DevOps Project

Problem Statement:

Create an end-to-end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as scm, maven as the build tool, deploy in a docker instance and create a docker image, store the docker image in ECR, Kubernetes deployment using ECR image. Build sample java web app usi

Approach:

Requirements:

✓ CI/CD pipeline System
✓ Git - local version control system.
✓ GitHub - As Distributed version control system.
✓ Jenkins - Continuous Integration tool.
✓ Maven - As a Build Tool.
✓ docker -Containerization
✓ Kubernetes - As Container Management Tool

Step-1:

➢ Setup CI/CD with GitHub, Jenkins, Maven & Tomcat.
➢ Setup Jenkins
➢ Setup & Configure Maven, Git.
➢ Setup Tomcat Server.
➢ Integrating GitHub, Maven, Tomcat Server with Jenkins
➢ Create a CI and CD Job.
➢ Test the Deployment

Step-2:

➢ Setup CI/CD with GitHub, Jenkins, Maven & Docker.
➢ Setting up the docker Environment.
➢ Create an Image and Container on Docker Host.
➢ Integrate Docker Host with Jenkins.
➢ Create CI/CD Job on Jenkins to build and deploy on container.

Step-3:

➢ Build and Deploy on Container.
➢ CI/CD with GitHub, Jenkins, Maven & Kubernetes.
➢ Setting up the Kubernetes (EKS).
➢ Write pod service and deployment manifest file.
➢ CI/CD Job to build code on Jenkins & Deploy it on Kubernetes.

Step-4:

➢ Deploy artifacts on the Kubernetes

➢ Write codes in the artifacts of docker and Kubernetes which we want to run.

➢ Now build the code in Jenkins.

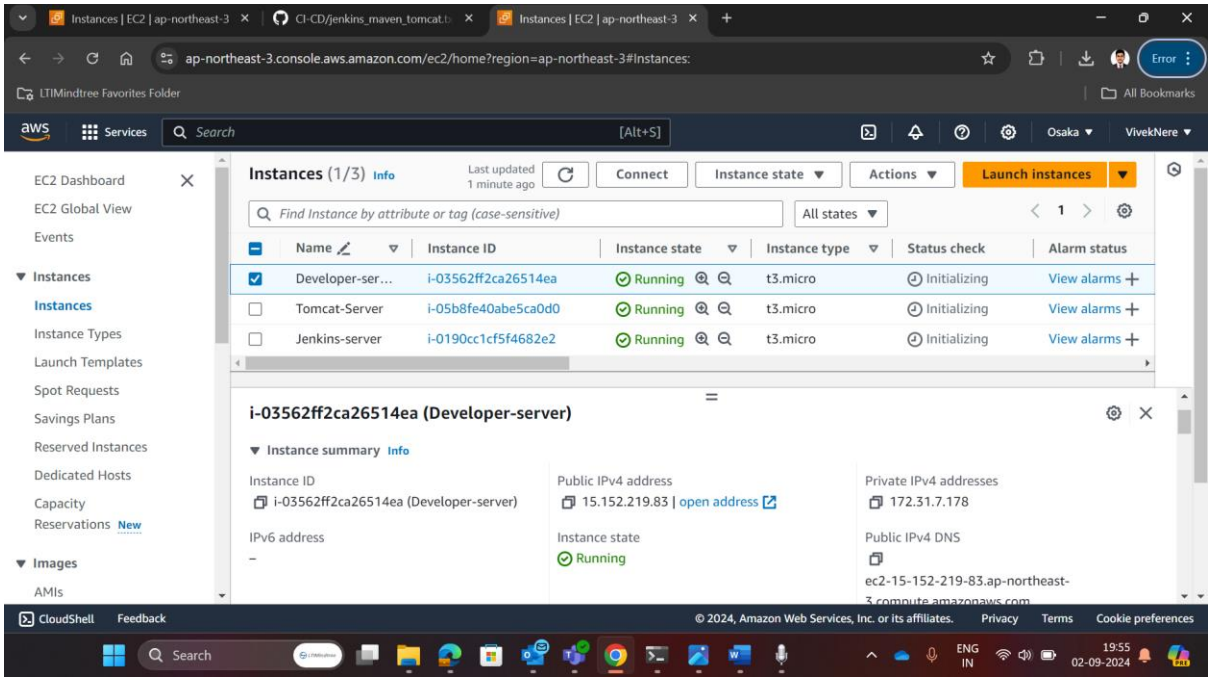
➢ Check in Kubernetes the pods are getting created or not.

➢ Now copy the service IP and paste it in the browser and check the output.

Solution:

#1: Git-Client:

Created Instances on AWS name as developer-server, Jenkins server and Tomcat server with same key-pair and same security group.



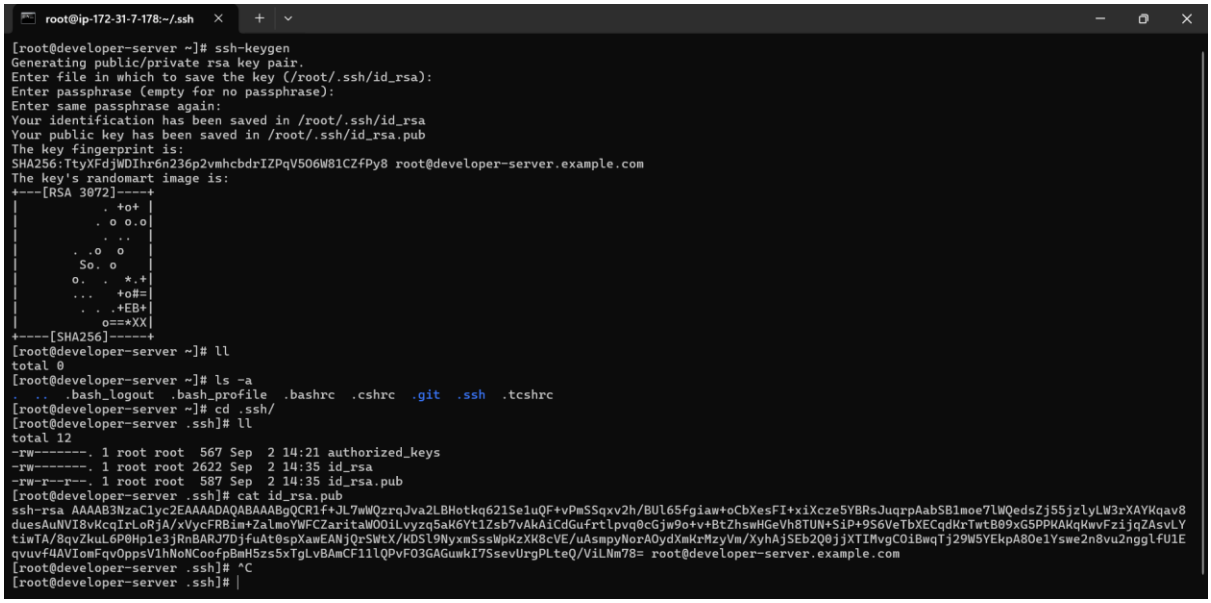
Connected developer server instance with local system through ssh and gave it a localhost name as developer-server

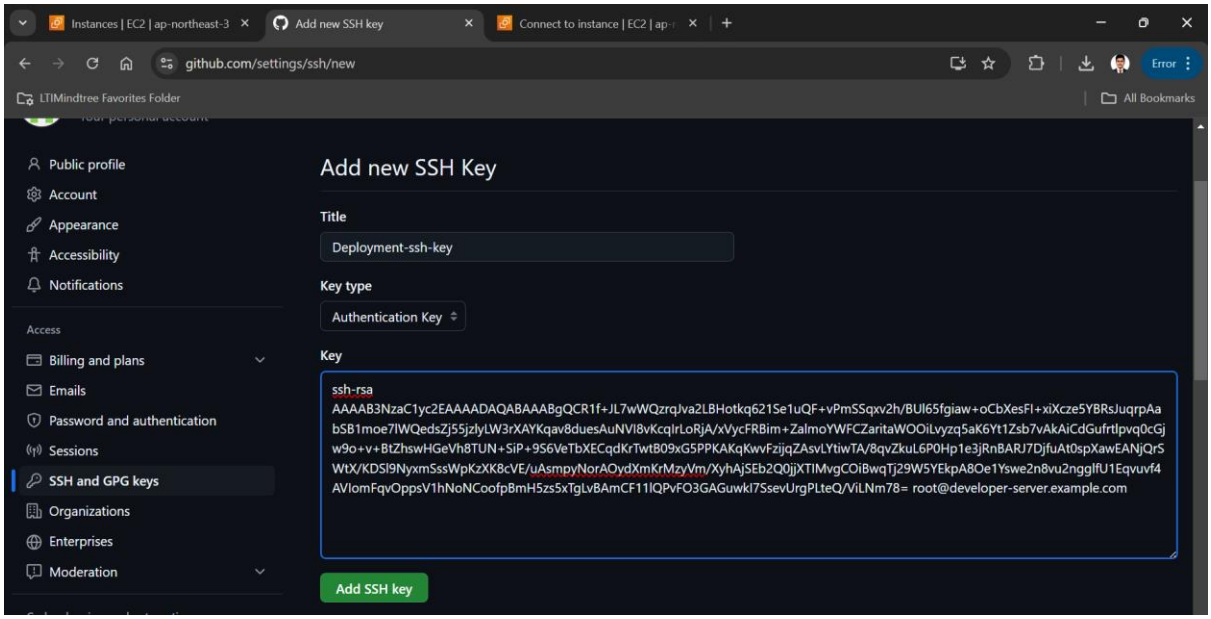
```
root@ip-172-31-7-178:~
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\10748090>cd Downloads

C:\Users\10748090\Downloads>ssh -i "deployment-key-pair.pem" ec2-user@ec2-15-152-219-83.ap-northeast-3.compute.amazonaws.com
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Mon Sep  2 14:28:15 2024 from 136.226.233.19
[ec2-user@ip-172-31-7-178 ~]$ sudo su -
[root@ip-172-31-7-178 ~]# hostnamectl set-hostname developer-server.example.com
[root@ip-172-31-7-178 ~]# bash
[root@developer-server ~]#
```

Installed and initialized git in developer-server

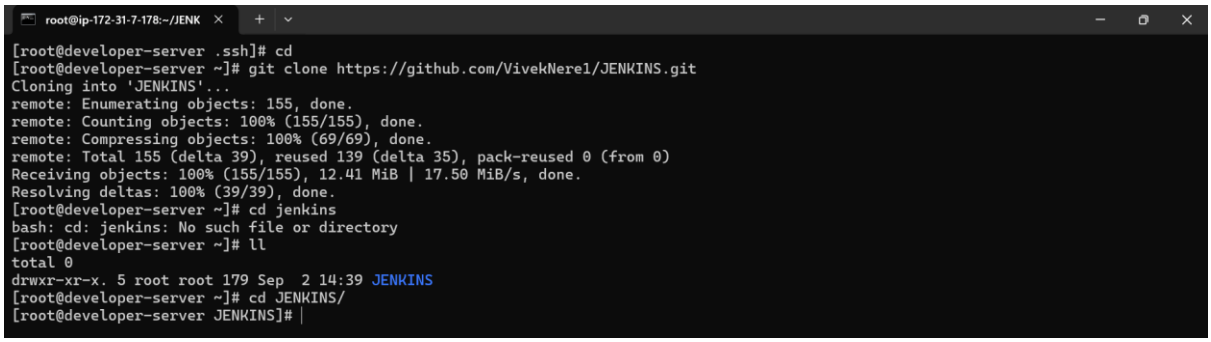
```
root@ip-172-31-7-178:~
[root@developer-server ~]# yum install git -y
Last metadata expiration check: 0:11:43 ago on Mon Sep  2 14:21:36 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture          Version                  Repository                  Size
================================================================================
Installing:
 git                     x86_64                2.40.1-1.amzn2023.0.3    amazonlinux                 54 k
Installing dependencies:
 git-core                x86_64                2.40.1-1.amzn2023.0.3    amazonlinux                4.3 M
 git-core-doc            noarch                2.40.1-1.amzn2023.0.3    amazonlinux                2.6 M
 perl-Error              noarch                1:0.17029-5.amzn2023.0.2 amazonlinux                 41 k
 perl-File-Find          noarch                1.37-477.amzn2023.0.6    amazonlinux                 26 k
 perl-Git                noarch                2.40.1-1.amzn2023.0.3    amazonlinux                 42 k
 perl-TermReadKey        x86_64                2.38-9.amzn2023.0.2      amazonlinux                 36 k
 perl-lib                x86_64                0.65-477.amzn2023.0.6    amazonlinux                 15 k

Transaction Summary
================================================================================
Install  8 Packages

Total download size: 7.1 M
Installed size: 34 M
Downloading Packages:
(1/8): git-2.40.1-1.amzn2023.0.3.x86_64.rpm                646 kB/s |  54 kB     00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm        1.7 MB/s |  41 kB     00:00
(3/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm     1.6 MB/s |  26 kB     00:00
(4/8): git-core-doc-2.40.1-1.amzn2023.0.3.noarch.rpm        16 MB/s | 2.6 MB     00:00
(5/8): perl-Git-2.40.1-1.amzn2023.0.3.noarch.rpm           1.0 MB/s |  42 kB     00:00
(6/8): git-core-2.40.1-1.amzn2023.0.3.x86_64.rpm            19 MB/s | 4.3 MB     00:00
(7/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm     567 kB/s |  36 kB     00:00
(8/8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm           244 kB/s |  15 kB     00:00
--------------------------------------------------------------------------------

Complete!
[root@developer-server ~]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/.git/
```

Created and added public ssh-key to github-account for ssh-authentication

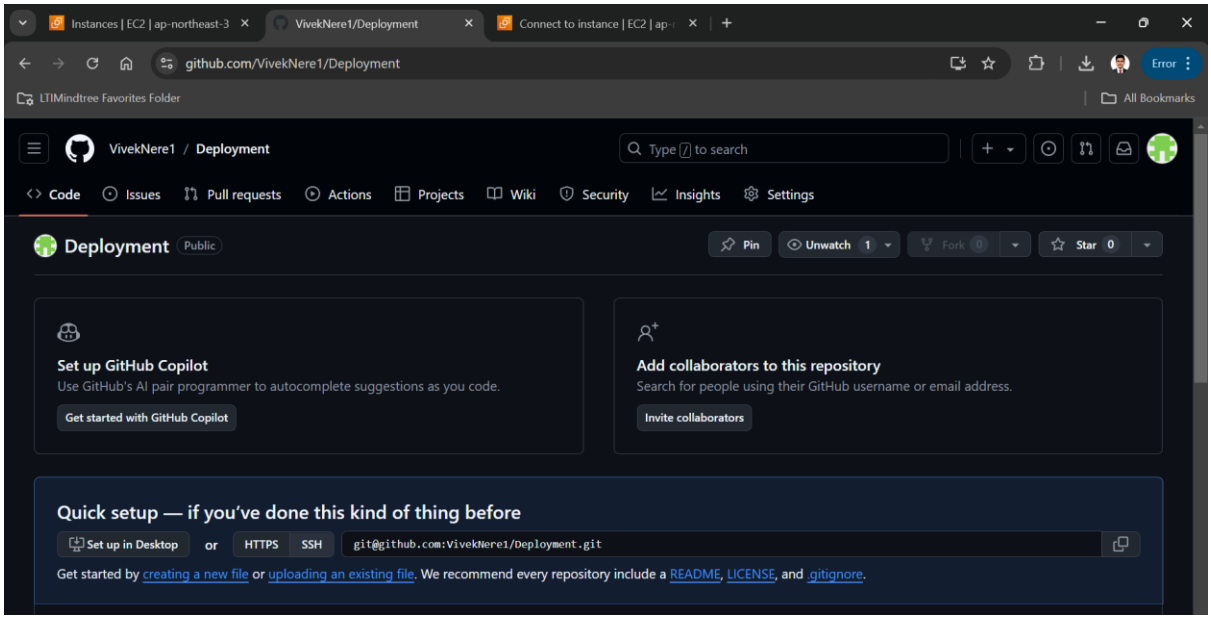




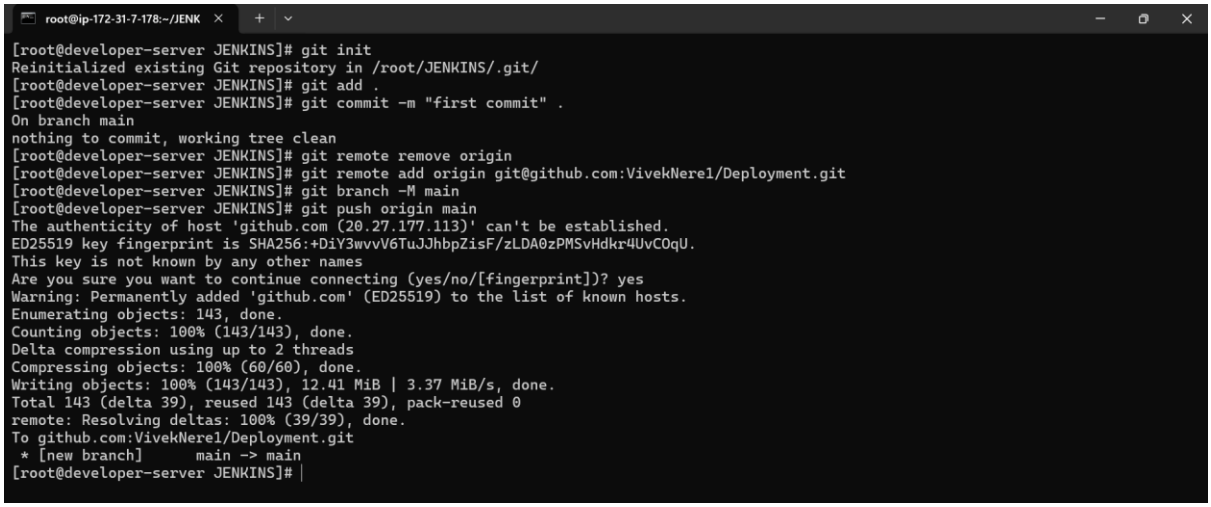
CLONE JAVA PROJECT FROM GIT REPOSITORTY WHERE IT IS PRESENT

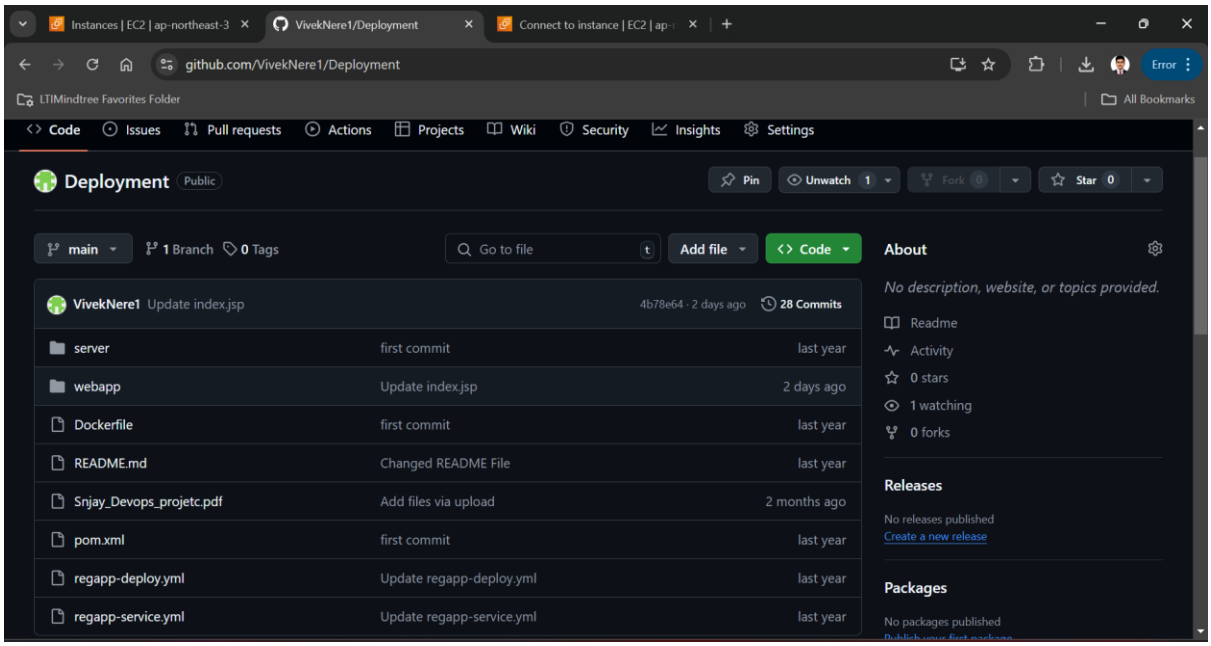


Created deployment name repository in my account to push the code in that repository which we have cloned just now

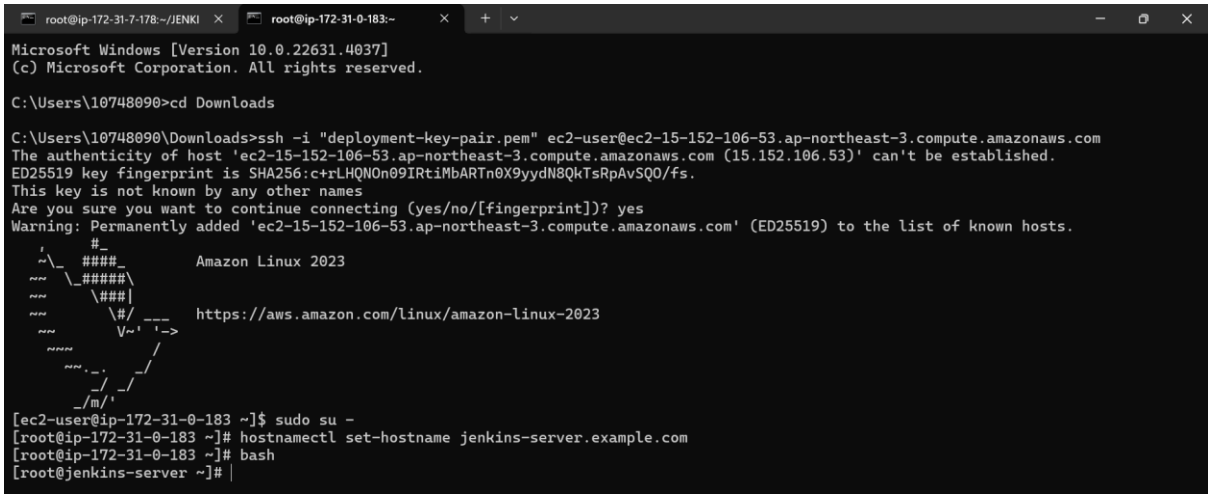


Now we have successfully push the code on my own branch

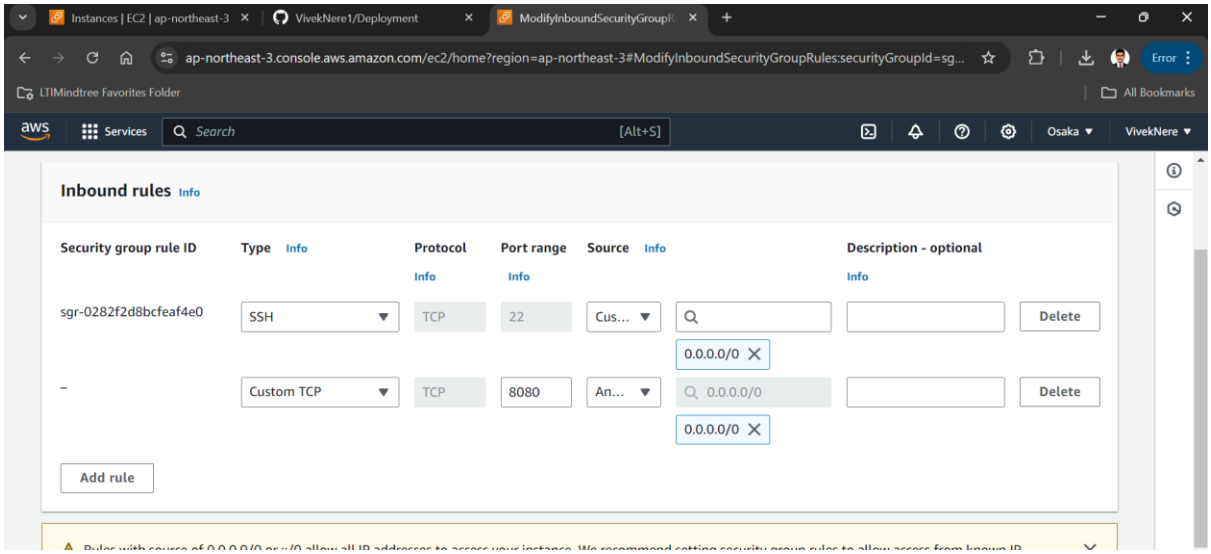




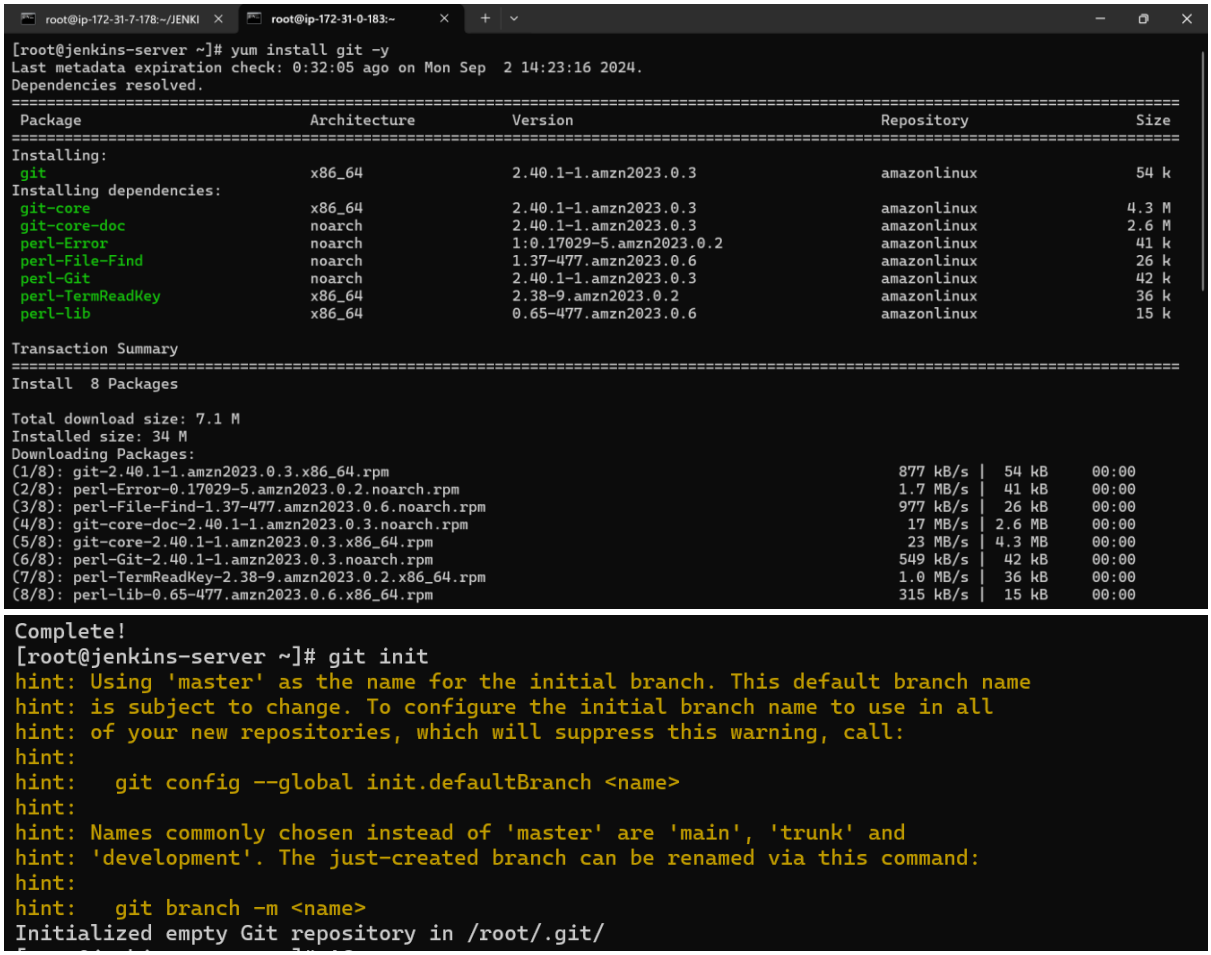
Now we have to open Jenkins-server instance in our machine using ssh and I have gave the hostname as Jenkins-server



Now we have to add one extra inbound rule in security group for Jenkins server instance with port number 8080



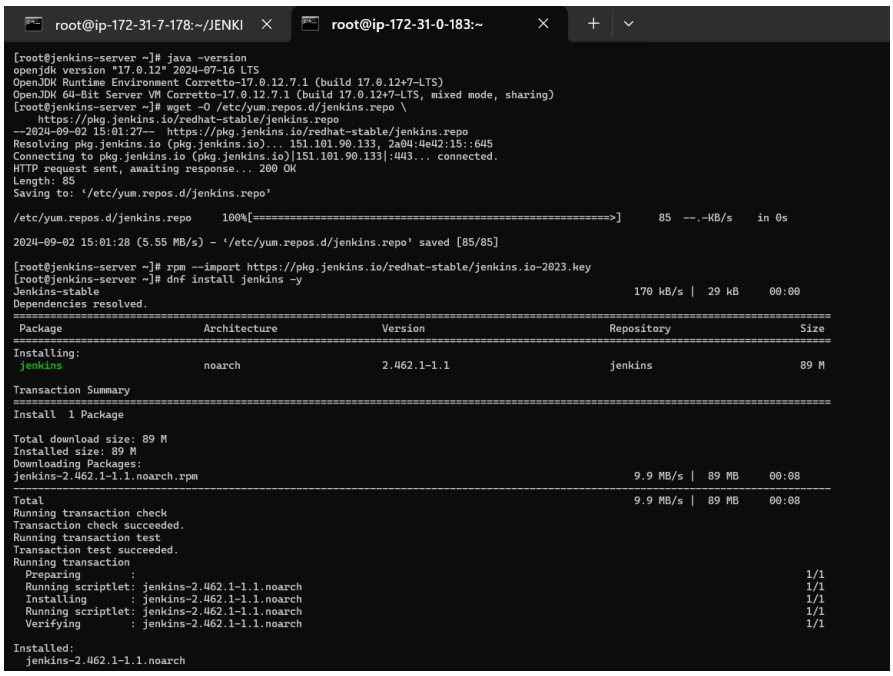
Now we have to install and initialized git in Jenkins-server




```
Complete!
[root@jenkins-server ~]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/.git/
```

#2: Jenkin & Maven :

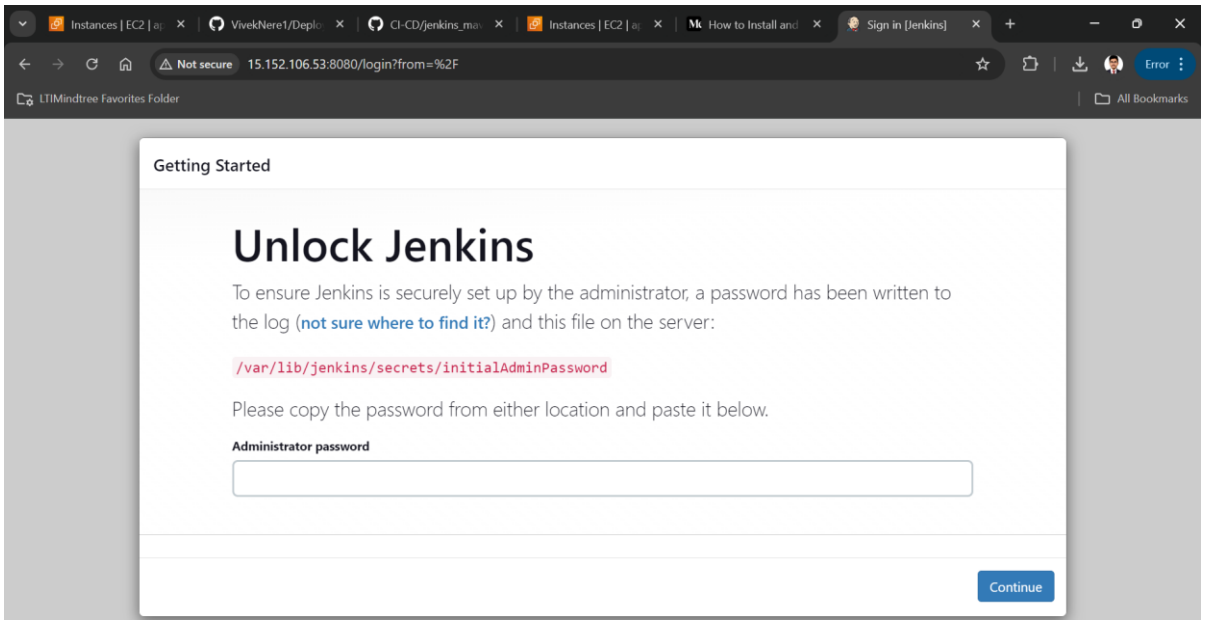
Now we have to install and configure java and Jenkins on Jenkins-server

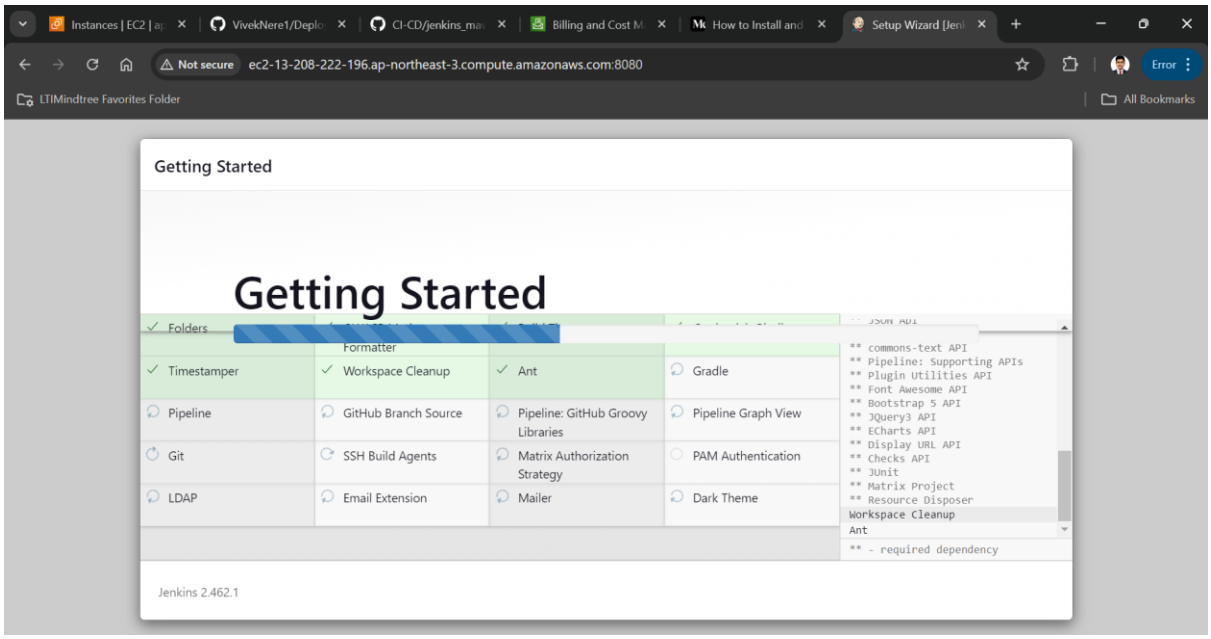


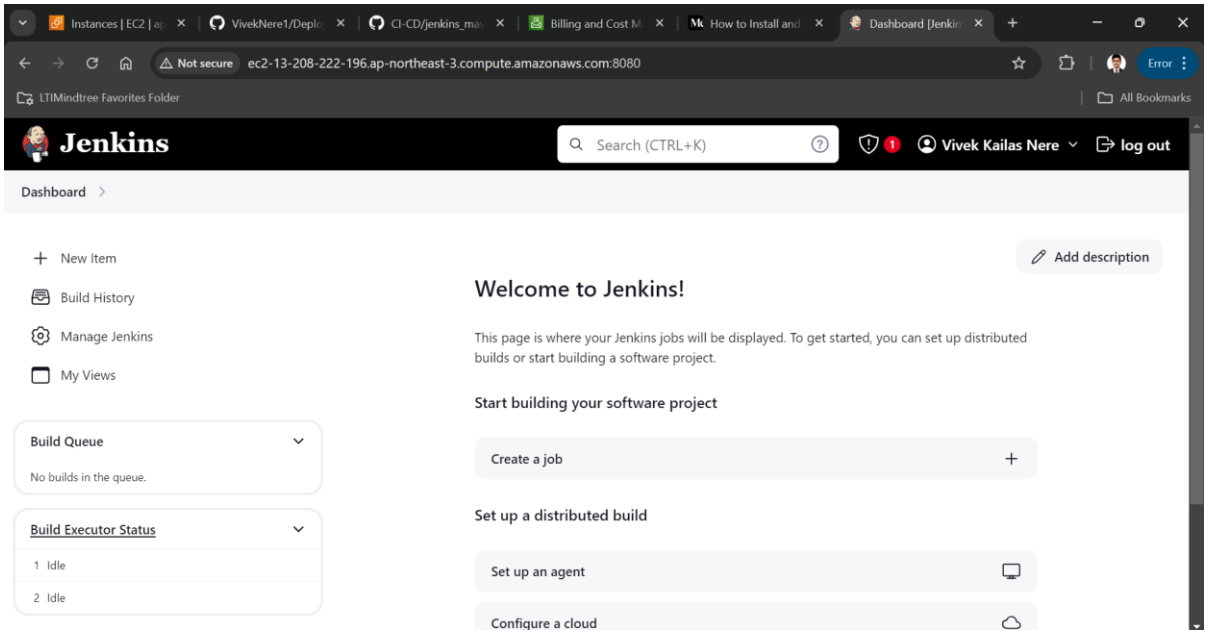
Enable and started Jenkins

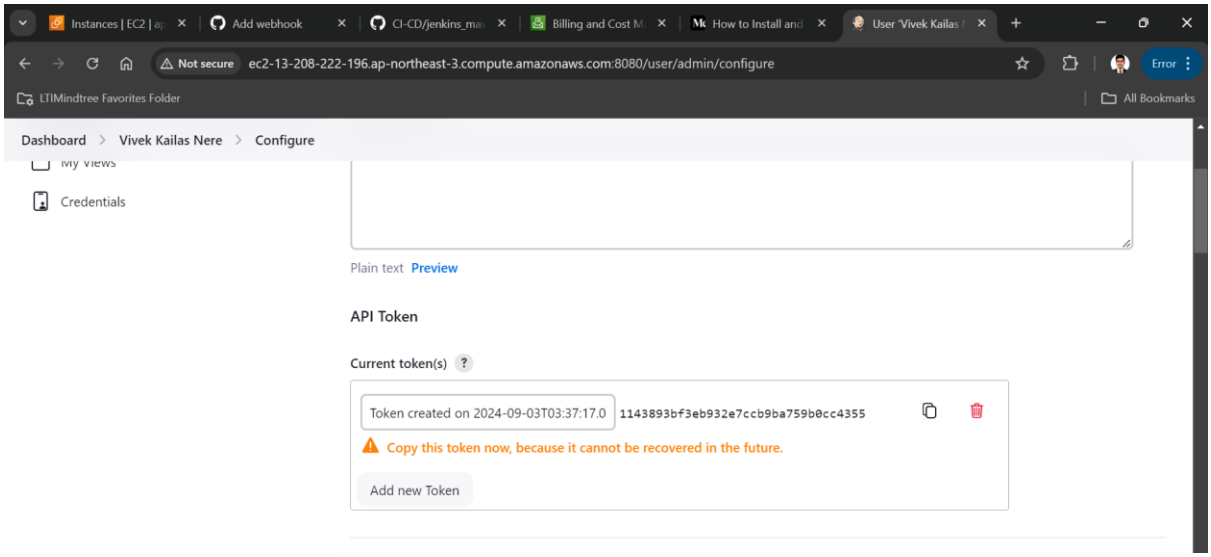
```
[root@jenkins-server ~]# sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins-server ~]# sudo systemctl start jenkins
[root@jenkins-server ~]# ^C
[root@jenkins-server ~]#
```

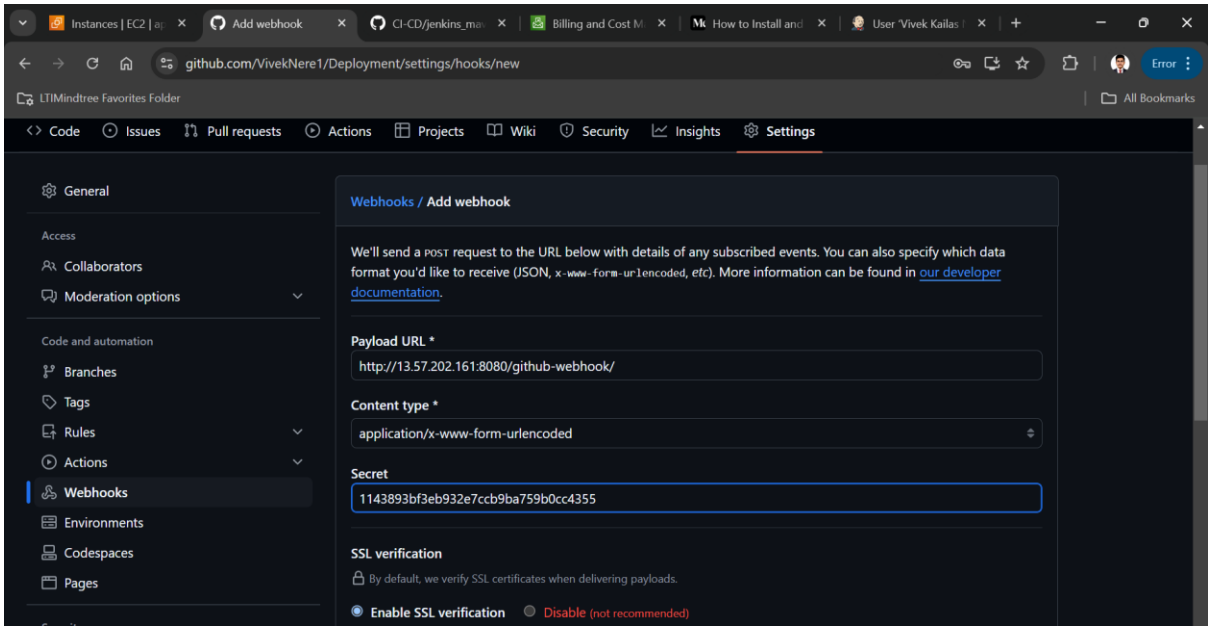
Accessed Jenkins dashboard via, Jenkins-server machine's public ip address and 8080 port, and completed login process.

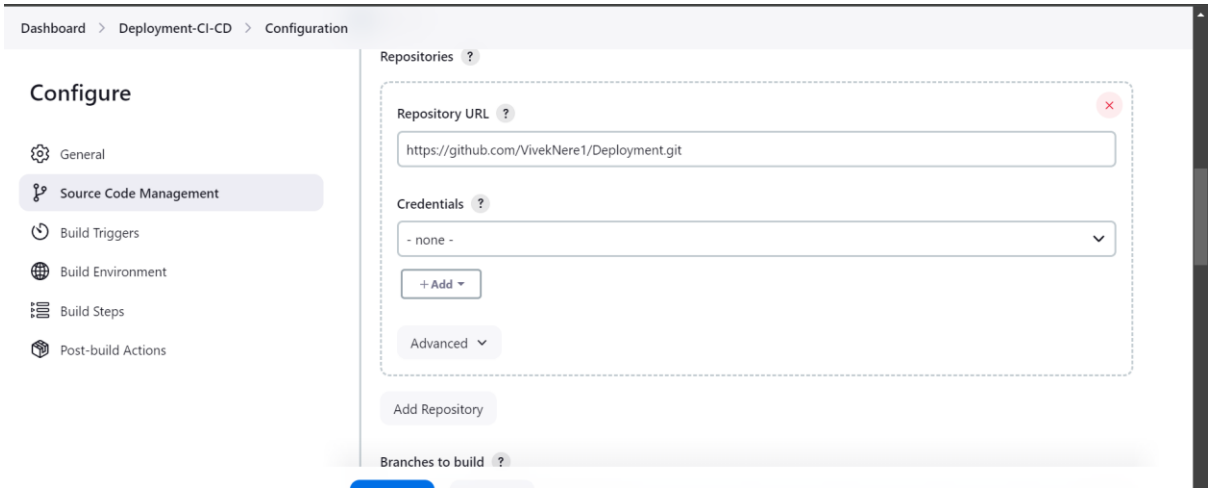


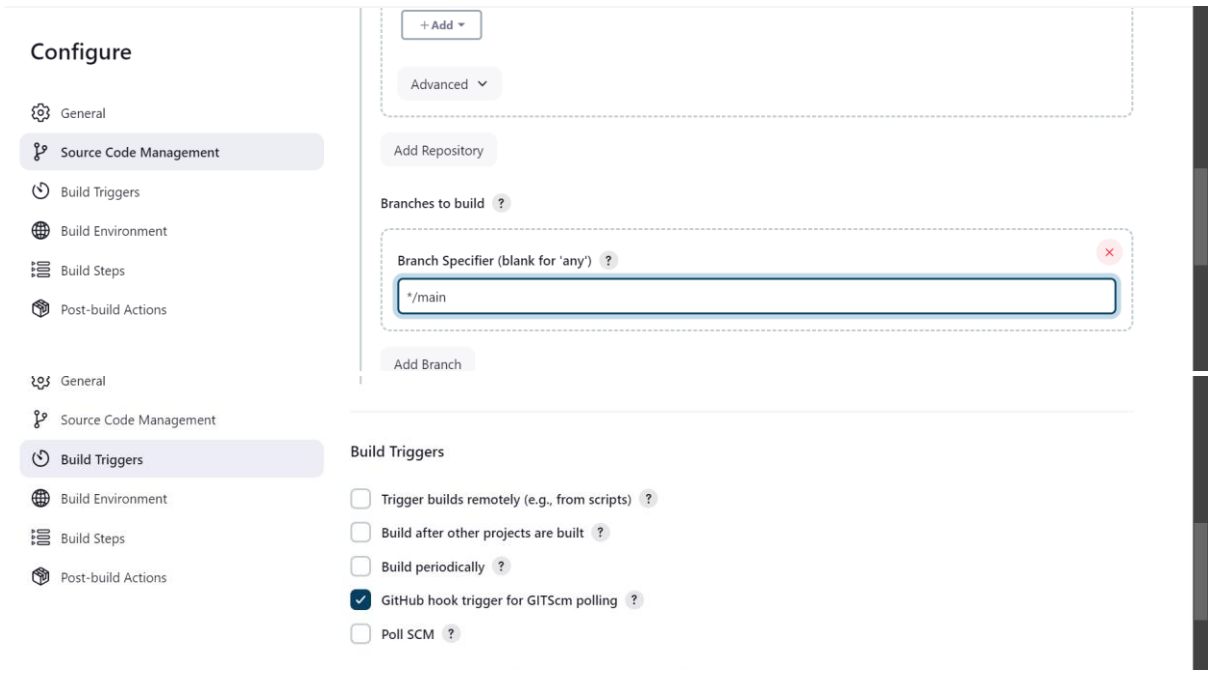


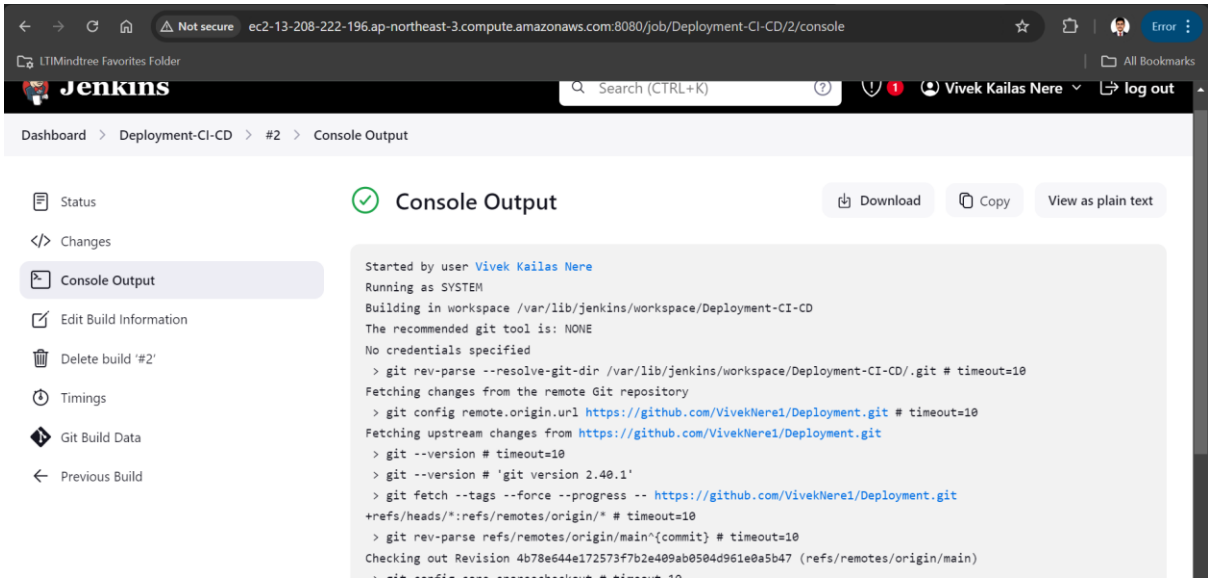

Not secure ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080
Jenkins
Search (CTRL+K)
Vivek Kailas Nere
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud

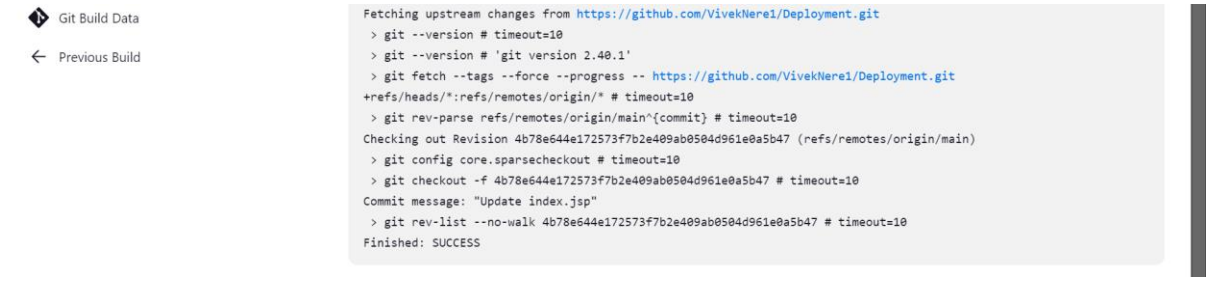

Not secure ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/user/admin/configure
Dashboard
Vivek Kailas Nere
Configure
Credentials
Plain text
Preview
API Token
Current token(s)
Token created on 2024-09-03T03:37:17.0
1143893bf3eb932e7ccb9ba759b0cc4355
Copy this token now, because it cannot be recovered in the future.
Add new Token


github.com/VivekNere1/Deployment/settings/hooks/new
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Webhooks / Add webhook
We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.
Payload URL *
http://13.57.202.161:8080/github-webhook/
Content type *
application/x-www-form-urlencoded
Secret
1143893bf3eb932e7ccb9ba759b0cc4355
SSL verification
By default, we verify SSL certificates when delivering payloads.
Enable SSL verification
Disable (not recommended)

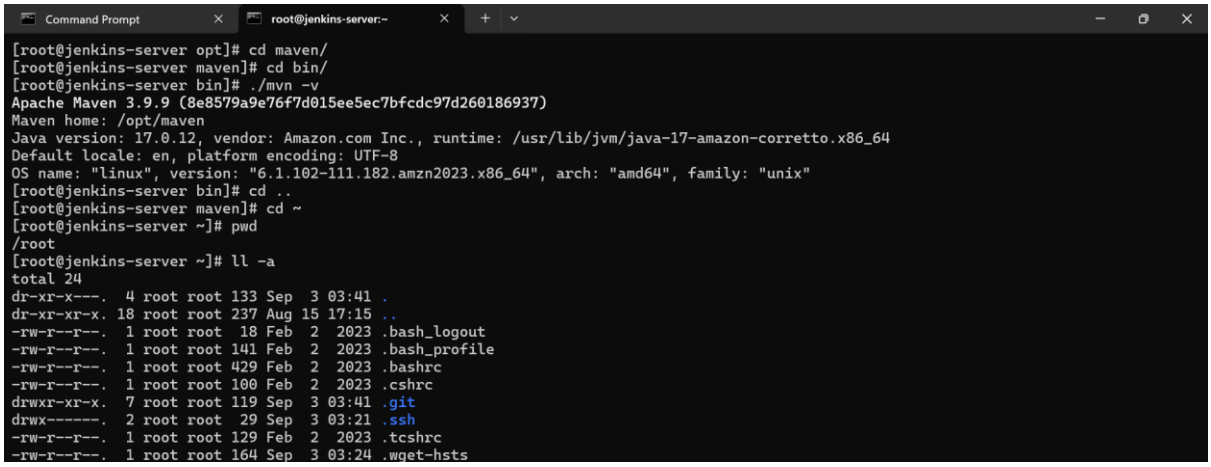

Dashboard > Deployment-CI-CD > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Repositories
Repository URL
https://github.com/VivekNere1/Deployment.git
Credentials
- none -
+ Add
Advanced
Add Repository
Branches to build
+ Add
Advanced


Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Add Branch
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM


Not secure
ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/job/Deployment-CI-CD/2/console
Error
LTIMindtree Favorites Folder
All Bookmarks
Jenkins
Search (CTRL+K)
Vivek Kailas Nere
log out
Dashboard > Deployment-CI-CD > #2 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#2'
Timings
Git Build Data
Previous Build
Console Output
Download
Copy
View as plain text
Started by user Vivek Kailas Nere
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Deployment-CI-CD
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Deployment-CI-CD/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/VivekNere1/Deployment.git # timeout=10
Fetching upstream changes from https://github.com/VivekNere1/Deployment.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/VivekNere1/Deployment.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4b78e644e172573f7b2e409ab0504d961e0a5b47 (refs/remotes/origin/main)

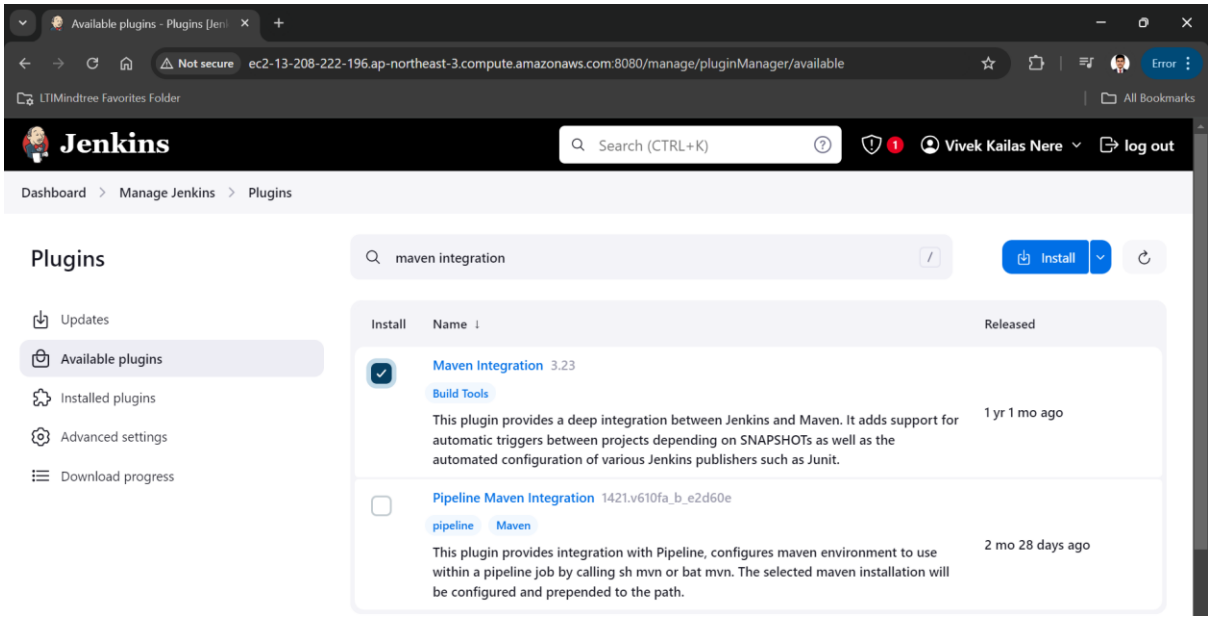


#Installation and Configuration of maven:

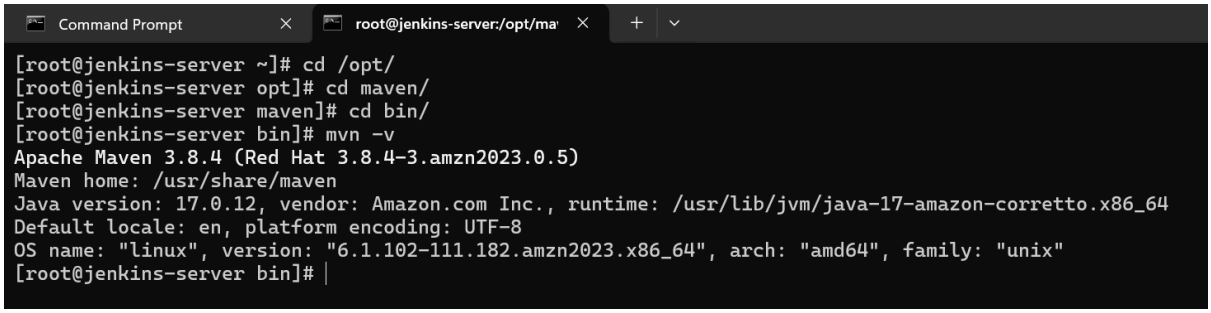


#Install maven plugins and configure Jenkins for maven

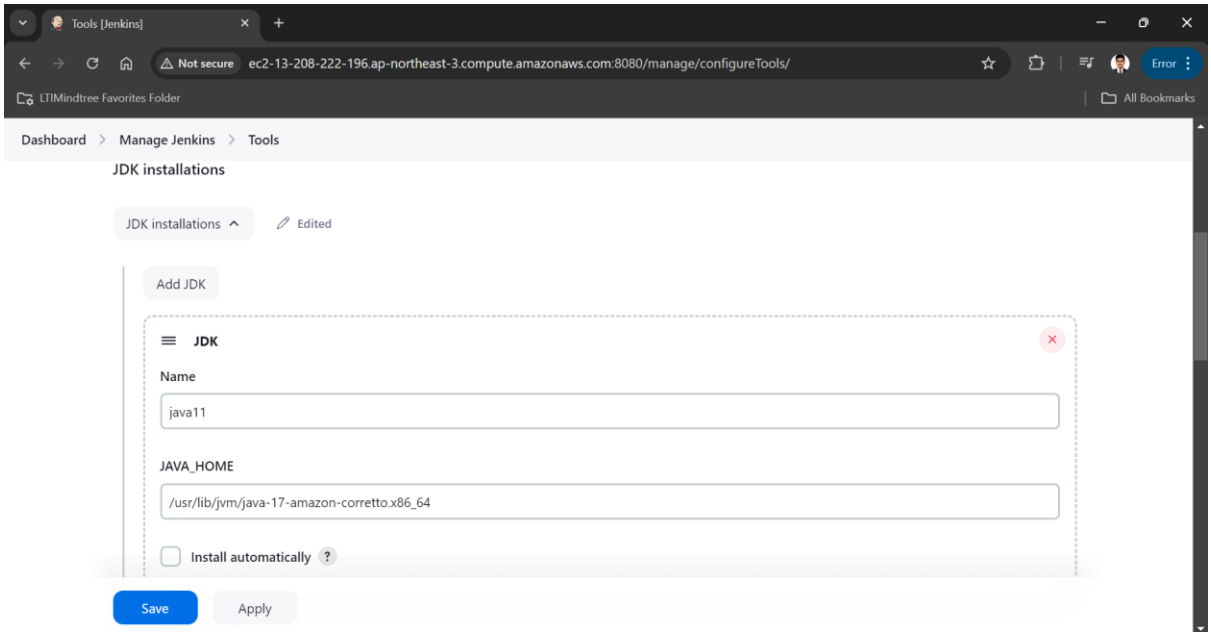
in Jenkins -> manage Jenkins -> plugins -> available plugins -> install maven integration

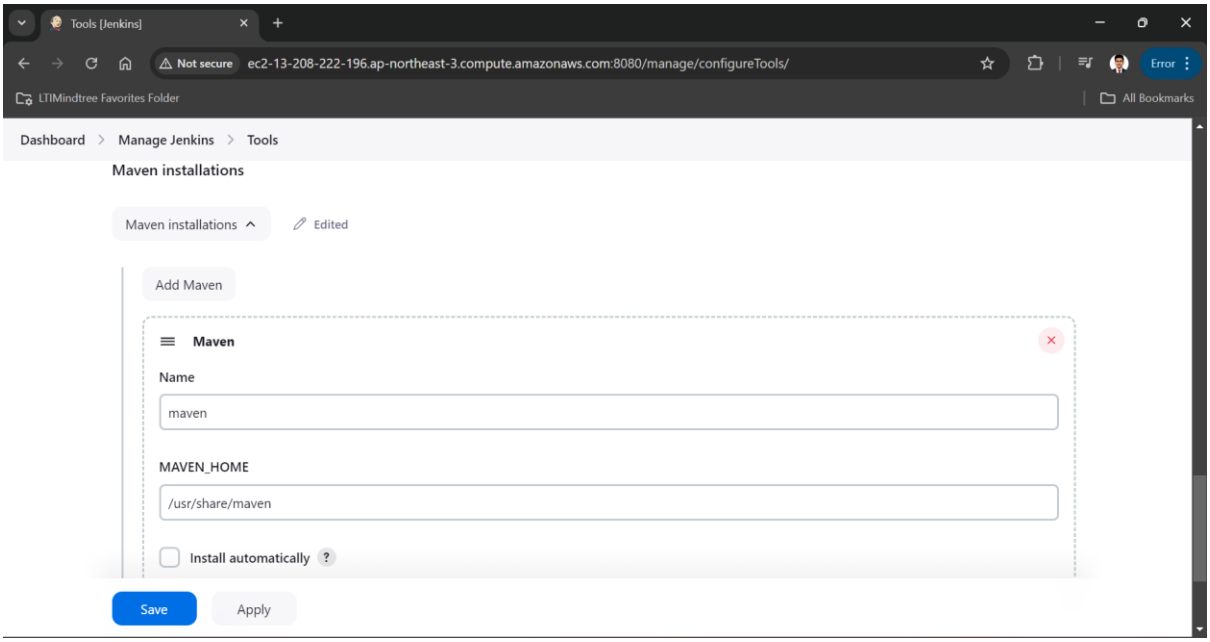


To find out the home directory of maven and java

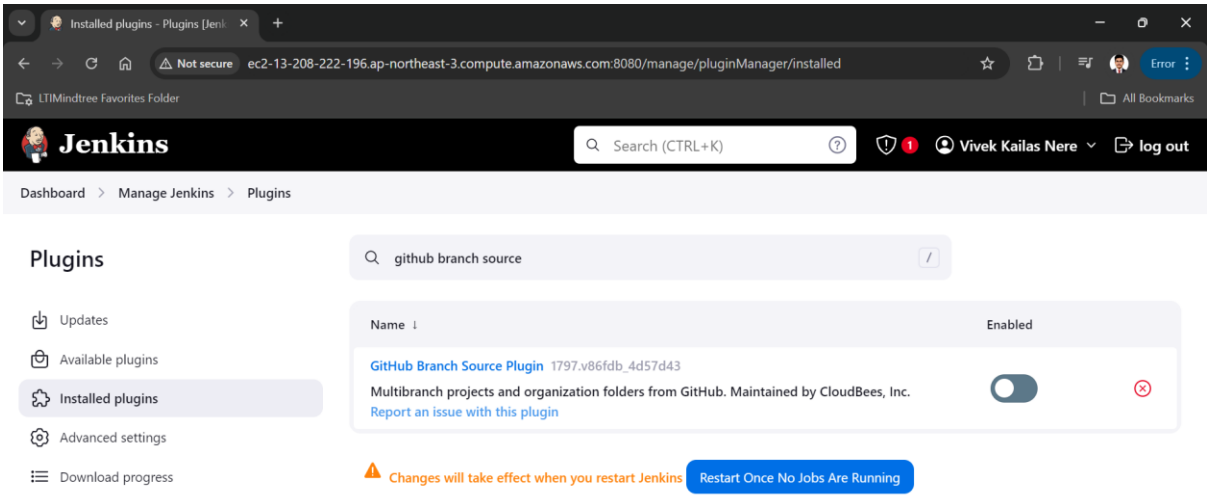


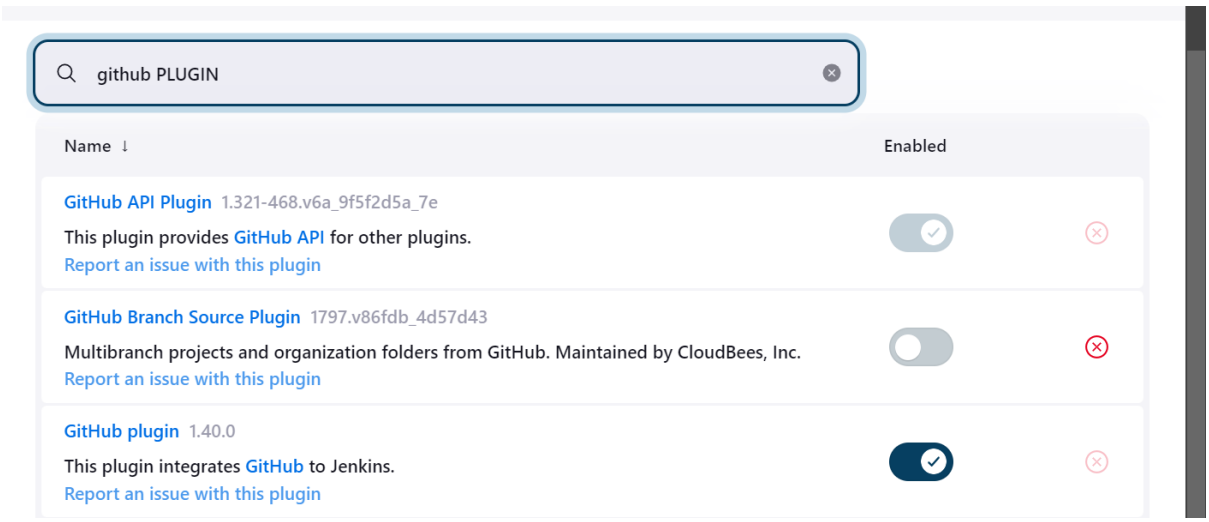
- now we need to configure this plugins for that go to manage Jenkins tools-> add JDK -> name=java11 JAVA_HOME=/usr/lib/jvm/jre-17-openjdk then add maven ==> name=maven , MAVEN_HOME= /usr/share/maven

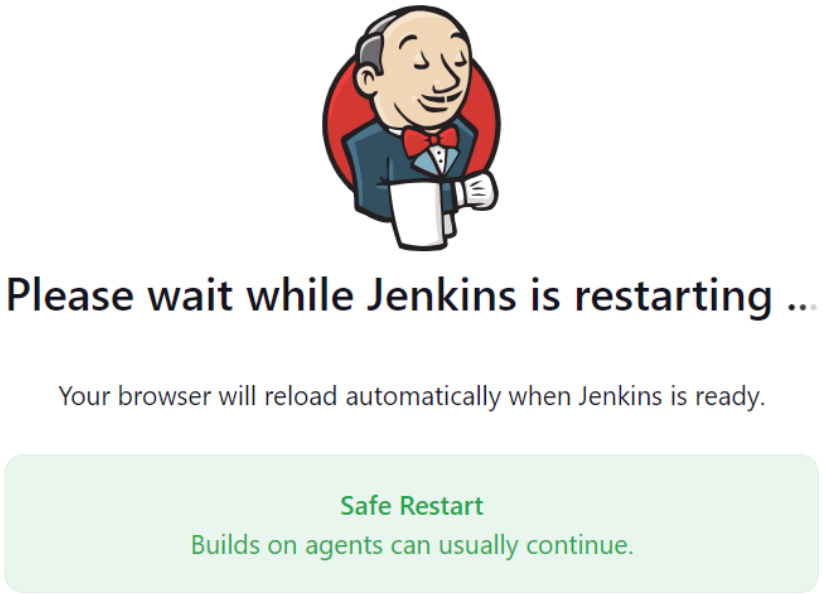




- now manage Jenkins -> plugins -> Intsalled plugins -> Disable the GITHUB branch source plugins & enable GitHub plugin and restart.



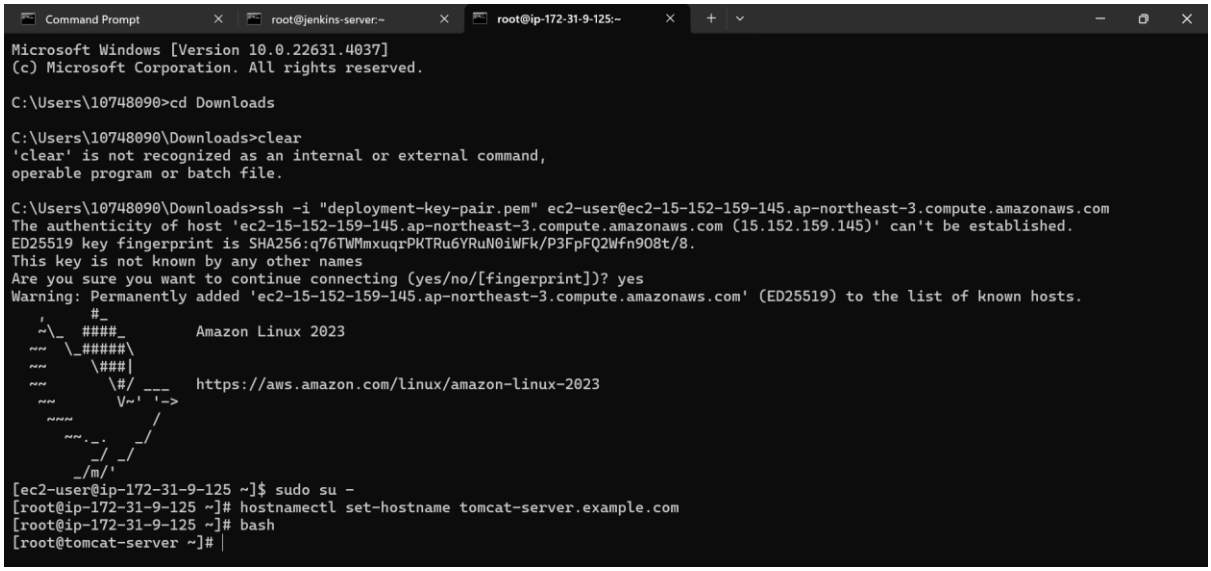




#3: Tomcat

#Installation and configuration of Tomcat:

Open the tomcat-server instance into local machine by using ssh and set a hostname as tomcat-server



Installed java in tomcat-server

```
[root@tomcat-server ~]# dnf install java-17-amazon-corretto -y
Last metadata expiration check: 14:33:58 ago on Mon Sep  2 14:24:17 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
java-17-amazon-corretto	x86_64	1:17.0.12+7-1.amzn2023.1	amazonlinux	187 k
Installing dependencies:				
alsa-lib	x86_64	1.2.7.2-1.amzn2023.0.2	amazonlinux	504 k
cairo	x86_64	1.17.6-2.amzn2023.0.1	amazonlinux	684 k
dejavu-sans-fonts	noarch	2.37-16.amzn2023.0.2	amazonlinux	1.3 M
dejavu-sans-mono-fonts	noarch	2.37-16.amzn2023.0.2	amazonlinux	467 k
dejavu-serif-fonts	noarch	2.37-16.amzn2023.0.2	amazonlinux	1.0 M
fontconfig	x86_64	2.13.94-2.amzn2023.0.2	amazonlinux	273 k
fonts-filesystem	noarch	1:2.0.5-12.amzn2023.0.2	amazonlinux	9.5 k
freetype	x86_64	2.13.2-5.amzn2023.0.1	amazonlinux	423 k
giflib	x86_64	5.2.1-9.amzn2023.0.1	amazonlinux	49 k
google-noto-fonts-common	noarch	20201206-2.amzn2023.0.2	amazonlinux	15 k
google-noto-sans-vf-fonts	noarch	20201206-2.amzn2023.0.2	amazonlinux	492 k
graphite2	x86_64	1.3.14-7.amzn2023.0.2	amazonlinux	97 k
harfbuzz	x86_64	7.0.0-2.amzn2023.0.1	amazonlinux	868 k
java-17-amazon-corretto-headless	x86_64	1:17.0.12+7-1.amzn2023.1	amazonlinux	91 M
javapackages-filesystem	noarch	6.0.0-7.amzn2023.0.6	amazonlinux	12 k
langpacks-core-font-en	noarch	3.0-21.amzn2023.0.4	amazonlinux	10 k
libICE	x86_64	1.0.10-6.amzn2023.0.2	amazonlinux	71 k
libSM	x86_64	1.2.3-8.amzn2023.0.2	amazonlinux	42 k
libX11	x86_64	1.7.2-3.amzn2023.0.4	amazonlinux	657 k
libX11-common	noarch	1.7.2-3.amzn2023.0.4	amazonlinux	152 k
libXau	x86_64	1.0.9-6.amzn2023.0.2	amazonlinux	31 k
libXext	x86_64	1.3.4-6.amzn2023.0.2	amazonlinux	41 k
libXi	x86_64	1.7.10-6.amzn2023.0.2	amazonlinux	40 k
libXinerama	x86_64	1.1.4-8.amzn2023.0.2	amazonlinux	15 k

Download tar file of tomcat from tomcat official website in /opt folder

```
[root@tomcat-server ~]# cd /opt/
[root@tomcat-server opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
--2024-09-03 05:06:13--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12122732 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.93.tar.gz'

apache-tomcat-9.0.93.tar.gz       100%[=============================================================>]  11.56M  --.-KB/s    in 0.03s

2024-09-03 05:06:14 (396 MB/s) - 'apache-tomcat-9.0.93.tar.gz' saved [12122732/12122732]

[root@tomcat-server opt]#
```

Extract tar file:

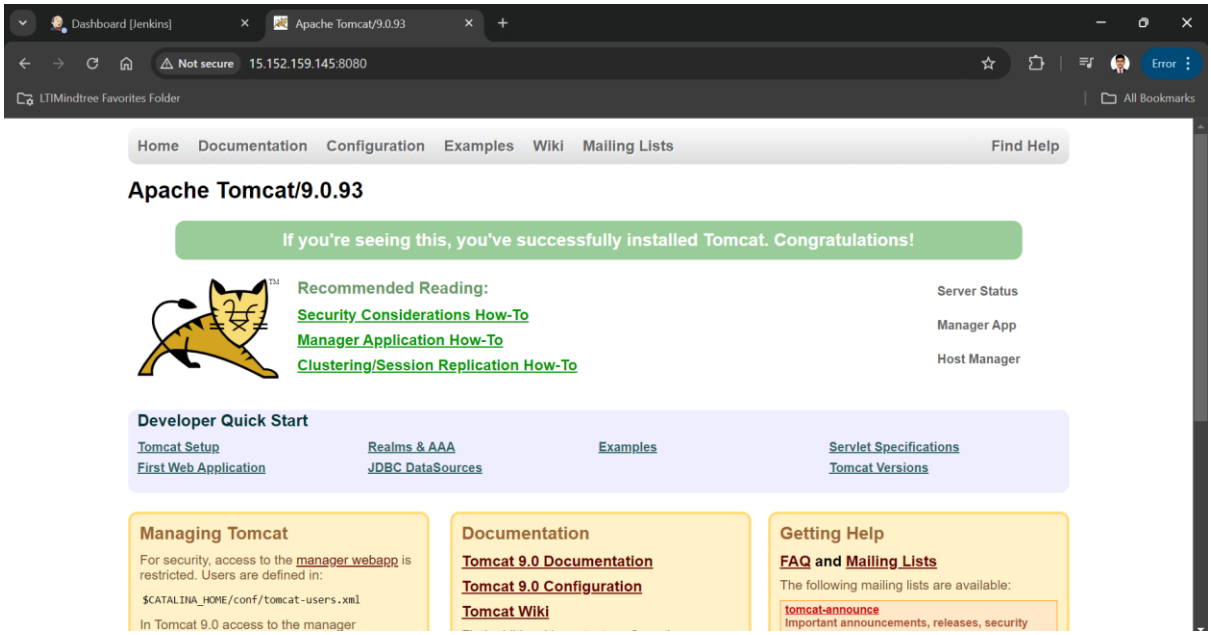
```
[root@tomcat-server opt]# ls
apache-tomcat-9.0.93.tar.gz  aws
[root@tomcat-server opt]# tar xvzf apache-tomcat-9.0.93.tar.gz
apache-tomcat-9.0.93/conf/
apache-tomcat-9.0.93/conf/catalina.policy
apache-tomcat-9.0.93/conf/catalina.properties
apache-tomcat-9.0.93/conf/context.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xsd
```

Move in tomcat folder

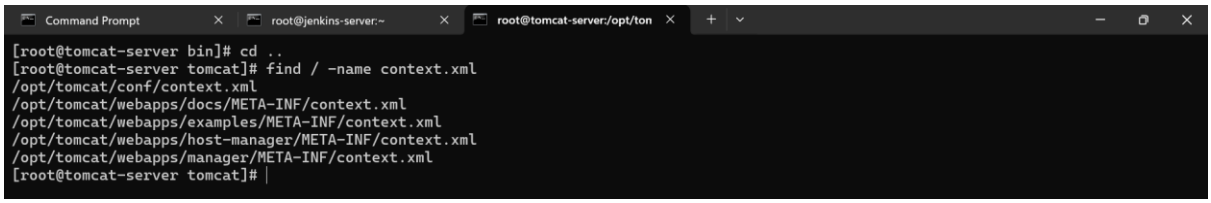
Now we have to start the tomcat server

```
[root@tomcat-server opt]# mv apache-tomcat-9.0.93 tomcat
[root@tomcat-server opt]# cd tomcat
[root@tomcat-server tomcat]# cd bin
[root@tomcat-server bin]# ./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat-server bin]#
```

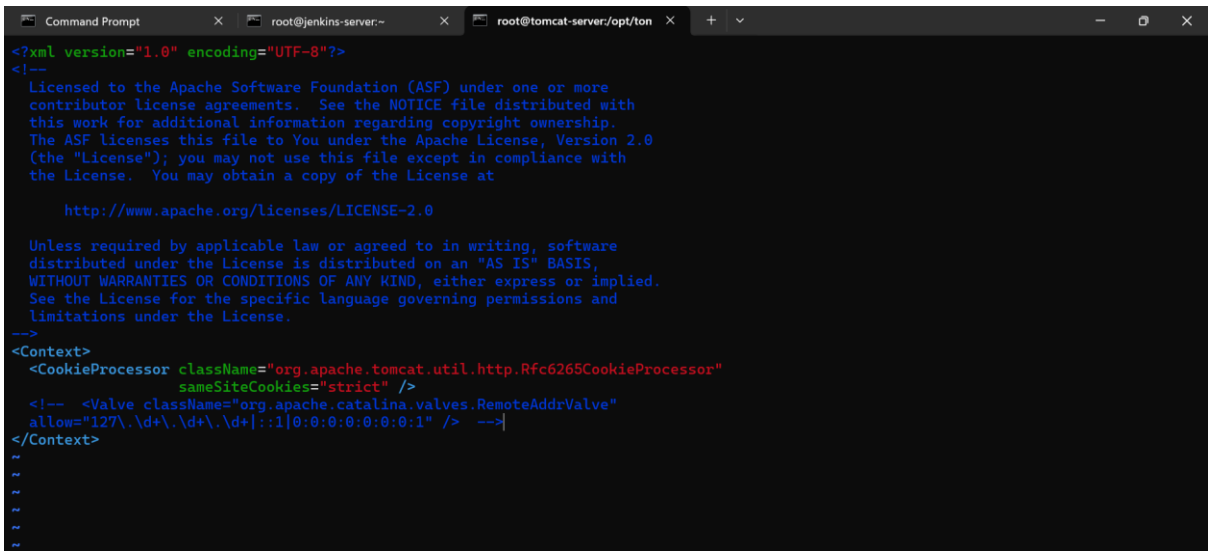
in google open http://(ip of tomcat server instance):8080



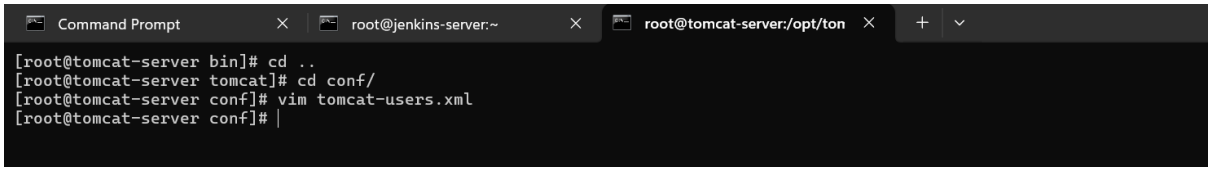
now we have make some changes in context.xml file for that we need to find that file in tomcat

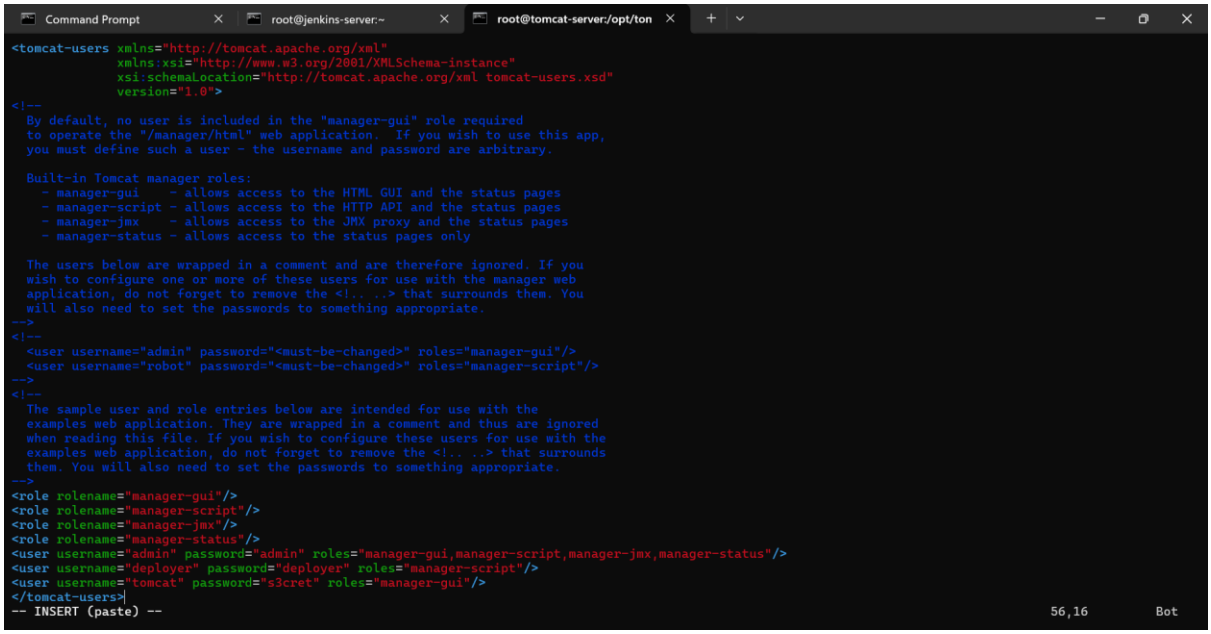


#comment value tag sections in below all files
[root@tomcat-server bin]# vim /opt/tomcat/webapps/examples/META-INF/context.xml
[root@tomcat-server bin]# vim /opt/tomcat/webapps/host-manager/META-INF/context.xml
[root@tomcat-server bin]# vin /opt/tomcat/webapps/manager/META-INF/context.xml
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> --!>

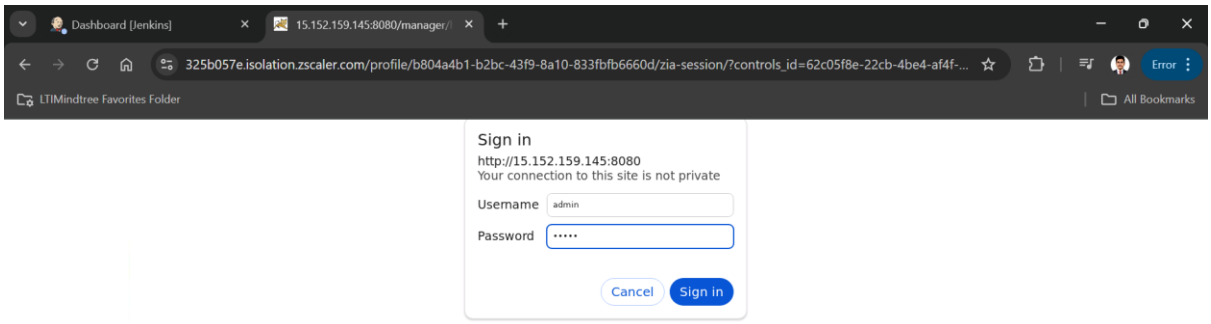


Now we need to change tomcat user configuration info:



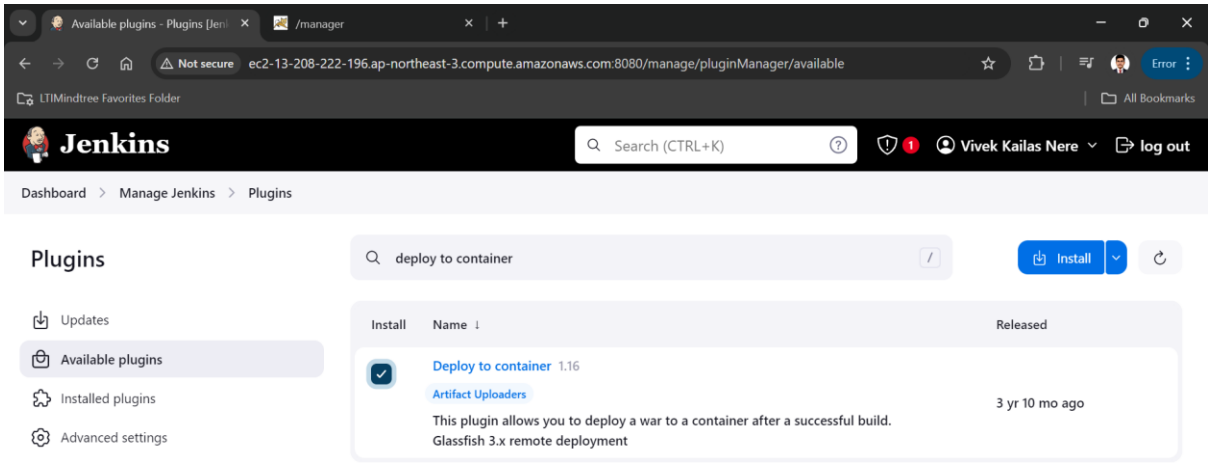


now we need to reboot the tomcat server

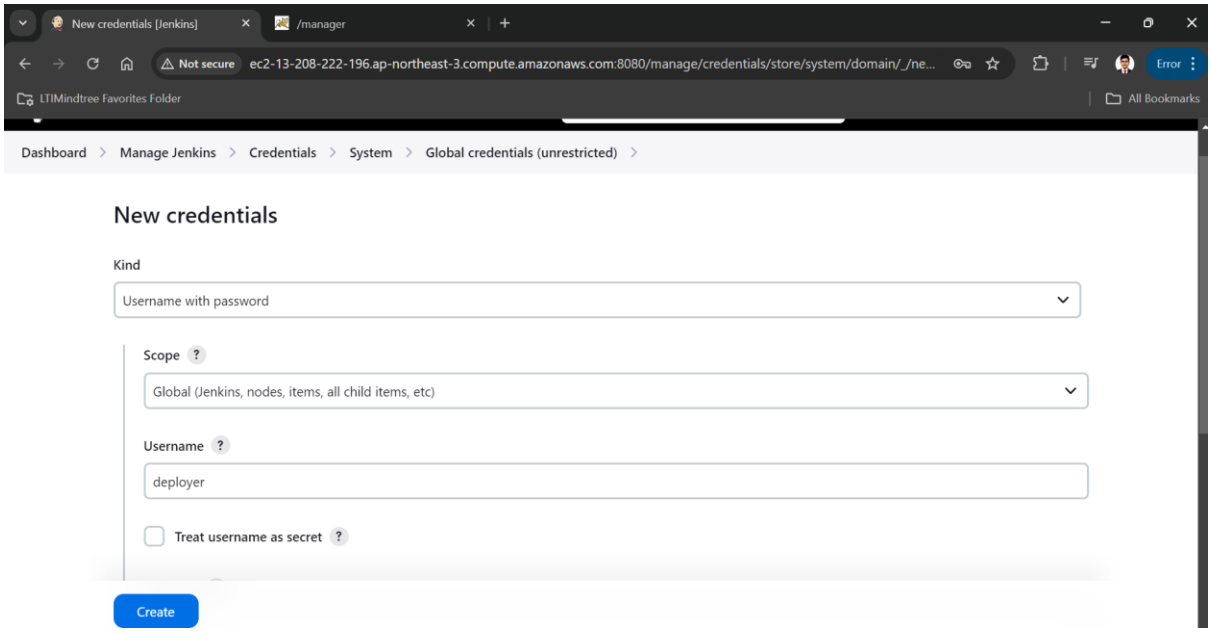


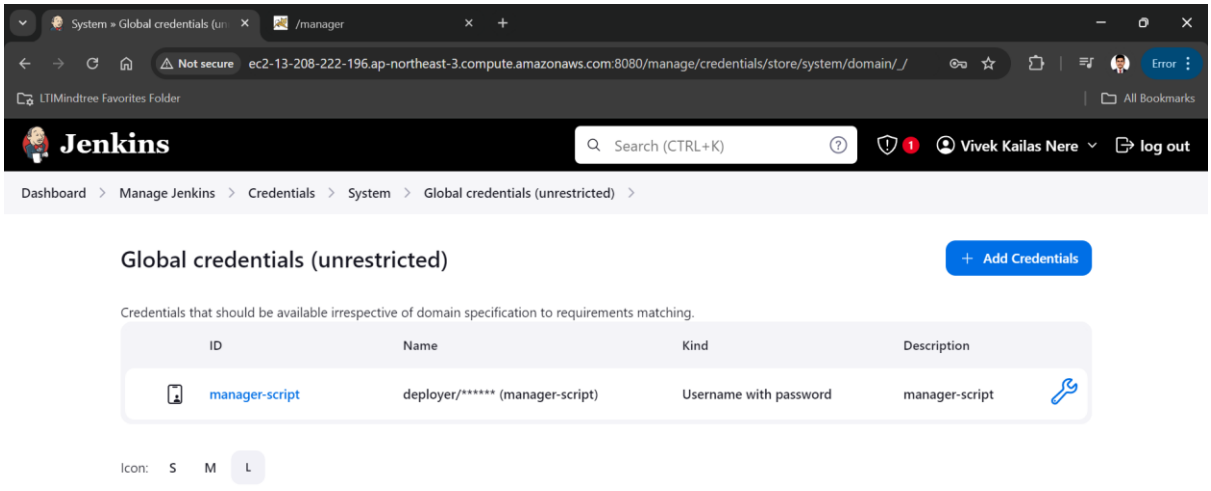
Integrating tomcat server with Jenkins:
there is no predefined tomcat plugin in Jenkins so we need to install "deploy to container plugin" then configure the tomcat server with credentials

manage genkins -> plugins -> available plugins -> deploy to container

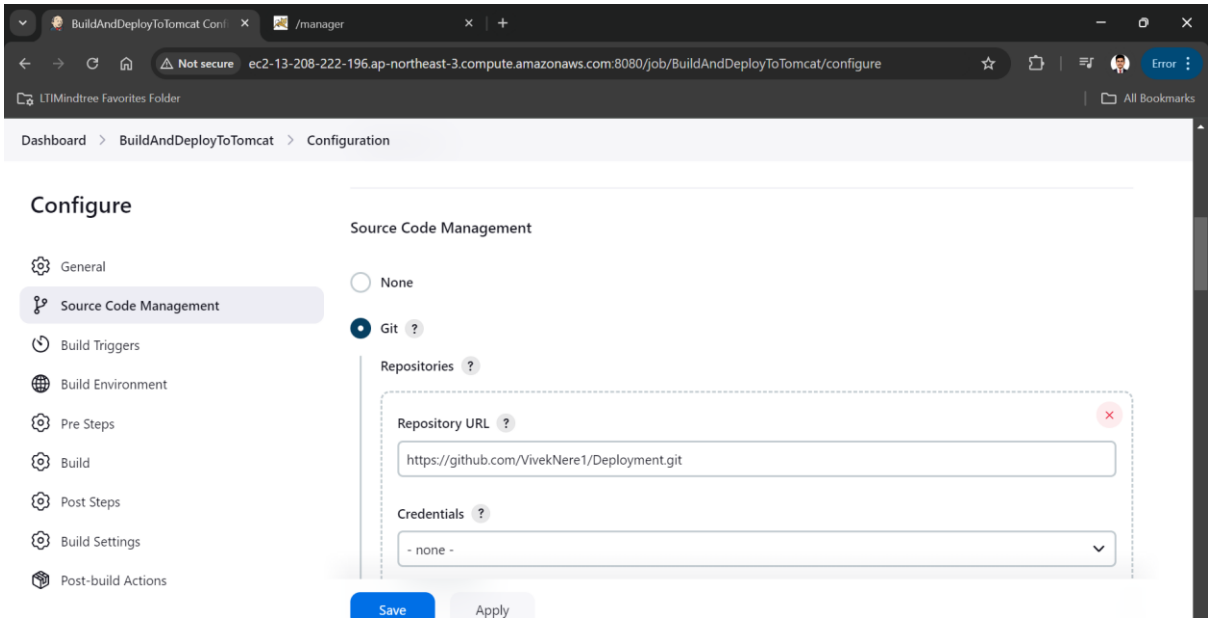


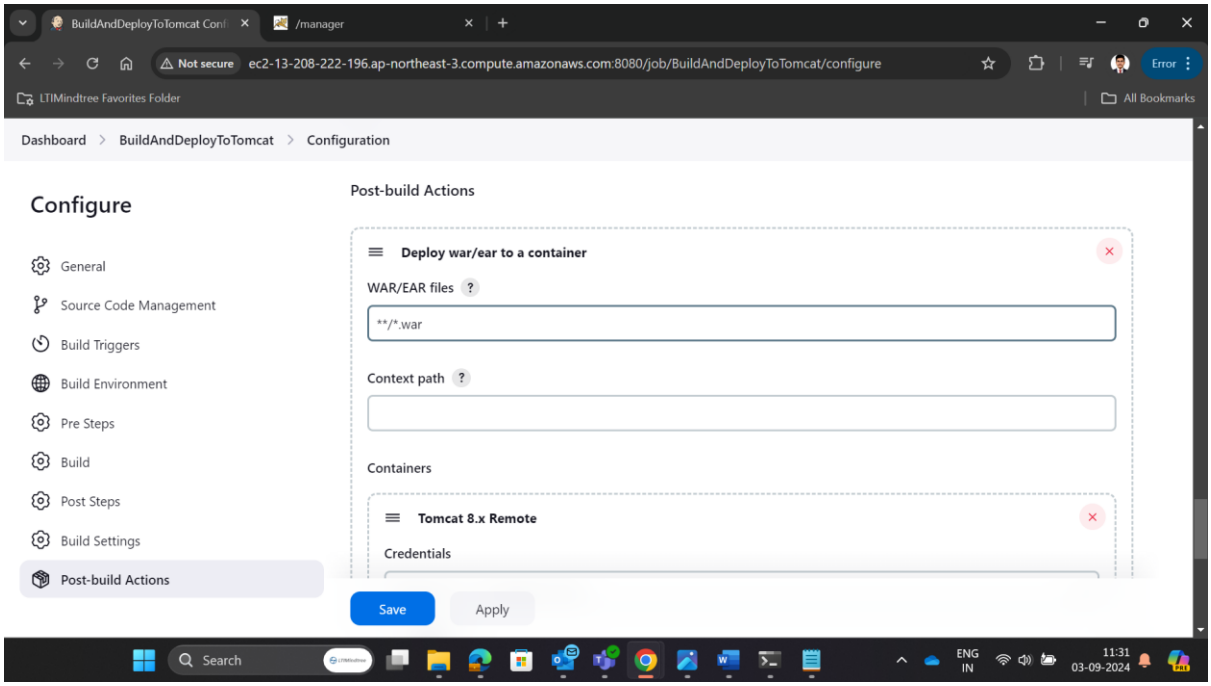
now we need to add credentials of tomcat to the Jenkins
manage Jenkins -> credentials -> system -> global credentials -> add credentials(<user username="deployer" password="deployer" roles="manager-script"/>) id and description=tomcat-credentials

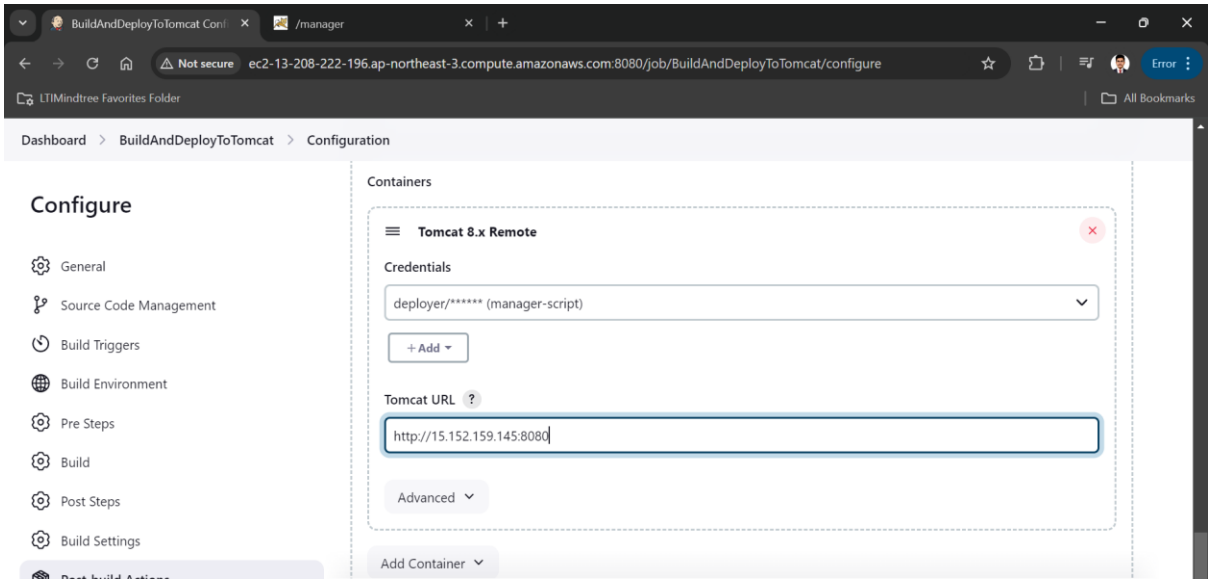




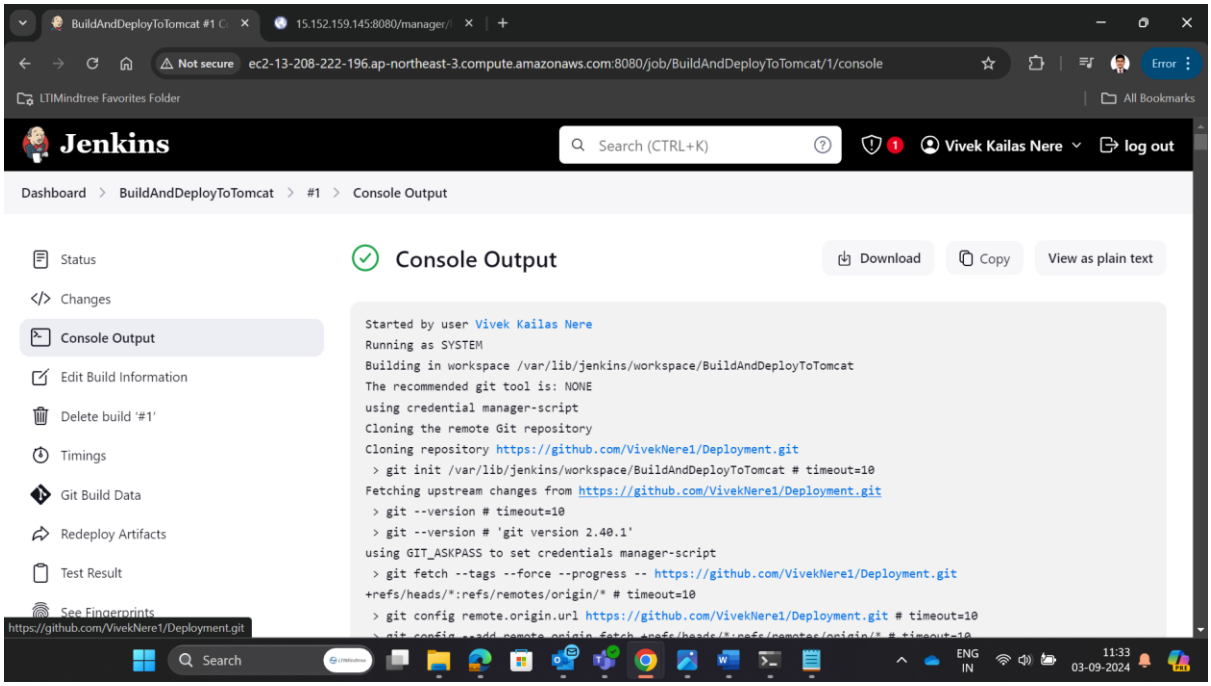
then go to dashboard and create new item name as BuildAndDeployToTomcat as a maven project
add git repository http url
make branch main
and at the last click on "Add post-build action" and select "Deploy war/ear to a container"
in war/ear file add "**/*.war"
add container Tomcat 8.x
add tomcat-server instance url
save and apply

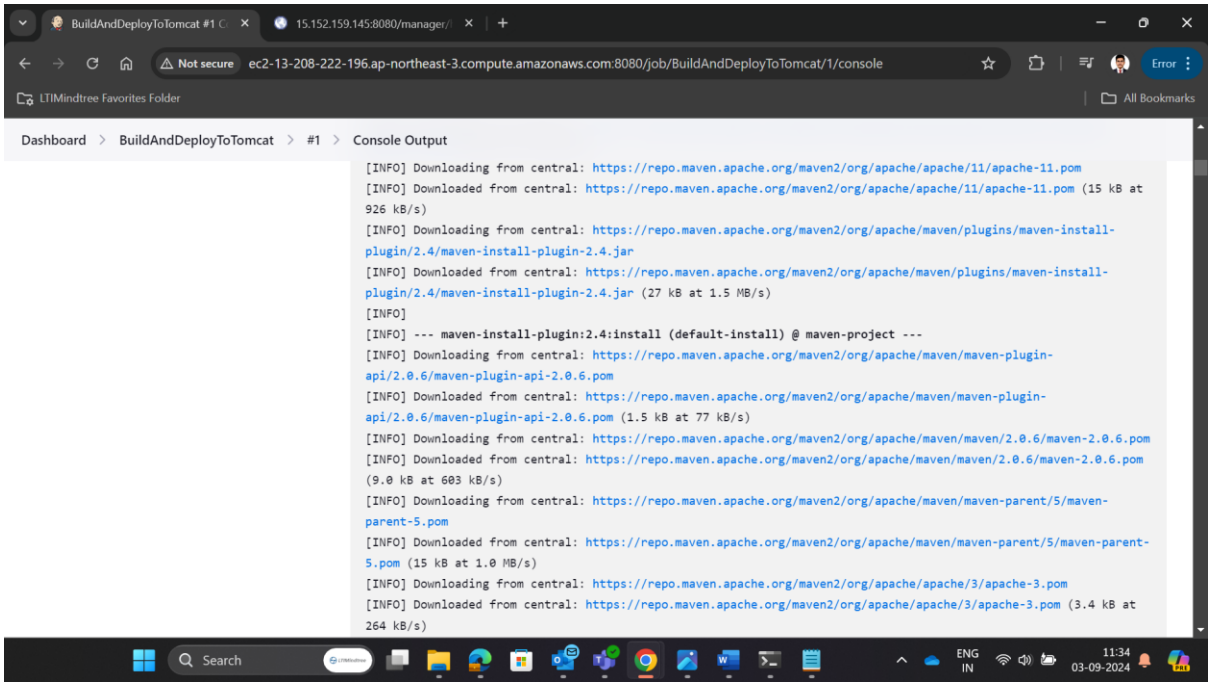


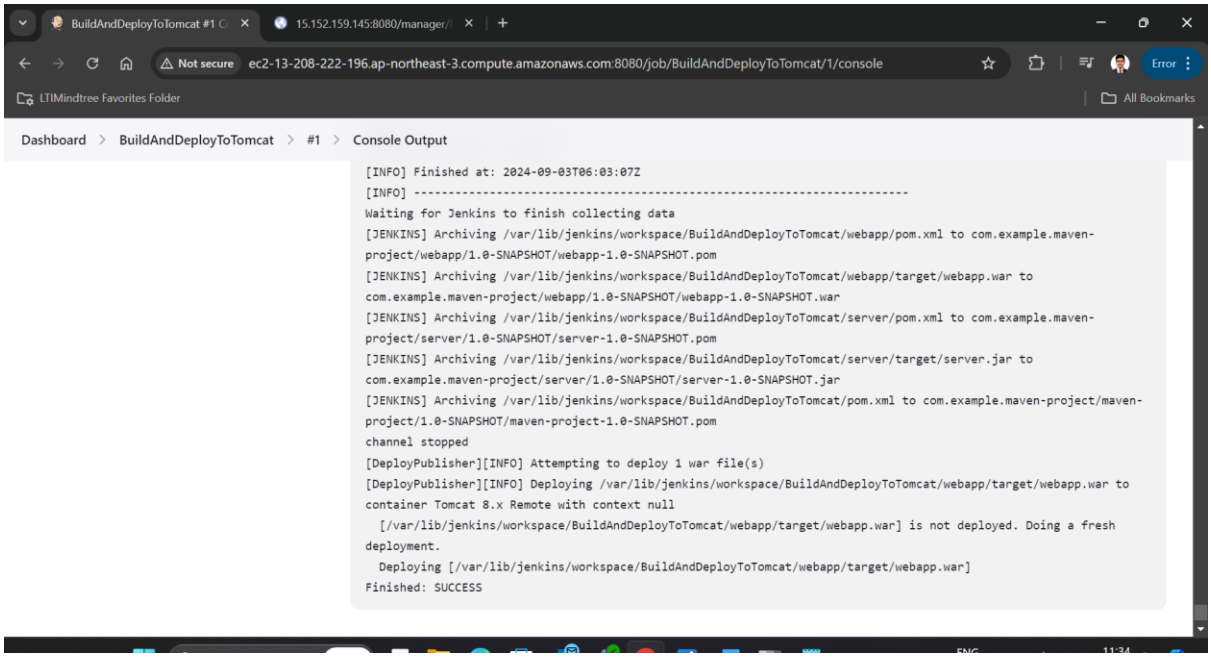


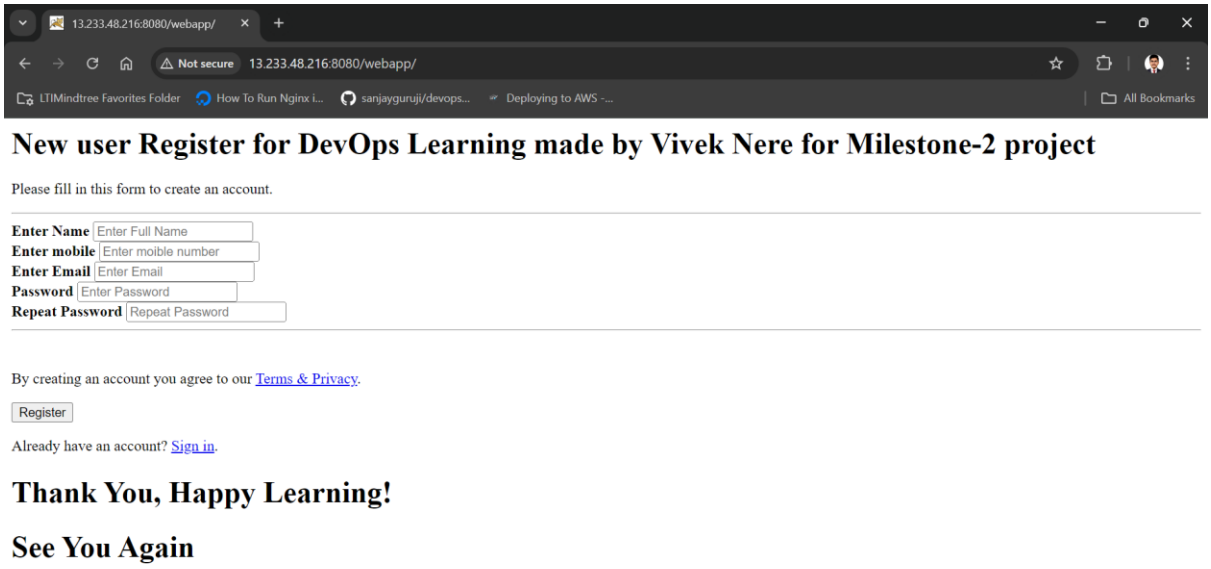


We have suceesfully deployed project on tomcat server



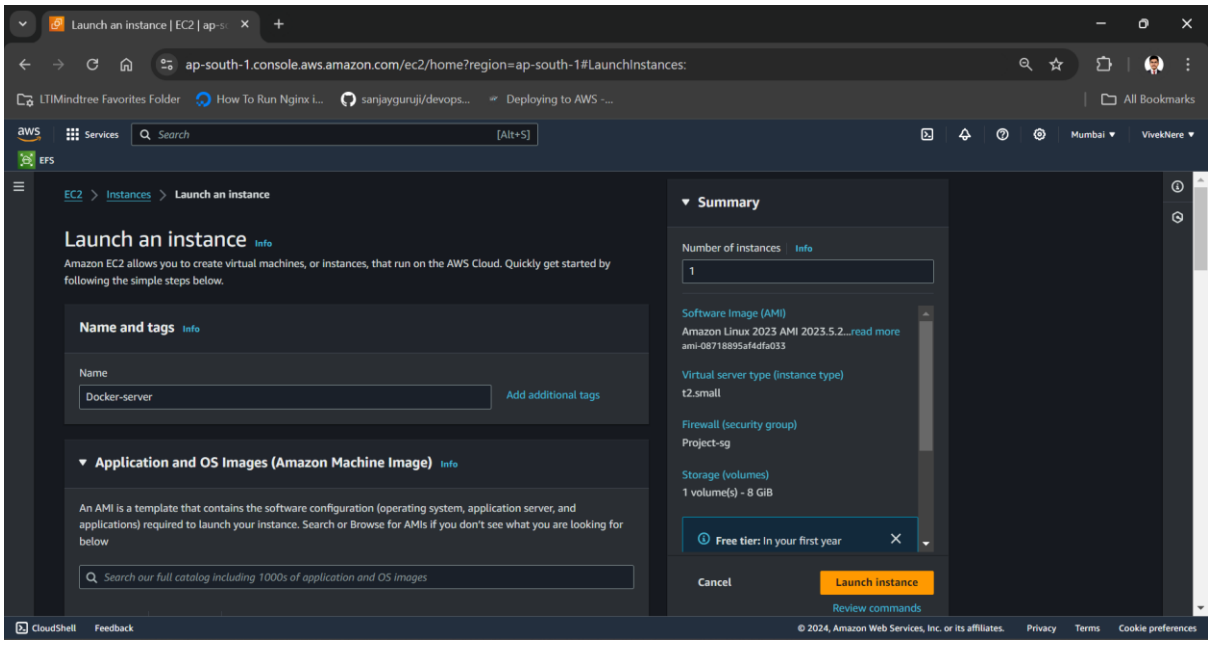

BuildAndDeployToTomcat #1
15.152.159.145:8080/manager/
Not secure
ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/job/BuildAndDeployToTomcat/1/console
Error
LTIMindtree Favorites Folder
All Bookmarks
Dashboard > BuildAndDeployToTomcat > #1 > Console Output
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/11/apache-11.pom (15 kB at 926 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-install-plugin-2.4.jar (27 kB at 1.5 MB/s)
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ maven-project ---
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom (1.5 kB at 77 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom (9.0 kB at 603 kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 1.0 MB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom (3.4 kB at 264 kB/s)
Search
ENG
IN
11:34
03-09-2024

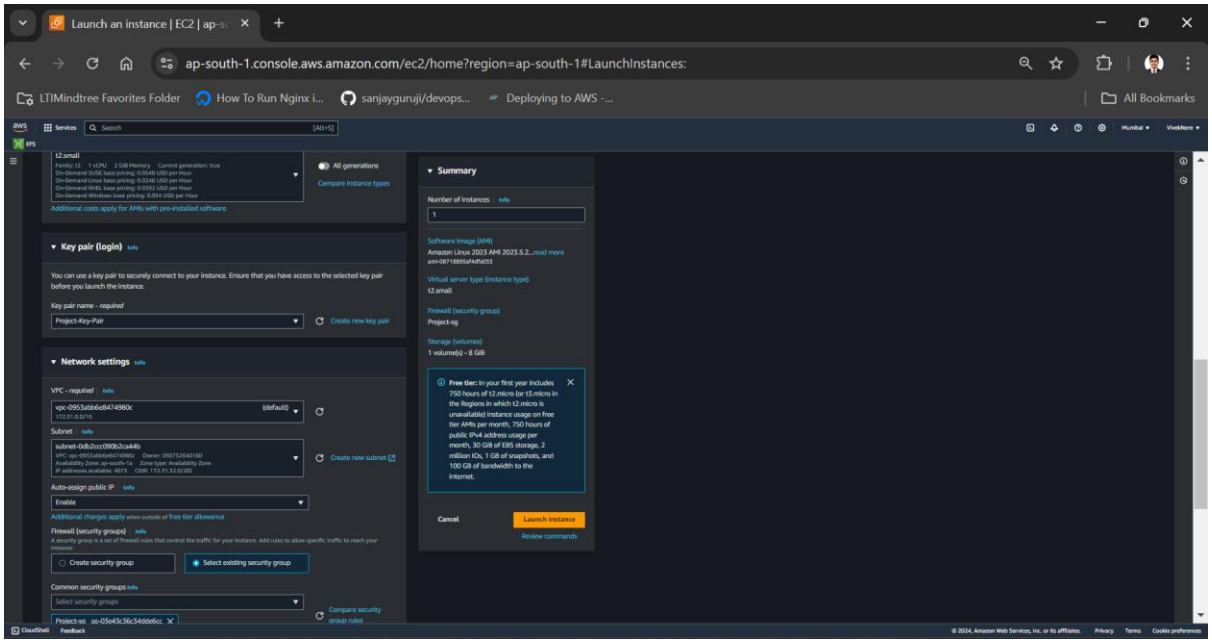

BuildAndDeployToTomcat #1
15.152.159.145:8080/manager/
Not secure
ec2-13-208-222-196.ap-northeast-3.compute.amazonaws.com:8080/job/BuildAndDeployToTomcat/1/console
Error
LTIMindtree Favorites Folder
All Bookmarks
Dashboard > BuildAndDeployToTomcat > #1 > Console Output
[INFO] Finished at: 2024-09-03T06:03:07Z
[INFO] --
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployToTomcat/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war to container Tomcat 8.x Remote with context null
[/var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/BuildAndDeployToTomcat/webapp/target/webapp.war]
Finished: SUCCESS

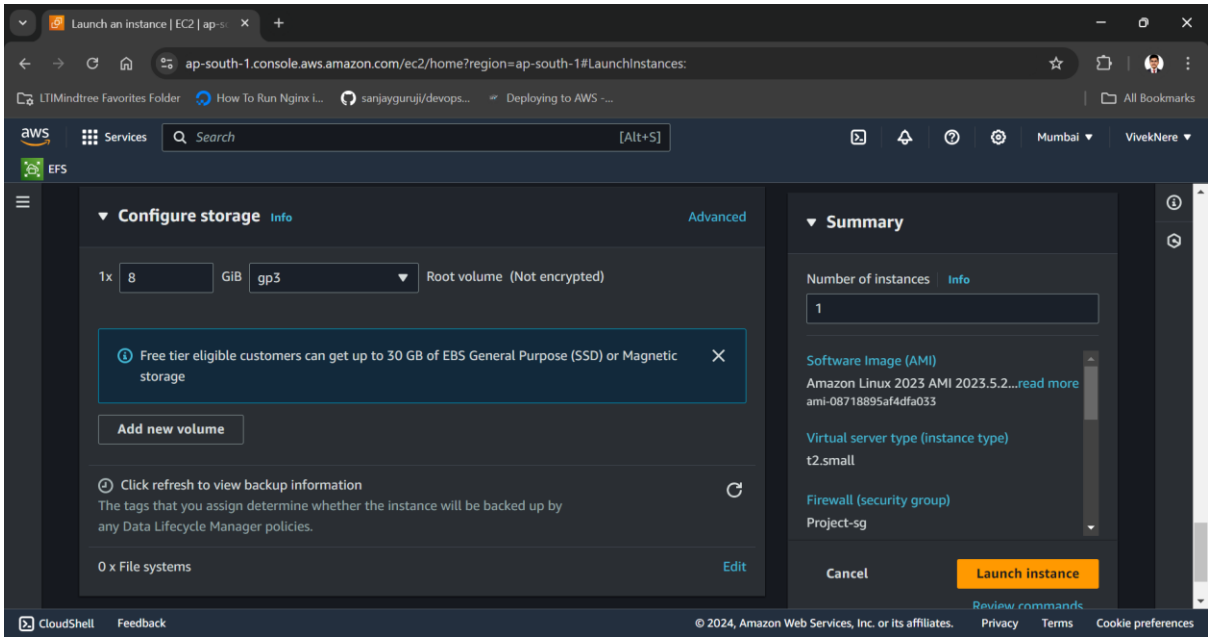


#4: Docker

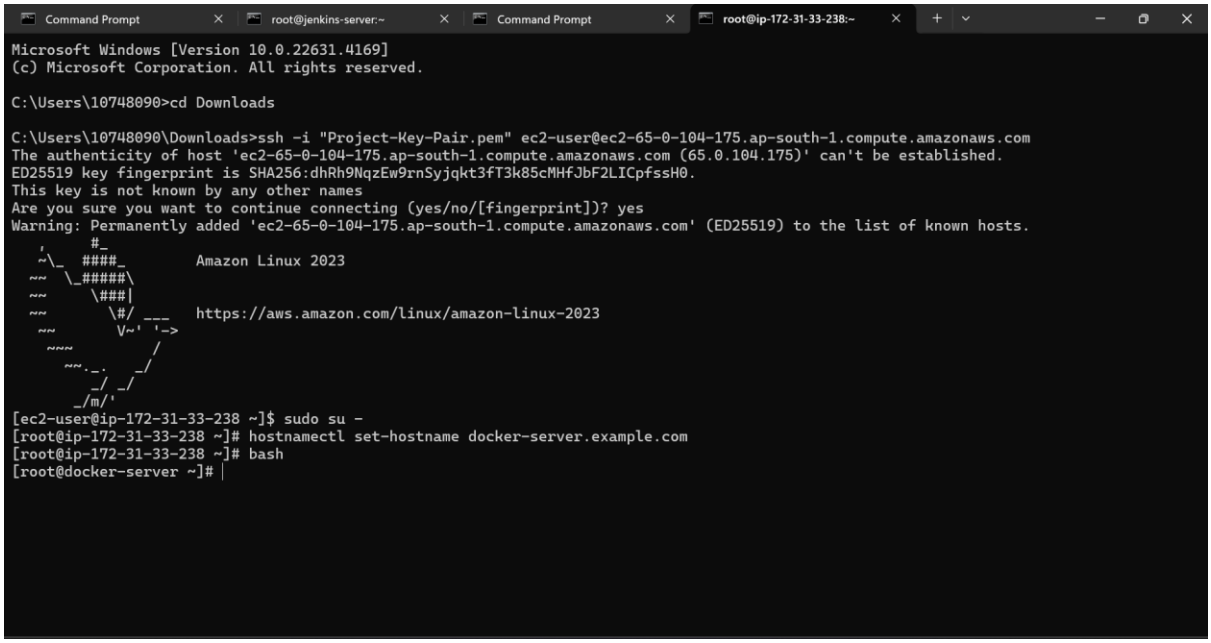
Docker Installation and Configuration with jenkins :
First create an instance for docker using amazon linux as OS with instance type t2.micro and same key-pair and security group which we have use in previous instances.



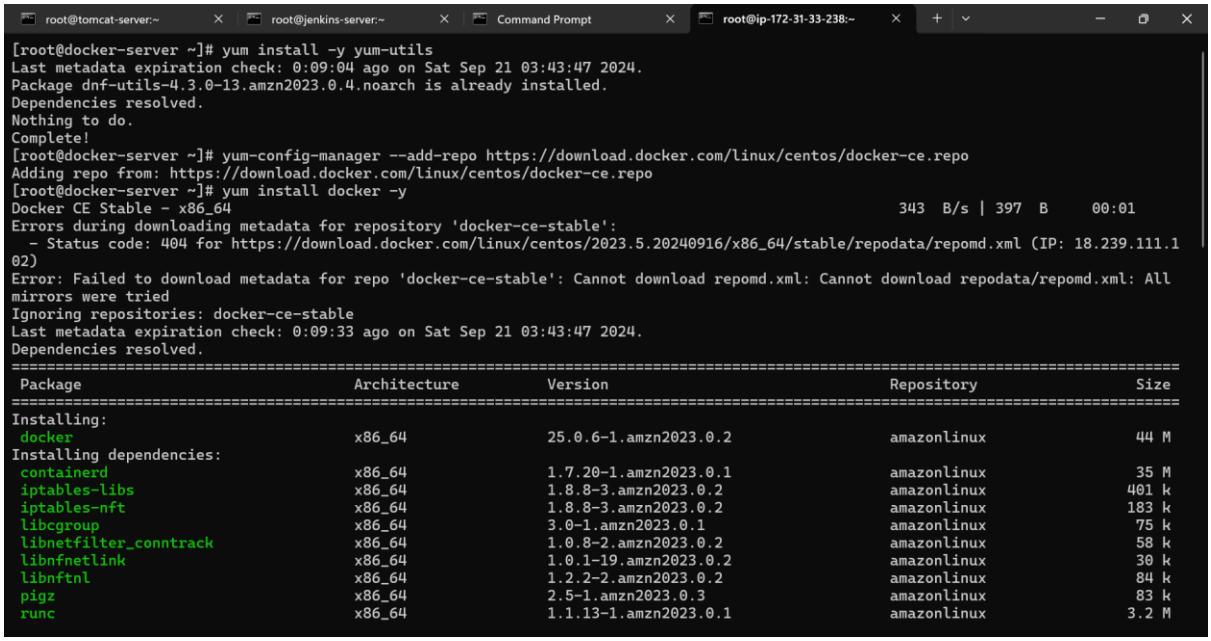


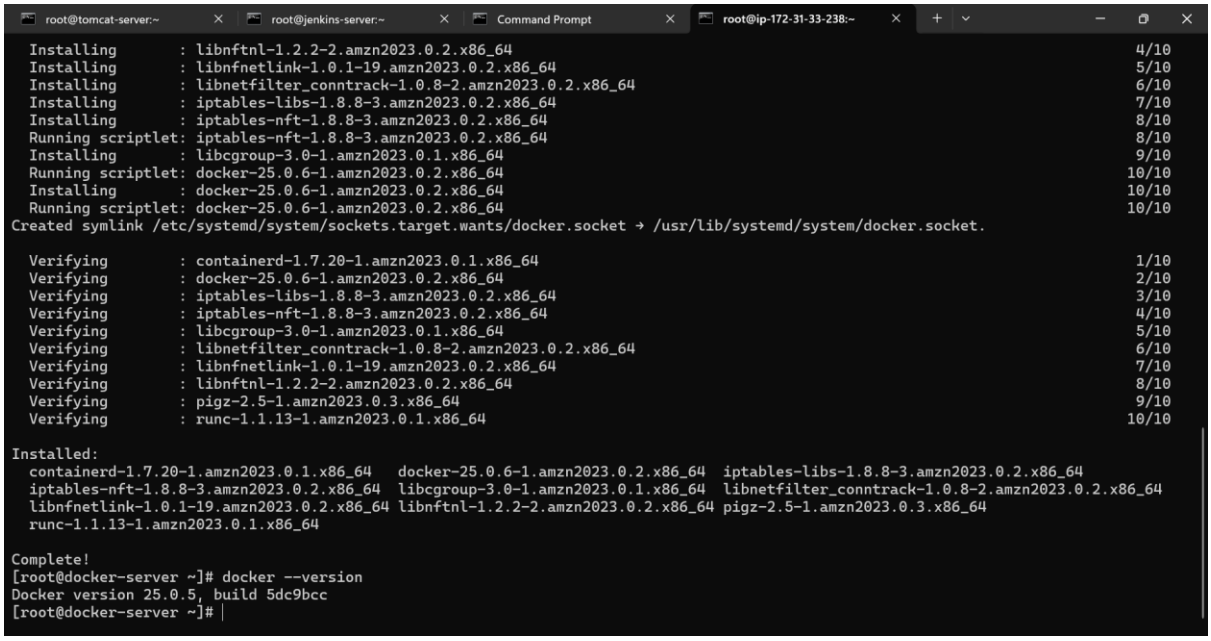


Now connect this docker ec2 in terminal and give hostname as docker-server to it.

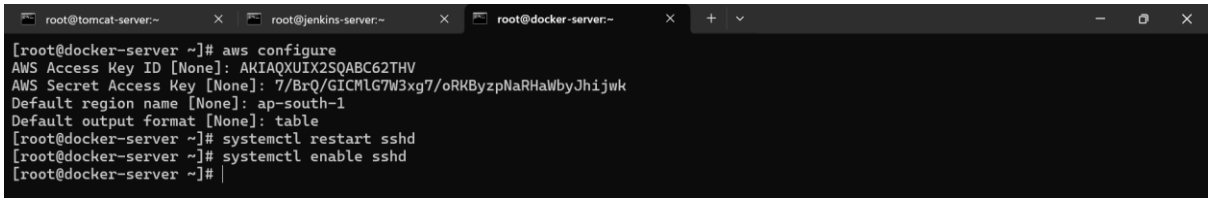


Now we have to install docker in docker-instance.

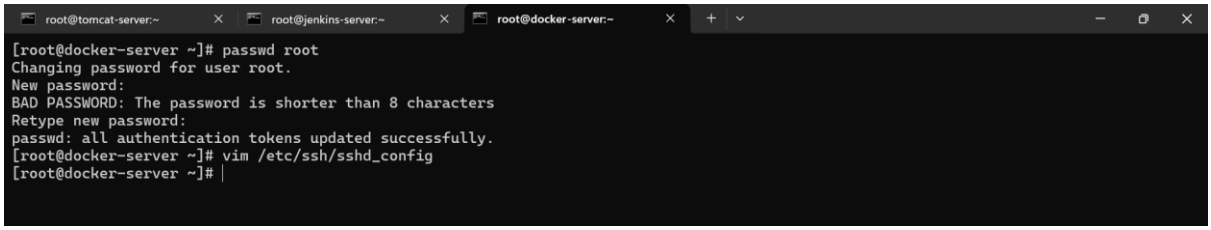


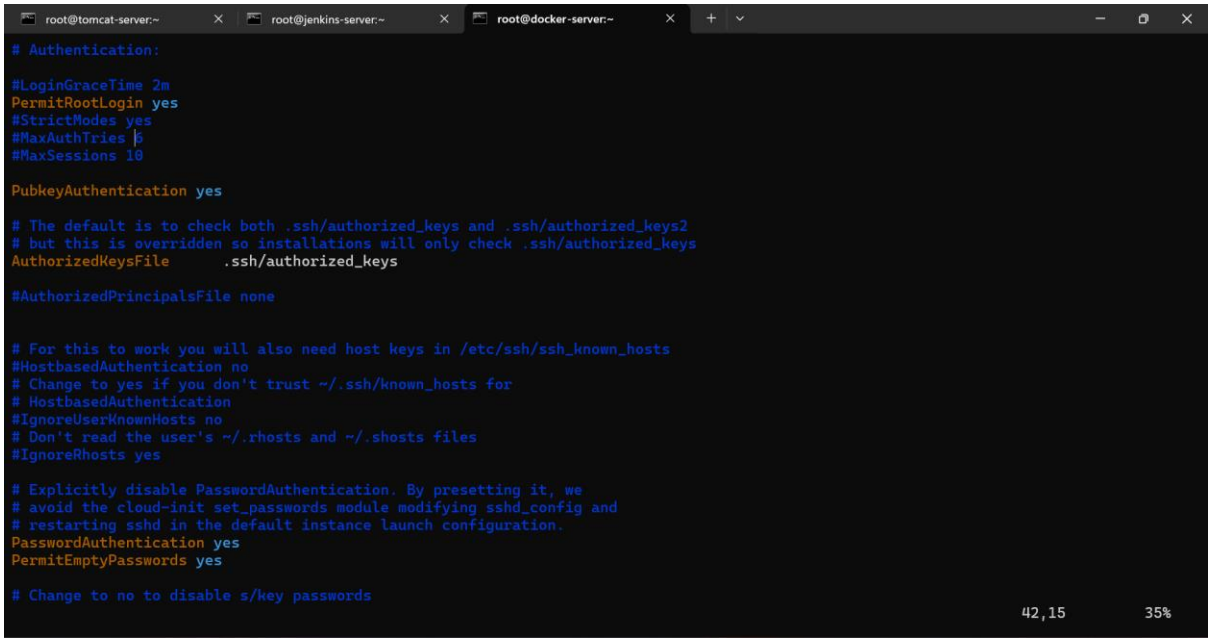


Now aws configure in docker-server

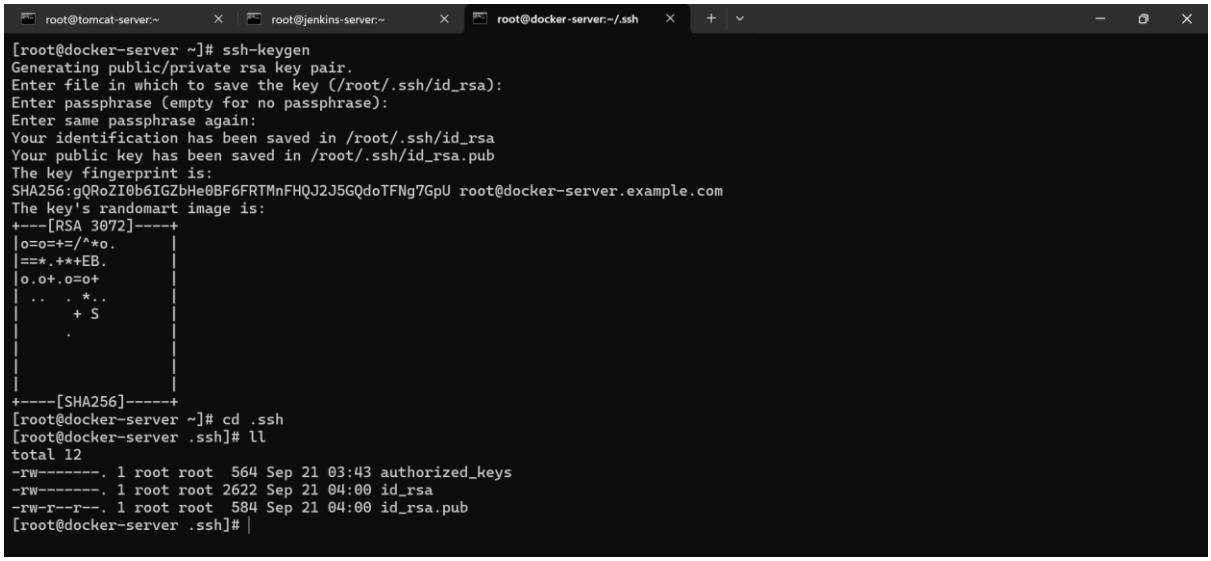


Now we have to set password to the docker-instance and give password related permissions in sshd_config file.

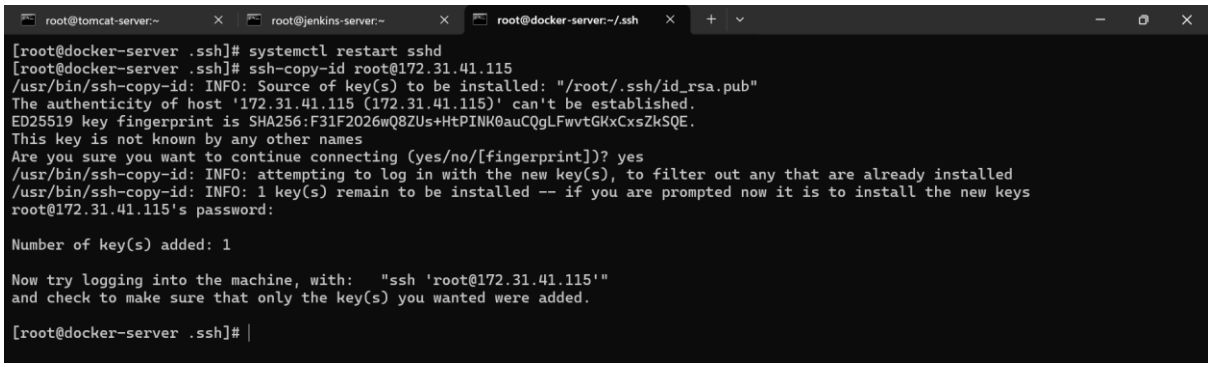


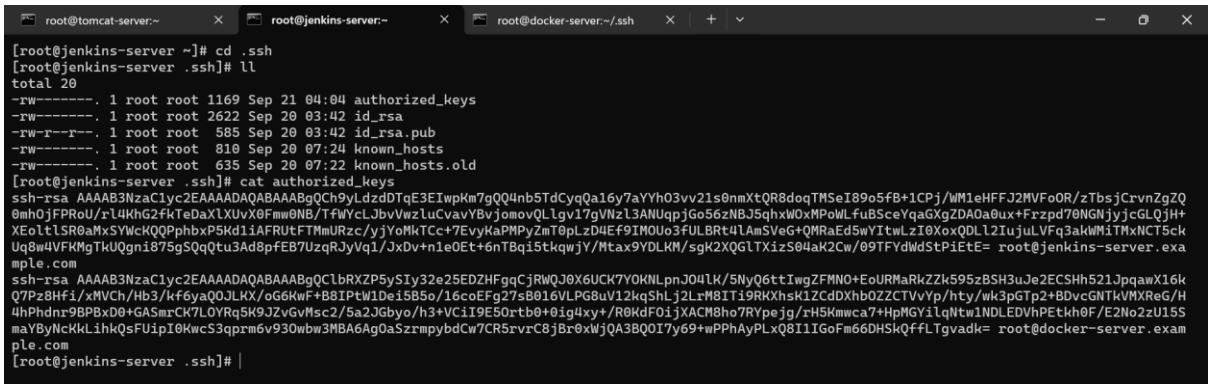


Now we have to created ssh key in docker-instance.

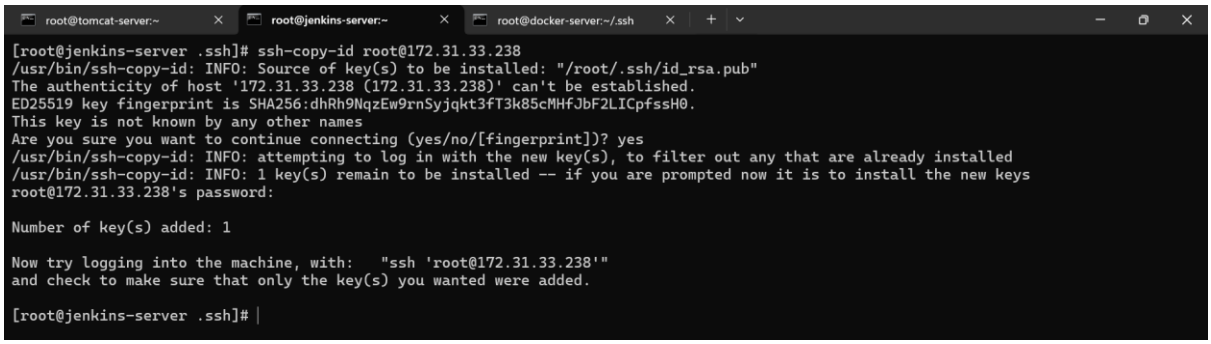


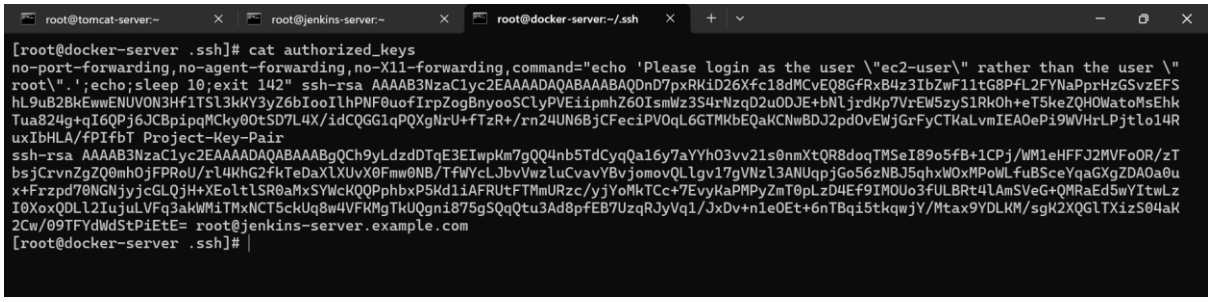
Now we need to transfer this docker ssh-key in jenkins-server.



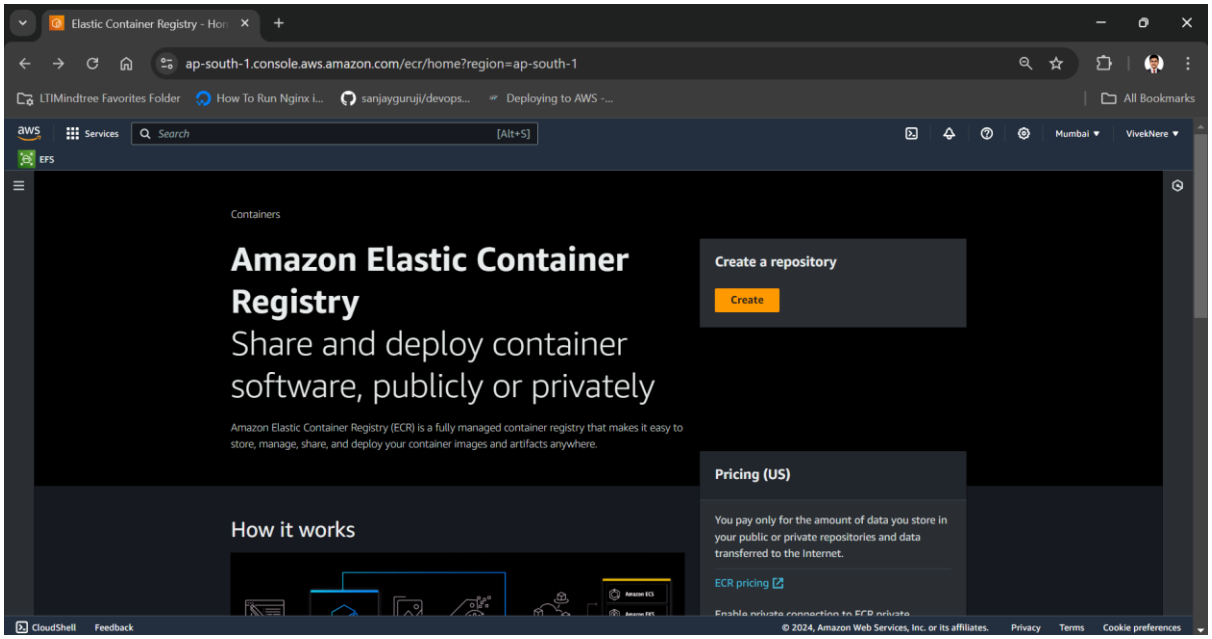


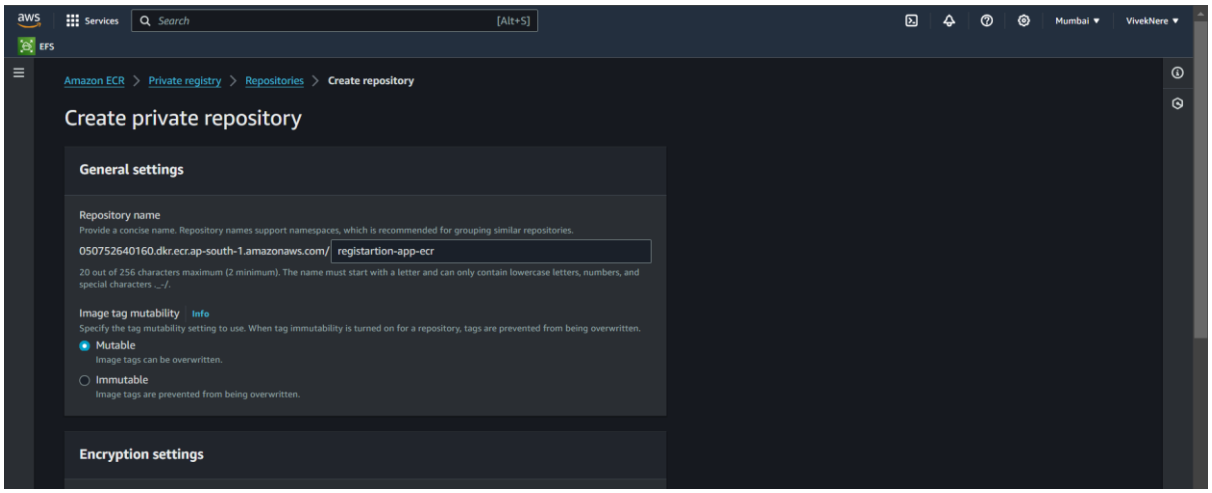
pass the ssh-key of jenkins-server in docker-server.

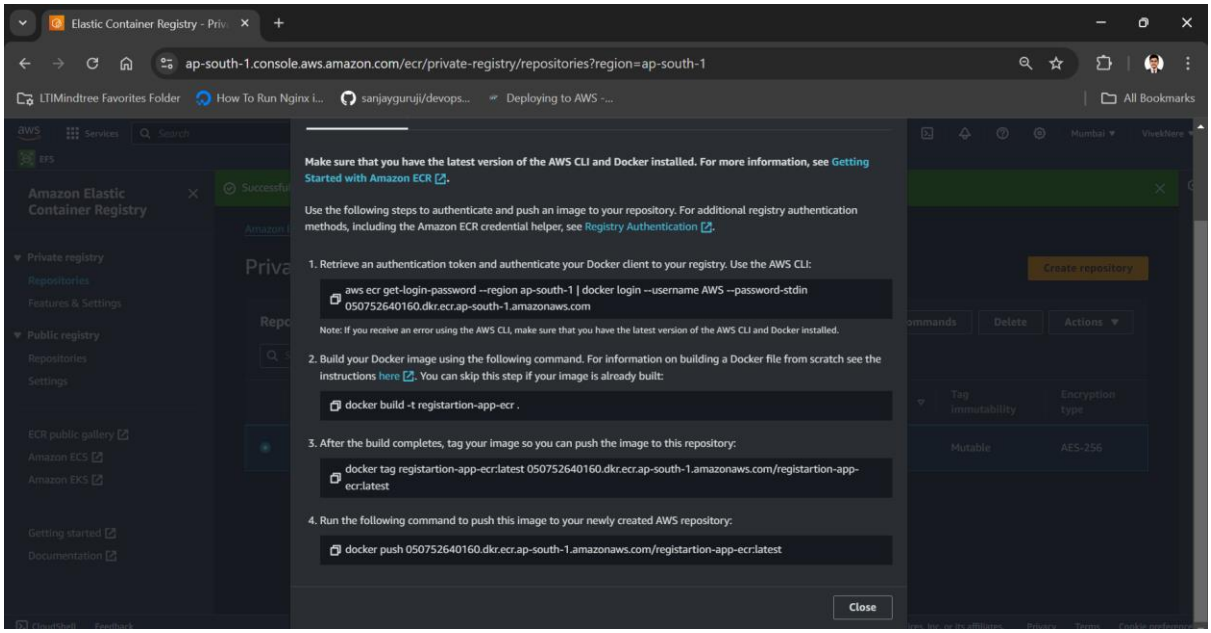




Now we have to create one ECR container to store our docker image.



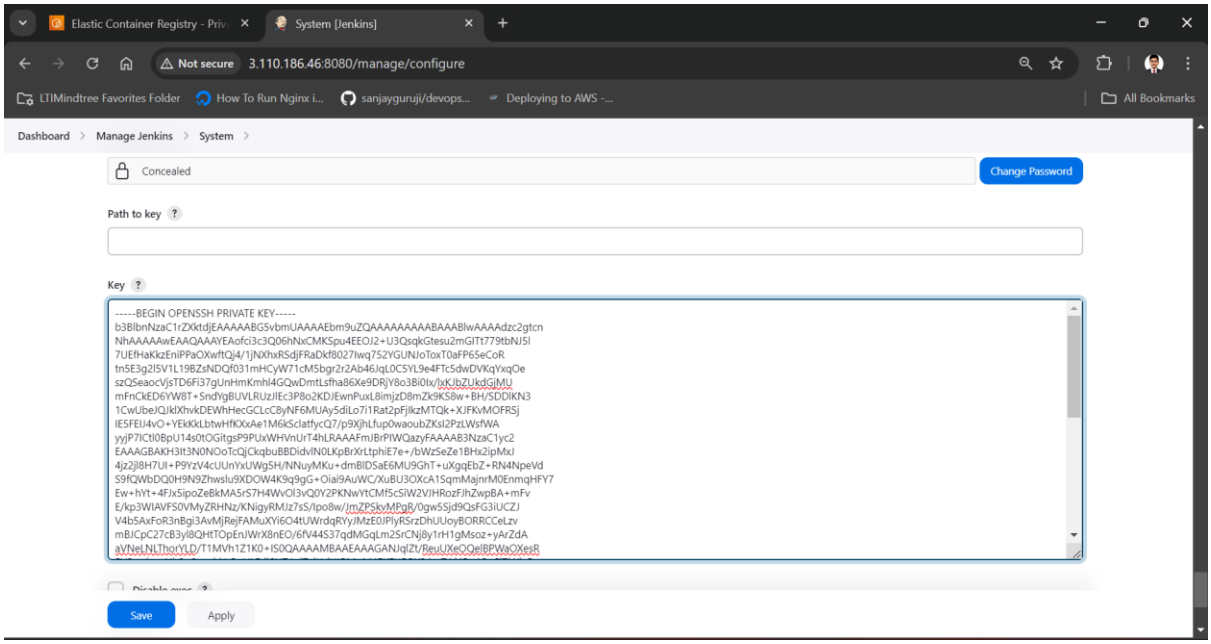




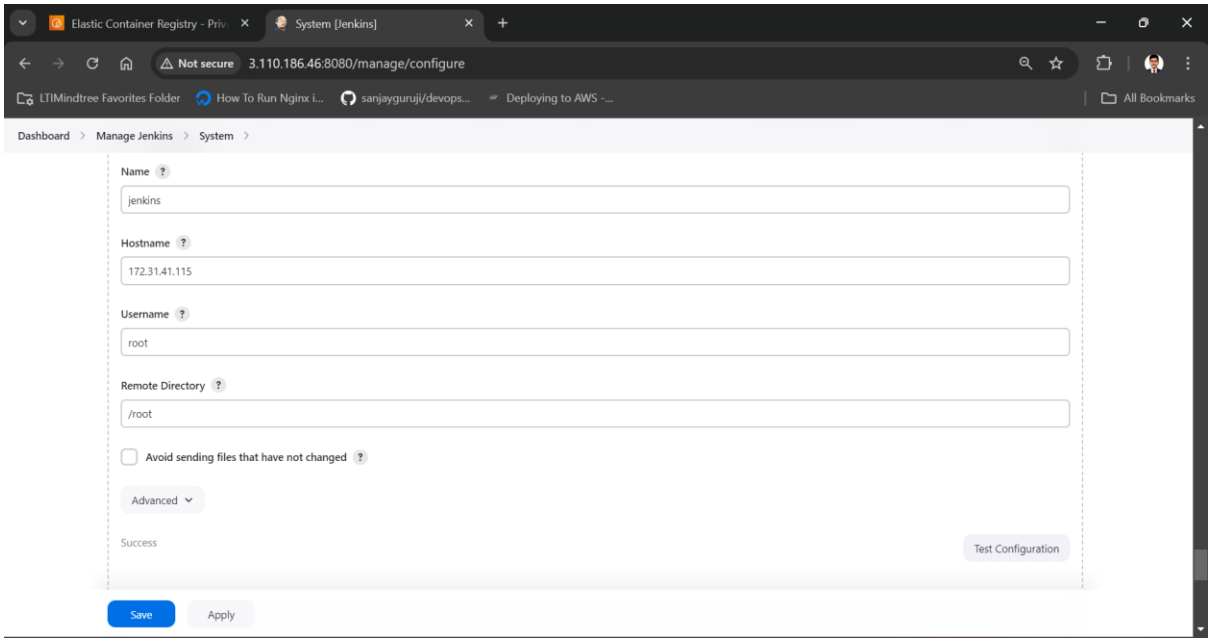
Now we have to configure our docker in jenkins for that we have make some changes in jenkins system.

Dashboard > Manage jenkins > System

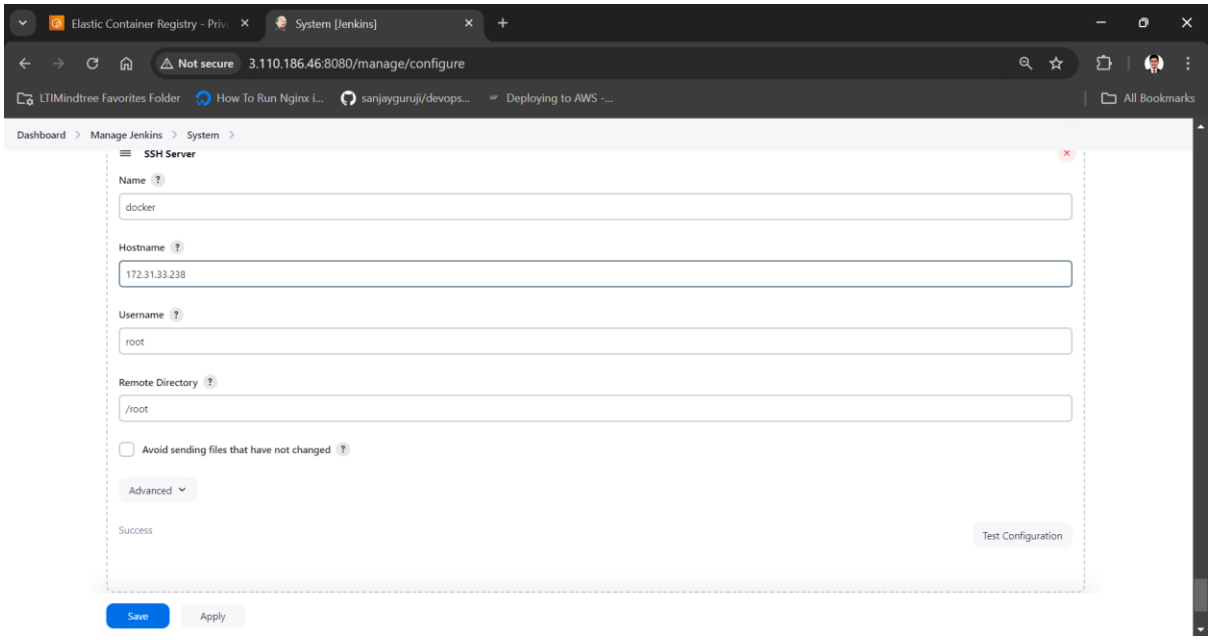
In keys we have to paste the private key of jenkins



Add jenkins ssh server with hostname as ip and root username ans path as /root



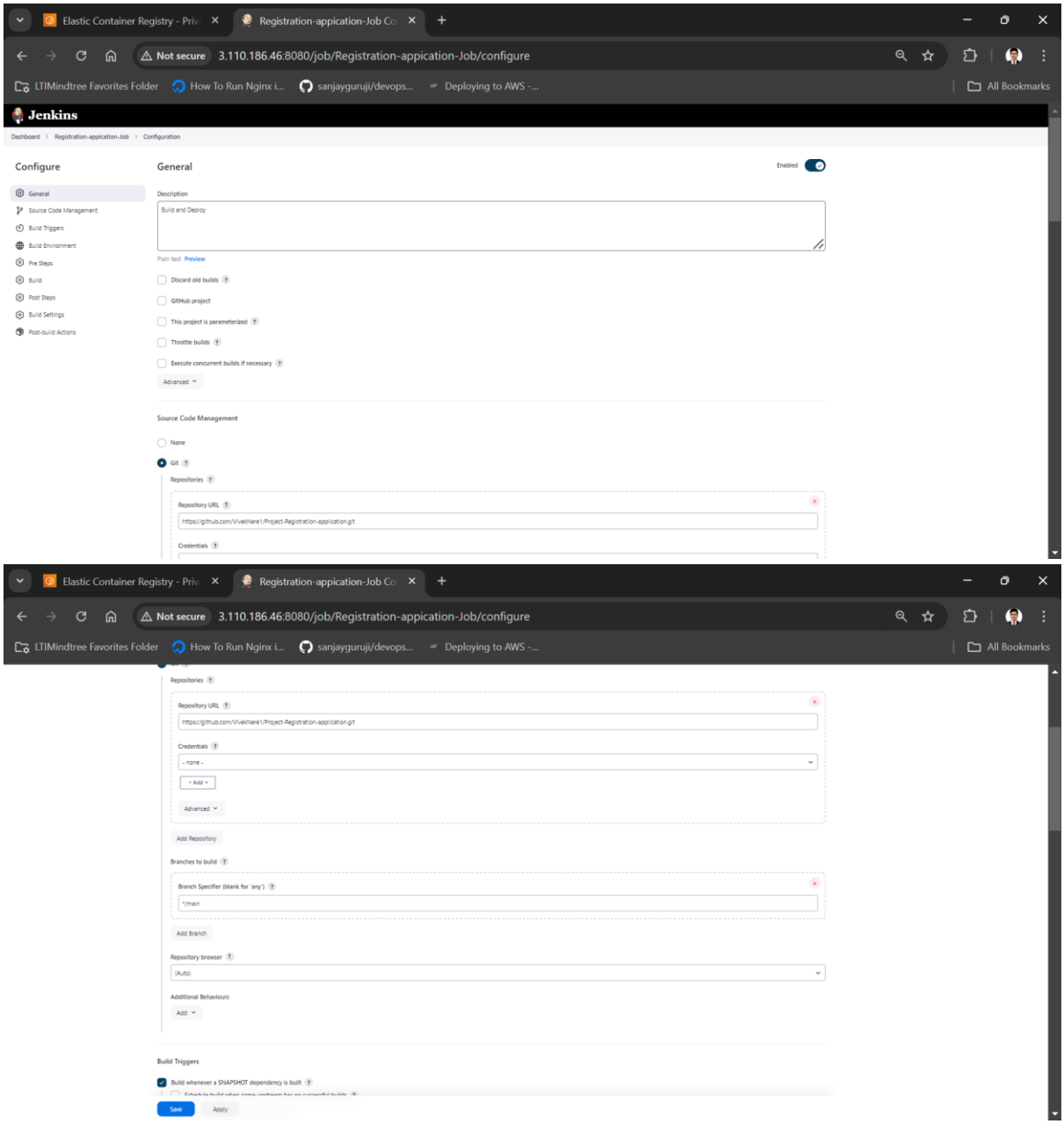
Add docker ssh server with hostname as ip and root username ans path as /root



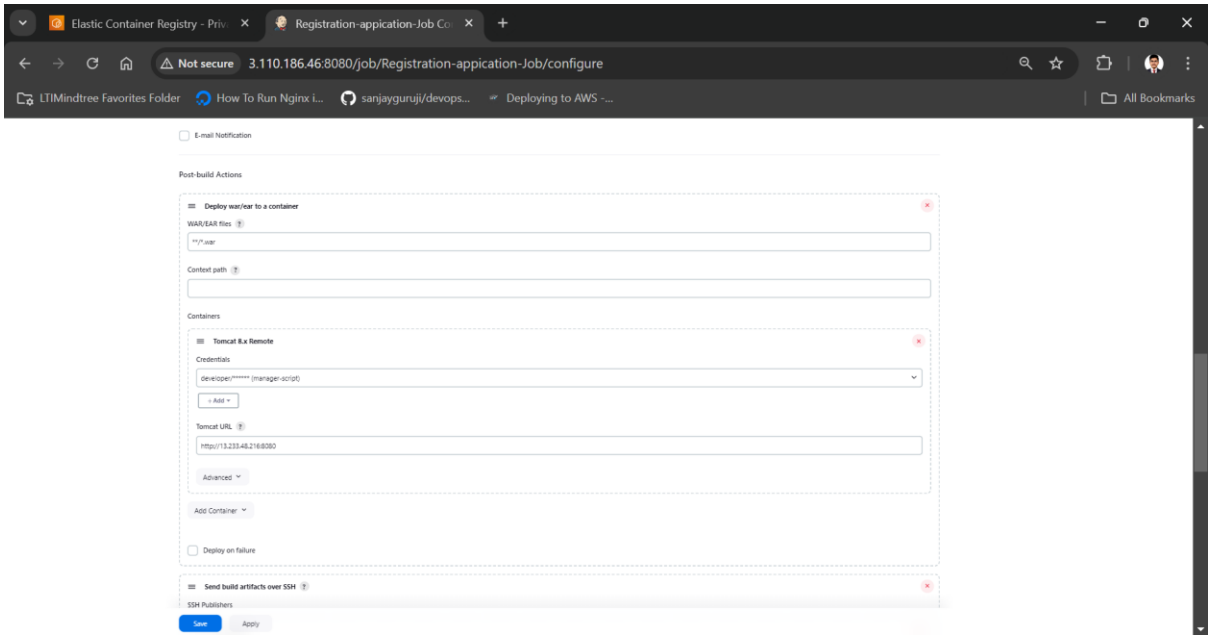
Apply and save.

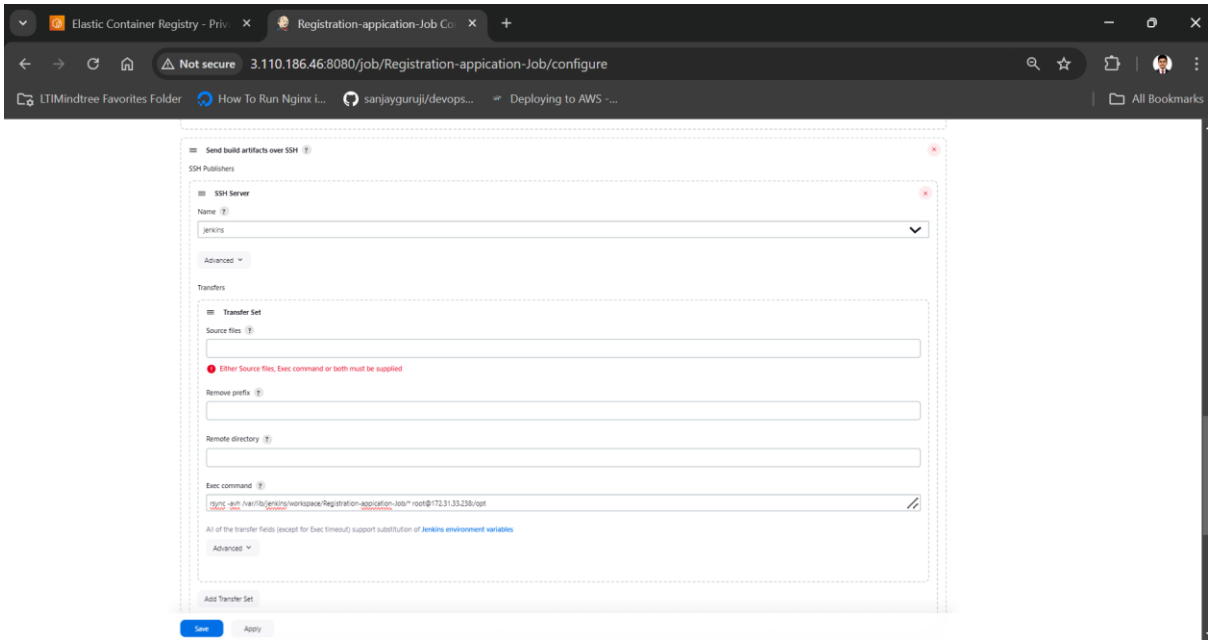
Create job name as Registration-application-job.

Now select our job in jenkins and in ‘Add post-build action’ add ‘Send build artifacts over SSH’

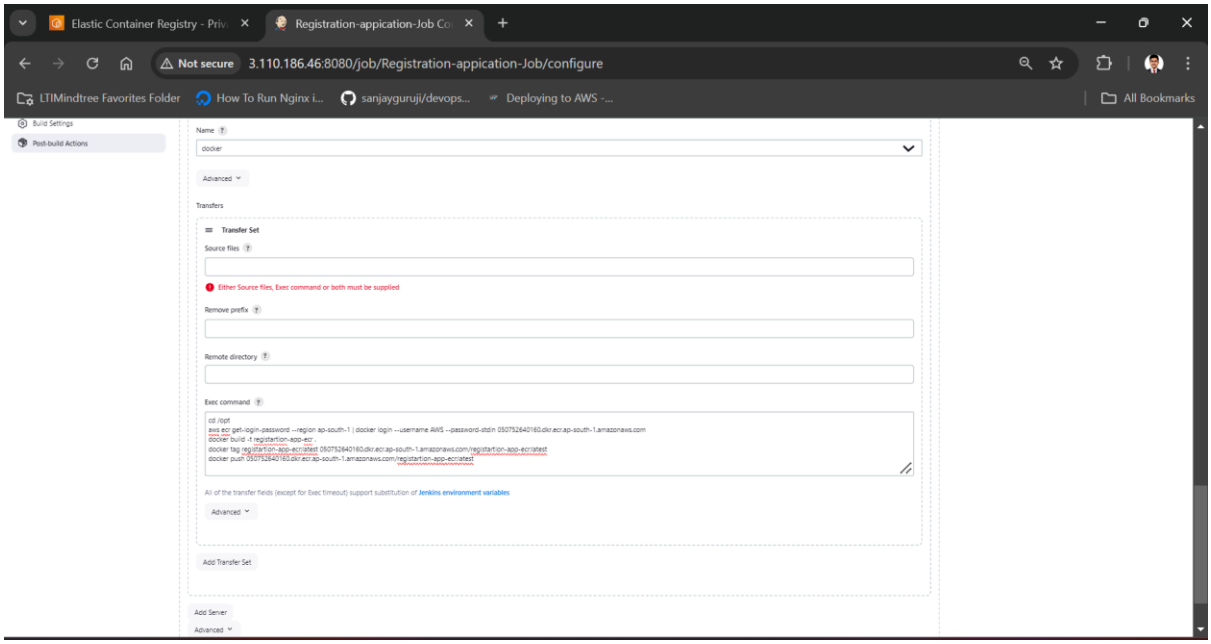

Elastic Container Registry - Priv
Registration-appication-Job Co
Not secure
3.110.186.46:8080/job/Registration-appication-Job/configure
LTIMindtree Favorites Folder
How To Run Nginx i...
sanjayguruji/devops...
Deploying to AWS -...
All Bookmarks
Jenkins
Dashboard
Registration-appication-Job
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Pre Steps
Build
Post Steps
Build Settings
Post-build Actions
General
Enabled
Description
Build and Deploy
Plain text
Preview
Discard old builds
GitHub project
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Advanced
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/VivekNere1/Project-Registration-application.git
Credentials

Elastic Container Registry - Priv
Registration-appication-Job Co
Not secure
3.110.186.46:8080/job/Registration-appication-Job/configure
LTIMindtree Favorites Folder
How To Run Nginx i...
sanjayguruji/devops...
Deploying to AWS -...
All Bookmarks
Repositories
Repository URL
https://github.com/VivekNere1/Project-Registration-application.git
Credentials
- none -
+ Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add
Build Triggers
Build whenever a SNAPSHOT dependency is built
Save
Apply

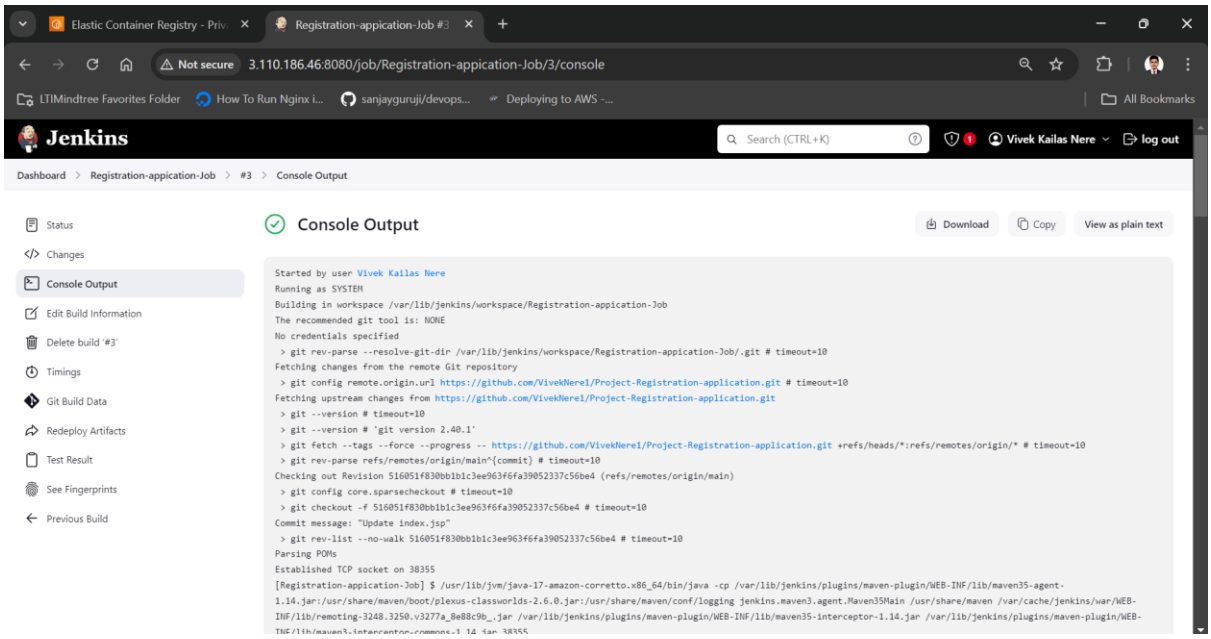


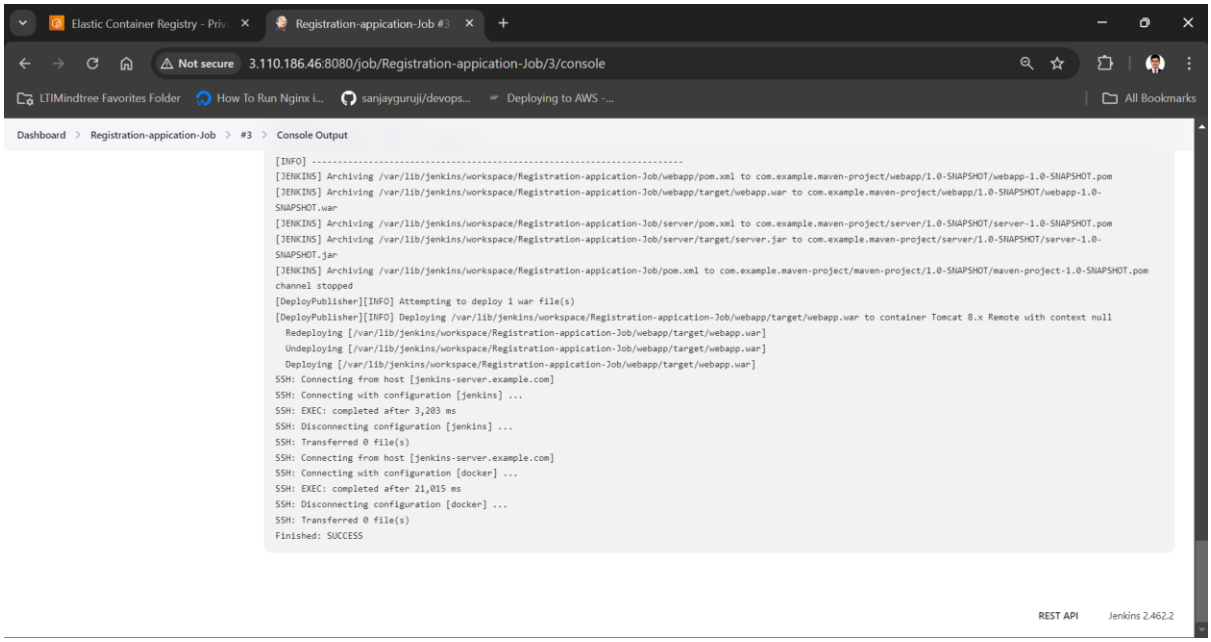


Created by: Vivek Nere



Appy and save.

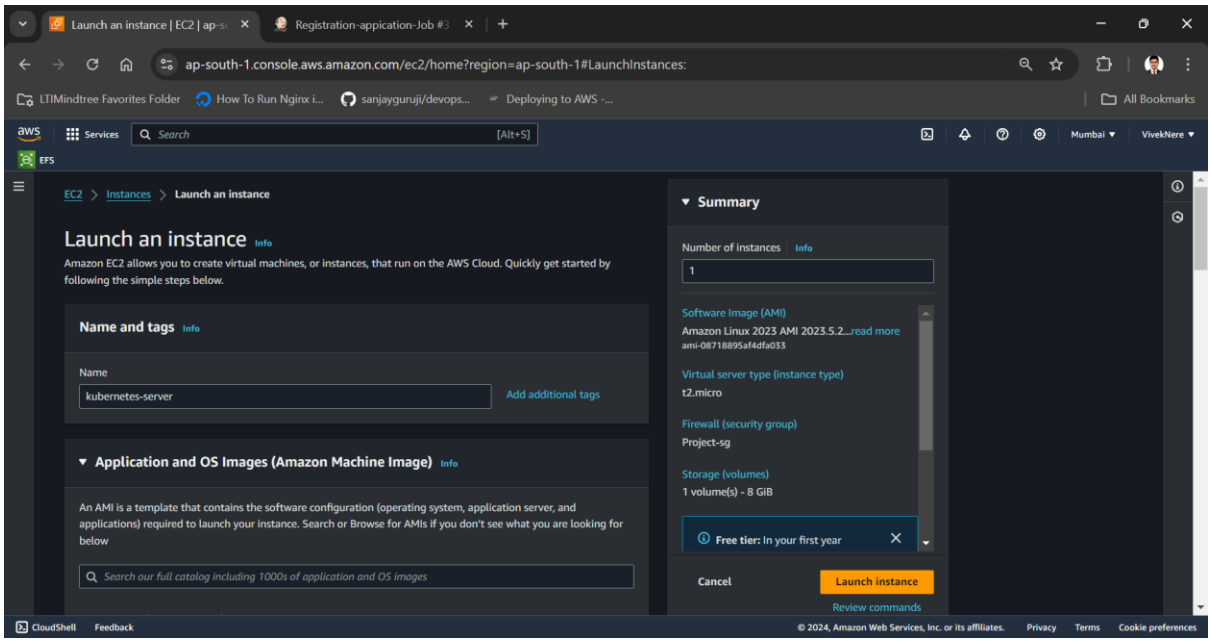


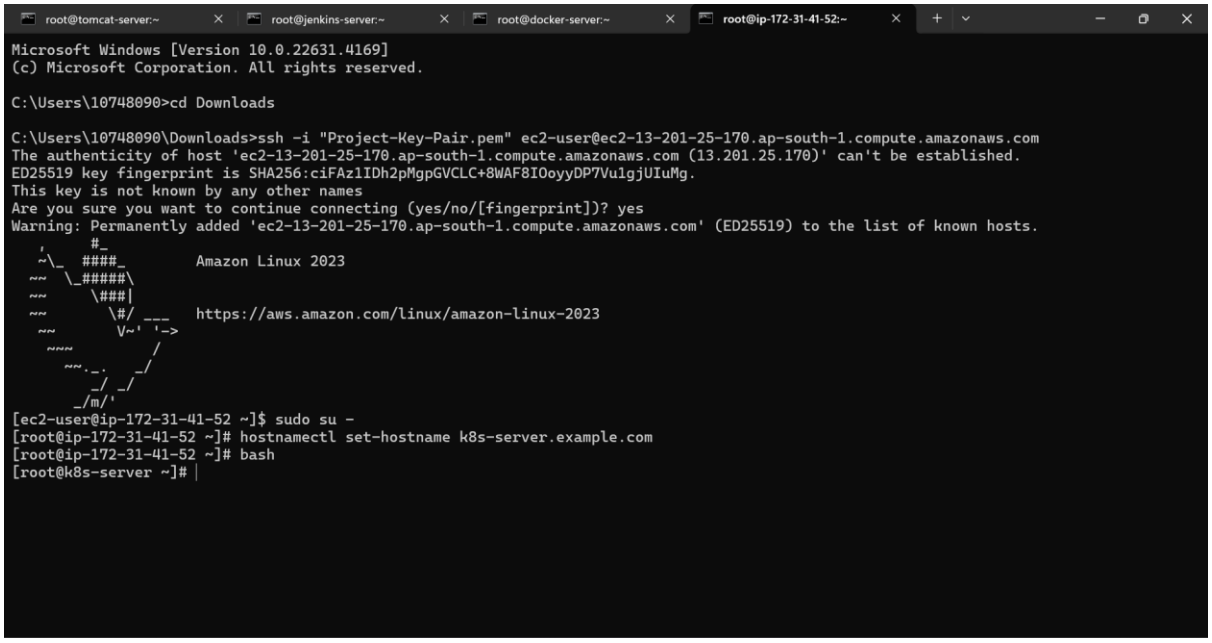


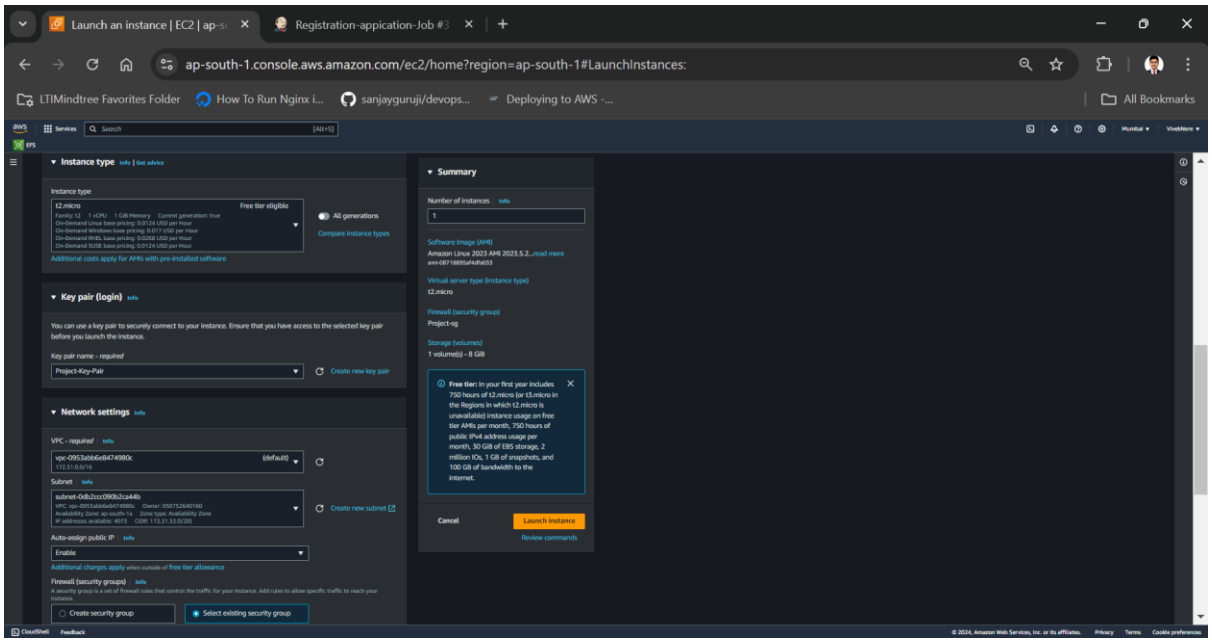
From above we can see our docker build get successful!

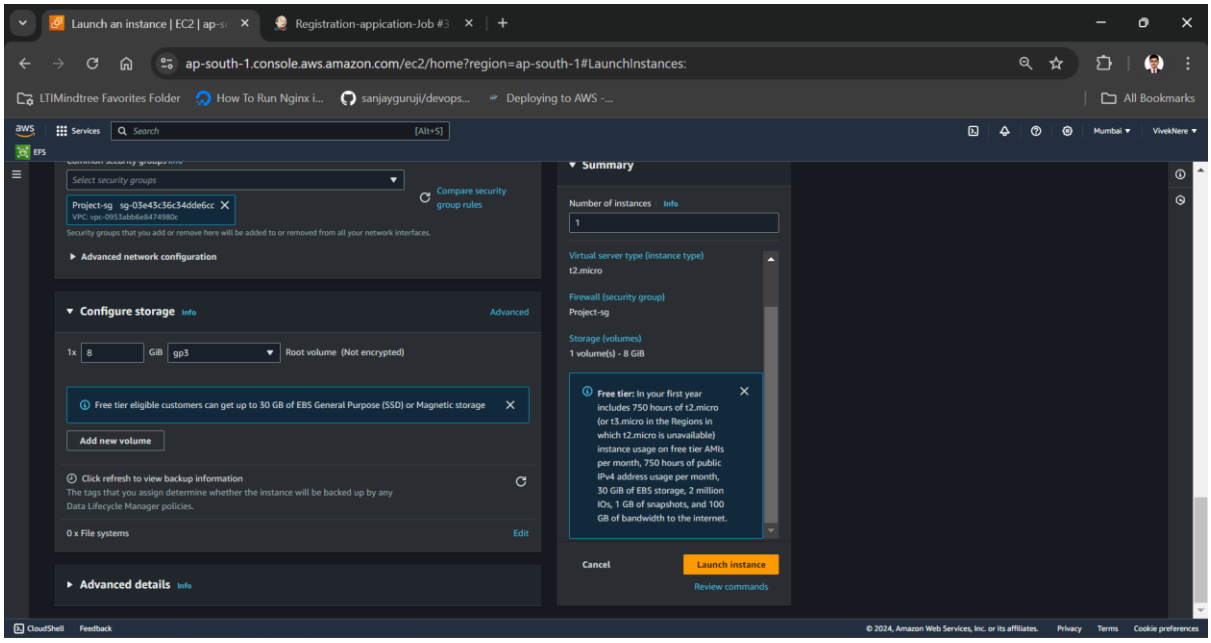
#5: Kubernetes EKS

kubernetes Installation and Configuration with jenkins:
First create an instance for k8s using amazon linux as OS with instance type t2.micro and same key-pair and security group which we have use in previous instances.

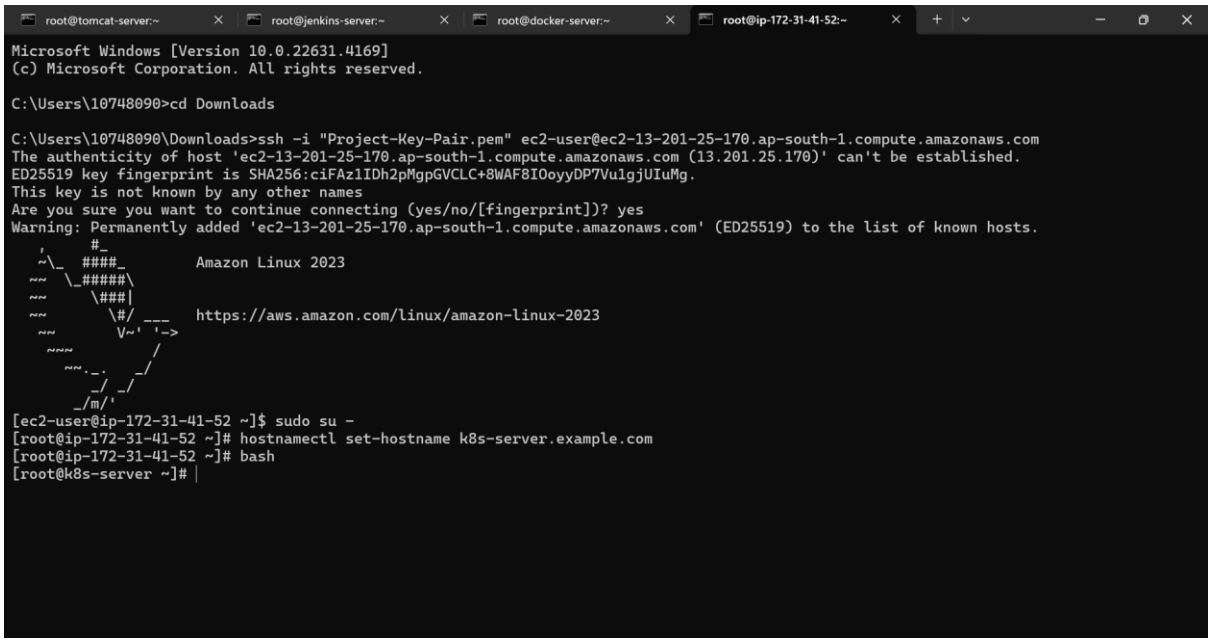



root@tomcat-server:~
root@jenkins-server:~
root@docker-server:~
root@ip-172-31-41-52:~
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.
C:\Users\10748090>cd Downloads
C:\Users\10748090\Downloads>ssh -i "Project-Key-Pair.pem" ec2-user@ec2-13-201-25-170.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com (13.201.25.170)' can't be established.
ED25519 key fingerprint is SHA256:ciFAz1IDh2pMgpGVCLC+8WAF8IOoyyDP7Vu1gjUIuMg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-201-25-170.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-41-52 ~]$ sudo su -
[root@ip-172-31-41-52 ~]# hostnamectl set-hostname k8s-server.example.com
[root@ip-172-31-41-52 ~]# bash
[root@k8s-server ~]#


Launch an instance | EC2 | ap-s
Registration-appication-Job #3
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
LTIMindtree Favorites Folder
How To Run Nginx i...
sanjayguruji/devops...
Deploying to AWS -...
All Bookmarks
Instance type
Key pair (login)
Network settings
Summary
Launch instance
Cancel

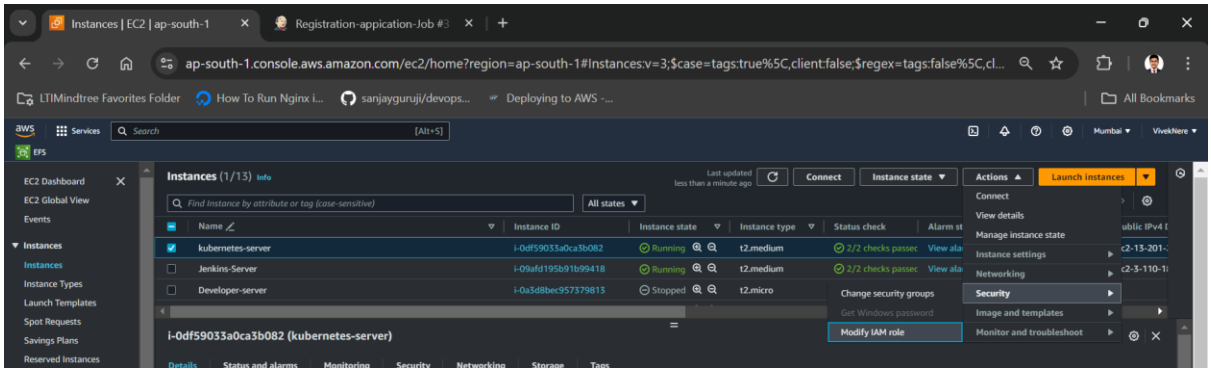


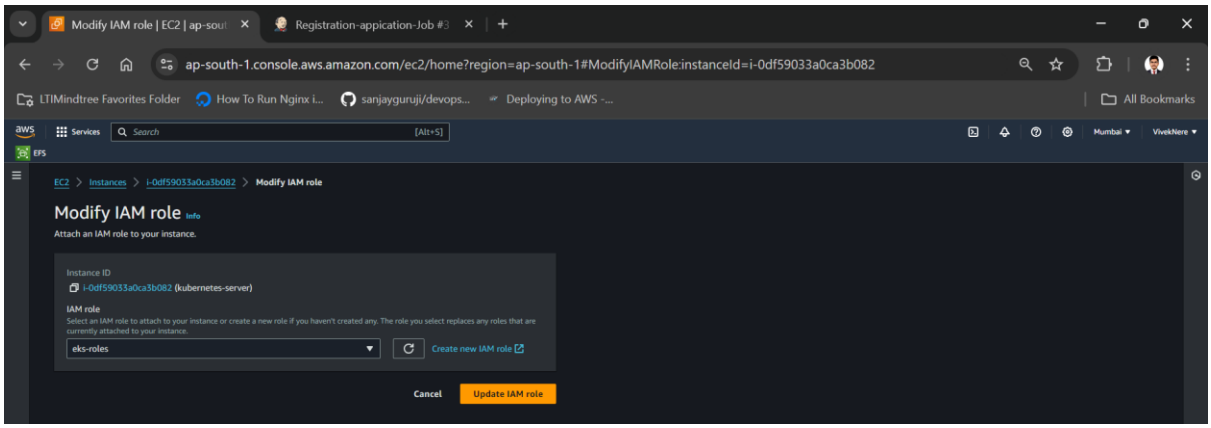
Now connect this kubernetes ec2 in terminal and give hostname as docker-server to it.



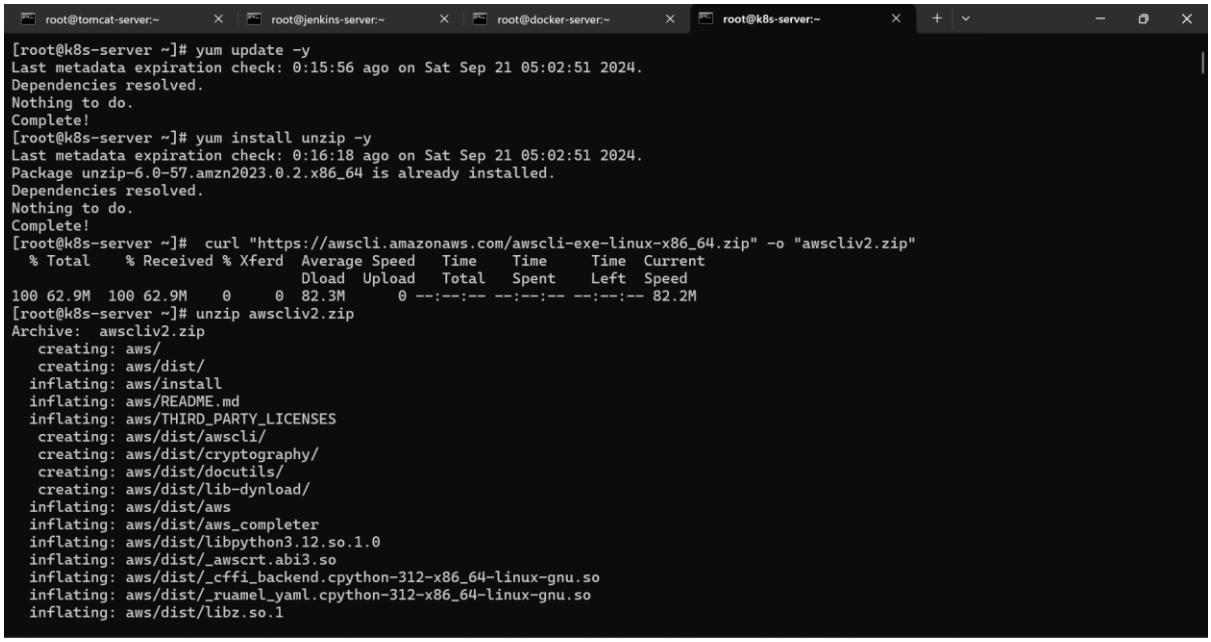
Create EKS cluster in kubernetes-server

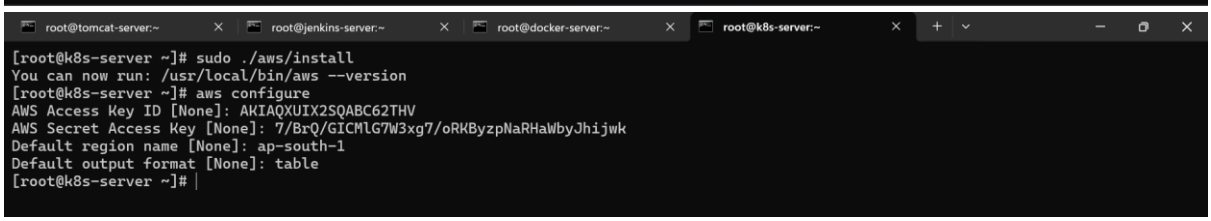
Create one IAM role with ECR fullaccess , EKSpolicy and IAM full access policy and attach that role to this kubernetes instance.



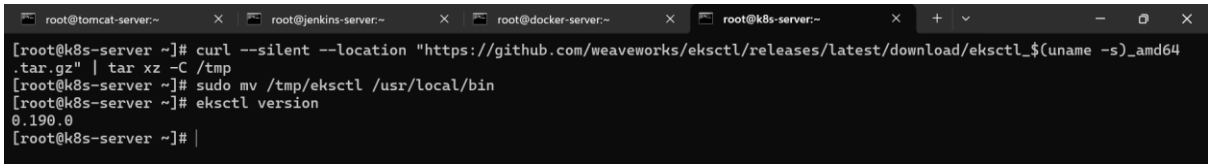


Install aws cli in k8s ec2

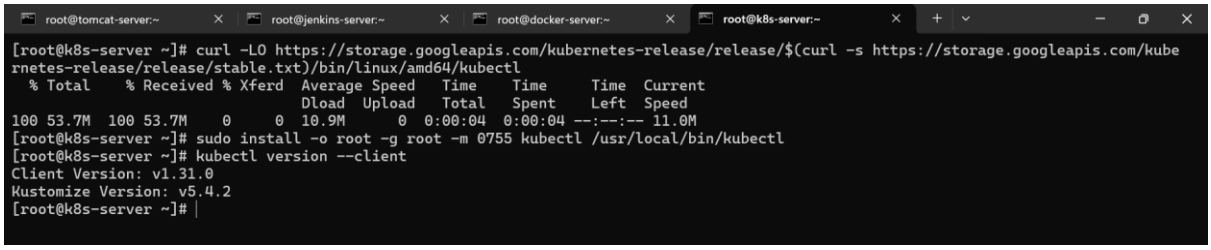




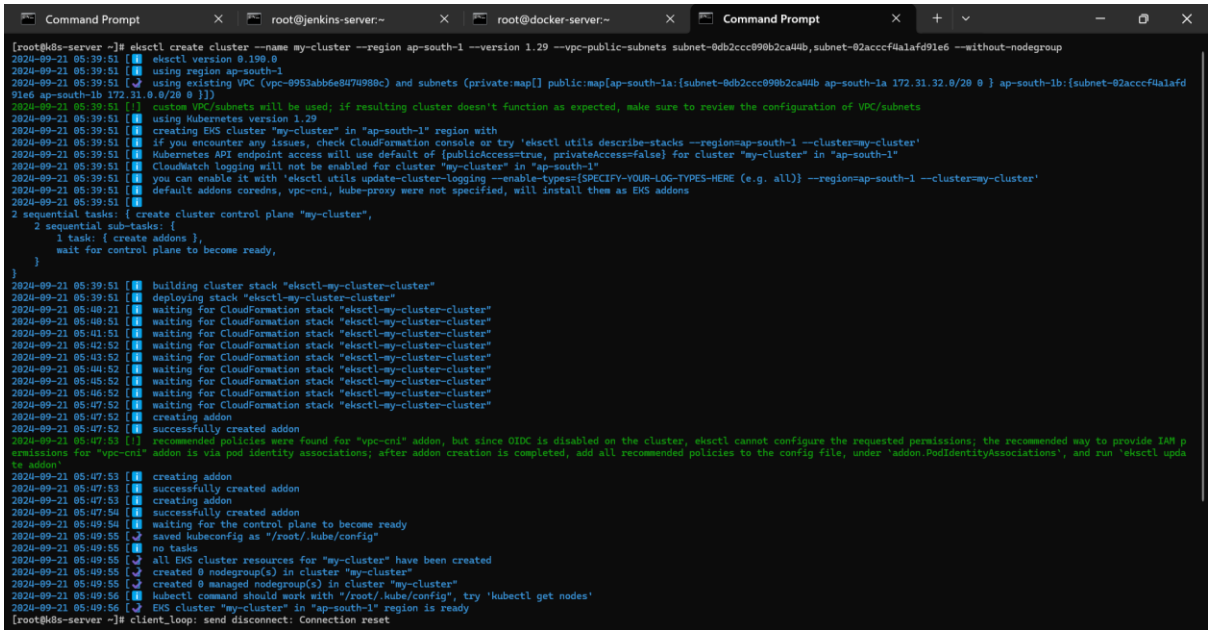
Install EKS tool:

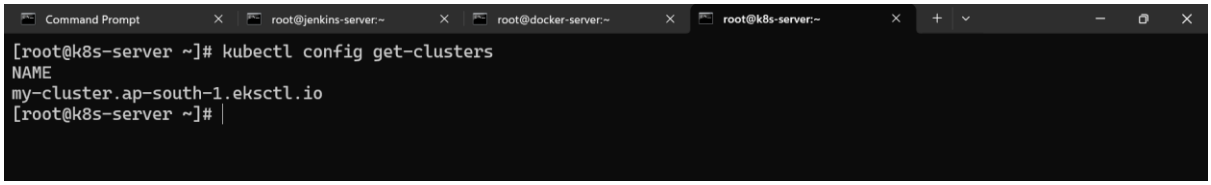


Install Kubectl:

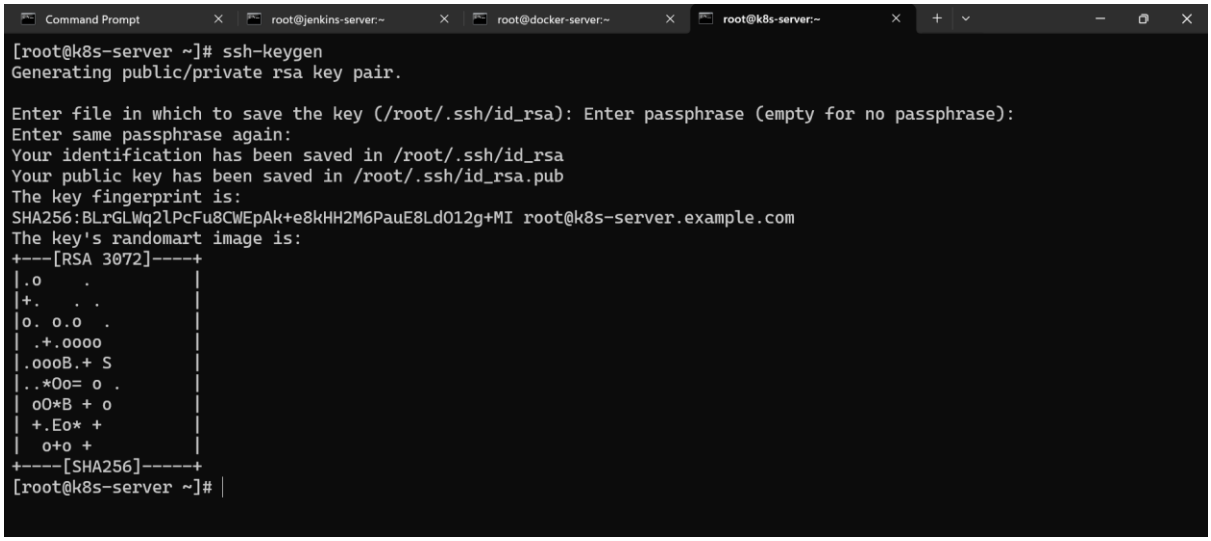


Now create cluster on EKS:

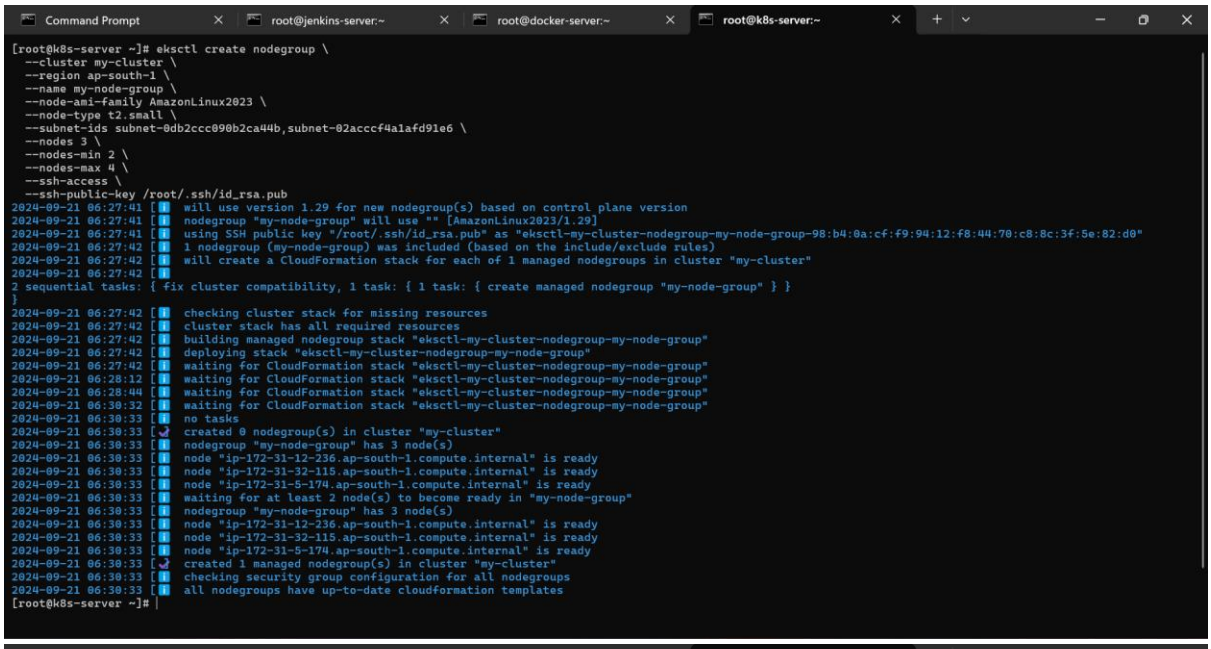


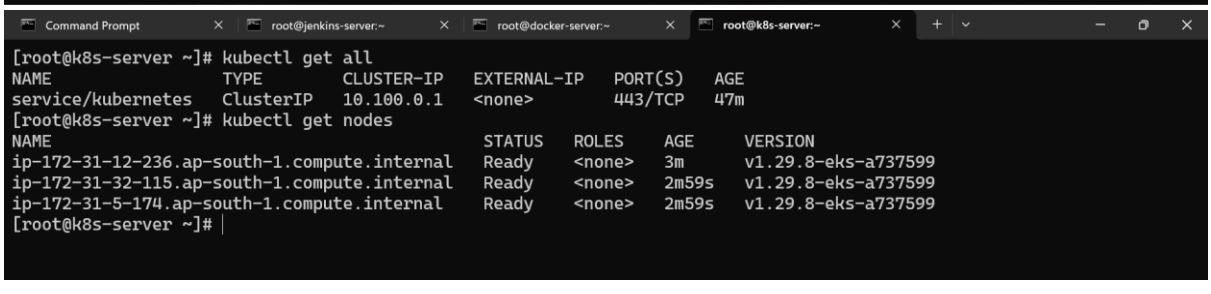


Create ssh-keygen in k8s ec2:

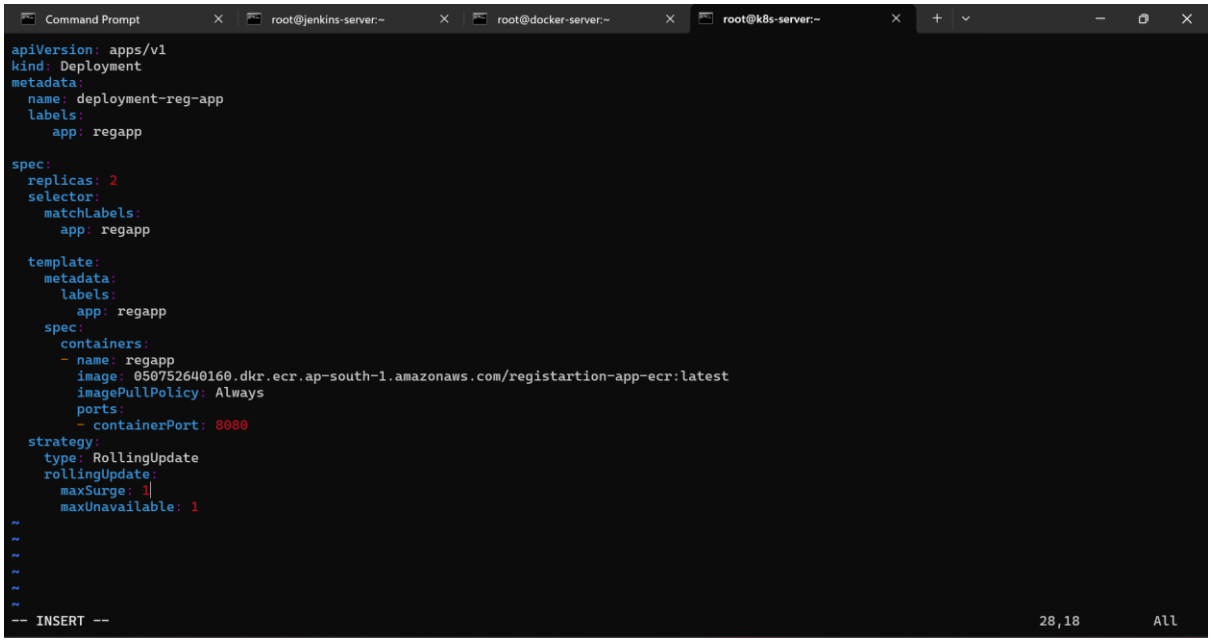


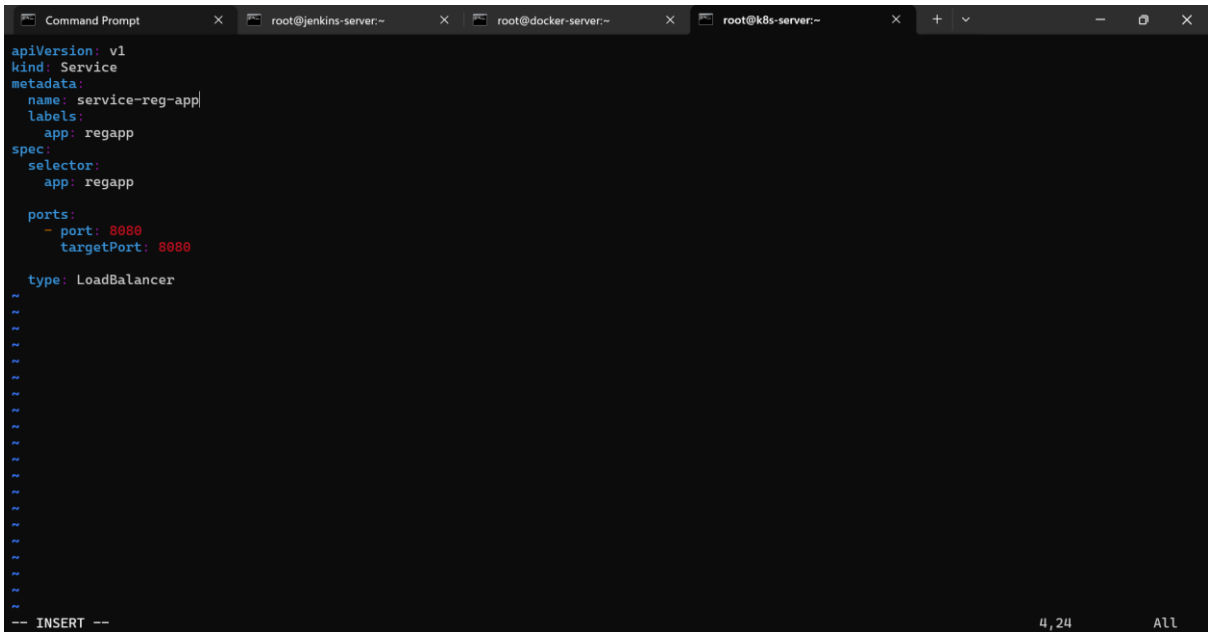
Now we have to create nodes in our cluster:



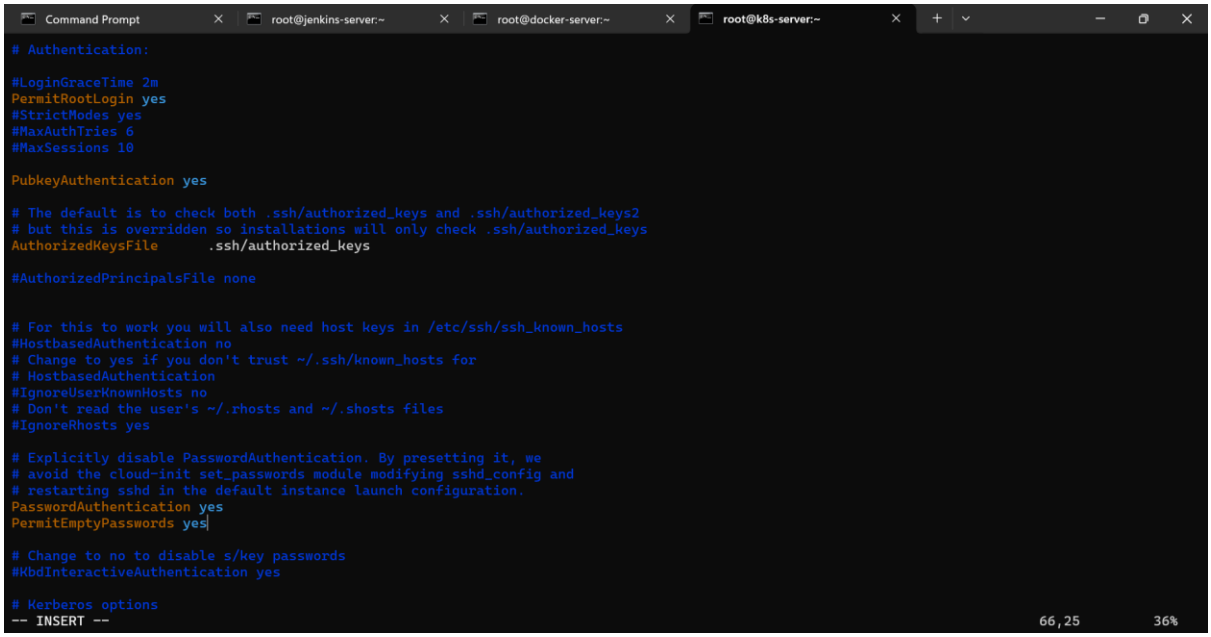


Now we have to create one deployment.yml file ans service.yml file

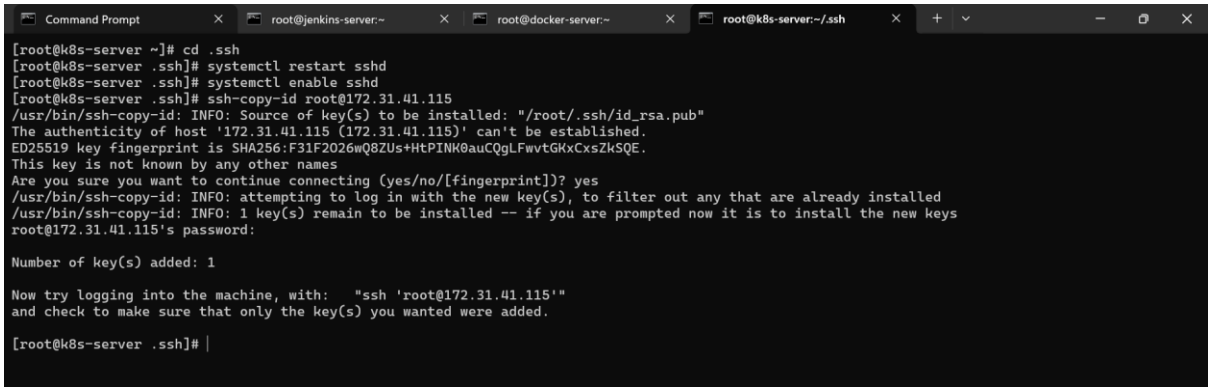


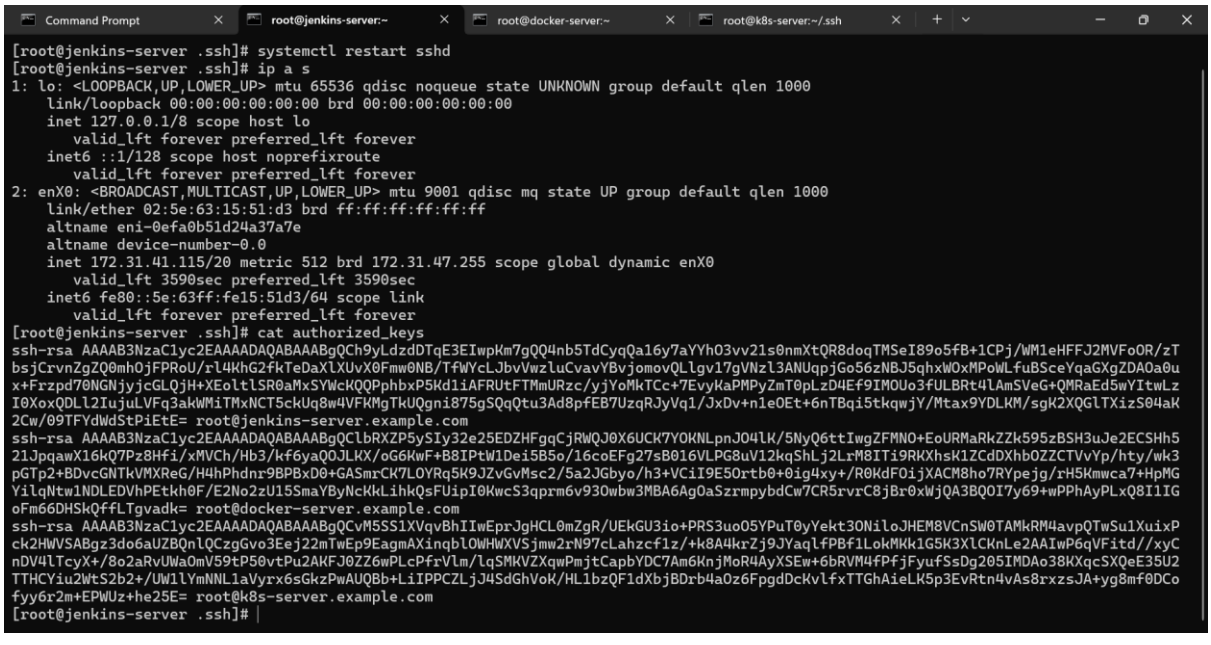


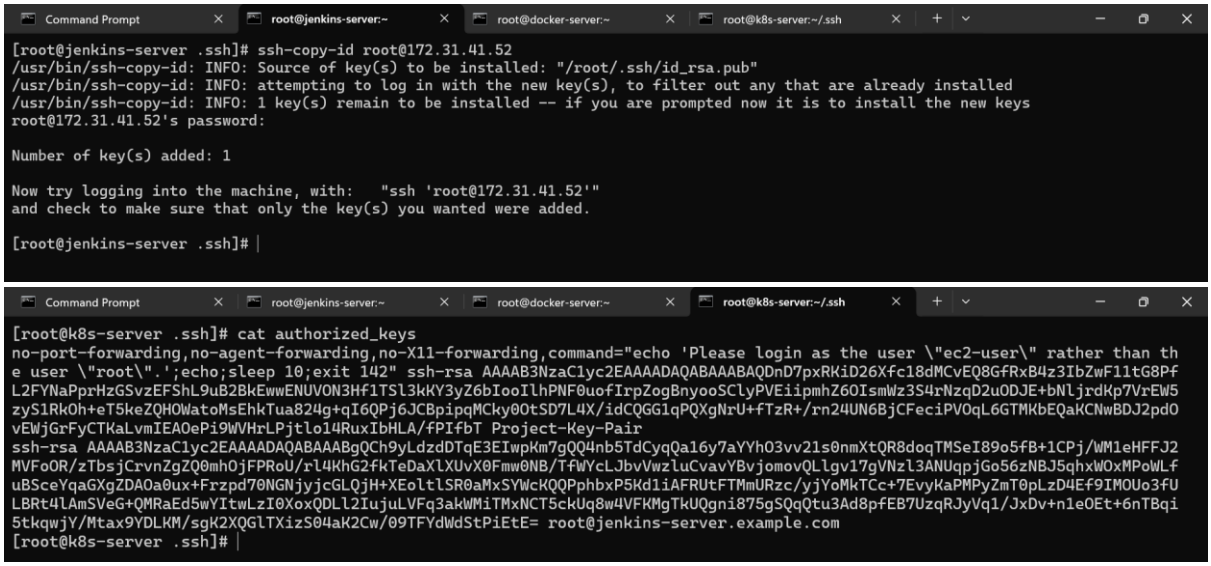
Give ssh-key and password related permission in sshd_config file of kubernetes-server.



Now we need to transfer this kubernetes ssh-key in jenkins-server And jenkins ssh-key in kubernetes-server

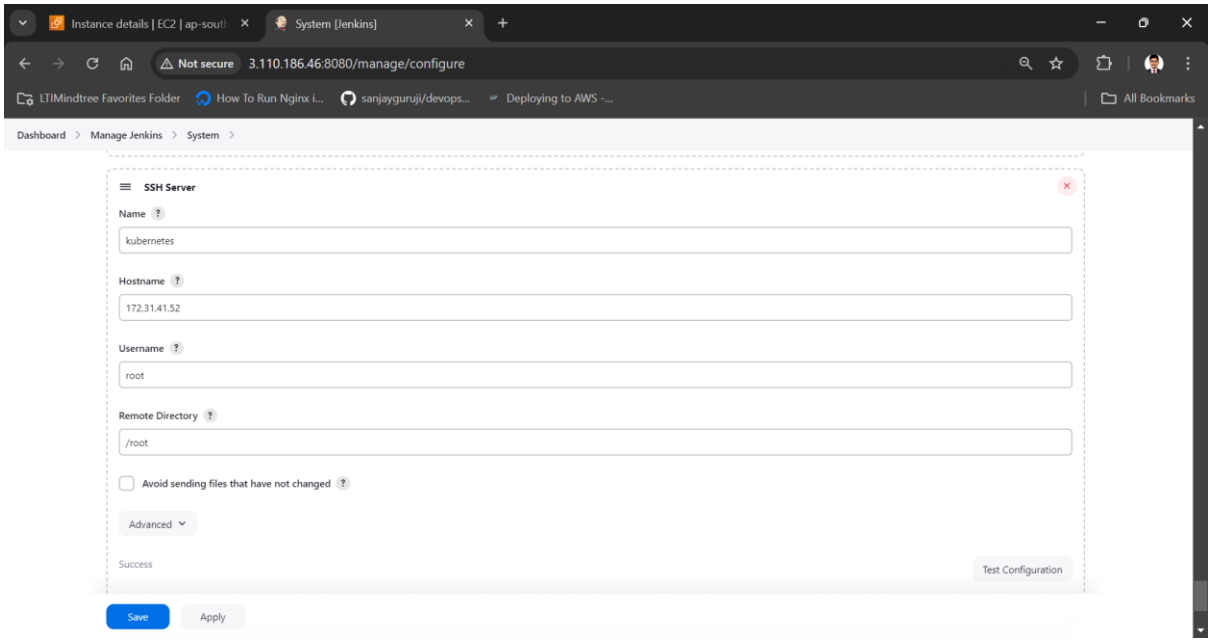




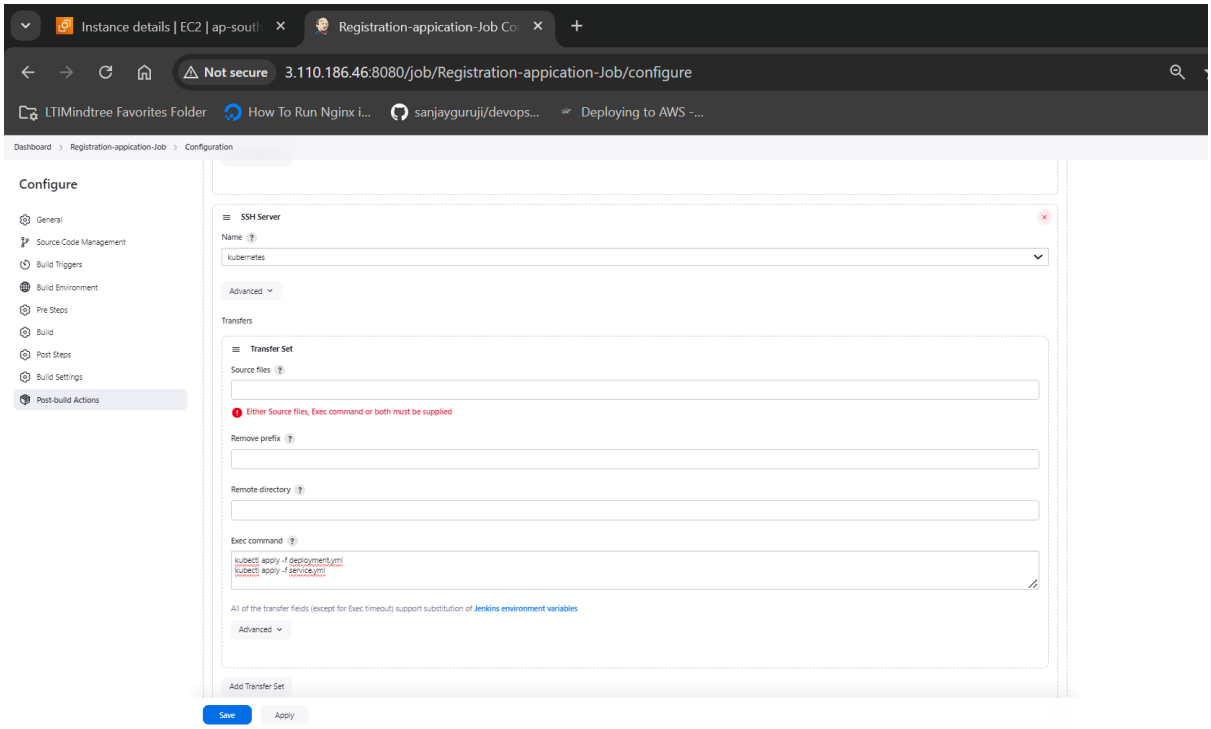


Now we have to configure our kubernetes in jenkins for that we have make some changes in jenkins system.

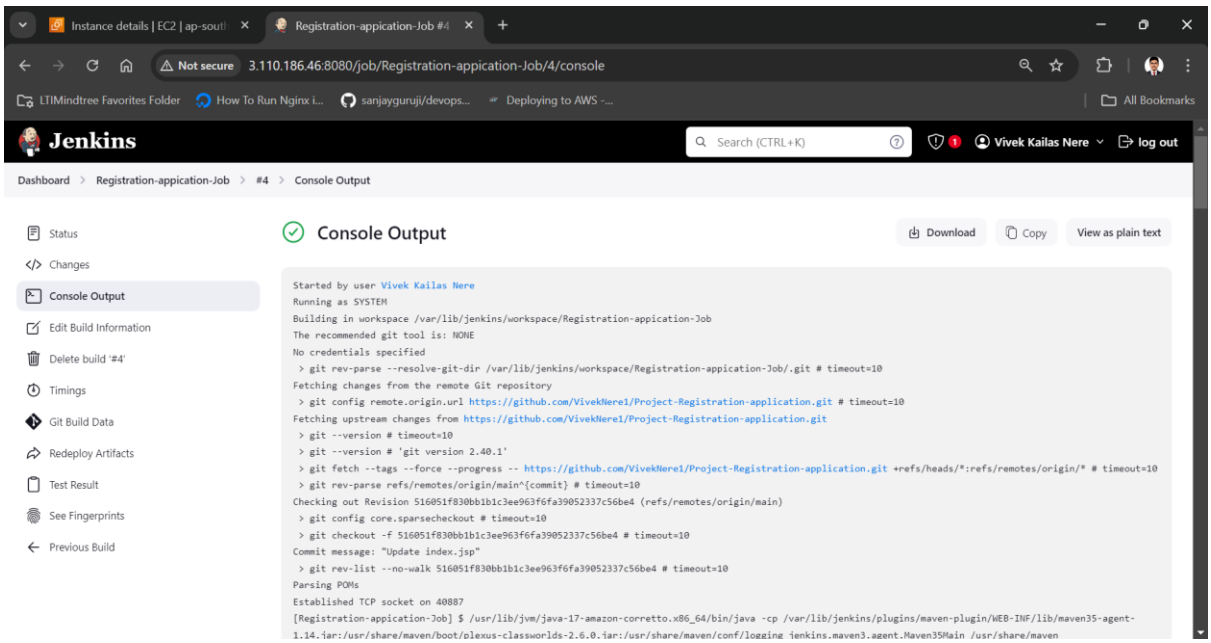
Dashboard > Manage jenkins > System

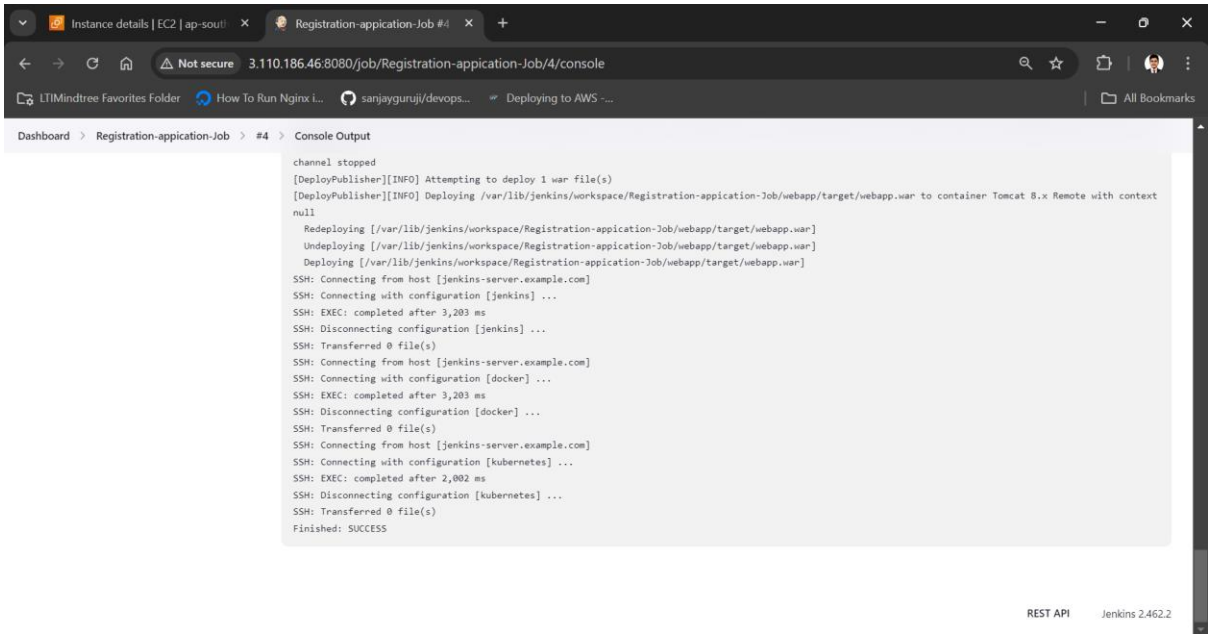


Now in our job **Registration-appication-Job** we have to add ssh-server with exec commands of kubernetes to deploy and create service by Jenkins build.



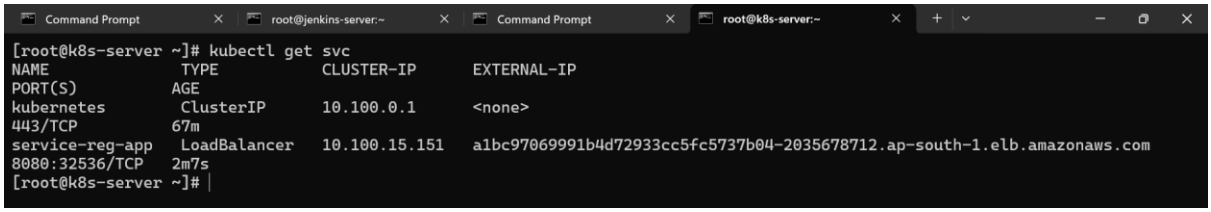
Apply save and build now.





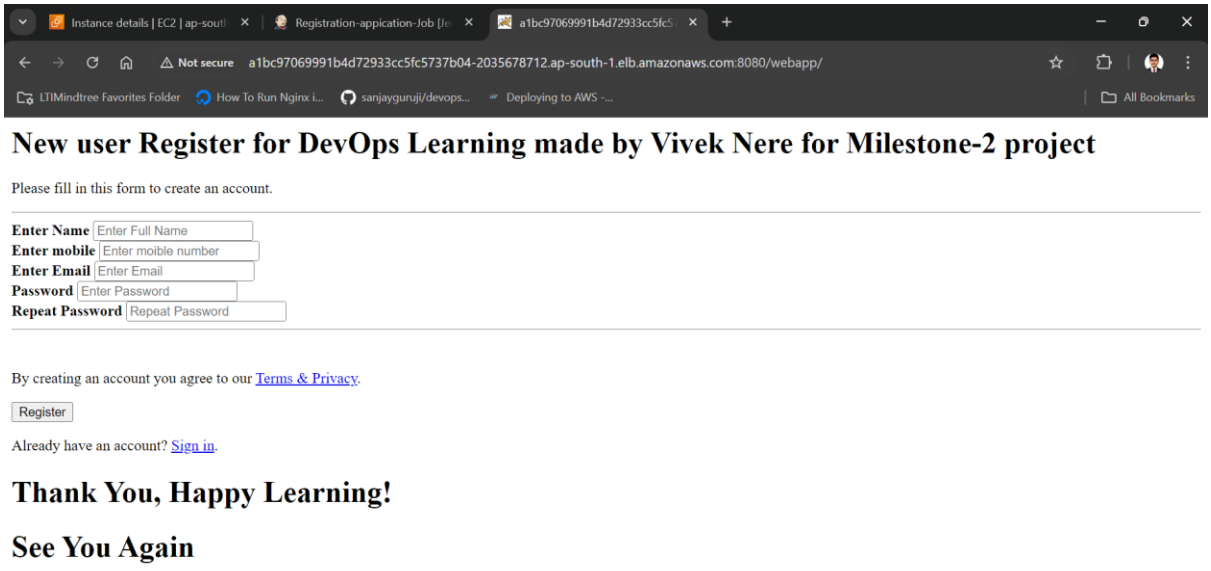
Our Kubernetes configuration with Jenkins build get successful!

Now we have to check in our Kubernetes-server whether service get created or not.

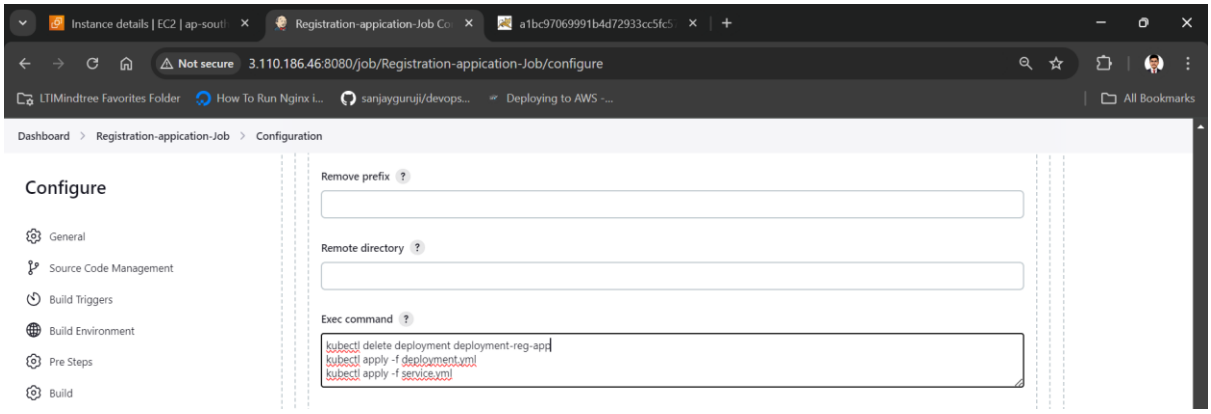


Our service get deployed successfully.

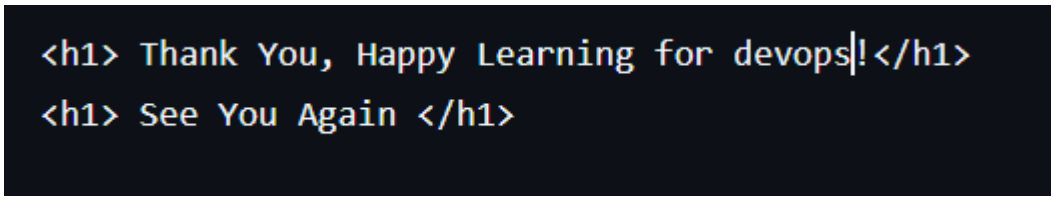
Now we have to check on chrome.



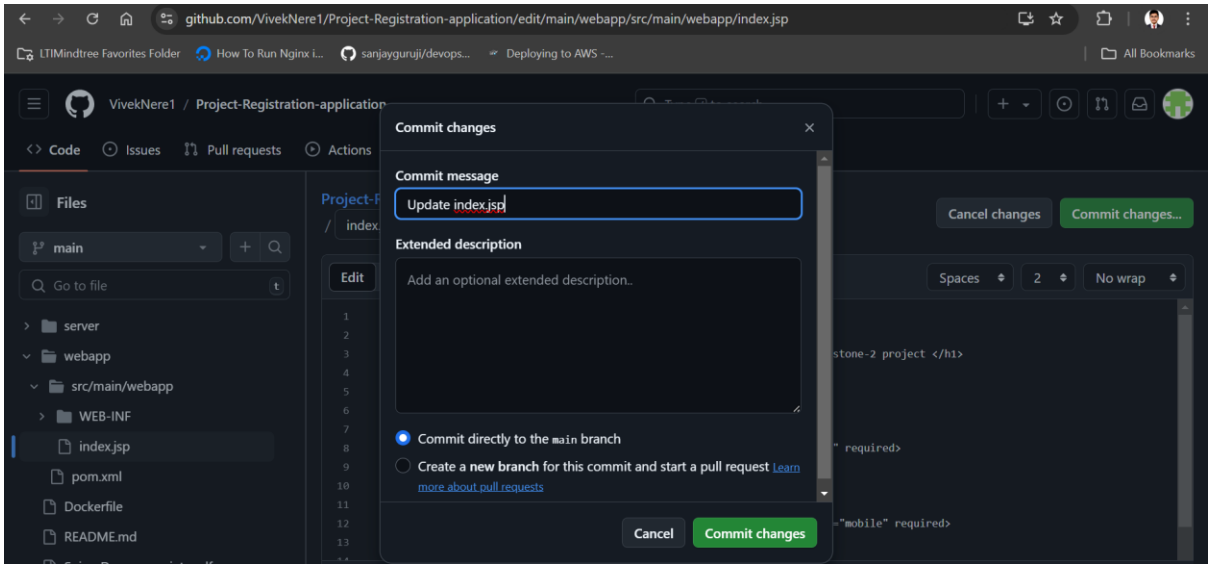
On every change in git we need to delete our old deployment and add new one so for that we have to add one more command in Kubernetes ssh-server exec commands

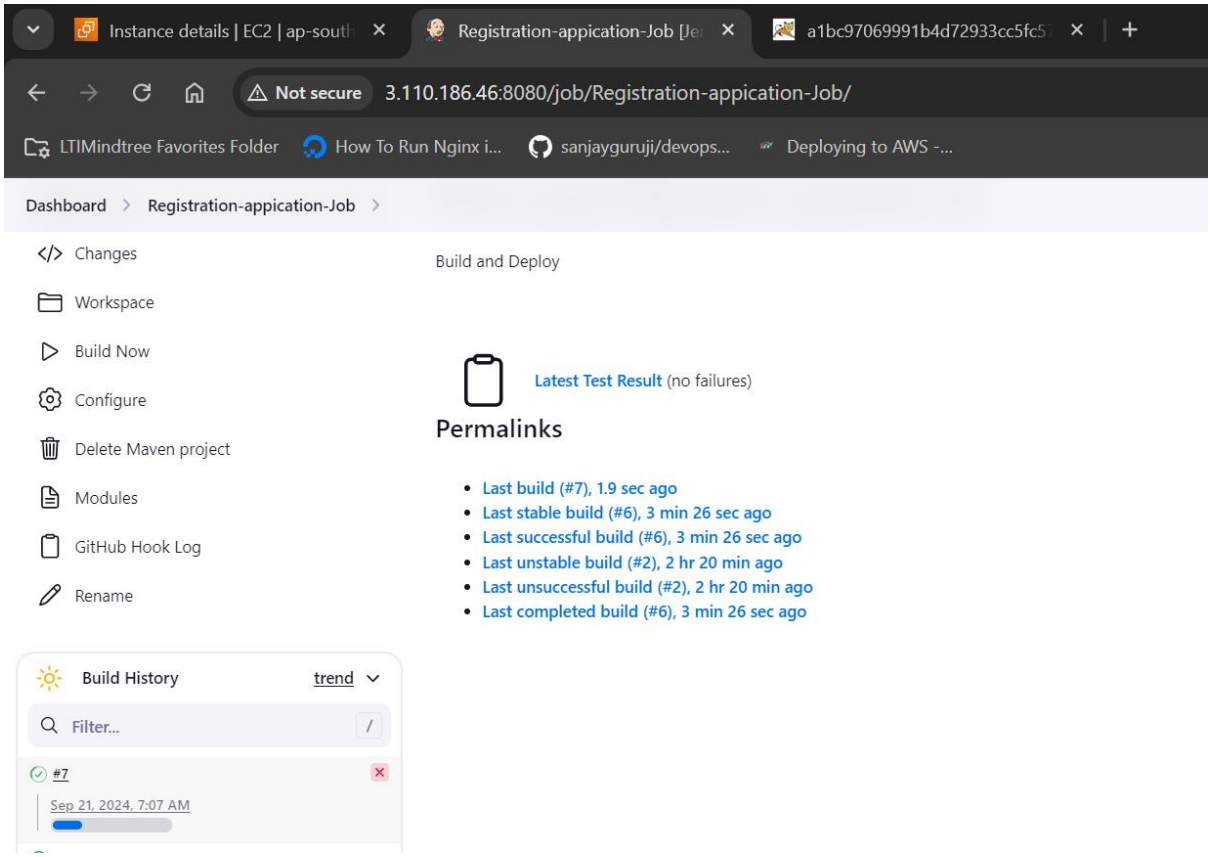


Now we have to made change in code

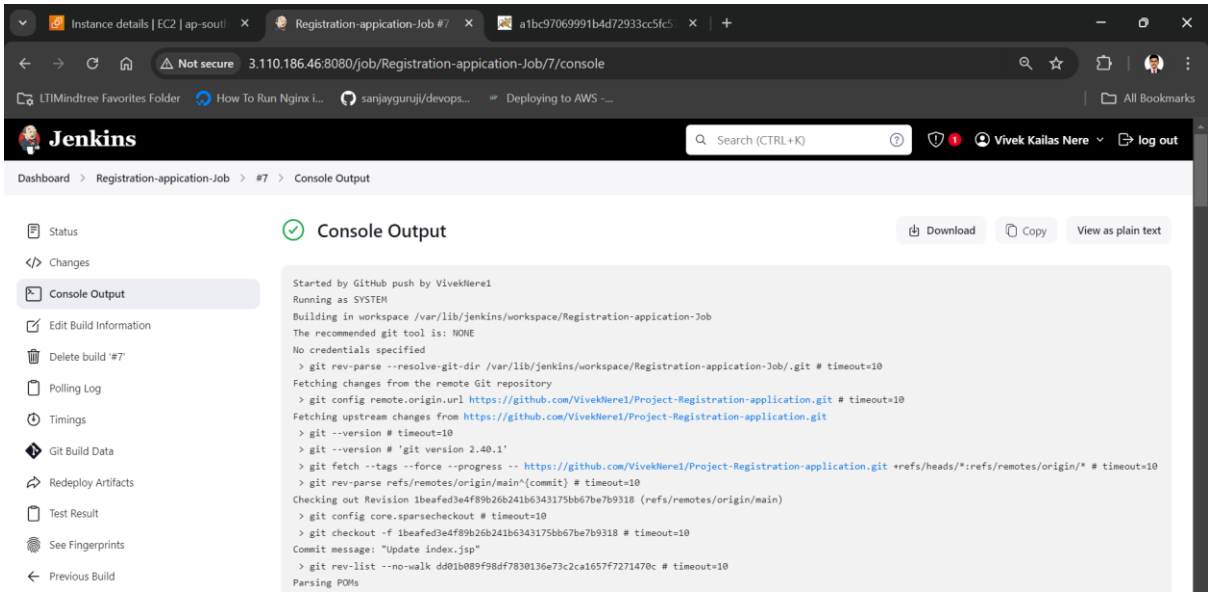


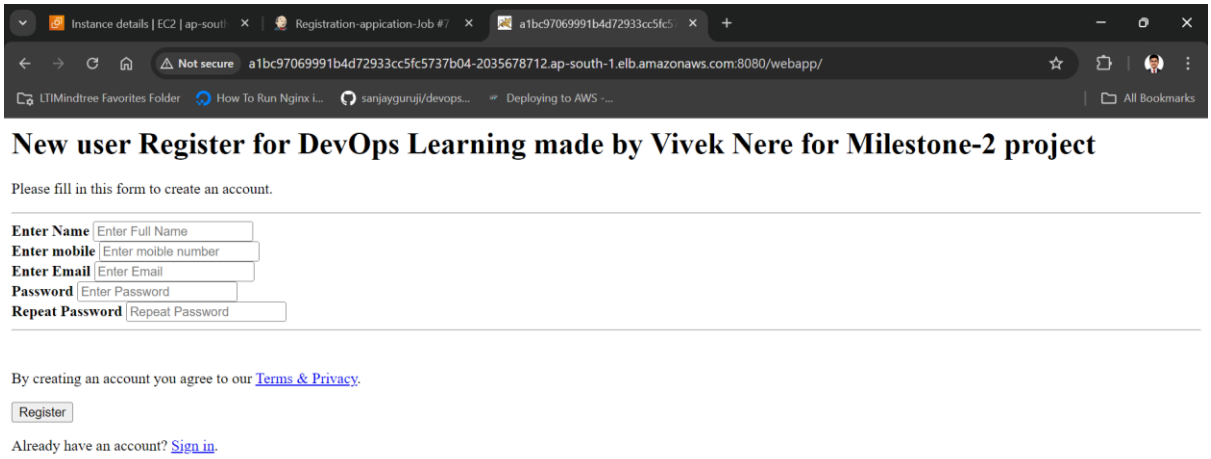
After committing it will automatically reflect in service deployment.





Our build is successful





Changes get automatically reflected on server with the help of docker and Kubernetes integration with Jenkins hence we have successfully Created an end to end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, GitHub as the SCM, Maven as the Build tool, Deploy in a docker instance and create a Docker image, Store the docker image in ECR, Achieve Kubernetes deployment using the ECR image. Build a sample java web app using maven.