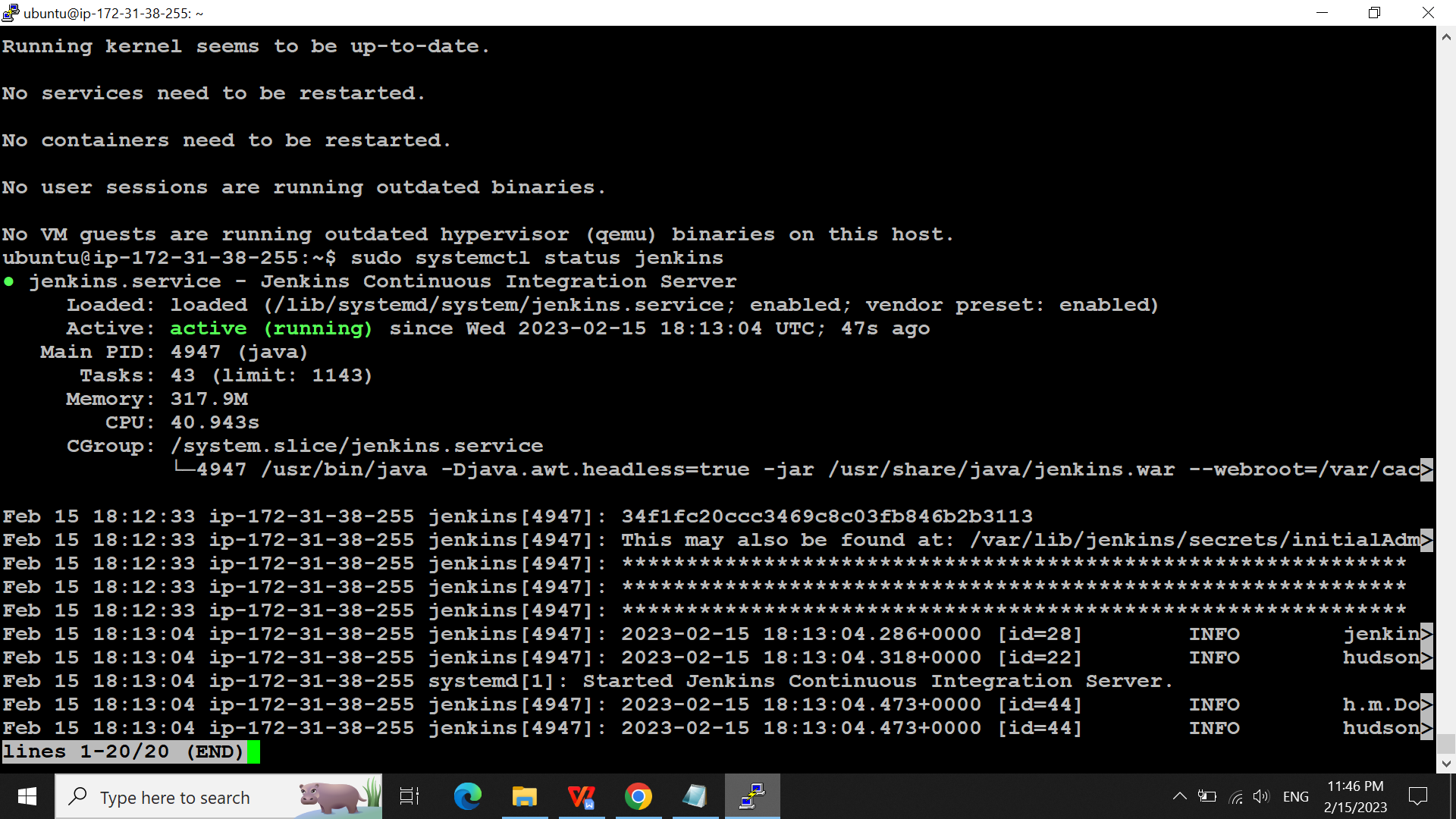
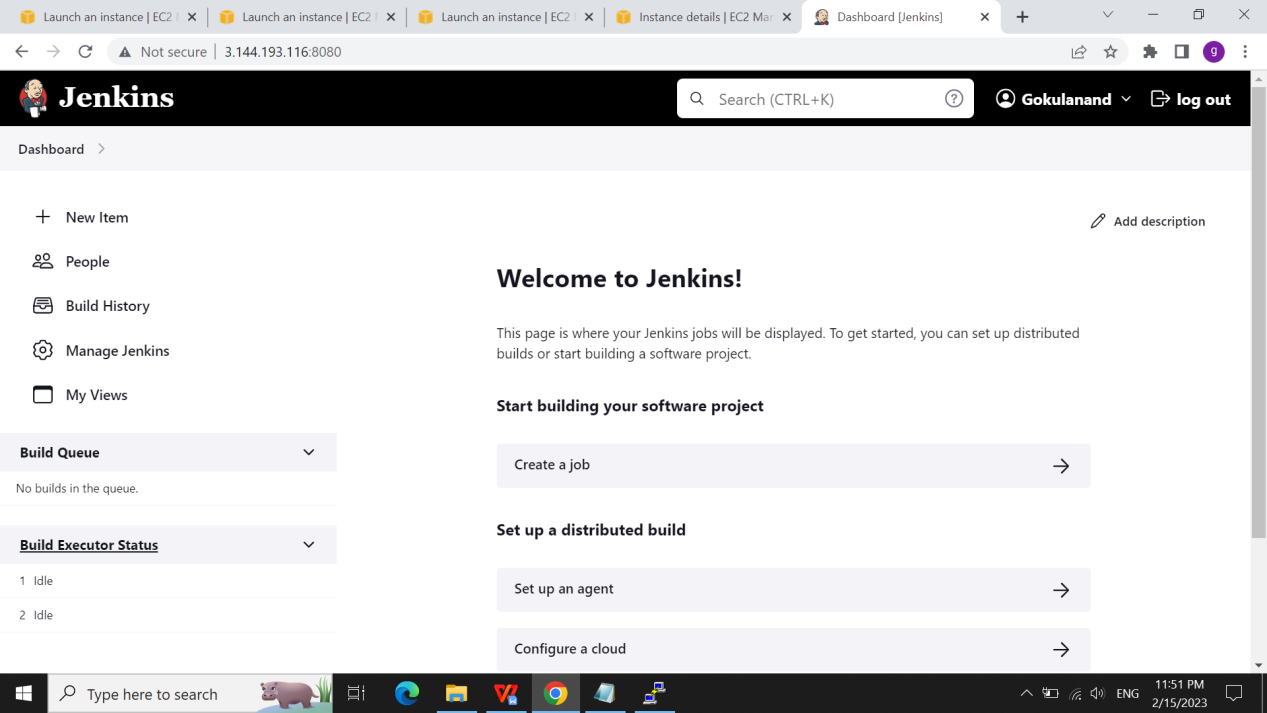


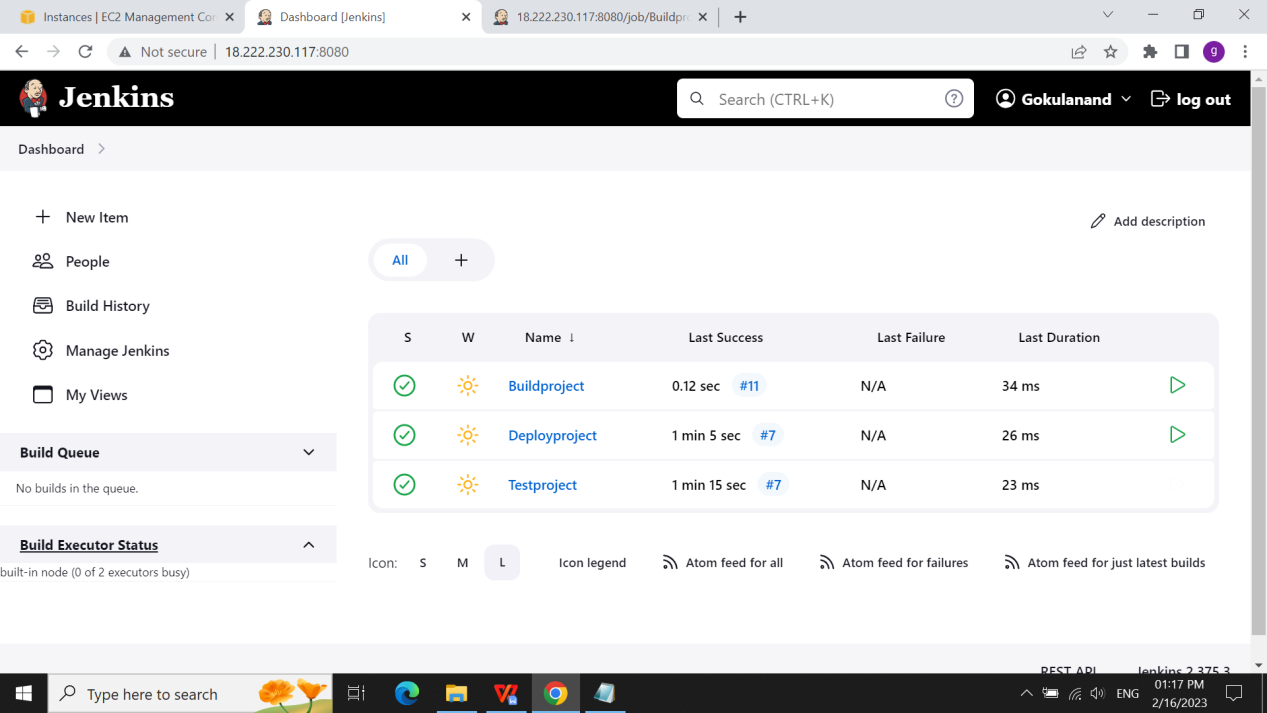
1. sudo apt-get update
2. Sudo apt-get upgrade
3. sudo apt install openjdk-11-jdk -y
4. openjdk version "11.0.17" 2022-10-18 -->java package
5. wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg
6. sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
7. sudo apt update
8. sudo apt install jenkins
9. sudo systemctl start jenkins.service
10. sudo systemctl status jenkins





Starting the job using plugins by:

1. job ordering
2. Periodically triggering
3. Remotely triggering



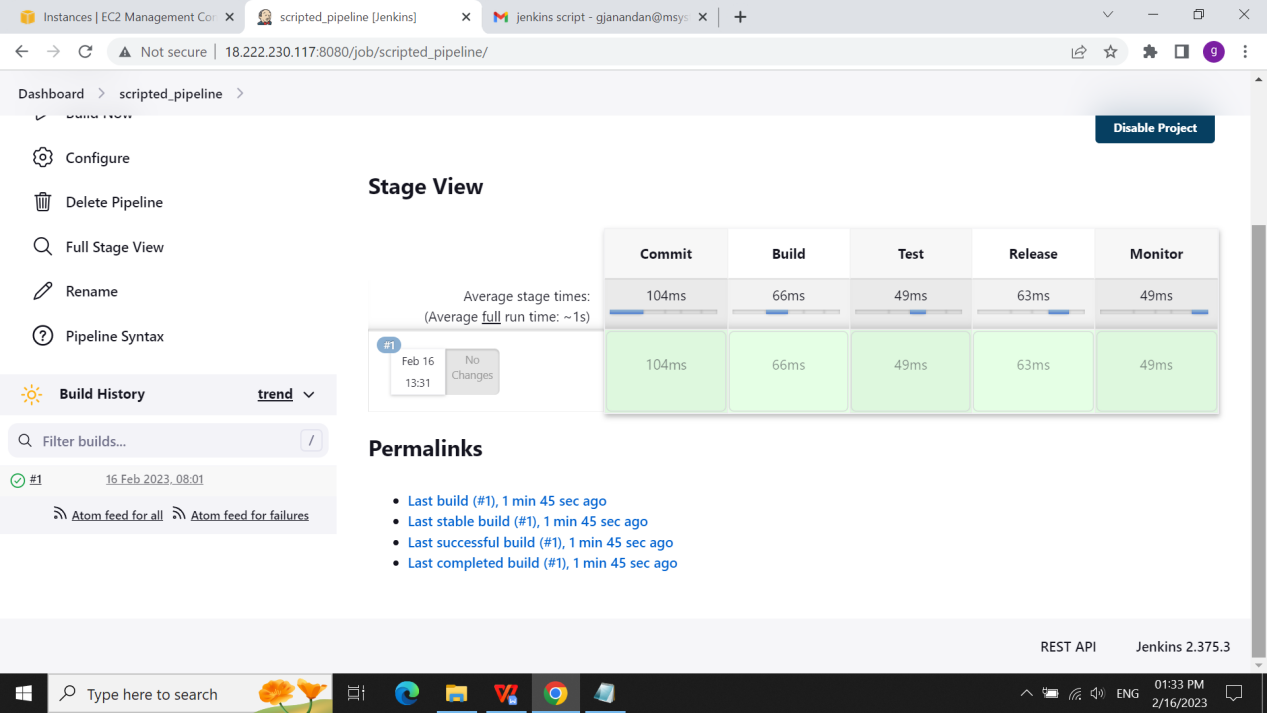
Drawback of plugins is:

Cant add the job in between

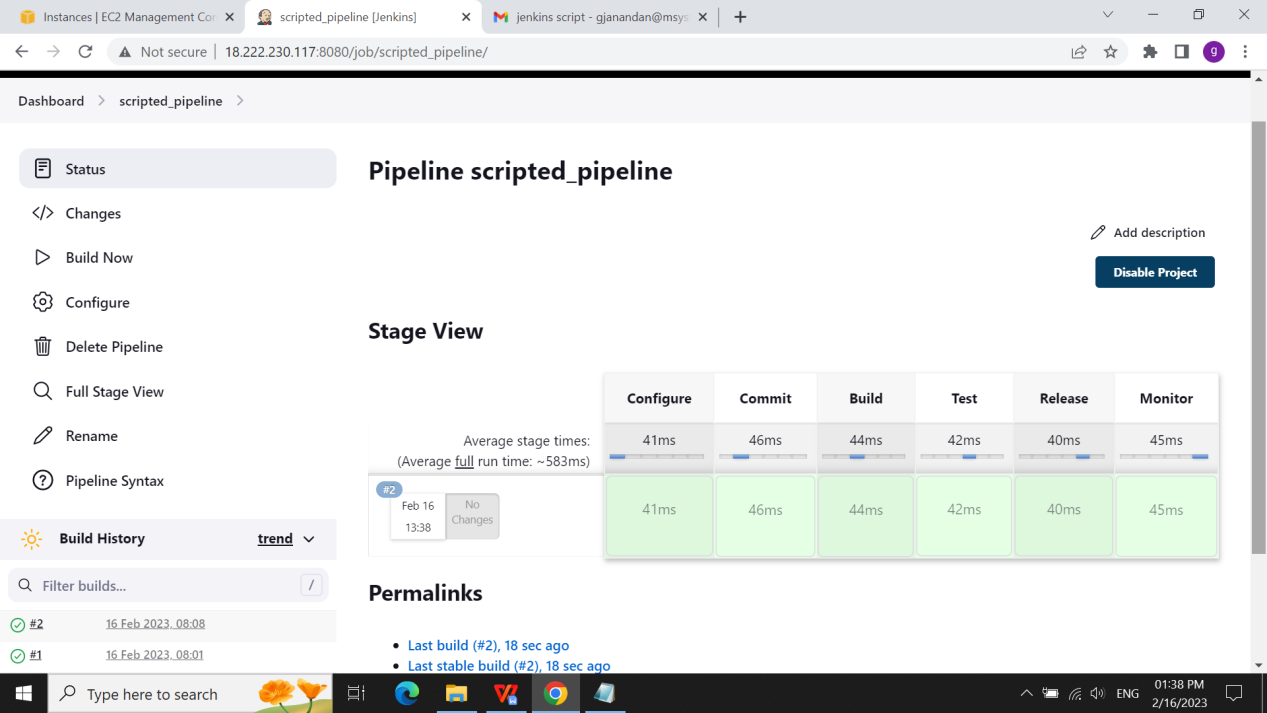
Mannual interruption is high

Jobs using Groovy script:

1. scripted pipeline (pipeline is created only in one node)
2. Declarative pipeline (making a script as a file like (template) using that we can create in n no of node)



Adding the jobs in between using pipeline:



1. **Master and Slave setup**:

\*Connecting via ssh connection for that we need ssh key gen

\*Knownhost-giving the previleage for host accesing using rsa key

\*For that key gen making pub.rsa

\*Integrating the slave via manage node in jenkins console

\*In slave:

Creating a temp job path,after running the job path in master it

Will reflect in slave.

&Master: need to install:  
\*ubuntu

Java

\*jenkins

Slave:

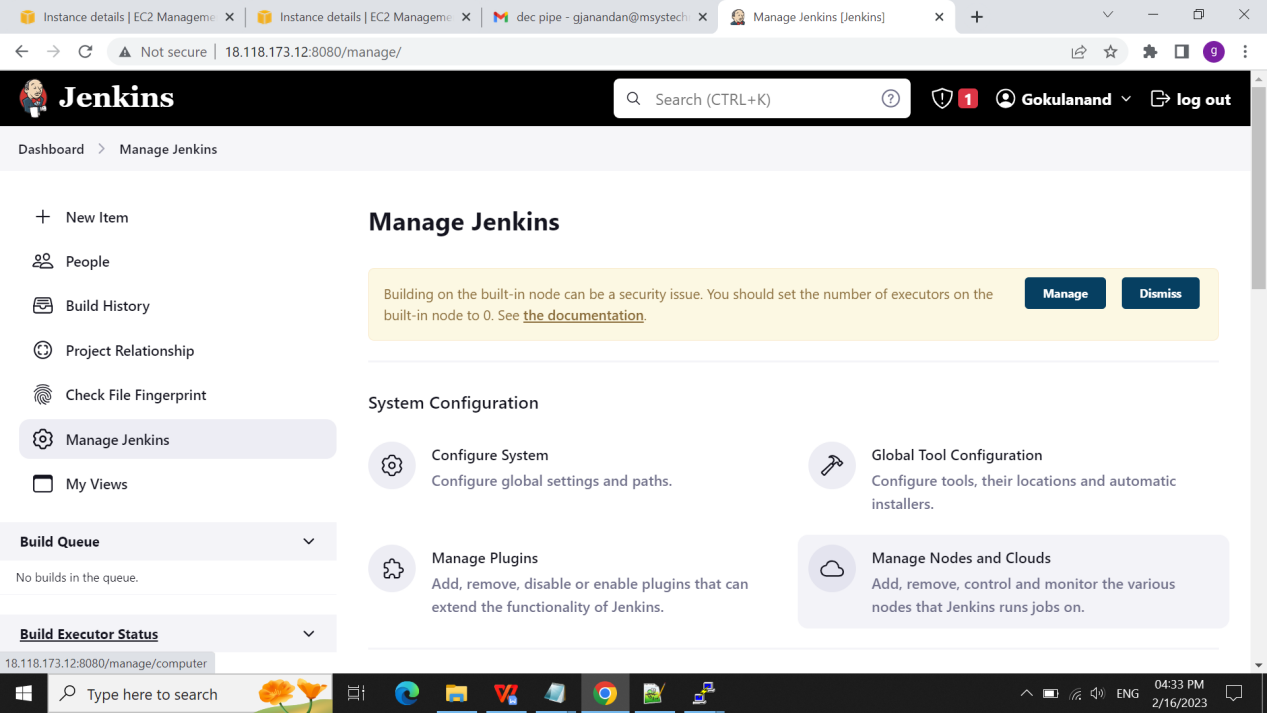
1. Ubuntu
2. Java

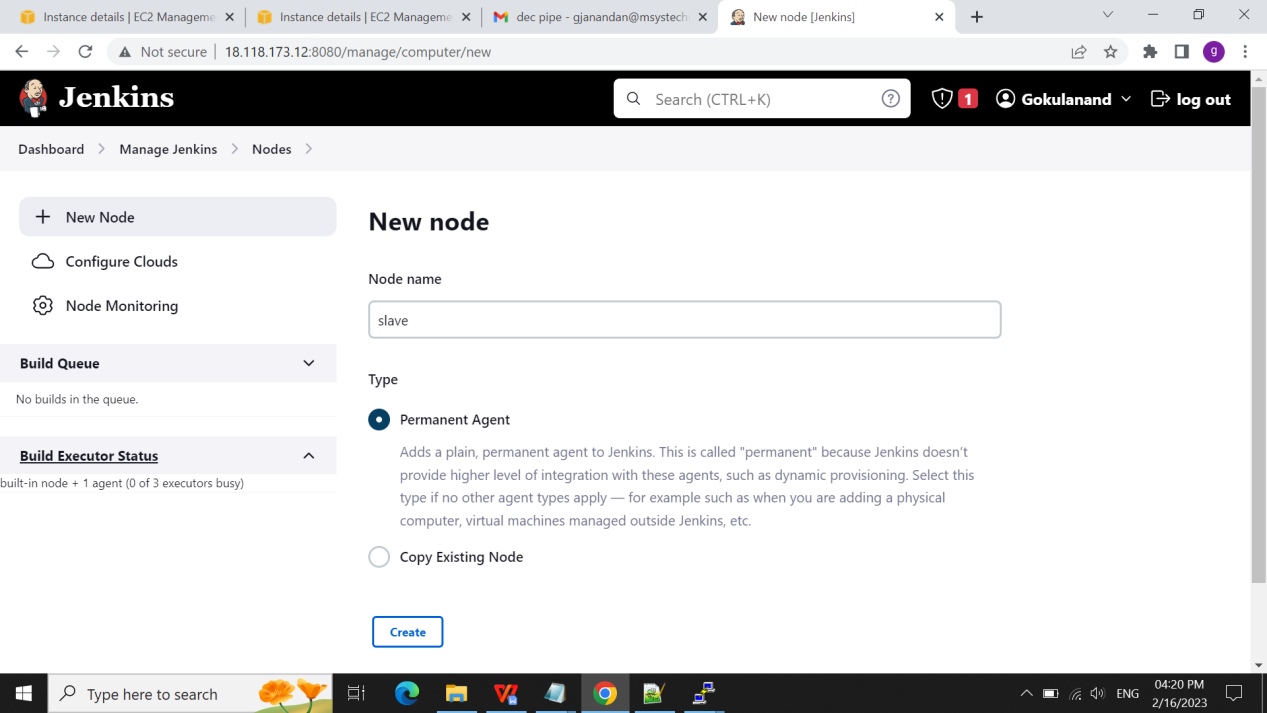
In master machine:

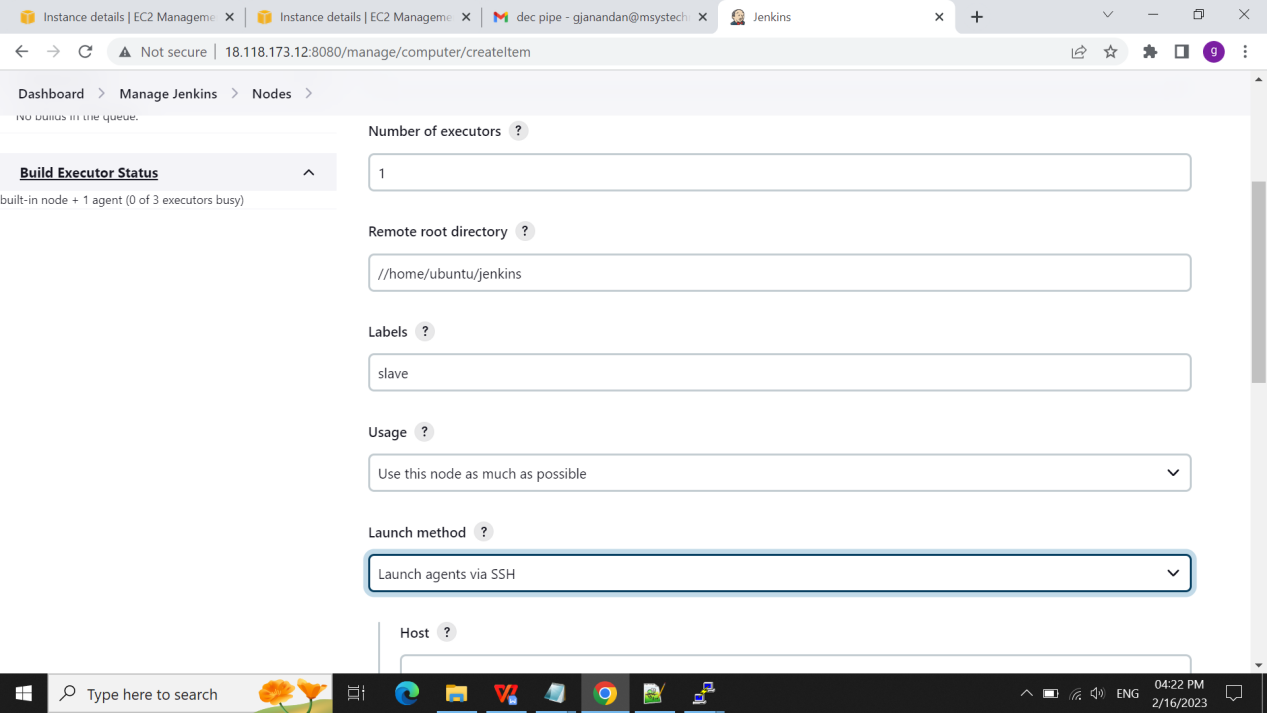
1. find a hidden file--> ls -lart
2. Cd .ssh
3. Vi authorized key --> have a pub key of the server
4. Ssh localhost-->permission will get deny due to the key
5. ls -lrt --> knownhost
6. touch id\_rsa.pub
7. touch id\_rsa
8. Cp authorized key id\_rsa.pub
9. Ls -lrt
10. Vi id\_rsa (copy the pem file)
11. Chmod 400 \*
12. Ssh localhost (now will have the permission)

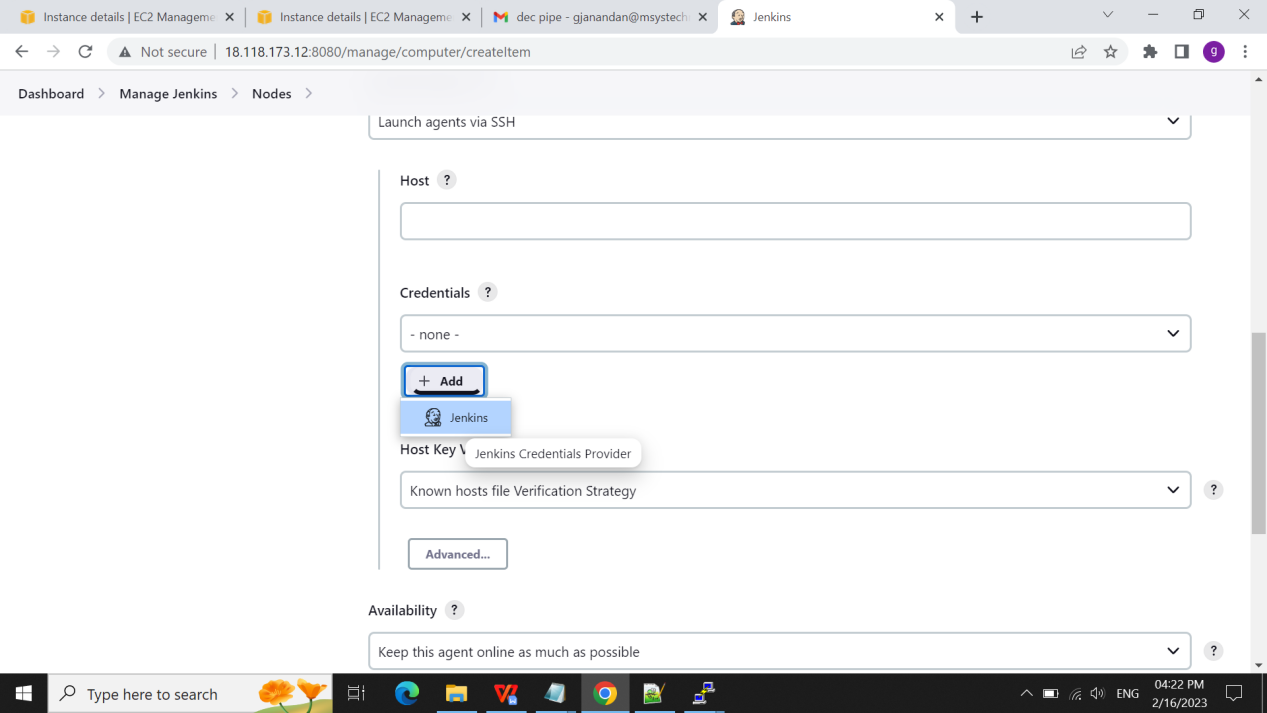
In Slave machine:

1. create temp job landing path
2. Mkdir->jenkins
3. Pwd

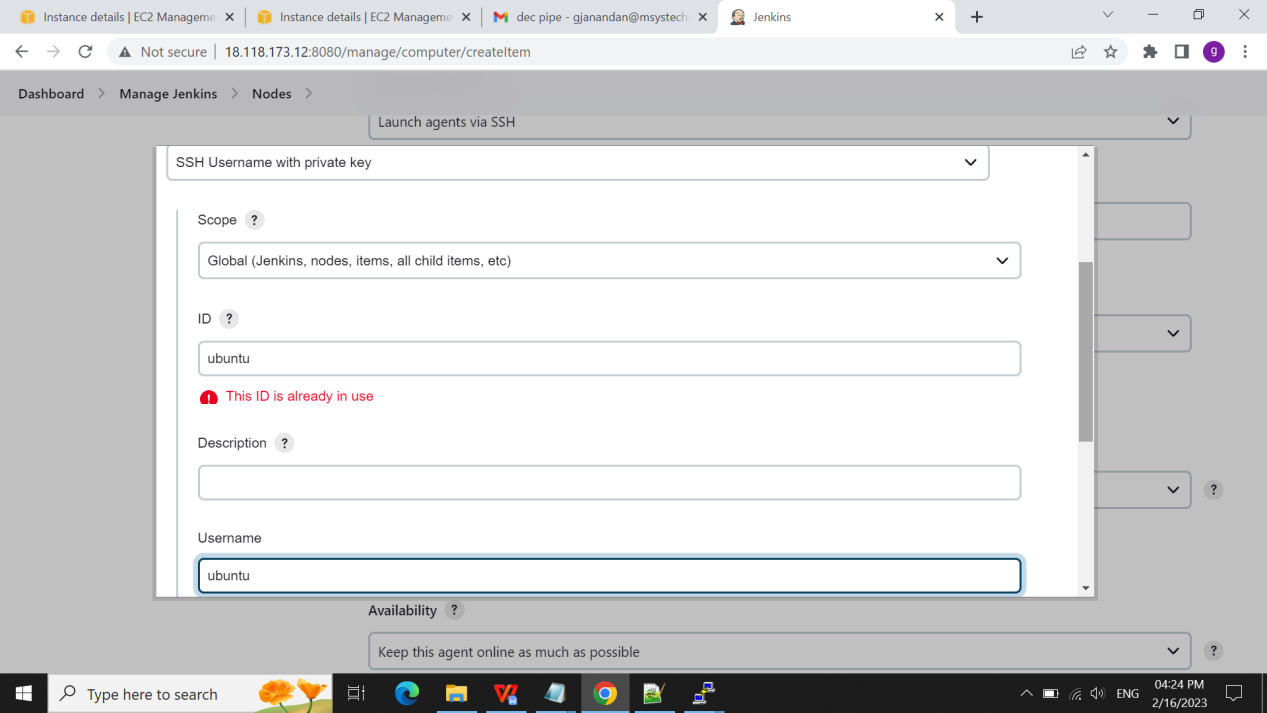


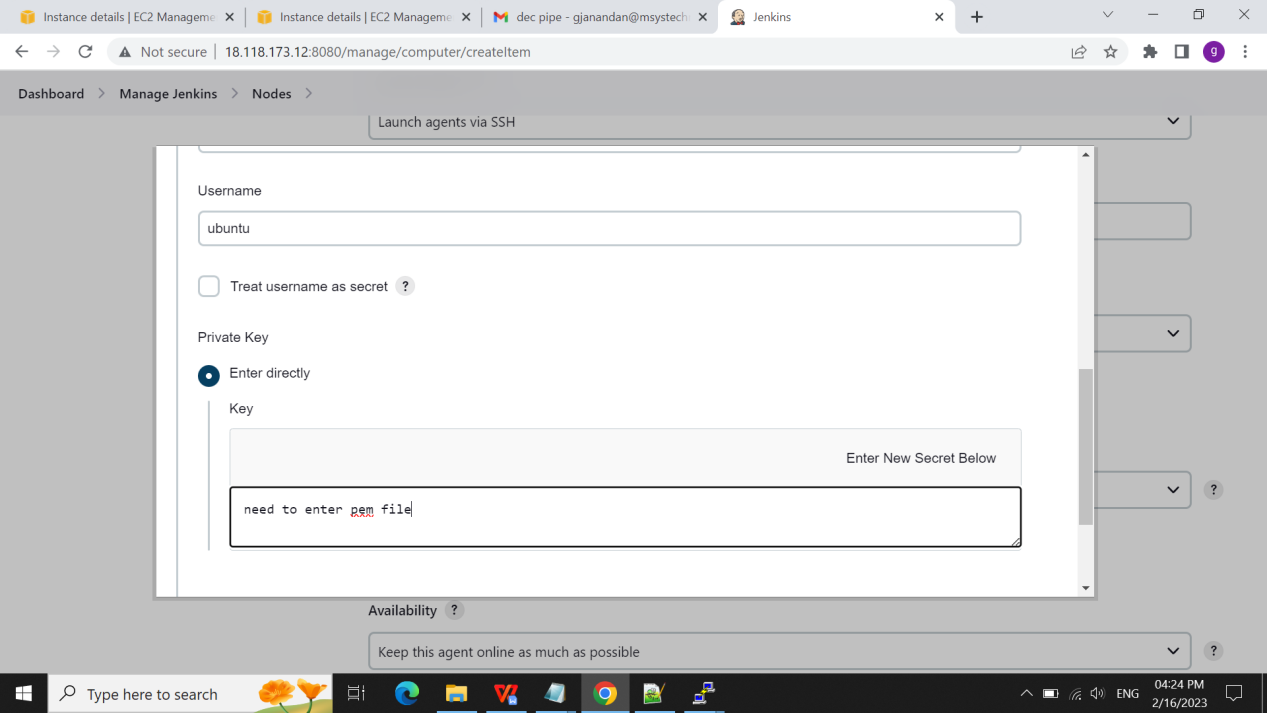


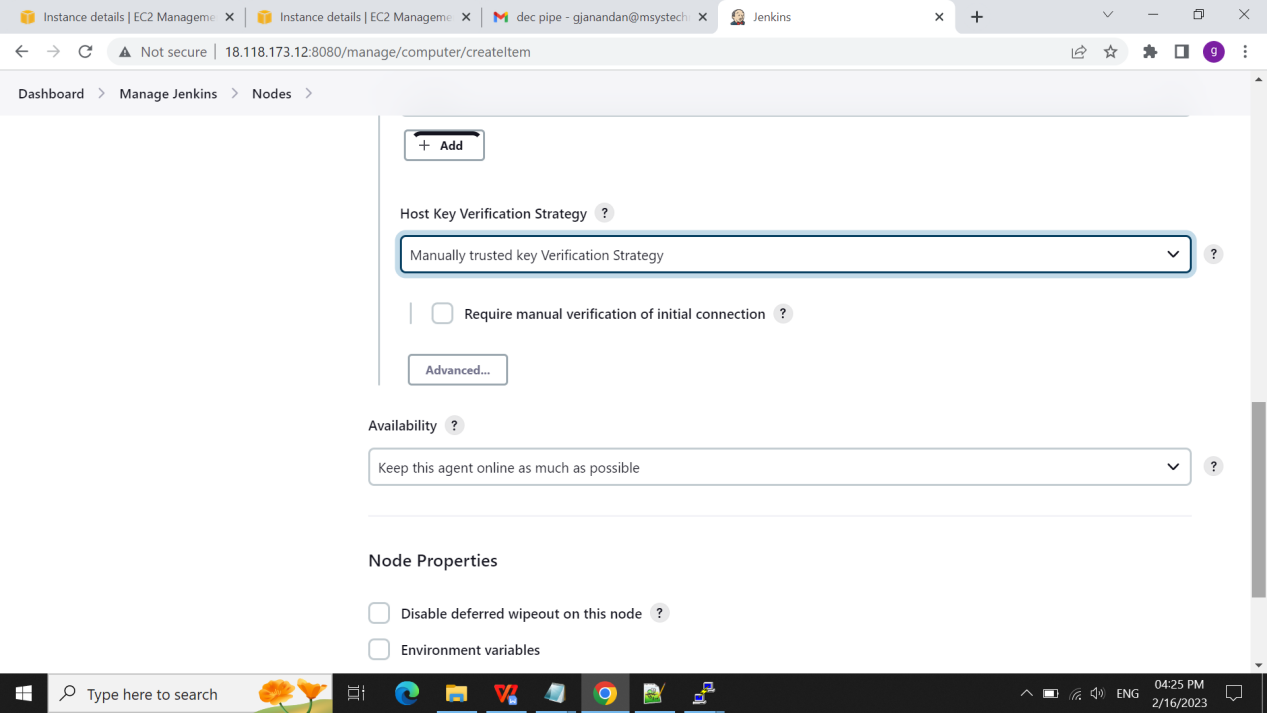




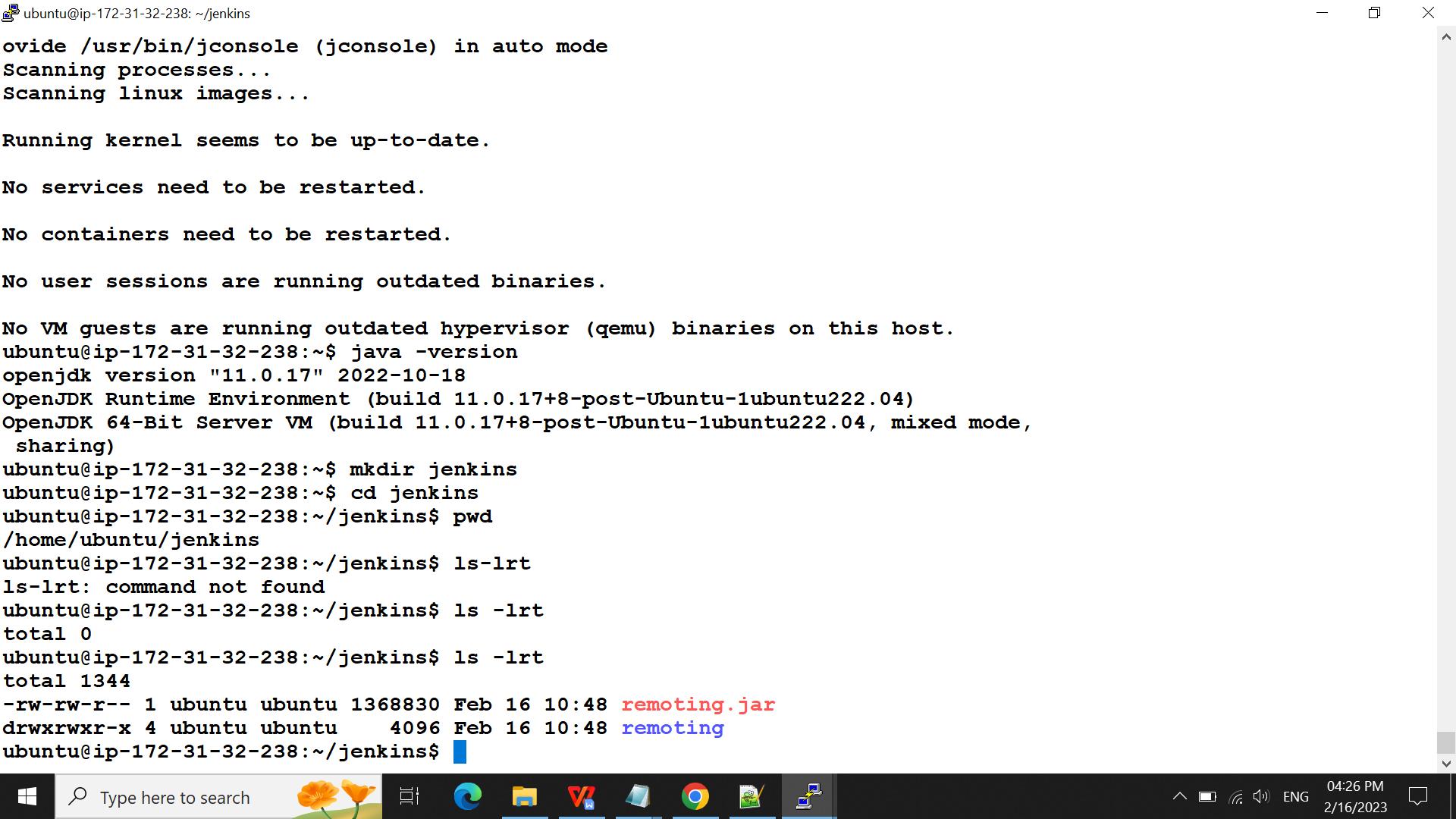
Above At host need to give slave ip.



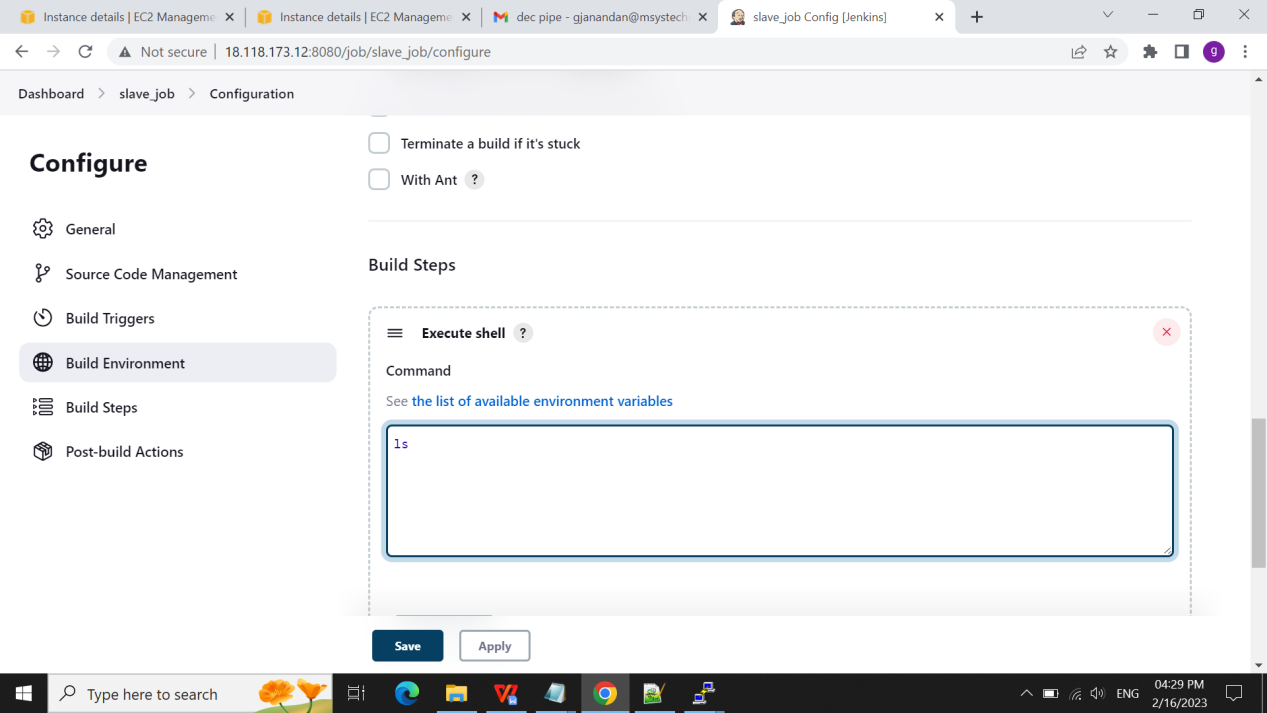


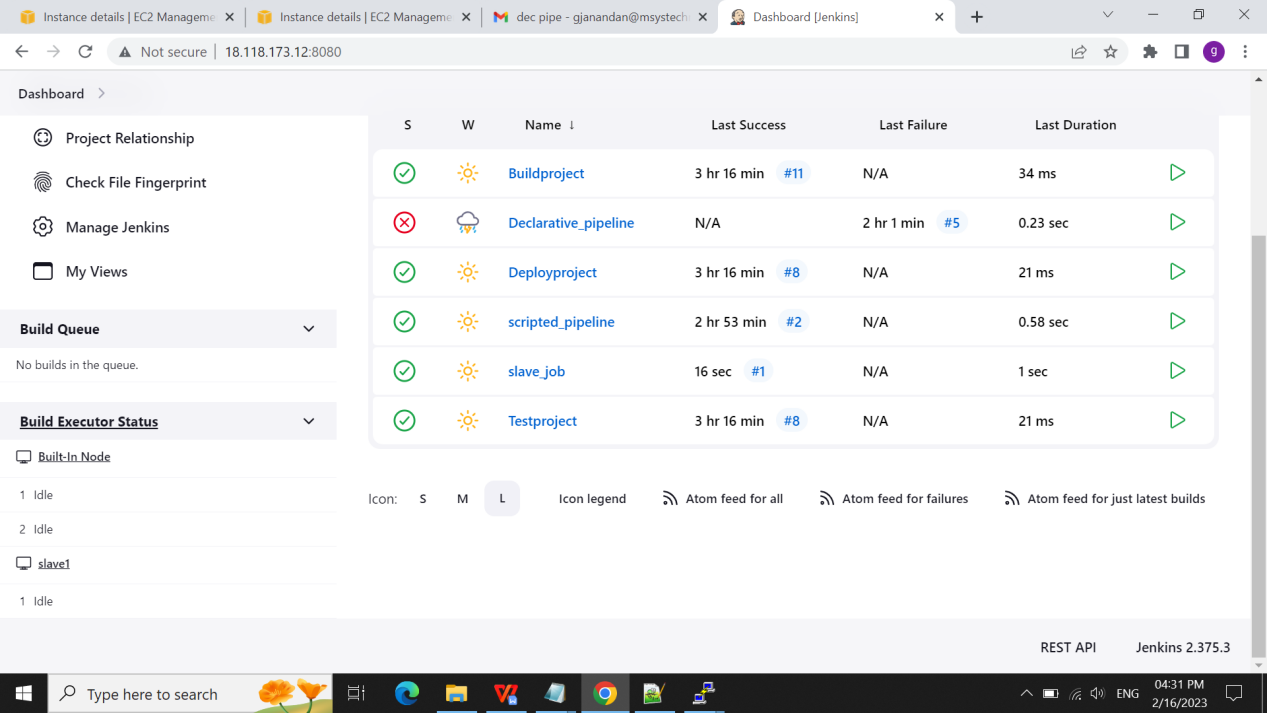


Checking in slave that remote jar is created r not

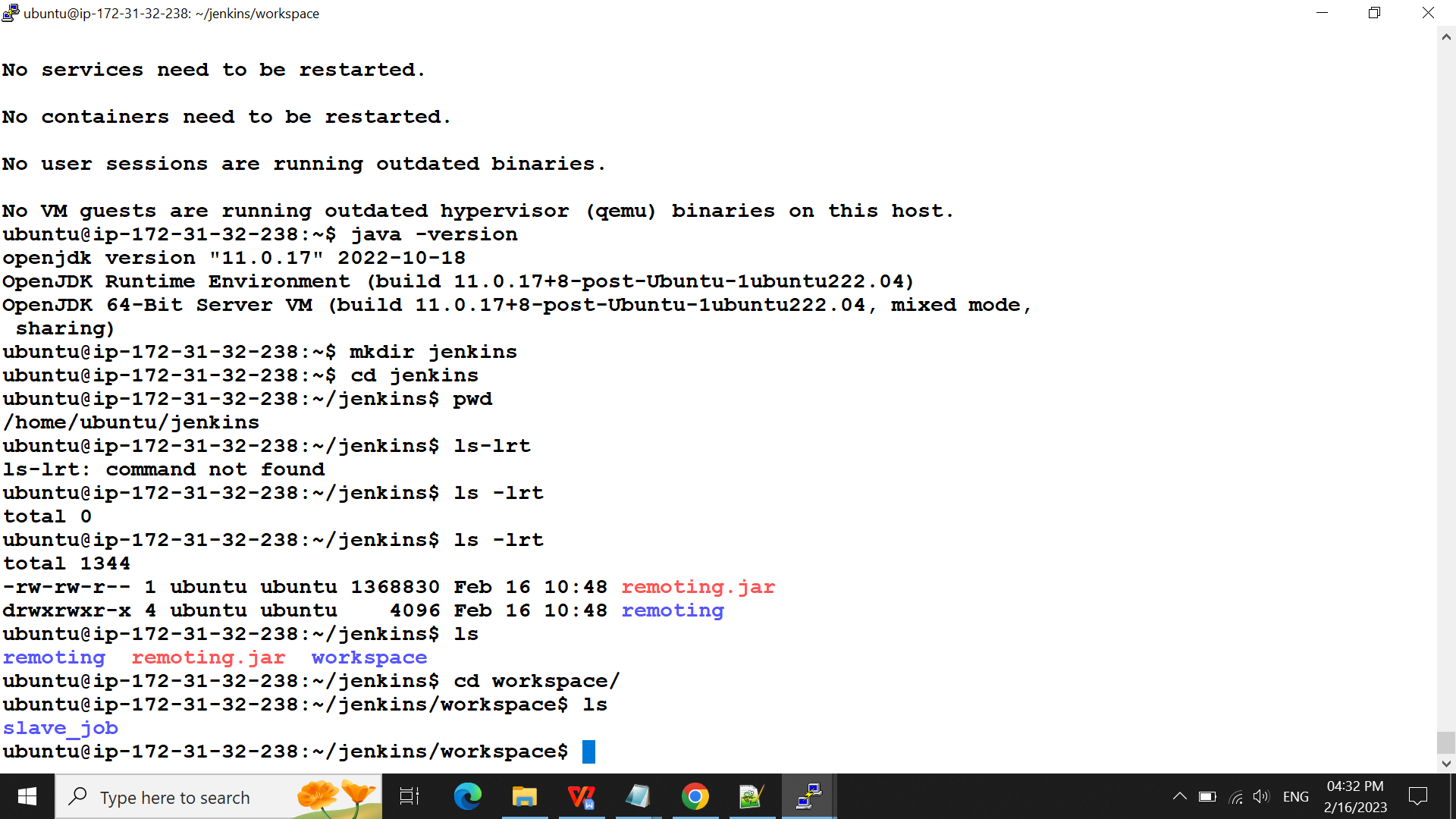


Creating a slave job in freestyle project:





Now check the slave machine wheather job is running in slave or not:



1. Jenkins Auto deployment using Tomcat:
2. installing a java,jenkins and changing the jenkins port no by cd/etc/default
3. If above is not working means do this process
4. Vi /usr/lib/systemd/system/jenkins.service(there change envprt=http=9090)and do

systemctl daemon-reload' and restart it.

Now go to home dir and install tomcat

\*wget <https://archive.apache.org/dist/tomcat/tomcat-10/v10.0.8/bin/apache-tomcat-10.0.8.tar.gz> and untar this

\* now create cd /opt

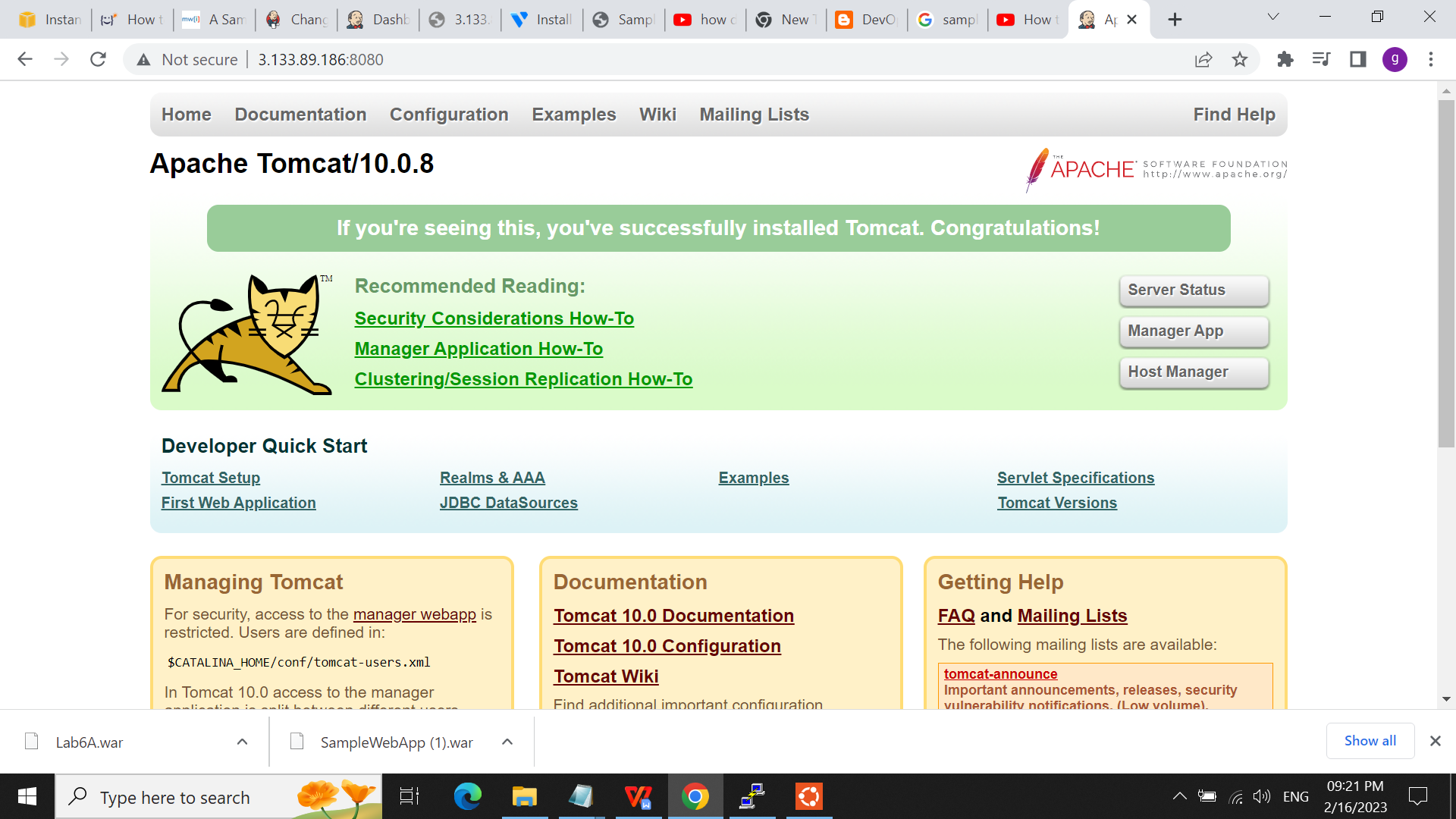
\*From opt -cp - cp /home/ubuntu/apache .

\*cd apache

\*Cd bin

\*./start.sh

\*hit in web will have tomcat



Now create a job and verfiy in cd var/lib/jenkins--> ls cd (workspace) ls -->cd -->tomcatworkpace

Pwd-->/var/lib/jenkins/workspace/tomcat pipeline

In the above path cp the having files(index.html) here

We have a war have that war file should be put into the tomcat pipeline

Cp /home/ubuntu/sample.war .

Now need to integrate jenkins and the tomcat

In jenkins go to -->mange jenkins-->instal plugin-->dedploy to container--> go to the job config-->post buid action-->select deploy war container

1. \*\*/\*.war
2. Contextpath -->war file name
3. Add conatiner username deployer ,password deployer
4. Give tomcat url -->apply and save (successfully integrated the jenkin nd tomcat app)

Need toset cred for tomcat in /opt/tomcaat/cd bin/vi tomcat-usr.xml and copy the cred

Del all by pressing ctl u from last till top of version=’1.0>

<role rolename=”manager-gui”/>

<role rolename=”manager-script”/>

<role rolename=”manager-jmx”/>

<role rolename=”manager-status”/>

<user username=”admin” password=”admin” roles=”manager-gui,manager-script,manager-jmx,manager-status”/>

<user username=”deployer” password=”deployer” roles=”manager-script’/>

Now go to webapps in tomcat:

Cd webapps/manager/META-INF/ -->ls

Vi content.xmi--> <! -->(disable this bcz according to tomcat it will follow the cronjob)

Come back to tomcat:

Cd bin-->./shutdown.sh-->./startup.sh

NOW IF WE RUN THE JOB TOMCAT WILL AUTODEPLY THE WARFILE.