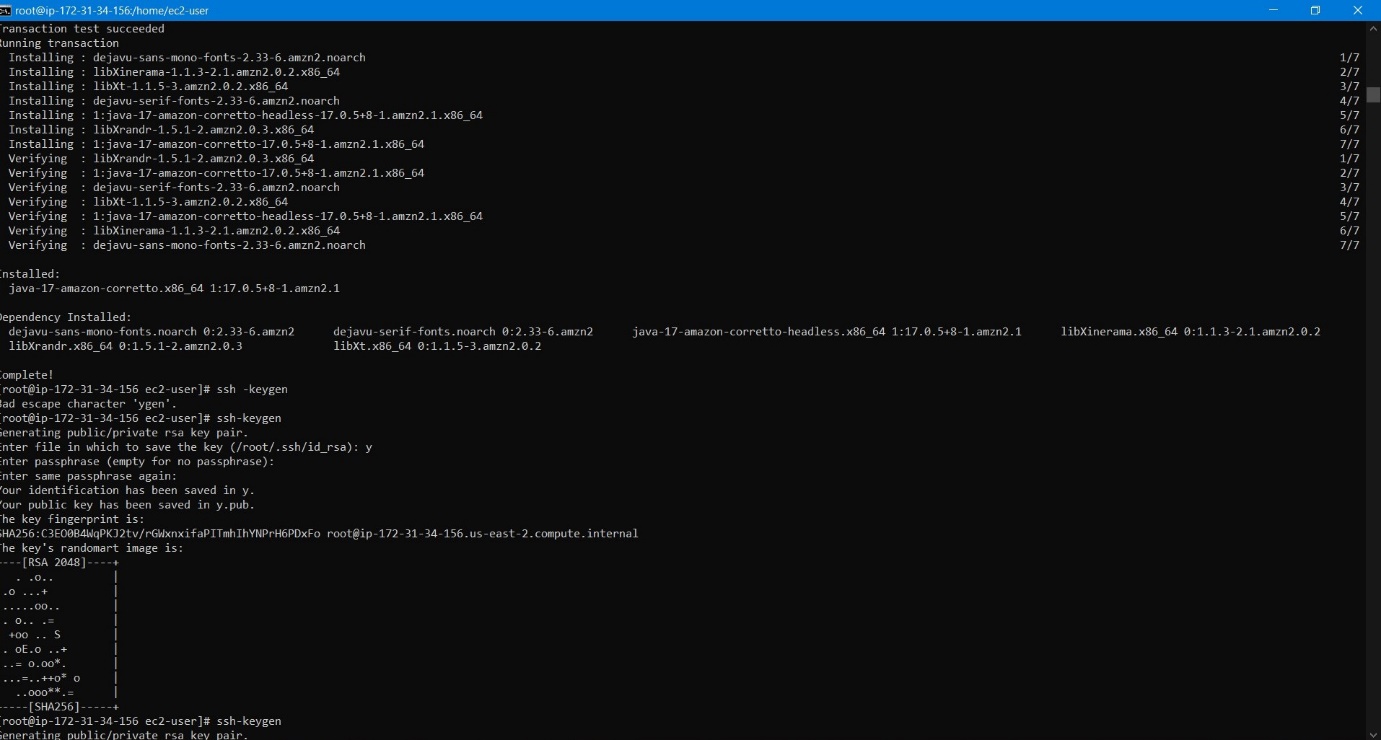
**4. MASTER SLAVE NODE**

****

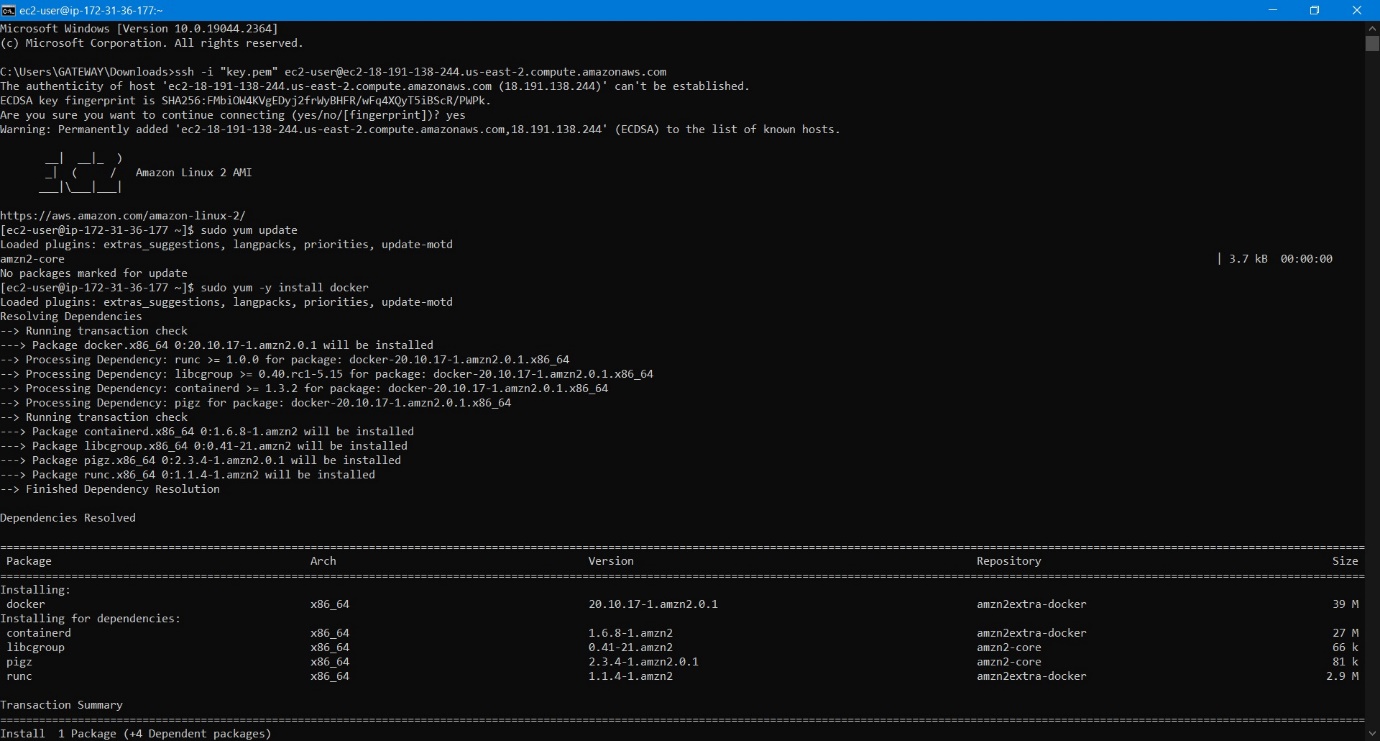
**Procedures followed**

* Mange Jenkins in UI. No external codes used apart from Jenkins installation.
* Configure Jenkins in two installation one is master and another one is worker node.
* Connect internally through IP host.

**5.SSH-KEYGEN**

****

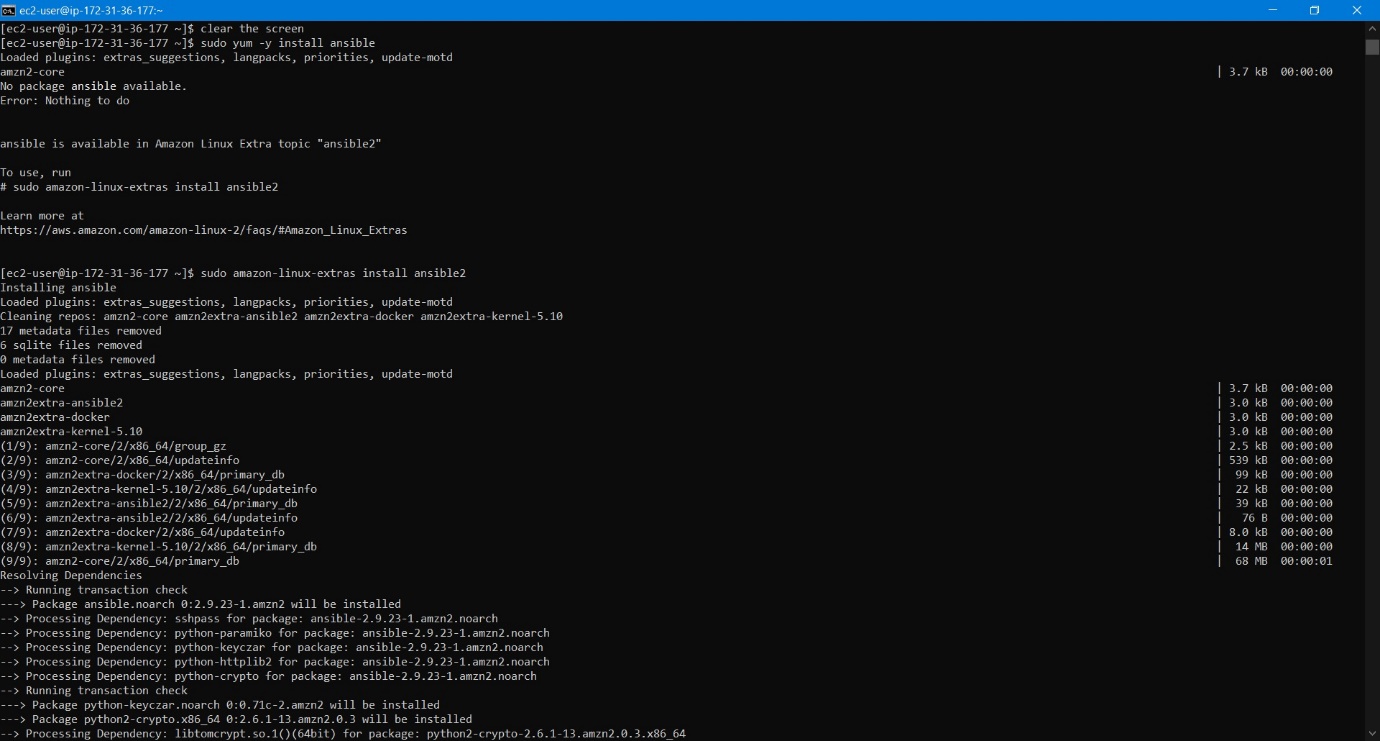
**6. DOCKER INSTALLATION**

****

**Commands used in docker**

1. docker run –dt –name test centos
2. docker ps
3. docker image ls
4. docker build –t <repo\_name>
5. docker run –t <image\_name>
6. docker candy
7. docker push <repo\_name>

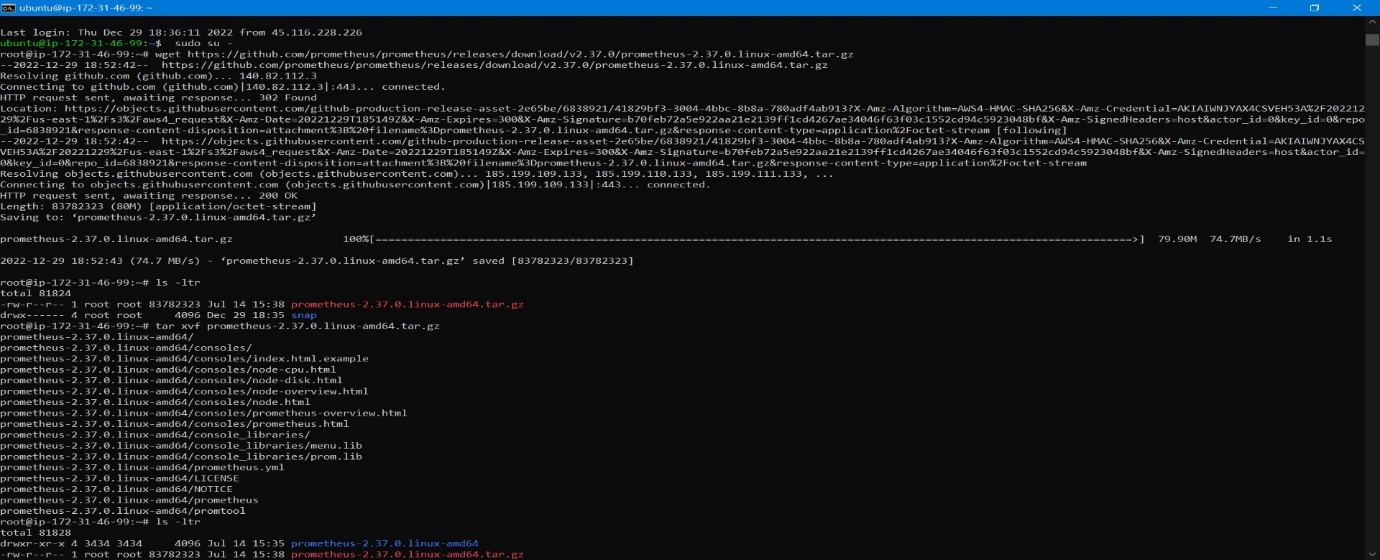
Pushed to repo. Account created..

**7.ANSIBLE INSTALLATION **

**Commands used**

1. kubectl cluster-info

2. kubectl apply -f task4\_ansible.yaml

**8. PROMETHEUS INSTALLATION** 

**Commands used**

1. wget <https://github.com/prometheus/prometheus/releases/download/v2.37.0/prometheus-2.37.0.linux-amd64.tar.gz>
2. ls –ltr
3. tar xvf prometheus-2.37.0.linux-amd64.tar.gz
4. rm -rf prometheus-2.37.0.linux-amd64.tar.gz

5. kubectl apply -f task4\_ansible.yaml  
6. mkdir -p /etc/Prometheus

7. mkdir -p /var/lib/Prometheus

8. mv prometheus promtool /usr/local/bin/

9. mv consoles console\_libraries /etc/Prometheus

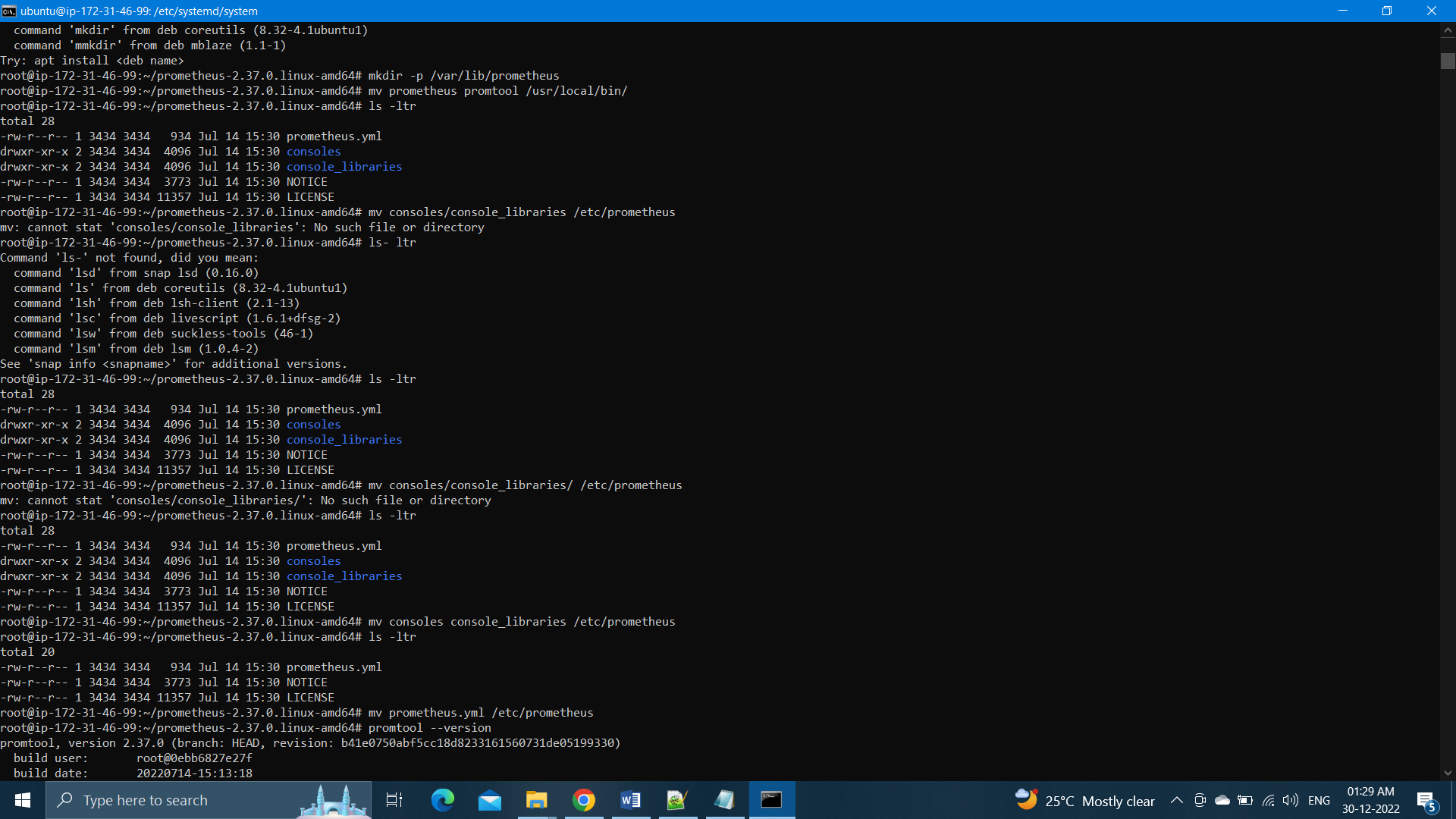
10. mv prometheus.yml /etc/Prometheus

11. groupadd --system Prometheus

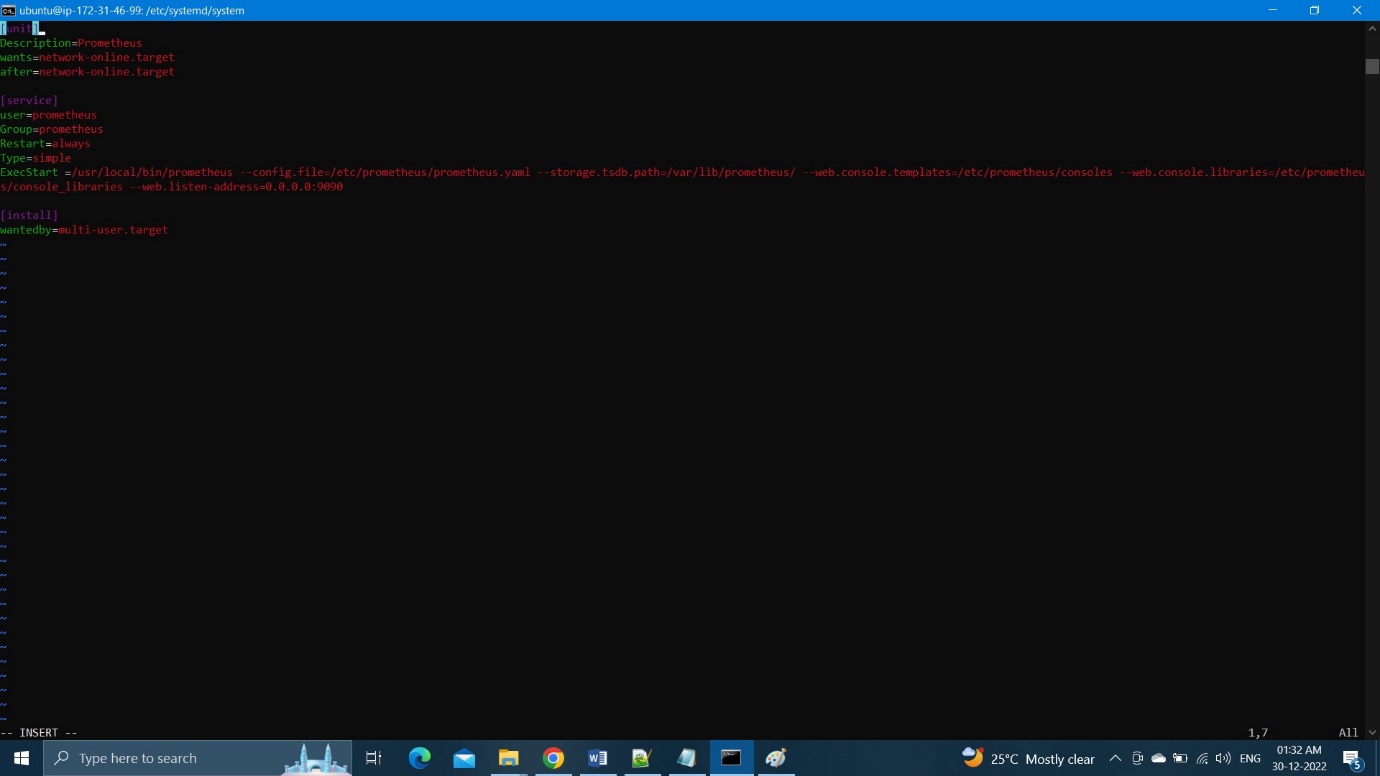
12. useradd -s /sbin/nologin --system -g prometheus Prometheus  
13.chown -R prometheus:prometheus /etc/prometheus/ /var/lib/prometheus/

14. chmod -R 775 /etc/prometheus/ /var/lib/prometheus/

15. vi /etc/systemd/system/prometheus.service

**8a.working of Prometheus**

**8b.Prometheus config file**

  
  
1. Sudo systemctl start Prometheus

2. sudo systemctl enable prometheus

Copy the ip address and paste it on internet explorer set IP end with 9090.

**CONCLUSION:**  
Code details and required content to run the project attached to github.

Closely monitored and validated each part is working fine and ready to deliver the product to the company.  
  
**OUTPUT:**   
Product is ready to launch