DevOps Project

PROBLEM STATEMENT Create 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.

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.

yum

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:

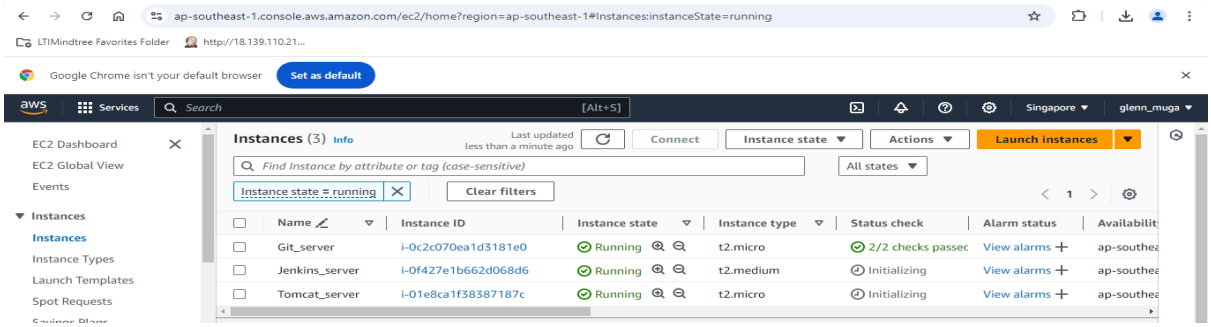
We will create 3 instances,

1 Git Client server

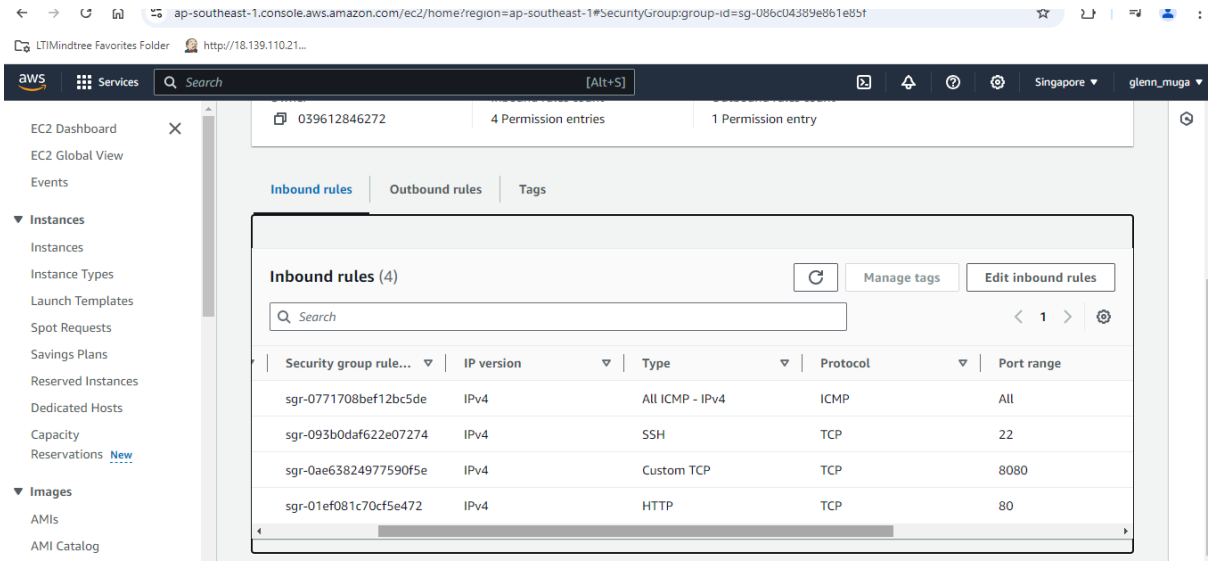
1 Jenkins server

1 tomcat server in the beginning

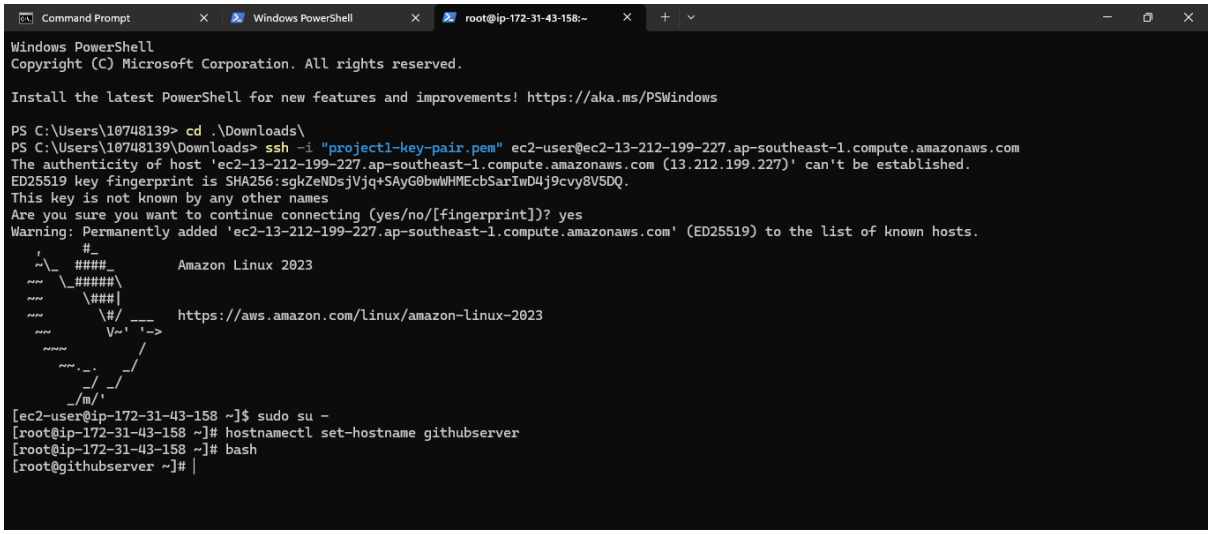
Choose t2. medium instance type for Jenkins since it requires 2 core cpu



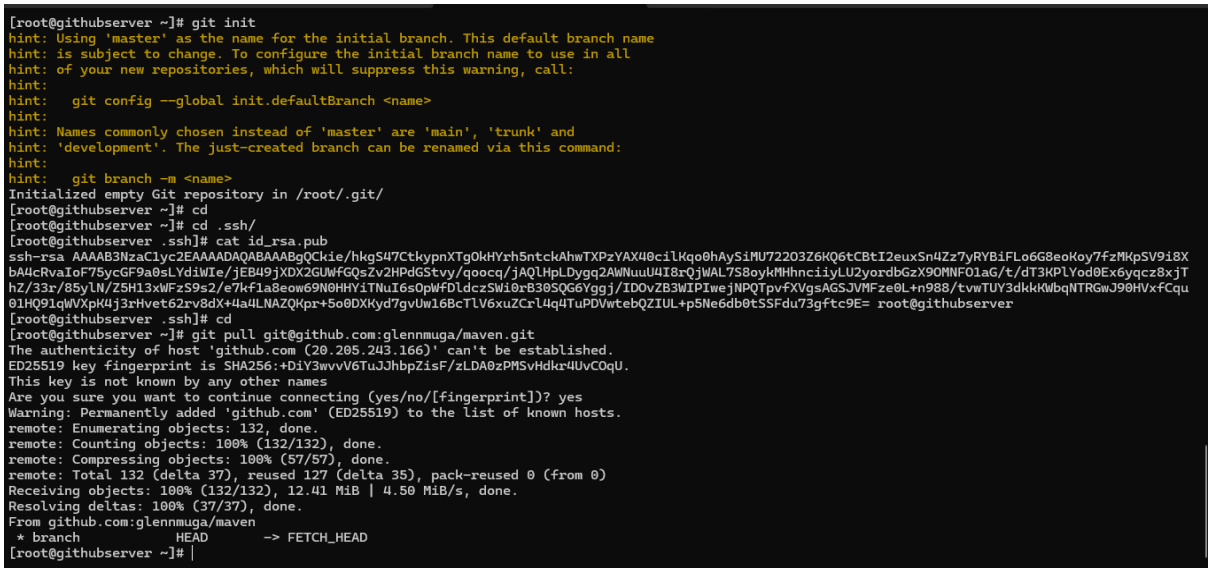
Add Security group rules as suggested below



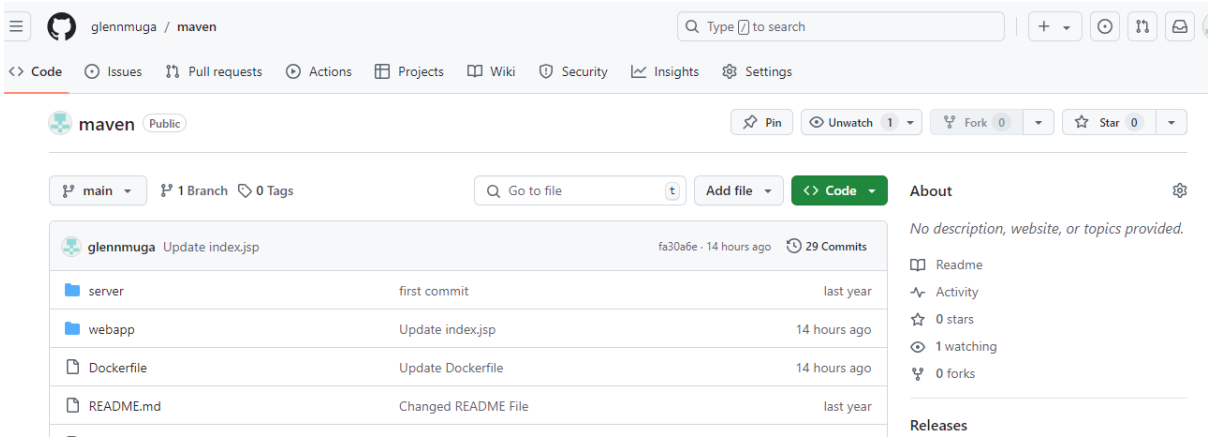
Connect the instances with the terminal



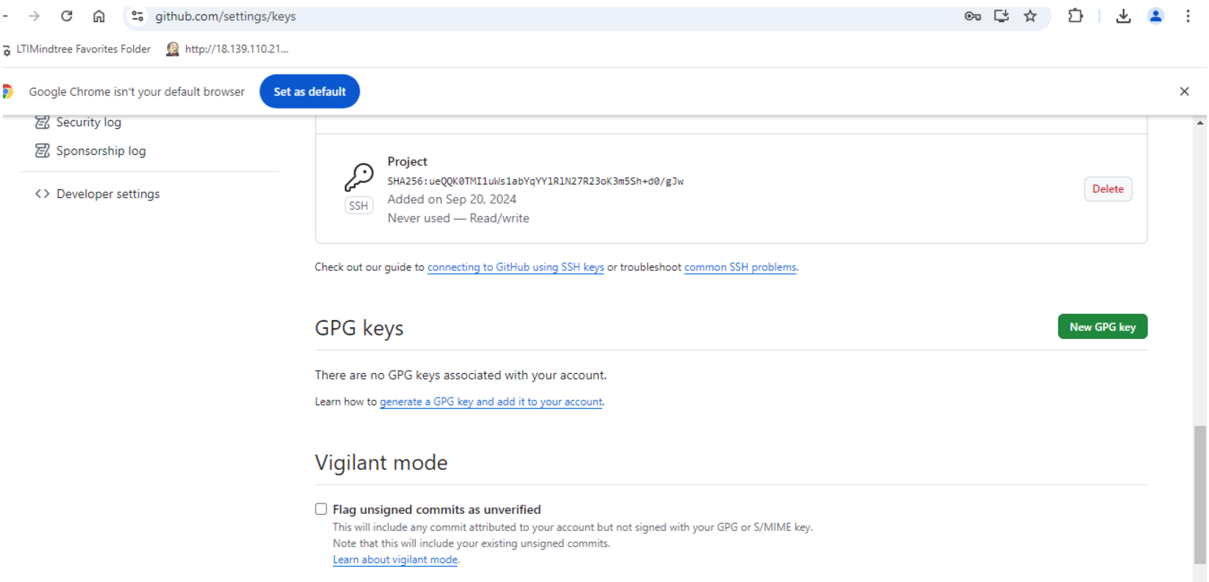
Initialize Git repository , ssh-keygen to connect the git client machine with the git repository.



The files are successfully pushed in the git repository.



Add ssh key to github repo to create a link between the github termial and web server



Add the git repo to your terminal

```
 * branch            HEAD       -> FETCH_HEAD
[root@githubserver ~]# git branch
* master
[root@githubserver ~]# ls
Dockerfile  README.md  Snjay_Devops_projetc.pdf  pom.xml  regapp-deploy.yml  regapp-service.yml  server  webapp
[root@githubserver ~]# git branch -M main
[root@githubserver ~]# git branch
* main
[root@githubserver ~]# |
```

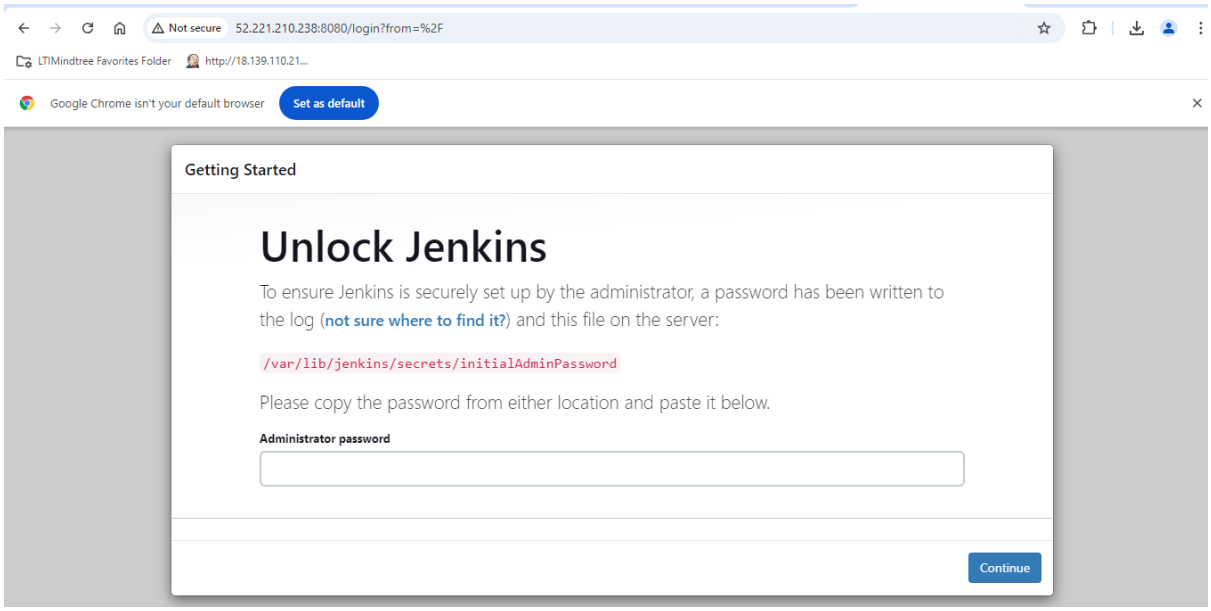
```
ot@jenkinsserver ~]# yum install git -y
t metadata expiration check: 0:21:47 ago on Fri Sep 20 12:21:23 2024.
endencies resolved.
================================================================================================================
ckage                  Architecture         Version                             Repository              Size
================================================================================================================
talling:
t                      x86_64               2.40.1-1.amzn2023.0.3               amazonlinux              54 k
talling dependencies:
t-core                 x86_64               2.40.1-1.amzn2023.0.3               amazonlinux             4.3 M
t-core-doc             noarch               2.40.1-1.amzn2023.0.3               amazonlinux             2.6 M
rl-Error               noarch               1:0.17029-5.amzn2023.0.2            amazonlinux              41 k
rl-File-Find           noarch               1.37-477.amzn2023.0.6               amazonlinux              26 k
rl-Git                 noarch               2.40.1-1.amzn2023.0.3               amazonlinux              42 k
rl-TermReadKey         x86_64               2.38-9.amzn2023.0.2                 amazonlinux              36 k
rl-lib                 x86_64               0.65-477.amzn2023.0.6               amazonlinux              15 k

nsaction Summary
================================================================================================================
tall  8 Packages

al download size: 7.1 M
talled size: 34 M
nloading Packages:
8): git-2.40.1-1.amzn2023.0.3.x86_64.rpm                                     746 kB/s |  54 kB     00:00
8): git-core-doc-2.40.1-1.amzn2023.0.3.noarch.rpm                             21 MB/s | 2.6 MB     00:00
8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm                             759 kB/s |  41 kB     00:00
8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm                          1.6 MB/s |  26 kB     00:00
8): perl-Git-2.40.1-1.amzn2023.0.3.noarch.rpm                                1.6 MB/s |  42 kB     00:00
8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm                          1.8 MB/s |  36 kB     00:00
8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm                                732 kB/s |  15 kB     00:00
8): git-core-2.40.1-1.amzn2023.0.3.x86_64.rpm                                 20 MB/s | 4.3 MB     00:00
```

```
================================================================================================================================
 Package                         Architecture                    Version                         Repository                  Size
================================================================================================================================
Installing:
 jenkins                         noarch                          2.462.2-1.1                     jenkins                     89 M

Transaction Summary
================================================================================================================================
Install  1 Package

Total download size: 89 M
Installed size: 89 M
Downloading Packages:
^[[Cjenkins-2.462.2-1.1.noarch.rpm                                                                  19 MB/s |  89 MB     00:04

--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                               19 MB/s |  89 MB     00:04
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                        1/1
  Running scriptlet: jenkins-2.462.2-1.1.noarch                                                                             1/1
  Installing       : jenkins-2.462.2-1.1.noarch                                                                             1/1
  Running scriptlet: jenkins-2.462.2-1.1.noarch                                                                             1/1
  Verifying        : jenkins-2.462.2-1.1.noarch                                                                             1/1

Installed:
  jenkins-2.462.2-1.1.noarch
```

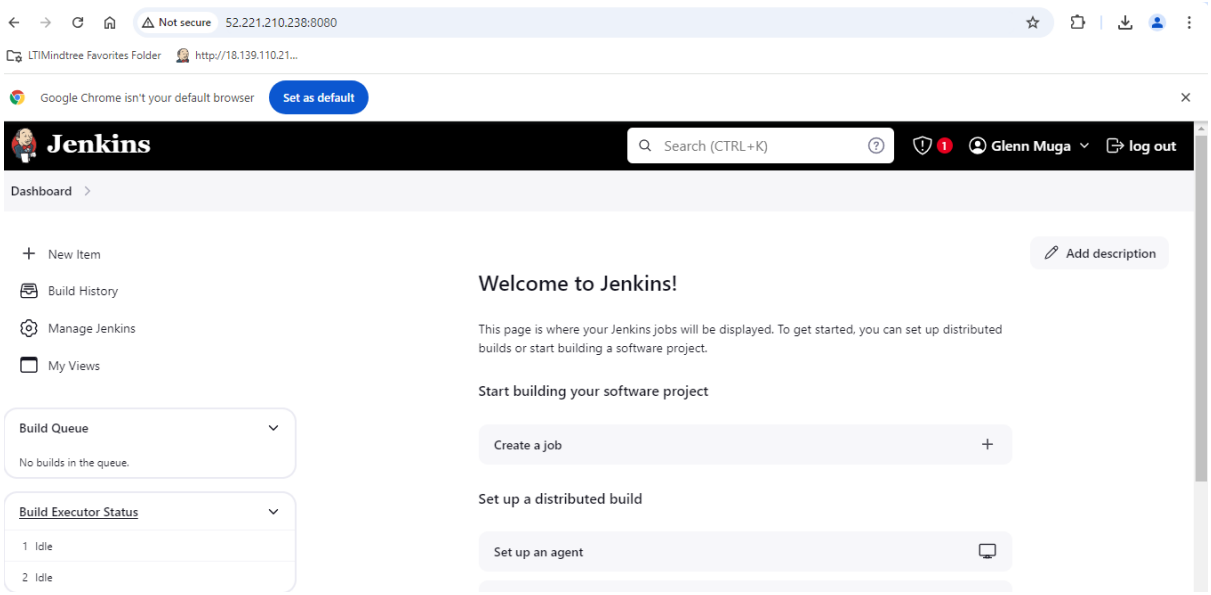
Install Jenkins on your Jenkins terminal

```
[root@jenkins ~]# wget -O /etc/yum.repos.d/jenkins.repo     https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-09-20 13:35:03--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.46.133, 2a04:4e42:48::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo        100%[=====================================================================>]      85  --.-KB/s    in 0s

2024-09-20 13:35:03 (1.90 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]
```

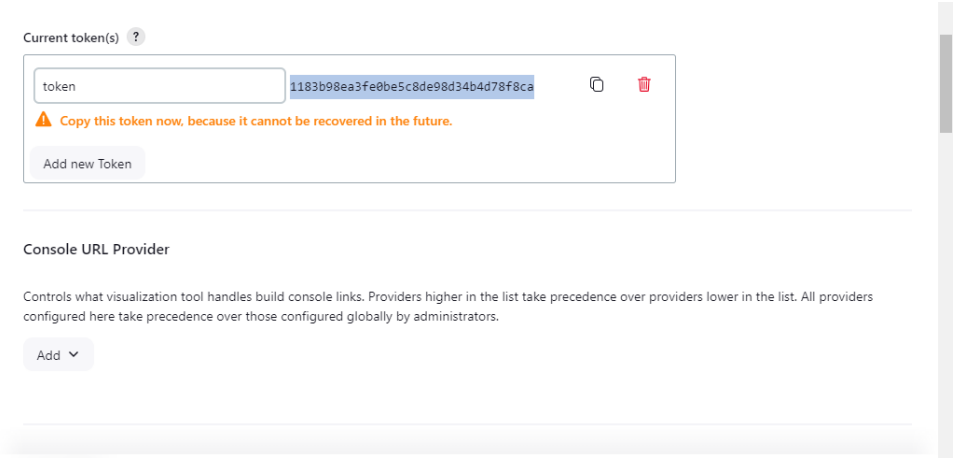
Open the Jenkins server using the public ip of the instance



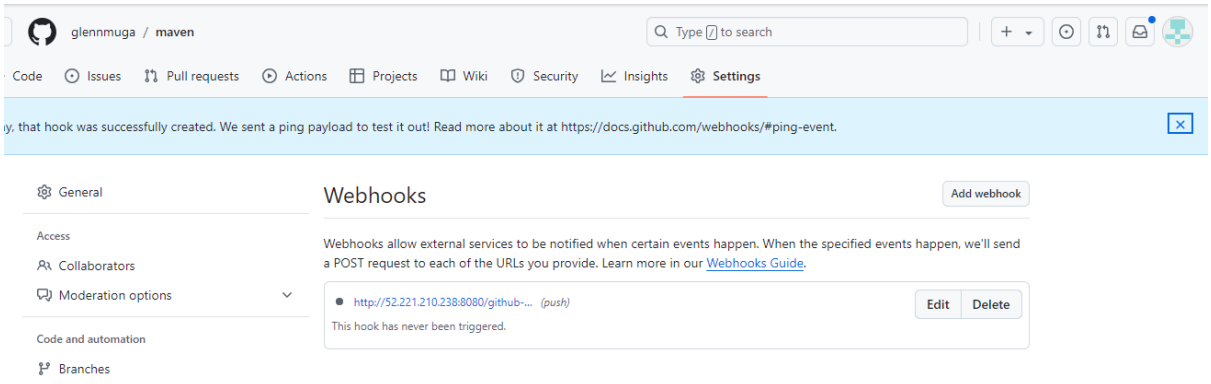
You can see we have now entered the Jenkins dashboard



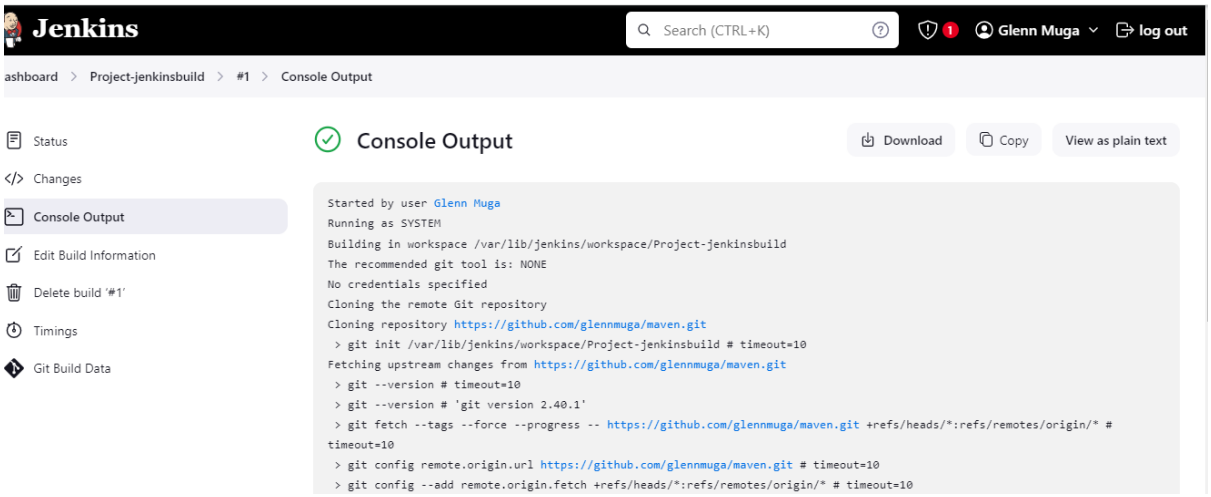
Create API token in Jenkins



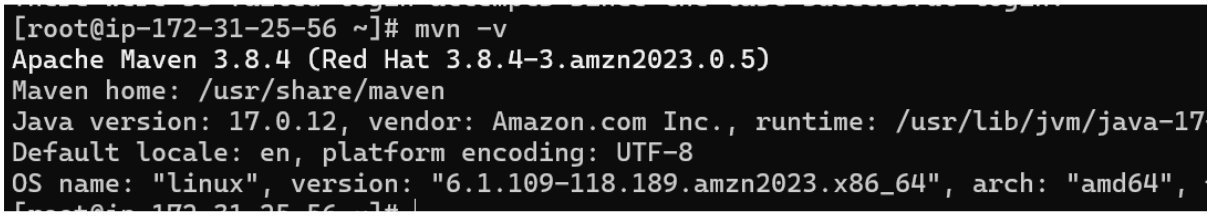
To connect Jenkins with GitHub repository we need to add Webhooks



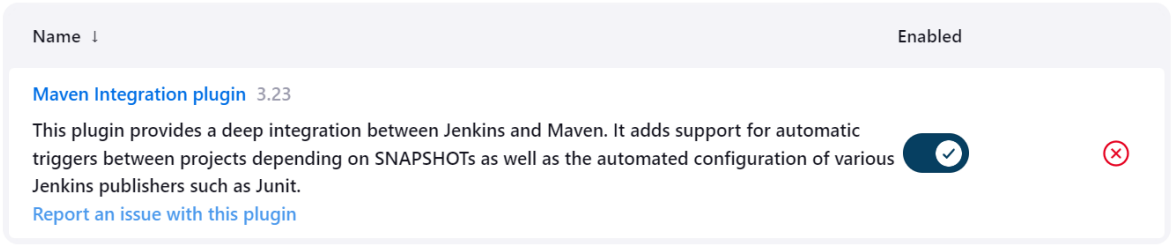
After adding the following we can build Jenkins and test the build is successful.



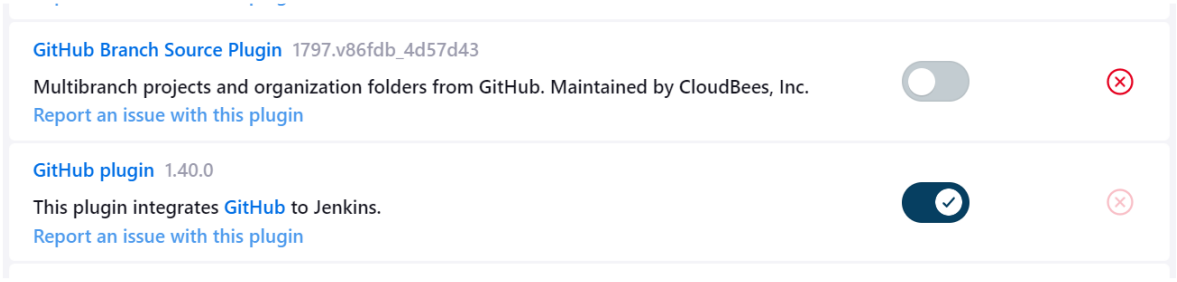
Now we will install maven on our server :



In Jenkins> dashboard > manage Jenkins > available plugins > maven integration >install



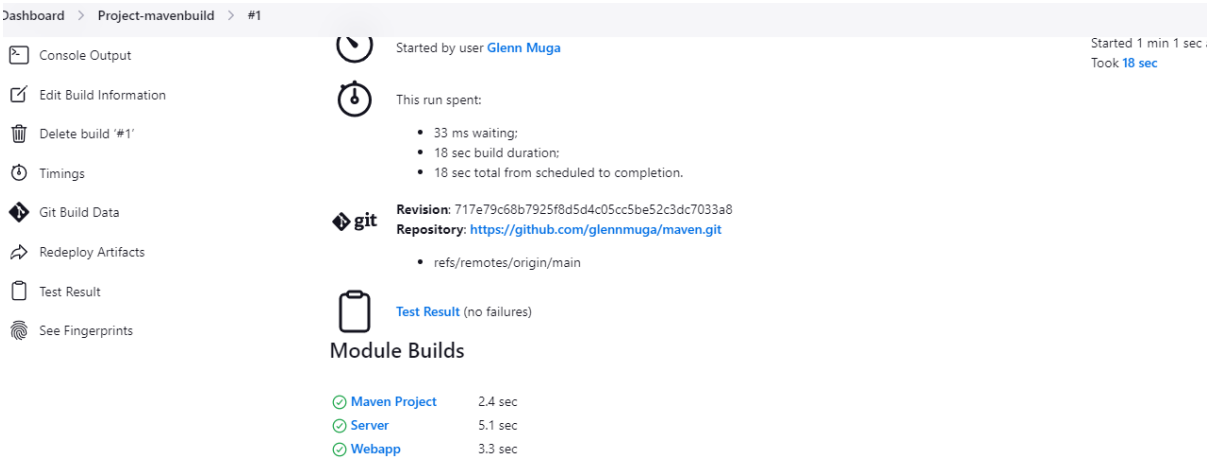
In installed plugins > type github > disable github branch source plugin and enable github plugin.



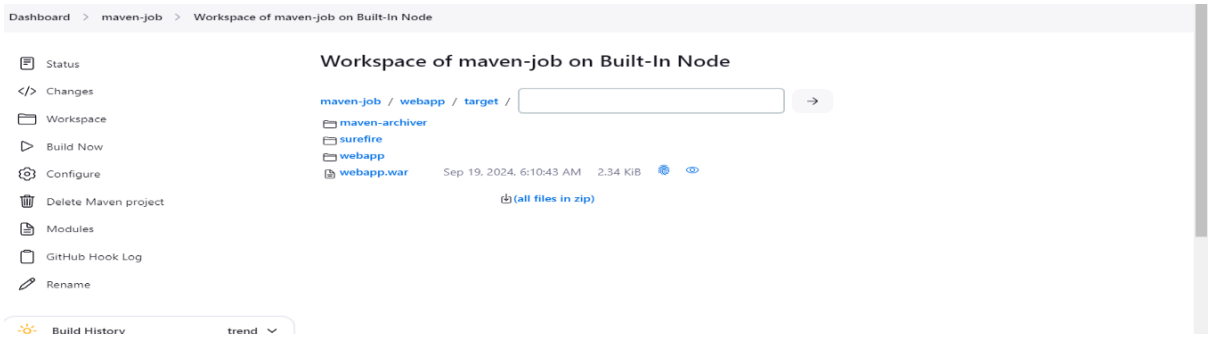
After installing restart Jenkins

In Manage Jenkins add tools and paste the java and maven path from Jenkins-server

You can now build maven and see the successful build below , we have received our git repo modules as seen below



The webapp .war file is visible which indicates that maven is building artifacts successfully.



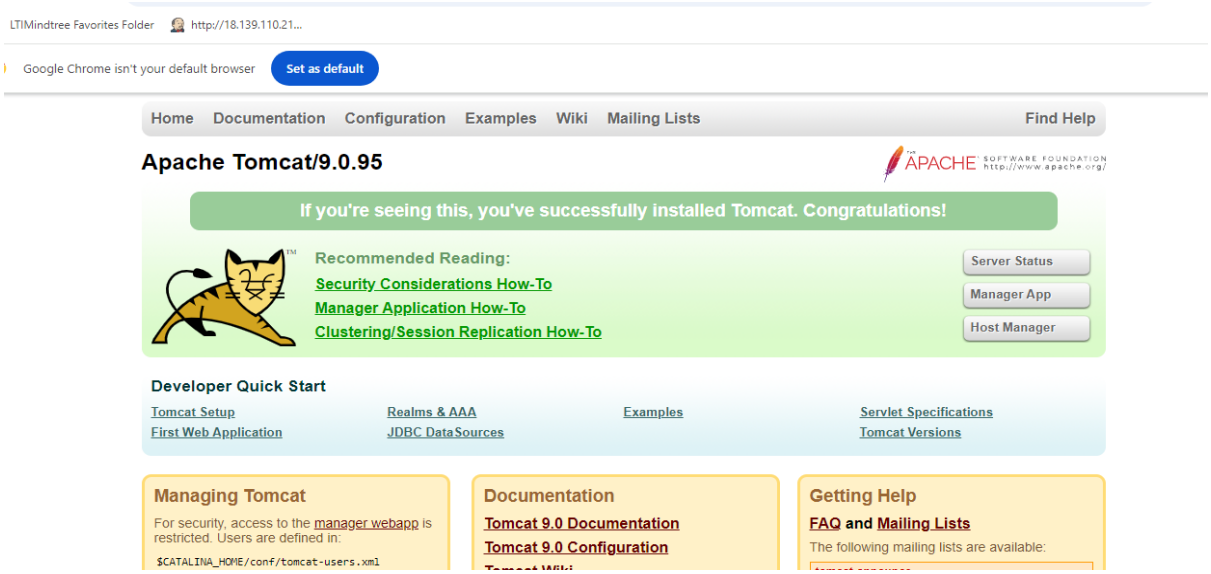
Now we will build Apache tomcat server

```
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64                    libpng-2:1.6.37-10.amzn2023.0.6.x86_64
libxcb-1.13.1-7.amzn2023.0.2.x86_64                          pixman-0.40.0-3.amzn2023.0.3.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch

omplete!
root@tomcatserver ~]# java --version
penjdk 17.0.12 2024-07-16 LTS
penJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7-LTS)
penJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7-LTS, mixed mode, sharing)
root@tomcatserver ~]#
```

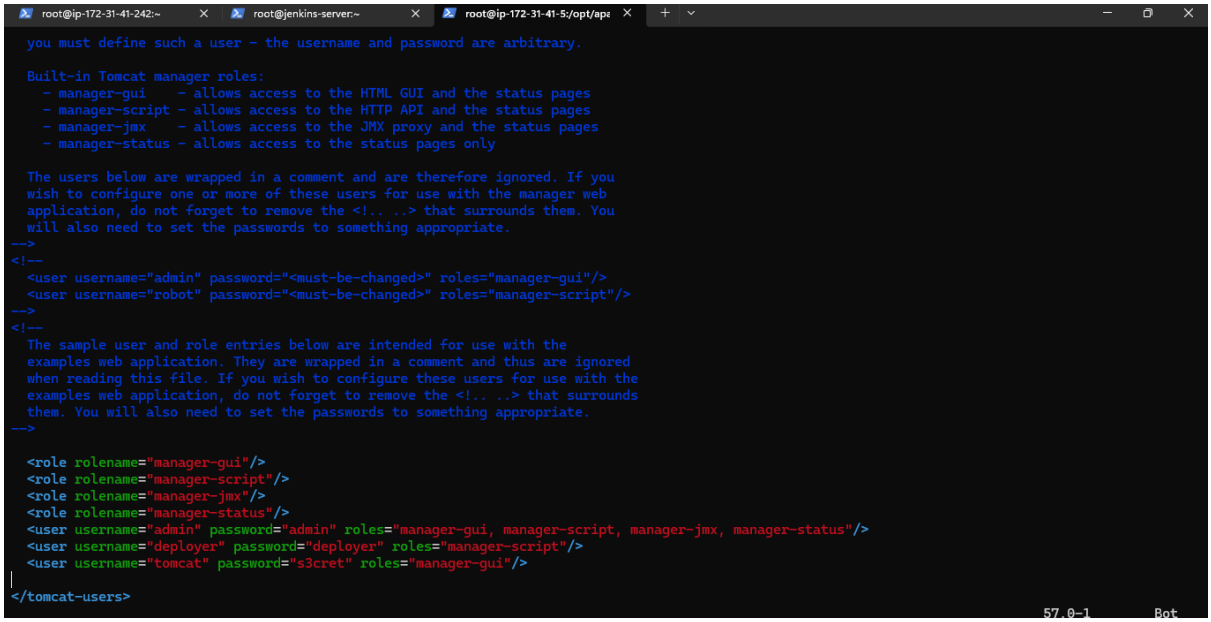
Download the tomcat using wget command

```
[root@tomcatserver opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.tar.gz
--2024-09-20 14:00:03--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.95/bin/apache-tomcat-9.0.95.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: 12715996 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.95.tar.gz'

apache-tomcat-9.0.95.tar.gz         100%[=====================================================================>]  12.13M  --.-KB/s    in 0.1s

2024-09-20 14:00:04 (88.4 MB/s) - 'apache-tomcat-9.0.95.tar.gz' saved [12715996/12715996]
```

Open tomcat using the public ip

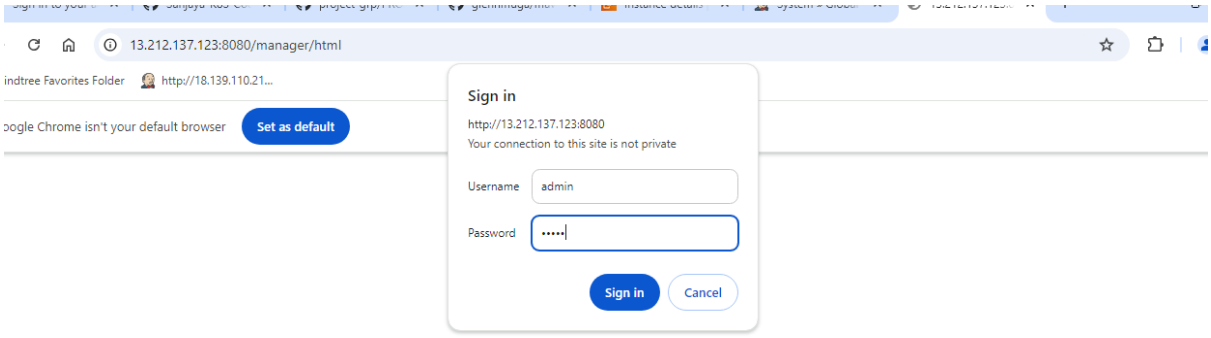


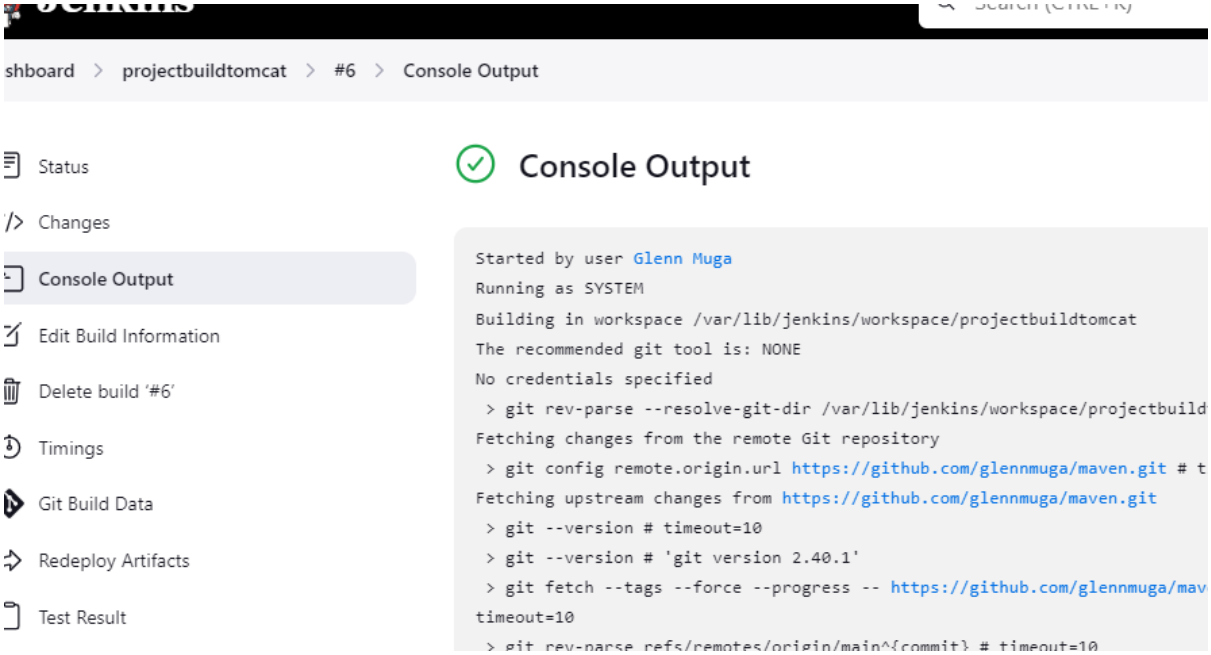
Add user's admin, deployer and assign roles using vim tomcat-users.xml



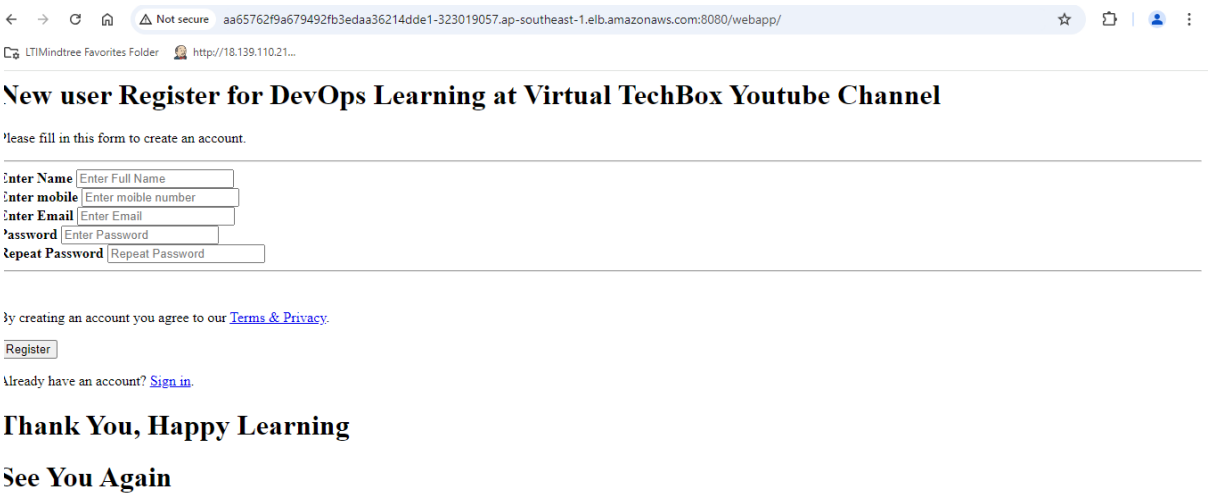
Shutdown and start the tomcat

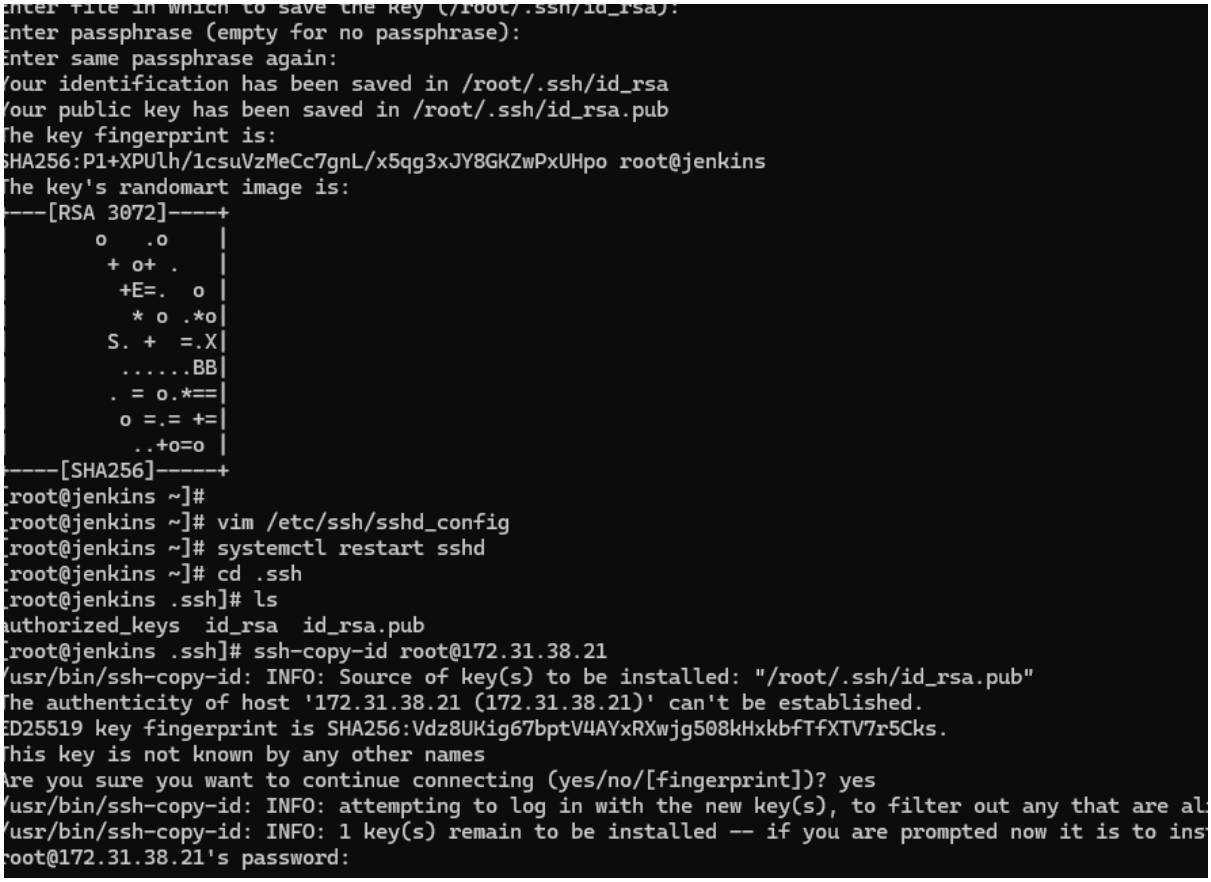
Now open the Apache tomcat server





Expose the web application on tomcat using public Ip on 8080 Port





Create global credentials for tomcat server in Jenkins machine

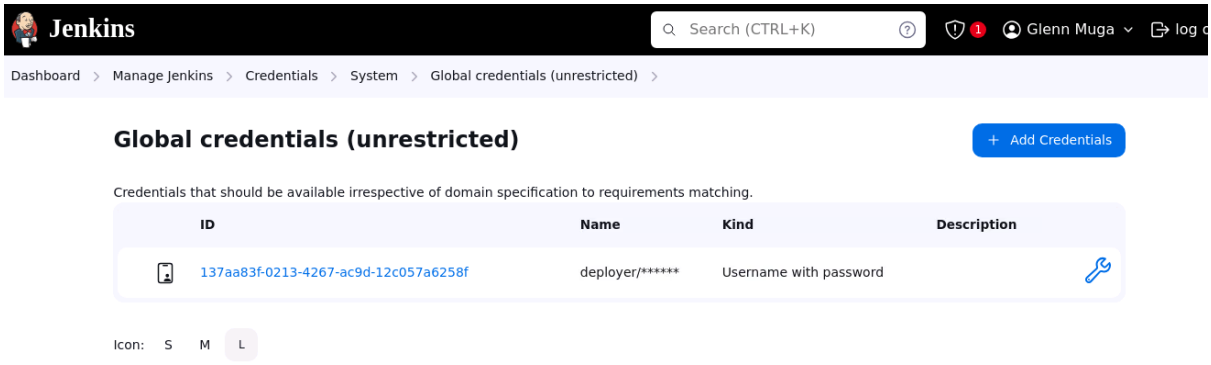
To integrate tomat server with Jenkins:

Go to Manage Jenkins > plugin > available plugins > deploy to container > install

Manage Jenkins > Credentials > system > global credentials > add

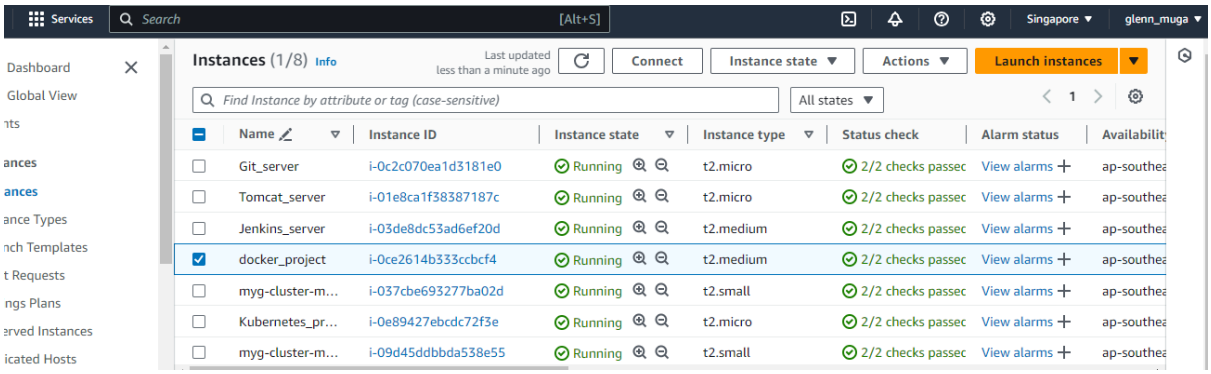
Username: deployer

Password: deployer



Docker server

Launch docker server



Install docker and start the service

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.20-1.amzn2023.0.1.x86_64                                                    1/10
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                                                        2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                  3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                   4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                        5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                         6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                  7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                       8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                             9/10
  Verifying        : runc-1.1.13-1.amzn2023.0.1.x86_64                                                         10/10

Installed:
  containerd-1.7.20-1.amzn2023.0.1.x86_64        docker-25.0.6-1.amzn2023.0.2.x86_64        iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64       libcgroup-3.0-1.amzn2023.0.1.x86_64        libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64      libnftnl-1.2.2-2.amzn2023.0.2.x86_64       pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.13-1.amzn2023.0.1.x86_64

Complete!
[root@docker ~]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[root@docker ~]#
```

```
omplete!
root@docker ~]# sudo service docker start
edirecting to /bin/systemctl start docker.service
root@docker ~]# docker ps
ONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@docker ~]# ssh-keygen
enerating public/private rsa key pair.
nter file in which to save the key (/root/.ssh/id_rsa):
nter passphrase (empty for no passphrase):
nter same passphrase again:
our identification has been saved in /root/.ssh/id_rsa
our public key has been saved in /root/.ssh/id_rsa.pub
he key fingerprint is:
HA256:9rOw0WgIkFRMsh05nX8mJNQ/VNVlN1VyKyosAyGFE4c root@docker
he key's randomart image is:
---[RSA 3072]----+
   o*OB.o   ...oo@|
  . E*.= o .    +=|
   + oo + o   . . |
    .   . + = . .  |
     .  oS* o      |
      . oo+.       |
       . = +       |
        . + o      |
         . .       |
----[SHA256]-----+
root@docker ~]#
```

Do SSH-KEYGEN to connect the server with docker server

```
Last login: Fri Sep 20 13:49:15 2024 from 167.103.3.72
[ec2-user@jenkins ~]$ ssh-keygen
Generating public/private rsa key pair.

Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa): Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LYmVC9bdkps2CRyTiYddaY6jz69GZFk5srCW02qcBHU ec2-user@jenkins
The key's randomart image is:
+---[RSA 3072]----+
|     ..+E+.o     |
|    . ++*=*o     |
|     .o**O+..    |
|     .*+O=.=     |
|     +.BS.B      |
|      * .o .     |
|     . +         |
|       +         |
|      ..o.       |
+----[SHA256]-----+
[ec2-user@jenkins ~]$
```

Configure AWS

```
+----[SHA256]-----+
[root@docker ~]# aws configure
AWS Access Key ID [None]: AKIA2HVQ5KTOIQPDURGS
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]:
[root@docker ~]#
```

Copy public Ip address of Jenkins in docker

Copy public Ip of Jenkins, docker in Jenkins

```
----[SHA256]-----+
root@docker .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCYRkZ6TfXpEZQWFal3QiiGok5e5gE1BmkqZ7tbEBM3nl8CisgRPY3Am2eyiVeQcYPxCOCbblEZ0CD6kdcOUrysjRrunBgva7EvomsC8r/cANDL
djZDkcxCNlhY4jG5UMhgGyCizyJ58z//iqPYRZuIH9ct79ZiRJBujr7FEGVrmC2weCWnH+5QtgAd7Veqkzk+kpswbQGBQNt1CQQdUEW4GNu6dVlsLeAOvC+CnN21A28bOiM9oPaD42XQsDzCZS8
MB9AIpsAdnhHSbssRgLpRpVeJLV/vKw+iHQT96iYLWPM3XUEKEOpNgDKucO8Rr0XDJCVbtCHZJSeQ84AJVgsbqRITnrNaKg8LjB7ZsvQQyuUlrbBH4Rj8Cx5I7EdcfM7MLCgZLQ5xtaV6E7GnvW
ZlbPAfpajZoloA34PBOwNf+4cDHqTwVBrvlwbdDM6qud72mmCQZAkr9vKAhXXfV4w9sabTeV3Ku1QFzcj3/EOPbBv375hKyaSOclCHdwlM= root@docker
root@docker .ssh]# ls
uthorized_keys  id_rsa  id_rsa.pub
root@docker .ssh]# cat authorized_keys
o-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
eep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCNLFyOBH8y5CtTL4khOLaMqDkoCwwNTWtz8D7pFtkPA4rEUjYN72gl4cGgDmRuifci6yZ+c3zfGohPmBzHcjzxZ5V65O
T77Fnd/ZJvI3882xlQUJVx35wvOs4KLz/lulj/N6g7DOpXHX1kLqjgwLzjzLt/JWm2xmNt7vAQGZMe+jcBb7zn8SIyK01DBPHYOVtN+EkgnZZ6tWfrMo7iwDzMPQK5ravpZfDbkD0HuN5eTQ/GL
kF3/vl0GuahWi8NaWsDw3sHtyF70MwK9AaBybX0bt1GBs3Nltg17VW+VnJialVi7MyVlaVCQ5jIiK29oelB5T31vdMSZx6tMXFWnd project1-key-pair
root@docker .ssh]# vim authorized_keys
root@docker .ssh]# cd
root@docker ~]# ping 172.31.35.236
```

```
[root@jenkins .ssh]#  cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQChtBVPA/DBaAWOpu1qNtWOQpoBLn0mEGh60hC4yw+4YCt2jlcIU61drBGyWAVG9f9cWgRLr3Y2uiaBFXFedf9IMJv9UkoTUqy+FqWJ3fpaW2Al
CIe2rqZaZEPzX0RnmasNYgp1uAdepQVc1+7zl8T81TV0Qib8QGF2/grsE8g2VmkysvqUf6rkp2/YVh7F3dkfBUa8rU3W1mxKV3B2FhDNu7xpHnJyjdNN85C4t+RXbOw+POBBYDkvyFx0N4otHszT
sVXpA//UV5+5aP+UqavRb7P1jN4rW4mAmbOy+szQON6bWxMhZM7eJvABoIt7aEUvydyjRiekM5BPQSxYMxcWwBnxoQG8lLwW91dUhESBBlqZYwVM0q0gby+u7JZ2Z2A9uKoJLKcqdTbp8SjltwE6
MqJ3M3V47hpjY1k5OswXQ9nx1cPnmXtJddWiF9C62siITEKczhn8uhvJb09/XcS1ACBNprpyZlTjJu2INqP3bWsKrmxS/c0B2NztNbGxdRk= root@jenkins
[root@jenkins .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCNLFyOBH8y5CtTL4khOLaMqDkoCwwNTWtz8D7pFtkPA4rEUjYN72gl4cGgDmRuifci6yZ+c3zfGohPmBzHcjzxZ5V65O
BT77Fnd/ZJvI3882xlQUJVx35wvOs4KLz/lulj/N6g7DOpXHX1kLqjgwLzjzLt/JWm2xmNt7vAQGZMe+jcBb7zn8SIyK01DBPHYOVtN+EkgnZZ6tWfrMo7iwDzMPQK5ravpZfDbkD0HuN5eTQ/GL
hkF3/vl0GuahWi8NaWsDw3sHtyF70MwK9AaBybX0bt1GBs3Nltg17VW+VnJialVi7MyVlaVCQ5jIiK29oelB5T31vdMSZx6tMXFWnd project1-key-pair
[root@jenkins .ssh]# vim authorized_keys
[root@jenkins .ssh]# vim authorized_keys
[root@jenkins .ssh]# cd
[root@jenkins ~]# ping 172.31.38.21
PING 172.31.38.21 (172.31.38.21) 56(84) bytes of data.
64 bytes from 172.31.38.21: icmp_seq=1 ttl=127 time=2.74 ms
64 bytes from 172.31.38.21: icmp_seq=2 ttl=127 time=1.28 ms
64 bytes from 172.31.38.21: icmp_seq=3 ttl=127 time=0.989 ms
^Z
[4]+  Stopped                 ping 172.31.38.21
```

Now we can ping the terminals of each other and check the terminals are connecte

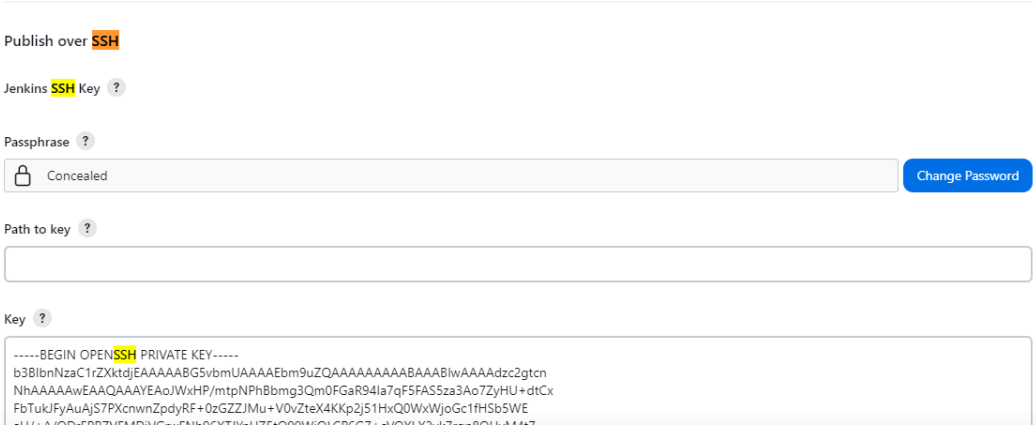
```
[root@docker ~]# ping 172.31.35.236
PING 172.31.35.236 (172.31.35.236) 56(84) bytes of data.
64 bytes from 172.31.35.236: icmp_seq=1 ttl=127 time=0.902 ms
64 bytes from 172.31.35.236: icmp_seq=2 ttl=127 time=0.757 ms
64 bytes from 172.31.35.236: icmp_seq=3 ttl=127 time=1.11 ms
64 bytes from 172.31.35.236: icmp_seq=4 ttl=127 time=0.880 ms
^Z
[1]+  Stopped                 ping 172.31.35.236
[root@docker ~]#
```

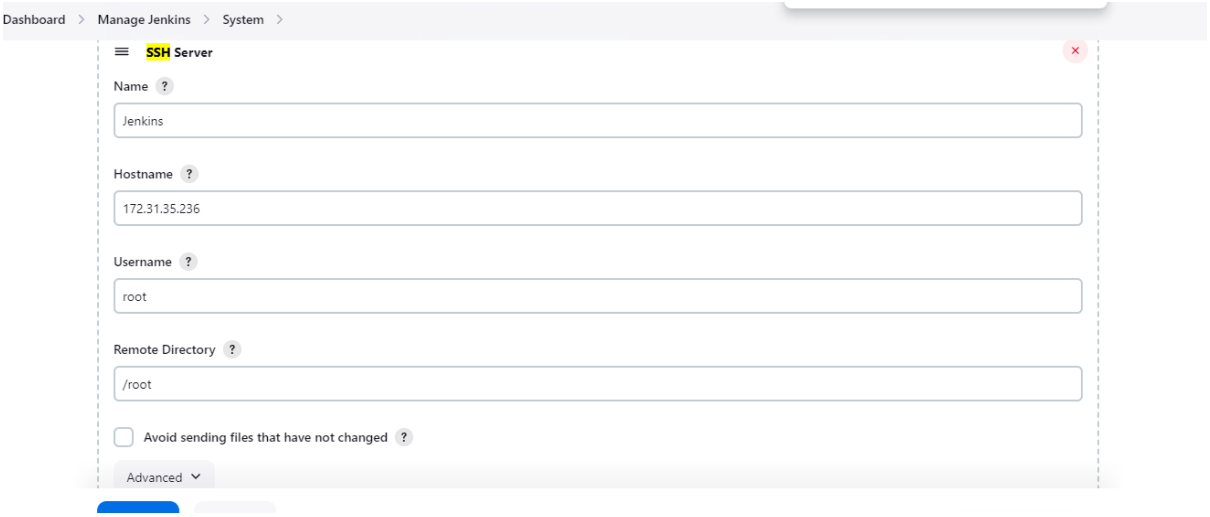
```
       valid_lft forever preferred_lft forever
[root@jenkins .ssh]# ssh-copy-id root@172.31.35.236
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.35.236 (172.31.35.236)' can't be established.
ED25519 key fingerprint is SHA256:1wlRi/mJ/MBmwHN7IpzUKCONsMmSl+9APS9zptx0e90.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.35.236's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.31.35.236'"
and check to make sure that only the key(s) you wanted were added.

[root@jenkins .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCNLFyOBH8y5CtTL4khOLaMqDkoCwwNTWtz8D7pFtkPA4rEUjYN72gl4cGgDmRuifci6yZ+c3zfGohPmBzHcjzxZ5V65O
BT77Fnd/ZJvI3882xlQUJVx35wvOs4KLz/lulj/N6g7DOpXHX1kLqjgwLzjzLt/JWm2xmNt7vAQGZMe+jcBb7zn8SIyK01DBPHYOVtN+EkgnZZ6tWfrMo7iwDzMPQK5ravpZfDbkD0HuN5eTQ/GL
hkF3/vl0GuahWi8NaWsDw3sHtyF70MwK9AaBybX0bt1GBs3Nltg17VW+VnJialVi7MyVlaVCQ5jIiK29oelB5T31vdMSZx6tMXFWnd project1-key-pair
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCh3Et/b2zMms72UNVpApsAYMdqBrWlEWW839NXCld9cd185c8EexeNDX/sc7H5mVq1UT9LdqkXU7/NhkgTbxW2xtJ8zwOI0N2L+soaM20FDRmP
RfQpMlXV/ddxF86/JZJQz45BmqSpIIrhu4azYbliHb1M2nD7hx8nLuMp9mhwTK42T2lhJYMWXQBKASldjOKmLNRDplxQeVVvgtgTuq+liFMyBGH6oDVn+SSCVBWHv12VwdE03eodO078b6Fwk8La
Ng/GWiKNdAId0P8pGUpWW2UJJUaa+HvrSAscGl2DlC05CrvMl2aGAKdJhq6qWzq3U3cyHncMEeHr83xq/pKdyp/EQVPRXPcrlIgWxGWz0qIIdopHAJX87L3OKtmteKcqHzgBGmOQR2CqpzP/DwKm
hOCVa/qK0OewWUb1sbLfX2v9lmA67Mi+fICP9BzCFcDxuEyKVf16n/d/Bcdxnhh ZfcL2ej7KZcKkyF3W8XY1iQjEVFSr/5xL3fHpraZP6ec= root@docker
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCglbEc/+a2k0+EFuaDdCbQUZpH3ghruoXkUBLnNrcCjtnIdT520LEVtO6QkXIC4CNLs9dyfCdml3JEX7TMZlkky75XS9m15fgoqnaPnUfFDRbF
aOgZzV8dJvlYRpT/4D9AOsQ89lUUwOJUavAU1v3pdMlidRnm0731aNAsI/obv5xU5ctfa+Tuuqnw4fIzi3uHapcly7R0gwiaCgx5h6hF+AiHdvFYzJ7DLYPSzQuTRjft2hvoh5j39YHdC1xeH6EA
udOqOGsjp1uzwkhKteQZh4vC9+8IgCkTQcijEO5LqdM2ZLGdqthbvqp2luVm4NsYWzPZmpAlokxSkPp5EuTmzHc2ROQGgUy1ji5f2qizN6rCOx50VxY22eHD+ChaGR11Nlqg+Yfr2FTZ98Hnmk9z
1X15rgc1A2evnkpilF3t5qLJsxWD7UBFaIhsmd6r5PlMpsOARruau+vlkEIQPHUT/G+/wSD5cGTAtOkOxFacnN5MlTYZ5FNIDaMo985E1Ek= root@jenkins
```

```
[root@docker ~]# vim /etc/ssh/sshd_config
[root@docker ~]# systemctl restart sshd
[root@docker ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:67:44:63:19:fd brd ff:ff:ff:ff:ff:ff
    altname eni-072069ec1693257f9
    altname device-number-0.0
    inet 172.31.38.21/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
       valid_lft 3295sec preferred_lft 3295sec
    inet6 fe80::467:44ff:fe63:19fd/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:dc:35:28:ce brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
[root@docker ~]# cd .ssh
[root@docker .ssh]# ssh-copy-id root@172.31.35.236
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.35.236 (172.31.35.236)' can't be established.
ED25519 key fingerprint is SHA256:1wlRi/mJ/MBmwHN7IpzUKCONsMmSl+9APS9zptx0e90.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.35.236's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@172.31.35.236'"
```

```
[root@docker ~]# sudo service docker start
Redirecting to /bin/systemctl start docker.service
[root@docker ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@docker ~]# docker images
REPOSITORY                                                    TAG       IMAGE ID       CREATED              SIZE
project-devops                                                latest    dd08530fc78d   About a minute ago   474MB
039612846272.dkr.ecr.ap-southeast-1.amazonaws.com/project-devops   latest    dd08530fc78d   About a minute ago   474MB
[root@docker ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@docker ~]# cd /opt
[root@docker opt]# ll
```

Add ssh server of Jenkins and docker in Jenkins dashboard to connect terminal to jenkins

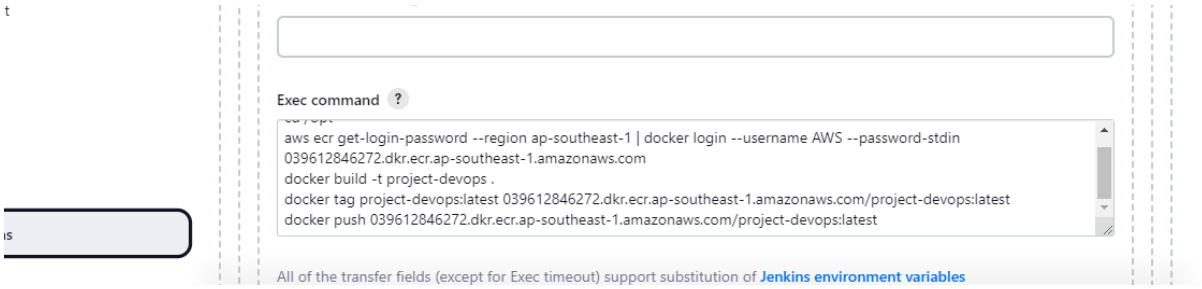


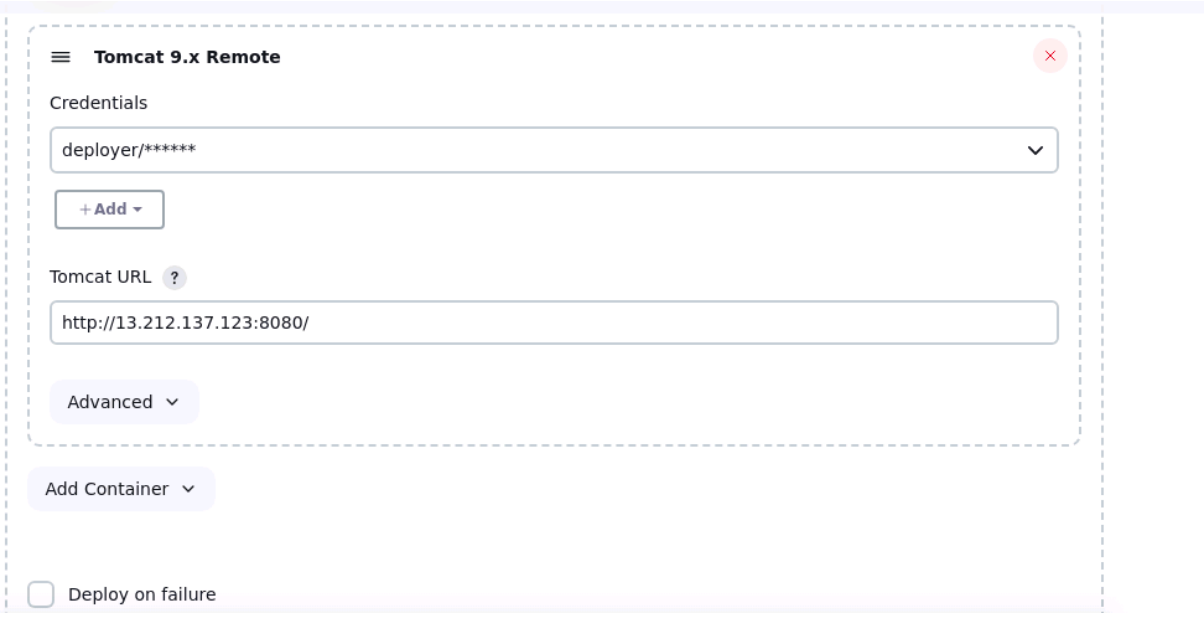


From post build actions select build artifacts over ssh

we are connecting Jenkins with docker by providing docker IP

The commands for creating the image and pushing image into ECR





Build again and see build is successful

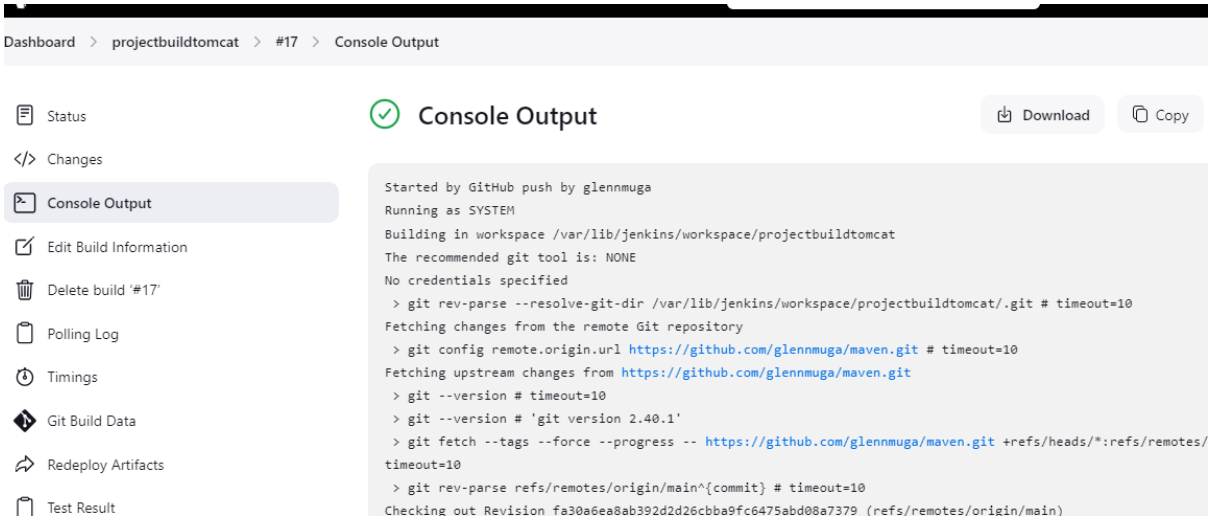
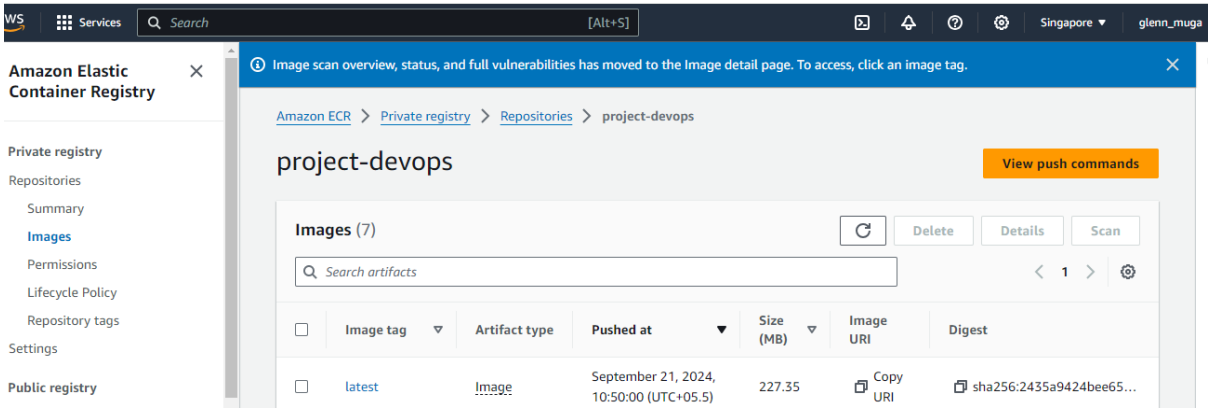
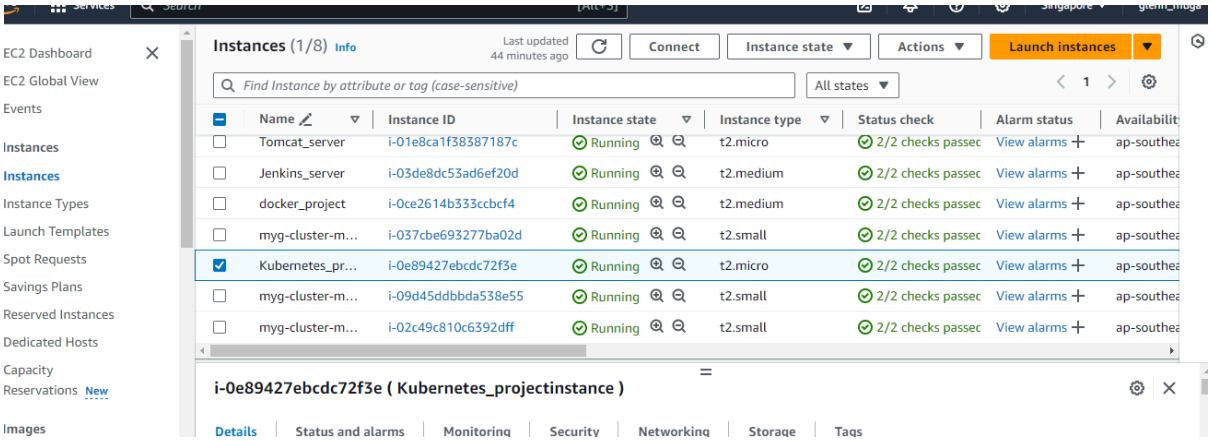


Image is successfully pushed into ECR

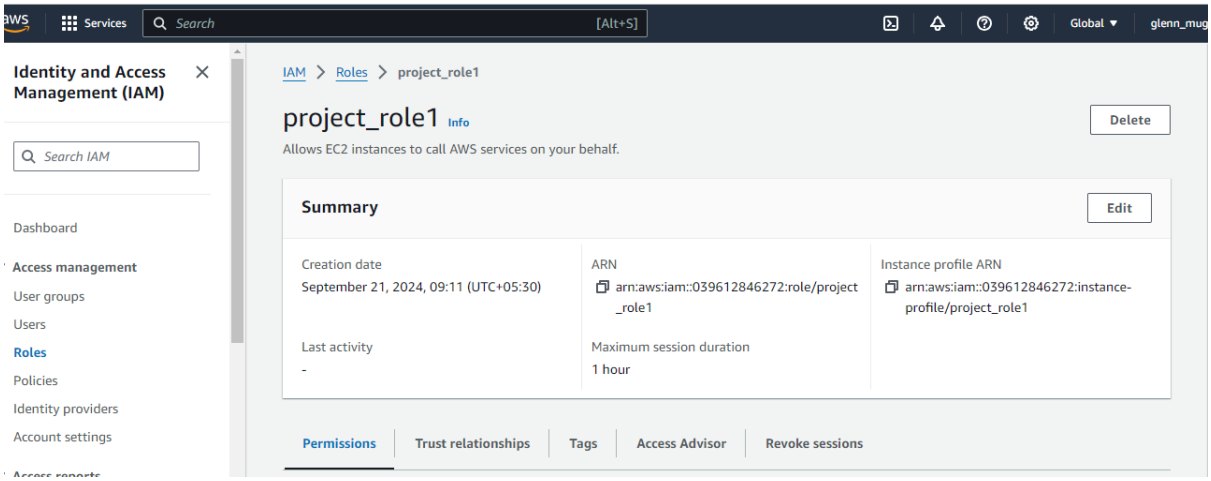


Kubernetes cluster

Launch the Kubernetes server



Attach the role to the Kubernetes instance with the access to permissions of eks, elasticcontairregistry and iam full access.



```
[ec2-user@ip-172-31-44-151 ~]$ sudo su -
[root@ip-172-31-44-151 ~]# hostnamectl set-hostname kubernetes
[root@ip-172-31-44-151 ~]# bash
[root@kubernetes ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:x0K4vpksweUMhpN535uOzBsXJl2D0+ojFXOXs8rYyPg root@kubernetes
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|        . o   .  |
|   +   . * = +   |
|  = + .+ O o o   |
|   = *o.S o .    |
|    o.+B.O .     |
|      .= Bo+     |
|     .+ Xo.      |
|      .XoE       |
+----[SHA256]-----+
[root@kubernetes ~]# yum update -y
```

Configure AWS

```
[root@kubernetes ~]# aws configure
AWS Access Key ID [None]: AKIAQSOI4CDAIFRIMSPX
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]: table
[root@kubernetes ~]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar
xz -C /tmp
[root@kubernetes ~]# sudo mv /tmp/eksctl /usr/local/bin
[root@kubernetes ~]# eksctl version
0.190.0
```

Aws user has been configured here through access key and secret key generated in the Aws console.

Install EKSCTL and KUBECTL

Create EKS cluster with a name, region, subnets and without nodes.

```
release/stable.txt)/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 53.7M  100 53.7M    0     0  12.8M      0  0:00:04  0:00:04 --:--:-- 12.8M
[root@kubernetes ~]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@kubernetes ~]# kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
[root@kubernetes ~]# eksctl create cluster --name myg-cluster --region ap-southeast-1 --version 1.29 --vpc-public-subnets subnet-0d9b69111e1d6682
ubnet-04aafe39cb446a1a1 --without-nodegroup
2024-09-21 03:56:25 [ℹ]  eksctl version 0.190.0
2024-09-21 03:56:25 [ℹ]  using region ap-southeast-1
2024-09-21 03:56:25 [✔]  using existing VPC (vpc-00d163923f3f54b32) and subnets (private:map[] public:map[ap-southeast-1a:{subnet-04aafe39cb446a1
ap-southeast-1a 172.31.32.0/20 0 } ap-southeast-1b:{subnet-0d9b69111e1d66829 ap-southeast-1b 172.31.16.0/20 0 }])
2024-09-21 03:56:25 [!]  custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuratio
f VPC/subnets
2024-09-21 03:56:25 [ℹ]  using Kubernetes version 1.29
2024-09-21 03:56:25 [ℹ]  creating EKS cluster "myg-cluster" in "ap-southeast-1" region with
2024-09-21 03:56:25 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-southeast-1 -
uster=myg-cluster'
2024-09-21 03:56:25 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "myg-cluster" in
p-southeast-1"
2024-09-21 03:56:25 [ℹ]  CloudWatch logging will not be enabled for cluster "myg-cluster" in "ap-southeast-1"
2024-09-21 03:56:25 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --re
n=ap-southeast-1 --cluster=myg-cluster'
2024-09-21 03:56:25 [ℹ]  default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
```

Create node group

```
[root@kubernates ~]# eksctl create nodegroup   --cluster pro-cluster   --region ap-south-1   --name my-node-group   --node-ami-family Ubuntu2004   -
-node-type t2.small   --subnet-ids subnet-0b85fc64db0bb9227,subnet-0f87e8992a2e28e8f   --nodes 3   --nodes-min 2   --nodes-max 4   --ssh-access   --
ssh-public-key /root/.ssh/id_rsa.pub
2024-09-21 10:04:18 [ℹ]  will use version 1.29 for new nodegroup(s) based on control plane version
2024-09-21 10:04:20 [ℹ]  nodegroup "my-node-group" will use "ami-0ab6dcbf35da05038" [Ubuntu2004/1.29]
2024-09-21 10:04:20 [ℹ]  using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-pro-cluster-nodegroup-my-node-group-81:5c:62:2a:10:ce:ef:9e:89:8f:0
f:db:05:0d:09:35"
2024-09-21 10:04:21 [ℹ]  1 existing nodegroup(s) (my-node-group) will be excluded
2024-09-21 10:04:21 [ℹ]
2 sequential tasks: { fix cluster compatibility, no tasks
}
2024-09-21 10:04:21 [ℹ]  checking cluster stack for missing resources
2024-09-21 10:04:21 [ℹ]  cluster stack has all required resources
2024-09-21 10:04:21 [ℹ]  no tasks
2024-09-21 10:04:21 [✔]  created 0 nodegroup(s) in cluster "pro-cluster"
2024-09-21 10:04:21 [✔]  created 0 managed nodegroup(s) in cluster "pro-cluster"
2024-09-21 10:04:21 [ℹ]  checking security group configuration for all nodegroups
2024-09-21 10:04:21 [ℹ]  all nodegroups have up-to-date cloudformation templates
```

Create a deployment file

root@tomcatserver:~ | root@githubserver:~ | root@jenkins:~ | root@docker:~ | root@ip-172-31-44-151:~

```
piVersion: apps/v1
ind: Deployment
etadata:
 name: gm-deployment
 labels:
   app: regapp
pec:
 replicas: 2
 selector:
   matchLabels:
     app: regapp
 template:
   metadata:
     labels:
       app: regapp
   spec:
     containers:
     - name: regapp
       image: 039612846272.dkr.ecr.ap-southeast-1.amazonaws.com/project-devops:latest
       imagePullPolicy: Always
       ports:
       - containerPort: 8080
```

Create a service file

```
iVersion: v1
d: Service
tadata:
ame: my-service
abels:
  app: webapp
c:
elector:
  app: webapp
orts:
  - port: 8080
    targetPort: 8080
ype: LoadBalancer
```

Apply the both files

Kubectl apply -f deployment.yaml

Kubectl apply -f service.yaml

Check the deployment and services

```
[root@kubernates ~]# kubectl get deployment
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
my-project-deployment   2/2     2            2           17m
[root@kubernates ~]# kubectl get service
NAME                    TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)
AGE
kubernetes              ClusterIP      10.100.0.1       <none>                                                                   443/TCP
24h
my-project-deployment   LoadBalancer   10.100.67.24     a40c802597f1448999986e330c1e3714-1187773118.ap-south-1.elb.amazonaws.com   8080:32211/TCP
23h
my-service              LoadBalancer   10.100.105.146   a19c5d1b1adf343f58419b884309def6-2055312661.ap-south-1.elb.amazonaws.com   8080:32279/TCP
23h
```

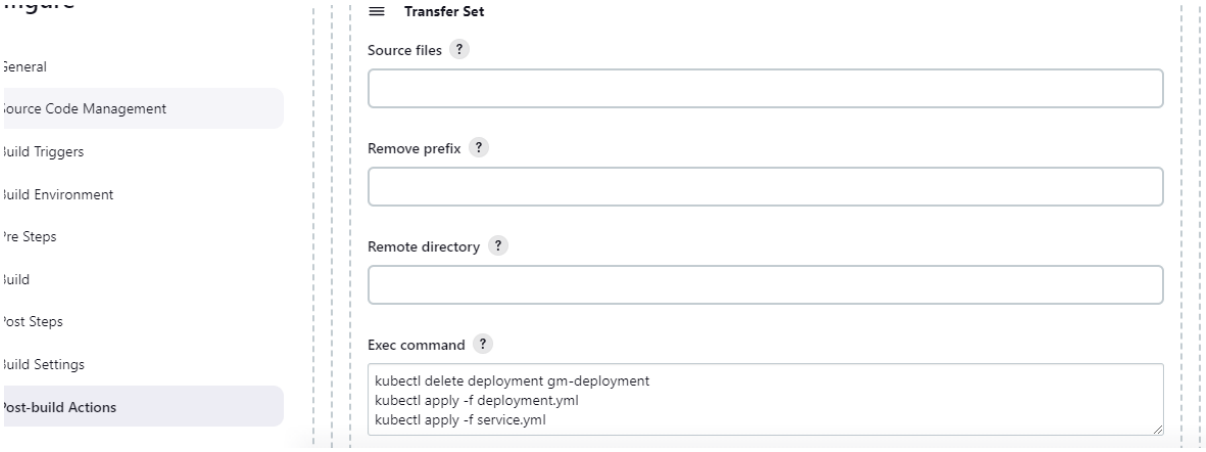
Copy ssh public keys of Jenkins into Kubernetes and Kubernetes into Jenkins

```
[root@kubernates .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzsQ17OsWUQebP3KYnH/RH/yLdysQbHRk1ZH6DaYGZa7DtF/rD4Y8okiO+NbegBYKnWSvmyA7lSK29hjtZHgiGStgybA
HVlu8BnIGZiJeLJHzGSeMYwIpztHIFzltMXjodEFH7Jlpe8vA8Hh2OfhEKkRwi4A1dwCggpb+QaE3XbXpZ+ugDFzKNKpcy44jMUBeCIDOb0AidyC8+pdQrxFhtVGwRFxpTShJDWYGlJrVPMRYxJL
nZTqU+vvtdiM/M6CCpJKCLeOAEtj371OqwTqn1PSy1NISxAeCb3leLbl56fHle+2Nn8Um6MQQXVsG1T+RAMOG87asZOy8/6HzAXVbz project
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCMipExiZ5omSs0pxnxPp7Q+UOMQZb/RemDM+sk8xN9pX0kTiSRTm4cYZ5DpM4wAIHsUVhs+u2PVF+TIWZ5ufjwPREqU3lqgEU3JJjgQJHXnqU4
dUXOFT6h0pAfWxxqM+JtSD5RiNaANqx5owPgiQILtnORz+OqqFWScFffJ8S/JxJeRMQFOzU0GeQTK/d5CeYZljcaT5E3Hy4GMrzh9fmgxCFOYCEXdo+tTcFeFHlSm4zKZcfotsniPpocVvbz9kPj
EJShisvAeJZik4RImS8l6fvWWf6nIJOCU62fQ/XzojhSrJkafZnnex0R5nB3y3CoRM5CKXIaww8wFiCmtZ5o4JWbEh2+ubXljOlOn+nDgbr7evKz8r4qINBY1yE1gAJvr7v9DLCcq/qNLX2Z6h1o
KNxJvJka4FkYN7V3k/S3VwLuT0KgcyLM6XXsp64ERgwx40STwjTkENpw5W+jgvJ0qK+xHdPYIjNW6B+ILrh09qspLrD7GFAXsiHBPhJ939E= root@jenkins.example.com
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDIEjXmG0ooGmOi8eUgydOI+YGch39cYP0wbAp0JpUy3f1OuqIt0ne1AcB8panINRuaLHeNBQ166DZEnRBoM61MPfaiy2nNSJCNJcDYnHege/0w
dp/qiQGvNGT/5f28H1Q814GOFajIysoPsPSPGhIesPIEu+8nNBbYEjZDLFJMy7fuCk618MHa1hsA+dIc4YKMflUt05DyL5mIRp5hqInAIJ9kZLmITproqYQ5zXj7amyAoRVqMV9OUR2He8ypsd4s
2i0er/PlKF9vk0ghjMjQtj9J305X3W0AusLEtlaWQZlcf88rBIXkD5fMZHbQ3EAOy+v4nVmyNI4/sBhKs7KlHRN944FOignGc8EWaiOwhHet5S8Ns638R8XPuFrGclAXR/QwWNENnMTXCRbFtd8h
Ok4ADdF9eUdZRodd7PMVGXIBHMqWHSdDV9SiXC29NLc1VWWQ88F0O03UTCNxoCB4DtRYrCxH1HvGPEO7sGIdYfxqR8ZzDwMHPkghf7hteY8= root@ip-172-31-40-81.ap-south-1.compute
.internal
```

```
[root@jenkins .ssh]# cat authorized_keys
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;s
leep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCzsQ17OsWUQebP3KYnH/RH/yLdysQbHRk1ZH6DaYGZa7DtF/rD4Y8okiO+NbegBYKnWSvmyA7lSK29hjtZHgiGStgybA
HVlu8BnIGZiJeLJHzGSeMYwIpztHIFzltMXjodEFH7Jlpe8vA8Hh2OfhEKkRwi4A1dwCggpb+QaE3XbXpZ+ugDFzKNKpcy44jMUBeCIDOb0AidyC8+pdQrxFhtVGwRFxpTShJDWYGlJrVPMRYxJL
nZTqU+vvtdiM/M6CCpJKCLeOAEtj371OqwTqn1PSy1NISxAeCb3leLbl56fHle+2Nn8Um6MQQXVsG1T+RAMOG87asZOy8/6HzAXVbz project
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCMipExiZ5omSs0pxnxPp7Q+UOMQZb/RemDM+sk8xN9pX0kTiSRTm4cYZ5DpM4wAIHsUVhs+u2PVF+TIWZ5ufjwPREqU3lqgEU3JJjgQJHXnqU4
dUXOFT6h0pAfWxxqM+JtSD5RiNaANqx5owPgiQILtnORz+OqqFWScFffJ8S/JxJeRMQFOzU0GeQTK/d5CeYZljcaT5E3Hy4GMrzh9fmgxCFOYCEXdo+tTcFeFHlSm4zKZcfotsniPpocVvbz9kPj
EJShisvAeJZik4RImS8l6fvWWf6nIJOCU62fQ/XzojhSrJkafZnnex0R5nB3y3CoRM5CKXIaww8wFiCmtZ5o4JWbEh2+ubXljOlOn+nDgbr7evKz8r4qINBY1yE1gAJvr7v9DLCcq/qNLX2Z6h1o
KNxJvJka4FkYN7V3k/S3VwLuT0KgcyLM6XXsp64ERgwx40STwjTkENpw5W+jgvJ0qK+xHdPYIjNW6B+ILrh09qspLrD7GFAXsiHBPhJ939E= root@jenkins.example.com
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbNdb50bPXHx3e9q0O3F9/w98xa9CVL8lSnZwXRdEnememw6T7t+1DnSEYTB0dpa6Bk6vtJhRQxJB/Gywx9sJCLUM5lGRj9w0eTgjYgR/+CBd/
jWUyZ5uLfPEpbe5Qz7D2SuY7nWbR3IZNtmVwsRIDSr23Oy3e8gE6euPwTZPnU67Z0wkxyeBur7dDGI/AzIJ5MMQZD3ewX79274flp7evn9NyCM7tDXVq0JRG3ud9V+zTHD0+IscuWY+dxMne85HV
oupPzfpRcx2lAuRDvlz66FLGvx82WpU67jrawiEF25tZ+HpxNzqdGUGoDNwTAuClFeytP4FZMCZ0pCCwrLWnhlFTtkmAFVu/s4Wk55O26zqYnc8+O6p8O9AJeaoF4btmwaccS7pQMGfFcOmFe4+b
1CZDn+TYiiVyDRPR8D6ebqgDXLoUn8xb090orDfWmSalEztPgYoehJ2fmx5ODKYgt76W9GhdBVI2jBqdhEOlRNwO5C9tP2vFUreQs9LYrTE= root@docker.example.com

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDIEjXmG0ooGmOi8eUgydOI+YGch39cYP0wbAp0JpUy3f1OuqIt0ne1AcB8panINRuaLHeNBQ166DZEnRBoM61MPfaiy2nNSJCNJcDYnHege/0w
dp/qiQGvNGT/5f28H1Q814GOFajIysoPsPSPGhIesPIEu+8nNBbYEjZDLFJMy7fuCk618MHa1hsA+dIc4YKMflUt05DyL5mIRp5hqInAIJ9kZLmITproqYQ5zXj7amyAoRVqMV9OUR2He8ypsd4s
2i0er/PlKF9vk0ghjMjQtj9J305X3W0AusLEtlaWQZlcf88rBIXkD5fMZHbQ3EAOy+v4nVmyNI4/sBhKs7KlHRN944FOignGc8EWaiOwhHet5S8Ns638R8XPuFrGclAXR/QwWNENnMTXCRbFtd8h
Ok4ADdF9eUdZRodd7PMVGXIBHMqWHSdDV9SiXC29NLc1VWWQ88F0O03UTCNxoCB4DtRYrCxH1HvGPEO7sGIdYfxqR8ZzDwMHPkghf7hteY8= root@ip-172-31-40-81.ap-south-1.compute
.internal
```

Add Kubernetes server into Jenkins

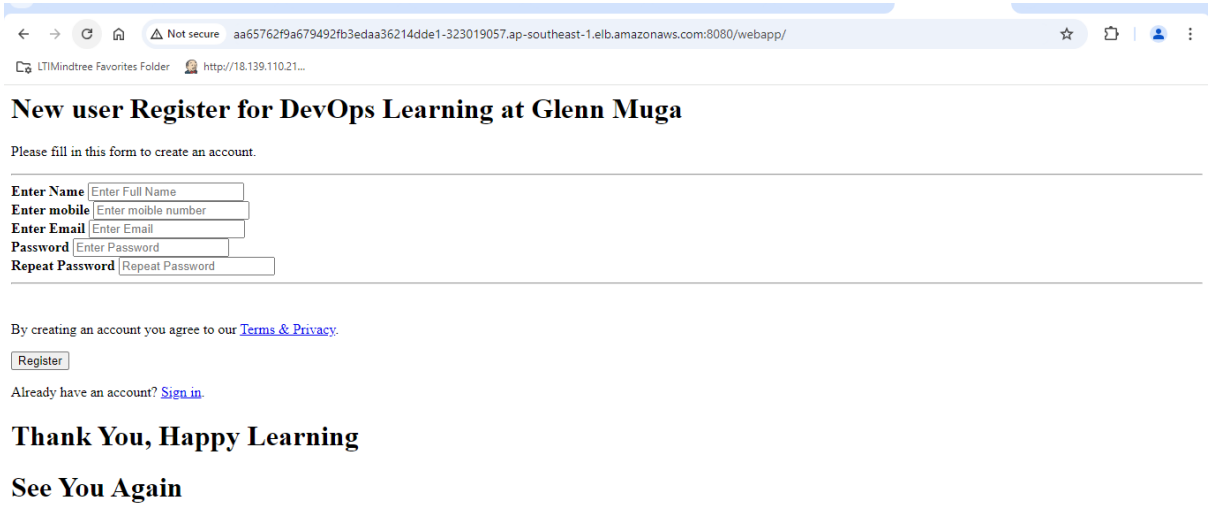
Add build artifacts



Now build the job

Expose application using external Ip address

It is successfully deployed and running



In this way we have automated the whole process. If there are any modifications in the jsp file from github jenkins will automatically build the project and the changes will be reflected on the live webpage automatically.

If there are any errors in the modification the build will be failed and the webpage will not be updated. It will show the last successful built which will help in preventing our webpage from crashing.

By- Glenn Muga.