

MILESTONE-2 PROJECT-Manaswini-P.S No-10835497

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

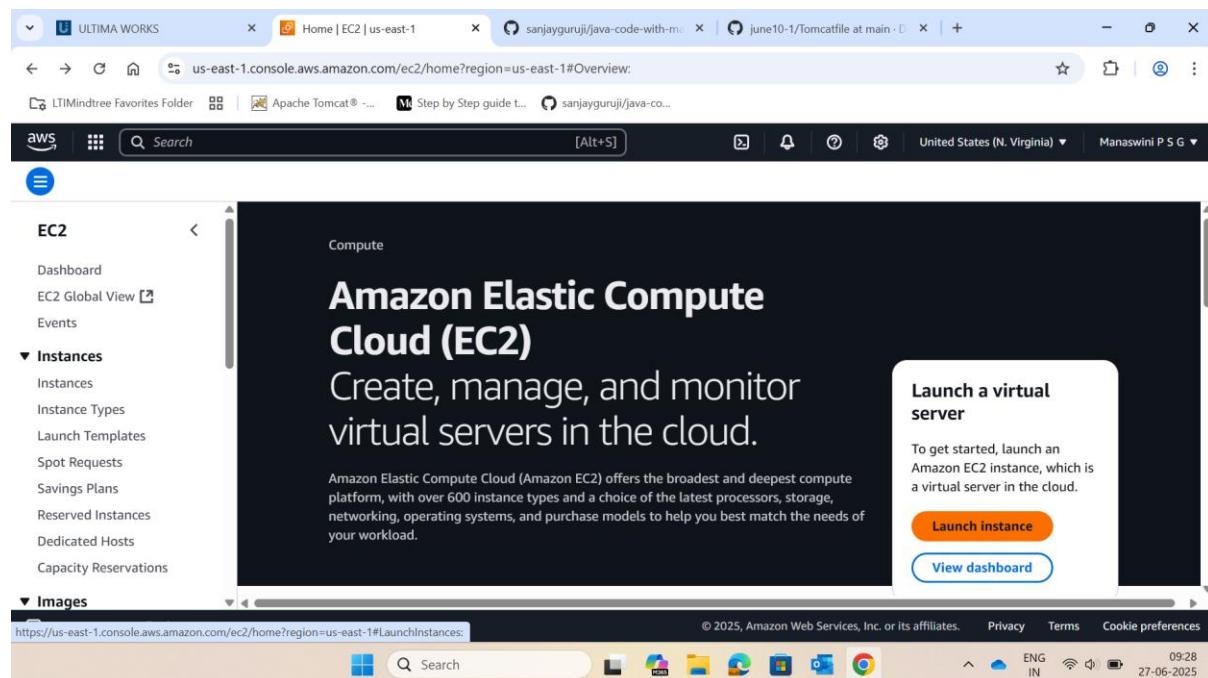
Phase 3: Push Docker Image to AWS ECR & Deploy to Kubernetes

- 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.

Procedure:



Open AWS and go to EC2. Now launch an instance.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' section, the name 'dev-server' is entered. The 'Software Image (AMI)' is set to 'Amazon Linux 2023 AMI 2023.7.2...'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Firewall (security group)' is 'New security group'. The summary on the right shows 1 instance.

Configure -

Name: dev-server

Instance type: t2.micro

Storage: 12 GiB

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the storage section, a root volume of 12 GiB is selected with an gp3 type. A note says 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. The summary on the right shows 1 instance.

Now launch the instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with 'EC2' selected. The main area displays a table of instances with one row selected: 'dev-server' (Instance ID: i-0219ccc4c0545e873, State: Running, Type: t2.micro). Below the table, the instance details for 'i-0219ccc4c0545e873 (dev-server)' are shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is active.

Now select the instance and click on connect.

The screenshot shows the 'Connect' page for the selected instance. It has tabs for EC2 Instance Connect, Session Manager, SSH client (which is selected), and EC2 serial console. The 'SSH client' section shows the instance ID 'i-0219ccc4c0545e873 (dev-server)' and provides instructions for connecting using an SSH client. It includes a command-line example: 'ssh -i "lti-mahape.pem" ec2-user@ec2-44-210-139-180.compute-1.amazonaws.com'. The bottom of the page includes standard AWS footer links.

Copy the example link from SSH Client, and paste it into the terminal.

The screenshot shows a Windows PowerShell window. The command 'ssh -i "lti-mahape.pem" ec2-user@ec2-44-210-139-180.compute-1.amazonaws.com' is pasted into the terminal. The output shows the command being run.

```
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\10835497> cd downloads
PS C:\Users\10835497\downloads> ssh -i "lti-mahape.pem" ec2-user@ec2-44-210-139-180.compute-1.amazonaws.com
```

Now paste the link and connect it into 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\10835497> cd downloads
PS C:\Users\10835497\downloads> ssh -i "lti-mahape.pem" ec2-user@ec2-44-210-139-180.compute-1.amazonaws.com
The authenticity of host 'ec2-44-210-139-180.compute-1.amazonaws.com (44.210.139.180)' can't be established.
ED25519 key fingerprint is SHA256:HYLxbskt6Nyxl0vid+C3um7Y3RRMyiqKULnKkpExvQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-210-139-180.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-89-215 ~]$ 

```

The instance is connected to the terminal.

```

[ec2-user@ip-172-31-89-215 ~]$ sudo su -
[root@ip-172-31-89-215 ~]# hostnamectl set-hostname dev-server.example.com
[root@ip-172-31-89-215 ~]# bash
[root@dev-server ~]# yum install git
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
=====
Transaction Summary
=====
Install 8 Packages
Total download size: 7.5 M
Installed size: 37 M

```

Now change it to root user and change the hostname. Install git to pull the repository.

Code

java-code-with-maven Public

main · 1 Branch · 0 Tags

Go to file Add file Code

About

No description, website, or topics provided.

Readme · Activity · 0 stars · 1 watching · 6 forks

Report repository

Releases

No releases published

Packages

This is the repository to pull.

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

After installing git, initialize it.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository](#)

Required fields are marked with an asterisk (*).

Owner *	Repository name *
 ManaswiniSeshagiri	/ <input type="text" value="proj"/>
<small>proj is available.</small>	

Great repository names are short and memorable. Need inspiration? How about [studious-disco](#) ?

Description (optional)

 Public
Anyone on the internet can see this repository. You choose who can commit.

 Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

Create a new repository.

```
[root@dev-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:
SHA256:68RHBILnFMGmwIplwfPxm3HvyRqAA+5tAIuR57Bkvd0 root@dev-server.example.com
The key's randomart image is:
+--[RSA 3072]----+
| ...o ++= |
| . * + * . |
|=*= + = . |
|=X..+ = .. |
| + .o..ES.. |
| .. o ..o o |
| . o =.. |
| . o o+ |
| . o. |
+---[SHA256]----+
[root@dev-server ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAAQDciReLUQtTpTnZrM2gx6KvwCGB28oMGLTBZ5K8FrkrRN+LN1rkBrdv5qEQNb3Qy48Aysk40hWpu2kAzt0
0uwdnagJMK6KIwA9RQWLWT2uTCPFT0d3gnyc6CqUrGbc0WD7WaJUw5PeqHfkf/oINHutjNMGuJ/D/ZNw0jEU5sE9v9yUCato7Lz8qWT8II1RAfmSjS7oAxU
Y0hEn2HEqj0mkWqn89U09/7dko7KXpRSkx+U79L7ybMzEFta7/RmdpC2eK3WkPsZGJ12E0jXArermr201UyWQKSH05AhD/xN1mo/C/+0fPrctsOLynY7qxOad
MgXeZdBEZW69PCZ9y+2B/khoXAJtc006CcxEVvaHuMYtLQS6BzLvI67BoXDy4Qz7YjQv+iDgG0kXib5b6ZyUSQbKmN2xs6in6s/W8Kx1jrXMUnGAwEVFAM
6nIPTNJmq4Nz8rA0VLkYA3DxrGK0wBcWlQ7/ZxsJWIOf3zGtuhtIENEuwy/GwCxUL4EZPU= root@dev-server.example.com
[root@dev-server ~]#
```

Do SSH key generation in terminal.

You need to add the ssh key in github.

The screenshot shows a GitHub repository page for 'ManaswiniSeshagiri / proj'. The sidebar on the right has a 'Settings' link under the 'Your projects' section. The main content area shows sections for GitHub Copilot setup and adding collaborators.

Go to github and click on Settings.

The screenshot shows the GitHub 'Settings' page for the user 'ManaswiniSeshagiri'. The left sidebar has a 'SSH and GPG keys' section selected. The main content area shows the 'SSH keys' section, which is currently empty. A 'New SSH key' button is visible.

Click on create new SSH key and paste it.

The screenshot shows the 'Add new SSH Key' form. The 'Title' field contains 'proj'. The 'Key type' dropdown is set to 'Authentication Key'. The 'Key' field contains a very long SSH key string. At the bottom is a green 'Add SSH key' button.

The screenshot shows the GitHub account settings page for a user named 'sanjayguruji'. The left sidebar has a 'SSH and GPG keys' section selected. The main area displays the 'SSH keys' section, which lists a single key named 'proj'. This key has a SHA256 fingerprint of 'SHA256:68RHB1lnFMGmwIplwfPxm3HvyRqjAA+5tAIuR57Bkv0d', was added on June 27, 2025, and has never been used with read/write permissions. A 'Delete' button is visible next to the key entry. Below this, there's a link to a guide on connecting to GitHub using SSH keys and troubleshooting common SSH problems.

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

proj
SHA256:68RHB1lnFMGmwIplwfPxm3HvyRqjAA+5tAIuR57Bkv0d
Added on Jun 27, 2025
Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

Successfully created.

```
[root@dev-server ~]# mkdir proj
[root@dev-server ~]# cd proj
[root@dev-server proj]# git clone https://github.com/sanjayguruji/java-code-with-maven.git
Cloning into 'java-code-with-maven'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 29 (from 2)
Receiving objects: 100% (31/31), 4.87 KiB | 2.44 MiB/s, done.
Resolving deltas: 100% (1/1), done.
[root@dev-server proj]#
```

Now make a directory and open it.

```
[root@dev-server proj]# git add .
warning: adding embedded git repository: java-code-with-maven
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint: git submodule add <url> java-code-with-maven
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint: git rm --cached java-code-with-maven
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config advice.addEmbeddedRepo false"
[root@dev-server proj]# git commit -m "first commit"
[master (root-commit) bc204d1] first commit
Committer: root <root@dev-server.example.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
[root@dev-server proj]# git commit -m "first commit"
[master (root-commit) bc204d1] first commit
Committer: root <root@dev-server.example.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 160000 java-code-with-maven
[root@dev-server proj]# git branch -M main
[root@dev-server proj]# git remote add origin git@github.com:ManaswiniSeshagiri/proj.git
```

```
[root@dev-server proj]# git push -u origin main
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Writing objects: 100% (2/2), 213 bytes | 213.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:ManaswiniSeshagiri/proj.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Successfully pulled the git repo to our repo.

The screenshot shows a GitHub repository page for 'ManaswiniSeshagiri/proj'. The top navigation bar includes links for LTMindtree favorites, Apache Tomcat, Step by Step guide, and Sanjay Gurujir's Java code. The main content area has a heading 'Quick setup — if you've done this kind of thing before' with a note about creating a new file or uploading an existing one. Below this, there are three sections: '...or create a new repository on the command line' with the following commands:

```
echo "# proj" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:ManaswiniSeshagiri/proj.git
git push -u origin main
```

The second section, '...or push an existing repository from the command line', contains these commands:

```
git remote add origin git@github.com:ManaswiniSeshagiri/proj.git
git branch -M main
git push -u origin main
```

A 'ProTip!' at the bottom suggests using the URL for adding GitHub as a remote.

Now reload the repository

The screenshot shows a GitHub repository named "proj". The repository is public and has 1 branch and 0 tags. The commit history shows one commit from last year. The repository contains files such as README, Dockerfile, and deployment scripts. The "About" section indicates no description, website, or topics provided. The "Releases" section shows no releases published, and the "Packages" section shows no packages published.

Now you can see the repository is successfully pulled.

The screenshot shows the AWS EC2 Instances launch wizard. The current step is "Launch an instance". A message box says: "It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices". There are buttons for "Do not show me this message again" and "Take a walkthrough". The configuration details are as follows:

- Name and tags**: Name is set to "jenkins-server".
- Software Image (AMI)**: Amazon Linux 2023 AMI 2023.7.2...read more
- Virtual server type (instance type)**: t2.micro
- Firewall (security group)**: New security group

Create a new instance.

Name: jenkins-server

Instance type: t2.medium

Storage: 8 GiB

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'Instances' selected. The main area displays two instances: 'jenkins-server' (running, t2.medium) and 'dev-server' (running, t2.micro). Below the instances, the details for 'jenkins-server' are shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

Now connect the instance to terminal.

```
PS C:\Users\10835497> cd downloads
PS C:\Users\10835497\downloads> ssh -i "lti-mahape.pem" ec2-user@ec2-52-23-168-92.compute-1.amazonaws.com
The authenticity of host 'ec2-52-23-168-92.compute-1.amazonaws.com (52.23.168.92)' can't be established.
ED25519 key fingerprint is SHA256:EhiAbimj8Z2cAk/PSLSghXCX+7AqBzgIcfnrD/M/Oog.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-23-168-92.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-172-31-80-90 ~]$ sudo su -
[root@ip-172-31-80-90 ~]# hostnamectl set-hostname jenkins-server.example.com
[root@ip-172-31-80-90 ~]# bash
[root@jenkins-server ~]# sudo yum update -y
```

Now the instances is successfully connected.

```
[root@ip-172-31-80-90 ~]# bash
[root@jenkins-server ~]# sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                               191 kB/s | 17 kB     00:00
No match for argument: -y
Error: No packages marked for upgrade.
[root@jenkins-server ~]# wget -O /etc/yum.repos.d/jenkins.repo \https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2025-06-27 04:53:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|: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

2025-06-27 04:53:07 (10.8 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@jenkins-server ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@jenkins-server ~]# sudo yum upgrade
Jenkins-stable                                         689 kB/s | 31 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@jenkins-server ~]# sudo yum install java-17-amazon-corretto -y
Last metadata expiration check: 0:00:11 ago on Fri Jun 27 04:53:36 2025.
Dependencies resolved.
=====
 Package           Architecture Version       Repository      Size
=====
 Installing:
```

```
[root@jenkins-server ~]# sudo yum install jenkins -y
Last metadata expiration check: 0:00:53 ago on Fri Jun 27 04:53:36 2025.
Dependencies resolved.
=====
 Package           Architecture      Version       Repository      Size
=====
 Installing:
 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          30 MB/s | 90 MB  00:02
 Total                                         30 MB/s | 90 MB  00:03
 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
  Installing : jenkins-2.504.3-1.1.noarch          1/1
  Running scriptlet: jenkins-2.504.3-1.1.noarch          1/1
 1/1
```

```
[root@jenkins-server ~]# systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins-server ~]# systemctl start jenkins
[root@jenkins-server ~]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-06-27 04:55:45 UTC; 5s ago
       Main PID: 26347 (java)
         Tasks: 52 (limit: 4656)
        Memory: 592.5M
          CPU: 17.755s
        CGroup: /system.slice/jenkins.service
                 └─26347 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenki
Jun 27 04:55:41 jenkins-server.example.com jenkins[26347]: ea891b48a10c4f49878215982d12cc21
Jun 27 04:55:41 jenkins-server.example.com jenkins[26347]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPass
Jun 27 04:55:41 jenkins-server.example.com jenkins[26347]: ****
Jun 27 04:55:41 jenkins-server.example.com jenkins[26347]: ****
Jun 27 04:55:45 jenkins-server.example.com jenkins[26347]: 2025-06-27 04:55:45.468+0000 [id=33]      INFO      jenkins
Jun 27 04:55:45 jenkins-server.example.com jenkins[26347]: 2025-06-27 04:55:45.479+0000 [id=23]      INFO      hudson
Jun 27 04:55:45 jenkins-server.example.com systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Jun 27 04:55:45 jenkins-server.example.com jenkins[26347]: 2025-06-27 04:55:45.648+0000 [id=49]      INFO      hudson
Jun 27 04:55:45 jenkins-server.example.com jenkins[26347]: 2025-06-27 04:55:45.646+0000 [id=49]      INFO      hudson
```

Installing Jenkins in terminal.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a navigation sidebar with 'EC2' selected. Under 'Instances', it lists two instances: 'jenkins-server' (running, t2.medium, 2/2 checks passed) and 'dev-server' (running, t2.micro, 2/2 checks passed). Below the instances, there's a section for 'i-069de71102ce90ba6 (jenkins-server)' showing its public IPv4 address (52.23.168.92), private IPv4 address (172.31.80.90), and carrier IP addresses (ephemeral). At the bottom, there are links for 'CloudShell' and 'Feedback'.

Now copy Public IPv4 and paste with IPv4 and :8080

The screenshot shows a web browser window with the URL `52.23.168.92:8080/login?from=%2F`. The title bar says "Not secure". The main content is titled "Getting Started" and "Unlock Jenkins". It instructs the user to copy the password from `/var/lib/jenkins/secrets/initialAdminPassword`. A text input field is provided for the "Administrator password". A blue "Continue" button is at the bottom right.

```
[root@jenkins-server ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
ea891b48a10c4f49878215982d12cc21
[root@jenkins-server ~]#
```

Copy the password from terminal and paste

The screenshot shows the "Customize Jenkins" step. It says "Plugins extend Jenkins with additional features to support many different needs." Two options are shown: "Install suggested plugins" (selected) and "Select plugins to install". Both options have descriptions below them. At the bottom left is the Jenkins version "Jenkins 2.504.3".

Do basic configuration and create it.

The screenshot shows the Jenkins dashboard at 52.23.168.92:8080. The top navigation bar includes links for LTIMindtree Favorites Folder, Apache Tomcat®, Step by Step guide to..., and the user's session (sanjayguruji/java-co...). The dashboard features a sidebar with links for New Item, Build History, Manage Jenkins, and My Views. The main content area includes sections for Welcome to Jenkins!, Start building your software project, and Set up a distributed build.

This screenshot is similar to the first one, but a context menu has been opened from the user icon in the top right. The 'Security' option is highlighted in the dropdown menu, which also lists Builds, My Views, Account, Appearance, Preferences, Experiments, and Credentials.

Now go to security and create API token.

The screenshot shows the Jenkins Security page at 52.23.168.92:8080/user/manaswinipsq/security. The left sidebar shows links for Status, Builds, My Views, Account, Appearance, Preferences, Security (which is selected), Experiments, and Credentials. The main content area is titled 'Security' and contains a 'Status' section and an 'API Token' section. In the 'API Token' section, a token named 'proj' has been generated with the value '11ed63c439b9a159f99f788290581d07b4'. A warning message says '⚠ Copy this token now, because it cannot be recovered in the future.' Below the token is a button to 'Add new Token'.

Copy the token and create webhook

The screenshot shows the GitHub repository settings for 'proj'. The 'General' tab is selected. On the left, there's a sidebar with options like Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Models), and Webhooks (which is currently selected). The main area shows the repository name 'proj' and a 'Rename' button. It also includes sections for 'Template repository' (unchecked) and 'Require contributors to sign off on web-based commits' (unchecked). Below that is the 'Default branch' section, which says 'main'.

Go to repository settings.

The screenshot shows the GitHub repository settings for 'proj'. The 'Webhooks' tab is selected. The sidebar on the left shows the same navigation as before. The main area is titled 'Webhooks' and contains a sub-section 'Add webhook'. It explains that webhooks allow external services to be notified of events. A link to the 'Webhooks Guide' is provided.

Create a webhook.

The screenshot shows the GitHub repository settings for 'proj'. The 'Webhooks' tab is selected. The sidebar on the left shows the same navigation as before. The main area is titled 'Webhooks / Add webhook'. It provides instructions on how payloads will be sent to the URL. The form fields include 'Payload URL *' (set to 'http://52.23.168.92:8080/github-webhook/'), 'Content type *' (set to 'application/json'), 'Secret' (containing the value '11ed63c439b9a159f99f788290581d07b4'), and 'SSL verification' (with a note about verifying certificates).

Paste the tomcat ipv4 with :8080 and add /github-webhook/

Dashboard > Manage Jenkins

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Java 17 end of life in Jenkins

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.

More Info

Ignore

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.

Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

52.23.168.92:8080/manage/pluginManager

Now go to Manage Jenkins and click plugins.

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Plugin	Status
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
Uninstructable API	Success

After installing, restart the Jenkins.

Dashboard > Manage Jenkins

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

Java 17 end of life in Jenkins

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.

More Info

Ignore

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.

Clouds

Appearance

52.23.168.92:8080/manage/configureTools

Go to tools.

```
[root@jenkins-server ~]# 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-server ~]# |
```

From the terminal copy the path and add it.

Dashboard > Manage Jenkins > Tools

Add Maven

Maven

Name: maven

MAVEN_HOME: /usr/share/maven

Install automatically ?

Add Maven

Save Apply

Add maven and jdk path.

Jenkins

Dashboard > All > New Item

New Item

Enter an item name: proj

Select an item type:

 Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

OK

Now create a new maven project.

The screenshot shows the Jenkins configuration interface for a project named 'proj'. The left sidebar lists various configuration tabs: General, Source Code Management (selected), Triggers, Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The main panel is titled 'Git' and shows 'Source Code Management' settings. It includes fields for 'Repository URL' (set to 'https://github.com/ManaswiniSeshagiri/proj.git') and 'Credentials' (set to '- none -'). There is also an 'Advanced' dropdown and a 'Save' button at the bottom.

Add the github repository link and change to main and create item.

The screenshot shows the Jenkins 'Builds' page. At the top, there is a search bar labeled 'Filter' and a 'Today' section. Below it, a green checkmark icon indicates a successful build, labeled '#3 6:12 AM'. The build log is displayed in a large text area:

```
[INFO] Total time: 12.135 s
[INFO] Finished at: 2025-06-27T06:12:57Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```

Now build the project and see for green tick.

```
[INFO] Total time: 12.135 s
[INFO] Finished at: 2025-06-27T06:12:57Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/proj/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```

REST API Jenkins 2.504.3

Now project successfully created and built.

The screenshot shows the AWS EC2 'Launch an instance' wizard. The top navigation bar includes the AWS logo, a search bar, and links for CloudShell, Feedback, and Help. The main content area has a breadcrumb trail: EC2 > Instances > Launch an instance. The first step, 'Name and tags', is completed with the name 'tomcat-server'. The second step, 'Application and OS Images (Amazon Machine Image)', is currently expanded, showing options for an AMI (Amazon Linux 2023 AMI 2023.7.2...), virtual server type (t2.micro), and firewall (New security group). A large orange 'Launch instance' button is visible at the bottom right of this step. A summary box on the right lists the instance details: 1 instance, Software Image (AMI) as Amazon Linux 2023 AMI 2023.7.2..., Virtual server type as t2.micro, and Firewall as New security group. A 'Preview code' link is also present in the summary.

Now create a new instance.

Name: dev-server

Instance type: t2.micro

Storage: 10 GiB

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays the following information:

Instances (1/3) Info

Last updated less than a minute ago

Actions ▾ **Launch instances** ▾

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/> tomcat-server	i-089dfbcfe066c12d1	Running Q Q	t2.micro	Initializing	View alarms +
<input type="checkbox"/> Unselect instance: tomcat-server	59de71102ce90ba6	Running Q Q	t2.medium	2/2 checks passed	View alarms +
<input type="checkbox"/> dev-server	i-0219ccc4c054e873	Running Q Q	t2.micro	2/2 checks passed	View alarms +

i-089dfbcfe066c12d1 (tomcat-server)

Details | **Status and alarms** | **Monitoring** | **Security** | **Networking** | **Storage** | **Tags**

Instance summary Info

Instance details Info

AMI ID: Monitoring Platform details

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Connect the instance to terminal.

```
PS C:\Users\10835497> cd downloads
PS C:\Users\10835497\downloads> ssh -i "lti-mahape.pem" ec2-user@ec2-44-212-76-137.compute-1.amazonaws.com
The authenticity of host 'ec2-44-212-76-137.compute-1.amazonaws.com (44.212.76.137)' can't be established.
ED25519 key fingerprint is SHA256:09oti6H2aSkPozlt8rW6wrz41P4TpHwNjOdy2orVSZo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-212-76-137.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

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

[ec2-user@ip-172-31-80-142 ~]$ sudo su -
[root@ip-172-31-80-142 ~]# hostnamectl set-hostname tomcat-server.example.com
[root@ip-172-31-80-142 ~]# bash
```

```
[root@tomcat-server ~]# yum install java-1.8*
Last metadata expiration check: 0:00:40 ago on Fri Jun 27 05:24:17 2025.
Dependencies resolved.
=====
 Package                               Architecture Version      Repository    Size
=====
Installing:
 java-1.8.0-amazon-corretto          x86_64        1:1.8.0_452.b09-2.amzn2023   amazonlinux   38 M
 java-1.8.0-amazon-corretto-devel    x86_64        1:1.8.0_452.b09-2.amzn2023   amazonlinux   63 M
Installing dependencies:
 adwaita-cursor-theme                noarch       47.0-1.amzn2023.0.1        amazonlinux   325 k
 adwaita-icon-theme                  noarch       47.0-1.amzn2023.0.1        amazonlinux   285 k
 adwaita-icon-theme-legacy           noarch       46.2-2.amzn2023            amazonlinux   2.2 M
 als-a-lib                           x86_64        1.2.7.2-1.amzn2023.0.2      amazonlinux   504 k
 at-spi2-atk                         x86_64        2.54.0-1.amzn2023.0.1      amazonlinux   90 k
=====

```

```
[root@tomcat-server ~]# cd /
[root@tomcat-server ~]# cd /opt
[root@tomcat-server opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.106/bin/apache-tomcat-9.0.106.tar.gz
--2025-06-27 05:27:08-- 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 0.1s

2025-06-27 05:27:08 (111 MB/s) - 'apache-tomcat-9.0.106.tar.gz' saved [13007554/13007554]

[root@tomcat-server opt]# tar -zvxf 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
apache-tomcat-9.0.106/conf/jaspic-providers.xsd
```

```
[root@tomcat-server bin]# chmod +x startup.sh
[root@tomcat-server bin]# chmod +x shutdown.sh
[root@tomcat-server bin]# ln -s /opt/apache-tomcat-9.0.106/bin/startup.sh /usr/local/bin/tomcatup
[root@tomcat-server bin]# ln -s /opt/apache-tomcat-9.0.106/bin/startup.sh /usr/local/bin/tomcatdown
[root@tomcat-server bin]# tomcatdown
Using CATALINA_BASE: /opt/apache-tomcat-9.0.106
Using CATALINA_HOME: /opt/apache-tomcat-9.0.106
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.106/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/apache-tomcat-9.0.106/bin/bootstrap.jar:/opt/apache-tomcat-9.0.106/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@tomcat-server bin]# cd apache-tomcat-9.0.106
bash: cd: apache-tomcat-9.0.106: No such file or directory
[root@tomcat-server bin]# cd ..
[root@tomcat-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
```

Install tomcat and configure it.

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Developer Quick Start

- Tomcat Setup
- First Web Application
- Realms & AAA
- JDBC DataSources
- Examples
- Servlet Specifications
- Tomcat Versions

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager

Documentation

- [Tomcat 9.0 Documentation](#)
- [Tomcat 9.0 Configuration](#)
- [Tomcat Wiki](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)
- Important announcements, releases, security

Now paste the ipv4 address with : 8080 and paste it.

The screenshot shows the Apache Tomcat Web Application Manager. At the top, there's a navigation bar with icons for back, forward, search, and other browser functions. Below it is a toolbar with links for 'Manager', 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. The main content area is titled 'Tomcat Web Application Manager' and contains a table of applications. The table has columns for Path, Version, Display Name, Running, Sessions, and Commands. The 'Welcome to Tomcat' application is running and has 0 sessions. The 'Tomcat Documentation' and 'Servlet and JSP Examples' applications are also running and have 0 sessions. Each row in the table includes a set of buttons for Start, Stop, Reload, Undeploy, and session expiration settings.

Go to manager app and give credentials . You'll get this page.

The screenshot shows the Jenkins Credentials page. At the top, there's a breadcrumb navigation: Dashboard > Manage Jenkins > Credentials. The main section is titled 'Credentials' and shows two categories: 'Stores scoped to User: Manaswini P S G' and 'Stores from parent'. Under 'Stores scoped to User', there's a table with columns for P, Store, and Domains. It shows one entry for 'User: Manaswini P S G' under '(global)'. Under 'Stores from parent', there's a similar table showing one entry for 'System' under '(global)'.

Go to credentials and click global.

The screenshot shows the Jenkins Global credentials (unrestricted) page. At the top, there's a navigation bar with icons for search, notifications, and log out, and the text 'Manaswini P S G'. The main content area is titled 'Global credentials (unrestricted)' and contains a table for managing credentials. The table has columns for ID, Name, Kind, and Description. A message at the bottom of the table says 'This credential domain is empty. How about adding some credentials?'. There are buttons for '+ Add Credentials' and 'Icon: S M L'.

Add a credential.

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username ?
deployer

Treat username as secret ?

Password ?
.....

ID ?

Create

Create username and password.

Dashboard > proj > Configuration

Configure

Connect and manage your code repository to automatically pull the latest code for your builds.

General

Source Code Management

Triggers

Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Git

Repositories

Repository URL ?

https://github.com/ManaswiniSeshagiri/proj.git

Credentials

- none -

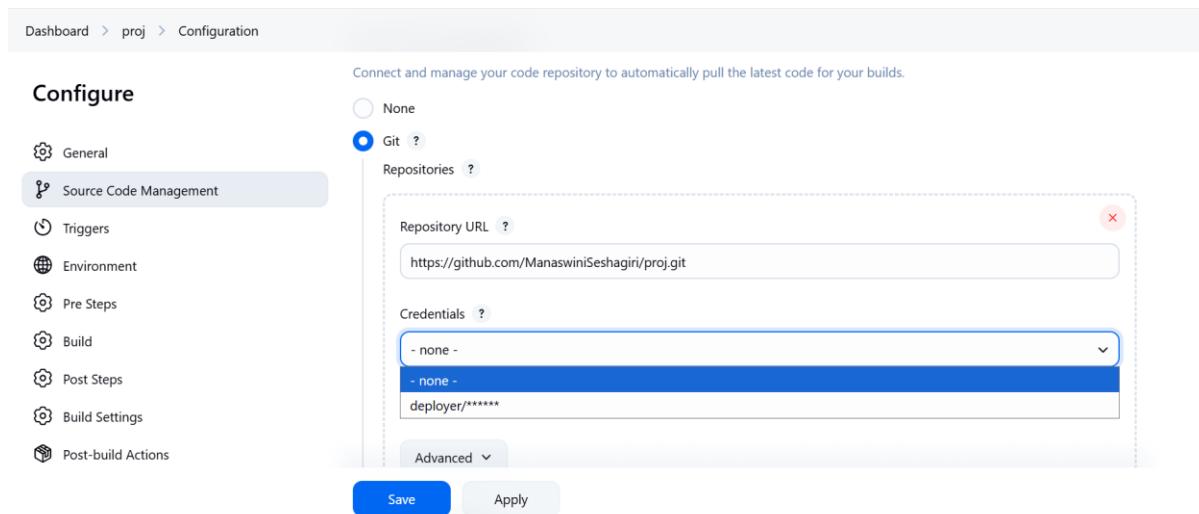
- none -

deployer/*****

Advanced

Save

Apply



Go to maven project and click configure and add the credentials in git.

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Context path ?

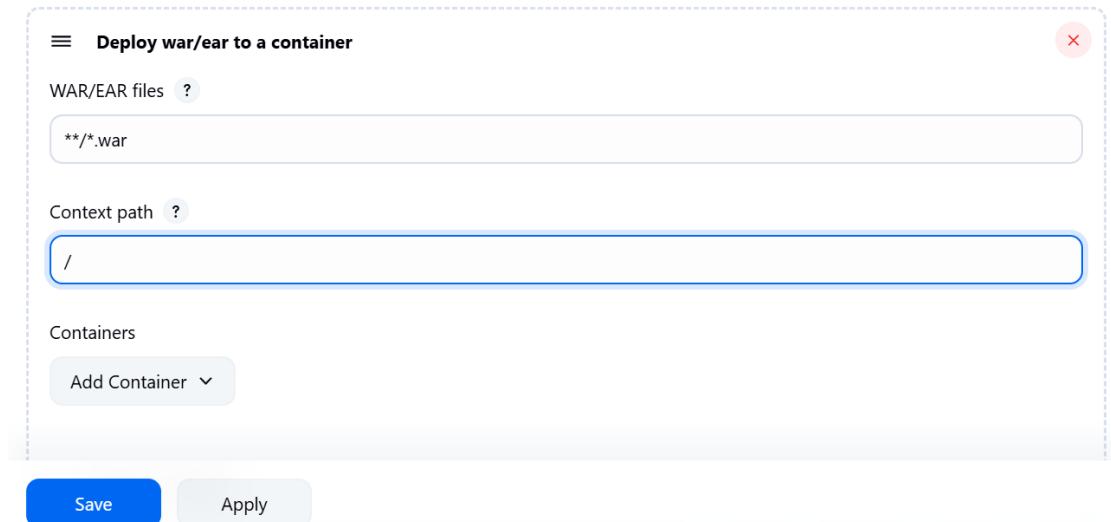
/

Containers

Add Container ▾

Save

Apply



Add post-build actions and add deploy war/ear.

Dashboard > proj > Configuration

Configure

General

Source Code Management

Triggers

Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Tomcat 9.x Remote

Credentials

- none -

- none -

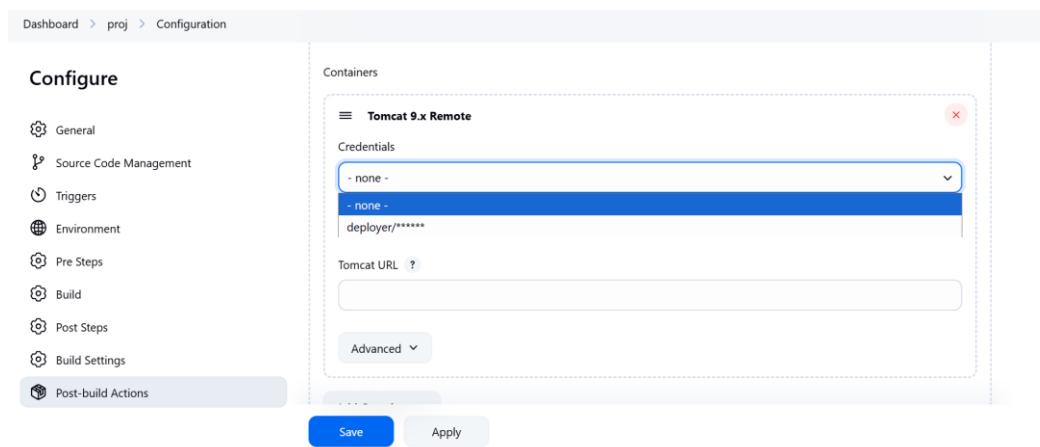
deployer/*****

Tomcat URL ?

Advanced

Save

Apply



Add container and credentials.

 Build Now

 Configure

 Delete Maven project

 Modules

 Rename

Builds

...



 Filter

/

Today

 #4 6:24 AM

▼

Now click build now.

New user Register for DevOps Learning at Virtual TechBox Youtube Channel

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

Now go to tomcat and refresh. You'll find the web page.

← → ⌛ ⌂ Not secure 52.23.168.92:8080/job/proj/configure ☆ ⌁ ⌂ ⌃ ⌄

LTIMindtree Favorites Folder 📂 Apache Tomcat® -... 📺 Step by Step guide t... 🎙 sanjayguruji/java-co...

Dashboard > proj > Configuration

Configure

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

Build whenever a SNAPSHOT dependency is built ?

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

GitHub Branches

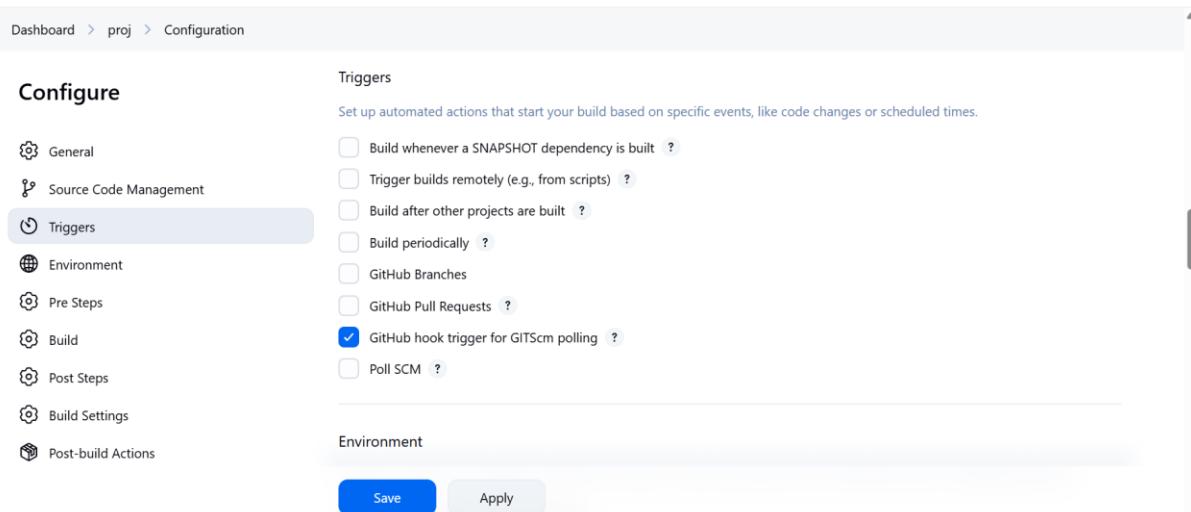
GitHub Pull Requests ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Environment

Save **Apply**



Now make changes in the repository and go to project configuration and click Github hook trigger for GITScm polling for automatic build if any changes made.

Builds

... ↗

 Filter

/

Pending

 #5



| Finished waiting

Today

 #4 6:24 AM



 #3 6:12 AM



 #2 6:02 AM



You can see automatic build.

← → ⌛ ⌂ ⌂ Not secure 44.212.76.137:8080 ☆ 🔍 ⓘ

CTM Mindtree Favorites Folder 📁 | 🖥 Apache Tomcat® -... 📚 Step by Step guide t... 🌐 sanjayguruji/java-co...

This is Manaswini's form

Please fill in this form to create an account.

Enter Name	<input type="text"/>
Enter mobile	<input type="text"/>
Enter Email	<input type="text"/>
Password	<input type="password"/>
Repeat Password	<input type="password"/>

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

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

Thank You, Happy Learning

See You Again

You'll find the changes in the web page.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The top navigation bar includes the AWS logo, search bar, and account information ('United States (N. Virginia) Manaswi P S G'). The main steps are 'EC2 > Instances > Launch an instance'. The current step is 'Launch an instance' with a sub-step 'Name and tags'. A summary panel on the right shows: 1 instance, Amazon Linux 2023 AMI 2023.7.2..., t2.micro instance type, and a new security group. Buttons for 'Cancel' and 'Launch instance' (highlighted in orange) are present.

Create a new instance

Name: docker-server

Instance type: t2.micro

```
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\10835497> cd downloads
PS C:\Users\10835497\downloads>
PS C:\Users\10835497\downloads> ssh -i "lti-mahape.pem" ec2-user@ec2-13-221-185-143.compute-1.amazonaws.com
The authenticity of host 'ec2-13-221-185-143.compute-1.amazonaws.com (13.221.185.143)' can't be established.
ED25519 key fingerprint is SHA256:ssub7cLB5opXfKBC/aw/3YeKW4WBUBPoS3QVmalxSKE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-221-185-143.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-172-31-89-163 ~]$ sudo su -
[root@ip-172-31-89-163 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-89-163 ~]# bash
```

Connect the instance with terminal.

```
[root@docker ~]# yum install docker*
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
=====
 Package           Architecture   Version      Repository    Size
=====
Installing:
 docker            x86_64        25.0.8-1.amzn2023.0.4  amazonlinux  44 M
Installing dependencies:
 container-selinux  noarch       3:2.233.0-1.amzn2023
 containerd         x86_64       2.0.5-1.amzn2023.0.1  amazonlinux  25 M
 iptables-libs     x86_64       1.8.8-3.amzn2023.0.2  amazonlinux  401 k
 iptables-nft      x86_64       1.8.8-3.amzn2023.0.2  amazonlinux  183 k
 libcgroup          x86_64       3.0-1.amzn2023.0.1   amazonlinux  75 k
 libnetfilter_conntrack x86_64       1.0.8-2.amzn2023.0.2  amazonlinux  58 k
 libnfnetwork      x86_64       1.0.1-19.amzn2023.0.2  amazonlinux  30 k
 libnftnl           x86_64       1.2.2-2.amzn2023.0.2  amazonlinux  84 k
 pigz              x86_64       2.5-1.amzn2023.0.3   amazonlinux  83 k
 runc              x86_64       1.2.4-2.amzn2023.0.1   amazonlinux  3.4 M
=====
Transaction Summary
=====
Install 11 Packages

Total download size: 74 M
Installed size: 288 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): container-selinux-2.233.0-1.amzn2023.noarch.rpm           1.7 MB/s | 55 kB  00:00
(2/11): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm           9.8 MB/s | 401 kB  00:00
```

Install docker in terminal.

```
[root@docker .aws]# cat config
[default]
region = us-east-1
output = table
[root@docker .aws]# cat credentials
[default]
aws_access_key_id = Pwz00uR0uIqss+sBJqvqh/z7hDqmQznIXJTkVdH
aws_secret_access_key = Pwz00uR0uIqss+sBJqvqh/z7hDqmQznIXJTkVdH
[root@docker .aws]#
```

Create a new user in IAM and give the access key id and secret access key id.

```
[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 12:45:5a:d7:a1:b7 brd ff:ff:ff:ff:ff:ff
    altname eni-025fc3a9bb529233
    altname device-number=0.0
    inet 172.31.89.163/20 metric 512 brd 172.31.95.255 scope global dynamic enX0
        valid_lft 2619sec preferred_lft 2619sec
    inet6 fe80::1045:5aff:fed7:a1b7/64 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
    link/ether 02:42:89:9f:d2:ed 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 ~]# systemctl restart sshd
[root@docker ~]# systemctl enable sshd
[root@docker ~]#
```

Restart and enable ssh in terminal.

Dashboard > proj > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions**

Source files ?

Remove prefix ?

Remote directory ?

/opt

Exec command ?

```
rsync -avh /var/lib/jenkins/workspace/proj/ root@172.31.80.142:/opt
```

Save **Apply**

Take the details of Jenkins terminal and connect.

Dashboard > proj > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions**

Source files ?

Remove prefix ?

Remote directory ?

Exec command ?

```
cd /opt/
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
334935311897.dkr.ecr.us-east-1.amazonaws.com
docker build -t proj .
docker tag proj:latest 334935311897.dkr.ecr.us-east-1.amazonaws.com/proj:latest
```

Save **Apply**

Take the details of docker terminal and connect.

Dashboard > proj >

- </> Changes
- Workspace
- ▷ Build Now
- Configure
- Delete Maven project
- Modules
- GitHub Hook Log
- Rename

Builds ...

Filter

Today #16 9:07 AM

Latest Test Result (no failures)

Permalinks

- Last build (#16), 2 min 3 sec ago
- Last stable build (#16), 2 min 3 sec ago
- Last successful build (#16), 2 min 3 sec ago
- Last unstable build (#15), 4 min 12 sec ago
- Last unsuccessful build (#15), 4 min 12 sec ago
- Last completed build (#16), 2 min 3 sec ago

Test Result Trend

The chart shows a series of green squares representing successful builds, spanning from build #1 to #16. A legend at the top right indicates that green dots represent 'Passed', grey dots represent 'Skipped', and red dots represent 'Failed'.

Check for the build.

The screenshot shows the AWS ECR console. On the left, there's a sidebar with sections for 'Amazon Elastic Container Registry', 'Private registry' (Repositories, Images, Permissions, Lifecycle Policy, Repository tags, Features & Settings), and 'Public registry' (Repositories, Settings). The main area is titled 'Images (1)' and shows a table with one row. The table columns are: Image tag, Artifact type, Pushed at, Size (MB), Image URI, Digest, and Last recorded pull time. The single entry is 'latest' (Image type), pushed on June 28, 2025, at 14:38:13 (UTC+05.5), with a size of 227.79 MB, an image URI starting with 'Copy URI', and a digest 'sha256:33859afa...'. There are buttons for 'Delete', 'Details', 'Scan', and 'View push commands'.

Now check for image if available in Elastic Container Registry(ECR).

The screenshot shows the AWS EC2 Instances launch wizard. The top navigation bar includes 'CloudShell' and 'Feedback'. A blue banner at the top says 'It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices' with 'Do not show me this message again' and 'Take a walkthrough' buttons. The main form is titled 'Launch an instance' and contains fields for 'Name and tags' (Name: 'eks-server', Add additional tags button), 'Application and OS Images (Amazon Machine Image)' (Info link, note about AMIs, search/browse link), 'Summary' (Number of instances: 1, Software Image (AMI) set to 'Amazon Linux 2023 AMI 2023.7.2...read more' (ami-05ffe3c48a9991135), Virtual server type (instance type) set to 't2.medium', Firewall (security group) set to 'launch-wizard-1'). The bottom navigation bar includes 'CloudShell' and 'Feedback'.

Now create a new instance:

Name: eks-server

Instance type: t2.medium

```

PS C:\Users\10835497> cd downloads
PS C:\Users\10835497\downloads> ssh -i "ltim.pem" ec2-user@ec2-3-94-79-231.compute-1.amazonaws.com
The authenticity of host 'ec2-3-94-79-231.compute-1.amazonaws.com (3.94.79.231)' can't be established.
ED25519 key fingerprint is SHA256:T8sDp2SXWCmxZ4SVqU7ehkmeO+XNWTxH9XL+qFyM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-94-79-231.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-172-31-87-205 ~]$ sudo su -
[root@ip-172-31-87-205 ~]# hostnamectl set-hostname eks.example.com
[root@ip-172-31-87-205 ~]# bash
[root@eks ~]#
[root@eks ~]# cd
[root@eks ~]# 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:pMCCR57PA3Wq3qV4fm8SfyWPylGQlcuU7od/WaHnQ root@eks.example.com
The key's randomart image is:

```

Connect the instance to terminal and generate ssh key.

```

[root@eks ~]# yum install unzip -y
Last metadata expiration check: 0:00:55 ago on Sat Jun 28 10:23:09 2025.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@eks ~]# 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  103M      0 --:--:-- --:--:-- 103M
[root@eks ~]# 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/

```

Do the installation process.

```

[root@eks ~]# eksctl create cluster --name manas-proj --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 11:11:01 [■] eksctl version 0.210.0
2025-06-28 11:11:01 [■] using region us-east-1
2025-06-28 11:11:01 [■] setting availability zones to [us-east-1f us-east-1d]
2025-06-28 11:11:01 [■] subnets for us-east-1f - public:192.168.0.0/19 private:192.168.64.0/19
2025-06-28 11:11:01 [■] subnets for us-east-1d - public:192.168.32.0/19 private:192.168.96.0/19
2025-06-28 11:11:01 [■] nodegroup "ng-222a2373" will use "" [AmazonLinux2023/1.32]
2025-06-28 11:11:01 [■] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-manas-proj-nodegroup-ng-222a2373-0f:32:e3:f2:63:67:d
2:7a:2d:35:c3:78:a9:df:98:63"
2025-06-28 11:11:01 [■] using Kubernetes version 1.32
2025-06-28 11:11:01 [■] creating EKS cluster "manas-proj" in "us-east-1" region with managed nodes
2025-06-28 11:11:01 [■] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-06-28 11:11:01 [■] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-e
ast-1 --cluster=manas-proj'
2025-06-28 11:11:01 [■] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "man
as-proj" in "us-east-1"
2025-06-28 11:11:01 [■] CloudWatch logging will not be enabled for cluster "manas-proj" in "us-east-1"
2025-06-28 11:11:01 [■] 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=manas-proj'
2025-06-28 11:11:01 [■] default addons kube-proxy, coredns, metrics-server, vpc-cni were not specified, will install them as EKS add
ons
2025-06-28 11:11:01 [■]
2 sequential tasks: { create cluster control plane "manas-proj",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-222a2373",
  }
}
2025-06-28 11:11:01 [■] building cluster stack "eksctl-manas-proj-cluster"

```

Create a new cluster.

```
[root@eks ~]# kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-28-23.ec2.internal  Ready   <none>  86s   v1.32.3-eks-473151a
ip-192-168-3-78.ec2.internal  Ready   <none>  90s   v1.32.3-eks-473151a
ip-192-168-33-139.ec2.internal Ready   <none>  90s   v1.32.3-eks-473151a
[root@eks ~]# 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 ~]# cd .ssh/
[root@eks .ssh]# ssh-copy-id root@172.31.84.30
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.84.30 (172.31.84.30)' can't be established.
ED25519 key fingerprint is SHA256:FIIPow6GH7qmHW0pdhpNhfj7AmWdR59gdFG15HCqehXTs.
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@172.31.84.30's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.84.30'"
```

```
[root@jenkins ~]# cd .ssh/
[root@jenkins .ssh]# ssh-copy-id root@172.31.87.205
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.87.205 (172.31.87.205)' can't be established.
ED25519 key fingerprint is SHA256:T8sDp2bSXWCVmxB4SVqU7ehkmeO+XNWtxH9XL+qEFyM.
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@172.31.87.205's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.87.205'" and check to make sure that only the key(s) you wanted were added.

[root@jenkins .ssh]# |
```

The screenshot shows the Jenkins Manage Jenkins interface with the 'SSH Server' configuration page open. The 'Name' field contains 'eks-boot-strap'. The 'Hostname' field contains '172.31.87.205'. The 'Username' field contains 'root'. The 'Remote Directory' field is empty. At the bottom, there are 'Save' and 'Apply' buttons.

Now add the details of eks-server instance into Jenkins.

```
[root@eks .ssh]# cd
[root@eks ~]# mkdir /code
[root@eks ~]# cd /code
[root@eks code]# pwd
/code
[root@eks code]# cd /opt/
[root@eks opt]# vim regapp-deploy.yml

[root@eks opt]# vim regapp-service.yml

[root@eks opt]# |
```

Make a directory and create two yml files.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: regapp-deployment
  labels:
    app: regapp

spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp

  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
        - name: regapp
          image: 491087500210.dkr.ecr.us-east-1.amazonaws.com/proj:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxSurge: 1
          maxUnavailable: 1
      ~
      ~
      ~
-- INSERT --
```

regapp-deploy.yml is created

The screenshot shows the GitHub Code interface for a project named 'proj'. The left sidebar lists files in the 'main' branch, including 'index.jsp', 'pom.xml', 'Dockerfile', 'README.md', 'pom.xml', 'regapp-deploy.yml', and 'regapp-service.yml'. The right panel displays the content of 'index.jsp'. The code is a JSP page with form fields for Name, mobile number, and email.

```

1 <form action="action_page.php">
2   <div class="container">
3     <h1>New user Register for DevOps Learning at Manaswini</h1>
4     <p>Please fill in this form to create an account.</p>
5     <br>
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>
16    <input type="text" placeholder="Enter Email" name="email" id="email" required>
17    <br>

```

Make changes in the jsp file.

The Jenkins dashboard for the 'proj' project shows the following details:

- Changes:** Shows the last 16 builds, all of which have passed.
- Workspace:** Build Now button.
- Configure:** Delete Maven project, Modules, GitHub Hook Log, Rename buttons.
- Permalinks:** Latest Test Result (no failures).
- Test Result Trend:** A chart showing the status of each build from #1 to #16. All builds are green (Passed).
- Builds:** A table showing the last build (#17) was successful at 11:35 AM today.

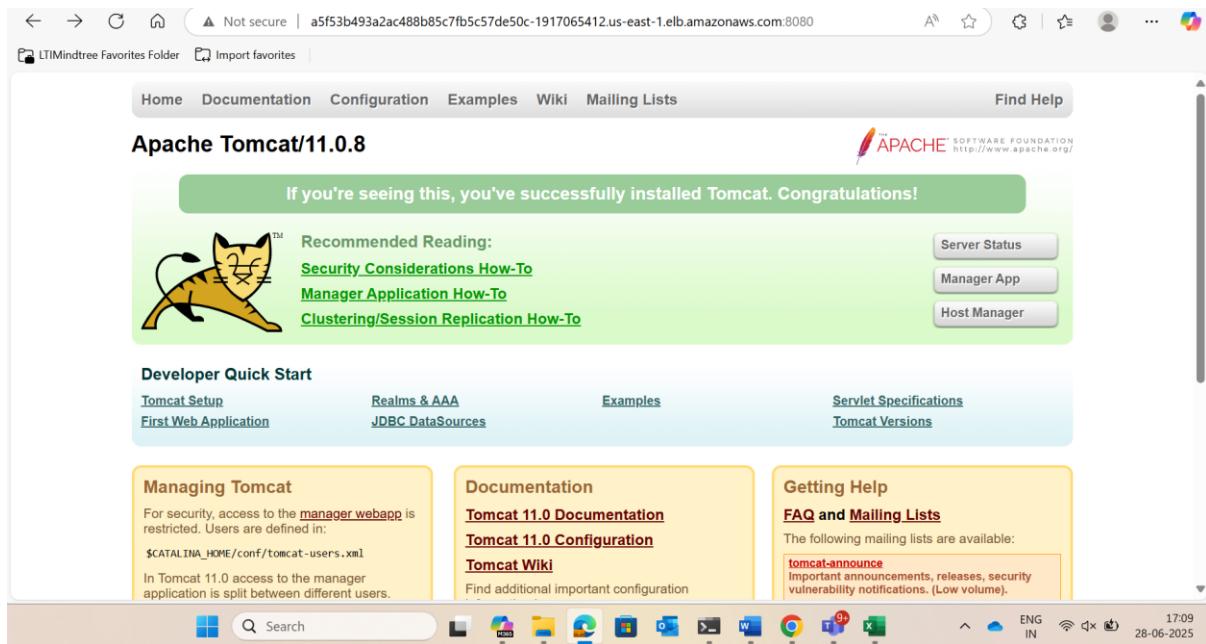
Now build and check it.

```

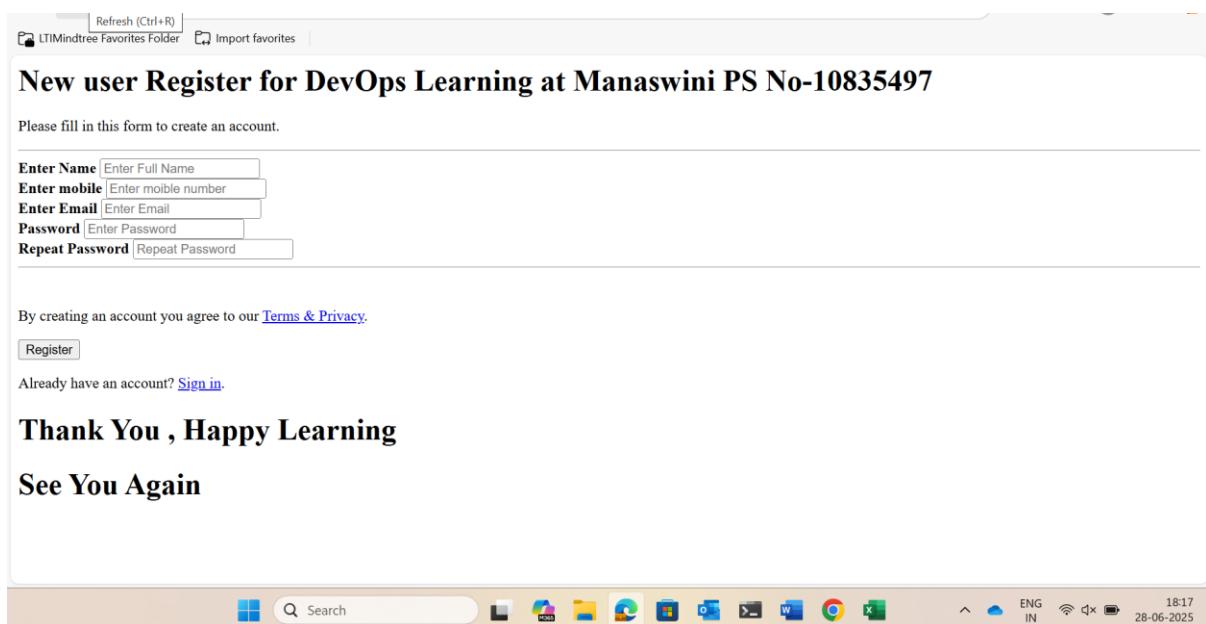
[root@eks opt]# kubectl get pod
NAME                               READY   STATUS    RESTARTS   AGE
regapp-deployment-64cc49bc97-2lwdb  1/1     Running   0          52s
regapp-deployment-64cc49bc97-grwdv  1/1     Running   0          52s
[root@eks opt]# kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)
AGE
kubernetes  ClusterIP  10.100.0.1  <none>        443/TCP
19m
regapp-service  LoadBalancer  10.100.120.15  a5f53b493a2ac488b85c7fb5c57de50c-1917065412.us-east-1.elb.amazonaws.com  8080:31180/
TCP 58s
[root@eks opt]#

```

Now paste the regapp-service file's external IP in browser with :8080 and check.



Opened in browser.



Now add /webapp to the link and you can see the page.