

# MILESTONE PROJECT

– ARUNVEL RADHAKRISHNAN (10835661)

## Question:

**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.**

## Tools required:

- **Git** - Tracks code changes locally; version control system
- **GitHub** - Hosts code repositories online; enables collaboration and integration.
- **Jenkins** - Automates build, test, and deployment; core of the CI/CD pipeline.
- **Maven** - Builds Java projects; manages dependencies and packaging.
- **Tomcat** - Deploys Java web apps, lightweight application server.
- **Docker** - Containerizes applications; ensures consistent environments.
- **AWS ECR** - Stores Docker images securely; used for Kubernetes deployments.
- **Kubernetes (EKS)** - Manages containers at scale; automates deployment and scaling.

## Phase 1: CI/CD with GitHub, Jenkins, Maven & Tomcat

- Set up Jenkins and install required plugins.
- Configure Maven and Git in Jenkins.
- Connect Jenkins to GitHub to pull code.
- Build Java app using Maven.
- Deploy the WAR file to Tomcat server.
- Test the deployment via browser.

## Phase 2: CI/CD with Docker Integration

- Install Docker on Jenkins host.
- Write a Docker file to containerize the Java app.

- Build Docker image using Jenkins.
- Run the container and expose the app.
- Automate the entire process in Jenkins pipeline.

### **Phase 3: Push Docker Image to AWS ECR & Deploy to Kubernetes**

- Create an ECR repository in AWS.
- Authenticate Docker with AWS CLI.
- Push Docker image to ECR.
- Set up EKS (Elastic Kubernetes Service).
- Write Kubernetes deployment and service YAML files.
- Deploy app using ECR image in Kubernetes.

### **Phase 4: Final Deployment & Validation**

- Trigger Jenkins pipeline to build and deploy.
- Validate pod creation in Kubernetes.
- Get service IP using kubectl get svc.
- Access the app in browser using service IP.

Confirm successful end-to-end deployment.

-Create an instance in AWS, Name of the instance: Arunvel-project

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the instance name is set to 'arunvel-project'. The 'Summary' pane on the right shows 1 instance being launched. The 'Software Image (AMI)' section lists 'Amazon Linux 2023 AMI 2023.7.2...' as the selected AMI. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. The 'Launch instance' button is visible at the bottom right.

-Instance type: t2. micro

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Instance type' step, the instance type is selected as 't2.micro'. The 'Summary' pane on the right shows 1 instance being launched. The 'Software Image (AMI)' section lists 'Amazon Linux 2023 AMI 2023.7.2...' as the selected AMI. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. The 'Launch instance' button is visible at the bottom right.

-create a security group: project-sg

The screenshot shows the AWS EC2 Instances Launch an instance page. In the 'Subnet' section, a new subnet is selected with the ID 'subnet-081441a5b98d5b6f0'. Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security groups)' section, 'Create security group' is selected. The security group name is set to 'project-sg'. Below it, a note states: 'This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_~!@#\$%^&\_!\$\*'. The 'Description' field contains 'launch-wizard-1 created 2025-06-27T05:59:43.366Z'. In the 'Inbound Security Group Rules' section, there is one rule: 'Security group rule 1 (TCP, 22, 0.0.0.0/0)'. The 'Launch instance' button is highlighted in orange at the bottom right.

-Open terminal move to the location where the key is available

->downloads

-Paste the SSH key

-Login to root

-set the hostname (arunvel-project)

```
root@ip-172-31-32-96: ~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-100-24-16-246.compute-1.amazonaws.com
The authenticity of host 'ec2-100-24-16-246.compute-1.amazonaws.com (100.24.16.246)' can't be established.
ED25519 key fingerprint is SHA256:hmD2iTA4CqCTcMRkmsSzNP7LZcUS37ySov/DKY9VtI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-24-16-246.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

               _#
  _\_\_ #####      Amazon Linux 2023
  ___ \_\#####\
  ~~ \#\#\#
  ~~  \#/
  ~~   V~`-->
  ~~~ /`-
  ~~~ /`-
  ~~~ /`-
  ~~~ /`-
[ec2-user@ip-172-31-32-96 ~]$ sudo su -
[root@ip-172-31-32-96 ~]# hostnamectl set-hostname arunvel-project.example.com
[root@ip-172-31-32-96 ~]# bash
[root@arunvel-project ~]# |
```

-Install git

```
root@ip-172-31-32-96:~| x + | Windows PowerShell - ENG IN 09:54
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

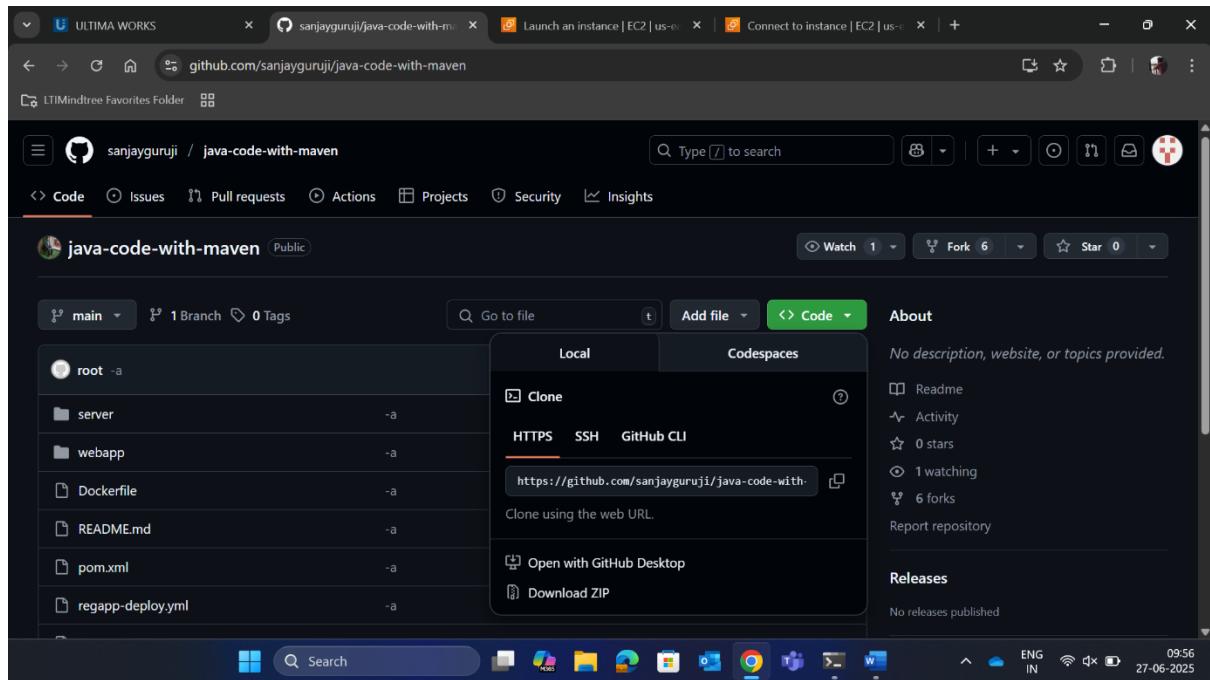
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd ..\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-100-24-16-246.compute-1.amazonaws.com
The authenticity of host 'ec2-100-24-16-246.compute-1.amazonaws.com (100.24.16.246)' can't be established.
ED25519 key fingerprint is SHA256:hdD2iT4CqCtMRkmsSzNP7LZcUS37ySov/DKY9VvT.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-24-16-246.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ ##### Amazon Linux 2023
~~ \#####\
~~ \###|
~~ /#| _> https://aws.amazon.com/linux/amazon-linux-2023
~~ \~`-->
~~~~
~~ .-` /` /
~~ .-` /` /
~~ /` /` /
[ec2-user@ip-172-31-32-96 ~]$ sudo su -
[root@ip-172-31-32-96 ~]# hostnamectl set-hostname arunvel-project.example.com
[root@ip-172-31-32-96 ~]# bash
[root@arunvel-project ~]# yum install git -y,
```

## -Cloning repository (java-code-with-maven)

-Copy the https code from the repository



-Create a directory (project-x)

-In that directory we are going to clone the git repository

```
[root@ip-172-31-32-96:/project] + - x
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : git-core-2.47.1-1.amzn2023.0.3.x86_64 1/1
Installing : git-core-doc-2.47.1-1.amzn2023.0.3.noarch 1/8
Installing : perl-lib-0.65-477.amzn2023.0.7.x86_64 2/8
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 3/8
Installing : perl-File-Find-1.37-477.amzn2023.0.7.noarch 4/8
Installing : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 5/8
Installing : perl-Git-2.47.1-1.amzn2023.0.3.noarch 6/8
Installing : git-2.47.1-1.amzn2023.0.3.x86_64 7/8
Installing : git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Running scriptlet: git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Verifying : git-2.47.1-1.amzn2023.0.3.x86_64 1/8
Verifying : git-core-2.47.1-1.amzn2023.0.3.x86_64 2/8
Verifying : git-core-doc-2.47.1-1.amzn2023.0.3.noarch 3/8
Verifying : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 4/8
Verifying : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8
Verifying : perl-Git-2.47.1-1.amzn2023.0.3.noarch 6/8
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/8
Verifying : perl-lib-0.65-477.amzn2023.0.7.x86_64 8/8

Installed:
git-2.47.1-1.amzn2023.0.3.x86_64          git-core-2.47.1-1.amzn2023.0.3.x86_64          git-core-doc-2.47.1-1.amzn2023.0.3.noarch
perl-Error-1.0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.47.1-1.amzn2023.0.3.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64

Complete!
[root@arunvel-project ~]# mkdir /project-x
[root@arunvel-project ~]# cd /project-x/
[root@arunvel-project project-x]#
```

-Paste the https link in the git clone command

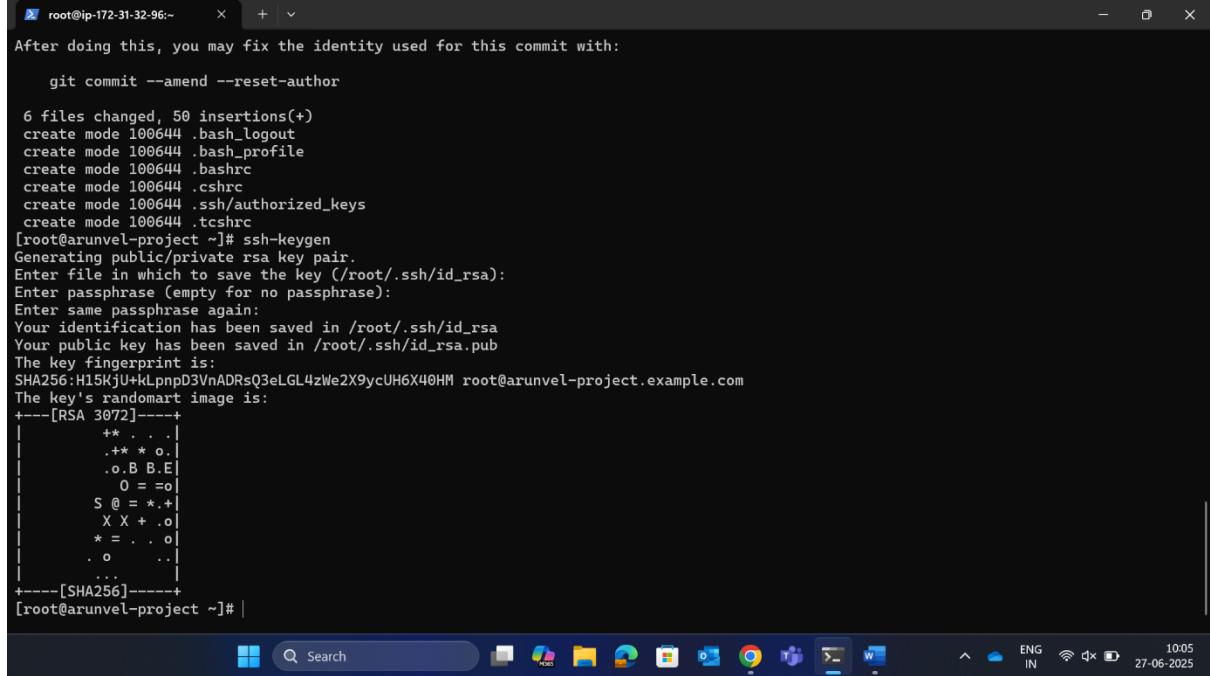
```
root@ip-172-31-32-96:/project ~% Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : git-core-2.47.1-1.amzn2023.0.3.x86_64 1/8
Installing : git-core-doc-2.47.1-1.amzn2023.0.3.noarch 2/8
Installing : perl-lib-0.65-477.amzn2023.0.7.x86_64 3/8
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/8
Installing : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8
Installing : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 6/8
Installing : perl-Git-2.47.1-1.amzn2023.0.3.noarch 7/8
Installing : git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Running scriptlet: git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Verifying : git-2.47.1-1.amzn2023.0.3.x86_64 1/8
Verifying : git-core-2.47.1-1.amzn2023.0.3.x86_64 2/8
Verifying : git-core-doc-2.47.1-1.amzn2023.0.3.noarch 3/8
Verifying : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 4/8
Verifying : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8
Verifying : perl-Git-2.47.1-1.amzn2023.0.3.noarch 6/8
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/8
Verifying : perl-lib-0.65-477.amzn2023.0.7.x86_64 8/8

Installed:
git-2.47.1-1.amzn2023.0.3.x86_64          git-core-2.47.1-1.amzn2023.0.3.x86_64          git-core-doc-2.47.1-1.amzn2023.0.3.noarch
perl-Error-1.0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.47.1-1.amzn2023.0.3.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64

Complete!
[root@arunvel-project ~]# mkdir /project-x
[root@arunvel-project ~]# cd /project-x/
[root@arunvel-project project-x]# git clone https://github.com/sanjayguruji/java-code-with-maven.git
```

-Generate key (SSH-keygen)

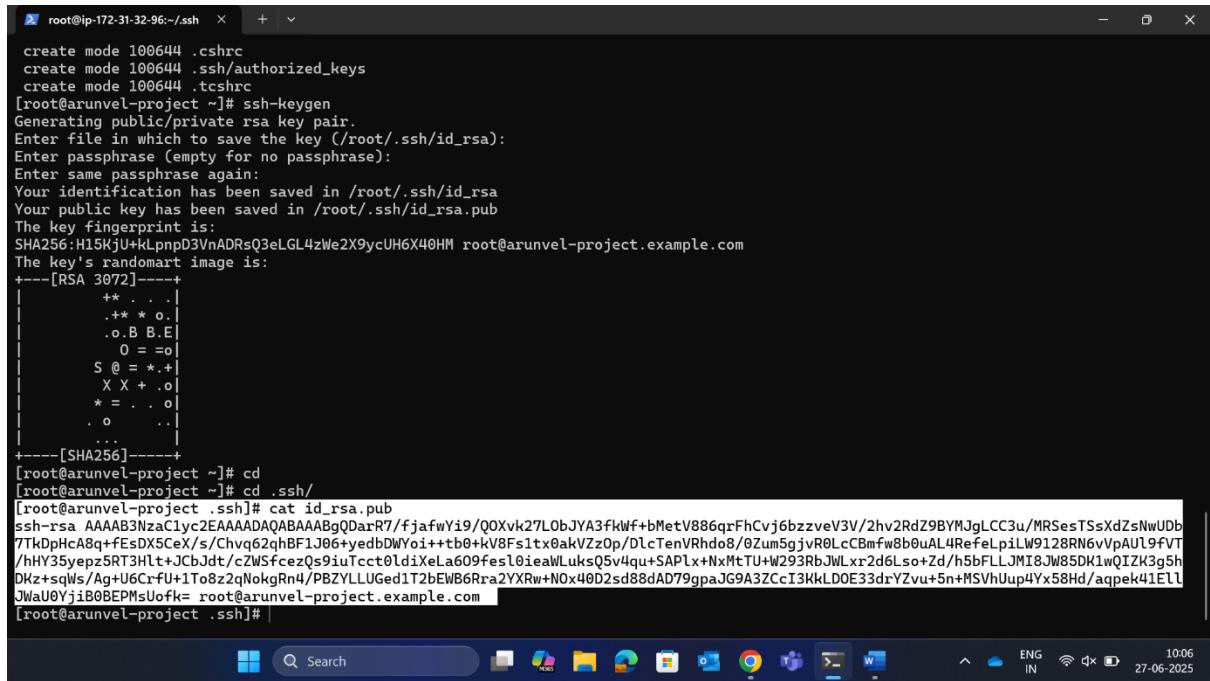
-Key generated



```
root@ip-172-31-32-96:~ After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author

6 files changed, 50 insertions(+)
create mode 100644 .bash_logout
create mode 100644 .bash_profile
create mode 100644 .bashrc
create mode 100644 .cshrc
create mode 100644 .ssh/authorized_keys
create mode 100644 .tcschrc
[root@arunvel-project ~]# 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:H15KjU+kLpnpD3VnADRsQ3eLGL4zWe2X9ycUH6X40HM root@arunvel-project.example.com
The key's randomart image is:
+---[RSA 3072]---+
| +* . . .
| .++ * o |
| .o.B B.E |
| O = =o |
| S @ = *.+ |
| X X + .o |
| * = . . o |
| . o . . |
| ... |
+---[SHA256]---+
[root@arunvel-project ~]#
```

-Copy the key



```
root@ip-172-31-32-96:~/ssh After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author

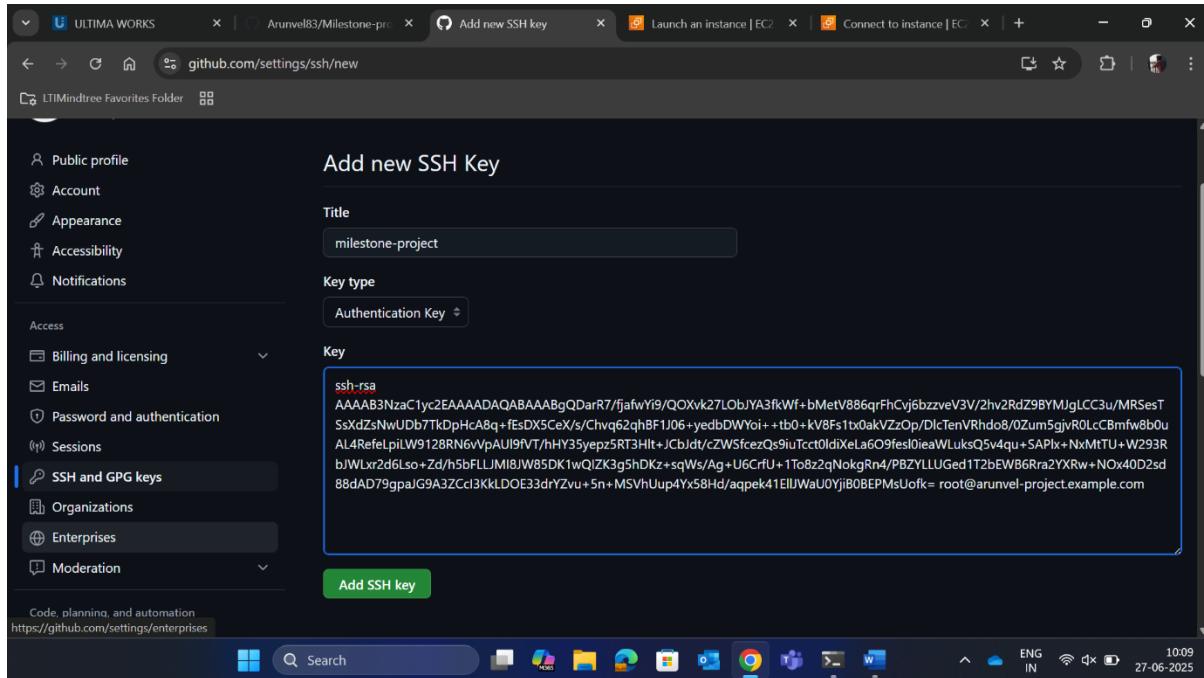
create mode 100644 .cshrc
create mode 100644 .ssh/authorized_keys
create mode 100644 .tcschrc
[root@arunvel-project ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:H15KjU+kLpnpD3VnADRsQ3eLGL4zWe2X9ycUH6X40HM root@arunvel-project.example.com
The key's randomart image is:
+---[RSA 3072]---+
| +* . . .
| .++ * o |
| .o.B B.E |
| O = =o |
| S @ = *.+ |
| X X + .o |
| * = . . o |
| . o . . |
| ... |
+---[SHA256]---+
[root@arunvel-project ~]# cd
[root@arunvel-project ~]# cd .ssh/
[root@arunvel-project .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDaR7/fjafwYi9/Q0Xvk27LobJYA3fkWf+bMetV886qrFhCvj6bzzeV3V/2hv2RdZ9BYMjgLCC3u/MRSesTSsXdZsNwUDb
7ThDphcA8q+feSDX5CeX/s/Chvg62ghBF1J06+yedbfDWYoi++tb0+kV8Fs1tx0akVZzOp/DlcTenVRhd8/0Zum5gjvR0LcBmfw8b0uAL4RefeLp1Lw9128RN6vPau19fVT
/hHY35yepz5RT3Hlt+JCbJdt/cZWsfcezQs9iuTcc0ldiXeLa609fes0ieawLuksQ5v4qu+SAPlx+NxMtTU+w293RbJWLxr2d6ls0+Zd/h5bFLJMI8JW85DK1wQIZK3g5h
Dkz+sqW5/Ag+U6CrfU+1To8z2qNokgRn4/PBZYLLUGed1T2bEWB6Rra2YXRw+N0x40D2sd88dAD79gpaJG9A3ZCcI3KkLDOE33drYZvu+5n+MSVhUp4Yx58Hd/aqpek41Ell
JWaU0YjiB0BEPMsUofk= root@arunvel-project.example.com
[root@arunvel-project .ssh]#
```

-Go to accounts > settings

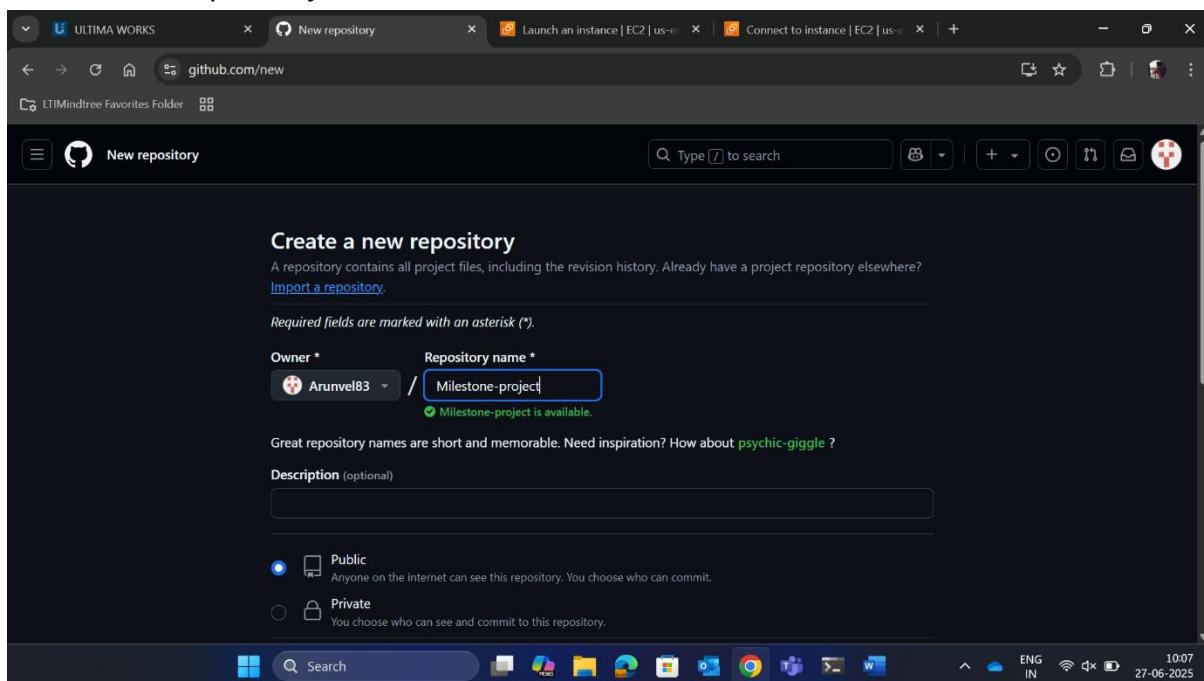
-Go to SSH and CPG keys

-Create a new SSH key

-Paste the key that we copied from the terminal (SSH ID)

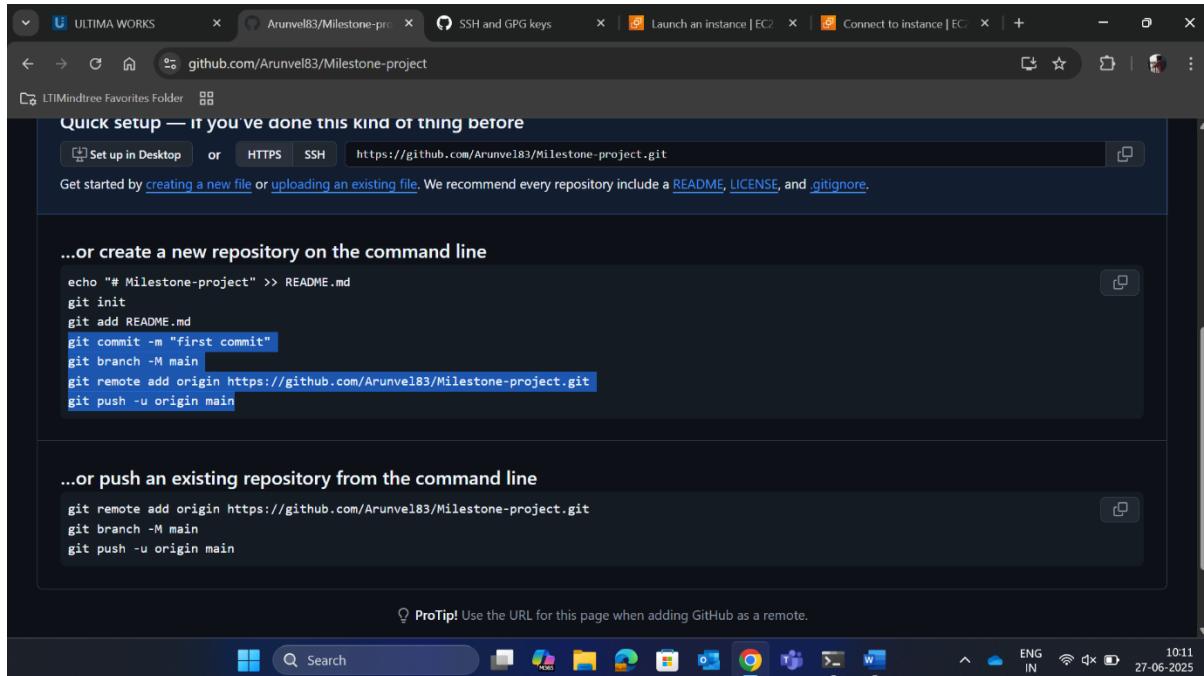


-Create new repository



-We need to push the cloned repo to the newly created repo

-Need to use these commands



The screenshot shows a browser window with multiple tabs open. The active tab is for a GitHub repository named 'Arunvel83/Milestone-project'. The page displays instructions for quick setup or creating a new repository from the command line. It includes a code block for initializing a local repository and pushing it to GitHub:

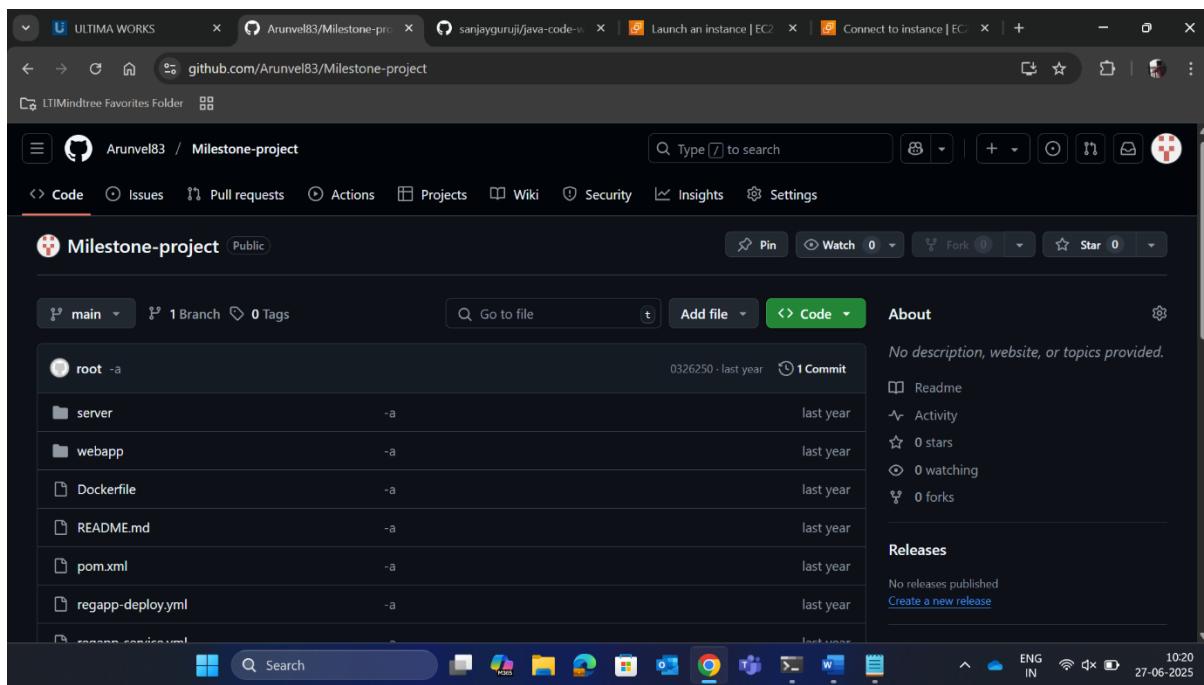
```
echo "# Milestone-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Arunvel83/Milestone-project.git
git push -u origin main
```

Below this, another code block shows how to push an existing repository from the command line:

```
git remote add origin https://github.com/Arunvel83/Milestone-project.git
git branch -M main
git push -u origin main
```

A note at the bottom says: "ProTip! Use the URL for this page when adding GitHub as a remote."

-Successfully pulled to my repo

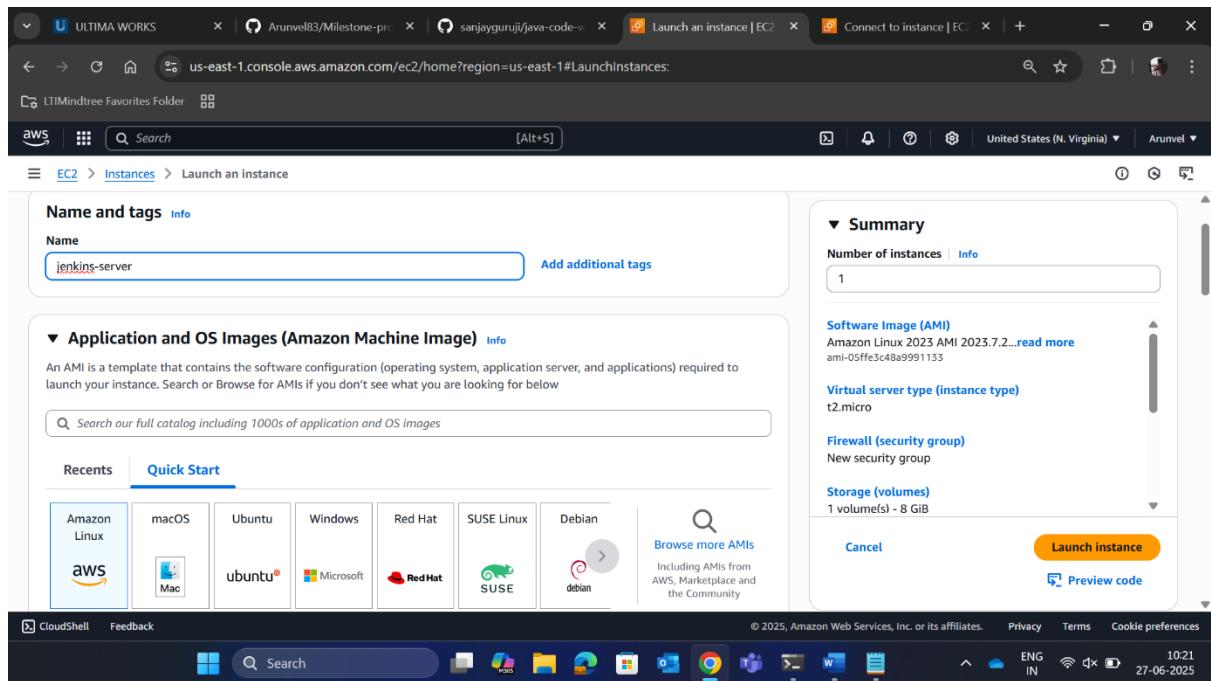


-Need to create new instance for jenkins

-We need to give t2.medium

-Choose the subnet,security group

-Need a storage of 12gib

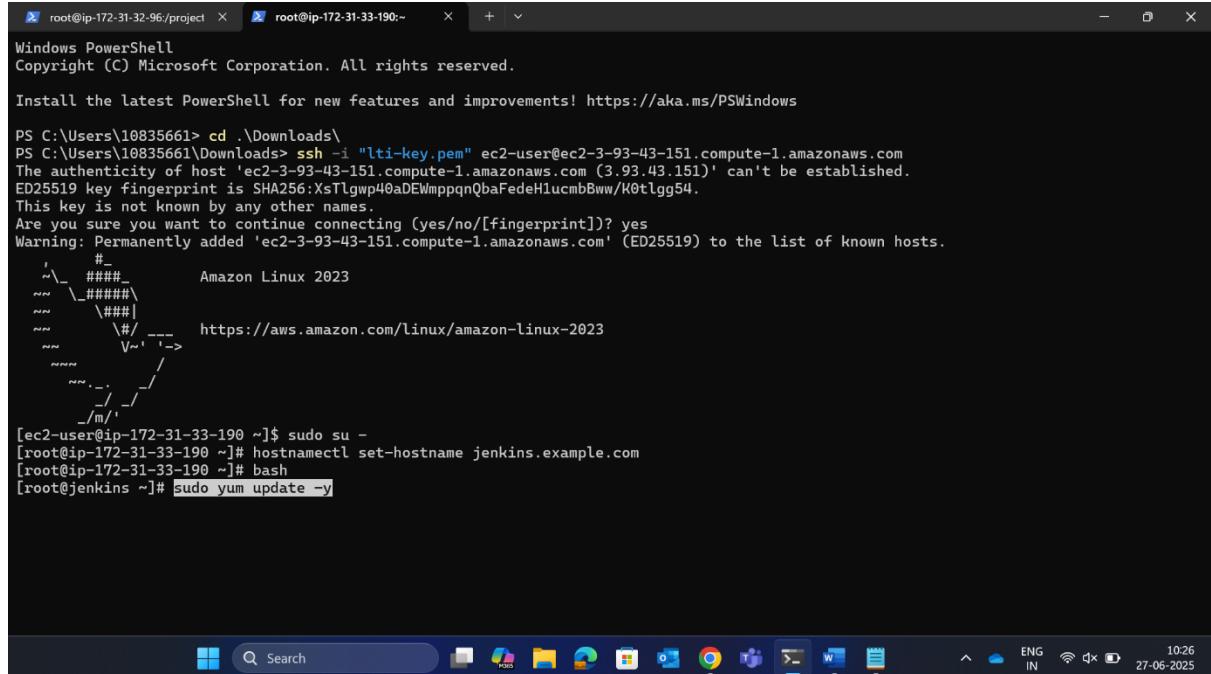


-Connect to the terminal

```
PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "ulti-key.pem" ec2-user@ip-172-31-33-190
The authenticity of host 'ip-172-31-33-190' can't be established.
ED25519 key fingerprint is SHA256:XsTlgwp40aDEWmpqqnQbaFedeH1ucmbBww/K0tlgg54.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ip-172-31-33-190' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-33-190 ~]$ |
```

## -Update all packages



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

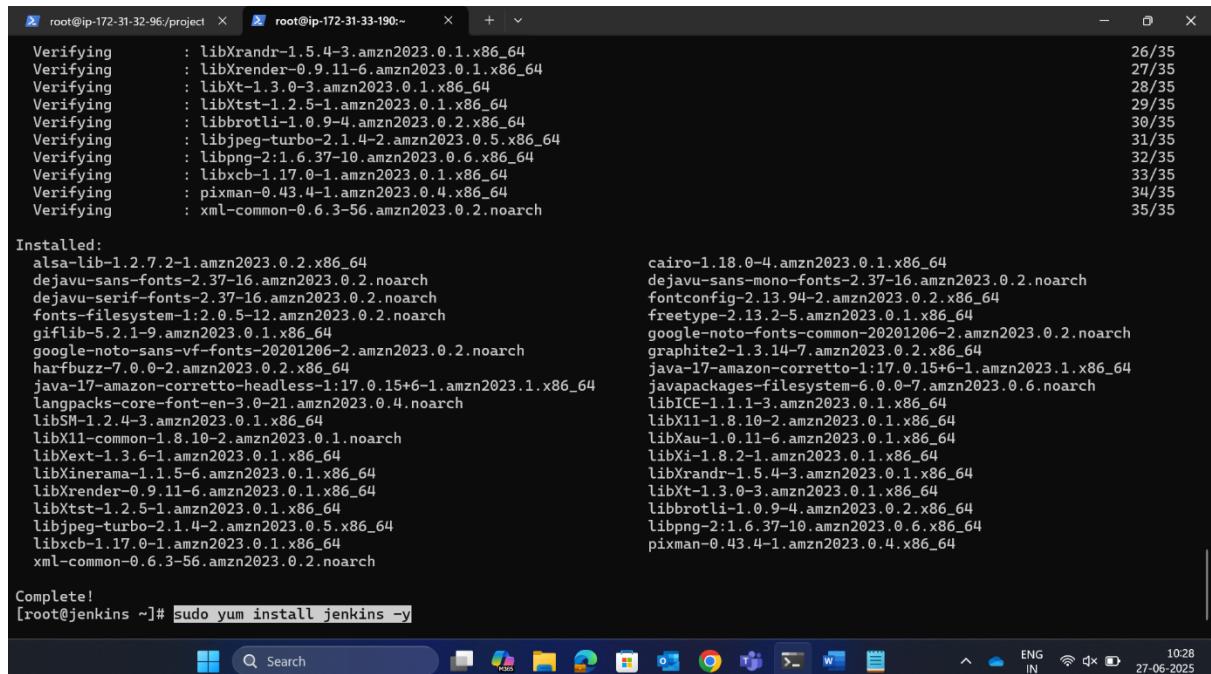
PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-3-93-43-151.compute-1.amazonaws.com
The authenticity of host 'ec2-3-93-43-151.compute-1.amazonaws.com (3.93.43.151)' can't be established.
ED25519 key fingerprint is SHA256:XstLgwp40aDEWmpqnQbaFedeHlucmbBww/K0tLgg54.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-93-43-151.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ #####          Amazon Linux 2023
~~ \_#####\_
~~ \###|
~~ \#/ ___  https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' '->
~~ /_
~~ .-' _/
~/ _/
/m/'

[ec2-user@ip-172-31-33-190 ~]$ sudo su -
[root@ip-172-31-33-190 ~]# hostnamectl set-hostname jenkins.example.com
[root@ip-172-31-33-190 ~]# bash
[root@jenkins ~]# sudo yum update -y

10:26 IN 27-06-2025
```

## -Install jenkins



```
root@ip-172-31-32-96:/project x root@ip-172-31-33-190:~ x + | ~
Verifying : libXrandr-1.5.4-3.amzn2023.0.1.x86_64 26/35
Verifying : libXrender-0.9.11-6.amzn2023.0.1.x86_64 27/35
Verifying : libXt-1.3.0-3.amzn2023.0.1.x86_64 28/35
Verifying : libXtst-1.2.5-1.amzn2023.0.1.x86_64 29/35
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 30/35
Verifying : libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64 31/35
Verifying : libpng-2.1.6.37-10.amzn2023.0.6.x86_64 32/35
Verifying : libxcb-1.17.0-1.amzn2023.0.1.x86_64 33/35
Verifying : pixman-0.43.4-1.amzn2023.0.4.x86_64 34/35
Verifying : xml-common-0.6.3-56.amzn2023.0.2.noarch 35/35

Installed:
alsa-lib-1.2.7.2-1.amzn2023.0.2.x86_64
dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch
dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch
fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
giflib-5.2.1-9.amzn2023.0.1.x86_64
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
harfbuzz-7.0.0-2.amzn2023.0.2.x86_64
java-17-amazon-corretto-headless-1:17.0.15+6-1.amzn2023.1.x86_64
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch
libSM-1.2.4-3.amzn2023.0.1.x86_64
libX11-common-1.8.10-2.amzn2023.0.1.noarch
libXext-1.3.6-1.amzn2023.0.1.x86_64
libXinerama-1.1.5-6.amzn2023.0.1.x86_64
libXrender-0.9.11-6.amzn2023.0.1.x86_64
libXtst-1.2.5-1.amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
libxcb-1.17.0-1.amzn2023.0.1.x86_64
xml-common-0.6.3-56.amzn2023.0.2.noarch

cairo-1.18.0-4.amzn2023.0.1.x86_64
dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch
fontconfig-2.13.94-2.amzn2023.0.1.x86_64
freetype-2.13.2-5.amzn2023.0.1.x86_64
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
graphite2-1.3.14-7.amzn2023.0.2.x86_64
java-17-amazon-corretto-1:17.0.15+6-1.amzn2023.1.x86_64
javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch
libICE-1.1.1-3.amzn2023.0.1.x86_64
libX11-1.8.10-2.amzn2023.0.1.x86_64
libXau-1.0.11-6.amzn2023.0.1.x86_64
libXi-1.8.2-1.amzn2023.0.1.x86_64
libXrandr-1.5.4-3.amzn2023.0.1.x86_64
libXt-1.3.0-3.amzn2023.0.1.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
libpng-2.1.6.37-10.amzn2023.0.6.x86_64
pixman-0.43.4-1.amzn2023.0.4.x86_64

Complete!
[root@jenkins ~]# sudo yum install jenkins -y

10:28 IN 27-06-2025
```

-Need to configure the security group

-Add the port 8080

The screenshot shows the 'Edit inbound rules' page for a security group. There are two rules listed:

- Security group rule ID: sgr-03c014f8e3180f92c, Type: SSH, Protocol: TCP, Port range: 22, Source: 0.0.0.0/0.
- Custom TCP, Port range: 8080, Source: Any..., Destination: 0.0.0.0/0.

Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

-copy the ip and open it in browser

The screenshot shows the 'Instances' page in the AWS EC2 console. Two instances are listed:

| Name            | Instance ID         | Instance state | Instance type | Status check      | Alarm status  | Availability Zone |
|-----------------|---------------------|----------------|---------------|-------------------|---------------|-------------------|
| jenkins-server  | i-06134b47462bccd2f | Running        | t2.medium     | 2/2 checks passed | View alarms + | us-east-1a        |
| arunvel-project | i-04df5d671d147e655 | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-east-1a        |

The 'jenkins-server' instance is selected. Its details are shown below:

**i-06134b47462bccd2f (jenkins-server)**

**Output ID**

**IP addresses**

| Public IPv4 address        | Private IPv4 addresses | IPv6 addresses |
|----------------------------|------------------------|----------------|
| 3.95.43.151   open address | 172.31.33.190          | -              |

Buttons at the bottom include 'CloudShell', 'Feedback', and 'Save changes'.

-It asks for password

-copy the command



-Paste it with -cat- command to get the password

```
root@ip-172-31-32-96:/project X root@ip-172-31-33-190:~ X + - 0 X
jenkins      noarch          2.504.3-1.1           jenkins          90 M
Transaction Summary
=====
Install 1 Package

Total download size: 90 M
Installed size: 90 M
Downloading Packages:
jenkins-2.504.3-1.1.noarch.rpm
34 MB/s | 90 MB 00:02
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch 1/1
Installing  : jenkins-2.504.3-1.1.noarch 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch 1/1
Verifying   : jenkins-2.504.3-1.1.noarch 1/1

Installed:
jenkins-2.504.3-1.1.noarch

Complete!
[root@jenkins ~]# sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins ~]# sudo systemctl start jenkins
[root@jenkins ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
f20d93d6c3264556947b222348f59b02
[root@jenkins ~]#
```

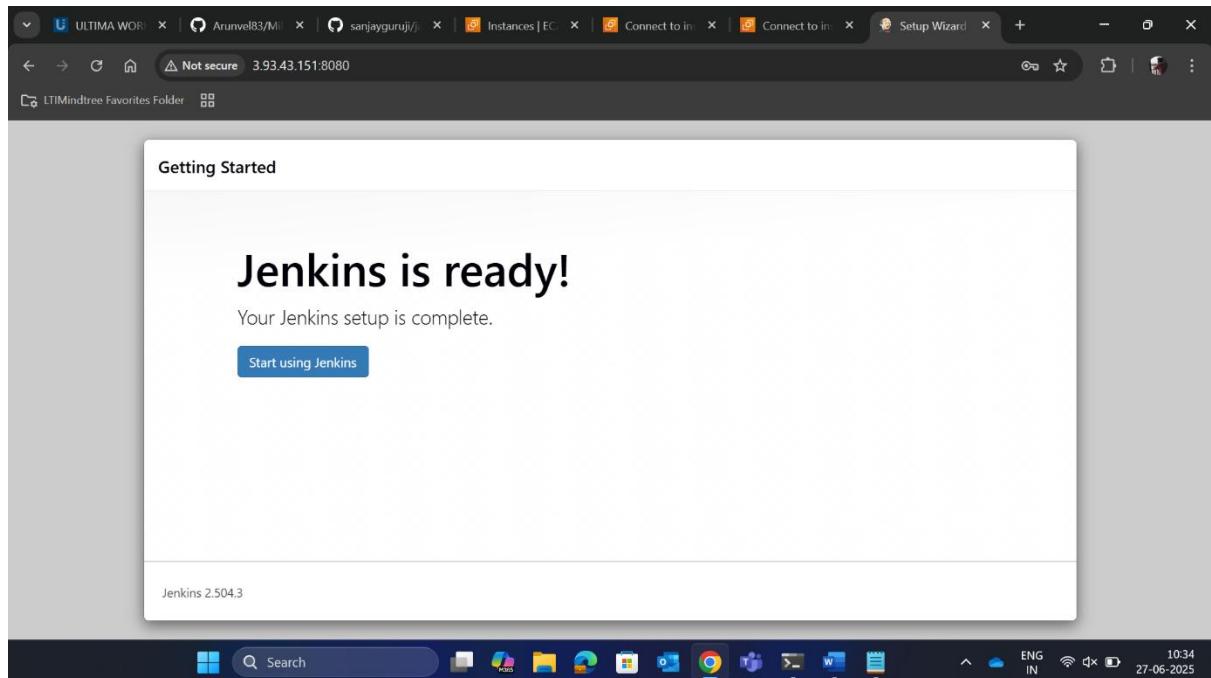
-Copy paste the password

The screenshot shows a web browser window with the URL `3.93.43.151:8080/login?from=%2F`. The page title is "Getting Started" and the main heading is "Unlock Jenkins". A text block states: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:" followed by the path `/var/lib/jenkins/secrets/initialAdminPassword`. Below this, a note says "Please copy the password from either location and paste it below." An "Administrator password" input field contains several dots, indicating a password has been entered. At the bottom right is a blue "Continue" button.

-Sign up

The screenshot shows a web browser window with the URL `3.93.43.151:8080`. The page title is "Getting Started" and the main heading is "Create First Admin User". There are three input fields: "Username" (containing "Arunvel"), "Password" (containing several dots), and "Confirm password" (containing several dots). At the bottom left is the Jenkins version "Jenkins 2.504.3". At the bottom right are two buttons: "Skip and continue as admin" and a blue "Save and Continue" button.

-Jenkins is ready!



-Go to profile > security > add token

-Add new token -copy the generated code

The screenshot shows the Jenkins Security page. On the left sidebar, under the 'Security' section, there is a list of options: Status, Builds, My Views, Account, Appearance, Preferences, Security (which is highlighted in grey), Experiments, and Credentials. The main content area is titled 'Security' and contains a sub-section 'API Token'. It displays a current token labeled 'milestone project' with the value '117a5bd3249a274cd10a0e652daa263fff6'. A warning message below it says '⚠️ Copy this token now, because it cannot be recovered in the future.' There is also a button 'Add new Token'. At the bottom of the page are 'Save' and 'Apply' buttons.

-Paste the code in github to create the webhooks

The screenshot shows the GitHub Webhooks settings page. In the top navigation bar, the URL is 'github.com/Arunvel83/Milestone-project/settings/hooks'. The main content area is titled 'Webhooks' and contains a single entry: 'http://3.93.43.151:8080/github-webhook/push' (push). Below this entry, a message says 'Last delivery was successful.' There are 'Edit' and 'Delete' buttons next to the webhook entry. On the left sidebar, under the 'Code' section, the 'Webhooks' option is selected. Other visible options include General, Access, Collaborators, Moderation options, Branches, Tags, Rules, Actions, Models, and Confluent. The status bar at the bottom indicates the date and time as '27-06-2025 10:45'.

-Need to install plugins in jenkins

The screenshot shows the Jenkins Manage Jenkins page under the System Configuration section. It includes links for System, Tools, Plugins, Nodes, Clouds, and Appearance.

**System Configuration**

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins. This is the selected tab.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**: Configure the look and feel of Jenkins.

**Security**

At the bottom, there is a navigation bar with links for Security, Credentials, and other system status indicators.

-Installing the required plugins

-Restart the jenkins after installing the plugins

The screenshot shows the Jenkins Available Plugins page. The user has selected the "Available plugins" tab. Three specific plugins are highlighted for installation:

- Maven Integration 3.26**: Provides deep integration between Jenkins and Maven. Last released 2 months ago.
- GitHub Integration 0.7.2**: GitHub Integration Plugin for Jenkins. Last released 5 months ago.
- Deploy to container 1.17**: Allows deployment to a container after a build. Last released 1 month ago.

At the bottom, there is a navigation bar with links for Updates, Available plugins, Installed plugins, Advanced settings, Download progress, and other system status indicators.

## -Install maven packages in jenkins machine

```
root@ip-172-31-32-96:/project X root@ip-172-31-33-190:~ X + v
jenkins noarch 2.504.3-1.1 jenkins 90 M
Transaction Summary
=====
Install 1 Package

Total download size: 90 M
Installed size: 90 M
Downloading Packages:
jenkins-2.504.3-1.1.noarch.rpm 34 MB/s | 90 MB 00:02
-----
Total 34 MB/s | 90 MB 00:02

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch 1/1
Installing : jenkins-2.504.3-1.1.noarch 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch 1/1
Verifying   : jenkins-2.504.3-1.1.noarch 1/1

Installed:
jenkins-2.504.3-1.1.noarch

Complete!
[root@jenkins ~]# sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins ~]# sudo systemctl start jenkins
[root@jenkins ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
f20d93d6c3264556947b222348f59b02
[root@jenkins ~]# yum install maven
```

## -Install git

```
root@ip-172-31-32-96:/project X root@ip-172-31-33-190:~ X + v
Verifying : maven-resolver-1:1.7.3-3.amzn2023.0.4.noarch 20/31
Verifying : maven-shared-utils-3.3.4-4.amzn2023.0.3.noarch 21/31
Verifying : maven-wagon-3.4.2-6.amzn2023.0.4.noarch 22/31
Verifying : plexus-cipher-1.8-3.amzn2023.0.3.noarch 23/31
Verifying : plexus-classworlds-2.6.0-10.amzn2023.0.4.noarch 24/31
Verifying : plexus-containers-component-annotations-2.1.0-9.amzn2023.0.4.noarch 25/31
Verifying : plexus-interpolation-1.26-10.amzn2023.0.4.noarch 26/31
Verifying : plexus-sec-dispatcher-2.0-3.amzn2023.0.3.noarch 27/31
Verifying : plexus-utils-3.3.0-9.amzn2023.0.4.noarch 28/31
Verifying : publicsuffix-list-20240212-61.amzn2023.noarch 29/31
Verifying : sisu-1.0.3.4-9.amzn2023.0.4.noarch 30/31
Verifying : slf4j-1.7.32-3.amzn2023.0.4.noarch 31/31

Installed:
apache-commons-cli-1.5.0-3.amzn2023.0.3.noarch
apache-commons-io-1.2.8.0-7.amzn2023.0.5.noarch
atinject-1.0.5-3.amzn2023.0.3.noarch
google-guice-4.2.3-8.amzn2023.0.6.noarch
httpcomponents-client-4.5.13-4.amzn2023.0.4.noarch
jakarta-annotations-1.3.5-13.amzn2023.0.3.noarch
java-17-amazon-corretto-devel-1:17.0.15+6-1.amzn2023.1.x86_64
jsoup-1.16.1-4.amzn2023.0.2.noarch
maven-1:3.8.4-3.amzn2023.0.5.noarch
maven-lib-1:3.8.4-3.amzn2023.0.5.noarch
maven-shared-utils-3.3.4-4.amzn2023.0.3.noarch
plexus-cipher-1.8-3.amzn2023.0.3.noarch
plexus-containers-component-annotations-2.1.0-9.amzn2023.0.4.noarch
plexus-sec-dispatcher-2.0-3.amzn2023.0.3.noarch
publicsuffix-list-20240212-61.amzn2023.noarch
slf4j-1.7.32-3.amzn2023.0.4.noarch

Complete!
[root@jenkins ~]# yum install git -y
```

-Copy the path for maven and java

```
root@ip-172-31-32-96:/project X root@ip-172-31-33-190:~ X + v
Verifying : git-core-2.47.1-1.amzn2023.0.3.x86_64 2/8
Verifying : git-core-doc-2.47.1-1.amzn2023.0.3.noarch 3/8
Verifying : perl-Error-1.0.17029-5.amzn2023.0.2.noarch 4/8
Verifying : perl-File-Find-1.37-477.amzn2023.0.7.noarch 5/8
Verifying : perl-Git-2.47.1-1.amzn2023.0.3.noarch 6/8
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/8
Verifying : perl-lib-0.65-477.amzn2023.0.7.x86_64 8/8

Installed:
git-2.47.1-1.amzn2023.0.3.x86_64      git-core-2.47.1-1.amzn2023.0.3.x86_64      git-core-doc-2.47.1-1.amzn2023.0.3.noarch
perl-Error-1.0.17029-5.amzn2023.0.2.noarch perl-File-Find-1.37-477.amzn2023.0.7.noarch perl-Git-2.47.1-1.amzn2023.0.3.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.7.x86_64

Complete!
[root@jenkins ~]# 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/
[root@jenkins ~]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.141-155.222.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins ~]#
```

-Add the paths in the jenkins > tools

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there's a message about Java 17 end-of-life: "Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)." Below this are three buttons: "Set up agent", "Set up cloud", and "Dismiss".

In the center, there's a prominent message: "Java 17 end of life in Jenkins". It states: "You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to [the documentation](#) for more details." There are "More Info" and "Ignore" buttons next to this message.

Below these messages, the "System Configuration" section is visible, featuring four main items:

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers.
- Plugins**: Add, remove, disable or enable plugins that can extend Jenkins.
- Nodes**: Add, remove, control and monitor the various nodes.

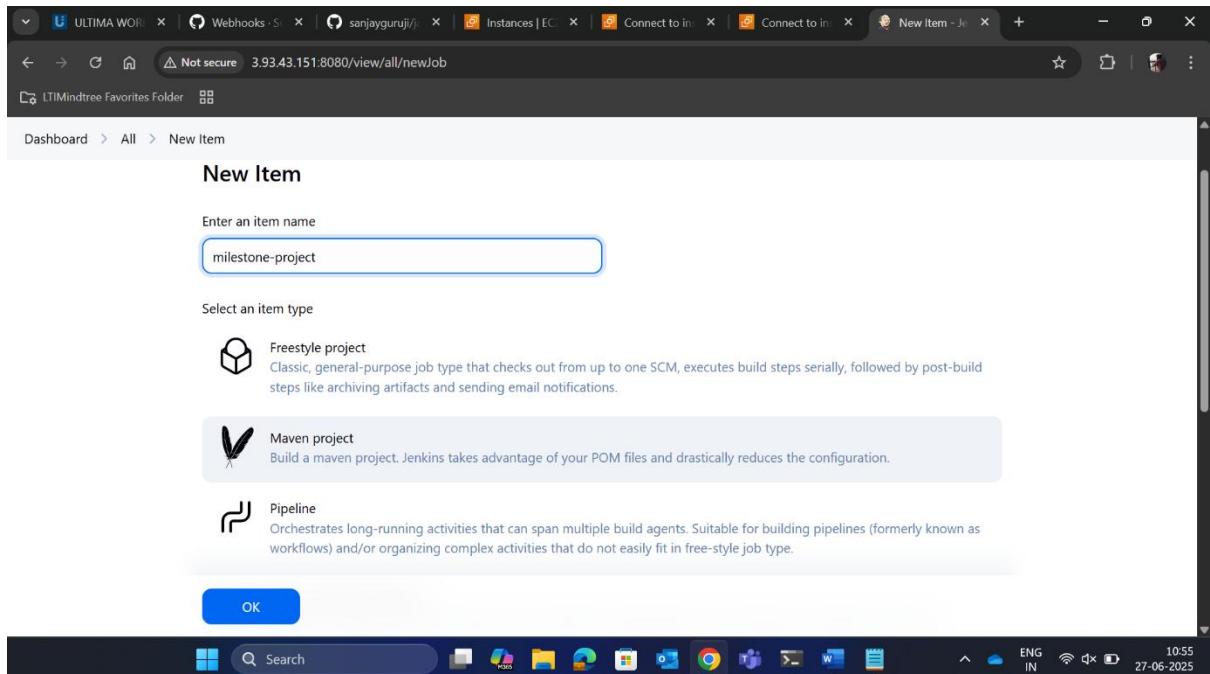
## -Java path

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. A new 'JDK installation' is being configured with the name 'java' and the JAVA\_HOME path set to '/usr/lib/jvm/java-17-amazon-corretto.x86\_64'. The 'Install automatically' checkbox is unchecked. At the bottom are 'Save' and 'Apply' buttons.

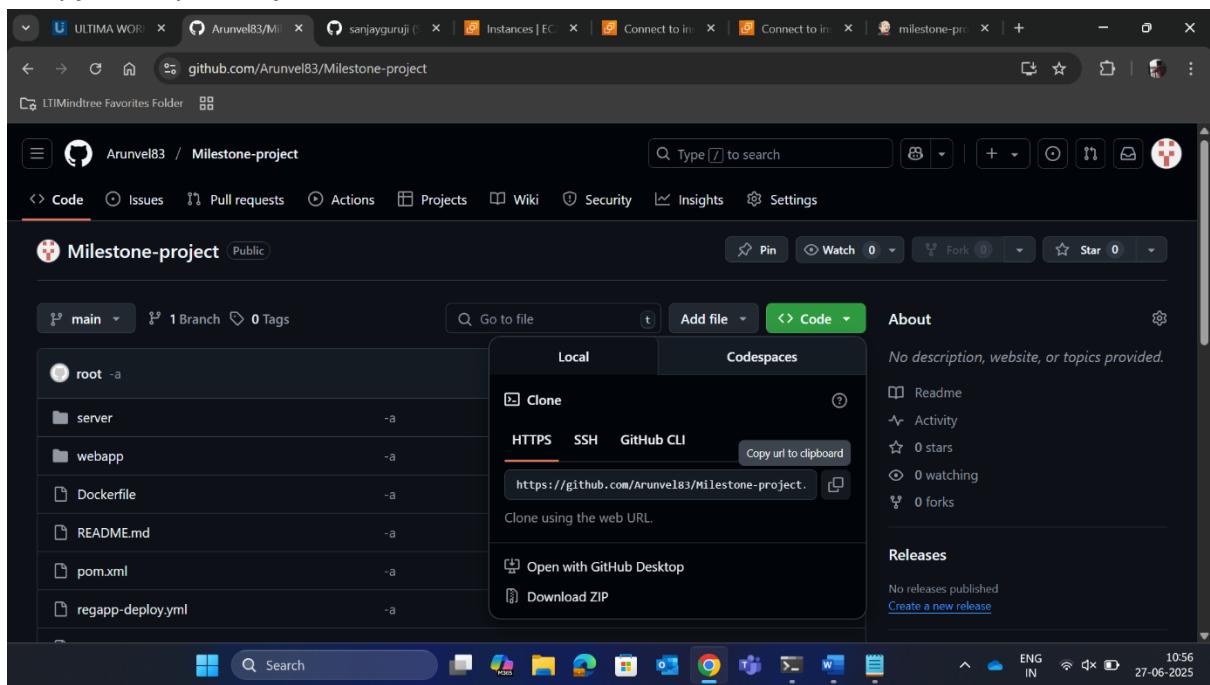
## -Maven path

The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Tools' section. A new 'Maven' configuration is being created with the name 'maven' and the MAVEN\_HOME path set to '/usr/share/maven'. The 'Install automatically' checkbox is unchecked. At the bottom are 'Add Maven', 'Save', and 'Apply' buttons.

-Create new item with maven type



-Copy the repository url



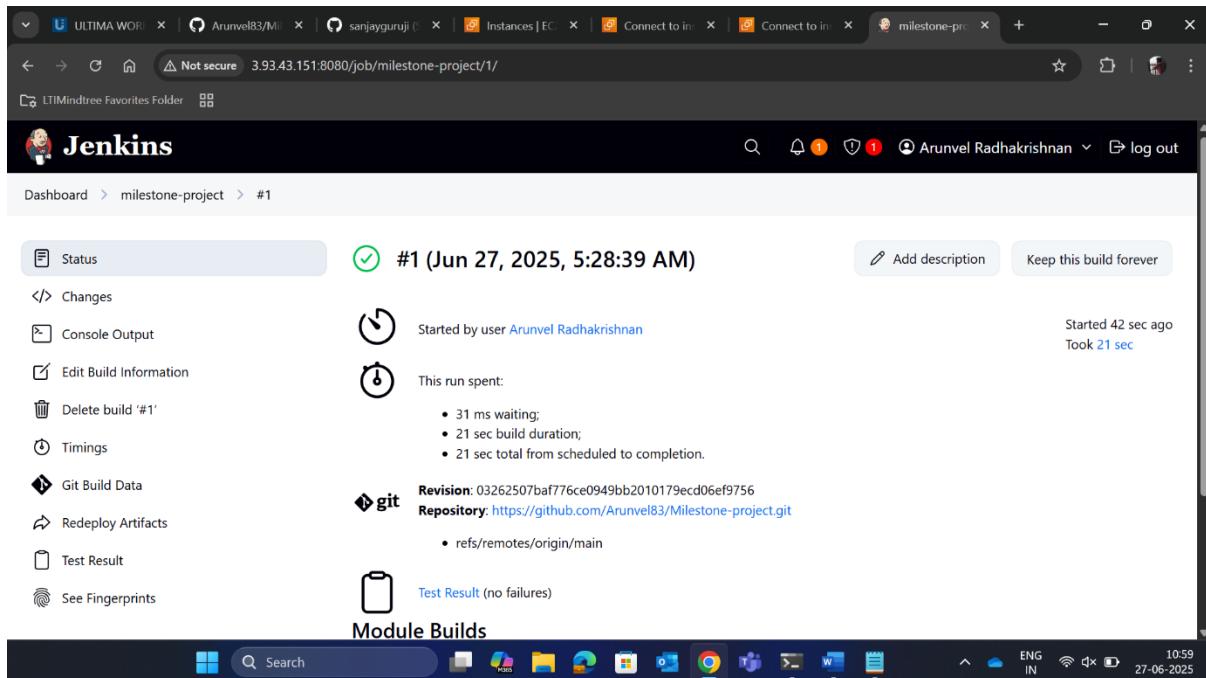
-Paste it in the item which we created- milestone-project > configuration

The screenshot shows a browser window with multiple tabs open. The active tab is '3.93.43.151:8080/job/milestone-project/configure'. The page title is 'Configure'. On the left, there's a sidebar with various Jenkins configuration options: General, Source Code Management (selected), Triggers, Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The main content area is titled 'Source Code Management' and contains instructions to 'Connect and manage your code repository to automatically pull the latest code for your builds.' A 'Git' configuration is shown with a 'Repository URL' input field containing 'https://github.com/Arunvel83/Milestone-project.git' and a 'Credentials' dropdown set to '- none -'. At the bottom are 'Save' and 'Apply' buttons.

-We need to build now

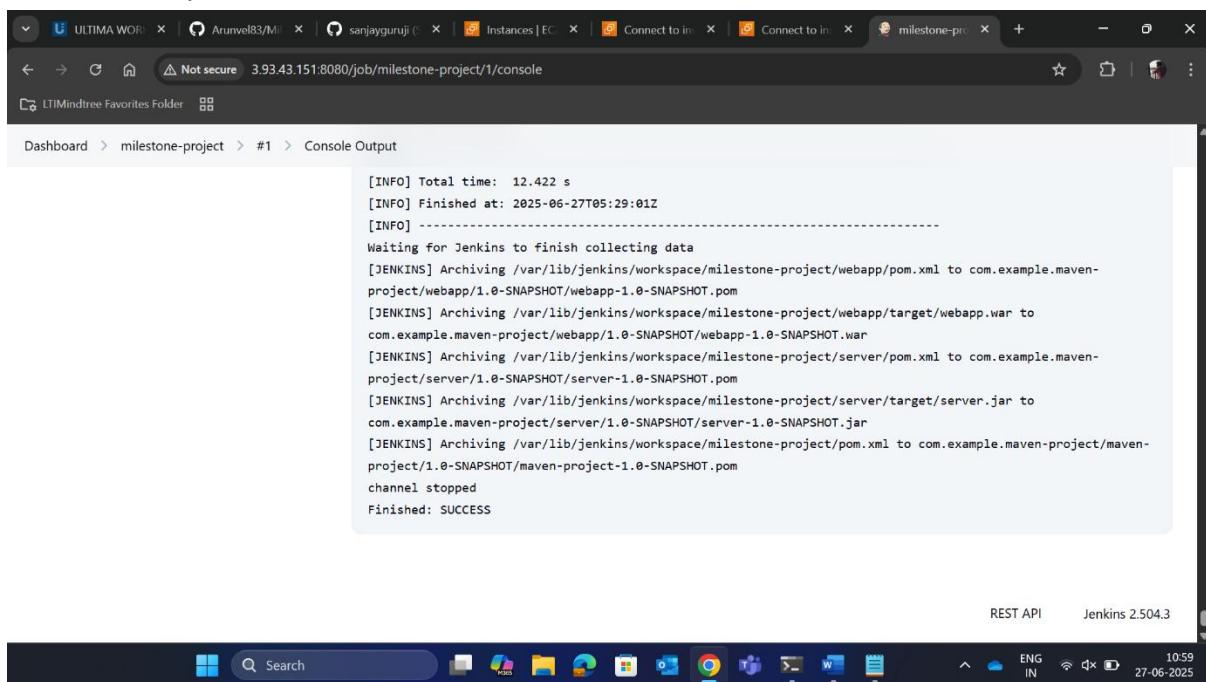
The screenshot shows a browser window with multiple tabs open. The active tab is '3.93.43.151:8080/job/milestone-project/'. The page title is 'Jenkins'. The main content area is titled 'milestone-project' and includes a 'Permalinks' section. On the left, there's a sidebar with links: Status, Changes, Workspace, Build Now (highlighted in blue), Configure, Delete Maven project, Modules, and Rename. Below the sidebar is a 'Builds' section. The bottom of the screen shows a Windows taskbar with various icons and system status indicators.

-Successfully built



The screenshot shows the Jenkins interface for a successful build. The top navigation bar includes tabs for ULTIMA WOR, Arunvel83/Mil, sanjayguruji, Instances | EC, Connect to in, Connect to in, milestone-project, and a plus sign. The main title is "Jenkins". The dashboard shows a green checkmark icon next to "#1 (Jun 27, 2025, 5:28:39 AM)". On the left, a sidebar lists build-related links: Status, Changes, Console Output, Edit Build Information, Delete build '#1', Timings, Git Build Data, Redeploy Artifacts, Test Result, and See Fingerprints. The central panel displays build statistics: Started by user Arunvel Radhakrishnan, Started 42 sec ago, Took 21 sec, and This run spent: 31 ms waiting, 21 sec build duration, 21 sec total from scheduled to completion. It also shows the Revision (03262507baf776ce0949bb2010179ecd06ef9756) and Repository (<https://github.com/Arunvel83/Milestone-project.git>). A "Test Result (no failures)" section is present. Below this is a "Module Builds" section. The bottom of the screen shows a Windows taskbar with various icons and the date/time: 10:59 IN 27-06-2025.

-Console output



The screenshot shows the Jenkins console output for build #1. The top navigation bar is identical to the previous screenshot. The main title is "Console Output". The central panel displays the build logs:

```
[INFO] Total time: 12.422 s
[INFO] Finished at: 2025-06-27T05:29:01Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/milestone-project/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/milestone-project/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/milestone-project/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/milestone-project/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/milestone-project/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```

The bottom of the screen shows a Windows taskbar with various icons and the date/time: REST API Jenkins 2.504.3 10:59 IN 27-06-2025.

-Creating a instance for tomcat-server

-Choose t2.medium

Name: apache-server

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recent AMIs

- Amazon Linux
- macOS
- Ubuntu
- Windows
- Red Hat
- SUSE Linux
- Debian

Browse more AMIs

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2... [read more](#)  
ami-05ffe3c48a9991133

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel [Launch Instance](#) [Preview code](#)

-Connect the instance to terminal

```
root@ip-172-31-32-96:/project ~ root@ip-172-31-33-190:~ ~ root@ip-172-31-45-232:~ + -
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-54-86-99-165.compute-1.amazonaws.com
The authenticity of host 'ec2-54-86-99-165.compute-1.amazonaws.com (54.86.99.165)' can't be established.
ED25519 key fingerprint is SHA256:tER+dbqdlwKI/I452pSXboE3bCb/uki7sM3LqaOBdGs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-86-99-165.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ #####_ Amazon Linux 2023
~~ \_#####\_
~~ \###|
~~ \#/ _--> https://aws.amazon.com/linux/amazon-linux-2023
~~ \~` _-->
~~ .- _- /_
~~ / _/
~/` /_/
~/m/`
```

```
[ec2-user@ip-172-31-45-232 ~]$ sudo su -
[root@ip-172-31-45-232 ~]# hostnamectl set-hostname apache-server.example.com
[root@ip-172-31-45-232 ~]# bash
[root@apache-server ~]#
```

## -Install java

```
PS C:\Users\10835661> cd .\Downloads\  
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-54-86-99-165.compute-1.amazonaws.com  
The authenticity of host 'ec2-54-86-99-165.compute-1.amazonaws.com (54.86.99.165)' can't be established.  
ED25519 key fingerprint is SHA256:tER+dbqdLwKI/I452pSXboE3bCb/uk17sM3LqaOBdGs.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-54-86-99-165.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
#  
~\_ ##### Amazon Linux 2023  
~~ \#####  
~~ \|##|  
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023  
~~ V~' '-->  
~~ /  
~~-. /  
~/ /  
/_/ /  
[ec2-user@ip-172-31-45-232 ~]$ sudo su -  
[root@ip-172-31-45-232 ~]# hostnamectl set-hostname apache-server.example.com  
[root@ip-172-31-45-232 ~]# bash  
[root@apache-server ~]# yum install java*
```

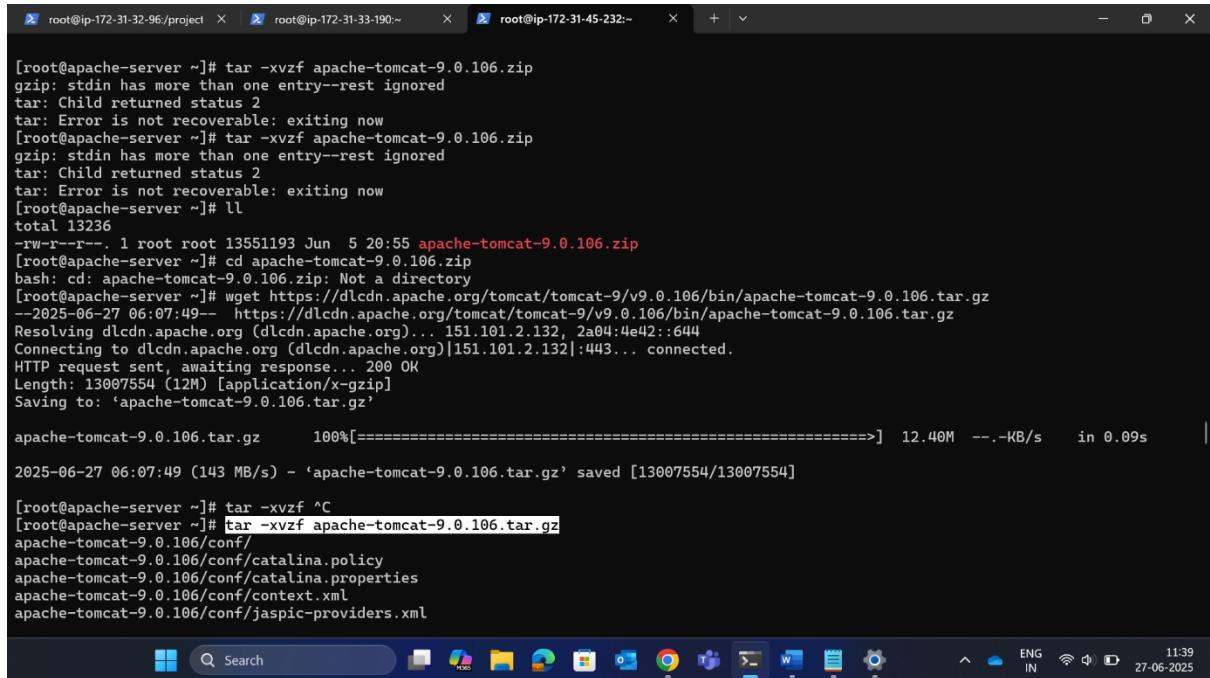
## -Copy the download link of tomacat 9

The screenshot shows a Microsoft Edge browser window with the URL [tomcat.apache.org/download-90.cgi](https://tomcat.apache.org/download-90.cgi). The page displays the Tomcat 9 download section. A context menu is open over a link labeled "Core" under the "Binary Distributions" section. The menu options include:

- Zip (pgp, sha512)
- Open link in new tab
- Open link in new window
- Open link in incognito window
- Aller (pgp, sha512)

The rest of the page includes sections for "Documentation", "Problems?", and links to various Tomcat versions and resources.

## -Extracting files



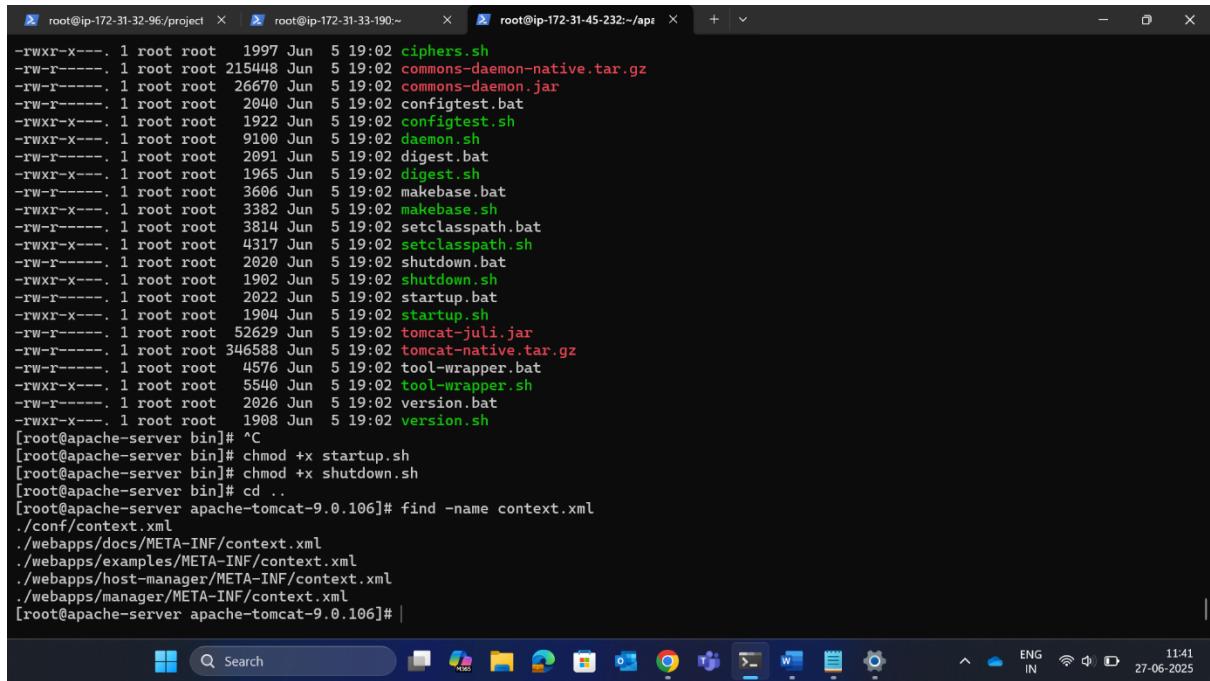
```
[root@apache-server ~]# tar -xvzf apache-tomcat-9.0.106.zip
gzip: stdin has more than one entry--rest ignored
tar: Child returned status 2
tar: Error is not recoverable: exiting now
[root@apache-server ~]# tar -xvzf apache-tomcat-9.0.106.zip
gzip: stdin has more than one entry--rest ignored
tar: Child returned status 2
tar: Error is not recoverable: exiting now
[root@apache-server ~]# ll
total 13236
-rw-r--r--. 1 root root 13551193 Jun  5 20:55 apache-tomcat-9.0.106.zip
[root@apache-server ~]# cd apache-tomcat-9.0.106.zip
bash: cd: apache-tomcat-9.0.106.zip: Not a directory
[root@apache-server ~]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.106/bin/apache-tomcat-9.0.106.tar.gz
--2025-06-27 06:07:49-- https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.106/bin/apache-tomcat-9.0.106.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: 13007554 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.106.tar.gz'

apache-tomcat-9.0.106.tar.gz      100%[=====] 12.40M --.-KB/s   in 0.09s

2025-06-27 06:07:49 (143 MB/s) - 'apache-tomcat-9.0.106.tar.gz' saved [13007554/13007554]

[root@apache-server ~]# tar -xvzf ^
[root@apache-server ~]# tar -xvzf apache-tomcat-9.0.106.tar.gz
apache-tomcat-9.0.106/conf/
apache-tomcat-9.0.106/conf/catalina.policy
apache-tomcat-9.0.106/conf/catalina.properties
apache-tomcat-9.0.106/conf/context.xml
apache-tomcat-9.0.106/conf/jaspic-providers.xml
```

## -Use these commands and find the files



```
-rwxr-x---. 1 root root 1997 Jun  5 19:02 ciphers.sh
-rw-r-----. 1 root root 215448 Jun  5 19:02 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 26670 Jun  5 19:02 commons-daemon.jar
-rw-r-----. 1 root root 2040 Jun  5 19:02 configtest.bat
-rwxr-x---. 1 root root 1922 Jun  5 19:02 configtest.sh
-rwxr-x---. 1 root root 9100 Jun  5 19:02 daemon.sh
-rw-r-----. 1 root root 2091 Jun  5 19:02 digest.bat
-rwxr-x---. 1 root root 1965 Jun  5 19:02 digest.sh
-rw-r-----. 1 root root 3606 Jun  5 19:02 makebase.bat
-rwxr-x---. 1 root root 3382 Jun  5 19:02 makebase.sh
-rw-r-----. 1 root root 3814 Jun  5 19:02 setclasspath.bat
-rwxr-x---. 1 root root 4317 Jun  5 19:02 setclasspath.sh
-rw-r-----. 1 root root 2020 Jun  5 19:02 shutdown.bat
-rwxr-x---. 1 root root 1902 Jun  5 19:02 shutdown.sh
-rw-r-----. 1 root root 2022 Jun  5 19:02 startup.bat
-rwxr-x---. 1 root root 1904 Jun  5 19:02 startup.sh
-rw-r-----. 1 root root 52629 Jun  5 19:02 tomcat-juli.jar
-rw-r-----. 1 root root 346588 Jun  5 19:02 tomcat-native.tar.gz
-rw-r-----. 1 root root 4576 Jun  5 19:02 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 Jun  5 19:02 tool-wrapper.sh
-rw-r-----. 1 root root 2026 Jun  5 19:02 version.bat
-rwxr-x---. 1 root root 1908 Jun  5 19:02 version.sh
[root@apache-server bin]\# ^C
[root@apache-server bin]\# chmod +x startup.sh
[root@apache-server bin]\# chmod +x shutdown.sh
[root@apache-server bin]\# cd ..
[root@apache-server apache-tomcat-9.0.106]\# find -name context.xml
./conf/context.xml
./webapps/META-INF/context.xml
./webapps/examples/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml
./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]\# |
```

-Configure the last three .xml files

```
root@ip-172-31-32-96:/project | root@ip-172-31-33-190:~ | root@ip-172-31-45-232:~/apse | + - X
-rwxr-x---. 1 root root 1997 Jun 5 19:02 ciphers.sh
-rw-r-----. 1 root root 215448 Jun 5 19:02 commons-daemon-native.tar.gz
-rw-r-----. 1 root root 26670 Jun 5 19:02 commons-daemon.jar
-rw-r-----. 1 root root 2040 Jun 5 19:02 configtest.bat
-rw-r-x---. 1 root root 1922 Jun 5 19:02 configtest.sh
-rw-r-x---. 1 root root 9100 Jun 5 19:02 daemon.sh
-rw-r-----. 1 root root 2091 Jun 5 19:02 digest.bat
-rw-r-x---. 1 root root 1965 Jun 5 19:02 digest.sh
-rw-r-----. 1 root root 3606 Jun 5 19:02 makebase.bat
-rw-r-x---. 1 root root 3382 Jun 5 19:02 makebase.sh
-rw-r-----. 1 root root 3814 Jun 5 19:02 setclasspath.bat
-rw-r-x---. 1 root root 4317 Jun 5 19:02 setclasspath.sh
-rw-r-----. 1 root root 2020 Jun 5 19:02 shutdown.bat
-rw-r-x---. 1 root root 1902 Jun 5 19:02 shutdown.sh
-rw-r-----. 1 root root 2022 Jun 5 19:02 startup.bat
-rw-r-x---. 1 root root 1904 Jun 5 19:02 startup.sh
-rw-r-----. 1 root root 52629 Jun 5 19:02 tomcat-juli.jar
-rw-r-----. 1 root root 346588 Jun 5 19:02 tomcat-native.tar.gz
-rw-r-----. 1 root root 4576 Jun 5 19:02 tool-wrapper.bat
-rw-r-x---. 1 root root 5540 Jun 5 19:02 tool-wrapper.sh
-rw-r-----. 1 root root 2026 Jun 5 19:02 version.bat
-rw-r-x---. 1 root root 1908 Jun 5 19:02 version.sh
[root@apache-server bin]# ^C
[root@apache-server bin]# chmod +x startup.sh
[root@apache-server bin]# chmod +x shutdown.sh
[root@apache-server bin]# cd ..
[root@apache-server apache-tomcat-9.0.106]# find -name context.xml
./conf/context.xml
./webapps/docs/META-INF/context.xml
./webapps/examples/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml
./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/examples/META-INF/context.xml
```

The screenshot shows a Linux terminal window with three tabs open. The current tab displays a file listing of Tomcat files, including various scripts and JAR files. Below the listing, commands are being entered to change file permissions for 'startup.sh' and 'shutdown.sh'. The terminal then navigates up to the parent directory and lists all XML configuration files under 'META-INF'. Finally, it uses the 'vi' editor to open the 'context.xml' file located in the 'examples' directory of the Tomcat distribution.

-Need to comment the valve statement

-Do the same for other two .xml files

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<Context>
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
        sameSiteCookies="strict" />
    <!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127.\d+.\d+\.\d+|::1|0:0:0:0:0:1" />-->
</Context>
~
~
~
~
~
~
~
-- INSERT --

```

The screenshot shows a Windows Notepad window displaying the 'context.xml' file. The file contains XML code for a Tomcat context. A specific 'Valve' element is highlighted with a red box, indicating it is the target for modification. The rest of the file includes standard Tomcat licensing and configuration directives.

-Open the tomcat-user.xml file

```
-rwxr-x---. 1 root root 9100 Jun 5 19:02 daemon.sh
-rw-r-----. 1 root root 2091 Jun 5 19:02 digest.bat
-rwxr-x---. 1 root root 1965 Jun 5 19:02 digest.sh
-rwxr-x---. 1 root root 3606 Jun 5 19:02 makebase.bat
-rwxr-x---. 1 root root 3382 Jun 5 19:02 makebase.sh
-rwxr-x---. 1 root root 3814 Jun 5 19:02 setclasspath.bat
-rwxr-x---. 1 root root 4317 Jun 5 19:02 setclasspath.sh
-rw-r-----. 1 root root 2020 Jun 5 19:02 shutdown.bat
-rwxr-x---. 1 root root 1902 Jun 5 19:02 shutdown.sh
-rw-r-----. 1 root root 2022 Jun 5 19:02 startup.bat
-rwxr-x---. 1 root root 1904 Jun 5 19:02 startup.sh
-rw-r-----. 1 root root 52629 Jun 5 19:02 tomcat-juli.jar
-rw-r-----. 1 root root 346588 Jun 5 19:02 tomcat-native.tar.gz
-rw-r-----. 1 root root 4576 Jun 5 19:02 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 Jun 5 19:02 tool-wrapper.sh
-rw-r-----. 1 root root 2026 Jun 5 19:02 version.bat
-rwxr-x---. 1 root root 1908 Jun 5 19:02 version.sh
[root@apache-server bin]# ^C
[root@apache-server bin]# chmod +x startup.sh
[root@apache-server bin]# chmod +x shutdown.sh
[root@apache-server bin]# cd ..
[root@apache-server apache-tomcat-9.0.106]# find -name context.xml
./conf/context.xml
./webapps/docs/META-INF/context.xml
./webapps/examples/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml
./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/examples/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/host-manager/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml" 24L, 1354B written
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# cd conf
[root@apache-server conf]# vi tomcat-users.xml
```

-Add the user with roles

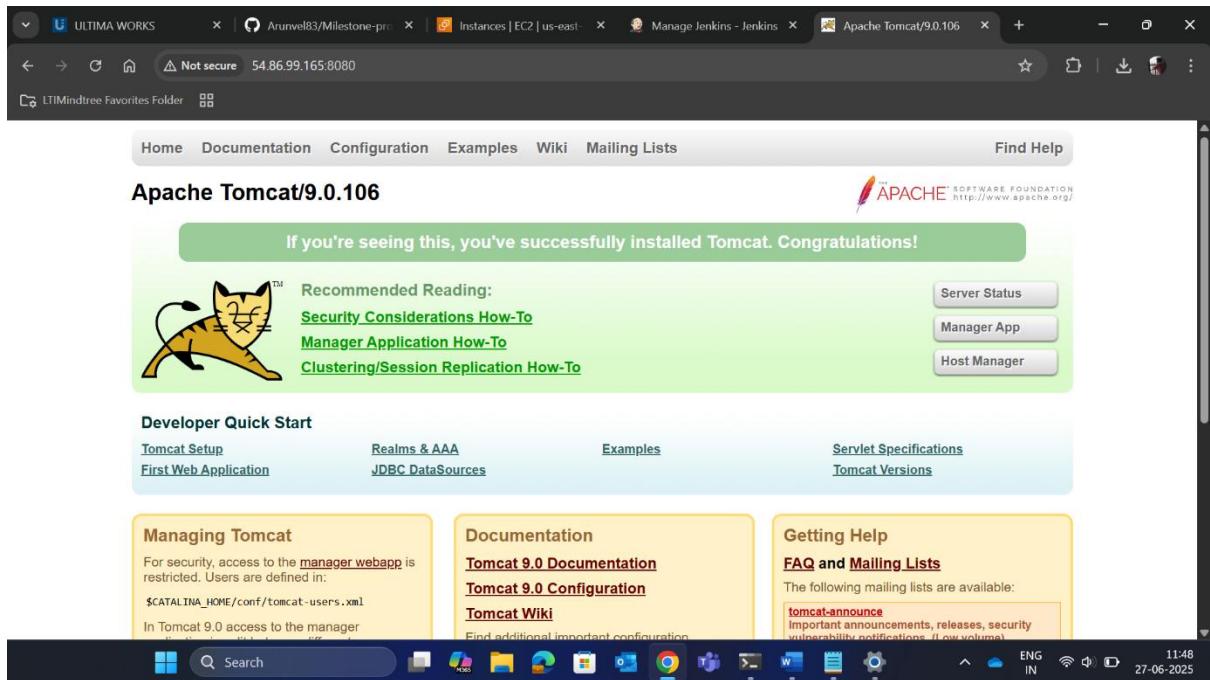
```
The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!...> that surrounds them. You
will also need to set the passwords to something appropriate.
-->
<!--
<user username="admin" password="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!...> that surrounds
them. You will also need to set the passwords to something appropriate.
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="" roles="tomcat"/>
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
-->
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>
<user username="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>
-- INSERT (paste) --
```

## -Start the tomcat server

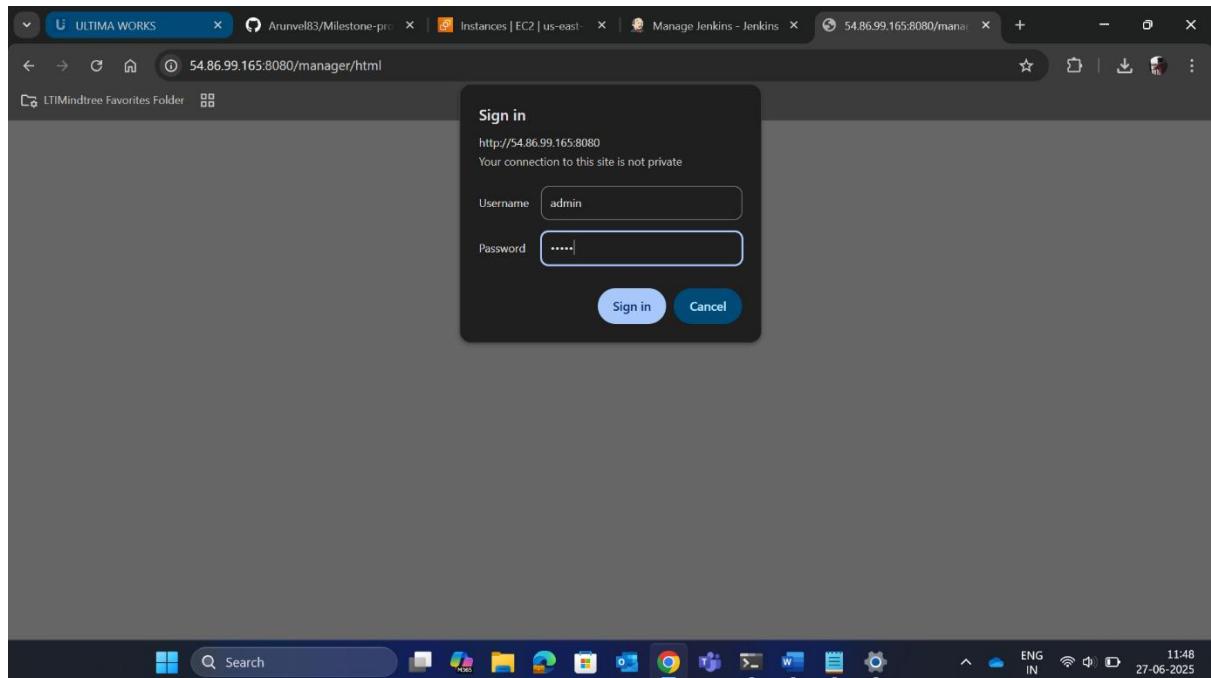
```
root@ip-172-31-32-96:/project ~ root@ip-172-31-33-190:~ root@ip-172-31-45-232:~/apache-tomcat-9.0.106 ~ + - x
-rw-r----. 1 root root 3814 Jun 5 19:02 setclasspath.bat
-rwxr-x---. 1 root root 4317 Jun 5 19:02 setclasspath.sh
-rw-r-----. 1 root root 2020 Jun 5 19:02 shutdown.bat
-rwxr-x---. 1 root root 1902 Jun 5 19:02 shutdown.sh
-rw-r----. 1 root root 2022 Jun 5 19:02 startup.bat
-rwxr-x---. 1 root root 1904 Jun 5 19:02 startup.sh
-rw-r----. 1 root root 52629 Jun 5 19:02 tomcat-juli.jar
-rw-r----. 1 root root 346588 Jun 5 19:02 tomcat-native.tar.gz
-rw-r----. 1 root root 4576 Jun 5 19:02 tool-wrapper.bat
-rwxr-x---. 1 root root 5540 Jun 5 19:02 tool-wrapper.sh
-rw-r----. 1 root root 2026 Jun 5 19:02 version.bat
-rwxr-x---. 1 root root 1908 Jun 5 19:02 version.sh
[root@apache-server bin]# ^C
[root@apache-server bin]# chmod +x startup.sh
[root@apache-server bin]# chmod +x shutdown.sh
[root@apache-server bin]# cd ..
[root@apache-server apache-tomcat-9.0.106]# find -name context.xml
./conf/context.xml
./webapps/docs/META-INF/context.xml
./webapps/examples/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml
./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/examples/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/host-manager/META-INF/context.xml
" ./webapps/host-manager/META-INF/context.xml" 24L, 1354B written
[root@apache-server apache-tomcat-9.0.106]# vi ./webapps/manager/META-INF/context.xml
[root@apache-server apache-tomcat-9.0.106]# cd conf
[root@apache-server conf]# vi tomcat-users.xml
"tomcat-users.xml" 63L, 3128B written
[root@apache-server conf]# cd
[root@apache-server ~]# cd apache-tomcat-9.0.106/
[root@apache-server apache-tomcat-9.0.106]# cd bin/
[root@apache-server bin]# ./startup.sh
```

-Copy the public ip and open the url with :8080

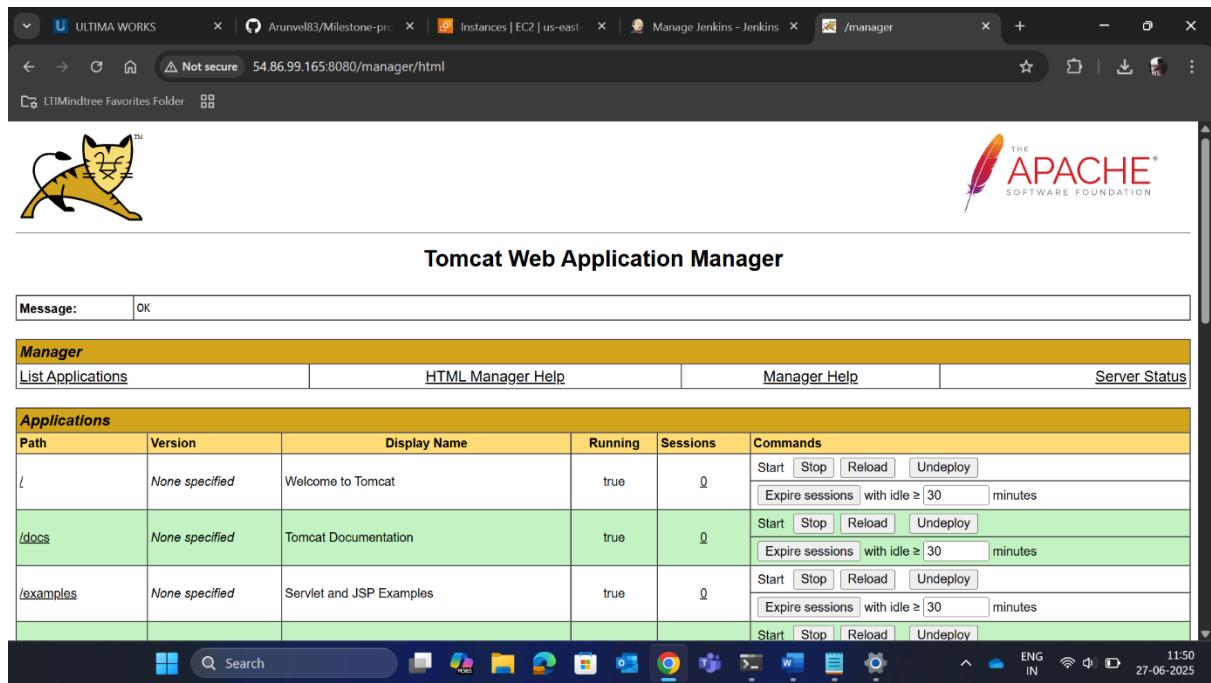
-Tomcat server is successfully deployed



-Go to the manager app > it asks for username and password :admin/admin



-Tomcat web app is successfully working



## -Configure the post build actions

The screenshot shows the Jenkins configuration interface for a project named 'milestone-project'. The left sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Pre Steps, Build, Post Steps, Build Settings (which is selected), and Post-build Actions. The main panel is titled 'Post-build Actions' and contains a section for 'Deploy war/ear to a container'. It includes fields for 'WAR/EAR files' (set to '\*\*/\*.war') and 'Context path' (set to '/'). Below these fields is a 'Containers' section. At the bottom of the panel are 'Save' and 'Apply' buttons.

## -Create credentials and add it :deployer/deployer

The screenshot shows the Jenkins configuration interface for a project named 'milestone-project'. The left sidebar lists various configuration sections. The 'Post-build Actions' section is selected. In the main panel, under the 'Containers' section, there is a 'Tomcat 9.x Remote' configuration. A dropdown menu for 'Credentials' shows two options: '- none -' and 'deployer/\*\*\*\*\*', with 'deployer/\*\*\*\*\*' currently selected. Below the dropdown is a 'Tomcat URL' field and an 'Advanced' button. At the bottom of the panel are 'Save' and 'Apply' buttons.

-Successfully built

The screenshot shows the Jenkins milestone-project dashboard. On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Maven project, Modules, and Rename. The main area has a title 'milestone-project' with a green checkmark icon. Below it is a 'Latest Test Result (no failures)' section with a small icon of a clipboard. To the right is a 'Test Result Trend' chart showing a single green bar from build #1 to #3, indicating all builds passed. A legend at the top of the chart identifies 'Passed' (green), 'Skipped' (grey), and 'Failed' (red). The bottom of the dashboard shows a 'Builds' section with a table header and a 'Builds' table below it. The Windows taskbar at the bottom shows various icons and the date/time: 27-06-2025, 12:00.

-Reload the tomcat url , it is changed

The screenshot shows a browser window with the URL 54.86.99.165:8080. The page title is 'New user Register for DevOps Learning at Virtual TechBox Youtube Channel'. It features a registration form with fields for 'Enter Name', 'Enter mobile', 'Enter Email', 'Password', and 'Repeat Password'. Below the form, a note says 'Please fill in this form to create an account.' There's also a link to 'Terms & Privacy'. At the bottom are 'Register' and 'Sign in' buttons. The Windows taskbar at the bottom shows various icons and the date/time: 27-06-2025, 12:00.

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

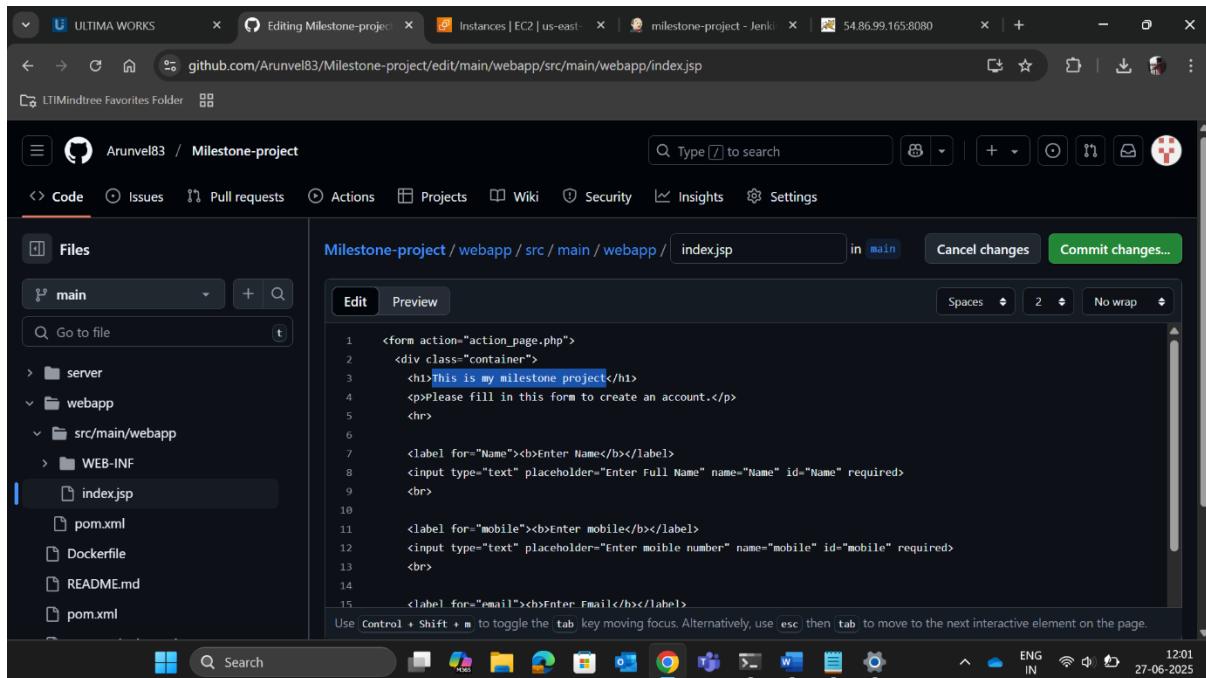
Already have an account? [Sign in](#).

**Thank You, Happy Learning**

**See You Again**

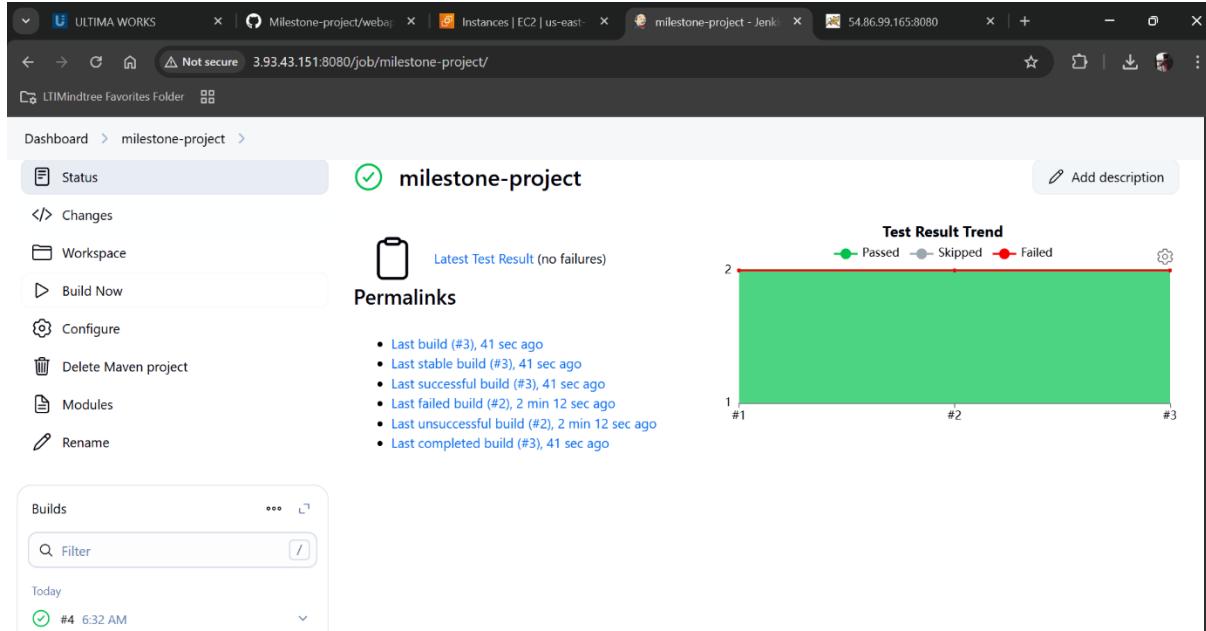


-If we change anything in the github html file after commit we need to build then it reflects in the url



```
<form action="action_page.php">
<div class="container">
<h1>This is my milestone project</h1>
<p>please fill in this form to create an account.</p>
<hr>
<label for="Name"><b>Enter Name</b></label>
<input type="text" placeholder="Enter Full Name" name="Name" id="Name" required>
<br>
<label for="mobile"><b>Enter mobile</b></label>
<input type="text" placeholder="Enter mobile number" name="mobile" id="mobile" required>
<br>
<label for="email"><b>Enter Email</b></label>
```

## -Build



Dashboard > milestone-project >

Status: ✓ milestone-project

Latest Test Result (no failures)

Test Result Trend

- Passed
- Skipped
- Failed

Permalinks

- Last build (#3), 41 sec ago
- Last stable build (#3), 41 sec ago
- Last successful build (#3), 41 sec ago
- Last failed build (#2), 2 min 12 sec ago
- Last unsuccessful build (#2), 2 min 12 sec ago
- Last completed build (#3), 41 sec ago

Builds

Filter

Today

#4 6:32 AM

-Changes reflects

The screenshot shows a web browser window with multiple tabs open. The active tab displays a registration form titled "This is my milestone project". The form asks for "Enter Name", "Enter mobile", "Enter Email", "Password", and "Repeat Password". Below the form, a note states: "By creating an account you agree to our [Terms & Privacy](#)". A "Register" button is present, along with a link to "Sign in".

**Thank You, Happy Learning**

**See You Again**

-To automate the process, Choose this option in the jenkins. Now we don't need to build the jenkins after commit, It reflects immediately.

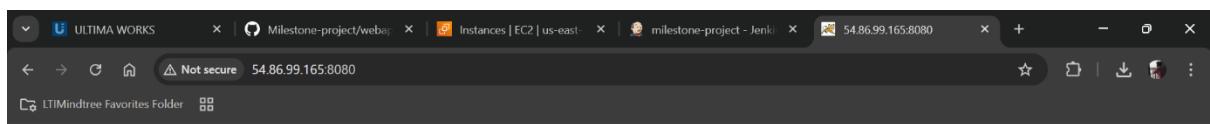
The screenshot shows the Jenkins configuration page for a job named "milestone-project". The left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The "Triggers" section is currently selected and expanded, showing various trigger options like "Build whenever a SNAPSHOT dependency is built", "Trigger builds remotely", "Build after other projects are built", "Build periodically", "GitHub Branches", "GitHub Pull Requests", and "GitHub hook trigger for GITScm polling". The "GitHub hook trigger for GITScm polling" option is checked. The "Environment" section at the bottom allows configuring workspace credentials and paths. At the bottom of the page are "Save" and "Apply" buttons.

The screenshot shows a GitHub code editor interface. The left sidebar displays a project structure under 'Arunvel83 / Milestone-project'. The 'Files' section shows files like 'main', 'server', 'webapp', 'src/main/webapp', 'WEB-INF', and 'index.jsp'. The 'index.jsp' file is selected and open in the main editor area. The code content is as follows:

```
1 <form action="action_page.php">
2 <div class="container">
3 <h1>This is my milestone project devops</h1>
4 <p>>Please fill in this form to create an account.</p>
5 <hr>
6
7 <label for="Name"><b>Enter Name</b></label>
8 <input type="text" placeholder="Enter Full Name" name="Name" id="Name" required>
9 <br>
10
11 <label for="mobile"><b>Enter mobile</b></label>
12 <input type="text" placeholder="Enter mobile number" name="mobile" id="mobile" required>
13 <br>
14
15 <label for="email"><b>Enter Email</b></label>
```

Below the code editor, a note says: "Use **Control + Shift + m** to toggle the **tab** key moving focus. Alternatively, use **esc** then **tab** to move to the next interactive element on the page."

-It reflects.



By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

Already have an account? [Sign in](#).

**Thank You, Happy Learning**

**See You Again**



## -Create new instance for docker

The screenshot shows the AWS EC2 'Launch an Instance' page. In the 'Name and tags' section, the name 'docker-server' is entered. Under 'Application and OS Images (Amazon Machine Image)', the 'Quick Start' tab is selected, showing various AMI options like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. On the right side, the 'Summary' section shows 1 instance being launched. The 'Software Image (AMI)' is set to Amazon Linux 2023.7.2. The 'Virtual server type (instance type)' is t2.micro. Other settings include a New security group and 1 volume(s) - 8 GiB. A large orange 'Launch instance' button is at the bottom right.

## -Connect to the terminal

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-18-209-176-240.compute-1.amazonaws.com
The authenticity of host 'ec2-18-209-176-240.compute-1.amazonaws.com (18.209.176.240)' can't be established.
ED25519 key fingerprint is SHA256:0vBmQQ2L0u6IIT5kuCE0PQs55u0/Q69+u9546GHgbXg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-209-176-240.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

               _#
  _\_\_ #####      Amazon Linux 2023
  ~~ \#####\
  ~~  \###|
  ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
  ~~    V~' '-->
  ~~     /
  ~~   /-
  ~~  /-
  ~~ /m'

[ec2-user@ip-172-31-36-46 ~]$ sudo su -
[root@ip-172-31-36-46 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-36-46 ~]# bash
[root@docker ~]#
```

-Install docker

```
root@arunvel-project~ X | root@jenkins~ X | root@apache-server~ X | root@ip-172-31-36-46~ X + - 14:30

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-18-209-176-240.compute-1.amazonaws.com
The authenticity of host 'ec2-18-209-176-240.compute-1.amazonaws.com (18.209.176.240)' can't be established.
ED25519 key fingerprint is SHA256:0vBmQ02LoU6IIT5kuCE0PQss55u0/Q69+u9546GHgbXg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-209-176-240.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ ##### Amazon Linux 2023
~~ \#####\
~~ \|##|
~~ ^#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ |V~! _-->
~~~~
~~ .-. / \
~~ /_ / \
~~ /m/ , [ec2-user@ip-172-31-36-46 ~]$ sudo su -
[root@ip-172-31-36-46 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-36-46 ~]# bash
[root@docker ~]# yum install docker*
```

-Configure the aws by creating user and use the access-key and secret-key to configure aws

```
root@arunvel-project-~:~|root@jenkins:~:~|root@apache-server:~/apache:~|root@docker:~:~|  
Experimental: false  
Insecure Registries:  
127.0.0.0/8  
Live Restore Enabled: false  
  
[root@docker ~]# yum install awscli -y  
Last metadata expiration check: 8:53:59 ago on Fri Jun 27 09:00:32 2025.  
Package awscli-2.25.0-1.amzn2023.0.1.noarch is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@docker ~]# aws configure  
AWS Access Key ID [*****X067]: AKIA2GDH70IHABOLXQ67  
AWS Secret Access Key [*****AjzZ]: wcvQT+m/bRJlCkpw3PSGIwCahMFYY3jh4W8AJzZ  
Default region name [us-east-1]: us-east-1  
Default output format [table]: table  
[root@docker ~]# vim .ssh/authorized_keys  
[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 fq_codel state UP group default qlen 1000  
    link/ether 0e:ba:c5:0a:30:73 brd ff:ff:ff:ff:ff:ff  
    altname eni-037a6ce656a59296  
    altname device-number-0.0  
    inet 172.31.36.46/20 metric 512 brd 172.31.47.255 scope global dynamic enX0  
        valid_lft 3535sec preferred_lft 3535sec  
    inet6 fe80::c8a:c5ff:fe0a:3073/64 scope link proto kernel_ll  
        valid_lft forever preferred_lft forever  
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
```

## -SSH key generation

```
root@arunvel-project~ root@jenkins~ root@apache-server:~/apache root@docker:~ + - _ X
valid_lft forever preferred_lft forever
[root@docker ~]# vim .ssh/id_rsa.pub
[root@docker ~]# 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:
SHA256:6RdAgrstPCTgrb8Kri4kIt3GPZ0d7zhyaecGN0X8ahBM root@docker.example.com
The key's randomart image is:
+---[RSA 3072]---+
|   . .
|   . .
|   . .
|   . o
| ... o S . o
|... = ... .+ +
|.o+.o+...*+.E
|.o*o.+ o0++o+.
|.oo=o .++o=o+-
+---[SHA256]---+
[root@docker ~]# vim .ssh/id_rsa.pub
1L, 577B written
[root@docker ~]# ssh root@172.31.33.190
#_
`__ _###_ Amazon Linux 2023
~~ \_####\_
~~ `###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ ^~' '->
~~ /

```

-Need to connect the docker and jenkins server. Generate the SSH and copy paste the key in authorized\_keys in both machines

-Restart sshd in both the machines

```
root@arunvel-project-~ X root@jenkins:~ X root@apache-server:~/apache X | root@docker:~ X + - _ X
/_/ _/
_/_m/_/_
Last login: Fri Jun 27 17:53:08 2025
[root@docker ~]# exit
logout
Connection to 172.31.36.46 closed.
[root@jenkins ~]# vim .ssh/authorized_keys
[root@jenkins ~]#
[root@jenkins ~]# 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 0e:fc:e2:77:e0:a3 brd ff:ff:ff:ff:ff:ff
    altname eni-07ac9629ce38759c1
    altname device-number-0.0
    inet 172.31.33.190/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
        valid_lft 1837sec preferred_lft 1837sec
    inet6 fe80::cfc:e2ff:fe77:e0a3/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
[root@jenkins ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@jenkins ~]# vim /etc/ssh/sshd_config
[root@jenkins ~]# systemctl restart sshd
[root@jenkins ~]# systemctl enable sshd
[root@jenkins ~]# |
```

```
The key's randomart image is:
+---[RSA 3072]---+
|   ...
|   . o
|   .
|   . o
|   .. o S . o
| ... = ... + +
| .o+.o+...*+E
| .o*o.+ oO++o.
|.oo=o .+++o+
+---[SHA256]---+
[root@docker ~]# vim .ssh/id_rsa.pub
1L, 577B written
[root@docker ~]# ssh root@172.31.33.190
' _#
`~\###'
~~\_###\
~~\###'
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~' `->
~~ \_/
~~ \_/
~~ \_/
~~ \_/
Last login: Fri Jun 27 17:51:55 2025
[root@jenkins ~]# exit
logout
Connection to 172.31.33.190 closed.
[root@docker ~]# vim /etc/ssh/sshd_config
[root@docker ~]# systemctl restart sshd
[root@docker ~]# systemctl enable sshd
[root@docker ~]# |
```

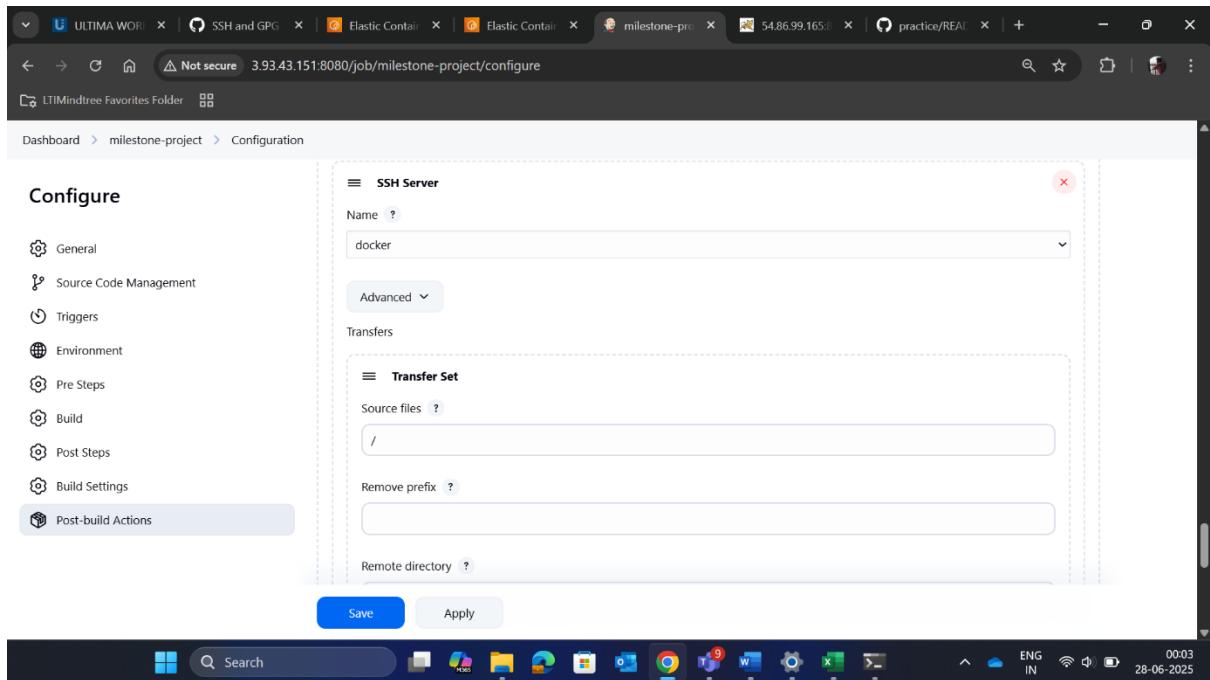
-In the jenkins > milestone-project > configuration > post build actions > under SSH server -add the docker and jenkins

The screenshot shows the Jenkins configuration page for the 'milestone-project'. The 'Post-build Actions' section is selected. Under 'SSH Publishers', there is a 'SSH Server' configuration with 'Name' set to 'jenkins'. A 'Transfer Set' is defined with 'Source files' set to '/'. Below the configuration are 'Save' and 'Apply' buttons.

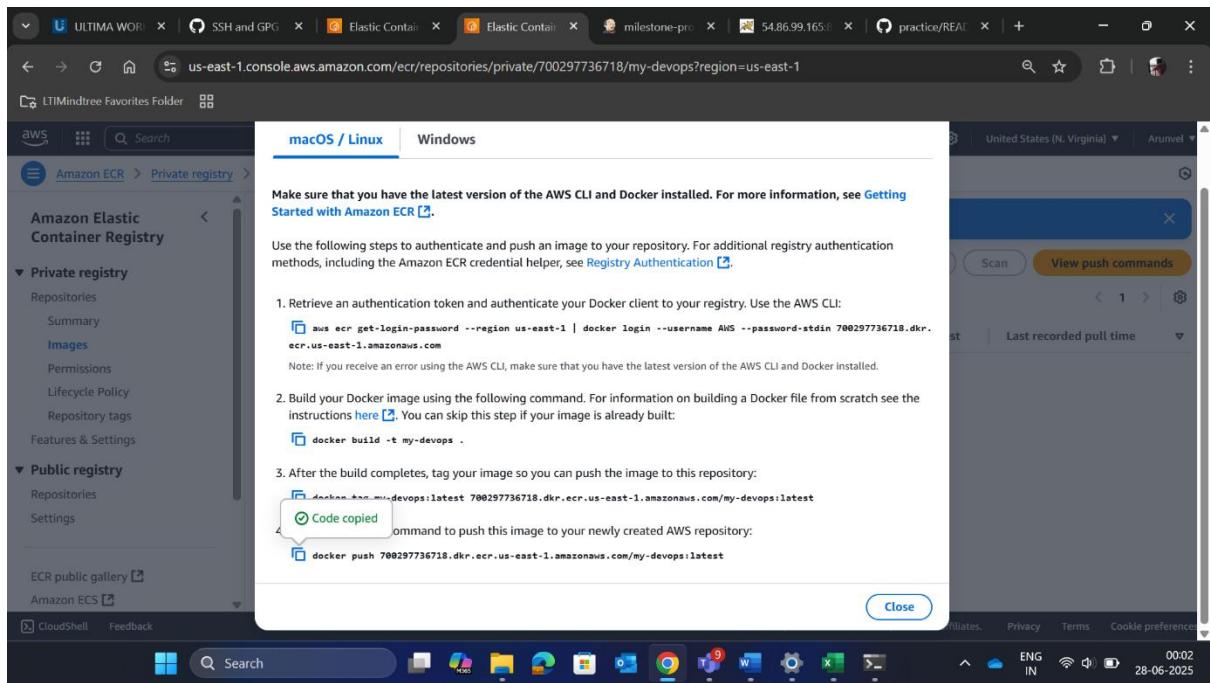
-Add the exec command

The screenshot shows the Jenkins configuration page for the 'milestone-project'. The 'Post-build Actions' section is selected. An 'Exec command' is added with the value 'rsync -avh /var/lib/jenkins/workspace/milestone-project/\* root@172.31.36.46:/opt'. Below the configuration are 'Save' and 'Apply' buttons.

-Same for the docker



-Go to ECR in aws > create a private registry – and copy paste these command to push



-Paste it in the Exec command of docker

The screenshot shows the Jenkins configuration interface for a job named 'milestone-project'. The 'Post-build Actions' section is selected. Under 'Exec command', the following script is pasted:

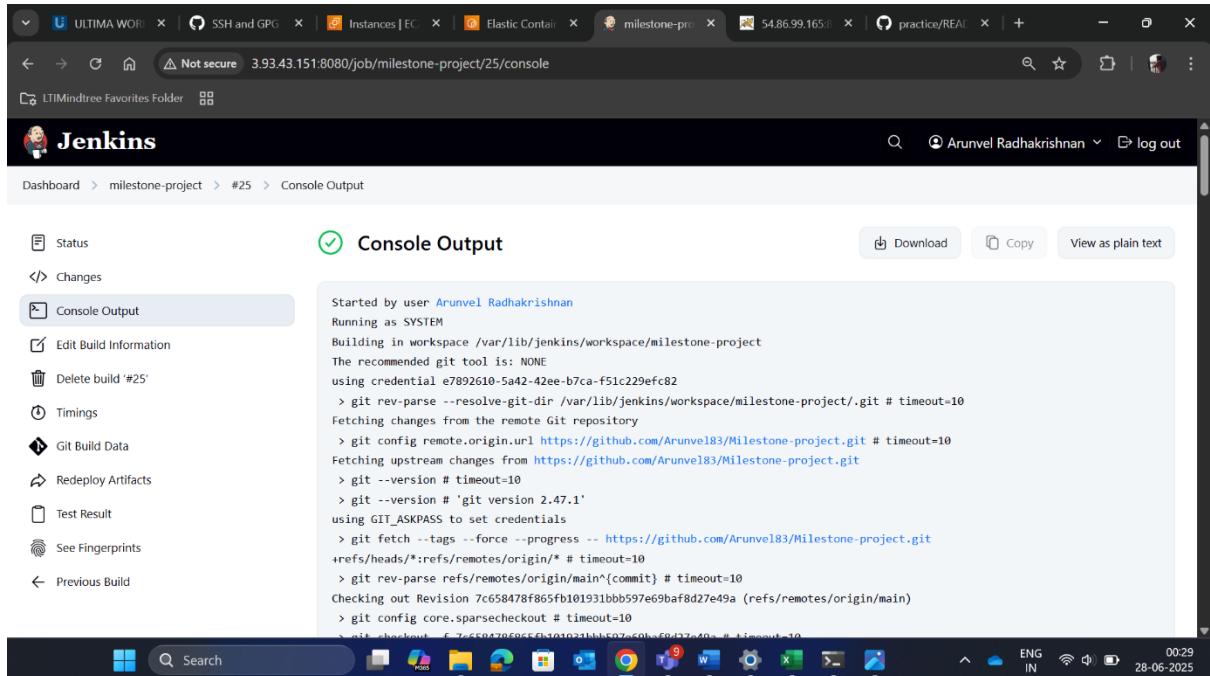
```
cd /opt
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 700297736718.dkr.ecr.us-east-1.amazonaws.com
docker build -t my-devops .
docker tag my-devops:latest 700297736718.dkr.ecr.us-east-1.amazonaws.com/my-devops:latest
```

Below the exec command, there is a note: "All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables". There are 'Advanced' and 'Add Transfer Set' buttons. At the bottom are 'Save' and 'Apply' buttons.

-Now build is successful

The screenshot shows the Jenkins project page for 'milestone-project'. The status is green with a checkmark icon, indicating a successful build. The 'Test Result Trend' chart shows a solid green area from build #1 to #25, with a legend for 'Passed' (green), 'Skipped' (grey), and 'Failed' (red). The 'Build Now' button is visible in the sidebar. The 'Builds' section shows the most recent build (#25) was run today at 6:30 PM.

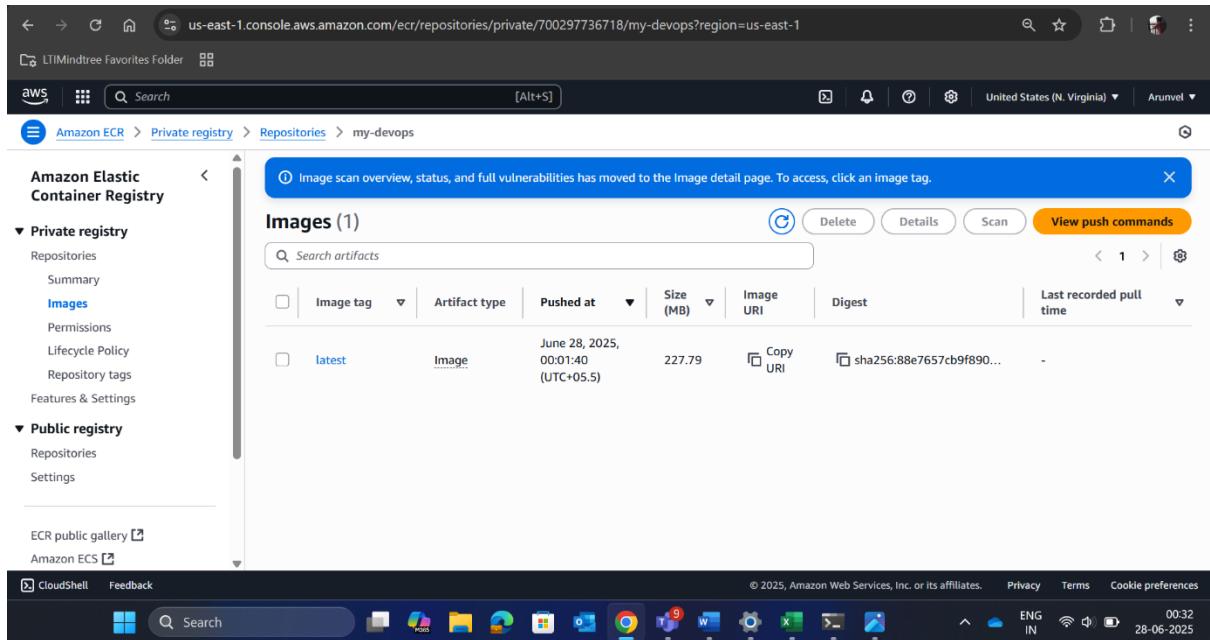
## -Console output



The screenshot shows the Jenkins interface with the 'Console Output' tab selected. The logs display the build process, starting with the user 'Arunvel Radhakrishnan' running as SYSTEM and building in workspace /var/lib/jenkins/workspace/milestone-project. The logs then show the execution of various git commands to clone the repository from GitHub, including fetching upstream changes and setting credentials using GIT\_ASKPASS.

```
Started by user Arunvel Radhakrishnan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/milestone-project
The recommended git tool is: NONE
using credential e7892610-5a42-42ee-b7ca-f51c229efc82
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/milestone-project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Arunvel83/Milestone-project.git # timeout=10
Fetching upstream changes from https://github.com/Arunvel83/Milestone-project.git
> git --version # timeout=10
> git --version # 'git version 2.47.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/Arunvel83/Milestone-project.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7c658478f865fb101931bbb597e69ba8d27e49a (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7c658478f865fb101931bbb597e69ba8d27e49a # timeout=10
```

## -Image is also created



The screenshot shows the Amazon ECR console with the 'Images' section selected. It displays a single image artifact named 'latest' which was pushed on June 28, 2025, at 00:01:40 UTC. The image size is 227.79 MB and its digest is sha256:88e7657cb9f890... . A note indicates that the image scan overview has moved to the image detail page.

| Image tag | Artifact type | Pushed at                          | Size (MB) | Image URI                | Digest                                   | Last recorded pull time |
|-----------|---------------|------------------------------------|-----------|--------------------------|--|-------------------------|
| latest    | Image         | June 28, 2025, 00:01:40 (UTC+05.5) | 227.79    | <a href="#">Copy URI</a> | <a href="#">sha256:88e7657cb9f890...</a> | -                       |

## -Create a new instance for kubernetes-cluster

The screenshot shows the AWS EC2 'Launch an instance' page. In the 'Name and tags' section, the name 'eks-server' is entered. Under 'Application and OS Images (Amazon Machine Image)', the search bar is empty. In the 'Summary' section, the number of instances is set to 1, the software image is 'Amazon Linux 2023 AMI 2023.7.2...', the instance type is 't2.micro', and the storage is '1 volume(s) - 8 GiB'. The 'Launch instance' button is highlighted.

## -Connect it with the terminal

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835661> cd .\Downloads\
PS C:\Users\10835661\Downloads> ssh -i "lti-key.pem" ec2-user@ec2-3-81-160-108.compute-1.amazonaws.com
The authenticity of host 'ec2-3-81-160-108.compute-1.amazonaws.com (3.81.160.108)' can't be established.
ED25519 key fingerprint is SHA256:iEDQeNwM6RuClJC+DVK/eQDl6riy97suh32/tcJFKVY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-81-160-108.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

               _#
  _\_\_ #####_   Amazon Linux 2023
~~~ \#####\_
~~~ \###\_
~~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~~ V~' '->
~~~ /_
~~~ .-. /_
~~~ /_/
~~~ /m/'

[ec2-user@ip-172-31-35-241 ~]$ sudo su -
[root@ip-172-31-35-241 ~]# hostnamectl set-hostname eks-server.example.com
[root@ip-172-31-35-241 ~]# bash
[root@eks-server ~]# 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:
```

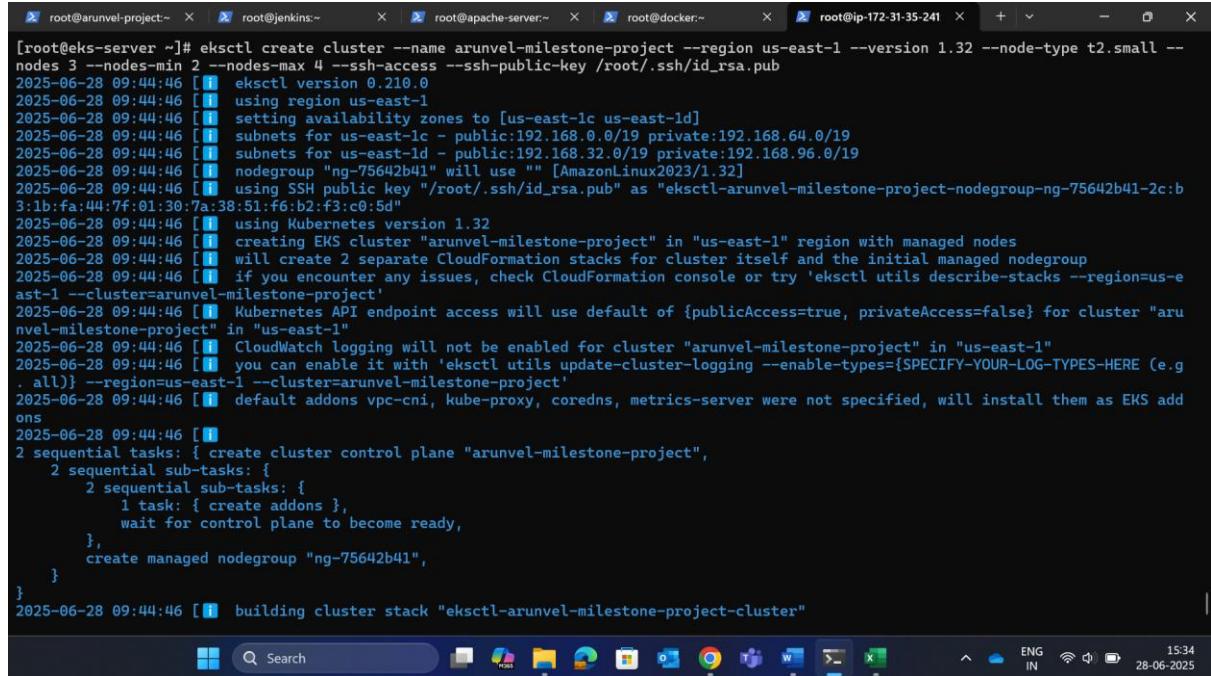
-SSH key generation, connect it with jenkins

```
|     =.=. +oo|  
|     . S + o. .|  
|     o = + .|  
|     o o . .|  
|     . + .o .|  
|     .=o+. .|  
+----[SHA256]----+  
[root@eks-server ~]# cat /etc/os-release  
NAME="Amazon Linux"  
VERSION="2023"  
ID="amzn"  
ID_LIKE="fedora"  
VERSION_ID="2023"  
PLATFORM_ID="platform:al2023"  
PRETTY_NAME="Amazon Linux 2023.7.20250623"  
ANSI_COLOR="#3399FF"  
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"  
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"  
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"  
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"  
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"  
VENDOR_NAME="AWS"  
VENDOR_URL="https://aws.amazon.com/"  
SUPPORT_END="2029-06-30"  
[root@eks-server ~]# vim /etc/ssh/sshd_config  
[root@eks-server ~]# systemctl restart sshd  
[root@eks-server ~]# systemctl enable sshd  
[root@eks-server ~]# yum update -y  
Amazon Linux 2023 Kernel Livepatch repository  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@eks-server ~]# yum install unzip -y
```

-Install awscli and unzip it

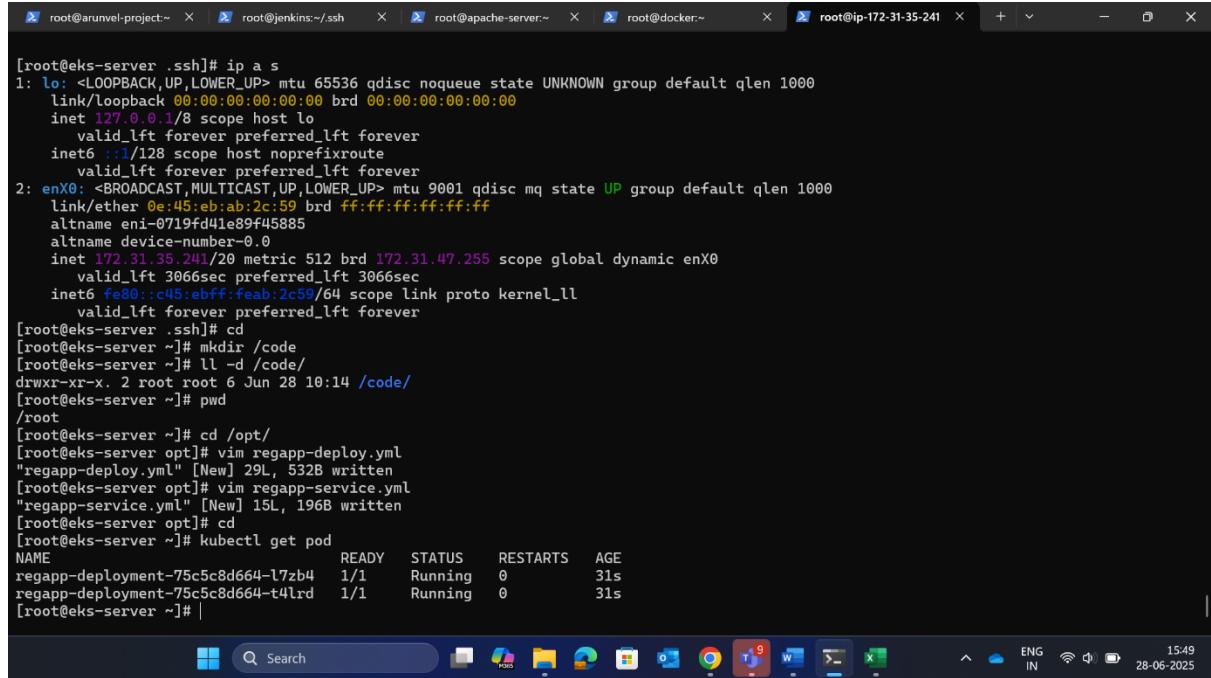
```
Dependencies resolved.
Nothing to do.
Complete!
[root@eks-server ~]# yum install unzip -y
Last metadata expiration check: 0:00:29 ago on Sat Jun 28 09:35:23 2025.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@eks-server ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received   % Xferd  Average Speed   Time     Time   Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 63.1M 100 63.1M    0      0  128M     0  --:--:--  --:--:--  --:--:-- 128M
[root@eks-server ~]# unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
inflating: aws/install
inflating: aws/README.md
inflating: aws/THIRD_PARTY_LICENSES
  creating: aws/dist/awscli/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
inflating: aws/dist/aws
inflating: aws/dist/aws_completer
inflating: aws/dist/libpython3.13.so.1.0
inflating: aws/dist/_awscrt.abi3.so
inflating: aws/dist/_ruamel_yaml.cpython-313-x86_64-linux-gnu.so
inflating: aws/dist/libbz.so.1
inflating: aws/dist/liblzma.so.5
inflating: aws/dist/libbz2.so.1
inflating: aws/dist/libffi.so.6
inflating: aws/dist/libuuid.so.1
```

### -Create kubernetes cluster



```
[root@eks-server ~]# eksctl create cluster --name arunvel-milestone-project --region us-east-1 --version 1.32 --node-type t2.small --nodes 3 --nodes-min 2 --nodes-max 4 --ssh-access --ssh-public-key /root/.ssh/id_rsa.pub
2025-06-28 09:44:46 [ℹ] eksctl version 0.210.0
2025-06-28 09:44:46 [ℹ] using region us-east-1
2025-06-28 09:44:46 [ℹ] setting availability zones to [us-east-1c us-east-1d]
2025-06-28 09:44:46 [ℹ] subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2025-06-28 09:44:46 [ℹ] subnets for us-east-1d - public:192.168.32.0/19 private:192.168.96.0/19
2025-06-28 09:44:46 [ℹ] nodegroup "ng-75642b41" will use "" [AmazonLinux2023/1.32]
2025-06-28 09:44:46 [ℹ] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-arunvel-milestone-project-nodegroup-ng-75642b41-2c:b3:1b:fa:44:7f:01:30:7a:38:51:f6:d2:f3:c0:5d"
2025-06-28 09:44:46 [ℹ] using Kubernetes version 1.32
2025-06-28 09:44:46 [ℹ] creating EKS cluster "arunvel-milestone-project" in "us-east-1" region with managed nodes
2025-06-28 09:44:46 [ℹ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-06-28 09:44:46 [ℹ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=arunvel-milestone-project'
2025-06-28 09:44:46 [ℹ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "arunvel-milestone-project" in "us-east-1"
2025-06-28 09:44:46 [ℹ] CloudWatch logging will not be enabled for cluster "arunvel-milestone-project" in "us-east-1"
2025-06-28 09:44:46 [ℹ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=arunvel-milestone-project'
2025-06-28 09:44:46 [ℹ] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-06-28 09:44:46 [ℹ]
2 sequential tasks: { create cluster control plane "arunvel-milestone-project",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-75642b41",
  }
}
2025-06-28 09:44:46 [ℹ] building cluster stack "eksctl-arunvel-milestone-project-cluster"
```

### -Need to configue the deployment and service files



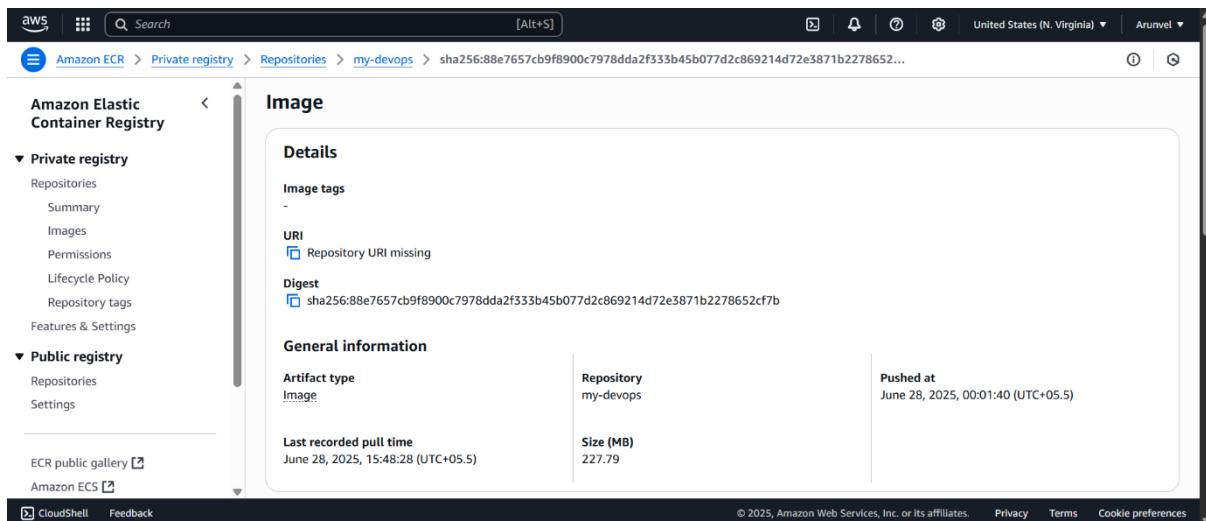
```
[root@eks-server .ssh]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
      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 0e:45:eb:ab:2c:59 brd ff:ff:ff:ff:ff:ff
    altname eni-0719fd41e89f45885
    altname device-number-0.0
    inet 172.31.35.241/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
      valid_lft 3066sec preferred_lft 3066sec
    inet6 fe80::c45:ebff:feab:2c59/64 scope link proto kernel ll
      valid_lft forever preferred_lft forever
[root@eks-server .ssh]# cd
[root@eks-server ~]# mkdir /code
[root@eks-server ~]# ll -d /code/
drwxr-xr-x. 2 root root 6 Jun 28 10:14 /code/
[root@eks-server ~]# pwd
/root
[root@eks-server ~]# cd /opt/
[root@eks-server opt]# vim regapp-deploy.yml
"regapp-deploy.yml" [New] 29L, 532B written
[root@eks-server opt]# vim regapp-service.yml
"regapp-service.yml" [New] 15L, 196B written
[root@eks-server opt]# cd
[root@eks-server ~]# kubectl get pod
NAME                      READY   STATUS    RESTARTS   AGE
regapp-deployment-75c5c8d664-l7zb4  1/1     Running   0          31s
regapp-deployment-75c5c8d664-t4lrd  1/1     Running   0          31s
[root@eks-server ~]# |
```

-Copy the details from github and paste in the terminal

The screenshot shows a GitHub repository named 'Milestone-project'. In the 'Code' tab, the 'regapp-deploy.yml' file is selected. The code content is as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: regapp-deployment
  labels:
    app: regapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp
  template:
    metadata:
      labels:
```

-Change the image id with our image id

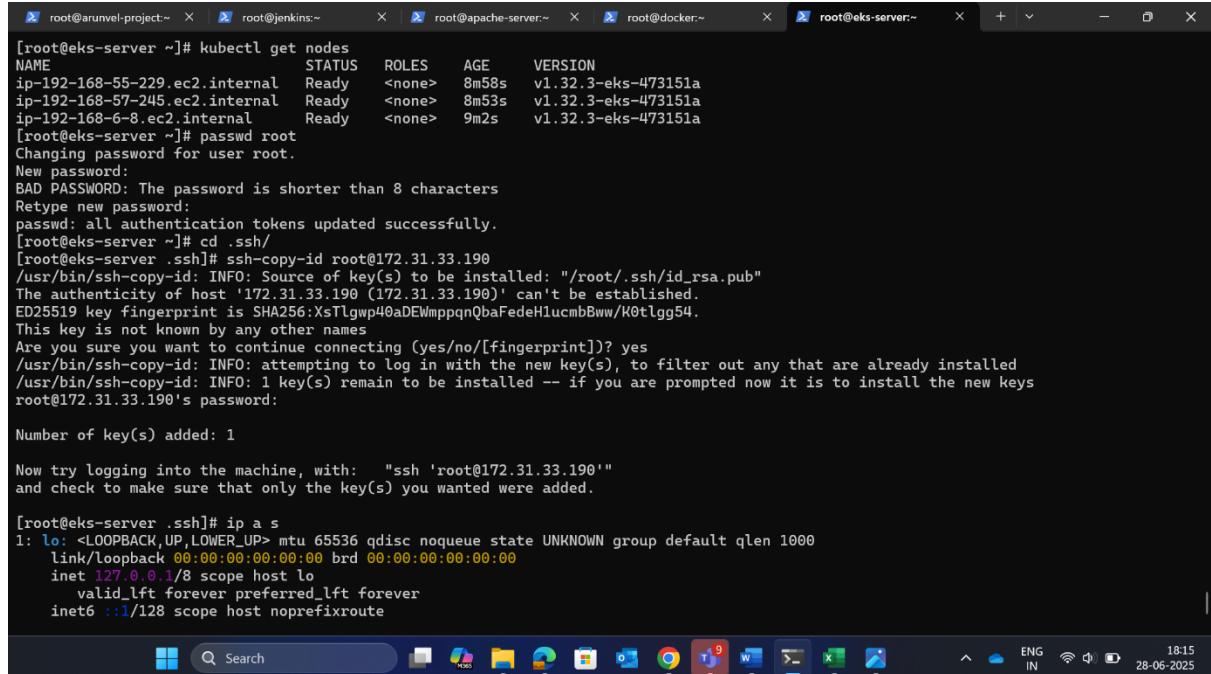


-Same for the service files

The screenshot shows a GitHub repository named 'Milestone-project'. In the 'Code' tab, the 'regapp-service.yml' file is selected. The code content is as follows:

```
apiVersion: v1
kind: Service
metadata:
  name: regapp-service
  labels:
    app: regapp
spec:
  selector:
    app: regapp
  ports:
    - port: 8080
      targetPort: 8080
  type: LoadBalancer
```

-Copy the SSH-id in both jenkins and kubernetes

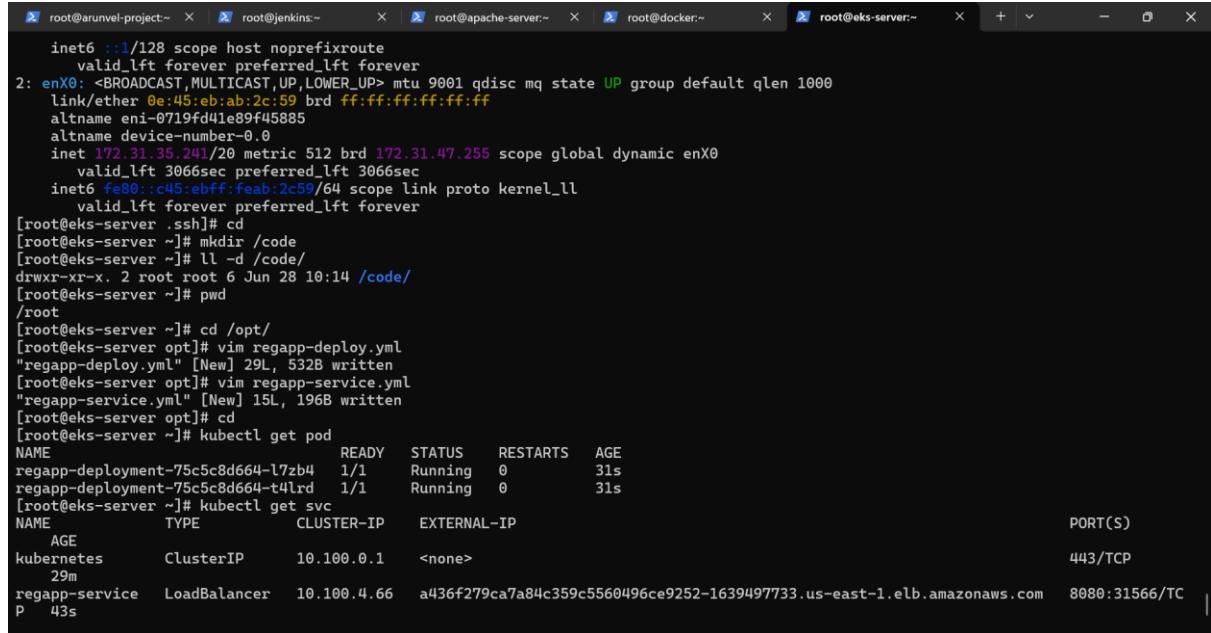


```
[root@eks-server ~]# kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-55-229.ec2.internal  Ready   <none>  8m58s  v1.32.3-eks-473151a
ip-192-168-57-245.ec2.internal  Ready   <none>  8m53s  v1.32.3-eks-473151a
ip-192-168-6-8.ec2.internal   Ready   <none>  9m2s   v1.32.3-eks-473151a
[root@eks-server ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@eks-server ~]# cd .ssh/
[root@eks-server .ssh]# ssh-copy-id root@172.31.33.190
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.33.190 (172.31.33.190)' can't be established.
ED25519 key fingerprint is SHA256:XsTlgwp40aDEWmpqQbaFedeH1ucmbBww/K0tlgg54.
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@172.31.33.190's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.33.190'"
and check to make sure that only the key(s) you wanted were added.

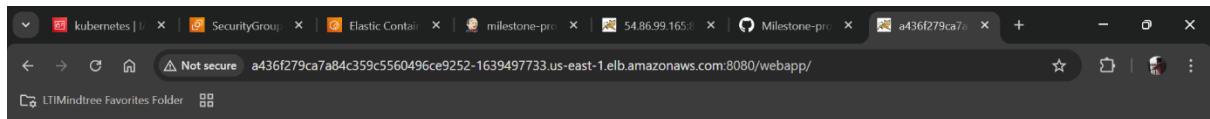
[root@eks-server .ssh]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
```

-get the link and paste it in the URL



```
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 0e:45:eb:ab:2c:59 brd ff:ff:ff:ff:ff:ff
        altname eni-0719fd41e89f45885
        altname device-number-0.0
        inet 172.31.35.241/20 metric 512 brd 172.31.47.255 scope global dynamic enX0
            valid_lft 3066sec preferred_lft 3066sec
            inet6 fe80::45:ebff:feab:2c59/64 scope link proto kernel_ll
                valid_lft forever preferred_lft forever
[root@eks-server ~]# cd
[root@eks-server ~]# mkdir /code
[root@eks-server ~]# ll -d /code/
drwxr-xr-x. 2 root root 6 Jun 28 10:14 /code/
[root@eks-server ~]# pwd
/root
[root@eks-server ~]# cd /opt/
[root@eks-server opt]# vim regapp-deploy.yml
"regapp-deploy.yml" [New] 29L, 532B written
[root@eks-server opt]# vim regapp-service.yml
"regapp-service.yml" [New] 15L, 196B written
[root@eks-server opt]# cd
[root@eks-server ~]# kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
regapp-deployment-75c5c8d664-l7zb4  1/1    Running   0          31s
regapp-deployment-75c5c8d664-t4lrd  1/1    Running   0          31s
[root@eks-server ~]# kubectl get svc
NAME              TYPE      CLUSTER-IP   EXTERNAL-IP                                     PORT(S)
AGE
kubernetes        ClusterIP  10.100.0.1   <none>                                         443/TCP
29m
regapp-service   LoadBalancer 10.100.4.66  a436f279ca7a84c359c5560496ce9252-1639497733.us-east-1.elb.amazonaws.com  8080:31566/TC
P 43s
```

-The URL works perfectly!



## This is my milestone project devops successful

Please fill in this form to create an account.

|                 |                       |
|-----------------|-----------------------|
| Enter Name      | [Enter Full Name]     |
| Enter mobile    | [Enter mobile number] |
| Enter Email     | [Enter Email]         |
| Password        | [Enter Password]      |
| Repeat Password | [Repeat Password]     |

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

Already have an account? [Sign in](#).

**Thank You, Happy Learning**

**See You Again**



-The build is successful!

