



## **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION)**



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH,  
Narayanguda, Hyderabad, Telangana – 500029**



### **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

#### **LAB RECORD**

#### **SOFTWARE ENGINEERING LAB**

**B. Tech. III YEAR I SEM (KR23)  
ACADEMIC YEAR  
2025-26**



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY  
(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad  
Narayanguda, Hyderabad, Telangana – 500029



## Certificate

This is to certify that following is a Bonafide Record of the workbook task done by

\_\_\_\_\_ bearing Roll No \_\_\_\_\_ of \_\_\_\_\_

Branch of \_\_\_\_\_ year B. Tech. Course in the \_\_\_\_\_

Subject during the Academic year \_\_\_\_\_ & \_\_\_\_\_ under our supervision.

Number of week tasks completed: \_\_\_\_\_

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner



# **KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

## **(AN AUTONOMOUS INSTITUTION)**



**Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad  
Narayanguda, Hyderabad, Telangana – 500029**

## **Daily Laboratory Assessment Sheet**

Name of the Lab:  
Branch & Section:

Student Name:  
HT. No:

## **Faculty Incharge**

## INDEX

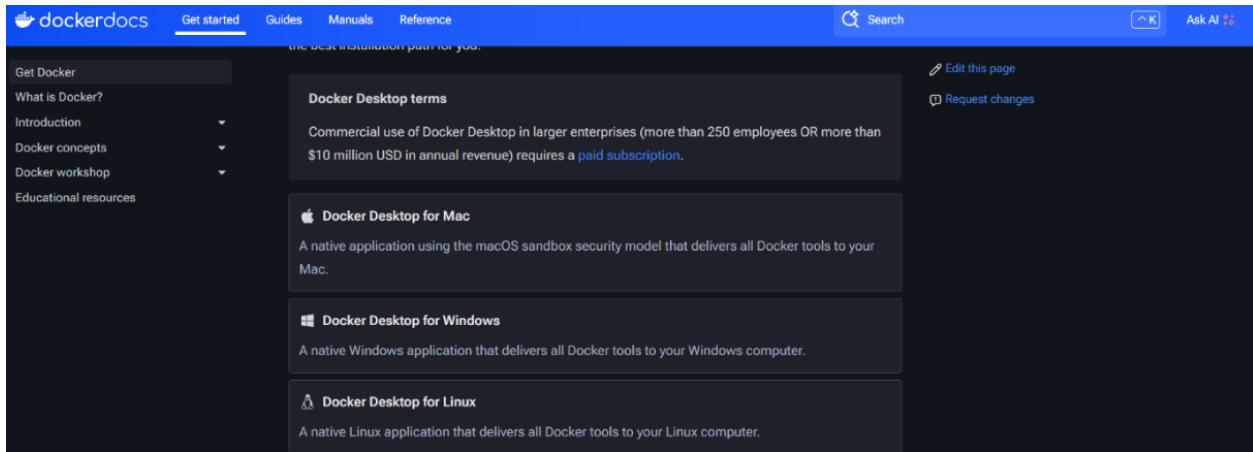
<b>Sr. NO.</b>	<b>CONTENTS</b>	<b>PAGE NO.</b>
1.	<p><b>Software Installation &amp; SRS Document</b></p> <ul style="list-style-type: none"> <li>a. Abstract</li> <li>b. Functional Requirements (FR)</li> <li>c. Non-Functional Requirements (NFR)</li> <li>d. User Identification</li> <li>e. Workflow of Each User</li> <li>f. Use Cases</li> </ul>	
2.	<p><b><u>Exploring git local and remote commands on the multi-folder project</u></b></p> <ul style="list-style-type: none"> <li>a. Pushing multi-folder project into private repository (by student).</li> <li>b. Students must explore all listed git commands on the multi-folder project in local and remote repository.</li> <li>c. Students must explore all git commands on given scenario-based question</li> </ul>	
3.	<p><b>Collaborative coding using git</b></p> <ul style="list-style-type: none"> <li>a. To work on collaborative coding by:</li> <li>b. Creating Organization.</li> <li>c. Coordinating with others through a shared repository</li> <li>d. To resolve conflicts when collaborating on same part of code.</li> <li>e. To create and apply patch.</li> </ul>	
4.	<p><b><u>Build and package Java and Web applications using Maven</u></b></p> <ul style="list-style-type: none"> <li>a. Understand the structure and lifecycle of a Maven project.</li> <li>b. Build and package Java and Web applications using Maven.</li> <li>c. Add dependencies using <b>pom.xml</b>, compile and test using plugins.</li> <li>d. Resolve errors and conflicts arising from dependency mismatches.</li> <li>e. Work with parent and multi-module Maven projects.</li> <li>f. Generate executable JARs and deployable WARs using Maven.</li> </ul>	
5.	<p><b>Docker CLI commands</b></p> <ul style="list-style-type: none"> <li>a. Learn how to pull, run, stop, start, remove, and inspect containers and images.</li> <li>b. Gain the ability to create, monitor, and troubleshoot running containers.</li> <li>c. Configure and manage networks for container communication.</li> <li>d. Create and manage persistent storage for containers.</li> <li>e. Learn how to list, remove, and manage images efficiently.</li> </ul>	
6.	<p><b><u>Docker</u></b></p> <ul style="list-style-type: none"> <li>a. Learn how to define and run multiple interdependent services (e.g., web server, database) in a single configuration file.</li> </ul>	

	<ul style="list-style-type: none"> <li>b. Gain skills in writing and interpreting docker-compose.yml files for service setup.</li> <li>c. Deploy the same setup across different machines without manual configuration.</li> <li>d. Configure container networking and persistent storage within Compose.</li> <li>e. Reduce setup time and enable faster iteration during application development.</li> </ul>	
7.	<p><b><u>Creating a Multi-Module Maven Project</u></b></p> <ul style="list-style-type: none"> <li>a. Build and package Java and Web applications using Maven.</li> <li>b. Add dependencies using <b>pom.xml</b>, compile and test using plugins.</li> <li>c. Resolve errors and conflicts arising from dependency mismatches.</li> <li>d. Work with parent and multi-module Maven projects.</li> <li>e. Generate executable JARs and deployable WARs using Maven</li> </ul>	
8.	<p><b><u>Jenkins Automation</u></b></p> <ul style="list-style-type: none"> <li>a. Hands-on practice on manual creation of Jenkins pipeline using Maven projects from Github</li> <li>b. Create the job and build the pipeline for maven-java and maven-web project.</li> </ul>	
9.	<p><b><u>Pipeline Creation using script</u></b></p> <ul style="list-style-type: none"> <li>a. Evaluation of Jenkins pipeline.</li> <li>b. WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE</li> <li>c. Hands-on practice on creation of scripted Jenkins pipeline.</li> <li>d. Take the screenshots for above task</li> </ul>	
10.	<p><b><u>Working with minikube and Nagios</u></b></p> <ul style="list-style-type: none"> <li>a. Hands-on practice of creating, running and scaling pods in minikube.</li> <li>b. Running Nginx server on specified port number by explaining the Nginx monitoring tool</li> <li>c. Running Nagios server and Understanding the Monitoring tool using Docker.</li> <li>d. AWS-free Trier account Creation steps</li> </ul>	
11.	<p><b><u>Jenkins-CI/CD</u></b></p> <ul style="list-style-type: none"> <li>a. CI-Continuous Integration using Webhooks.</li> <li>b. Sending E-mail Notification on Build Failure or success</li> </ul>	
12.	<p><b><u>Creation of virtual machine for Ubuntu OS and Deploying the web application</u></b></p> <ol style="list-style-type: none"> <li>1. Creation of virtual machine</li> <li>2. Deploying the web application</li> <li>3. Accessing it publicly</li> </ol>	

## **1. Software Installation & SRS Document:**

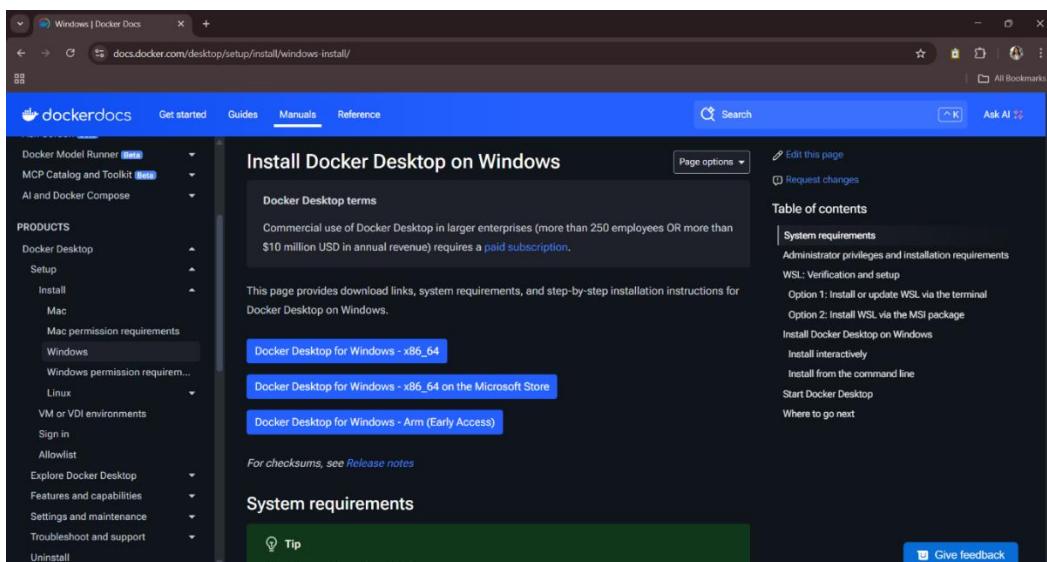
### **DOCKER- INSTALLATION**

Step-1: Go to docker website

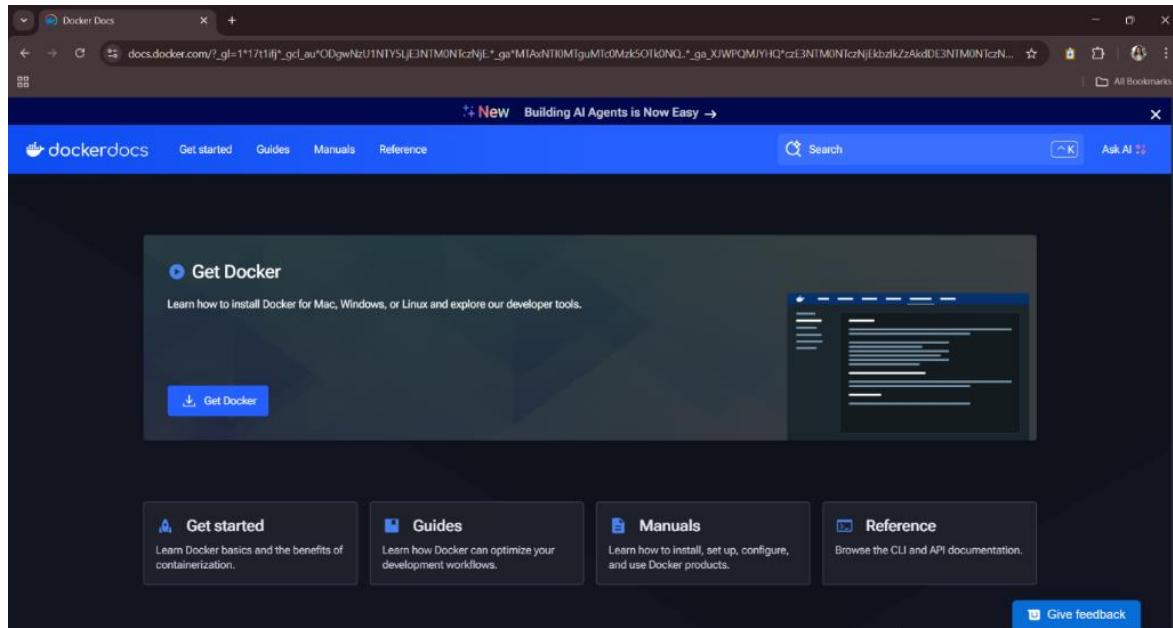


The screenshot shows the Docker Docs website with a dark theme. The top navigation bar includes links for 'Get started', 'Guides', 'Manuals', and 'Reference'. A search bar and an 'Ask AI' button are also present. On the left, a sidebar titled 'Get Docker' lists options like 'What is Docker?', 'Introduction', 'Docker concepts', 'Docker workshop', and 'Educational resources'. The main content area features a 'Docker Desktop terms' box stating that commercial use requires a paid subscription. Below are three sections: 'Docker Desktop for Mac', 'Docker Desktop for Windows', and 'Docker Desktop for Linux', each with a brief description.

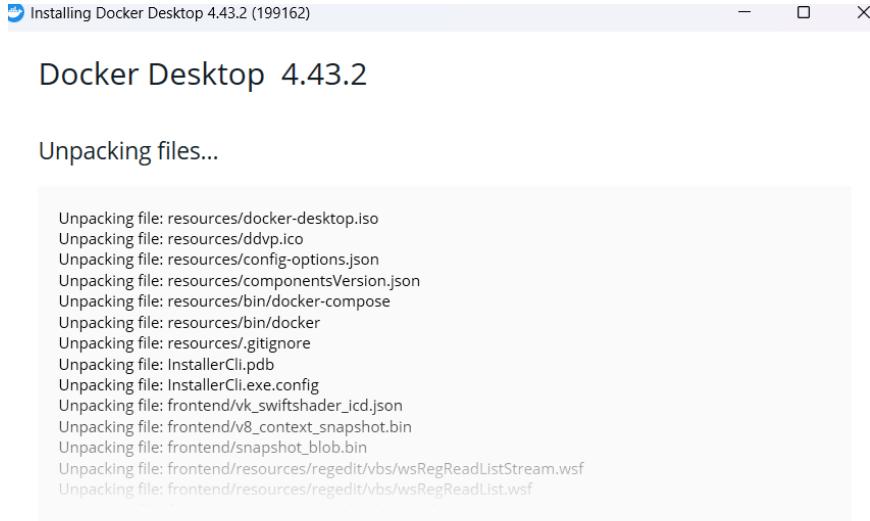
Step-2: Select the suitable one for your system



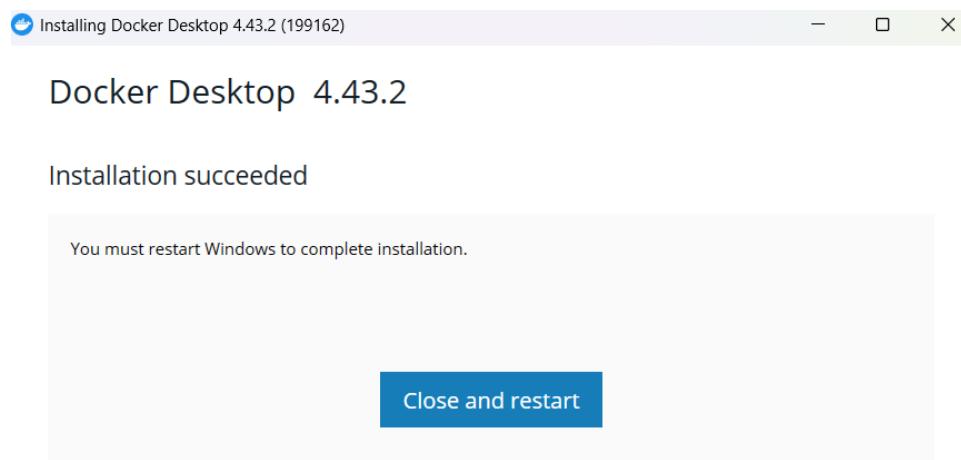
The screenshot shows the 'Install Docker Desktop on Windows' page from the Docker Docs website. The left sidebar includes sections for 'PRODUCTS' (Docker Model Runner, MCP Catalog and Toolkit, AI and Docker Compose) and 'Docker Desktop' (Setup, Install, Mac, Windows, Linux, VM or VDI environments, Sign in, Allowlist, Explore Docker Desktop, Features and capabilities, Settings and maintenance, Troubleshoot and support, Uninstall). The main content area has a 'Docker Desktop terms' box and a 'Table of contents' sidebar with links to system requirements, administrator privileges, WSL verification, and various installation options. Three download links are highlighted: 'Docker Desktop for Windows - x86\_64', 'Docker Desktop for Windows - x86\_64 on the Microsoft Store', and 'Docker Desktop for Windows - Arm (Early Access)'. A 'Tip' box and a 'Give feedback' button are at the bottom.



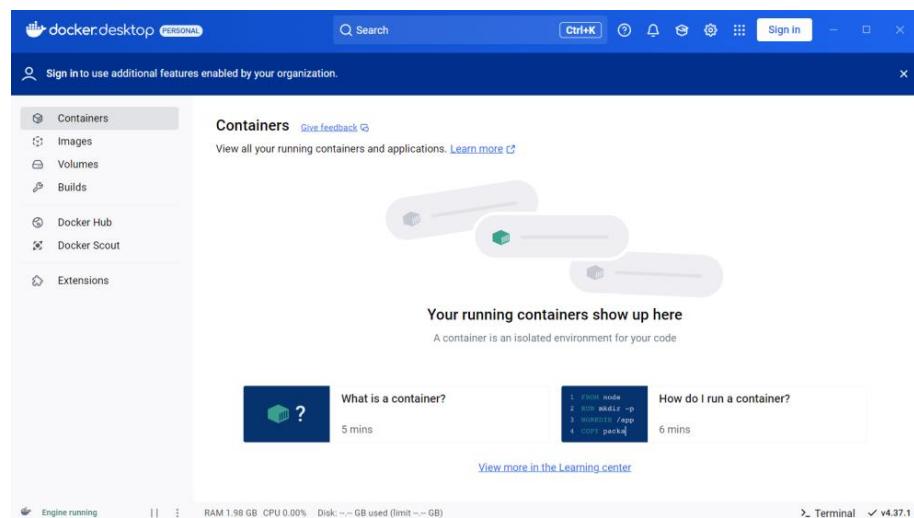
Step-3: After clicking on get docker it starts initializing



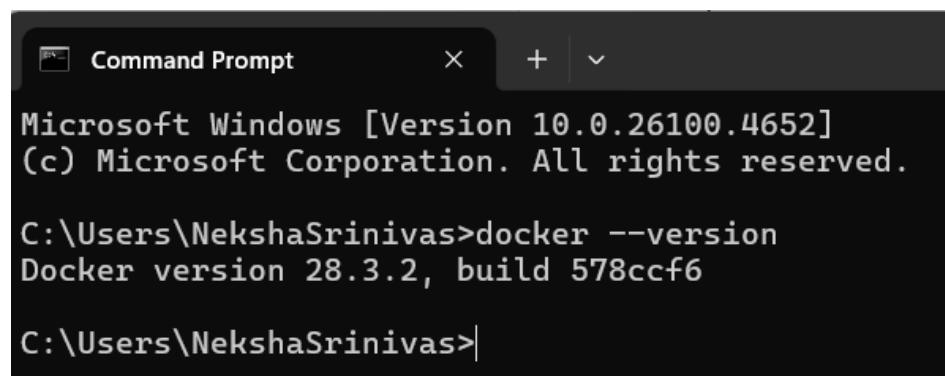
#### Step-4: Installation successful



#### Step-5: Docker interface



#### Step-6: docker version

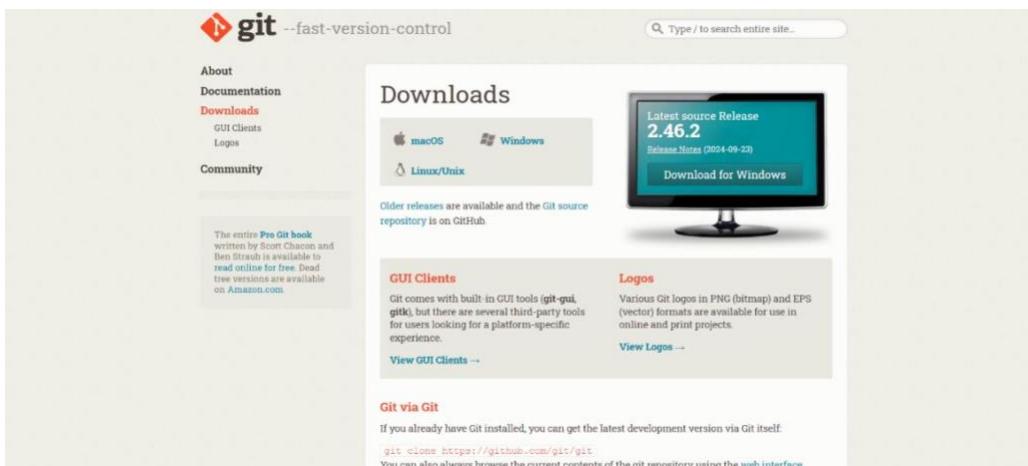


## GIT – INSTALLATION:

### Step-1: Go to Git website



Step-2: click on downloads and options will be displayed



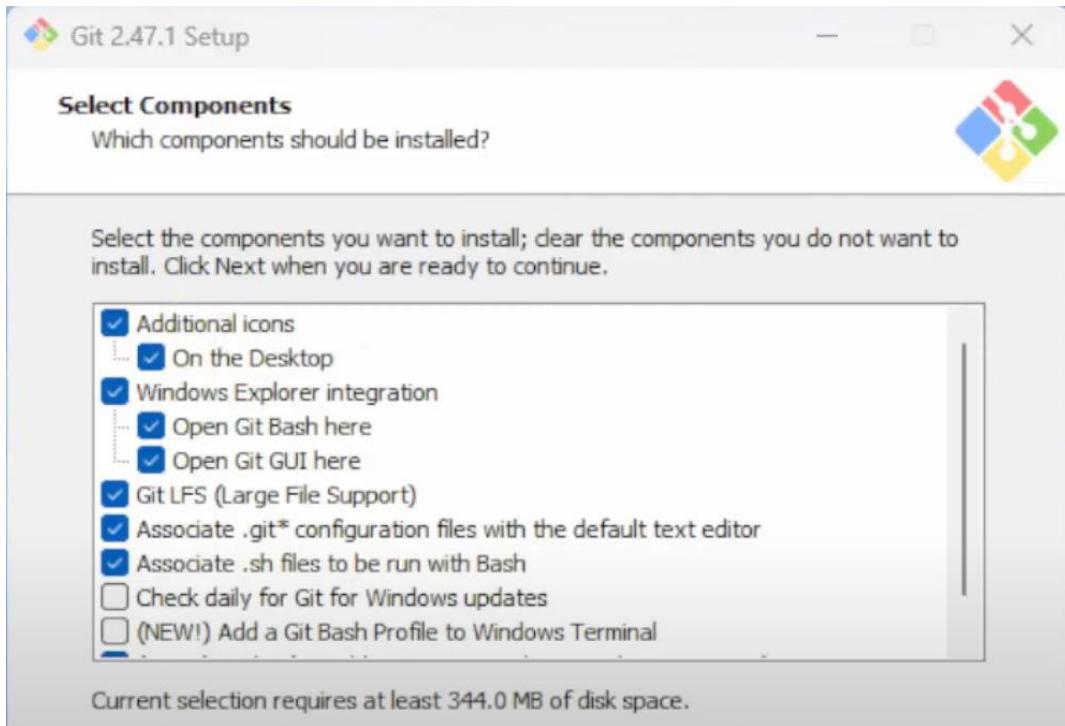
Step-3: Download for windows(suitable one for your system)



Step-4: License will be displayed click on next

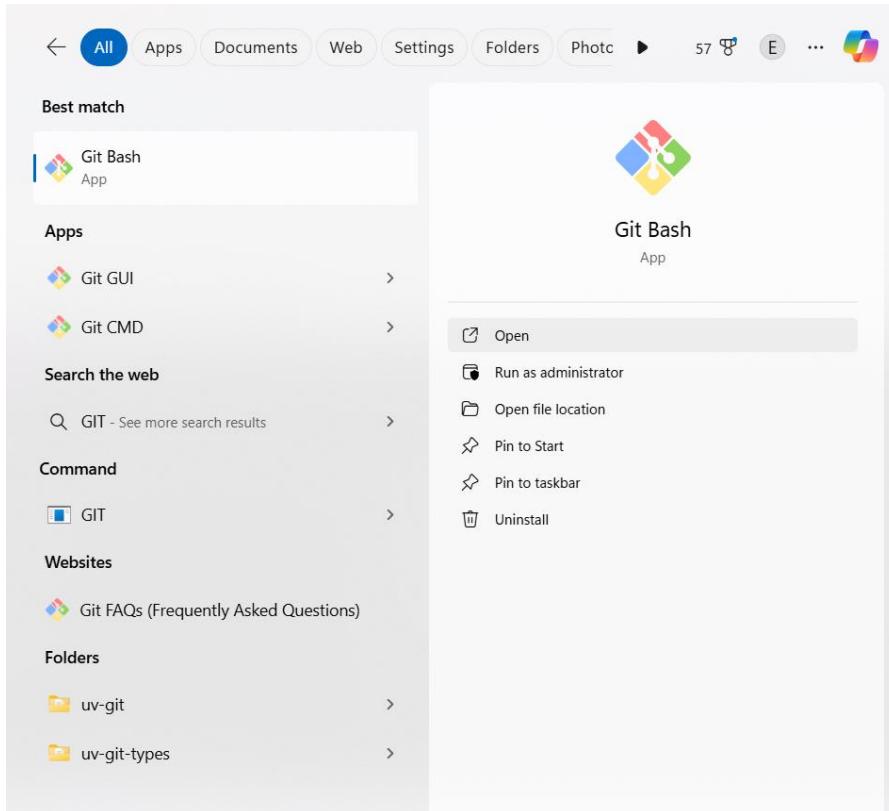


Step-5: Select the components and click next

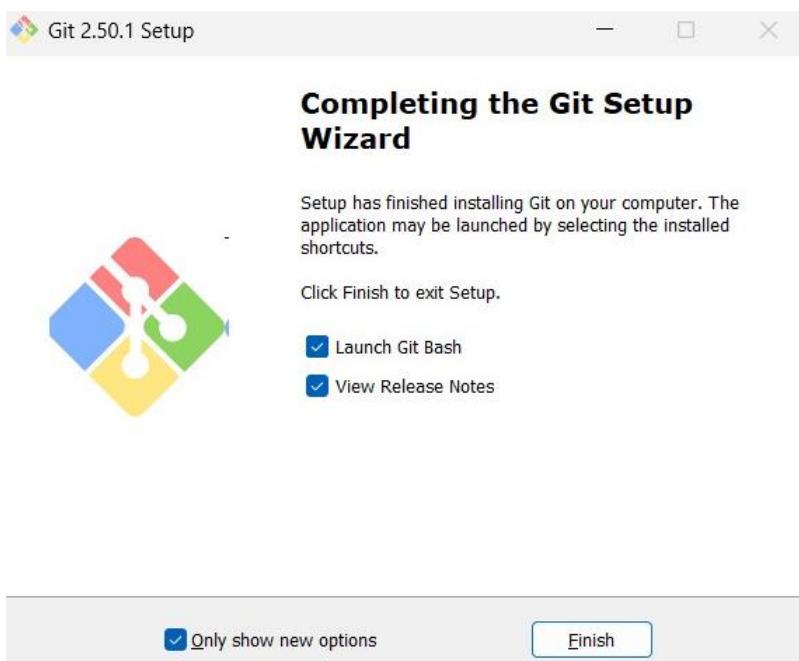


Git bash:

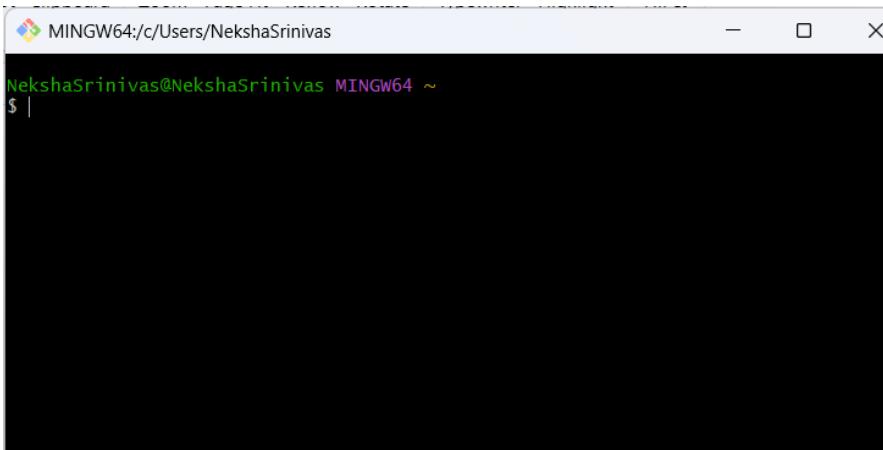
Step-1: Go to search bar and click git bash



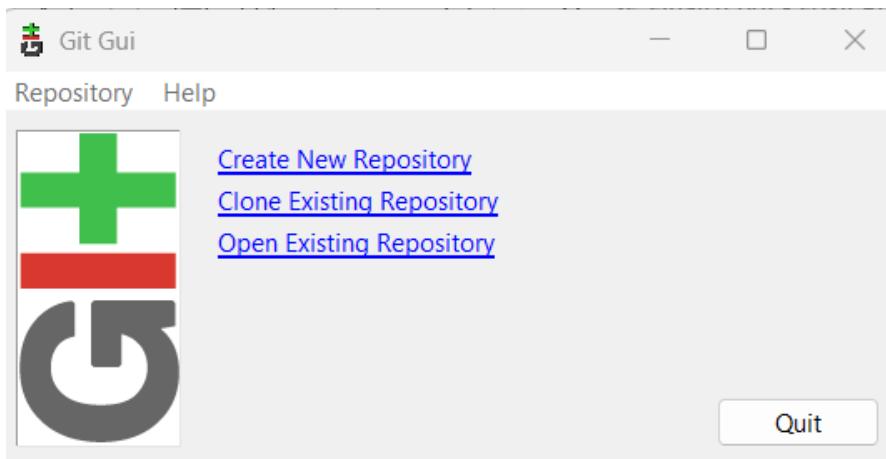
Step-2: Click on finish



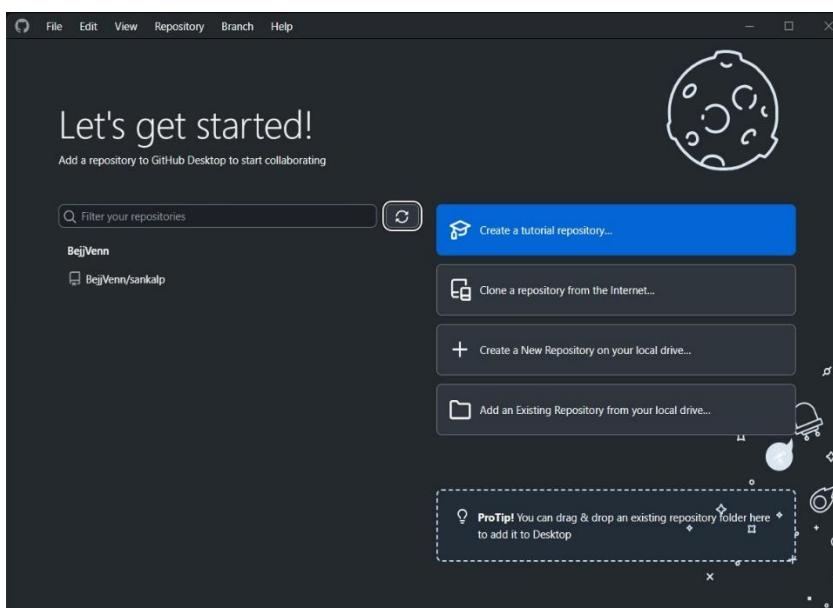
### Step-3: git bash interface



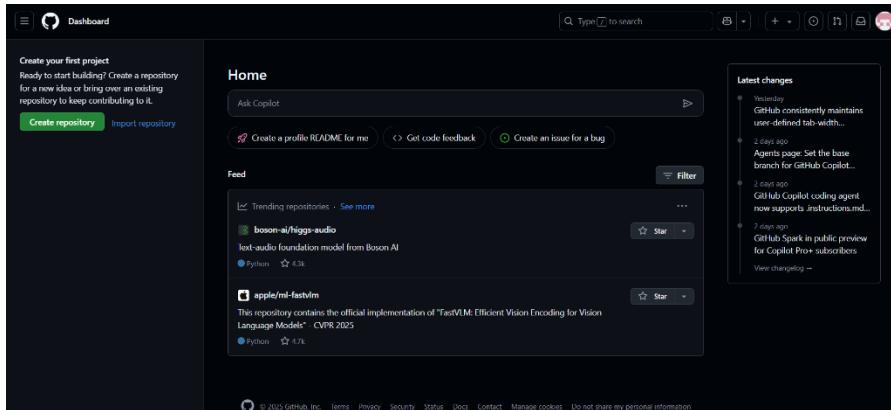
A screenshot of a terminal window titled "MINGW64:/c/Users/NekshaSrinivas". The window shows the command prompt "NekshaSrinivas@NekshaSrinivas MINGW64 ~" followed by a "\$ |". The rest of the screen is black, indicating it is a blank command line interface.



### GIT-ACCOUNT



## GIT-ACCOUNT



## Tomcat

A screenshot of the Apache Tomcat 9.0.98 homepage. The URL in the browser is 'localhost:8080'. The page features a large green banner at the top with the text 'If you're seeing this, you've successfully installed Tomcat. Congratulations!' and a cartoon cat icon. To the right of the banner are three buttons: 'Server Status', 'Manager App', and 'Host Manager'. Below the banner, there are four main sections: 'Developer Quick Start' (with links to 'Tomcat Setup', 'First Web Application', 'Realms &amp; AAA', 'JDBC DataSources', 'Examples', and 'Servlet Specifications'), 'Documentation' (with links to 'Tomcat 9.0 Documentation', 'Tomcat 9.0 Configuration', and 'Tomcat Wiki'), 'Getting Help' (with links to 'FAQ and Mailing Lists' and a list of mailing lists: 'tomcat-announce', 'tomcat-users', 'taglibs-user', and 'tomcat-dev'), and 'Managing Tomcat' (with links to 'Release Notes', 'Changelog', 'Migration Guide', 'Security Notices', 'SECURITY', 'MANAGING', 'RUNNING', 'TOMCAT', 'DEPLOYER', 'DATA-SOURCES', 'REALMS', and 'CONNECTORS'). At the bottom, there are links for 'Other Downloads' (Tomcat Connectors, Tomcat Native, Taglibs, Deployer), 'Other Documentation' (Tomcat Connectors, mod\_ik Documentation, Tomcat Native, Deployer), 'Get Involved' (Overview, Source Repositories, Mailing Lists, Wiki), 'Miscellaneous' (Contact, Legal, Sponsorship, Thanks), and 'Apache Software Foundation' (Who We Are, Heritage, Apache Home, Resources). A copyright notice at the very bottom reads 'Copyright ©1999-2025 Apache Software Foundation. All Rights Reserved'.

## Java and maven versions

```
Command Prompt
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

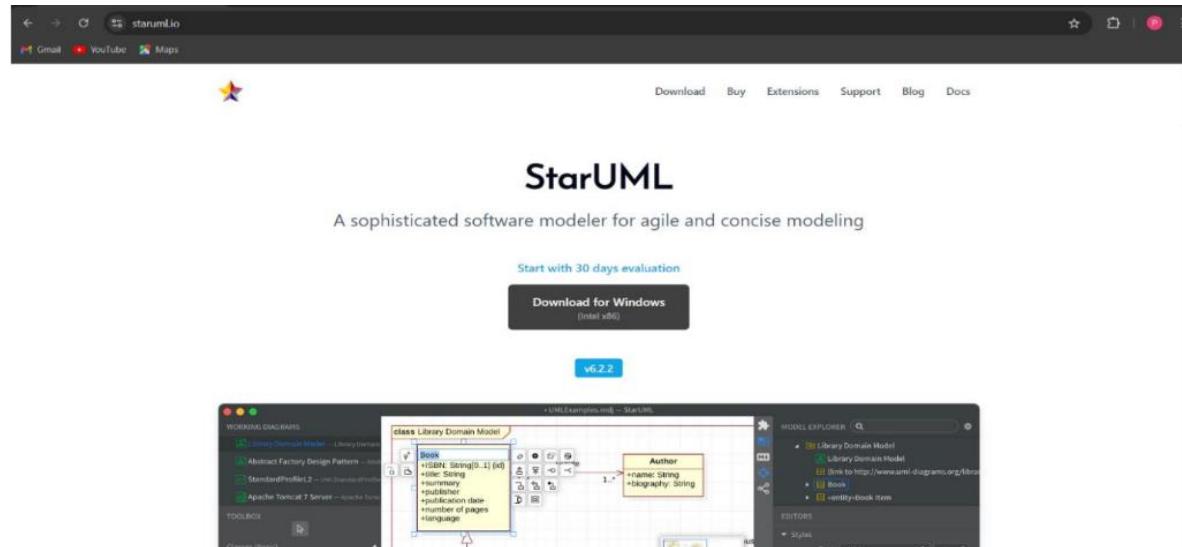
C:\Users\User>java --version
java 21.0.5 2024-10-15 LTS
Java(TM) SE Runtime Environment (build 21.0.5+9-LTS-239)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.5+9-LTS-239, mixed mode, sharing)

C:\Users\User>mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\apache-maven-3.9.9
Java version: 21.0.5, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\User>
```

## StarUML INSTALLATION

### Step-1: Go to startuml website



Step-2: from the given options select the suitable one for your system

The screenshot shows the Modelio download page. At the top, there is a navigation bar with links for Download, Buy, Extensions, Support, Blog, and Docs. Below the navigation bar, there is a large "Download" button. A message above the download button says "Start with 30 days evaluation". Below the download button, there are three main sections: "macOS 10.13 or higher" with two options ("macOS (Intel x86)" and "macOS (Apple arm64)"), "Windows 10 or higher" with two options ("Windows (x86-64bit)" and ".deb (x86-64bit)"), and "Ubuntu or Fedora" with two options ("Ubuntu (.deb)" and ".rpm (x86-64bit)"). Below these sections, a note says "If you want to download for previous versions, you can get a link for previous versions by [finding your license key](#)".

Step-3: Interface

The screenshot shows the Modelio interface. On the left, there is a "TOOLBOX" panel containing categories like "Classes (Basic)", "Class", "Interface", "Association", "Directed Association", "Aggregation", "Composition", "Dependency", "Generalization", and "Interface Realization". In the center, there is a workspace area with a grid background. On the right, there is a "MODEL EXPLORER" panel showing a tree structure with "Untitled" and "Model". At the bottom right of the workspace, there is a small preview window showing a UML class diagram. The bottom right corner of the interface has a status bar with "100%".

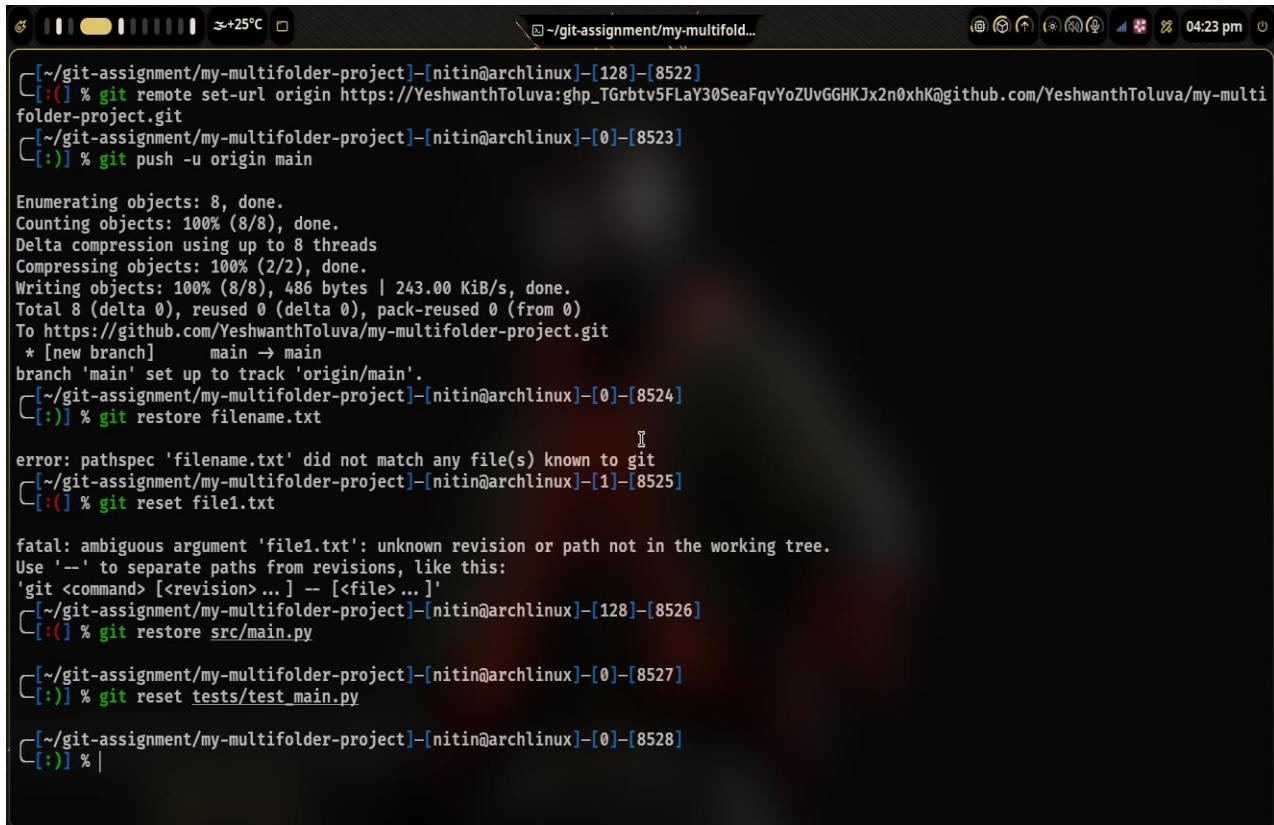
## 2. Exploring git local and remote commands on the multi-folder project

Github Global Configuration:

```
C:\Users\NekshaSrinivas>git config --global --list
core.editor="C:\Users\NekshaSrinivas\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=Edigirala-Neksha
user.email=edigiralaneksha@gmail.com

C:\Users\NekshaSrinivas>
```

Git Push to GitHub Public Repository with Remote Set



```
~/git-assignment/my-mulfolder-project-[nitin@archlinux]-[128]-[8522]
[::] % git remote set-url origin https://YeshwanthToluva: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@github.com/YeshwanthToluva/my-multi
folder-project.git
[::] % git push -u origin main

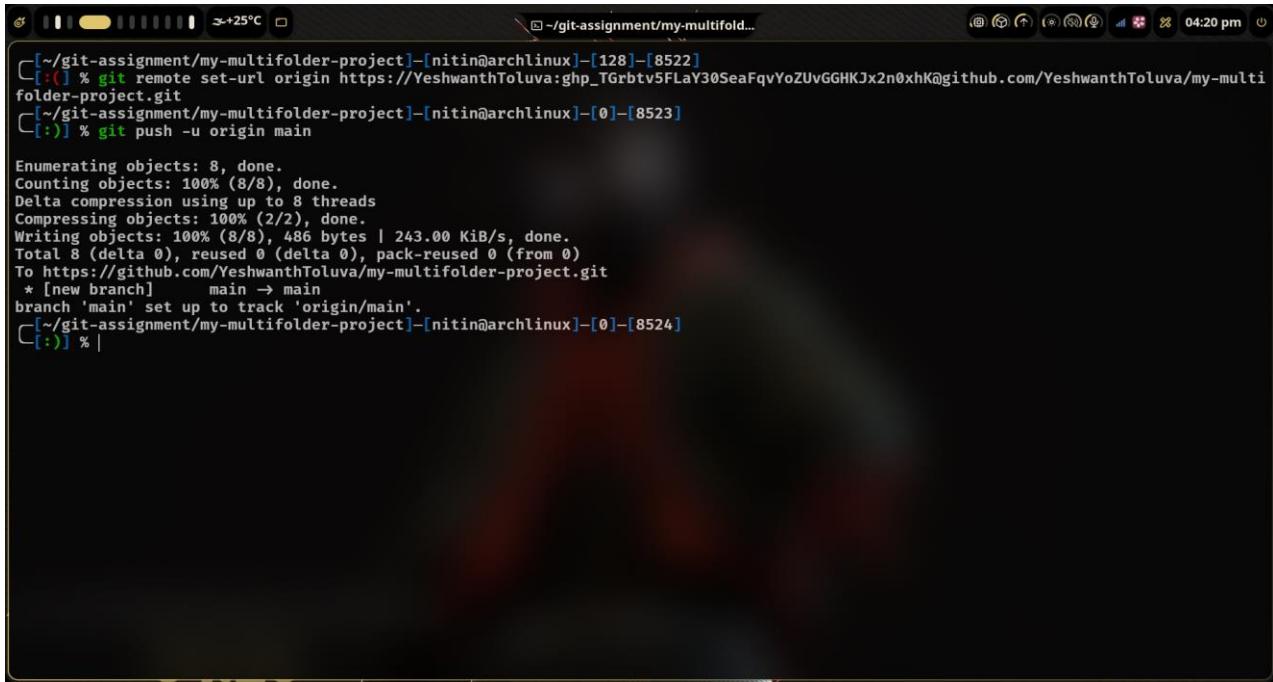
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (8/8), 486 bytes | 243.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YeshwanthToluva/my-mulfolder-project.git
 * [new branch]    main → main
branch 'main' set up to track 'origin/main'.
[::] % git restore filename.txt
error: pathspec 'filename.txt' did not match any file(s) known to git
[::] % git reset file1.txt
fatal: ambiguous argument 'file1.txt': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>... -- [<file> ... ]'
[::] % git restore src/main.py

[~/git-assignment/my-mulfolder-project-[nitin@archlinux]-[128]-[8526]
[::] % git reset tests/test_main.py

[~/git-assignment/my-mulfolder-project-[nitin@archlinux]-[0]-[8527]
[::] % git restore src/main.py

[~/git-assignment/my-mulfolder-project-[nitin@archlinux]-[0]-[8528]
[::] % |
```

## Scenario-Based Git Commands: Discarding and Unstaging Changes



A screenshot of a terminal window titled 'git-assignment/my-multifold...'. The terminal shows the following command sequence:

```
[~/git-assignment/my-multiproject] [nitin@archlinux] [128] [8522]
[::] % git remote set-url origin https://YeshwanthToluva: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@github.com/YeshwanthToluva/my-multi
[~/git-assignment/my-multiproject] [nitin@archlinux] [0] [8523]
[::] % git push -u origin main

Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (8/8), 486 bytes | 243.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/YeshwanthToluva/my-multiproject.git
 * [new branch]      main    -> main
branch 'main' set up to track 'origin/main'.
[~/git-assignment/my-multiproject] [nitin@archlinux] [0] [8524]
[::] % |
```

1. You've cloned a repository and made some changes to a local branch. Now you want to push these changes to the remote repository, but you're getting an error saying "rejected - non-fast-forward." How would you resolve this?

This error occurs when the remote branch has changes that your local branch doesn't. To resolve it:

**git pull --rebase origin <branch-name>**

This rebases your local changes on top of the latest remote changes. After resolving any conflicts, push your changes:

2. You've been working on a feature branch, and now you need to push it to the remote repository. However, the remote repository already has a main branch. How do you push your feature branch without affecting the main branch?

You can push your feature branch independently:

**git push origin feature/feat-1**

This creates a new remote branch and does not affect the main branch.

3. You cloned a remote repository, but after a while, the repository's structure changed and new branches were added. How would you keep your local repository updated with the latest changes from the remote repository?

Use the following commands:

**git fetch origin**

This updates your local copy with all branches and changes from the remote. You can then check out new branches using:

**git checkout branch-name**

4. A colleague has pushed some changes to the main branch, but you have local changes in the same branch. You want to pull their changes, but you want to avoid merge conflicts. What steps would you take?

Use rebase to integrate their changes on top of your work:

**git stash # Temporarily store your changes**

**git pull --rebase origin main**

**git stash pop # Apply your changes on top**

This reduces the chance of conflicts and keeps history clean.

5. You accidentally pushed a sensitive file (e.g., API keys) to the remote repository. How would you fix this situation?

Steps to remove the sensitive data:

**Remove the file and commit:**

**git rm --cached path/to/file**

**git commit -m "Remove sensitive file"**

**git push origin main**

If the secret is in history, use git filter-branch or BFG Repo-Cleaner to rewrite history:

**git filter-branch --force --index-filter \**

**"git rm --cached --ignore-unmatch path/to/file" \**

```
--prune-empty --tag-name-filter cat -- --all
```

Force push and rotate the secret.

6. You're working on a feature branch, and your manager requests that you integrate the latest changes from main into your feature branch. What steps would you take?

Use rebase or merge:

Rebase:

```
git checkout feature/your-feature
```

```
git fetch origin
```

```
git rebase origin/main
```

7. You cloned a remote repository, but later you find that you need to push your changes to a different remote repository. How do you configure your local repository to push to this new remote?

Then push your changes:

```
git push origin branch-name
```

8. After running git pull, you notice that your local branch is behind the remote branch. How would you proceed to bring your local branch up to date without losing your local changes?

Use stash or rebase:

```
git stash
```

```
git pull --rebase origin branch-name
```

```
git stash pop
```

This ensures a clean rebase and retains your changes.

9. You're working on a project with multiple collaborators, and you notice that your local changes conflict with changes that have been pushed by others. How would you resolve the conflicts?

Pull the latest changes:

**git pull origin branch-name**

Git will highlight conflicts. Open the files, manually resolve the <<<<<, =====, and >>>>> markers.

Mark as resolved and commit:

**git add .**

**git commit**

10. You've pushed a feature branch to a remote repository, but now you need to delete the branch from the remote. How would you do that?

Use the following command:

**git push origin --delete feature/branch-name**

This will remove the branch from the remote repository.

### **3. Collaborative coding using git**

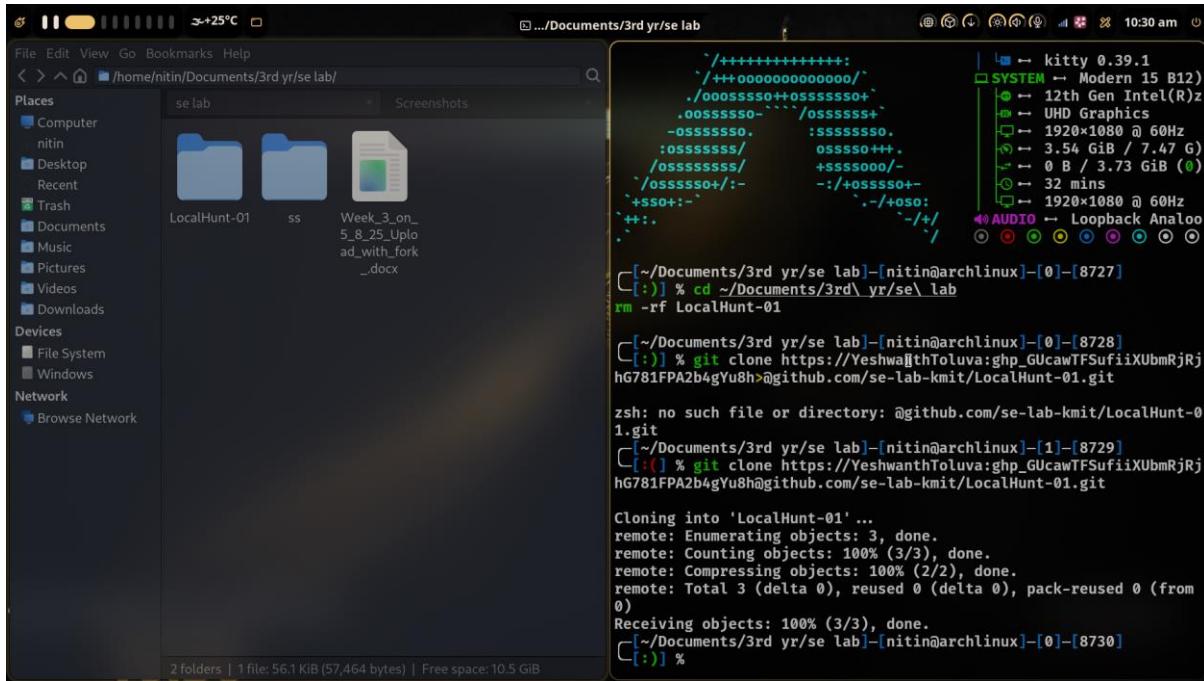
## GitHub Organization Members Page - se-lab-kmit Team Overview

The screenshot shows the GitHub Organization Members page for the 'se-lab-kmit' organization. The URL is [github.com/orgs/se-lab-kmit/people](https://github.com/orgs/se-lab-kmit/people). The page displays three members: Edigirala Neksha, Varshith-666, and YeshwanthToluva. Each member's profile picture, name, role (Member or Owner), and team count (0 teams) are visible. A sidebar on the left shows organization permissions: Members (3) and Security Managers. A search bar at the top allows users to find specific members. A banner at the top right encourages users to enable two-factor authentication.

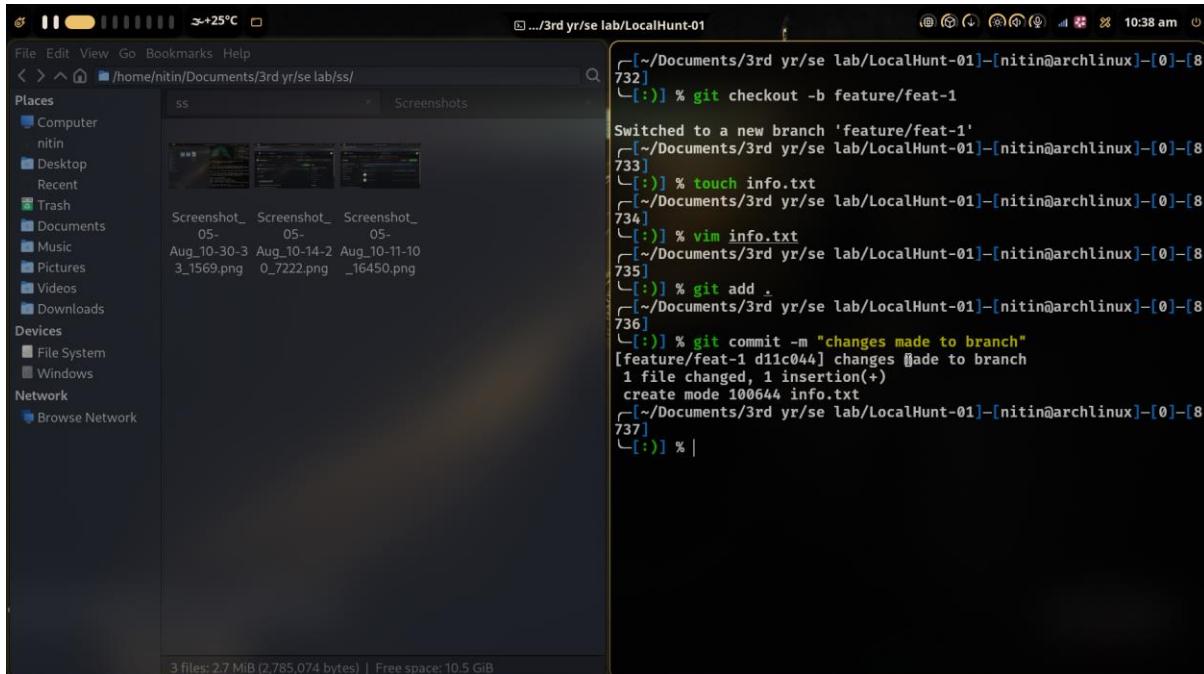
## GitHub Repository Overview - LocalHunt-01 Private Repository

The screenshot shows the GitHub Repository Overview for the 'LocalHunt-01' repository, which is private. The URL is [github.com/se-lab-kmit/LocalHunt-01](https://github.com/se-lab-kmit/LocalHunt-01). The repository has 1 branch and 0 tags. The most recent commit was made by 'YeshwanthToluva' with the message 'Initial commit'. The repository has 0 forks, 0 stars, and 0 releases. The README file contains the text 'TEsting the private repo of the organization'. The repository is part of the 'se-lab-kmit' organization.

## Terminal Git Clone Operations - LocalHunt-01 Repository Setup



Git Branch Operations - Feature Branch Creation and File Management



Git Push and Pull Request Creation - Feature Branch Workflow

```

733] [::] % touch info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
734] [::] % vim info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
735] [::] % git add .
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
736] [::] % git commit -m "changes made to branch"
[feature/feat-1 d11c044] changes made to branch
1 file changed, 1 insertion(+)
create mode 100644 info.txt
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
737] [::] % git push origin feature/feat-1

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature/feat-1' on GitHub by vis
iting:
remote: https://github.com/se-lab-kmit/LocalHunt-01/pull/new/f
eature/feat-1
remote:
To https://github.com/se-lab-kmit/LocalHunt-01.git
 * [new branch]      feature/feat-1 -> feature/feat-1
[~/Documents/3rd yr/se lab/LocalHunt-01-[nitin@archlinux]-[0]-[8
738] [::] %

```

"Screenshot\_05-Aug\_10-38-32\_17782.png" | 1.3 MiB (1,318,318 bytes)

## GitHub Repository Fork - simple-repo-se Overview and Setup

The screenshot shows a GitHub repository page for 'simple-repo-se'. The repository was forked from 'imagec/simple-repo'. It has 0 stars, 0 forks, and 0 releases. The 'About' section describes it as 'a simple repo for assignment'. The 'Code' tab is selected, showing the master branch with 1 branch and 0 tags. The branch is up-to-date with 'imagec/simple-repo:master'. The repository contains three files: README.RD, repo, and repo\_utils.py, all initialized 9 years ago.

## Git Commit and Status - README.RD File Modifications in Feature Branch

```

File Edit View Go Bookmarks Help
< > ^ _ /home/nitin/Documents/3rd yr/se lab/
Places Computer nitin Desktop Recent Trash Documents Music Pictures Videos Downloads Devices File System Windows Network Browse Network
se lab Screenshots
[8763] [:] % git add README.RD
git status
git commit -m "Added my name to README.RD"

On branch feature/feat-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.RD

[feature/feat-1 59b935a] Added my name to README.RD
1 file changed, 1 insertion(+), 1 deletion(-)
[~/Documents/3rd yr/se lab/simple-repo-se]-[nitin@archlinux]-[0]-
[8764] [:] %

```

3 folders | 1 file; 56.1 KiB (57,464 bytes) | Free space: 10.4 GiB

## Git Push to Forked Repository - Feature Branch Upload and Pull Request Creation

```

File Edit View Go Bookmarks Help
< > ^ _ /home/nitin/Documents/3rd yr/se lab/
Places Computer nitin Desktop Recent Trash Documents Music Pictures Videos Downloads Devices File System Windows Network Browse Network
se lab Screenshots
[8764] [:] % git status
git commit -m "Added my name to README.RD"

On branch feature/feat-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.RD

[feature/feat-1 59b935a] Added my name to README.RD
1 file changed, 1 insertion(+), 1 deletion(-)
[~/Documents/3rd yr/se lab/simple-repo-se]-[nitin@archlinux]-[0]-
[8764] [:] %

```

```

git status
git commit -m "Added my name to README.RD"

On branch feature/feat-1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.RD

[feature/feat-1 59b935a] Added my name to README.RD
1 file changed, 1 insertion(+), 1 deletion(-)
[~/Documents/3rd yr/se lab/simple-repo-se]-[nitin@archlinux]-[0]-
[8764] [:] %

```

```

git push origin feature/feat-1

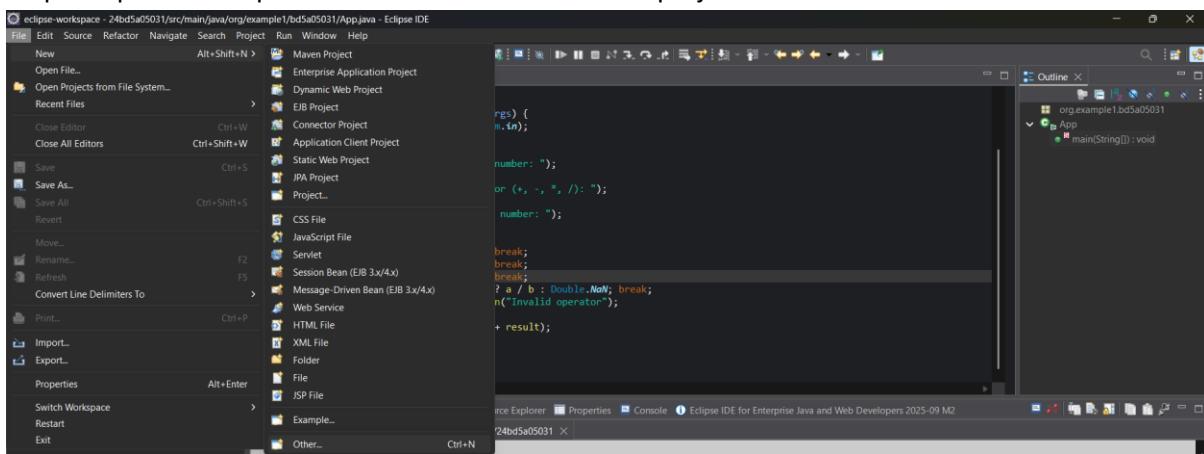
Username for 'https://github.com': YeshwanthToluva
Password for 'https://YeshwanthToluva@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 383 bytes | 383.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object
.
remote:
remote: Create a pull request for 'feature/feat-1' on GitHub by vis
iting:
remote:   https://github.com/YeshwanthToluva/simple-repo-se/pull
/nw/feature/feat-1
remote:
To https://github.com/YeshwanthToluva/simple-repo-se.git
 * [new branch]      feature/feat-1 -> feature/feat-1
[~/Documents/3rd yr/se lab/simple-repo-se]-[nitin@archlinux]-[0]-
[8765] [:] %

```

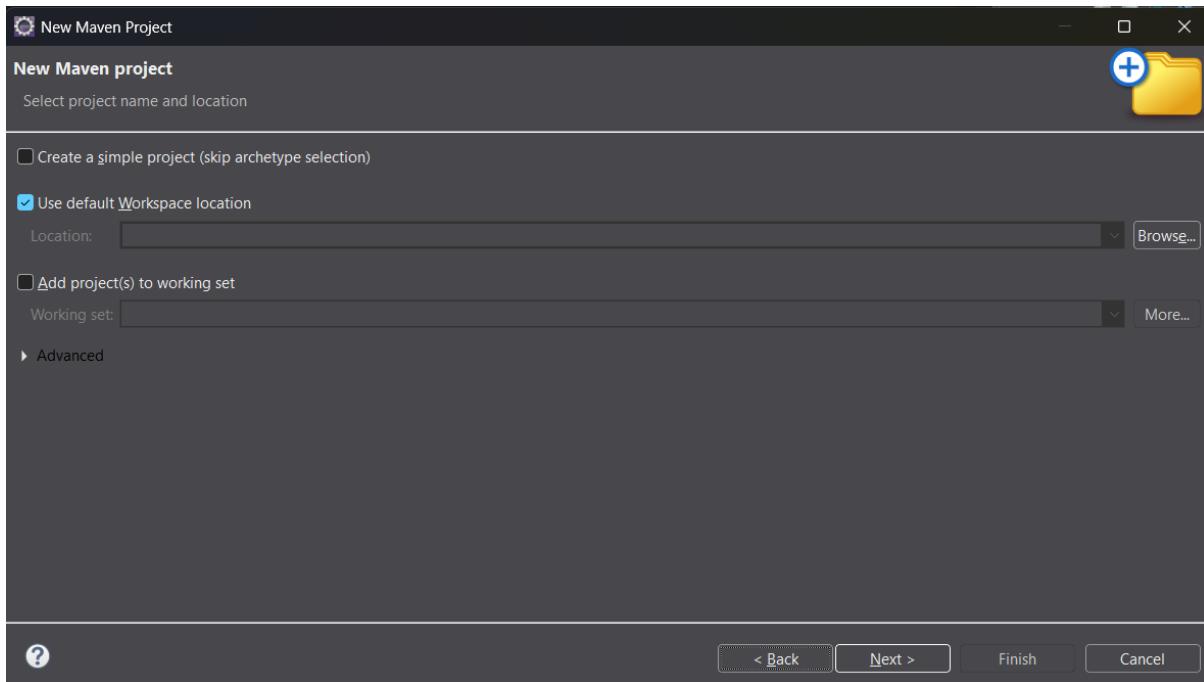
3 folders | 1 file; 56.1 KiB (57,464 bytes) | Free space: 10.4 GiB

## 4. Build and package Java and Web applications using Maven

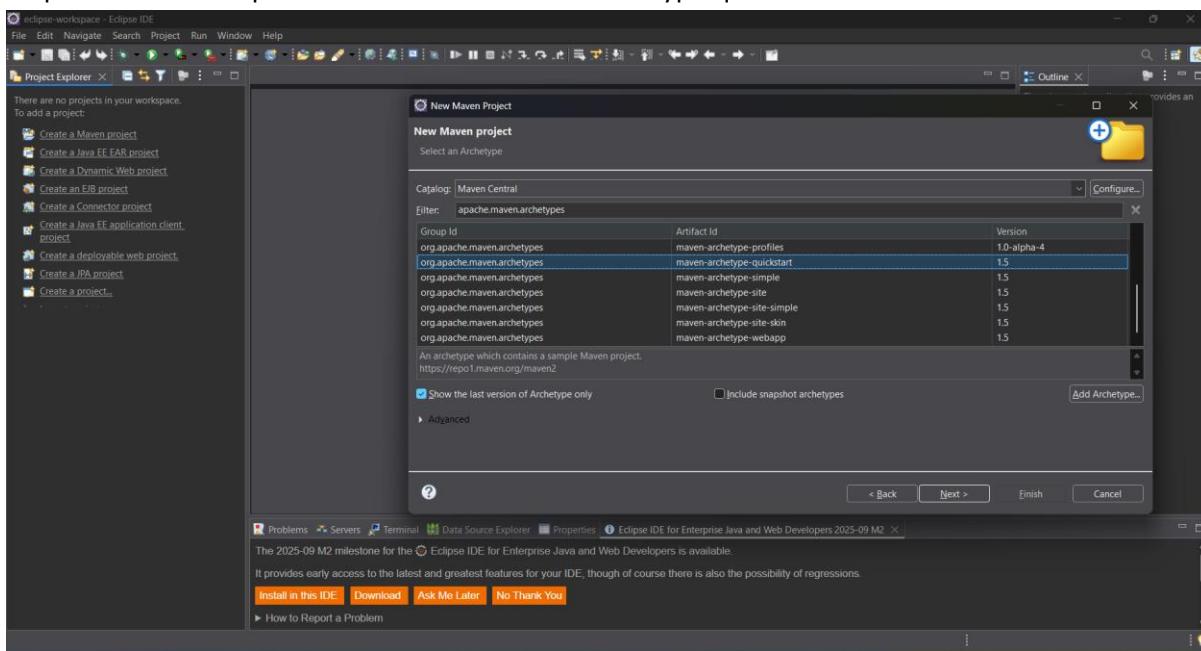
## Step-1: Open the eclipse and click on file>new>Maven project



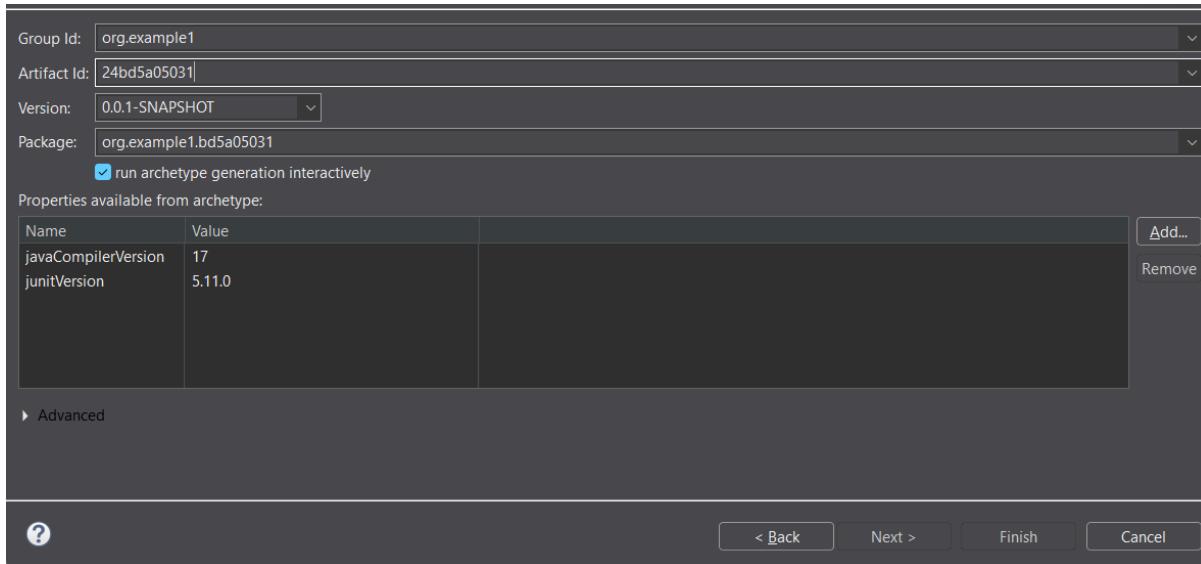
## Step-2: select the default workspace and click on next



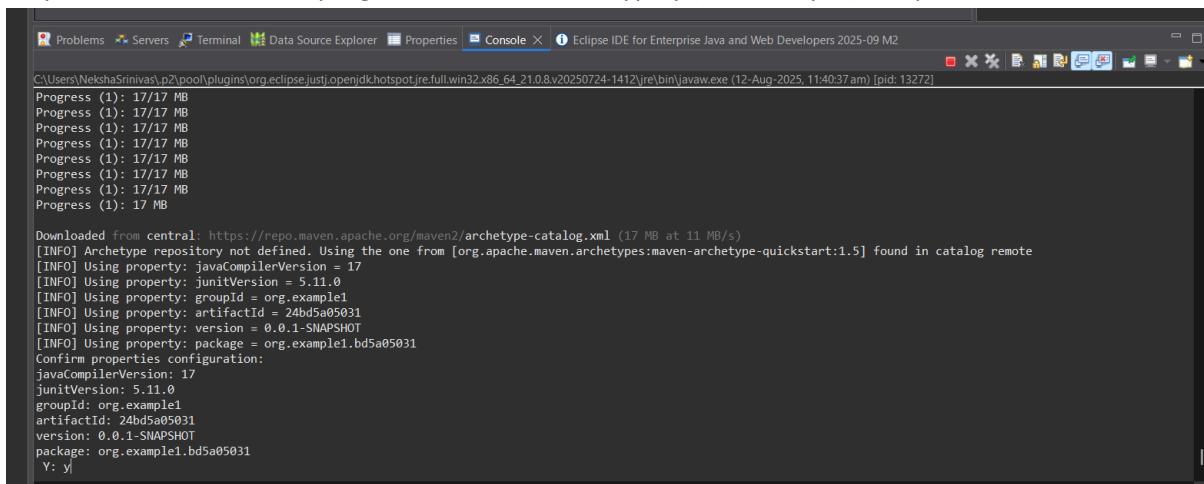
### Step-3: in the filter option select the one maven-archetype-quickstart



### Step-4: give the Group Id and Artifact Id and click on next



Step-5: In the console the progress will be showed type y (refers to yes) and press enter

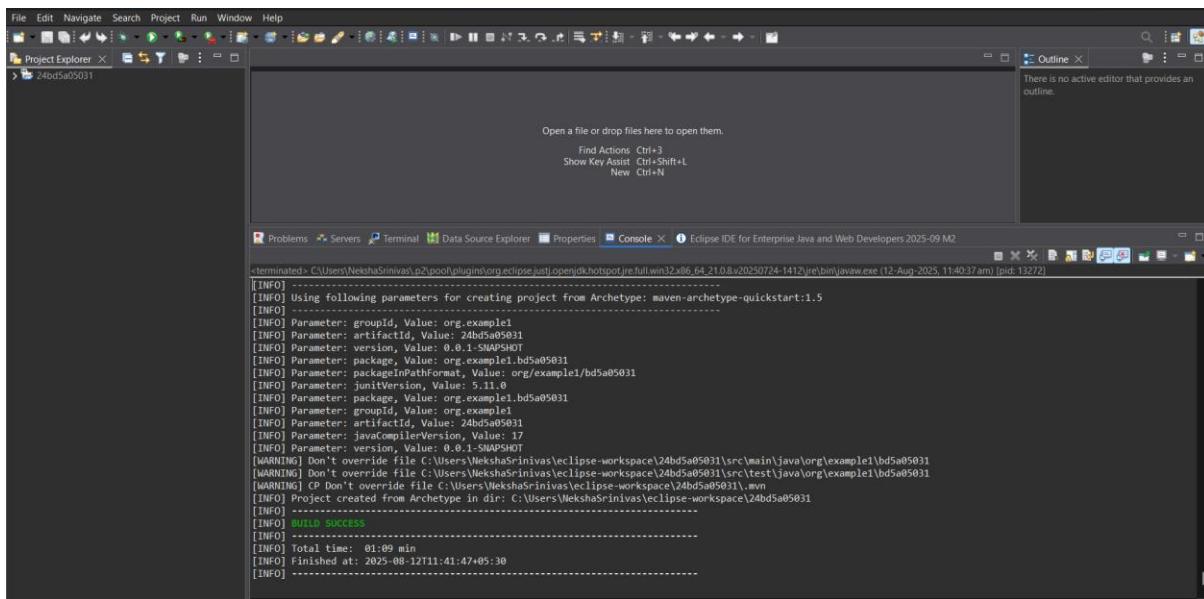


The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 11 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in catalog remote
[INFO] Using property: javaCompilerVersion = 17
[INFO] Using property: junitVersion = 5.11.0
[INFO] Using property: groupId = org.example1
[INFO] Using property: artifactId = 24bd5a05031
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = org.example1.bd5a05031
Confirm properties configuration:
javaCompilerVersion: 17
junitVersion: 5.11.0
groupId: org.example1
artifactId: 24bd5a05031
version: 0.0.1-SNAPSHOT
package: org.example1.bd5a05031
Y: y|
```

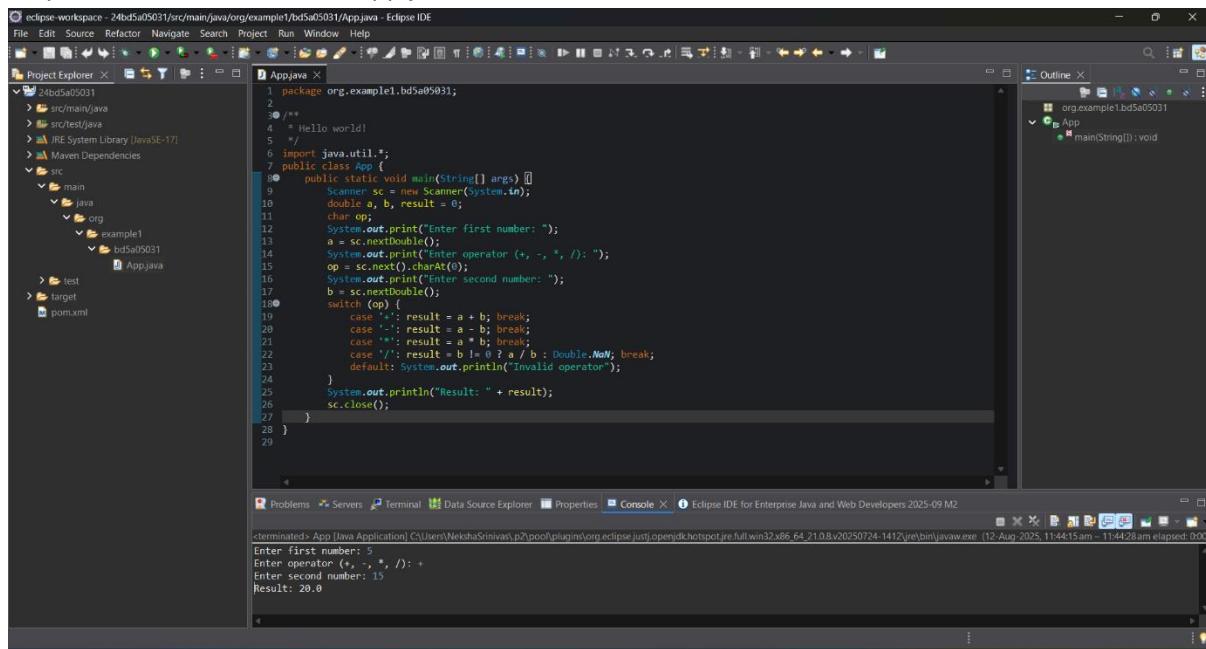
Step-6: BUILD SUCCESS will be shown



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
[terminated - C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.5
[INFO] -----
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: packageInPathFormat, Value: org/example1/bd5a05031
[INFO] Parameter: junitVersion, Value: 5.11.0
[INFO] Parameter: groupId, Value: org.example1.bd5a05031
[INFO] Parameter: artifactId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: javaCompilerVersion, Value: 17
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\main\java\org\example1\bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\test\java\org\example1\bd5a05031
[WARNING] CP Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\.mvn
[INFO] Project created from Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:09 min
[INFO] Finished at: 2025-08-12T11:41:47+05:30
[INFO] -----
```

## Step-6: write the code in the App.java file



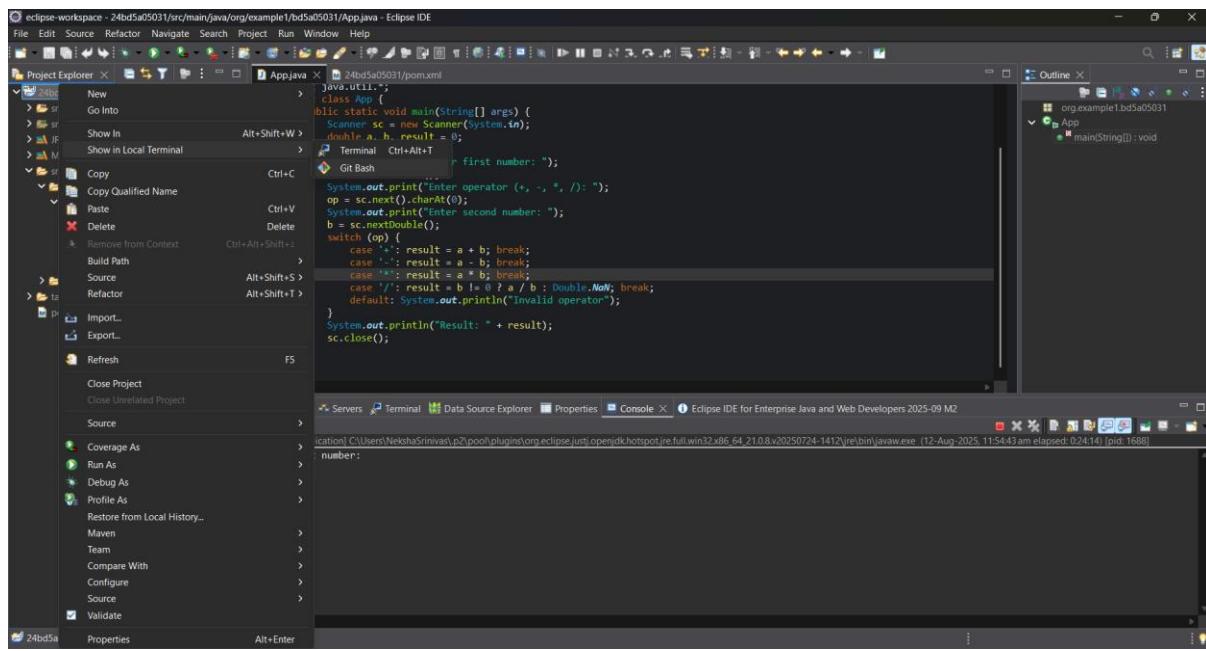
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure: 24bd5a05031, src/main/java, src/test/java, IRE System Library [JavaSE-17], Maven Dependencies, and src.
- App.java Content:**

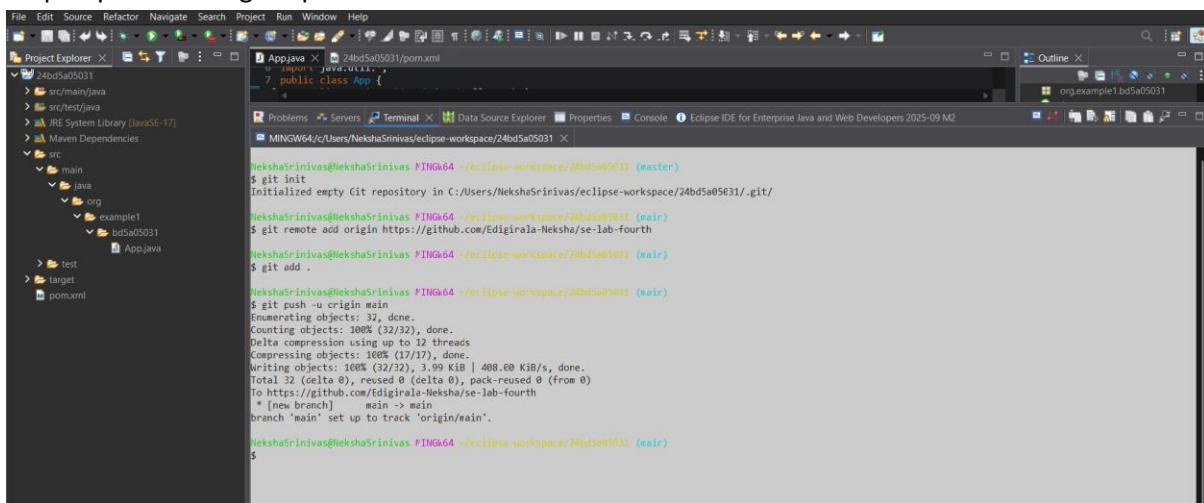
```
1 package org.example1.bd5a05031;
2
3 /**
4  * Hello world!
5  */
6 import java.util.*;
7 public class App {
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        double a, b, result = 0;
11        char op;
12        System.out.print("Enter first number: ");
13        a = sc.nextDouble();
14        System.out.print("Enter operator (+, -, *, /): ");
15        op = sc.next().charAt(0);
16        System.out.print("Enter second number: ");
17        b = sc.nextDouble();
18        switch (op) {
19            case '+': result = a + b; break;
20            case '-': result = a - b; break;
21            case '*': result = a * b; break;
22            case '/': result = b != 0 ? a / b : Double.NaN; break;
23            default: System.out.println("Invalid operator");
24        }
25        System.out.println("Result: " + result);
26        sc.close();
27    }
28 }
```
- Console Output:**

```
Enter first number: 5
Enter operator (+, -, *, /): +
Enter second number: 15
Result: 20.0
```

## Step-7: right click on the root folder and select show in git bash



## Step-8: push to the git repo



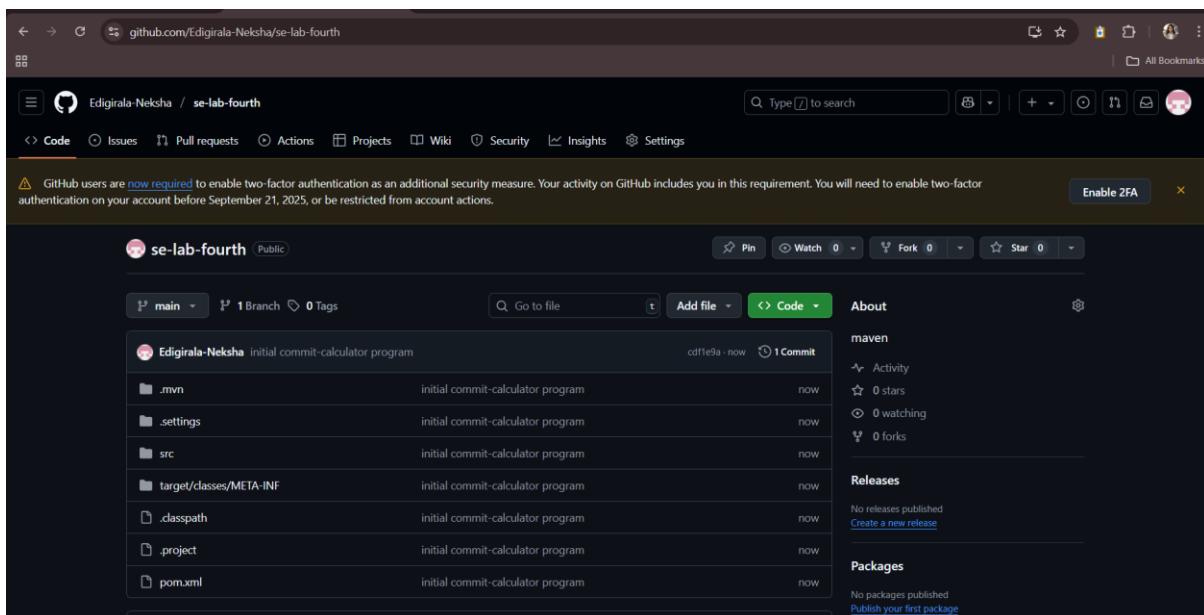
The screenshot shows the Eclipse IDE interface with the Project Explorer and Outline views. A terminal window is open in the bottom right corner, displaying the following command-line session:

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a05031/.git/
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git remote add origin https://github.com/Edigirala-Neksha/se-lab-fourth
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git add .

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a05031 (main)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 4 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (32/32), 3.09 KiB | 408.00 KiB/s, done.
Total 32 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Edigirala-Neksha/se-lab-fourth
 * [new branch]      main > main
branch 'main' set up to track 'origin/main'.
```

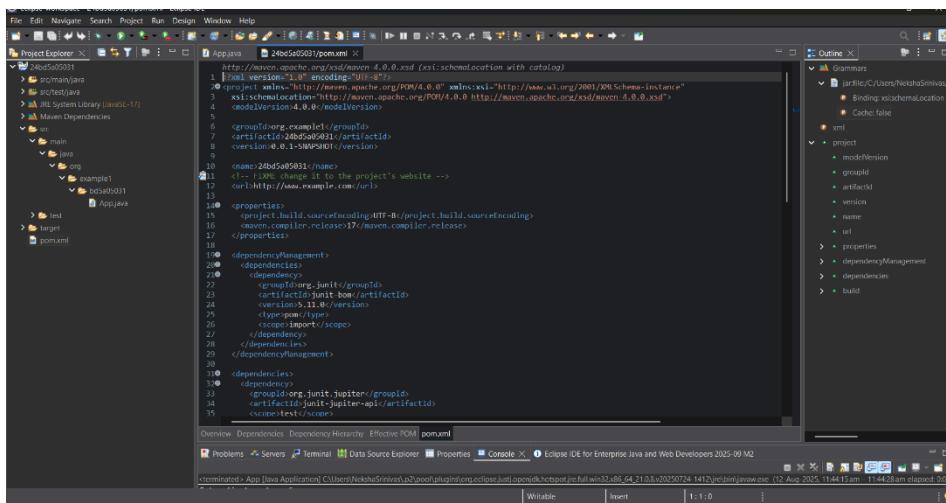
## Git repo:

Git repo link: <https://github.com/Edigirala-Neksha/se-lab-fourth>



pom.xml file:

Shows the structure-

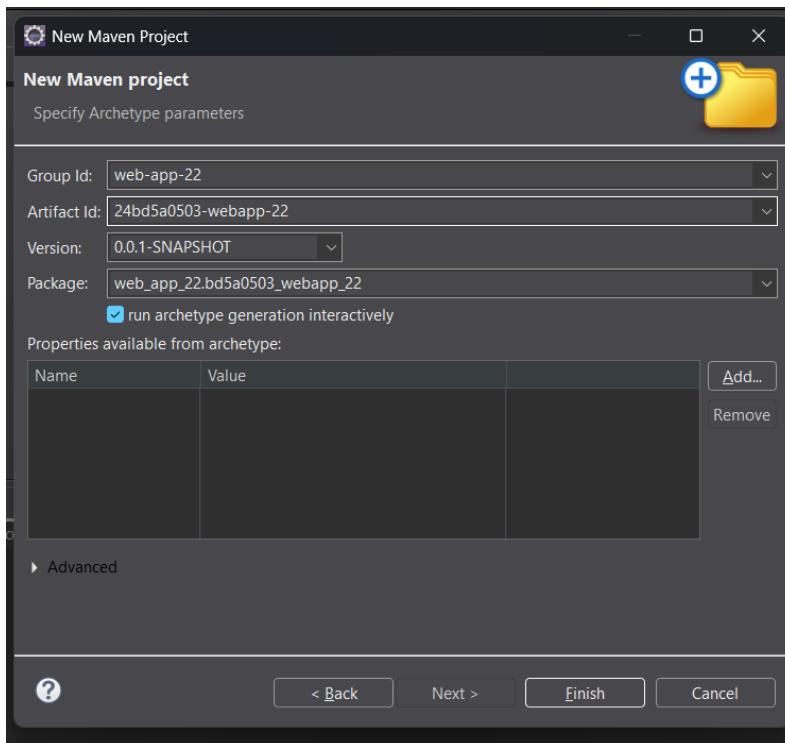


The screenshot shows the Eclipse IDE interface with the pom.xml file open. The Project Explorer view on the left displays the project structure with files like src/main/java, src/test/java, and pom.xml. The Outline view on the right shows the XML structure of the pom.xml file, including sections like `<groupId>`, `<artifactId>`, `<version>`, `<dependencies>`, and `<dependencyManagement>`. The code editor in the center contains the XML code for the pom.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>zbd5a0503</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>zbd5a0503</name>
  <url>http://www.example.com/url</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.release>17</maven.compiler.release>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>5.11</version>
        <scope>import</scope>
      </dependency>
      <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.11.1</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>5.11.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Creating maven-web project:

Step 1: Create a new maven project and give the details



## Step 2: Click y to continue the creation of project

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_21.0.8v20250724-1412\jre\bin\javaw.exe (02-Sept-2025, 7:19:56 pm) [pid: 13772]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 9.1 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in catalog remote
[INFO] Using property: groupId = web-app-22
[INFO] Using property: artifactId = 24bd5a0503-webapp-22
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = web_app_22.bd5a0503_webapp_22
Confirm properties configuration:
groupId: web-app-22
artifactId: 24bd5a0503-webapp-22
version: 0.0.1-SNAPSHOT
package: web_app_22.bd5a0503_webapp_22
Y: y
```

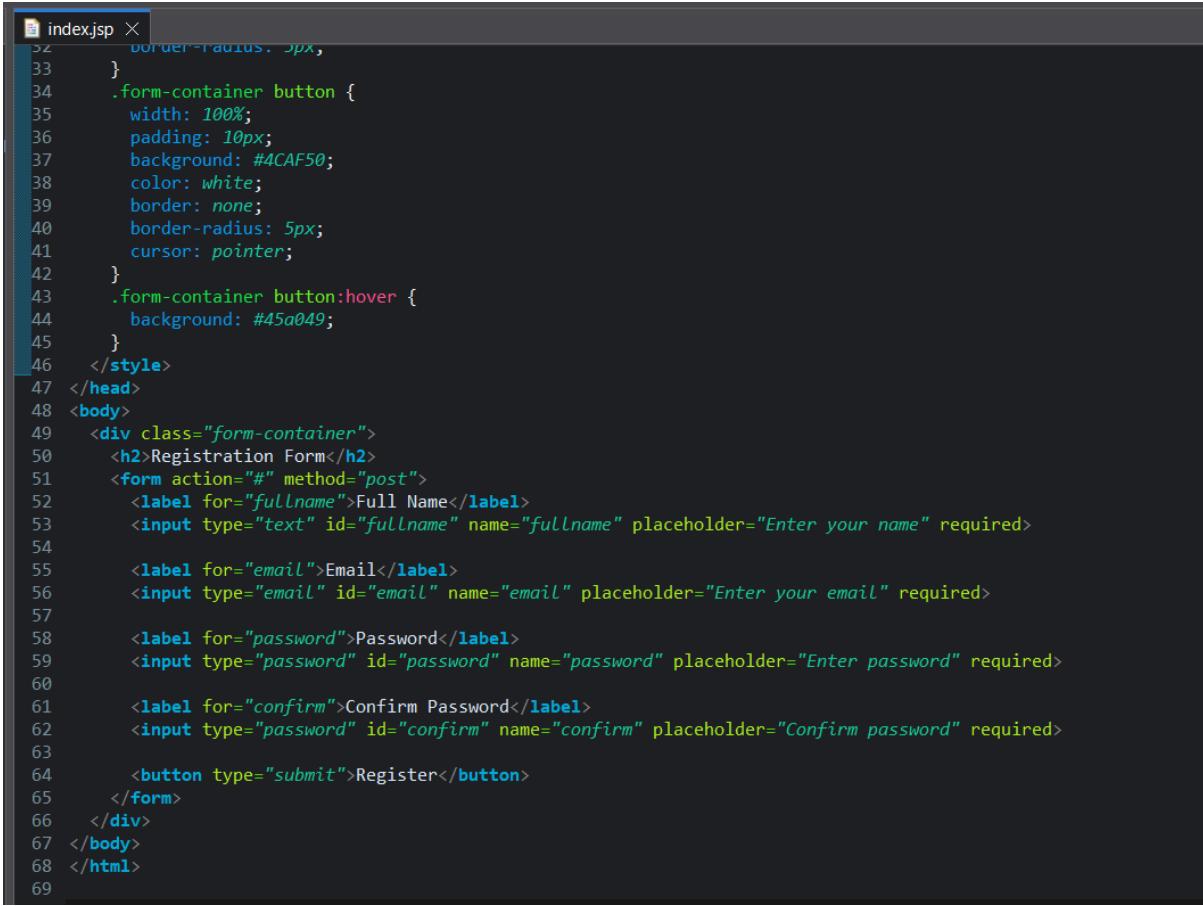
## Step 3: If the build is success it will show the message

```
package: web_app_22.bd5a0503_webapp_22
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\NekshaSrinivas\eclipse-workspace
[INFO] Parameter: package, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: groupId, Value: web-app-22
[INFO] Parameter: artifactId, Value: 24bd5a0503-webapp-22
[INFO] Parameter: packageName, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a0503-webapp-22
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 43.500 s
[INFO] Finished at: 2025-09-02T19:20:41+05:30
[INFO] -----
```

## Step 4: write the html code for the web page:

```
index.jsp X
⑥ <html>
  <body>
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Registration Form</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #f9f9f9;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
      }
      .form-container {
        background: #fff;
        padding: 20px 30px;
        border-radius: 10px;
        box-shadow: 0 4px 10px rgba(0,0,0,0.1);
        width: 300px;
      }
      .form-container h2 {
        text-align: center;
        margin-bottom: 20px;
      }
      .form-container input {
        width: 100%;
        padding: 10px;
        margin: 8px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
      }
      .form-container button {
        width: 100%;
        padding: 10px;
        background: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
      }
    </style>
  </head>
  <body>
    <div class="form-container">
      <h2>Registration Form</h2>
      <form>
        <div>
          <label>Name:</label>
          <input type="text" placeholder="Enter Name" required>
        </div>
        <div>
          <label>Email:</label>
          <input type="email" placeholder="Enter Email" required>
        </div>
        <div>
          <label>Password:</label>
          <input type="password" placeholder="Enter Password" required>
        </div>
        <div>
          <label>Confirm Password:</label>
          <input type="password" placeholder="Enter Confirm Password" required>
        </div>
        <div>
          <input type="checkbox" checked=""> I agree to the terms and conditions
        </div>
        <div>
          <button type="submit">Register</button>
        </div>
      </form>
    </div>
  </body>
</html>
```

## Web-page:



```
index.jsp X
  border-radius: 5px;
}
.form-container button {
  width: 100%;
  padding: 10px;
  background: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.form-container button:hover {
  background: #45a049;
}
</style>
</head>
<body>
<div class="form-container">
  <h2>Registration Form</h2>
  <form action="#" method="post">
    <label for="fullname">Full Name</label>
    <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>

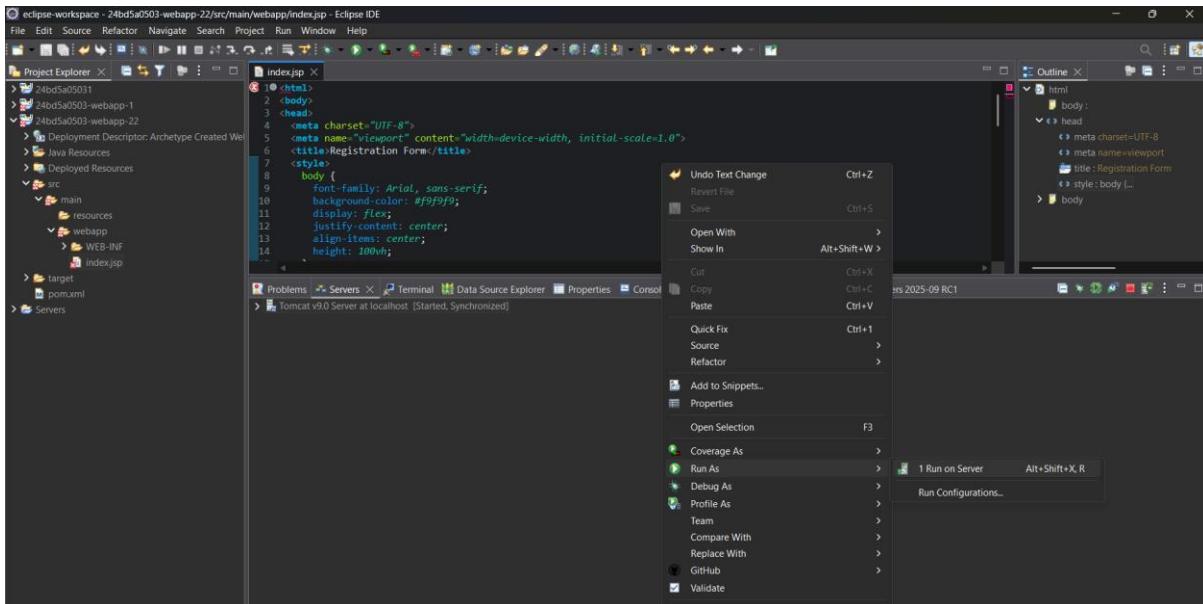
    <label for="email">Email</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required>

    <label for="password">Password</label>
    <input type="password" id="password" name="password" placeholder="Enter password" required>

    <label for="confirm">Confirm Password</label>
    <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>

    <button type="submit">Register</button>
  </form>
</div>
</body>
</html>
```

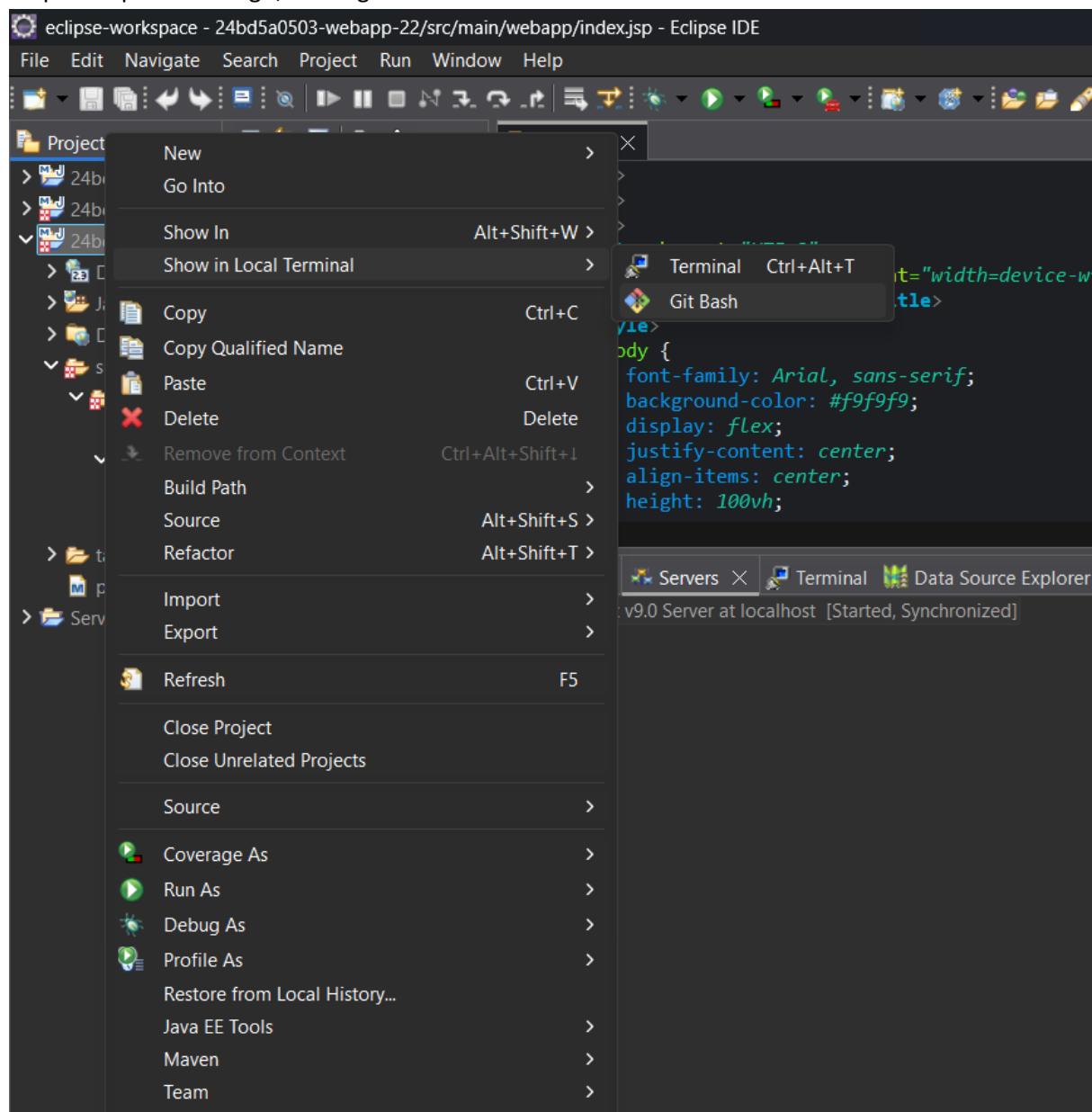
## Step 5: Select run on server



Step 6: It will show the following output:

A screenshot of a web browser window displaying a registration form. The browser's address bar shows the URL: `localhost:8080/24bd5a0503-webapp-22/index.jsp`. The main content area is a white card with a rounded border, titled "Registration Form". The form contains five input fields: "Full Name" (placeholder: "Enter your name"), "Email" (placeholder: "Enter your email"), "Password" (placeholder: "Enter password"), and "Confirm Password" (placeholder: "Confirm password"). Below these fields is a green rectangular button labeled "Register".

Step 7: To push it into git, select git bash from show in local terminal

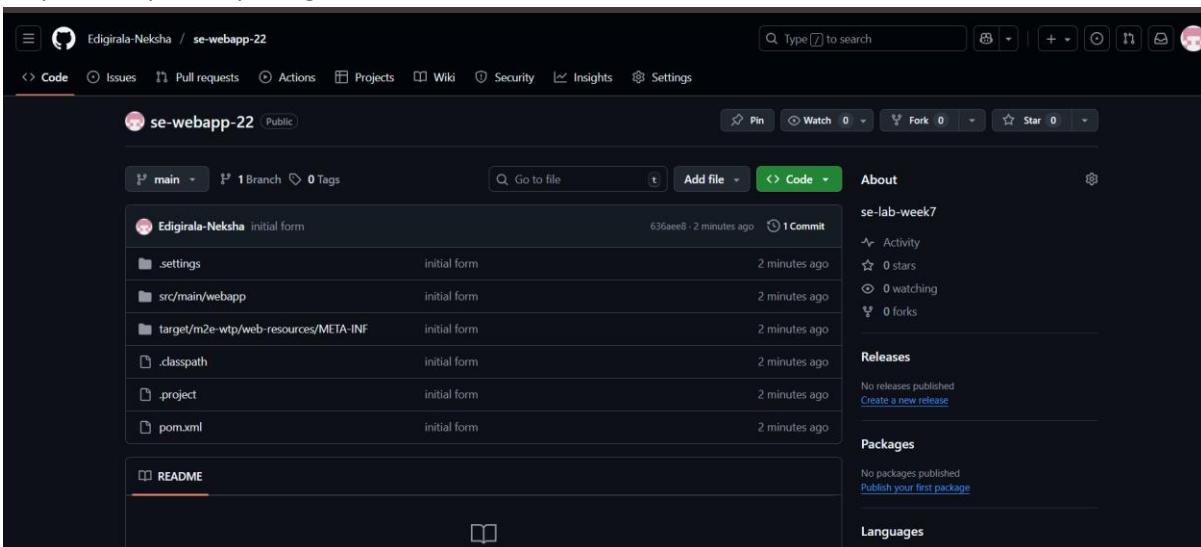


## Step 8: use the command of git to push the maven web project

```
MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05... MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05... MINGW64:/c/Users/NekshaSrinivas/eclipse-workspace/24bd5a05...  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (master)  
$ git init  
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22/.git/  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git add .  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git commit -m "initial form"  
[main (root-commit) 636aee8] initial form  
16 files changed, 254 insertions(+)  
create mode 100644 .classpath  
create mode 100644 .project  
create mode 100644 .settings/.jsdtscope  
create mode 100644 .settings/org.eclipse.jdt.core.prefs  
create mode 100644 .settings/org.eclipse.m2e.core.prefs  
create mode 100644 .settings/org.eclipse.wst.commonn.component  
create mode 100644 .settings/crg.eclipse.wst.commonn.project.facet.core.xml  
create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.container  
create mode 100644 .settings/crg.eclipse.wst.jsdt.ui.superType.name  
create mode 100644 .settings/crg.eclipse.wst.validation.prefs  
create mode 100644 pom.xml  
create mode 100644 src/main/webapp/WEB-INF/web.xml
```

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git branch  
* main  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$ git push origin main  
Enumerating objects: 29, done.  
Counting objects: 100% (29/29), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (18/18), done.  
Writing objects: 100% (29/29), 4.43 KiB | 283.00 KiB/s, done.  
Total 29 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/Edigirala-Neksha/se-webapp-22.git  
 * [new branch]      main -> main  
  
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)  
$
```

## Step 9: verify the repo in git hub



The screenshot shows a GitHub repository page for 'se-webapp-22'. The repository is public and has one branch named 'main'. The 'About' section shows the repository was created by 'Edigirala-Neksha' and has 1 commit. The 'Releases' section shows no releases published. The 'Languages' section shows Java as the primary language.

## **5. Docker CLI commands**

### **Installing Docker and Setting up Nginx**

#### **Introduction**

**Docker** is a platform that allows us to run applications inside lightweight containers. Containers are isolated environments that include everything needed to run an application. This makes it easy to set up and deploy software without worrying about dependencies or configurations on the host system.

In this task, we used Docker to run an **Ubuntu container**, install **nginx** inside it, and serve a customized homepage

#### **Step 1: Pulling the Ubuntu Image**

First, we pulled the latest Ubuntu image from Docker Hub.

```
PS C:\Users\NekshaSrinivas> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\NekshaSrinivas> cd SE-1
PS C:\Users\NekshaSrinivas\SE-1> docker --version
Docker version 28.3.2, build 578ccf6
PS C:\Users\NekshaSrinivas\SE-1> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
b71466b94f26: Pull complete
Digest: sha256:7c06e91f61fa88c08cc74f7e1b7c69ae24910d745357e0dfe1d2c0322aaf2
0f9
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

#### **Step 2: Running the Container**

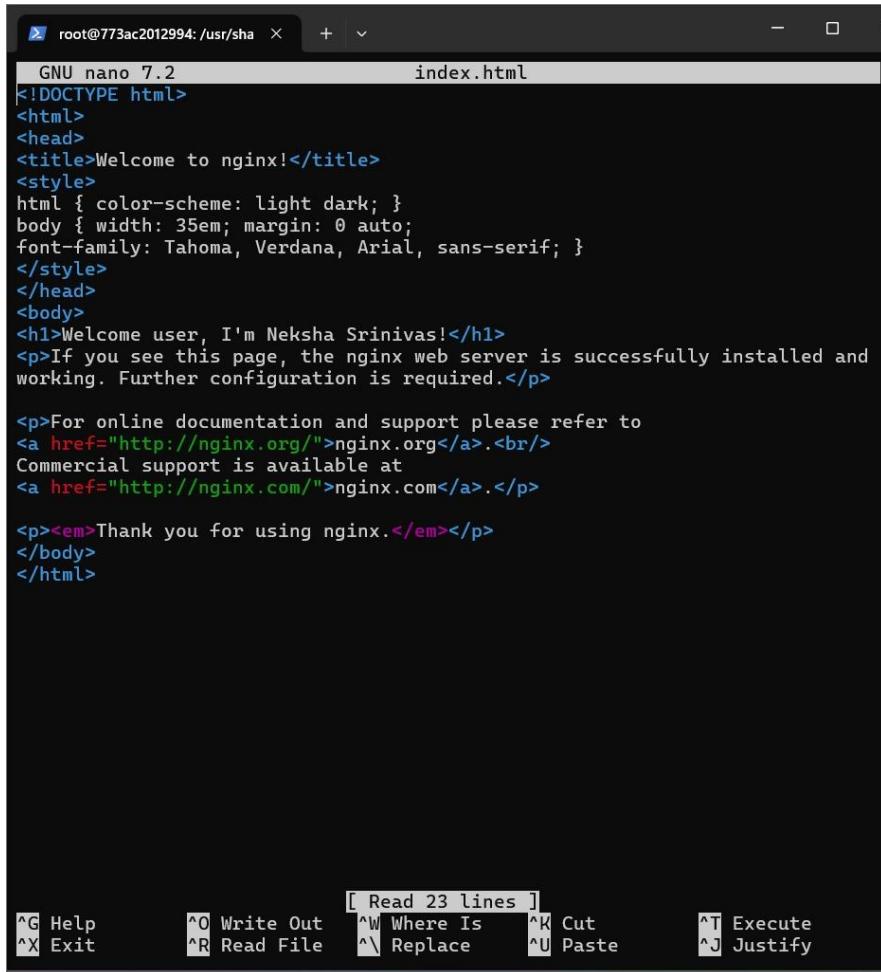
We created and started a new container named **myubuntu**, mapping port **3000** on the host to port **80** inside the container.

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -it -p 9090:80 --name myubuntu1
ubuntu:latest
root@773ac2012994:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1135 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1355 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2047 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [23.0 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
```

Step 3: Installing Nginx and redirecting to index.html page to edit the content

```
Processing triggers for libc-bin (2.39-0ubuntu8.5) ...
root@773ac2012994:/# ls
bin          dev    lib     mnt   root  sbin usr-is-merged  tmp
bin usr-is-merged etc    lib64  opt    run    srv           usr
boot        home   media  proc   sbin   sys           var
root@773ac2012994:/# cd usr
root@773ac2012994:/usr# ls
bin games include lib lib64 libexec local sbin share src
root@773ac2012994:/usr# cd share
root@773ac2012994:/usr/share# ls
apport         gcc      pam
base-files     gdb      pam-configs
base-passwd    info     perl5
bash-completion info.dir pixmaps
bug            keyrings polkit-1
common-licenses libc-bin profile
debconf        libgcrypt20 profile.md5sums
debianutils    lintian  sensible-utils
dict           locale   staff-group-for/usr-local
doc            man      tabset
doc-base       menu    terminfo
dot.bashrc     misc    util-linux
dot.profile    motd    vim
dot.profile.md5sums networks
dpkg           nginx
root@773ac2012994:/usr/share# cd nginx
root@773ac2012994:/usr/share/nginx# ls
html modules
root@773ac2012994:/usr/share/nginx# cd html
root@773ac2012994:/usr/share/nginx/html# ls
index.html
root@773ac2012994:/usr/share/nginx/html# nano index.html
bash: nano: command not found
root@773ac2012994:/usr/share/nginx/html# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
```

Step 4: navigate to index.html using command –“nano index.html” Changed the content of h1 tag

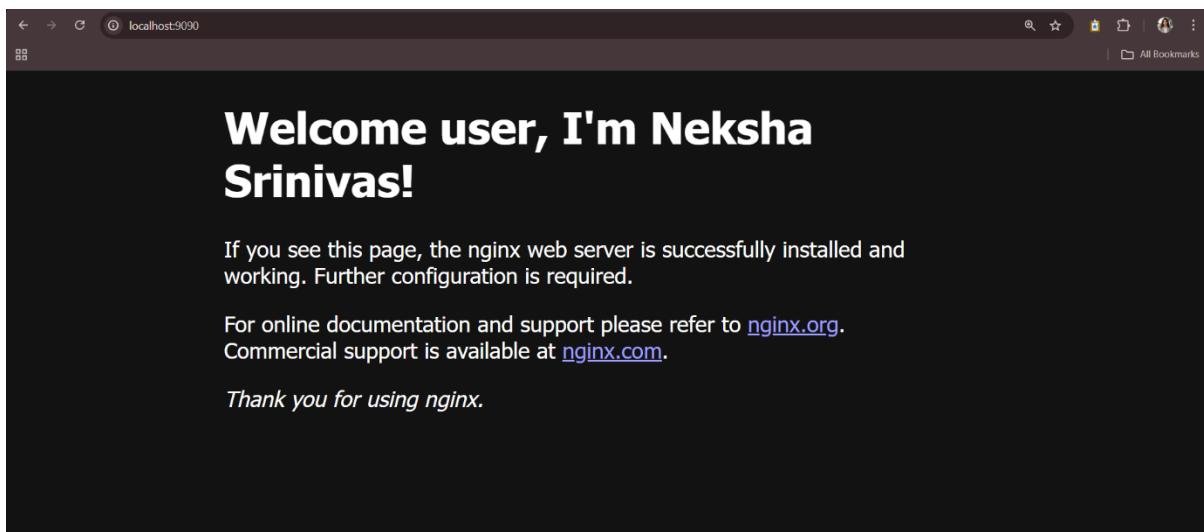


```
GNU nano 7.2                               index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome user, I'm Neksha Srinivas!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

[ Read 23 lines ]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

Step 5: Viewing the page from local host



## 6. Docker

### DOCKER IMAGE CREATION:

Image can be created in two ways:

1. Using Docker commit
2. Using docker file

Step 1: Created a new container of ubuntu so image can be created on that container

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -it -p 9090:80 --name ubuntu-cont-1 ubuntu:latest
root@74098c332e58:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1137 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2066 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1363 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [23.0 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packa
```

Step 2: Using commit the image is being created:

```
PS C:\Users\NekshaSrinivas\SE-1> docker commit ubuntu-cont-1 img-commit-1
sha256:153126502820131f25f36cc59f7c4557275621bcd7a54b48c8ffd4409685efd
PS C:\Users\NekshaSrinivas\SE-1> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
img-commit-1    latest    153126502820    7 seconds ago   326MB
mynginx         latest    de77ca8d52cb    30 hours ago   279MB
mypythonapp     latest    8a39b6d82115    30 hours ago   1.63GB
nginx           latest    33e0bbc7ca9e    12 days ago    279MB
ubuntu          latest    7c06e91f61fa    3 weeks ago    117MB
PS C:\Users\NekshaSrinivas\SE-1> docker run -it img-commit-1
root@909ab066a51f:/# git --version
git version 2.43.0
root@909ab066a51f:/# docker tag img-commit-1 nekshasrinivas/img-commit-1
bash: docker: command not found
root@909ab066a51f:/# exit
exit
PS C:\Users\NekshaSrinivas\SE-1> docker tag img-commit-1 nekshasrinivas/img-commmit-1
PS C:\Users\NekshaSrinivas\SE-1> docker push nekshasrinivas/img-commmit-1
Using default tag: latest
The push refers to repository [docker.io/nekshasrinivas/img-commmit-1]
4024494ad21b: Pushed
b71466b94f26: Mounted from library/ubuntu
latest: digest: sha256:153126502820131f25f36cc59f7c4557275621bcd7a54b48c8ffd4409685efd size: 751
```

### Step 3: Image creation using docker file

```
PS C:\Users\NekshaSrinivas\SE-1> mkdir image-creation

Directory: C:\Users\NekshaSrinivas\SE-1

Mode          LastWriteTime      Length Name
----          -----          ---- 
d----        26-08-2025       18:26      image-creation

PS C:\Users\NekshaSrinivas\SE-1> ls

Directory: C:\Users\NekshaSrinivas\SE-1

Mode          LastWriteTime      Length Name
----          -----          ---- 
d----        26-08-2025       18:26      image-creation
d----        25-08-2025       12:25      static_site
-a---        25-08-2025       12:15      36 app.py
-a---        25-08-2025       12:15      100 Dockerfile
-a---        23-08-2025       16:28      29739008 myapi.tar
-a---        05-08-2025       16:33      38 README.md

PS C:\Users\NekshaSrinivas\SE-1> cd image-creation
PS C:\Users\NekshaSrinivas\SE-1\image-creation> notepad Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\image-creation> ls

Directory: C:\Users\NekshaSrinivas\SE-1\image-creation

Mode          LastWriteTime      Length Name
----          -----          ---- 
-a---        26-08-2025       18:27      59 Dockerfile.txt

PS C:\Users\NekshaSrinivas\SE-1\image-creation> ren Dockerfile.txt Dockerfile
```

### Step 4: after writing the content in docker file use the command docker build

```
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker build -t img-dockerfile-1 .
[+] Building 44.0s (7/7) FINISHED
          docker:desktop-linux
=> [internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 96B                          0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest 0.1s
=> [internal] load .dockerrcignore                         0.1s
=> => transferring context: 2B                           0.0s
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:7c06e91f61fa88c 0.1s
=> => resolve docker.io/library/ubuntu:latest@sha256:7c06e91f61fa88c 0.0s
=> [2/3] RUN apt-get update                                12.5s
=> [3/3] RUN apt-get install git -y                      24.4s
=> exporting to image                                     6.5s
=> => exporting layers                                    4.5s
=> => exporting manifest sha256:99d816a6b717e709d838937a995f24d0121e 0.0s
=> => exporting config sha256:f021a40f65d4b684b65cd403292af90ec68210 0.0s
=> => exporting attestation manifest sha256:04dc38ee96b84e155b083e5 0.1s
=> => exporting manifest list sha256:9868ecb2df510b52e539c55076bf63c 0.0s
=> => naming to docker.io/library/img-dockerfile-1:latest 0.0s
=> => unpacking to docker.io/library/img-dockerfile-1:latest 1.8s
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker run -it img-dockerfile-1
root@adfe97a50685:/# docker --version
bash: docker: command not found
root@adfe97a50685:/# git --version
git version 2.43.0
root@adfe97a50685:/# exit
exit
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker tag img-dockerfile-1 nekshasrinivas/img-dockerfile-1
PS C:\Users\NekshaSrinivas\SE-1\image-creation> docker push nekshasrinivas/img-dockerfile-1
Using default tag: latest
The push refers to repository [docker.io/nekshasrinivas/img-dockerfile-1]
6a5ccfd4b031: Pushed
edd67216c21: Pushed
b71466b94f26: Mounted from nekshasrinivas/img-commmit-1
004a734bd8b1: Pushed
latest: digest: sha256:9868ecb2df510b52e539c55076bf63ccae47b54ab67e29de352ddbc3cb33b109 size: 855
```

## Step 5: checking the images

```
PS C:\Users\NekshaSrinivas\SE-1> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
img-dockerfile-1   latest   9868ecb2df51  5 minutes ago  326MB
nekshasrinivas/img-dockerfile-1   latest   9868ecb2df51  5 minutes ago  326MB
img-commit-1        latest   153126502820  12 minutes ago  326MB
nekshasrinivas/img-commmit-1    latest   153126502820  12 minutes ago  326MB
mynginx             latest   de77ca8d52cb  30 hours ago   279MB
mypythonapp         latest   8a39b6d82115  30 hours ago   1.63GB
nginx               latest   33e0bbc7ca9e  12 days ago    279MB
ubuntu              latest   7c06e91f61fa  3 weeks ago    117MB
PS C:\Users\NekshaSrinivas\SE-1> |
```

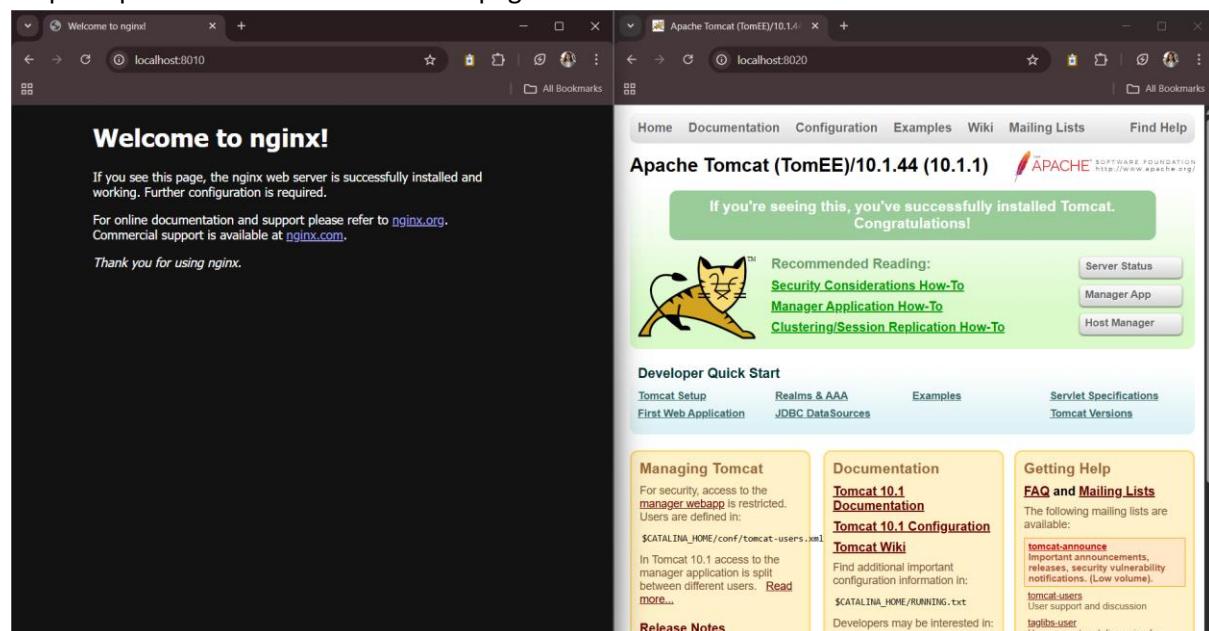
## DOCKER COMPOSE FILE:

Docker Compose is a tool used to define and run multi-container Docker applications. It allows you to define services, networks, and volumes that your application needs, all in a single file. This makes it easier to manage complex applications that require multiple containers (e.g., a web server and a database).

## Step 1: Running two servers at the same time on different ports

```
PS C:\Users\NekshaSrinivas\SE-1> docker run -d -p 8010:80 nginx
2ea4a201f197b93276310a7d23f2a46060ba9c7387f869e8a2a804931b66b2d9
PS C:\Users\NekshaSrinivas\SE-1> docker run -d -p 8020:8080 tomee
3a524036f6b212843be468585f80fb029aed07715a8e33a38e4eb306044765a2
PS C:\Users\NekshaSrinivas\SE-1> |
```

## Step 2: Open the local host to view the pages



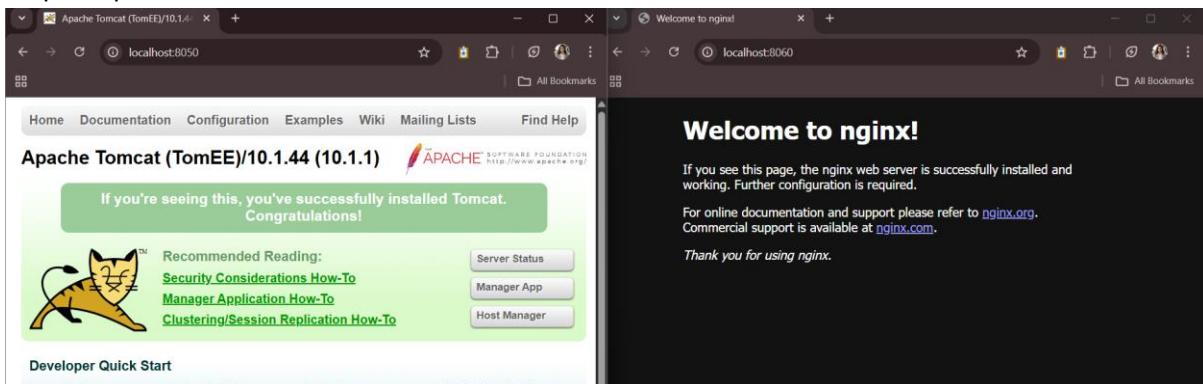
Step 3: Using docker file to run two servers parallelly

```
docker-compose.yml
C: > Users > NekshaSrinivas > SE-1 > comp-1-server > docker-compose.yml
1 services:
2   web:
3     image: nginx
4     ports:
5       - "8060:80"
6   db:
7     image: tomee
8     ports:
9       - "8050:8080"
10
```

Step 4: Use the docker-compose up -d command to execute the docker file

```
No configuration file provided, not found
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> ren Dockerfile docker-compose
.yml
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> docker-compose up -d
[+] Running 3/3
✓ Network comp-1-server_default  C...          0.1s
✓ Container comp-1-server-db-1    St...         0.6s
✓ Container comp-1-server-web-1  S...          0.7s
PS C:\Users\NekshaSrinivas\SE-1\comp-1-server> |
```

Step 5: Open the localhost to view the servers



## WORD-PRESS:

Step 1: Create a docker-compose file and write the content for wordpress and mysql

```
PS C:\Users\NekshaSrinivas\SE-1> cd mysql
PS C:\Users\NekshaSrinivas\SE-1\mysql> notepad docker-compose
PS C:\Users\NekshaSrinivas\SE-1\mysql> ls

Directory: C:\Users\NekshaSrinivas\SE-1\mysql

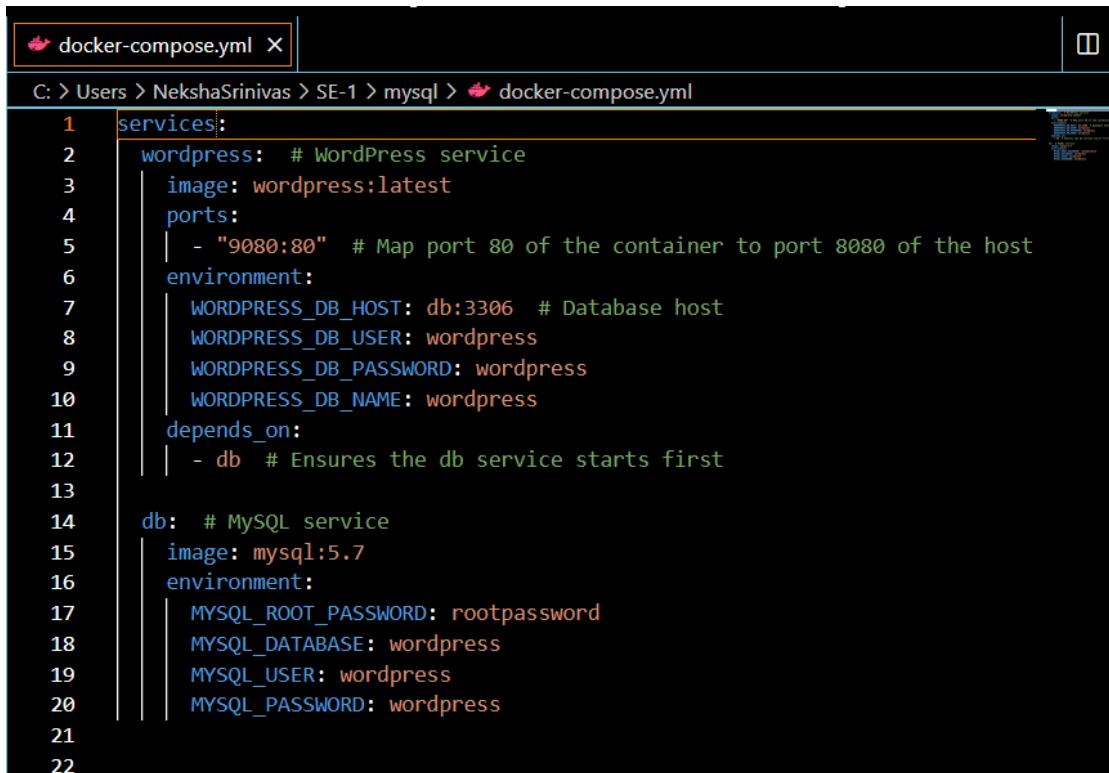
Mode                LastWriteTime        Length Name
----                -----          ----  --
-a----       26-08-2025      18:48           672 docker-compose.txt

PS C:\Users\NekshaSrinivas\SE-1\mysql>
PS C:\Users\NekshaSrinivas\SE-1\mysql> ren docker-compose.txt docker-compose
.yml
PS C:\Users\NekshaSrinivas\SE-1\mysql> ls

Directory: C:\Users\NekshaSrinivas\SE-1\mysql

Mode                LastWriteTime        Length Name
----                -----          ----  --
-a----       26-08-2025      18:48           672 docker-compose.yml
```

Step 2: docker-compose.yml file:



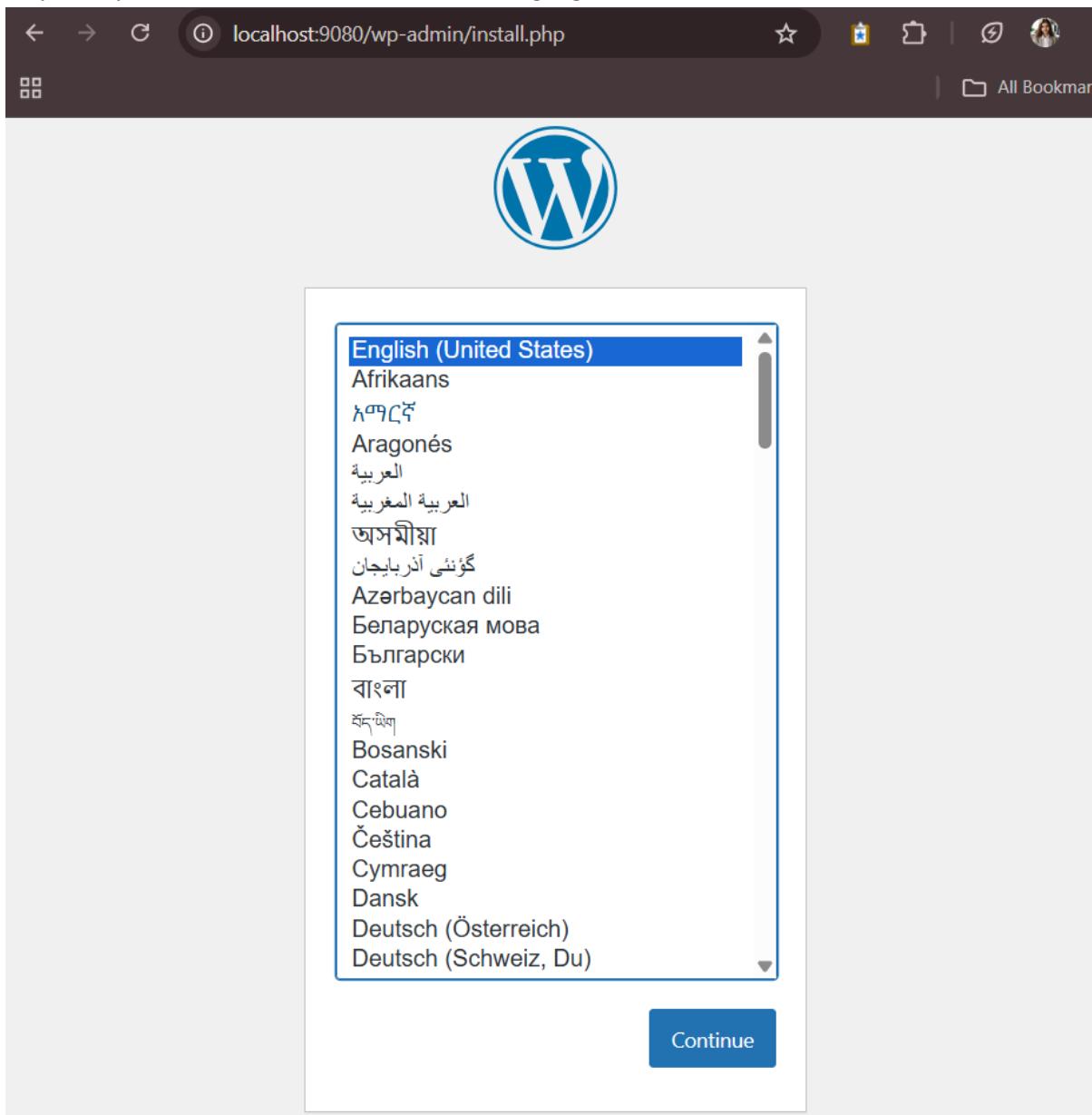
The screenshot shows a code editor window with the title "docker-compose.yml". The file content is as follows:

```
1 services:
2   wordpress: # WordPress service
3     image: wordpress:latest
4     ports:
5       - "9080:80" # Map port 80 of the container to port 8080 of the host
6     environment:
7       WORDPRESS_DB_HOST: db:3306 # Database host
8       WORDPRESS_DB_USER: wordpress
9       WORDPRESS_DB_PASSWORD: wordpress
10      WORDPRESS_DB_NAME: wordpress
11      depends_on:
12        - db # Ensures the db service starts first
13
14   db: # MySQL service
15     image: mysql:5.7
16     environment:
17       MYSQL_ROOT_PASSWORD: rootpassword
18       MYSQL_DATABASE: wordpress
19       MYSQL_USER: wordpress
20       MYSQL_PASSWORD: wordpress
```

Step 3: Use the docker-compose up -d command to start the compose

```
PS C:\Users\NekshaSrinivas\SE-1\mysql> docker-compose up -d
[+] Running 3/3
  ✓ Network mysql_default          Created              0.1s
  ✓ Container mysql-db-1           Started             0.8s
  ✓ Container mysql-wordpress-1   Start...            1.0s
PS C:\Users\NekshaSrinivas\SE-1\mysql>
```

Step 4: Open in the local host and select the language



## Step 5: Fill the details in the welcome page

### Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

**Site Title**

Hey

**Username**

Neksha Srinivas

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**

Sri@121318

 Hide

Medium

**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email**

edigiralaneksha@gmail.com

Double-check your email address before continuing.

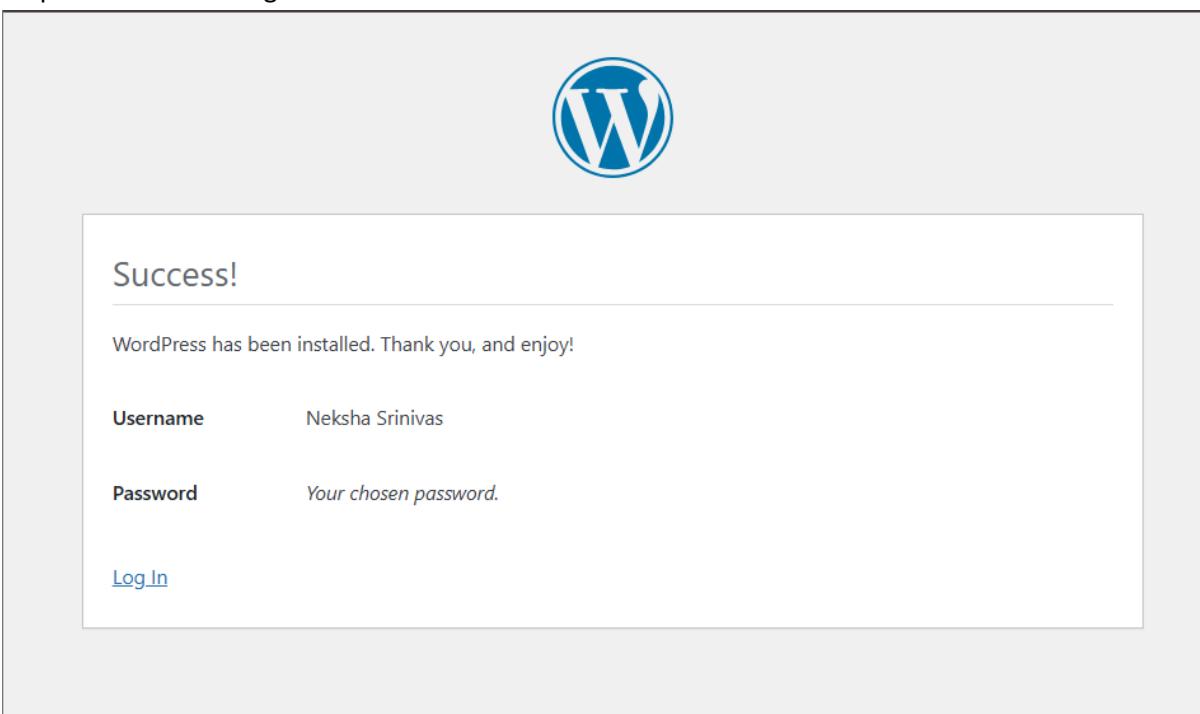
**Search engine visibility**

Discourage search engines from indexing this site

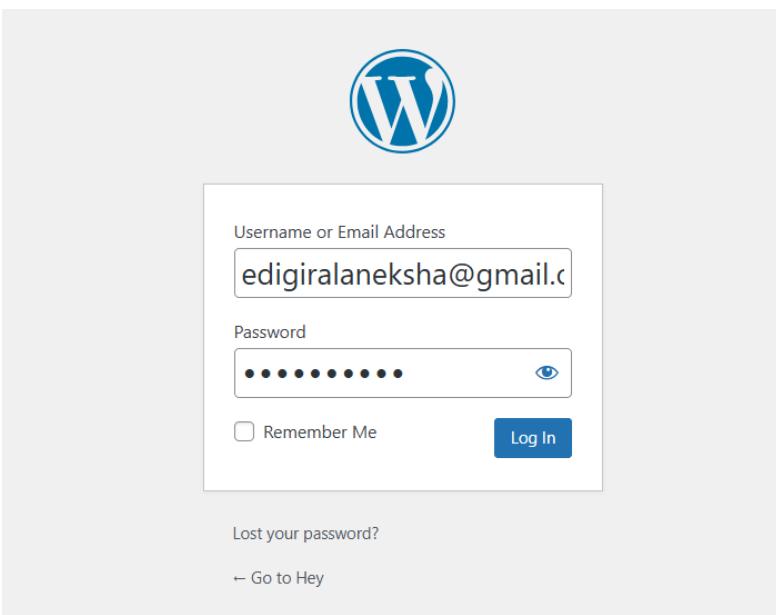
It is up to search engines to honor this request.

[Install WordPress](#)

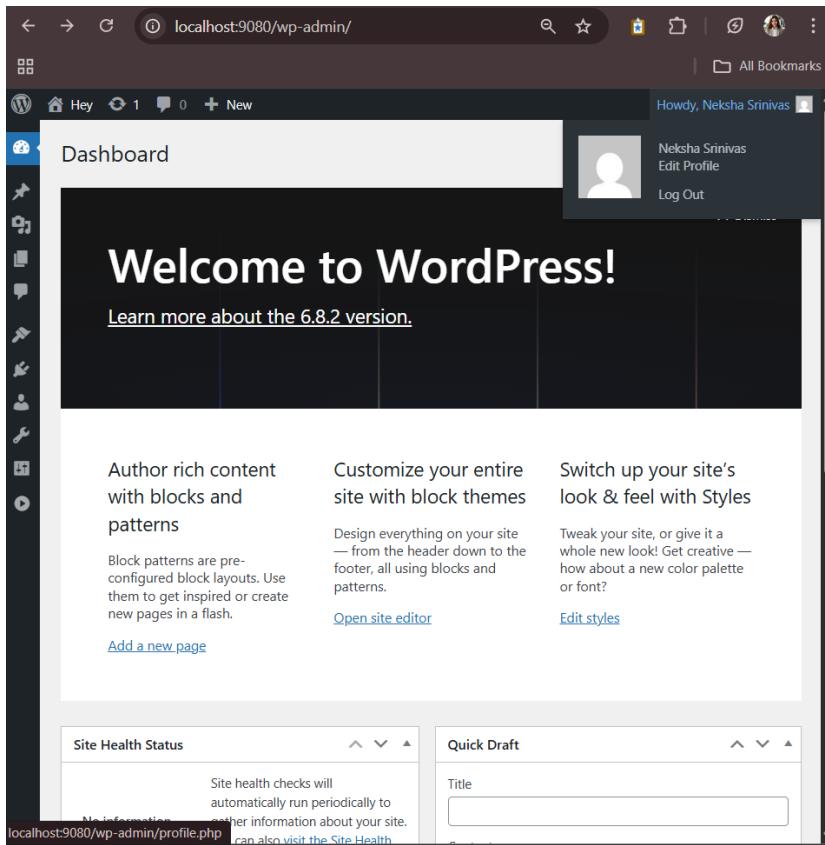
Step 6: Success message will be shown



Step 7: Use your credentials to log in



Step 7: The following page will be shown after login



Task:

Create a simple Flask app in app.py:

Step 1: create a separate folder

```
PS C:\Users\NekshaSrinivas\SE-1> mkdir custom_flask

Directory: C:\Users\NekshaSrinivas\SE-1

Mode                LastWriteTime         Length Name
----                -- -- -- -- -- -- -- -- --
d-----        28-08-2025      10:01                 custom_flask

PS C:\Users\NekshaSrinivas\SE-1> cd custom_flask
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad app.py
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> ren Dockerfile.txt Dockerfile
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> ls

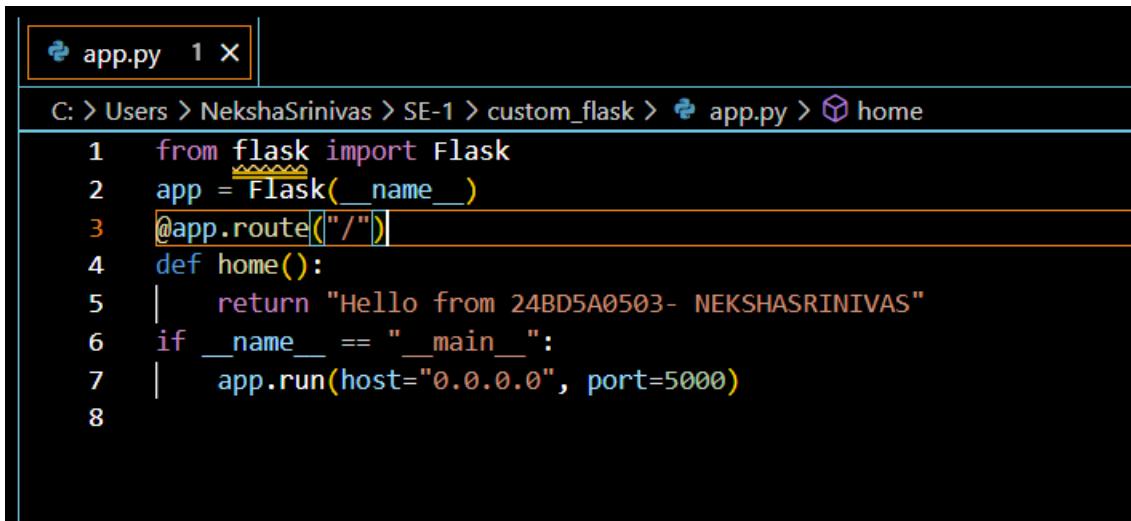
Directory: C:\Users\NekshaSrinivas\SE-1\custom_flask

Mode                LastWriteTime         Length Name
----                -- -- -- -- -- -- -- -- --
-a---        28-08-2025      10:02          187 app.py
-a---        28-08-2025      10:02          105 Dockerfile

PS C:\Users\NekshaSrinivas\SE-1\custom_flask> notepad docker-compose.yml
```

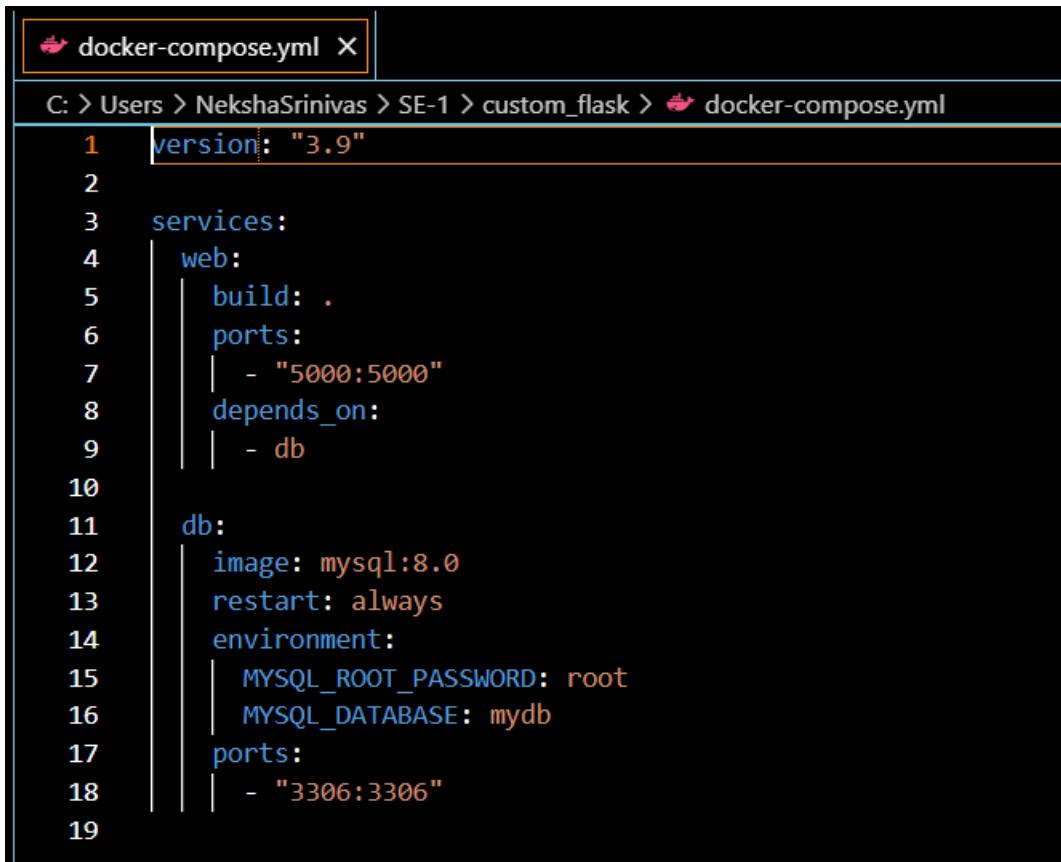
Step 2: write the content of app.py , docker-compose.yml & Dockerfile

app.py:



```
app.py 1 X
C: > Users > NekshaSrinivas > SE-1 > custom_flask > app.py > home
1 from flask import Flask
2 app = Flask(__name__)
3 @app.route("/")
4 def home():
5     return "Hello from 24BD5A0503- NEKSHASRINIVAS"
6 if __name__ == "__main__":
7     app.run(host="0.0.0.0", port=5000)
8
```

docker-compose.yml:



```
docker-compose.yml X
C: > Users > NekshaSrinivas > SE-1 > custom_flask > docker-compose.yml
1 version: "3.9"
2
3 services:
4     web:
5         build: .
6         ports:
7             - "5000:5000"
8         depends_on:
9             - db
10
11     db:
12         image: mysql:8.0
13         restart: always
14         environment:
15             MYSQL_ROOT_PASSWORD: root
16             MYSQL_DATABASE: mydb
17         ports:
18             - "3306:3306"
19
```

Dockerfile:

```
FROM python:3.10-slim
WORKDIR /app
COPY app.py /app/
RUN pip install flask
CMD ["python", "app.py"]
```

Step 3: run the compose using the command docker compose up --build:

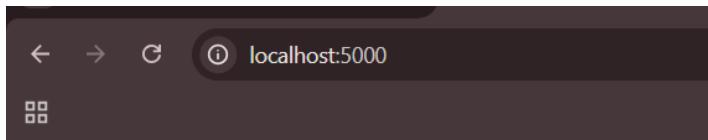
```
Mode           LastWriteTime      Length Name
----           -----          ---- 
-a---       28-08-2025     10:02        187 app.py
-a---       28-08-2025     10:03         82 docker-compose.yml
-a---       28-08-2025     10:02        105 Dockerfile

PS C:\Users\NekshaSrinivas\SE-1\custom_flask> docker compose up --build
validating C:\Users\NekshaSrinivas\SE-1\custom_flask\docker-compose.yml: additional properties 'web' not allowed
PS C:\Users\NekshaSrinivas\SE-1\custom_flask> docker compose up --build
time="2025-08-28T10:24:45+05:30" level=warning msg="C:\\\\Users\\\\NekshaSrinivas\\\\SE-1\\\\custom_flask\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 12/12
 ✓ db Pulled
   ✓ 04fa42a56901 Pull complete               75.7s
   ✓ 500d7b2546c4 Pull complete                1.8s
   ✓ ecc6cc933381 Pull complete                38.2s
   ✓ 5cd63fb67c17 Pull complete                38.5s
   ✓ 4d3eacc36b14 Pull complete                1.7s
   ✓ 9476b8faedba Pull complete                1.7s
   ✓ 789fa151603e Pull complete                3.5s
   ✓ 1756a372d796 Pull complete                1.8s
   ✓ bc0f5543b464 Pull complete                1.9s
   ✓ 131412d69359 Pull complete                67.6s
   ✓ 03ca01bc78d4 Pull complete                42.5s
   ✓ 03ca01bc78d4 Pull complete                1.8s
#1 [internal] load local bake definitions
#1 reading from stdin 542B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 142B 0.0s done
#2 DONE 0.1s

#3 [internal] load metadata for docker.io/library/python:3.10-slim
...
```

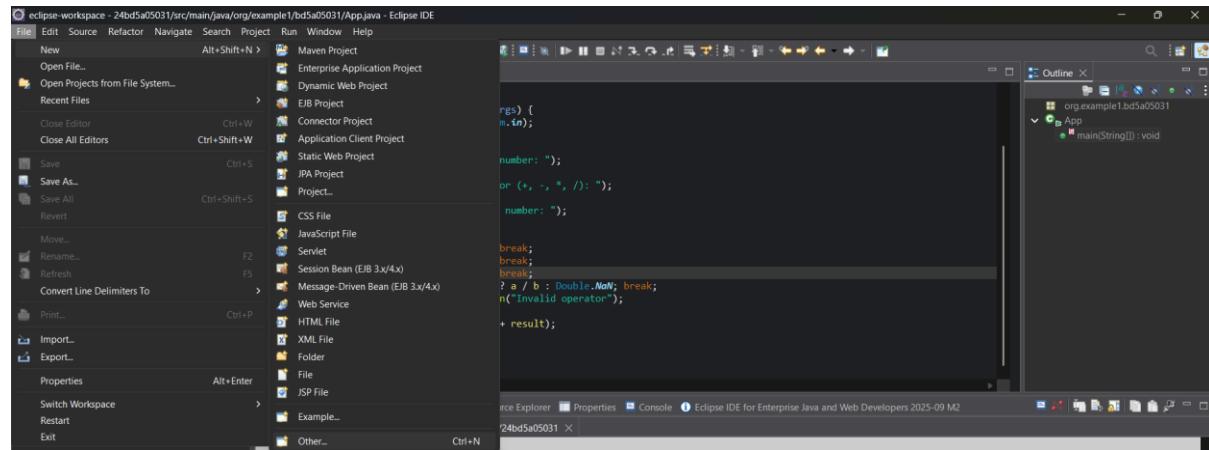
Step 4: Open the local host to view the custom page:



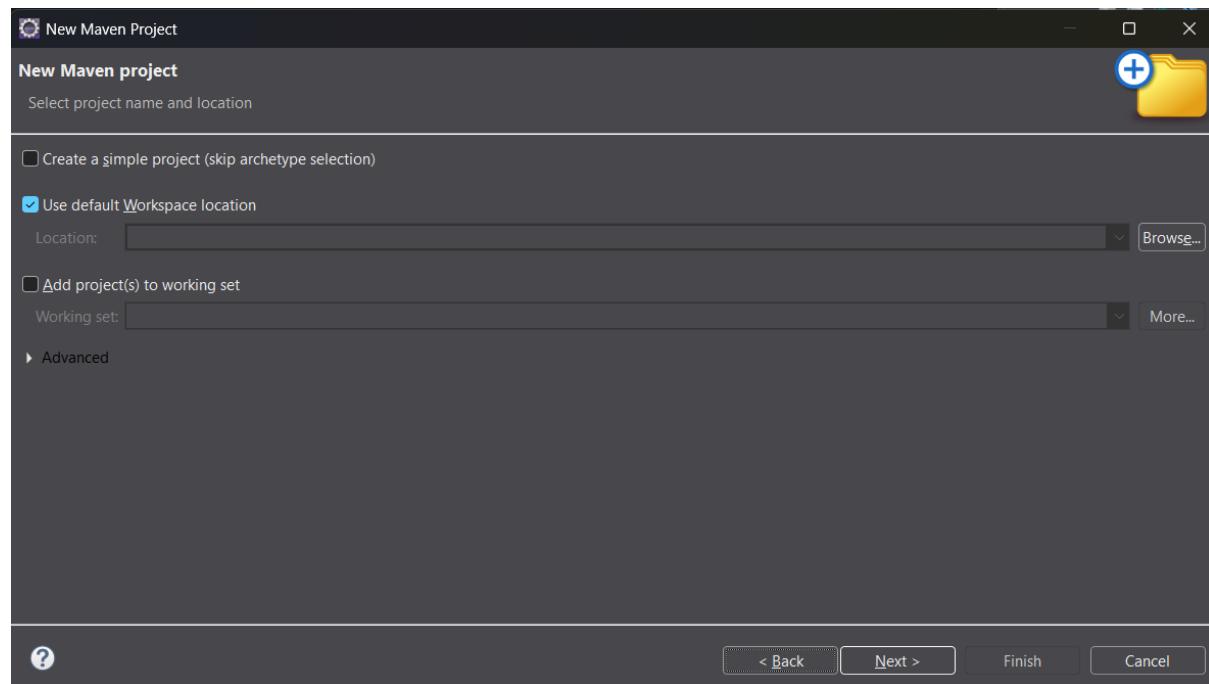
Hello from 24BD5A0503- NEKSHASRINIVAS

## 7. Creating a Multi-Module Maven Project

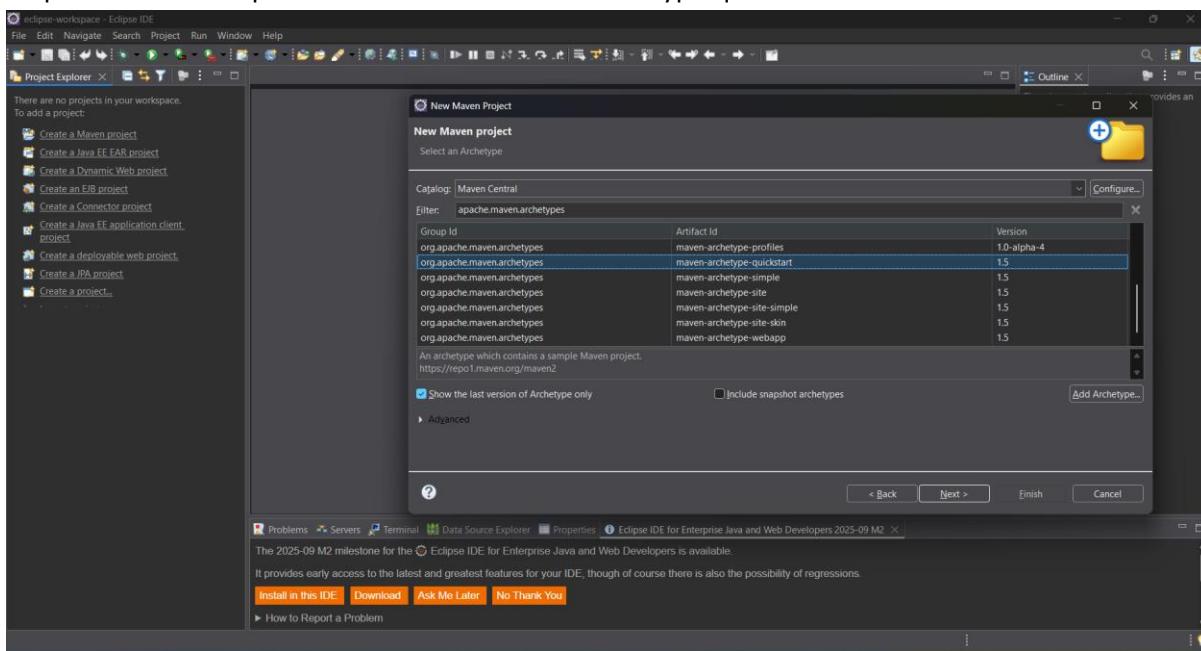
Step-1: Open the eclipse and click on file>new>Maven project



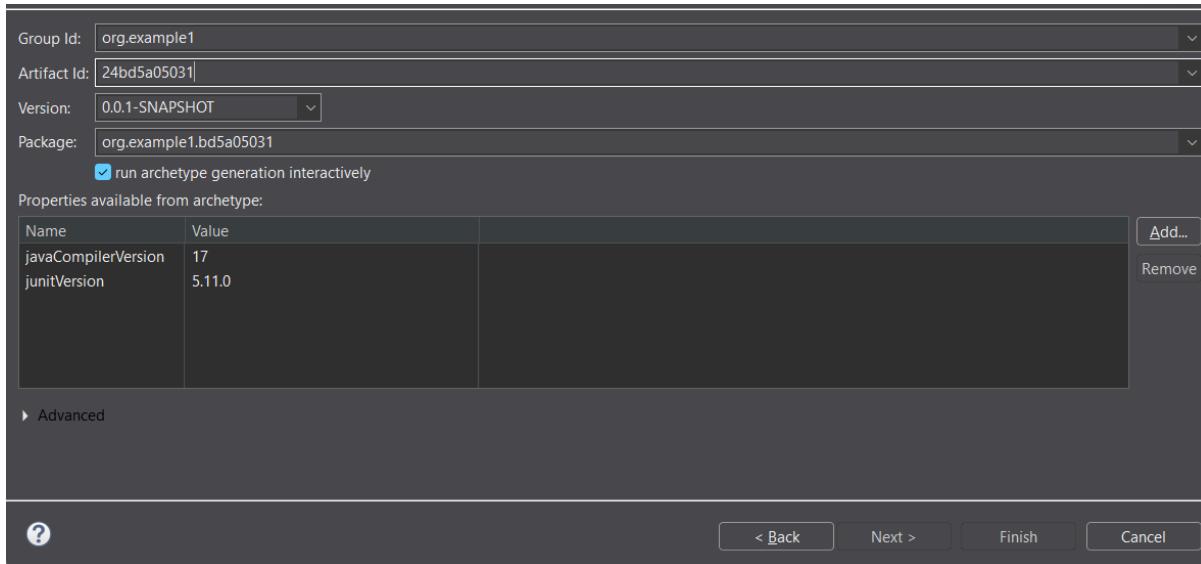
Step-2: select the default workspace and click on next



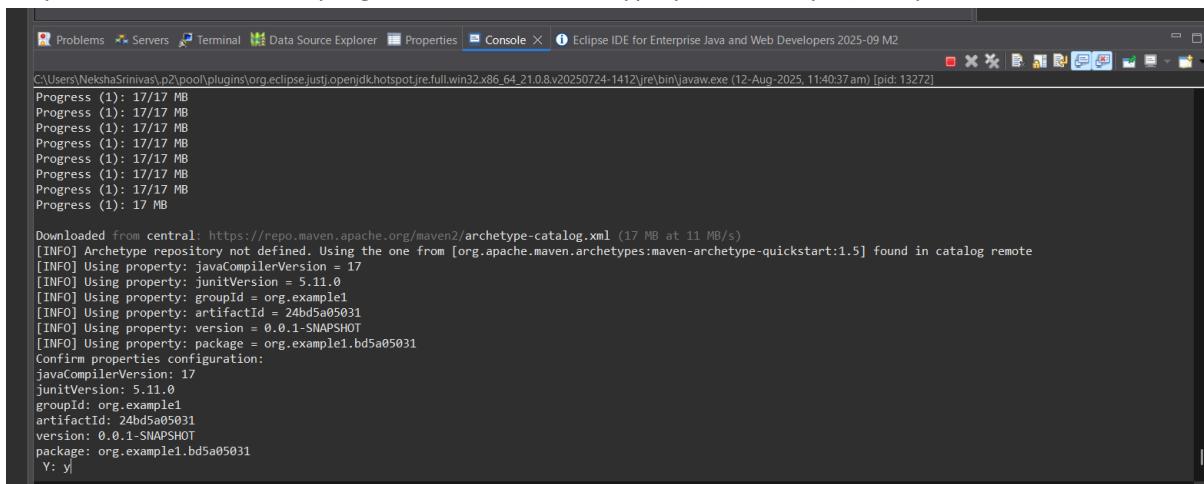
### Step-3: in the filter option select the one maven-archetype-quickstart



### Step-4: give the Group Id and Artifact Id and click on next



Step-5: In the console the progress will be showed type y (refers to yes) and press enter

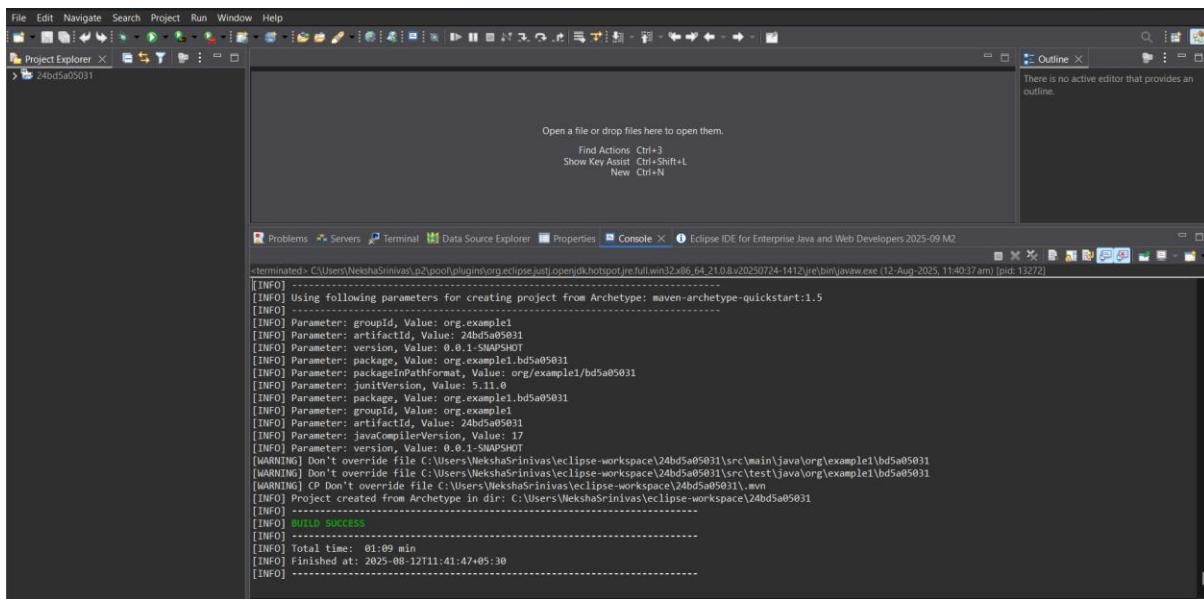


A screenshot of the Eclipse IDE interface. The top menu bar includes 'File', 'Edit', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace shows a 'Project Explorer' view on the left with a single project named '24bd5a05031'. To the right is an 'Outline' view which displays the message 'There is no active editor that provides an outline.' At the bottom is a 'Console' tab where the following output is displayed:

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 11 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in catalog remote
[INFO] Using property: javaCompilerVersion = 17
[INFO] Using property: junitVersion = 5.11.0
[INFO] Using property: groupId = org.example1
[INFO] Using property: artifactId = 24bd5a05031
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = org.example1.bd5a05031
Confirm properties configuration:
javaCompilerVersion: 17
junitVersion: 5.11.0
groupId: org.example1
artifactId: 24bd5a05031
version: 0.0.1-SNAPSHOT
package: org.example1.bd5a05031
Y: y|
```

Step-6: BUILD SUCCESS will be shown



A screenshot of the Eclipse IDE interface. The top menu bar includes 'File', 'Edit', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace shows a 'Project Explorer' view on the left with a single project named '24bd5a05031'. To the right is an 'Outline' view which displays the message 'There is no active editor that provides an outline.' At the bottom is a 'Console' tab where the following output is displayed:

```
[terminated - C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (12-Aug-2025, 11:40:37 am) [pid: 13272]
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.5
[INFO] -----
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: 24bd5a05031
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: org.example1.bd5a05031
[INFO] Parameter: packageInPathFormat, Value: org/example1/bd5a05031
[INFO] Parameter: junitVersion, Value: 5.11.0
[INFO] Parameter: groupId, Value: org.example1
[INFO] Parameter: artifactId, Value: org.example1
[INFO] Parameter: javaCompilerVersion, Value: 17
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\main\java\org\example1\bd5a05031
[WARNING] Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\src\test\java\org\example1\bd5a05031
[WARNING] CP Don't override file C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031\.mvn
[INFO] Project created from Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a05031
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:09 min
[INFO] Finished at: 2025-08-12T11:41:47+05:30
[INFO] -----
```

## Step-6: write the code in the App.java file

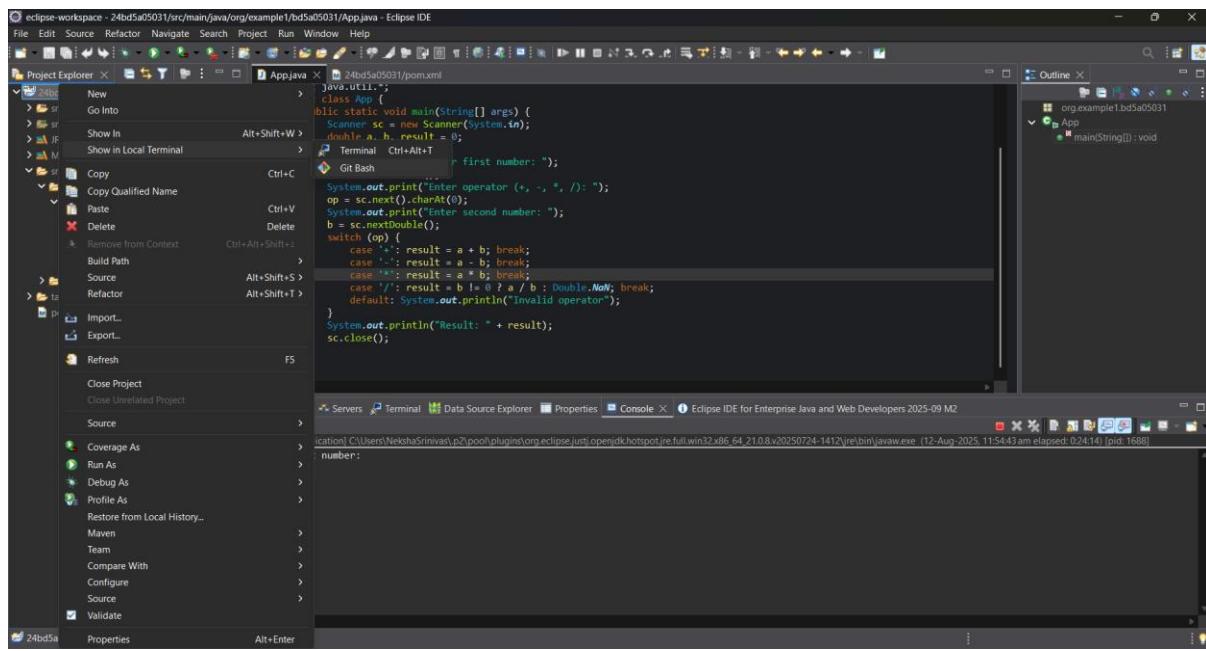
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a package named `org.example1.bd5a05031` containing a source folder `src` which has a Java class `App.java`.
- Code Editor (App.java):** Displays the following Java code:

```
1 package org.example1.bd5a05031;
2
3 /**
4  * Hello world!
5  */
6 import java.util.*;
7 public class App {
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        double a, b, result = 0;
11        char op;
12        System.out.print("Enter first number: ");
13        a = sc.nextDouble();
14        System.out.print("Enter operator (+, -, *, /): ");
15        op = sc.next().charAt(0);
16        System.out.print("Enter second number: ");
17        b = sc.nextDouble();
18        switch (op) {
19            case '+': result = a + b; break;
20            case '-': result = a - b; break;
21            case '*': result = a * b; break;
22            case '/': result = b != 0 ? a / b : Double.NaN; break;
23            default: System.out.println("Invalid operator");
24        }
25        System.out.println("Result: " + result);
26        sc.close();
27    }
28 }
```
- Console:** Shows the output of running the application:

```
Enter first number: 5
Enter operator (+, -, *, /): +
Enter second number: 10
Result: 20.0
```

## Step-7: right click on the root folder and select show in git bash



Step-8: push to the git repo

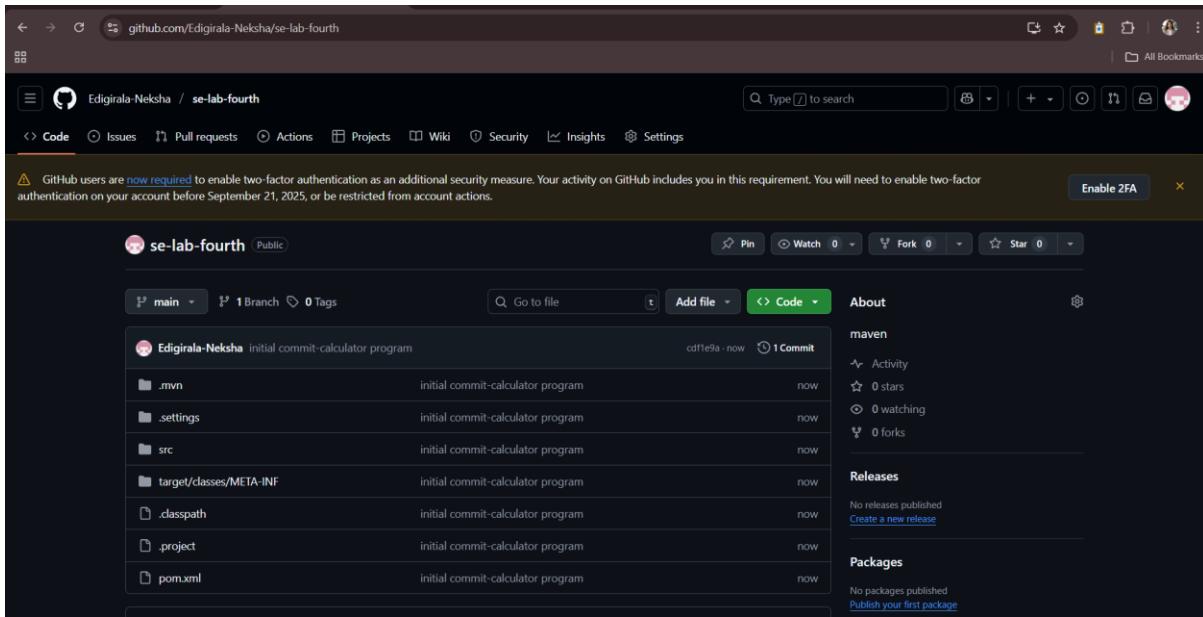
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "24bd5a0531". The structure includes a src folder containing main, test, and JRE System Library (JavaSE-17). The main folder contains Java and org packages, with example1 and bd5a0531 subfolders. An App.java file is selected.
- Terminal:** Displays a terminal session for a user named NekshaSrinivas on a MINGW64 system. The session starts with initializing an empty Git repository, adding a remote origin, and committing changes. It then pushes the changes to a GitHub repository named "edigirala-Neksha/se-lab-fourth".
- Outline:** Shows the class structure for App.java.

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0531 (master)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0531/.git/
$ git remote add origin https://github.com/edigirala-Neksha/se-lab-fourth
$ git add .
$ git commit -m "Initial commit"
[master (root-commit) 32d32] Initial commit
  1 file changed, 1 insertion(+)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 10 threads.
Compressing objects: 100% (27/27), done.
Writing objects: 100% (32/32), 3.99 KiB | 408.00 KiB/s, done.
Total 32 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/edigirala-Neksha/se-lab-fourth
 * [new branch]    main > main
branch 'main' set up to track 'origin/main'.
```

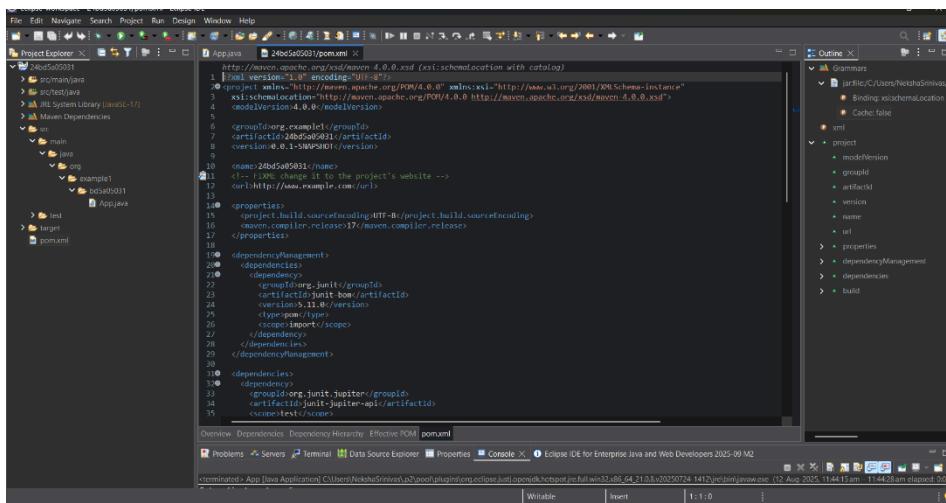
Git repo:

Git repo link: <https://github.com/Edigirala-Neksha/se-lab-fourth>



pom.xml file:

Shows the structure-

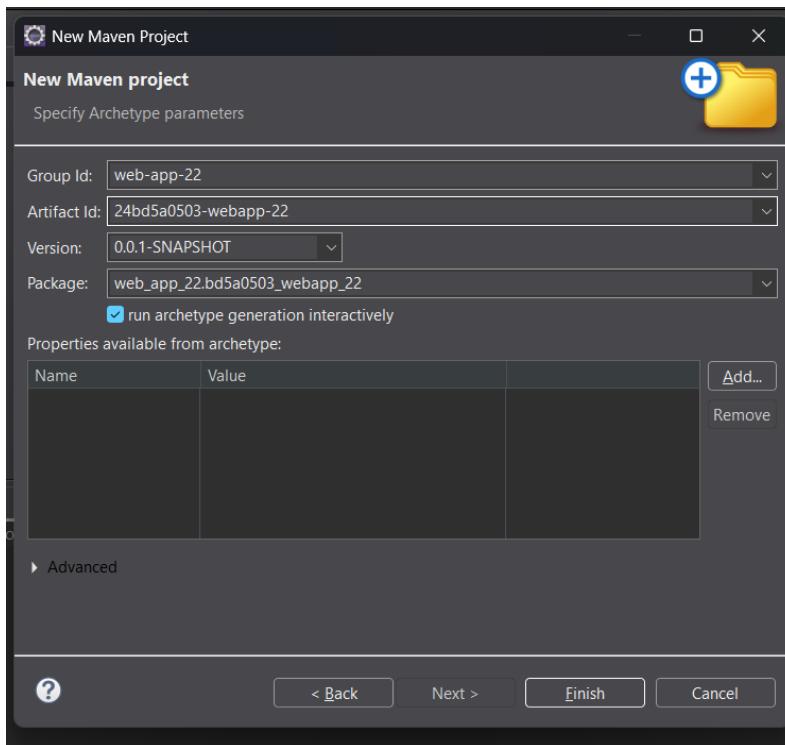


The screenshot shows the Eclipse IDE interface with the pom.xml file open. The Project Explorer view on the left displays the project structure with files like src/main/java, src/test/java, and pom.xml. The Outline view on the right shows the XML structure of the pom.xml file, including sections like `<groupId>`, `<artifactId>`, `<version>`, `<dependencies>`, and `<dependencyManagement>`. The code editor in the center contains the XML code for the pom.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>zbd5a0503</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>zbd5a0503</name>
  <url>http://www.example.com/url</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.release>17</maven.compiler.release>
  </properties>
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>5.11</version>
        <scope>import</scope>
      </dependency>
      <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.11.1</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>5.11.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Creating maven-web project:

Step 1: Create a new maven project and give the details



## Step 2: Click y to continue the creation of project

```
C:\Users\NekshaSrinivas\p2\pool\plugins\org.eclipse.jst\openjdk.hotspot.jre.full.win32.x86_64_21.0.8v20250724-1412\jre\bin\javaw.exe (02-Sept-2025, 7:19:56 pm) [pid: 13772]
Progress (1): 17/17 MB
Progress (1): 17 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (17 MB at 9.1 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in catalog remote
[INFO] Using property: groupId = web-app-22
[INFO] Using property: artifactId = 24bd5a0503-webapp-22
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = web_app_22.bd5a0503_webapp_22
Confirm properties configuration:
groupId: web-app-22
artifactId: 24bd5a0503-webapp-22
version: 0.0.1-SNAPSHOT
package: web_app_22.bd5a0503_webapp_22
Y: y
```

## Step 3: If the build is success it will show the message

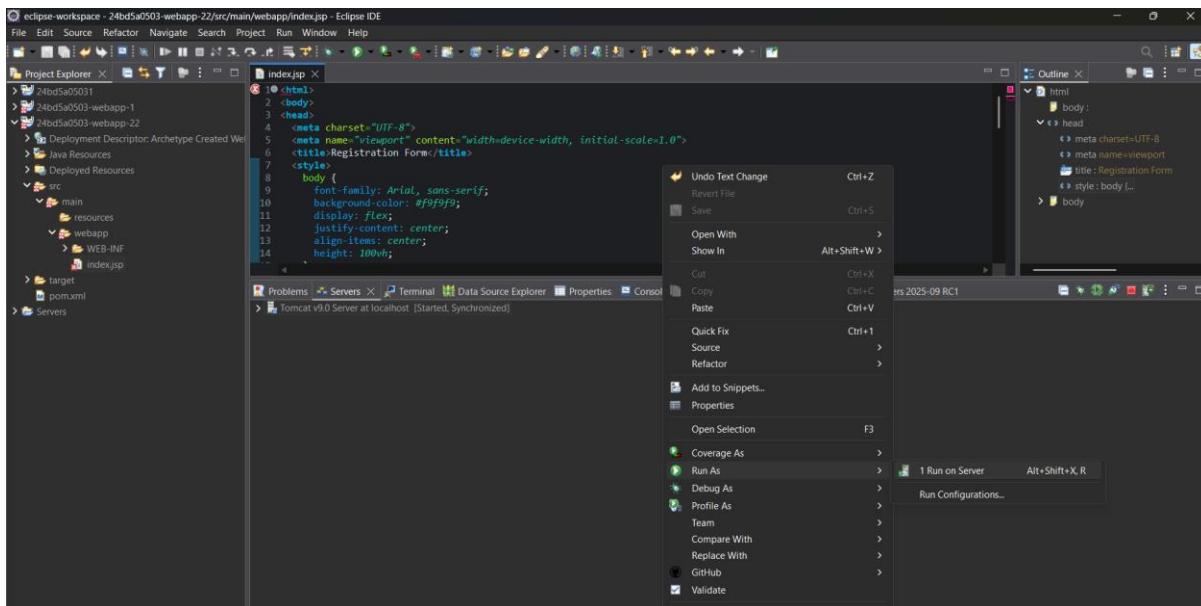
```
package: web_app_22.bd5a0503_webapp_22
Y: y
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: basedir, Value: C:\Users\NekshaSrinivas\eclipse-workspace
[INFO] Parameter: package, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: groupId, Value: web-app-22
[INFO] Parameter: artifactId, Value: 24bd5a0503-webapp-22
[INFO] Parameter: packageName, Value: web_app_22.bd5a0503_webapp_22
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\NekshaSrinivas\eclipse-workspace\24bd5a0503-webapp-22
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 43.500 s
[INFO] Finished at: 2025-09-02T19:20:41+05:30
[INFO] -----
```

## Step 4: write the html code for the web page:

```
index.jsp X
⑥ <html>
  <body>
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Registration Form</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        background-color: #f9f9f9;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
      }
      .form-container {
        background: #fff;
        padding: 20px 30px;
        border-radius: 10px;
        box-shadow: 0 4px 10px rgba(0,0,0,0.1);
        width: 300px;
      }
      .form-container h2 {
        text-align: center;
        margin-bottom: 20px;
      }
      .form-container input {
        width: 100%;
        padding: 10px;
        margin: 8px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
      }
      .form-container button {
        width: 100%;
        padding: 10px;
        background: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
      }
    </style>
  </head>
  <body>
    <div class="form-container">
      <h2>Registration Form</h2>
      <form>
        <div>
          <label>Name:</label>
          <input type="text" placeholder="Enter Name" required>
        </div>
        <div>
          <label>Email:</label>
          <input type="email" placeholder="Enter Email" required>
        </div>
        <div>
          <label>Password:</label>
          <input type="password" placeholder="Enter Password" required>
        </div>
        <div>
          <label>Confirm Password:</label>
          <input type="password" placeholder="Enter Confirm Password" required>
        </div>
        <div>
          <input type="checkbox" checked=""> I agree to the terms and conditions
        </div>
        <div>
          <button type="submit">Register</button>
        </div>
      </form>
    </div>
  </body>
</html>
```

```
index.jsp X
  border-radius: 5px;
33  }
34  .form-container button {
35    width: 100%;
36    padding: 10px;
37    background: #4CAF50;
38    color: white;
39    border: none;
40    border-radius: 5px;
41    cursor: pointer;
42  }
43  .form-container button:hover {
44    background: #45a049;
45  }
46 </style>
47 </head>
48 <body>
49 <div class="form-container">
50   <h2>Registration Form</h2>
51   <form action="#" method="post">
52     <label for="fullname">Full Name</label>
53     <input type="text" id="fullname" name="fullname" placeholder="Enter your name" required>
54
55     <label for="email">Email</label>
56     <input type="email" id="email" name="email" placeholder="Enter your email" required>
57
58     <label for="password">Password</label>
59     <input type="password" id="password" name="password" placeholder="Enter password" required>
60
61     <label for="confirm">Confirm Password</label>
62     <input type="password" id="confirm" name="confirm" placeholder="Confirm password" required>
63
64     <button type="submit">Register</button>
65   </form>
66 </div>
67 </body>
68 </html>
69
```

## Step 5: Select run on server



Step 6: It will show the following output:

The screenshot shows a web browser window with a dark header bar. The address bar displays the URL: `localhost:8080/24bd5a0503-webapp-22/index.jsp`. Below the header is a registration form titled "Registration Form". The form consists of five input fields: "Full Name" (placeholder: "Enter your name"), "Email" (placeholder: "Enter your email"), "Password" (placeholder: "Enter password"), and "Confirm Password" (placeholder: "Confirm password"). A green "Register" button is located at the bottom right of the form area.

← → ⌂ ⓘ localhost:8080/24bd5a0503-webapp-22/index.jsp

Registration Form

Full Name  
Enter your name

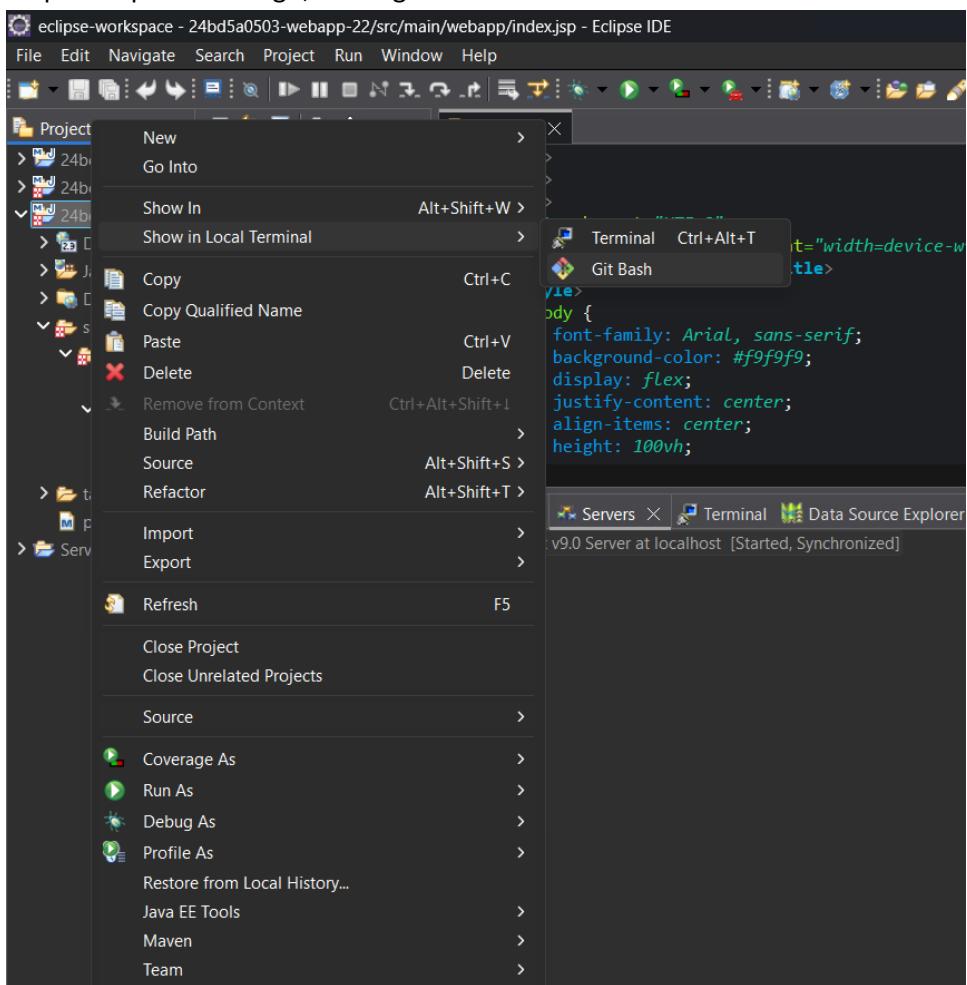
Email  
Enter your email

Password  
Enter password

Confirm Password  
Confirm password

Register

Step 7: To push it into git, select git bash from show in local terminal



Step 8: use the command of git to push the maven web project

```
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (master)
$ git init
Initialized empty Git repository in C:/Users/NekshaSrinivas/eclipse-workspace/24bd5a0503-webapp-22/.git/
NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git add .

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git commit -m "initial form"
[main (root-commit) 636aeef] initial form
 16 files changed, 254 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/.jsdtscope
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 create mode 100644 .settings/org.eclipse.wst.commonn.component
 create mode 100644 .settings/org.eclipse.wst.commonn.project.facet.core.xml
 create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container
 create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name
 create mode 100644 .settings/org.eclipse.wst.validationn.prefs
 create mode 100644 pom.xml
 create mode 100644 src/main/webapp/WEB-INF/web.xml
```

```

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git branch
* main

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$ git push origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 12 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (29/29), 4.43 KiB | 283.00 KiB/s, done.
Total 29 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Edigirala-Neksha/se-webapp-22.git
 * [new branch]      main -> main

NekshaSrinivas@NekshaSrinivas MINGW64 ~/eclipse-workspace/24bd5a0503-webapp-22 (main)
$
```

### Step 9: verify the repo in git hub

The screenshot shows a GitHub repository page for 'se-webapp-22'. The repository has 1 branch and 0 tags. The 'Code' tab is selected. The repository contains several files: '.settings', 'src/main/webapp', 'target/m2e-wtp/web-resources/META-INF', '.classpath', '.project', and 'pom.xml'. The 'README' file is also present. On the right side, there is an 'About' section for the repository, which includes information about the last commit (by 'se-lab-week7'), activity (0 stars, 0 forks, 0 watching), releases (no releases published), packages (no packages published), and languages (not specified).

## **8. Jenkins Automation**

### **Steps for MavenJava Automation**

#### **Step 1: Open Jenkins (localhost:8888)**

Click on "New Item" (left side menu) and name it as maven\_java > select freestyle project > click on "OK"

New Item

Enter an item name  
maven\_java

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.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

#### **Step 2: Configuration of maven\_java project**

Give the description

Configure

General

Enabled

Description

Java Build demo

Plain text Preview

Discard old builds ?

GitHub project

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Discard old builds ?

Github project

Plain text Preview

In the source code management select git and give the git repo link

The screenshot shows the Jenkins configuration page for a job named 'Mavenjava'. Under the 'Source Code Management' section, the 'Git' option is selected. A 'Repository URL' field contains the value `https://github.com/SarvikaSomishetty/eclipse-maven-projects.git`. Below it, a 'Credentials' dropdown is set to '- none -'. There is also a '+ Add' button and an 'Advanced' dropdown.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as clean

The screenshot shows the Jenkins configuration page for the same job. Under the 'Build Steps' section, two 'Invoke top-level Maven targets' steps are present. Both steps have 'Maven Version' set to 'MAVEN\_HOME' and 'Goals' set to 'clean'. The 'Advanced' dropdown is visible for both steps. At the bottom of the page, there are 'Save' and 'Apply' buttons.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as install

The screenshot shows the Jenkins configuration page for a job named "Mavenjava". The left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Build Steps (which is selected), and Post-build Actions. The main area is titled "Configure" and shows the "Goals" section with "clean" and "install" listed. A detailed view of the "Build Steps" section is shown, specifically the "Invoke top-level Maven targets" step. This step has "MAVEN\_HOME" set as the Maven Version and "install" set as the Goals. There is also an "Advanced" dropdown. At the bottom of the "Build Steps" section is a "Post-build Actions" section with a note about what happens after a build completes.

Goals

clean

Advanced

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Invoke top-level Maven targets

Maven Version

MAVEN\_HOME

Goals

install

Advanced

Add build step

Post-build Actions

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

Save Apply

In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give “\*\*/\*”

For the second post build action,

In the post build actions > click on add post build action > select build other projects > give projects to build as MavenJava\_Test

Click on apply and save

The screenshot shows the Jenkins configuration interface for a job named 'Mavenjava'. The left sidebar lists configuration sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' section is currently selected and highlighted.

The main content area is titled 'Post-build Actions' and contains two defined actions:

- Archive the artifacts**: Set to archive files matching the pattern '\*\*/\*'. An 'Advanced' dropdown is available.
- Build other projects**: Set to build the project 'MavenJava\_Test'. Trigger options include 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'.

At the bottom of the configuration page are 'Save' and 'Apply' buttons.

If the build is success:

The screenshot shows the Jenkins web interface for the 'maven\_web\_build' job. The job status is green, indicating success. The build name is 'maven\_web\_build'. The build number is #2, which was last updated 27 minutes ago. The build log link is provided. The workspace is labeled 'web build demo'. A 'Last Successful Artifacts' section is shown, along with a 'Downstream Projects' section listing 'maven\_web\_test'. A 'Permalinks' section provides links to the build history. The bottom of the page shows the Jenkins version as 2.489 and the REST API endpoint.

REST API Jenkins 2.489

### Step 3: Create Freestyle Project (e.g., MavenJava\_Test)

Click on new item > give item name as mavaen\_java\_test or MavenJava\_Test and select free style project and click ok

New Item

Enter an item name  
maven\_java\_test

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.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

### Step 4: Configuration of maven\_java project

Give the description

MavenJava\_Test Config [Jenkins]

localhost:8888/job/MavenJava\_Test/configure

Jenkins

Dashboard > MavenJava\_Test > Configuration

Configure General

Enabled

Description  
Test demo

Plain text [Preview](#)

Discard old builds ?

GitHub project

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

In the source code management select none and in environment select “delete workspace before build starts”

The screenshot shows the Jenkins configuration page for the 'MavenJava\_Test' job. Under 'Source Code Management', the 'None' radio button is selected. Under 'Environment', the checkbox for 'Delete workspace before build starts' is checked. At the bottom, there are 'Save' and 'Apply' buttons.

In the build steps> select add a build step> select “copy artifacts from another project” > give project name as Maven java and artifacts to copy as \*\*/\*

The screenshot shows the Jenkins configuration page for the 'MavenJava\_Test' job. Under 'Build Steps', a 'Copy artifacts from another project' step is added. The 'Project name' is set to 'Mavenjava'. The 'Which build' dropdown is set to 'Latest successful build' with the 'Stable build only' checkbox checked. The 'Artifacts to copy' field contains '\*\*/\*'. At the bottom, there are 'Save' and 'Apply' buttons.

In the post build actions> select archive the artifacts and enter files as \*\*/\*

Click on apply and save

The screenshot shows the Jenkins configuration page for the 'MavenJava\_Test' job. In the 'Post-build Actions' section, there are two steps defined:

- Invoke top-level Maven targets**: Maven Version is set to MAVEN\_HOME and Goals is set to test.
- Archive the artifacts**: Files to archive is set to \*\*/\*.

At the bottom, there are 'Save' and 'Apply' buttons.

In the dashboard you will find MavenJava and MavenJava\_Test

The dashboard shows the following projects:

S	W	Name	Last Success	Last Failure	Last Duration
✗	rainy	INTERNAL_JAVA	9 mo 3 days #34	40 sec #15454	0.67 sec
✓	sunny	Mavenjava	13 days #2	N/A	11 sec
✓	sunny	MavenJava_Test	13 days #3	N/A	3.4 sec
✗	rainy	new	9 mo 3 days #3	13 days #4	31 sec
✓	sunny	web_build	9 mo 9 days #8	N/A	8.2 sec
✗	rainy	web_deploy	N/A	9 mo 9 days #15	0.31 sec
✓	sunny	web_test	9 mo 9 days #12	N/A	3.4 sec

If you open the MavenJava file the following will be shown in case on no errors

The screenshot shows the Jenkins interface for the 'MavenJava' job. The top navigation bar includes links for 'Dashboard', 'MavenJava', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The main content area has tabs for 'Status' (highlighted), 'Changes', 'Workspace', and 'Build Now'. Under 'Status', there's a 'Builds' section showing three recent builds: #2 (11:46 AM), #1 (11:45 AM), and a 'Filter' input. To the right, under 'Last Successful Artifacts', a table lists various files with their sizes and 'view' links. Below this is a 'Downstream Projects' section with a single entry: 'MavenJava\_Test'. A 'Permalinks' section at the bottom contains a link to the last build.

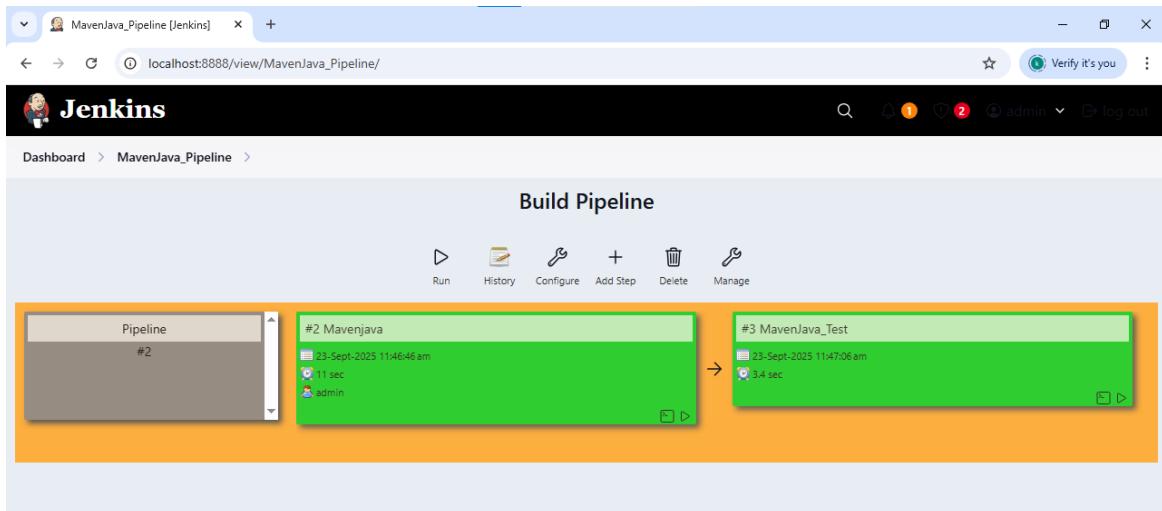
Name	Size	Action
.classpath	1.65 kB	view
.project	1.06 kB	view
.jdtScope	639 B	view
org.eclipse.jdt.core.prefs	616 B	view
org.eclipse.m2e.core.prefs	90 B	view
org.eclipse.wst.common.component	665 B	view
org.eclipse.wst.common.project.facet.core.xml	252 B	view
org.eclipse.wst.jdt.ui.superType.container	49 B	view
org.eclipse.wst.jdt.ui.superType.name	6 B	view
org.eclipse.wst.validation.prefs	50 B	view
Dockerfile	131 B	view
pom.xml	738 B	view
webapp/index.jsp	57 B	view
webapp/WEB-INF/web.xml	222 B	view
pom.properties	71 B	view
org.demo/index.jsp	57 B	view
org.demo/WEB-INF/web.xml	222 B	view
org.demo.war	1.64 kB	view

If you open the MavenJava\_Test file the following will be shown in case on no errors

The screenshot shows the Jenkins interface for the 'MavenJava\_Test' job. The top navigation bar includes links for 'Dashboard', 'MavenJava\_Test', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The main content area has tabs for 'Status' (highlighted), 'Changes', 'Workspace', and 'Build Now'. Under 'Status', there's a 'Builds' section showing three recent builds: #3 (11:47 AM), #2 (11:46 AM), and #1 (11:45 AM), with a 'Filter' input. To the right, under 'Last Successful Artifacts', a table lists various files with their sizes and 'view' links. Below this is an 'Upstream Projects' section with a single entry: 'MavenJava'. A 'Permalinks' section at the bottom contains links to both the last and stable builds.

Name	Size	Action
.classpath	1.65 kB	view
.project	1.06 kB	view
.jdtScope	639 B	view
org.eclipse.jdt.core.prefs	616 B	view
org.eclipse.m2e.core.prefs	90 B	view
org.eclipse.wst.common.component	665 B	view
org.eclipse.wst.common.project.facet.core.xml	252 B	view
org.eclipse.wst.jdt.ui.superType.container	49 B	view
org.eclipse.wst.jdt.ui.superType.name	6 B	view
org.eclipse.wst.validation.prefs	50 B	view
Dockerfile	131 B	view
pom.xml	738 B	view
webapp/index.jsp	57 B	view
webapp/WEB-INF/web.xml	222 B	view
pom.properties	71 B	view
org.demo/index.jsp	57 B	view
org.demo/WEB-INF/web.xml	222 B	view
org.demo.war	1.64 kB	view

## MavenJava\_pipeline

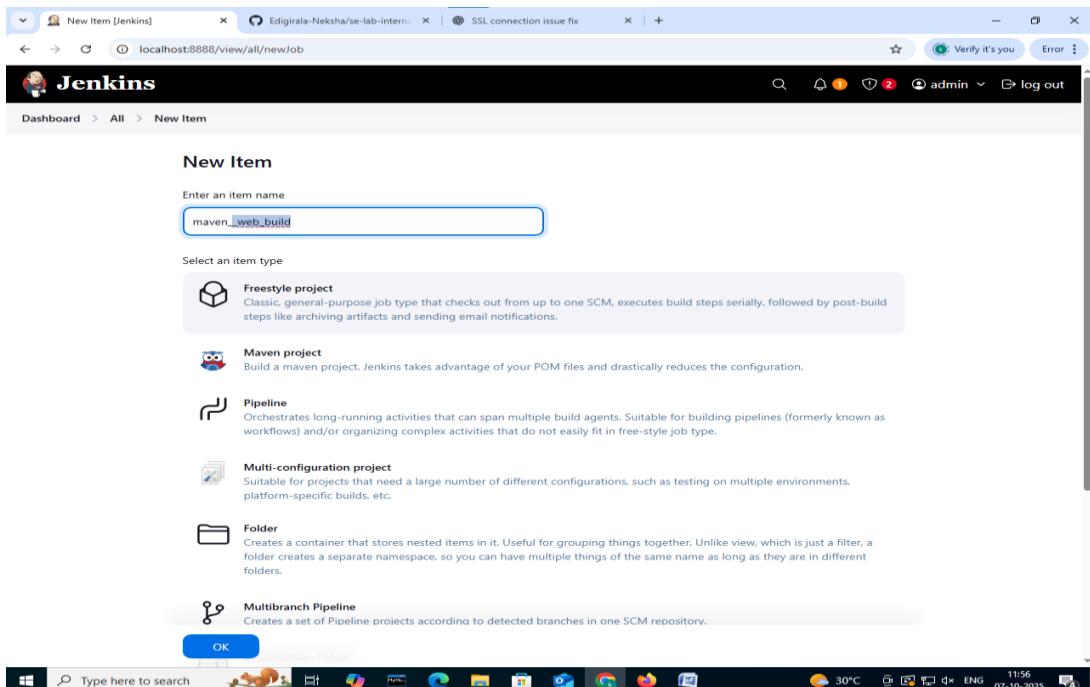


## II. Maven Web Automation Steps:

Create Freestyle Project (e.g., MavenWeb\_Build)

### **Step 1: Open Jenkins (localhost:8888)**

Click on "New Item" (left side menu) and name it as maven\_web\_build > select freestyle project > click on "OK"



## Step 2: Configuration of maven\_web\_build project

Give the description

The screenshot shows the Jenkins configuration interface for the 'maven\_web\_build' project. The top navigation bar includes tabs for 'maven\_web\_build Config [Jenkins]', 'Edigirala-Neksha/se-lab-intern...', and 'SSL connection issue fix'. The main title is 'localhost:8888/job/maven\_web\_build/configure'. The Jenkins logo and user information ('Verify it's you', 'admin', 'log out') are visible.

The left sidebar has a 'Configure' section with links to 'General', 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected, showing the 'Enabled' status (checked) and a 'Description' field containing 'web build demo'. Below this are several optional checkboxes: 'Discard old builds', 'GitHub project', 'Permission to Copy Artifact', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. A 'Plain text' link is also present.

The 'Source Code Management' section is expanded, with a sub-section titled 'Source Code Management' and a note: 'Connect and manage your code repository to automatically pull the latest code for your builds.' It contains two buttons: 'Save' (blue) and 'Apply'.

In the source code management select git and give the git repo link

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_build". The left sidebar lists "Configure", "General", "Source Code Management", "Triggers", "Environment", "Build Steps", and "Post-build Actions". The "Source Code Management" section is selected and expanded, showing the "Git" tab. Under "Repositories", there is one entry with the URL `https://github.com/Edigirala-Neksha/se-lab-internal-1.git`. Under "Branches to build", there is one entry with the specifier `*/*main`. At the bottom, there are "Save" and "Apply" buttons.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as clean

For the second build step,

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as install

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_build". The "Build Steps" section is active, displaying two "Invoke top-level Maven targets" steps. Both steps have "MAVEN\_HOME" selected for Maven Version and "clean" or "install" selected for Goals. The interface includes a sidebar with General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions options. At the bottom are Save and Apply buttons.

Configure

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Invoke top-level Maven targets ?

Maven Version: MAVEN\_HOME

Goals: clean

Advanced

Invoke top-level Maven targets ?

Maven Version: MAVEN\_HOME

Goals: install

Advanced

Add build step ▾

Save Apply

In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give “\*\*/\*”

For the second post build action,

In the post build actions > click on add post build action > select build other projects > give projects to build as maven\_web\_test

Click on apply and save

The screenshot shows the Jenkins configuration page for the 'maven\_web\_build' job. The 'Post-build Actions' section is active. It contains two entries:

- Archive the artifacts**: Set to archive files matching the pattern '\*\*/\*'. An 'Advanced' dropdown is visible.
- Build other projects**: Set to build the project 'maven\_web\_test'. The trigger option 'Trigger only if build is stable' is selected.

At the bottom, there are 'Save' and 'Apply' buttons. The status bar at the bottom right indicates 'Jenkins 2.489'.

Create Freestyle Project (e.g., MavenWeb\_Test):

### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_web\_test > select freestyle project > click on "OK"

The screenshot shows the Jenkins 'New Item' configuration page. The 'Item name' field contains 'maven\_web\_test'. Under 'Select an item type', the 'Freestyle project' option is selected, described as a '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.' Other options shown include 'Maven project', 'Pipeline', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. At the bottom, there is an 'OK' button.

### Step 2: Configuration of maven\_web\_test project

Give the description

The screenshot shows the Jenkins 'Configuration' page for the 'maven\_web\_test' project. The 'General' tab is selected. The 'Description' field contains 'test demo'. The 'Enabled' checkbox is checked. On the left sidebar, other tabs include 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. At the bottom, there are checkboxes for 'Discard old builds' and 'GitHub project'.

In the source code management select none and in environment select “delete workspace before build starts”

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_test". The left sidebar lists "Configure", "General", "Source Code Management", "Triggers", "Environment", "Build Steps", and "Post-build Actions". The "Source Code Management" section is active, showing a radio button for "None" selected over "Git". The "Triggers" section contains several options like "Trigger builds remotely", "Build after other projects are built", etc., all unselected. The "Environment" section has a checked checkbox for "Delete workspace before build starts" and several other unselected options: "Use secret text(s) or file(s)", "Provide Configuration files", "Add timestamps to the Console Output", and "Inspect build log for published build scans". At the bottom are "Save" and "Apply" buttons. The browser address bar shows "localhost:8888/job/maven\_web\_test/configure". The taskbar at the bottom includes icons for Start, Search, File Explorer, File History, Task View, Edge, File Manager, Photos, Camera, Google Chrome, Mozilla Firefox, and Notepad. System status shows 500... notifications, battery level, and network connection.

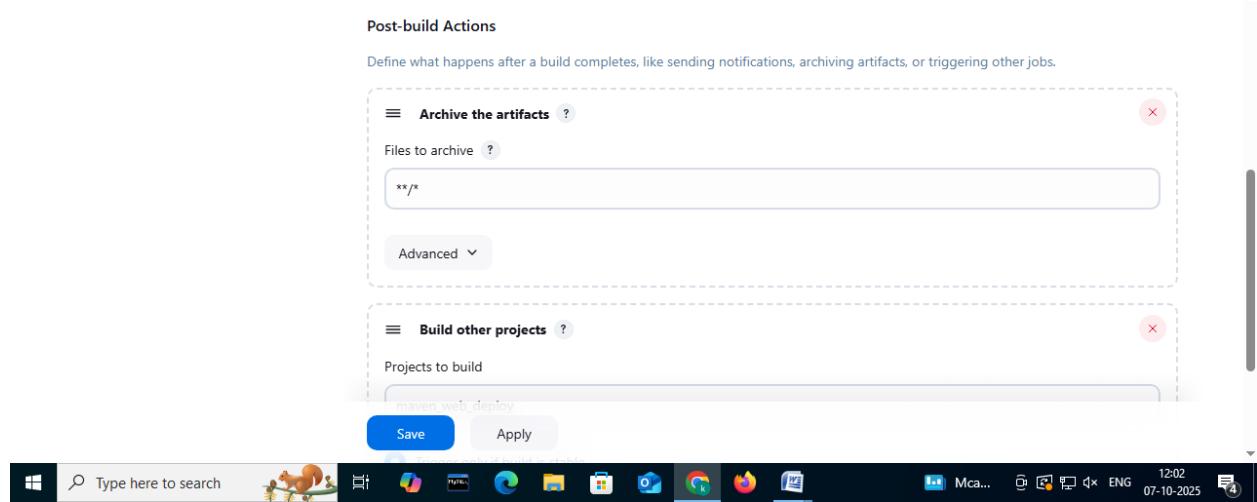
In the build steps click on add build step > select copy artifacts from another project > give project name as maven\_web\_build > give artifacts to copy as \*\*/\*

The screenshot shows the Jenkins configuration interface for the 'maven\_web\_test' job. The 'Build Steps' section is active, displaying a 'Copy artifacts from another project' step. The 'Project name' field is set to 'maven\_web\_build'. The 'Which build' dropdown is set to 'Latest successful build'. The 'Stable build only' checkbox is checked. The 'Artifacts to copy' field contains '\*\*/\*'. The 'Target directory' field is empty. There are also sections for 'Optional', 'Fingerprint Artifacts', and 'Include Build Number', with 'Fingerprint Artifacts' checked. At the bottom are 'Save' and 'Apply' buttons.

In the build steps click on add build step > give maven version as MAVEN\_HOME > select invoke top-level maven targets > goals as test

The screenshot shows the Jenkins configuration interface for the 'maven\_web\_test' job. The 'Build Steps' section is active, displaying an 'Invoke top-level Maven targets' step. The 'Maven Version' dropdown is set to 'MAVEN\_HOME'. The 'Goals' dropdown is set to 'test'. An 'Advanced' button is visible. At the bottom is an 'Add build step' button.

In the post build actions > click on add post build action > select the archive the artifacts > in the file to archive give \*\*/\*



In the post build actions > click on add post build action >select build other projects > give name as maven\_web\_deploy> select “trigger only if build is stable”



If the build is success:

The screenshot shows the Jenkins web interface for the job 'maven\_web\_test'. The status is green with a checkmark, indicating success. The build number is #4, and it was run 1 min 30 sec ago. The build name is 'test demo'. The interface includes sections for Upstream Projects (maven\_web\_build) and Downstream Projects (maven\_web\_deploy). A sidebar on the left lists options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. A 'Builds' section on the left shows a history of builds from #1 to #4. At the bottom right, there are links for REST API and Jenkins 2.489.

Status: **maven\_web\_test** (green checkmark)

test demo

Last Successful Artifacts: [maven\\_web\\_build](#)

Upstream Projects: [maven\\_web\\_build](#)

Downstream Projects: [maven\\_web\\_deploy](#)

Builds:

- Last build (#4), 1 min 30 sec ago
- Last stable build (#4), 1 min 30 sec ago
- Last successful build (#4), 1 min 30 sec ago
- Last completed build (#4), 1 min 30 sec ago

Permalinks:

- [Last build \(#4\)](#)
- [Last stable build \(#4\)](#)
- [Last successful build \(#4\)](#)
- [Last completed build \(#4\)](#)

REST API Jenkins 2.489



Create Freestyle Project (e.g., MavenWeb\_Deploy):

### Step 1: Open Jenkins (localhost:8888)

Click on "New Item" (left side menu) and name it as maven\_web\_deploy > select freestyle project > click on "OK"

New Item

Enter an item name

maven\_web\_deploy

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.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

## Step 2: Configuration of maven\_web\_deploy project

Give the description

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_deploy". The "General" tab is selected. The "Description" field contains the text "deploy demo". The "Enabled" switch is turned on. On the left sidebar, there are tabs for General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. Under "Build Steps", several checkboxes are listed: Discard old builds, GitHub project, Permission to Copy Artifact, This project is parameterized, Throttle builds, and Execute concurrent builds if necessary. Below the "General" section is a "Source Code Management" section with a note to connect and manage code repositories. At the bottom of the configuration page are "Save" and "Apply" buttons. The browser address bar shows "localhost:8888/job/maven\_web\_deploy/configure". The operating system taskbar at the bottom includes icons for File Explorer, Paint, File History, Internet Explorer, File Cabinet, Photos, Camera, Google Chrome, Mozilla Firefox, Microsoft Edge, Task View, Mail, and System. The date and time are shown as 07-10-2025 12:04.

In the source code management select none and in environment select “delete workspace before build starts”

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_deploy". The left sidebar lists "General", "Source Code Management", "Triggers", "Environment", "Build Steps", and "Post-build Actions". The "Source Code Management" tab is selected, showing options for "None" or "Git". The "Triggers" section contains several checkboxes for remote triggers, scheduled builds, and GitHub hooks. The "Environment" section has a checked checkbox for "Delete workspace before build starts" and an "Advanced" dropdown. The bottom of the screen shows a Windows taskbar with various icons and system status.

Configure

Source Code Management

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

None

Git ?

Triggers

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

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

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

Delete workspace before build starts

Advanced

Use secret text(s) or file(s) ?

Provide Configuration files ?

Add timestamps to the Console Output

Inspect build log for published build scans

Save Apply

Type here to search

NIFTY 12:07 07-10-2025

In the build steps click on add build step > select copy artifacts from another project > give project name as maven\_web\_test > give artifacts to copy as \*\*/\*

The screenshot shows the Jenkins configuration interface for a job named 'maven\_web\_deploy'. The 'Build Steps' section is active, displaying a 'Copy artifacts from another project' step. The 'Project name' field is set to 'maven\_web\_test'. The 'Which build' dropdown is set to 'Latest successful build', and the 'Stable build only' checkbox is checked. The 'Artifacts to copy' field contains '\*\*/\*'. The 'Target directory' and 'Parameter filters' fields are empty. At the bottom, there are checkboxes for 'Flatten directories', 'Optional', 'Fingerprint Artifacts' (which is checked), and 'Include Build Number'. Below these are 'Save' and 'Apply' buttons.

In the post build actions > click on add post build actions > select deploy war/ear to a container > enter war/ear files as \*\*/\*.war > context path as webpath > give the credentials and tomcat URL

The screenshot shows the Jenkins configuration interface for a job named "maven\_web\_deploy". The left sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The "Post-build Actions" section is currently selected and highlighted with a grey background. Under "Post-build Actions", there is a sub-section titled "Deploy war/ear to a container". This section contains fields for "WAR/EAR files" (set to "\*\*/\*.war") and "Context path" (set to "webpath"). Below this, there is a "Containers" section for "Tomcat 9.x Remote". It includes a "Credentials" dropdown containing "admin/\*\*\*\*\*" and a "Tomcat URL" field set to "https://localhost:8080/". At the bottom of the configuration area are "Save" and "Apply" buttons.



If the build is success:

The screenshot shows a Windows desktop environment with a Jenkins job status page open in a browser window. The browser tabs include 'maven\_web\_deploy [Jenkins]', 'Edigirala-Neksha/se-lab-intern...', 'Apache Tomcat/9.0.98', and 'Jenkins support for Java 21'. The Jenkins page for 'maven\_web\_deploy' shows a green checkmark icon and the text 'Status maven\_web\_deploy'. Below it, 'Changes' and 'Workspace' links are visible. A 'Build Now' button is present. The 'Upstream Projects' section lists 'maven\_web\_test' with a green checkmark icon. The 'Permalinks' section provides links to various build logs. The 'Builds' section displays a list of builds from today, with the most recent one (#13) marked as successful (green checkmark). The desktop taskbar at the bottom shows icons for File Explorer, Edge, Mail, OneDrive, and others, along with system status indicators like battery level, temperature (30°C), and date/time (07-10-2025).

## Create Pipeline View for MavenWeb

Click "+" beside "All" on the dashboard and Enter name as maven\_web\_pipeline

Select type as build pipeline view

The screenshot shows the Jenkins interface for creating a new view. The title bar says 'New view [Jenkins]'. The main area is titled 'New view' with a sub-section 'Build Pipeline View'. The 'Name' field contains 'maven\_web\_pipeline'. The 'Type' section has three options: 'Build Pipeline View' (selected), 'List View', and 'My View'. A 'Create' button is at the bottom. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', etc., and sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2).

Give the description and in the upstream directly the maven\_web\_build will be shown

Dashboard > maven\_web\_pipeline > Edit View

**Edit View**

Name: maven\_web\_pipeline

Description: Describe the purpose of this view.

Plain text: Preview

Build Pipeline View Title:

**Pipeline Flow**

Layout: Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

**Upstream / downstream config**

Select initial Job: maven\_web\_build

**Trigger Options**

Save Apply

Click on apply and save

Dashboard > maven\_web\_pipeline > Edit View

**Column Headers**

No header

Do not show any column headers

Refresh frequency (in seconds): 3

URL for custom CSS files:

Console Output Link Style

Lightbox

**Widgets**

Customize the widgets that show in this view.

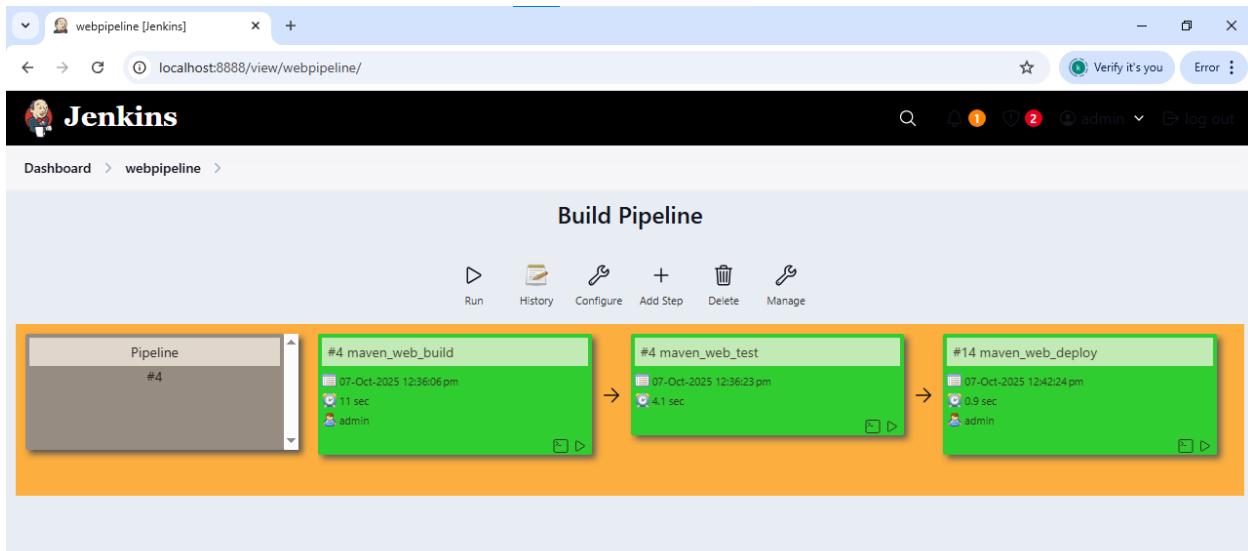
Filter build queue  
If checked, only jobs in this view will be shown in the queue.

Filter build executors  
If checked, only those build executors will be shown that could execute the jobs in this view.

Save Apply

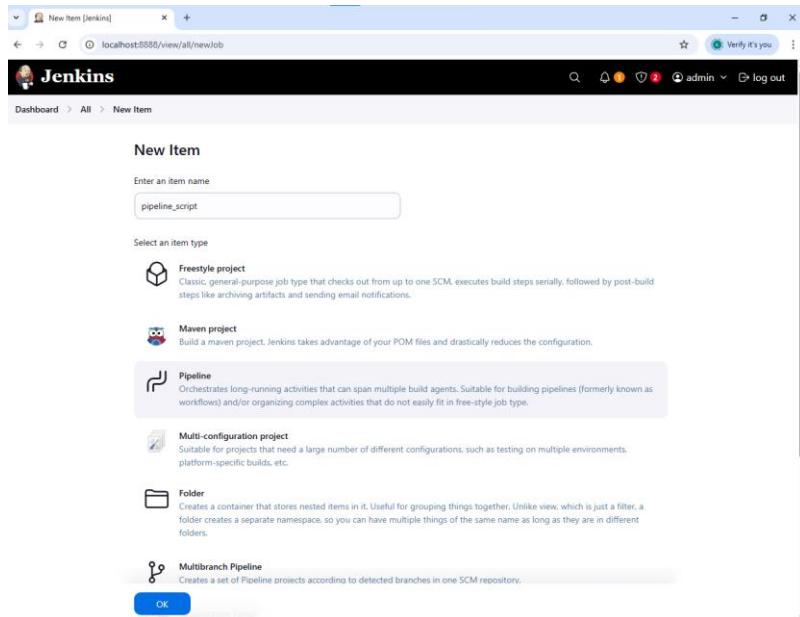
REST API Jenkins 2.489

In the stage view it we be shown as:

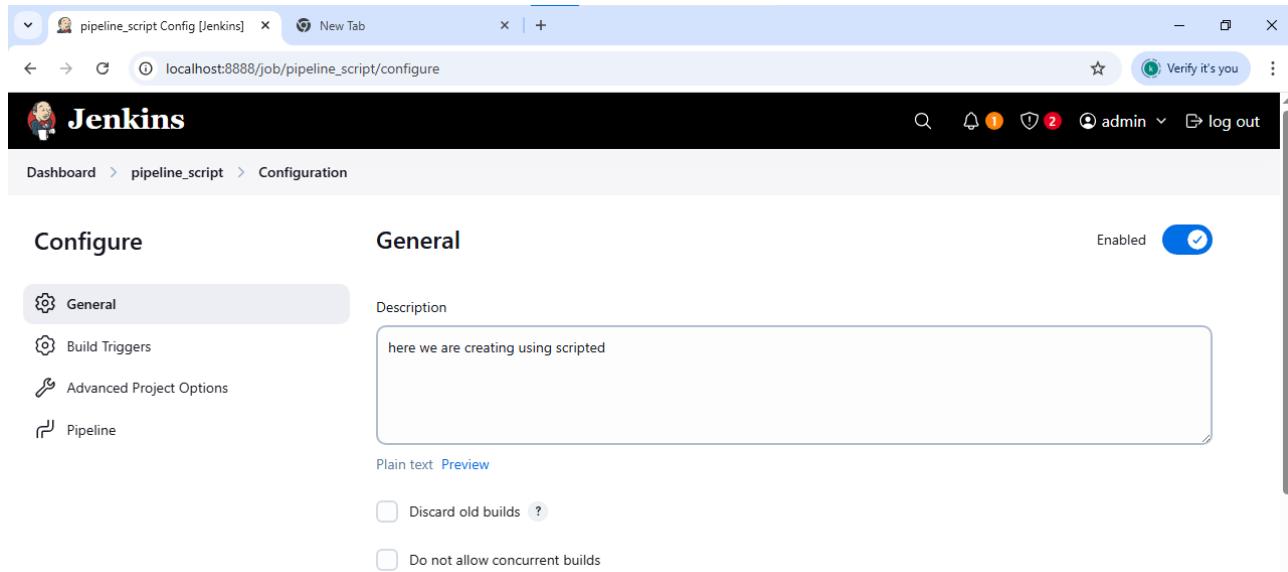


## **9.Pipeline Creation using script**

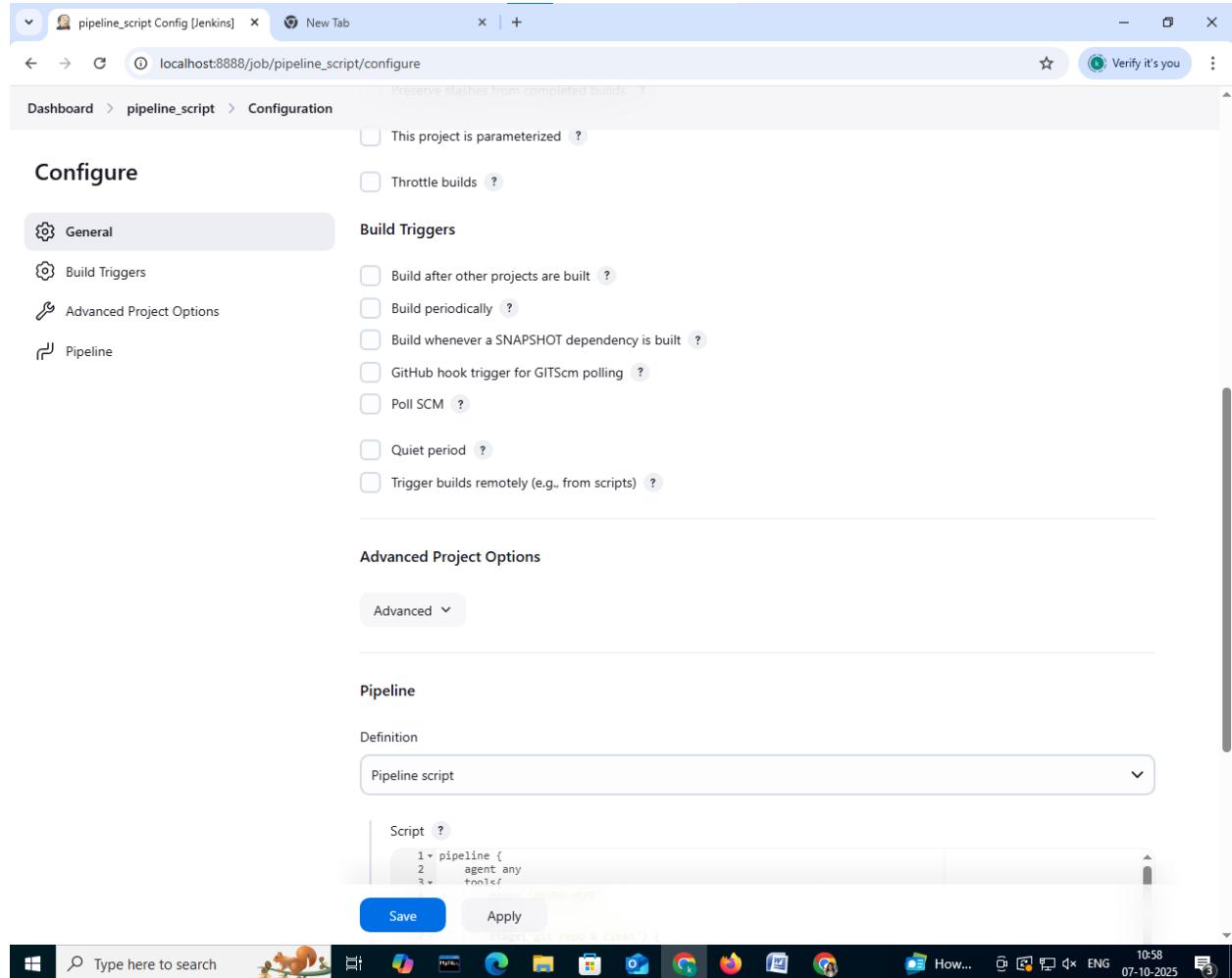
Step 1: In the Jenkins select the new item and give the name as pipeline\_script and select pipeline and click ok



Step 2: In the configuration, give the description



Step 3: In the pipeline section give definition as pipeline script and enter the script with git reop link and project name



The screenshot shows the Jenkins Pipeline configuration page for a job named "pipeline\_script". The "General" tab is selected. Under "Build Triggers", several options are listed: "Build after other projects are built", "Build periodically", "Build whenever a SNAPSHOT dependency is built", "GitHub hook trigger for GITScm polling", "Poll SCM", "Quiet period", and "Trigger builds remotely (e.g., from scripts)". The "Advanced Project Options" section is collapsed. The "Pipeline" section is expanded, showing the "Definition" field set to "Pipeline script". Below it is a code editor containing the following Groovy script:

```
1 > pipeline {
2   agent any
3   tools</pre>
```

At the bottom of the pipeline editor are "Save" and "Apply" buttons.

Step 4: click on apply and then save

The screenshot shows the Jenkins Pipeline configuration page for a job named "pipeline\_script". The "Advanced Project Options" tab is selected. The "Pipeline" section is set to "Definition" and "Pipeline script". The script content is as follows:

```
1 > pipeline {
2   agent any
3   tools{
4     maven 'MAVEN-HOME'
5   }
6   stages {
7     stage('git repo & clean') {
8       steps {
9         //bat "rmdir /s /q mavenjava"
10        bat "git clone https://github.com/SarvikaSomishetty/eclipse-maven-projects.git"
11        bat "mvn clean -f eclipse-maven-projects"
12      }
13    }
14    stage('install') {
15      steps {
16        bat "mvn install -f eclipse-maven-projects"
17      }
18  }
```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom of the page are two buttons: "Save" and "Apply".

Step 8: Check the stage view. If is successful.

The screenshot shows the Jenkins interface for the 'pipeline\_script' job. The top navigation bar includes the Jenkins logo, user information ('Verify it's you'), and links for search, notifications, security, and log out. The main content area displays the 'Status' of the job, which is green with a checkmark icon. A note says 'here we are creating using scripted'. Below this is the 'Stage View' section, which provides a breakdown of the build time by stage: Declarative: Tool Install (296ms), git repo & clean (5s), install (9s), test (3s), and package (4s). The 'Average stage times: (full run time: ~26s)' is also indicated. The 'Builds' section shows one build (#2) from Oct 07 at 11:02 AM, with a note 'No Changes'. The 'Permalinks' section lists four build links. On the left sidebar, there are links for Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax.

Declarative: Tool Install	git repo & clean	install	test	package		
296ms	5s	9s	3s	4s		
Oct 07 11:02	No Changes	296ms	5s	9s	3s	4s

**Builds**

- Oct 07 11:02 AM #2

**Permalinks**

- Last build (#2), 4 min 29 sec ago
- Last stable build (#2), 4 min 29 sec ago
- Last successful build (#2), 4 min 29 sec ago
- Last completed build (#2), 4 min 29 sec ago

## **10. Kubernetes Using Minikube:**

### **Step -1:**

#### **Start Minikube : Command- minikube start**

- First, you need to start your Kubernetes cluster using Minikube.
- When you start it, Minikube sets up a lightweight virtual machine on your system and runs a local Kubernetes node inside it.

#### **Step-2:Then check for the status Minikube status**

#### **Step-3:Create an image**

```
PS C:\Users\User>
PS C:\Users\User> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\Users\User> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Users\User> kubectl expose deployment mynginx --type=NodePort --port=80
service/mynginx exposed
PS C:\Users\User> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
mynginx-79bb8756c7-wpslj   1/1     Running   0          34s
```

#### **Step-4: Check the NGINX Service Details**

- After creating the service, check its details to see which port Kubernetes assigned to it.

```
  DownwardAPI:           true
QoS Class:             BestEffort
Node-Selectors:         <none>
Tolerations:           node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                       node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From           Message
  ----  -----   ----  ----
  Normal  Scheduled  68s  default-scheduler  Successfully assigned default/mynginx-79bb8756c7-wpslj to minikube
  Normal  Pulling   67s  kubelet        Pulling image "nginx"
  Normal  Pulled   65s  kubelet        Successfully pulled image "nginx" in 2.416s (2.416s including waiting). Image size: 159974475 bytes.
  Normal  Created   65s  kubelet        Created container nginx
  Normal  Started   64s  kubelet        Started container nginx
PS C:\Users\User> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Users\User> kubectl get service mynginx
Error from server (NotFound): services "mynginx" not found
PS C:\Users\User> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::]:8081 -> 80
```

### Step-5:check the detail of the kubectl .

```
PS C:\Users\User> kubectl describe pods
Name:           mynginx-79bb8756c7-wpslj
Namespace:      default
Priority:      0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Tue, 14 Oct 2025 12:38:19 +0530
Labels:        app=mynginx
               pod-template-hash=79bb8756c7
Annotations:   <none>
Status:        Running
IP:            10.244.0.16
IPs:
  IP:          10.244.0.16
Controlled By: ReplicaSet/mynginx-79bb8756c7
Containers:
  nginx:
    Container ID:  docker://675066efbd98a54ba39177103943b196de2c61f01d820ede859b48578f3e245e
    Image:         nginx
    Image ID:     docker-pullable://nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
    Port:          <none>
    Host Port:    <none>
    State:        Running
      Started:   Tue, 14 Oct 2025 12:38:22 +0530
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nh2rw (ro)
Conditions:
  Type          Status
  PodReadyToStartContainers  True
  Initialized    True
  Ready          True
  ContainersReady  True
  PodScheduled   True
Volumes:
  kube-api-access-nh2rw:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
  QoS Class:      BestEffort
  Node-Selectors:  <none>
  Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
```

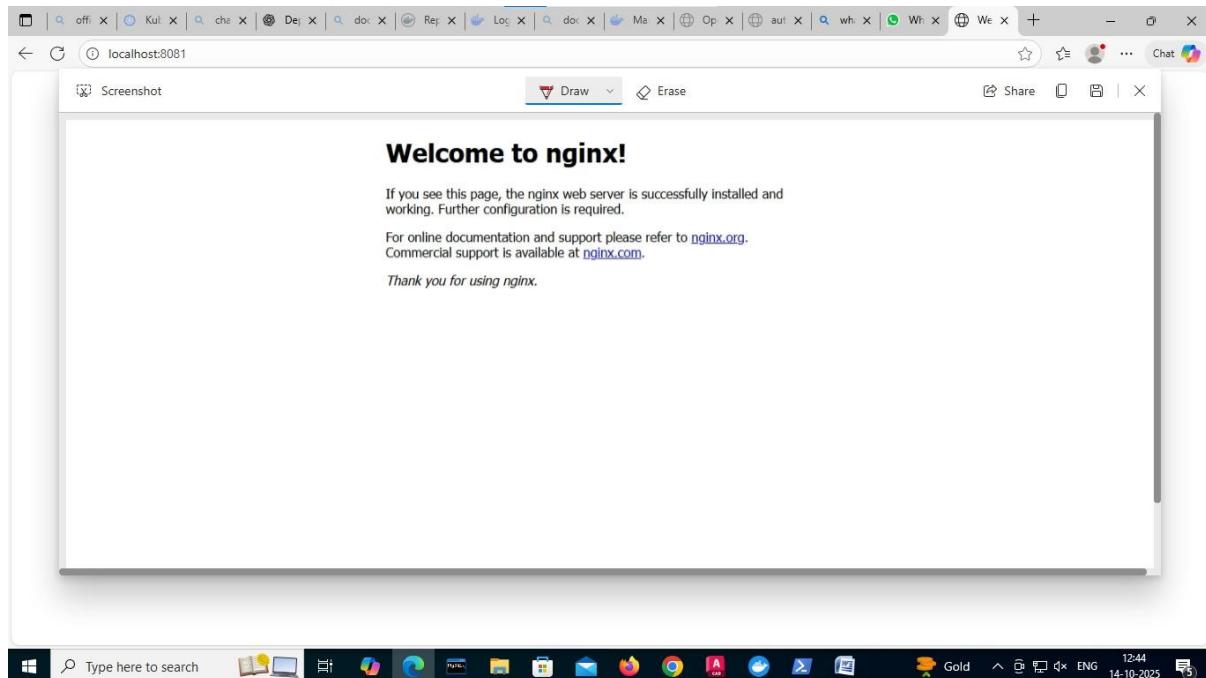
### Step-6:Check the NGINX Service Details

- After creating the service, check its details to see which port Kubernetes assigned to it.

```
PS C:\Users\User> kubectl port-forward svc/mynginx 8081:80
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
Handling connection for 8081
```

## Step-7: Open NGINX in the Browser

- Now that your service is exposed, you can open NGINX in your browser.



## 11. Jenkins-CI/CD

### Setting Up Jenkins CI-----using GitHub Webhook with Jenkins

Step 1: Take the authentication key from the ngrok and setup in ngrok terminal

```
tcp          start a TCP tunnel
tls          start a TLS endpoint
update      update ngrok to the latest version
version     print the version string

EXAMPLES:
# forward http traffic from assigned public URL to local port 80
ngrok http 80
# port 8080 available at baz.ngrok.dev
ngrok http --url baz.ngrok.dev 8080
# tunnel arbitrary TCP traffic to port 22
ngrok tcp 22
# secure your app with oauth
ngrok http 80 --oauth=google --oauth-allow-email=foo@foo.com

Paid Features:
ngrok http 80 --url mydomain.com                               # run ngrok with your own custom domain
ngrok http 80 --cidr-allow 2600:8c00::a03c:91ee:fe69:9695/32 # run ngrok with IP policy restrictions
Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Upgrade your account at https://dashboard.ngrok.com/billing/subscription to access paid features

Flags:
-h, --help      help for ngrok

Use "ngrok [command] --help" for more information about a command.

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Windows\System32>ngrok config add-authtoken 34gKWhQDcoITj34K6eN73XoYG6J_58fBgmpjM5ikZVdKVdyCe|
```

Step-2: Execute the following command using the port number on which Jenkins is running

```
C:\Windows\System32>ngrok.exe http 8888
```

- Following output will be given:

```
ngrok                                         (Ctrl+C to quit)

♦ Block threats before they reach your services with new WAF actions → https://ngrok.com/r/waf

Session Status        online
Account              Neksha Edigirala (Plan: Free)
Update               update available (version 3.32.0, Ctrl-U to update)
Version              3.24.0-msix
Region               India (in)
Latency              147ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://corkier-darla-handsome.ngrok-free.dev -> http://localhost:8888

Connections          ttl     opn     rt1     rt5     p50     p90
                     2       0       0.00    0.00   30.28   30.47

HTTP Requests
-----
11:35:59.377 IST POST /github-webhook/          200 OK
11:34:29.479 IST POST /github-webhook/          200 OK
```

Go to Jenkins:

Step-3: Create the Jenkins job in the source code management select the git and enter git repo url and make sure the branch is same (i.e., main)

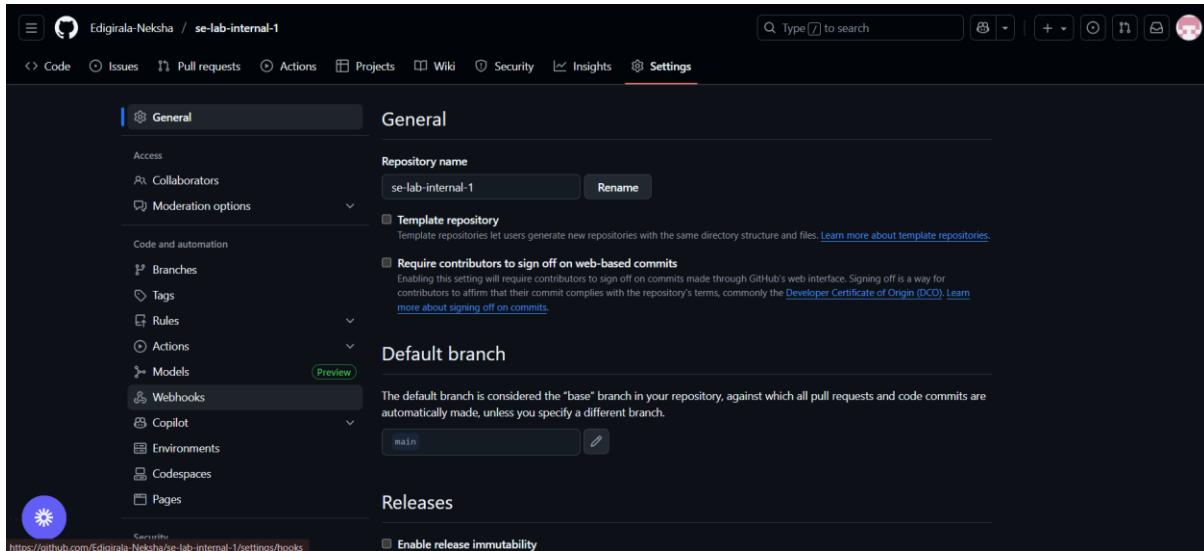
The screenshot shows the Jenkins job configuration page for 'job\_webhook\_java'. The 'Source Code Management' section is active, with 'Git' selected. The 'Repository URL' field contains 'https://github.com/Edigirala-Neksha/se-lab-internal-1.git'. The 'Branch Specifier' field contains '/main'. Other sections like General, Triggers, Environment, Build Steps, and Post-build Actions are visible but not selected.

Step-4: In the triggers section select “Github hook trigger for GITScm polling”

The screenshot shows the Jenkins job configuration page for 'job\_webhook\_java'. The 'Triggers' section is active. The 'GitHub hook trigger for GITScm polling' checkbox is checked. Other trigger options like 'Trigger builds remotely' and 'Build periodically' are unchecked. The 'Save' and 'Apply' buttons are at the bottom.

Click on apply and save

Step-6: open the git hub repo open setting of repo and then go to webhooks

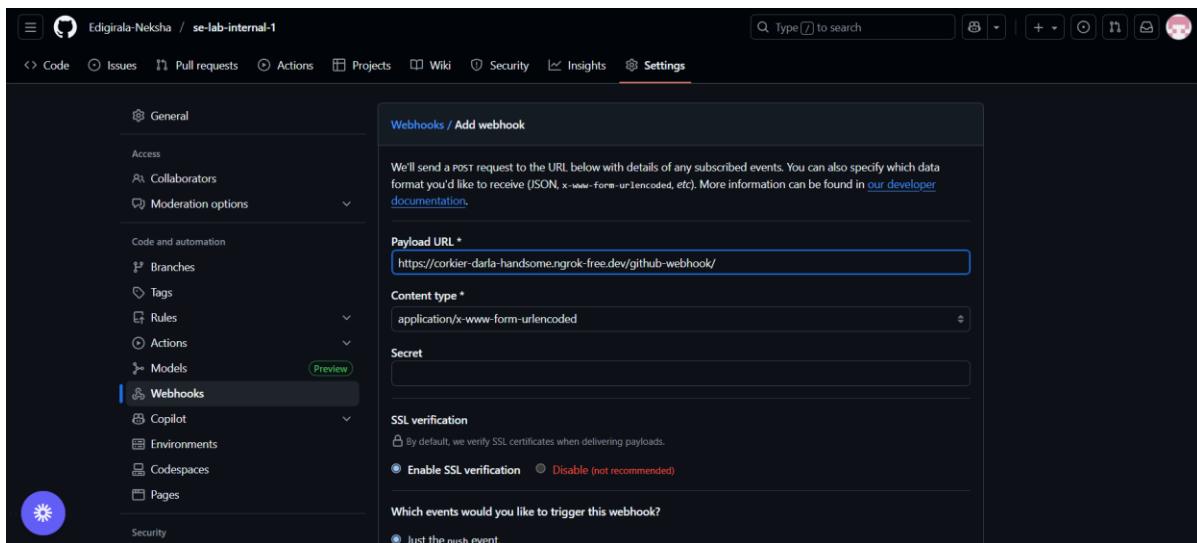


The screenshot shows the GitHub repository settings page for 'se-lab-internal-1'. The 'General' tab is selected. On the left sidebar, 'Webhooks' is highlighted. The main area displays the 'Default branch' section, which specifies 'main' as the default branch. Below it, there's a 'Releases' section with an 'Enable release immutability' checkbox. At the bottom left, there's a 'Copy URL' button and a URL link.

Step-7: Click on add a webhook and take the forwarding URL from ngrok and paste in payload URL and add /github-webhook/ along with the forwarding url

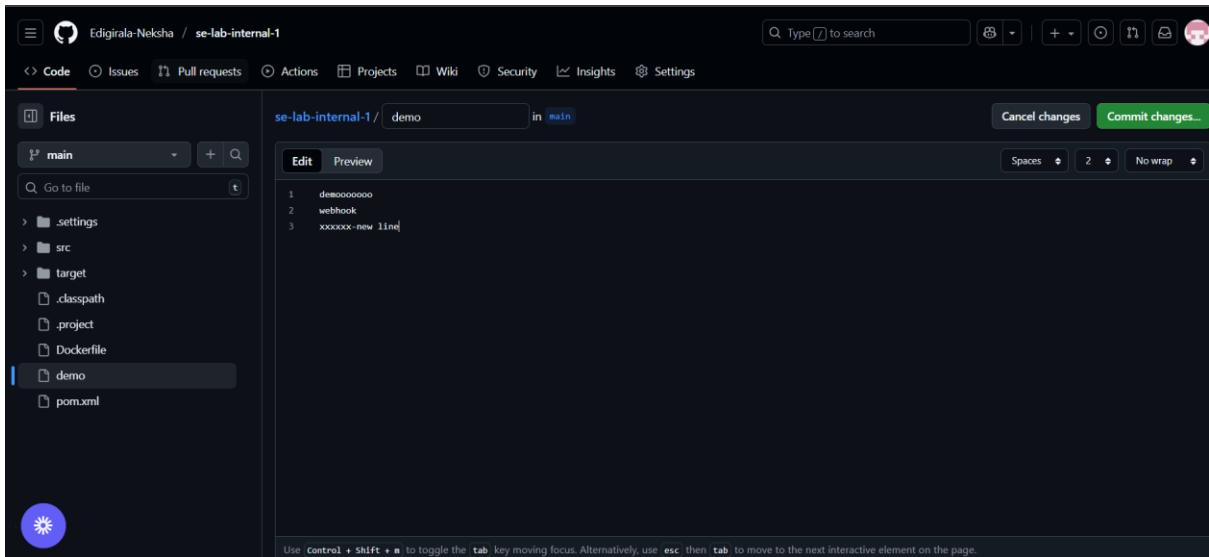
Forwarding URL: <https://corkier-darla-handsome.ngrok-free.dev>

Payload url: <https://corkier-darla-handsome.ngrok-free.dev/github-webhook/>



The screenshot shows the 'Add webhook' form within the GitHub repository settings. The 'Webhooks / Add webhook' header is visible. The 'Payload URL \*' field contains the value 'https://corkier-darla-handsome.ngrok-free.dev/github-webhook/'. The 'Content type \*' dropdown is set to 'application/x-www-form-urlencoded'. The 'SSL verification' section has the 'Enable SSL verification' radio button selected. The 'Which events would you like to trigger this webhook?' section has the 'Just the push event.' option selected.

## Step 8: make changes in the files in github

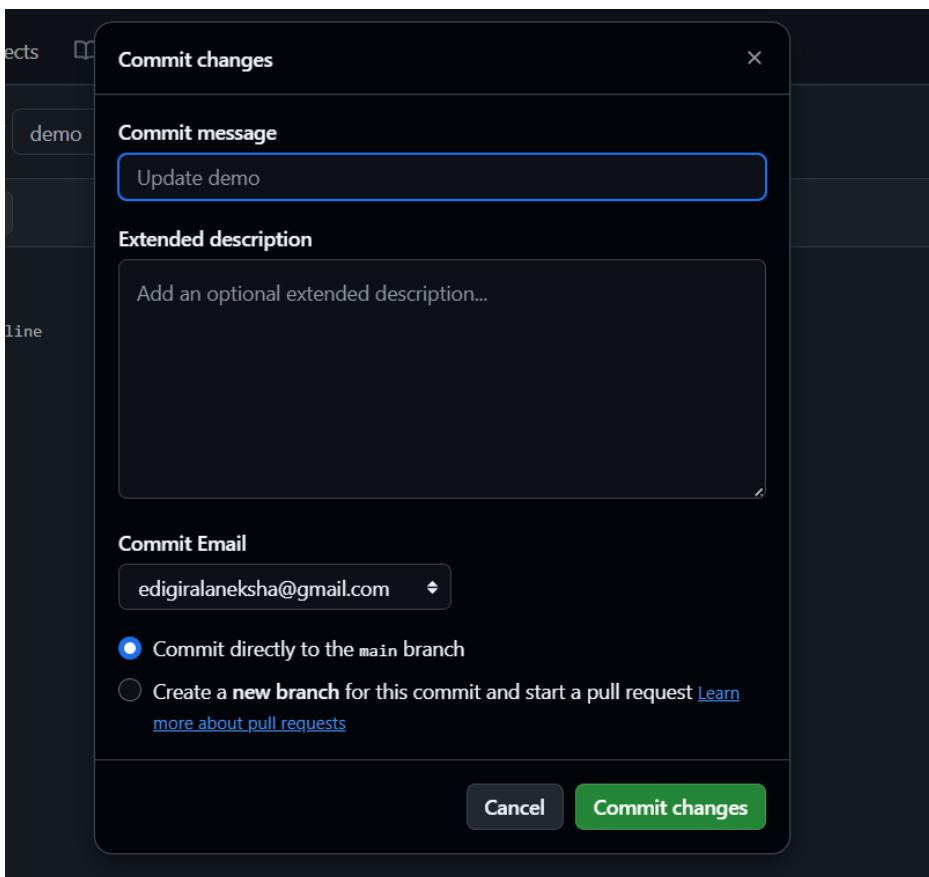


A screenshot of the GitHub interface showing a commit message being edited. The repository is "Edigirala-Neksha / se-lab-internal-1". The file "demo" is selected in the sidebar. The commit message editor shows the following content:

```
1 demooooooo
2 webhook
3 xxxxxxx-new line|
```

The "Commit changes" button is visible at the top right of the editor.

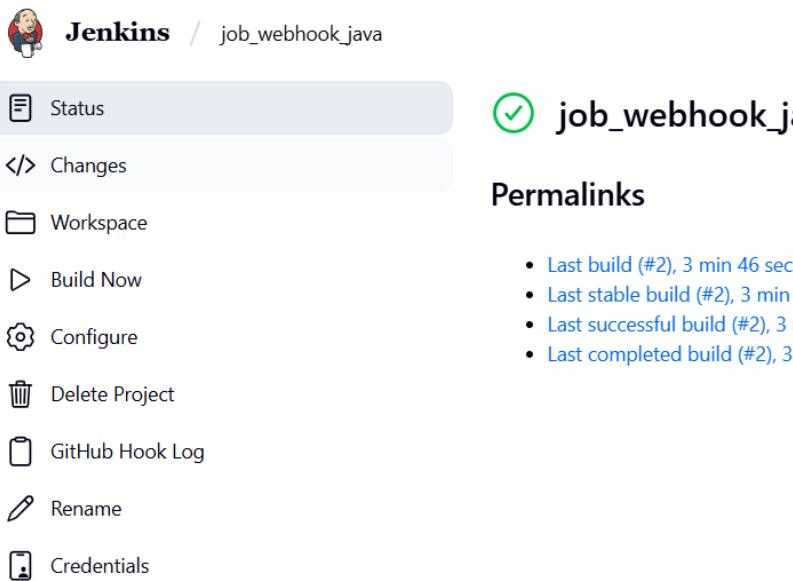
## Step 9: click on commit changes



Step 10: open Jenkins the build will start automatically

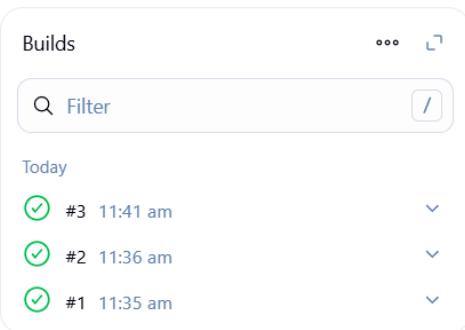
The screenshot shows the Jenkins interface for the 'job\_webhook\_java' project. The top navigation bar includes the Jenkins logo and the current path: Jenkins / job\_webhook\_java. A sidebar on the left contains links for Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, Rename, and Credentials. The main content area is titled 'job\_webhook\_java' with a green checkmark icon. Below it is a section titled 'Permalinks' with a bulleted list of four links: 'Last build (#2), 3 min 46 sec ago', 'Last stable build (#2), 3 min 46 sec ago', 'Last successful build (#2), 3 min 46 sec ago', and 'Last completed build (#2), 3 min 46 sec ago'. The bottom section, titled 'Builds', displays two entries: a 'Pending' entry for build #3, which is 'In the quiet period. Expires in 2.9 sec', and a 'Today' entry for build #2 at 11:36 am.

Build	Status	Time
#3	Pending	In the quiet period. Expires in 2.9 sec
#2	Today	11:36 am



The screenshot shows the Jenkins job\_webhook\_java dashboard. On the left is a sidebar with various project management options: Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, Rename, and Credentials. The main area displays a green checkmark icon followed by the text "job\_webhook\_java". Below this is a section titled "Permalinks" with a list of recent builds:

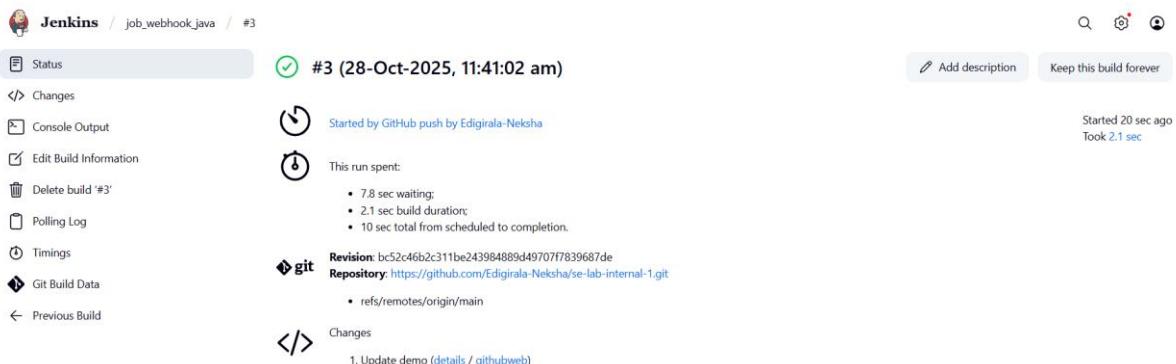
- Last build (#2), 3 min 46 sec ago
- Last stable build (#2), 3 min 46 sec ago
- Last successful build (#2), 3 min 46 sec ago
- Last completed build (#2), 3 min 46 sec ago



A detailed view of the build history for job\_webhook\_java. It shows three builds listed under the "Today" section:

- #3 11:41 am (status: green checkmark)
- #2 11:36 am (status: green checkmark)
- #1 11:35 am (status: green checkmark)

You can check status : started by git hub push



A detailed view of Jenkins build #3. The top navigation bar includes "Status", "Changes", "Console Output", "Edit Build Information", "Delete build #3", "Polling Log", "Timings", "Git Build Data", and "Previous Build". The main content area shows the build status as "Started by GitHub push by Edigirala-Neksha" at 28-Oct-2025, 11:41:02 am. It also displays the run time spent (7.8 sec waiting, 2.1 sec build duration, 10 sec total) and the git commit information (Revision: bc52c46b2c311be243984889d49707f7839687de, Repository: https://github.com/Edigirala-Neksha/se-lab-internal-1.git, ref: refs/remotes/origin/main). A note indicates "Keep this build forever".

## Setting Up Jenkins Email Notification Setup (Using Gmail with AppPassword)

### Step-1: Creation of app password

### **Gmail: Enable App Password (for 2-Step Verification)**

#### **ii. Enable 2-Step Verification**

#### **iii. Generate App Password for Jenkins**

- Go to:
  - Security → App passwords
- Select:
  - **App:** Other (Custom name)
  - **Name:** Jenkins-Demo
- Click **Generate**
- Copy the **16-digit app password**
  - Save it in a secure location (e.g., Notepad)

## **2. Jenkins Plugin Installation**

#### **i. Open Jenkins Dashboard**

#### **ii. Navigate to:**

- Manage Jenkins → Manage Plugins

#### **iii. Install Plugin:**

- Search for and install:
  - Email Extension Plugin

Jenkins / Manage Jenkins / Plugins

Plugins

Updates 16

Available plugins

Installed plugins

Advanced settings

Name

Health Enabled

Email Extension 1925.v1598902b\_56dd

This plugin is a replacement for Jenkins's email publisher. It allows to configure every aspect of email notifications: when an email is sent, who should receive it and what the email says

Report an issue with this plugin

100

Email Extension Template Plugin 233.v1eb\_88fc160b\_5

This plugin allows administrators to create global templates for the Extended Email Publisher.

Report an issue with this plugin

100

Mailer Plugin 522.va\_995fa\_cfb\_8b\_d

This plugin allows you to configure email notifications for build results.

Report an issue with this plugin

100

Pipeline

Failed to load: Pipeline (workflow-aggregator 608.v67378e9d3db\_1)  
- Failed to load: Pipeline: Basic Steps (workflow-basic-steps 1098.v808b\_fd7f8cf4)

### 3. Configure Jenkins Global Email Settings

Go to:

- Manage Jenkins → Configure System

---

#### A. E-mail Notification Section

Field	Value
SMTP Server	smtp.gmail.com
Use SMTP Auth	<input checked="" type="checkbox"/> Enabled
User Name	Your Gmail ID (e.g., archanareddykmit@gmail.com)
Password	Paste the 16-digit App Password
Use SSL	<input checked="" type="checkbox"/> Enabled
SMTP Port	465
Reply-To Address	Your Gmail ID (same as above)

#### ► Test Configuration

- Click: Test configuration by sending test e-mail
- Provide a valid email address to receive a test mail
- Should receive email from Jenkins

Jenkins / Manage Jenkins / System

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Advanced ^ Edited

Use SMTP Authentication ?

User Name  
edigiralaneksha@gmail.com

Password  
 Concealed Change Password

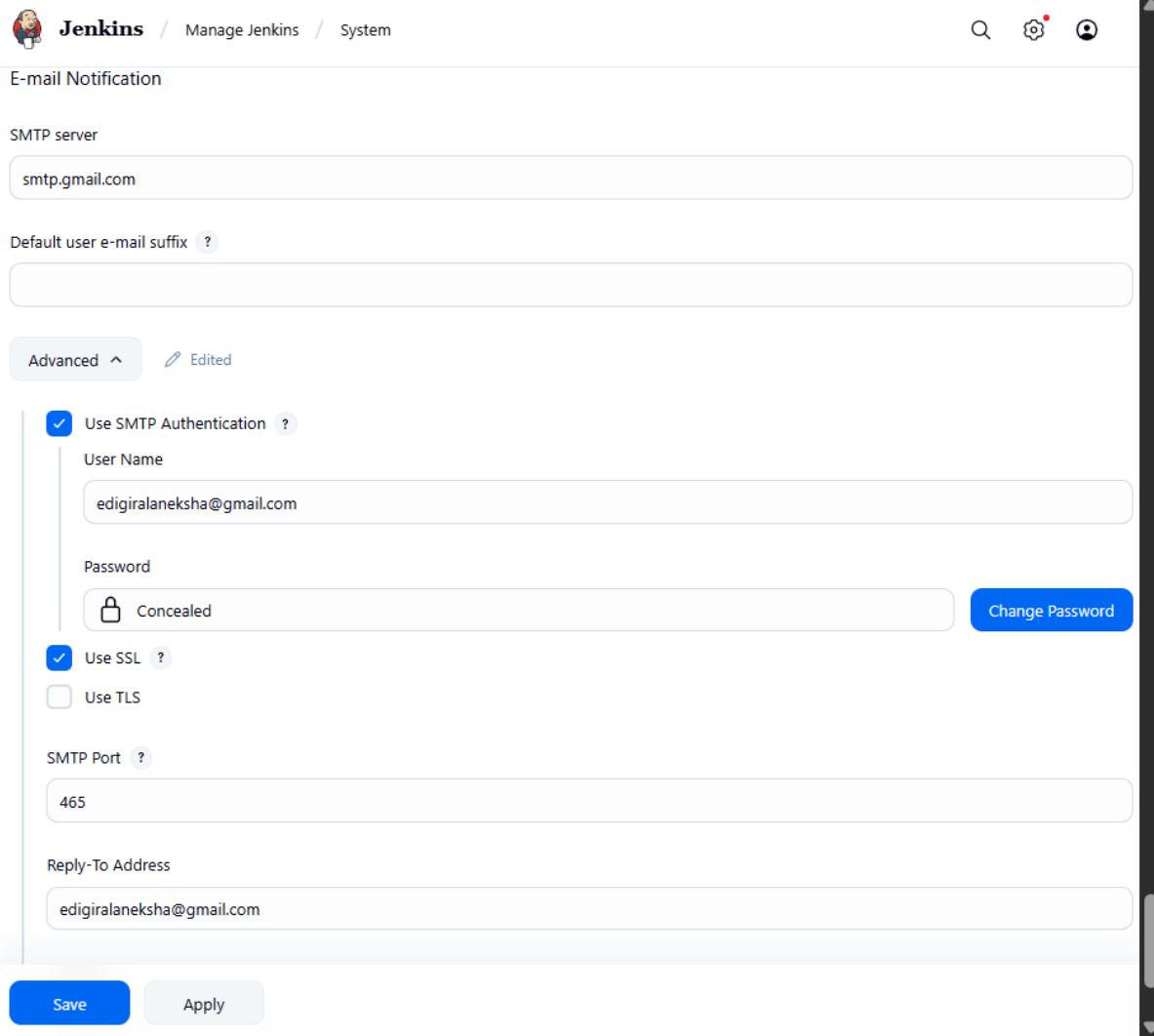
Use SSL ?

Use TLS

SMTP Port ?  
465

Reply-To Address  
edigiralaneksha@gmail.com

Save Apply



## B. Extended E-mail Notification Section

Field	Value
SMTP Server	smtp.gmail.com
SMTP Port	465
Use SSL	<input checked="" type="checkbox"/> Enabled
Credentials	Add Gmail ID and App Password as Jenkins credentials
Default Content Type	text/html or leave default

Field	Value
<b>Default Recipients</b>	Leave empty or provide default emails
<b>Triggers</b>	Select as per needs (e.g., Failure)

Extended E-mail Notification

SMTP server

SMTP Port

Advanced ^     Edited

Credentials

edigiralaneksha@gmail.com/\*\*\*\*\*\*\*\* (rst)

+ Add

Use SSL  
 Use TLS  
 Use OAuth 2.0

Advanced Email Properties

Save Apply

Default Triggers ^

Default Triggers ?

- Aborted
- Always
- Before Build
- Failure - 1st
- Failure - 2nd
- Failure - Any
- Failure - Still
- Failure - X
- Failure -> Unstable (Test Failures)
- Fixed
- Not Built
- Script - After Build
- Script - Before Build
- Status Changed
- Success
- Test Improvement
- Test Regression
- Unstable (Test Failures)
- Unstable (Test Failures) - 1st
- Unstable (Test Failures) - Still
- Unstable (Test Failures)/Failure -> Success

Content Token Reference ?

#### 4. Configure Email Notifications for a Jenkins Job

##### i. Go to:

- Jenkins → Select a Job → Configure

The screenshot shows the Jenkins configuration interface for a job named 'job\_webhook\_java'. The 'General' tab is selected. The 'Description' field contains 'java webhook'. The 'Advanced' dropdown is open, showing various build options like 'Discard old builds', 'GitHub project', and 'Notify when Job configuration changes'. The 'Source Code Management' section is expanded, showing 'Git' selected as the provider with a repository link. At the bottom are 'Save' and 'Apply' buttons.

Configure

General

Description

java webhook

Plain text Preview

Advanced

Source Code Management

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

None

Git

Repositories

Save Apply

##### ii. In the Post-build Actions section:

- Click: Add post-build action → **Editable Email Notification**

##### A. Fill in the fields:

Field	Value
<b>Project Recipient List</b>	Add recipient email addresses (comma-separated)
<b>Content Type</b>	Default (text/plain) or text/html
<b>Triggers</b>	Select events (e.g., Failure, Success, etc.)
<b>Attachments</b>	(Optional) Add logs, reports, etc.

### iii. Click Save

#### Post-build Actions

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

≡ **Editable Email Notification** ? ✖

Allows the user to disable the publisher, while maintaining the settings

Disable Extended Email Publisher ?

**Project From**

Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

edigiralaneksha@gmail.com,nekshasri99@gmail.com

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

\$DEFAULT\_REPLYTO

**Save** **Apply**

Jenkins / job\_webhook\_java

Rename Credentials

Builds

Filter

Today

- #4 11:45 am
- #3 11:41 am
- #2 11:36 am
- #1 11:35 am

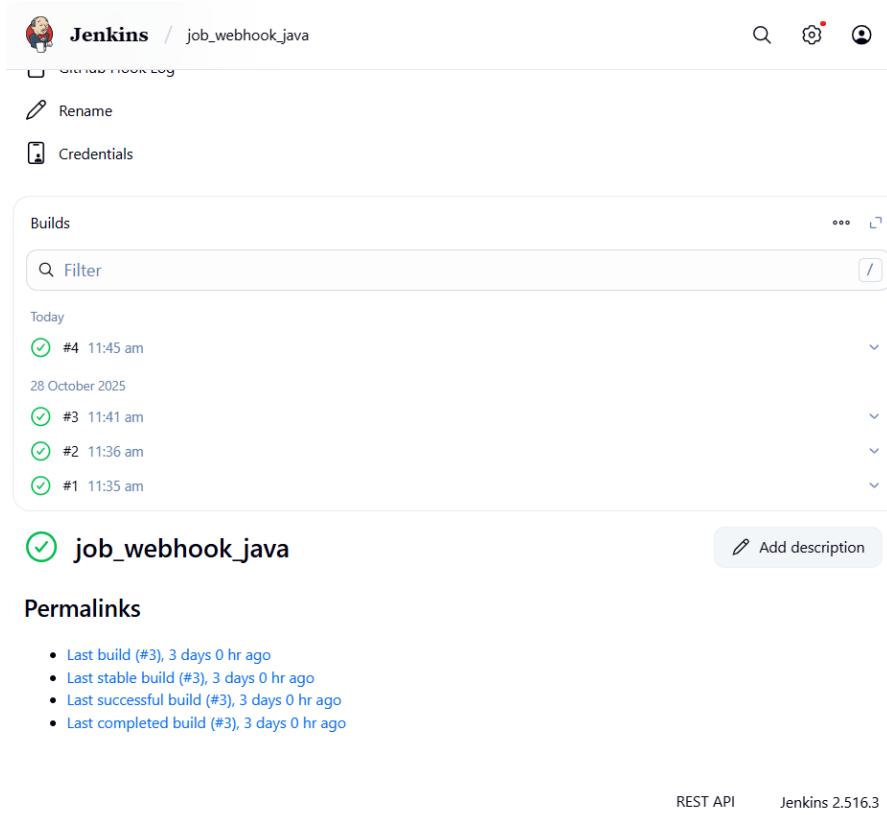
Add description

job\_webhook\_java

Permalinks

- Last build (#3), 3 days 0 hr ago
- Last stable build (#3), 3 days 0 hr ago
- Last successful build (#3), 3 days 0 hr ago
- Last completed build (#3), 3 days 0 hr ago

REST API Jenkins 2.516.3



Gmail Search mail

Compose

Inbox 84

- Starred
- Snoozed
- Sent
- Drafts 1
- Purchases 13
- More

Labels +

job\_webhook\_java - Build # 4 - Successful!

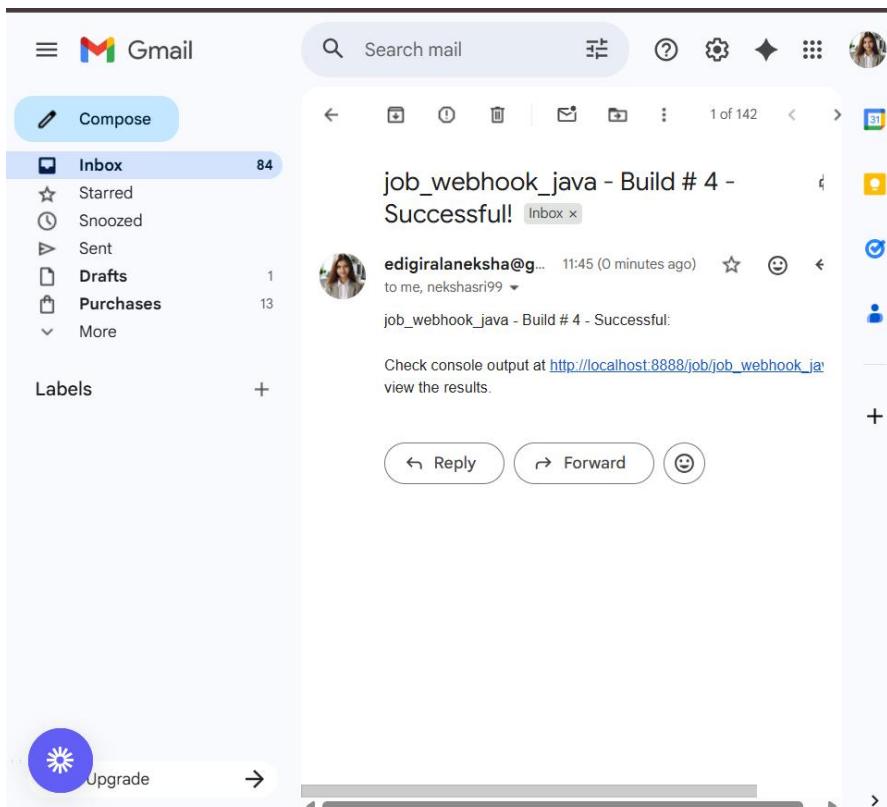
edigiralaneksha@g... 11:45 (0 minutes ago) to me, nekshasri99

job\_webhook\_java - Build # 4 - Successful!

Check console output at [http://localhost:8888/job/job\\_webhook\\_java](http://localhost:8888/job/job_webhook_java) view the results.

Reply Forward

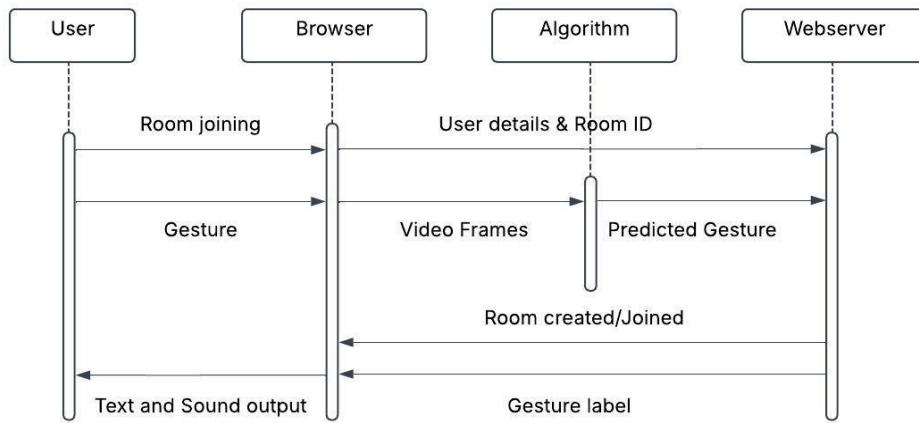
Upgrade



## TUNEORA – A Music Web App

### 1. Sequence Diagram:

A sequence diagram shows how objects interact in a particular scenario of a use case.  
It focuses on the time order of messages exchanged between different components in a system.



### 2. Class Diagram:

A class diagram represents the static structure of a system by showing classes, their attributes, methods, and relationships.

It is mainly used for object-oriented design and modeling data structures.

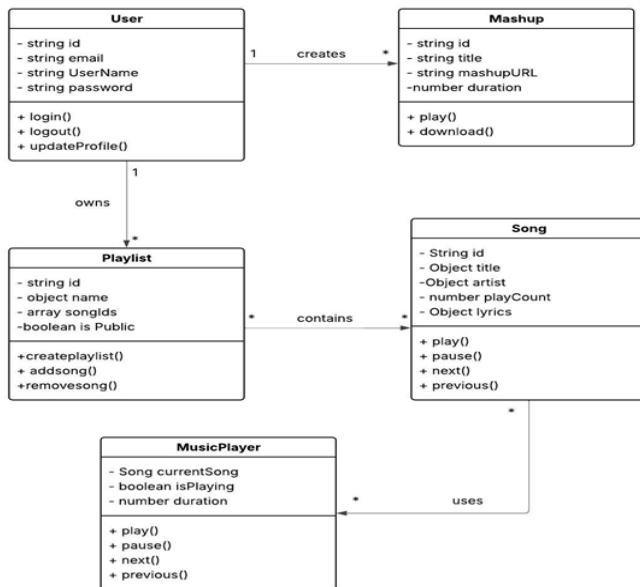
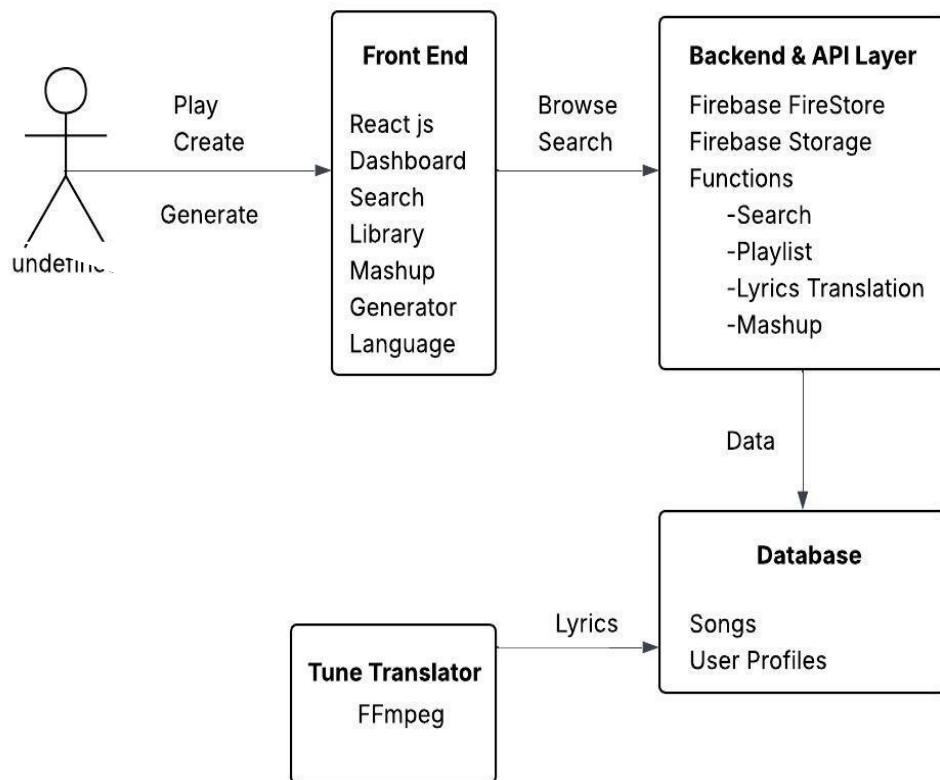


Fig 5: Class Diagram for TuneOra

### 3. Component Diagram:

A component diagram illustrates how different software components are connected and interact to form a complete system.

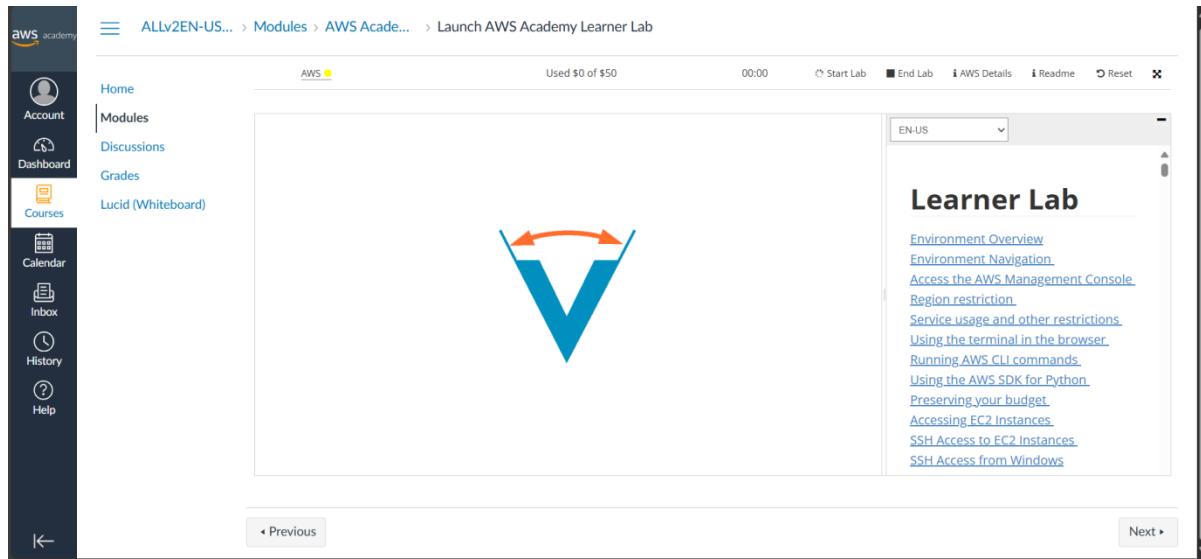
It helps visualize the organization and dependencies among modules or subsystems.



## **12.Creation of virtual machine for Ubuntu OS and Deploying the web application**

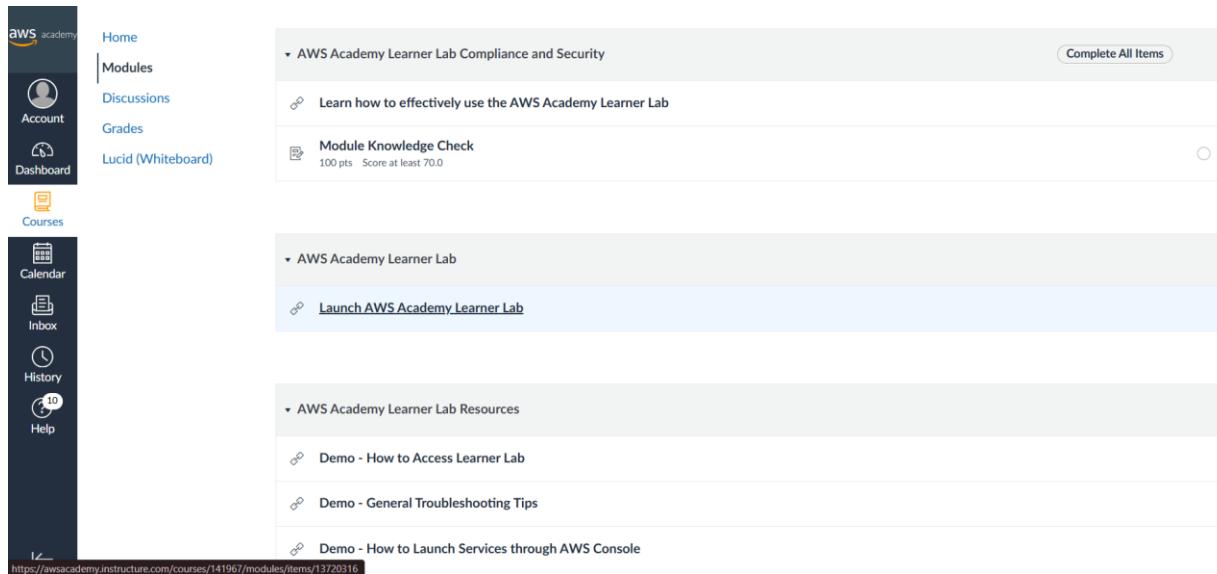
DEPLOYMENT OF INDEX.HTML USING EC2 INSTANCE in AWS

Step 1: Click on Modules.



The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a breadcrumb navigation: ALLv2EN-US... > Modules > AWS Academ... > Launch AWS Academy Learner Lab. Below the breadcrumb is a navigation bar with Home, Modules (which is selected), Discussions, Grades, and Lucid (Whiteboard). A large blue 'V' icon with a red double-headed arrow is centered in the main content area. To the right is a 'Learner Lab' panel with a dropdown menu set to EN-US. The panel contains links for Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, Preserving your budget, Accessing EC2 Instances, SSH Access to EC2 Instances, and SSH Access from Windows. At the bottom are 'Previous' and 'Next' buttons.

Step 2: Scroll down and select Lunch AWS Academy Lab



The screenshot shows the AWS Academy Learner Lab interface with the 'Modules' section selected in the sidebar. The main content area displays three sections: 'AWS Academy Learner Lab Compliance and Security' (with a 'Complete All Items' button), 'AWS Academy Learner Lab' (with a 'Launch AWS Academy Learner Lab' link), and 'AWS Academy Learner Lab Resources' (with links for 'Demo - How to Access Learner Lab', 'Demo - General Troubleshooting Tips', and 'Demo - How to Launch Services through AWS Console'). The URL at the bottom is https://awsacademy.instructure.com/courses/141967/modules/items/13720316.

### Step 3: click on start lab

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with various navigation links: Home, Modules (selected), Discussions, Grades, Lucid (Whiteboard), Courses, Calendar, Inbox, History, and Help. The main area has a title bar: ALLv2EN-US-LTI13-141967 > Modules > AWS Academy Learner Lab > Launch AWS Academy Learner Lab. Below the title bar, there are buttons for Start Lab, End Lab, AWS Details, Readme, and Reset. A terminal window shows a command prompt: eee\_l\_5353255@runweb195992:~\$ . To the right, a sidebar titled "Learner Lab" lists several links: Environment Overview, Environment Navigation, Access the AWS Management Console, Region restriction, Service usage and other restrictions, Using the terminal in the browser, Running AWS CLI commands, Using the AWS SDK for Python, Preserving your budget, Accessing EC2 Instances, SSH Access to EC2 Instances, and SSH Access from Windows. At the bottom, there are "Previous" and "Next" buttons.

### Step 4: click on AWS and in the services select EC2

The screenshot shows the AWS Academy Learner Lab interface. The navigation bar now has "AWS" selected. The main area shows a terminal window with a command prompt: eee\_l\_3940257@runweb155453:~\$ . To the right, the "Learner Lab" sidebar remains the same, listing the same set of links as in Step 3. At the bottom, there are "Previous" and "Next" buttons.

## Step 5: select instances and select instance click on launch instance

The screenshot shows the AWS EC2 Dashboard in the N. Virginia region. The left sidebar has sections for Instances, Images, and Elastic Block Store. The main area displays 'Resources' with counts for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key Pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this is the 'Launch instance' section, which includes a note about launching in the US East (N. Virginia) Region, a 'Launch instance' button, and a 'Migrate a server' link. To the right is the 'Service health' section showing the AWS Health Dashboard, Region (US East (N. Virginia)), Status (operating normally), and Zones. The top right shows account attributes like Default VPC and Settings, and a 'Explore AWS' section with cost reduction tips.

## Step 6: Give the name of the machine "week-122"

The screenshot shows the 'Launch an instance' wizard in the EC2 service. The top navigation bar shows tabs for Launch AWS Academy Learner, Launch an instance, Two-factor authentication - Git, and Verify it's you. The main page title is 'Launch an instance' with an 'Info' link. It says 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.' Under 'Name and tags', the 'Name' field contains 'week-122'. Other fields include 'Add additional tags'. The 'Application and OS Images (Amazon Machine Image)' section shows a search bar and a grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A 'Quick Start' tab is selected. On the right, there's a 'Summary' section with 'Number of instances' set to 1, 'Software Image (AMI)' (Canonical, Ubuntu, 24.04, amd64), 'Virtual server type (instance type)' (t3.micro), 'Firewall (security group)' (New security group), 'Storage (volumes)' (1 volume(s) - 8 GiB), and a large orange 'Launch instance' button. At the bottom, there are links for 'Cancel', 'Preview code', and 'CloudShell Feedback'. The footer includes copyright information and a status bar showing weather, time, and battery level.

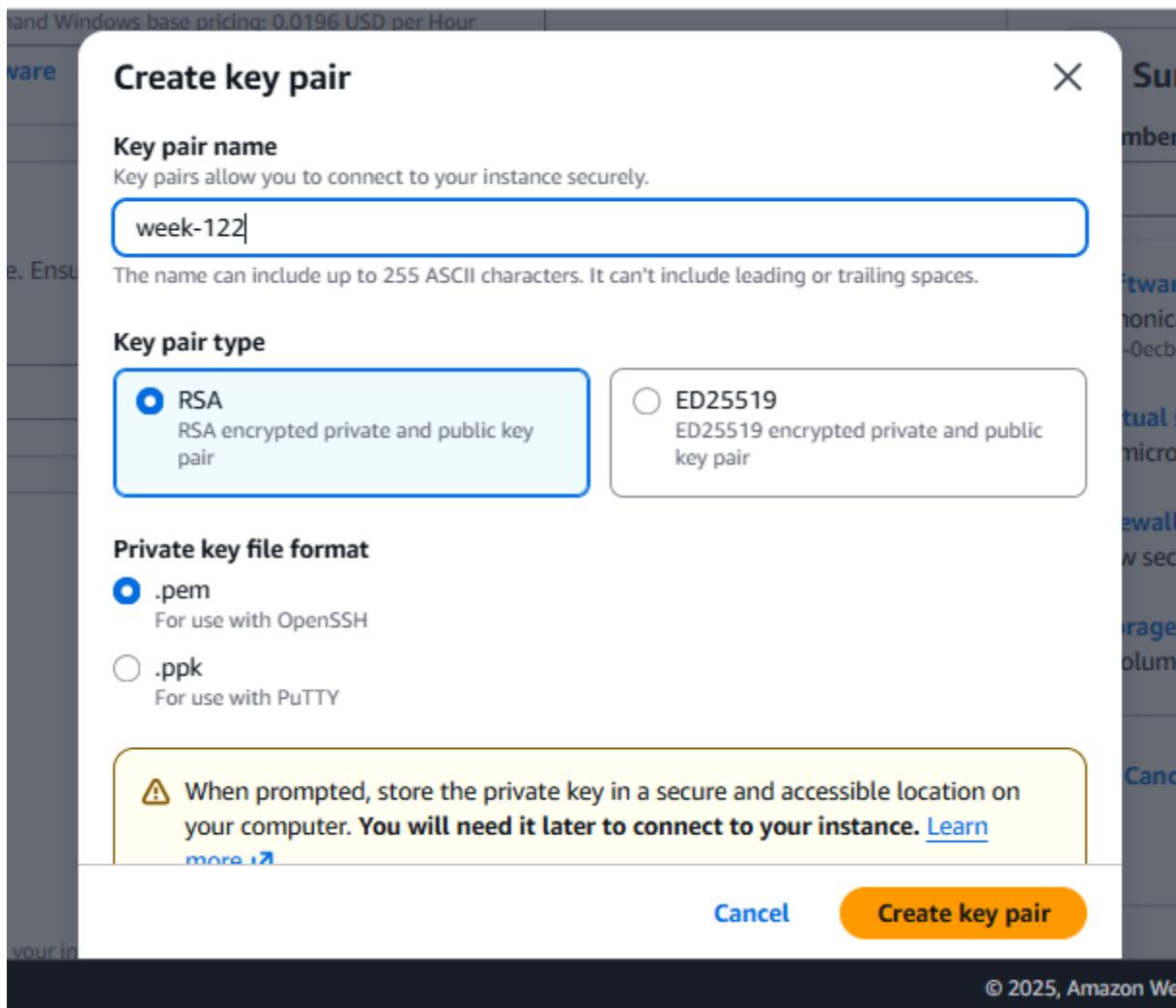
## Step 6: Select the ubuntu server

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Amazon Machine Image (AMI)' section, 'Ubuntu Server 24.04 LTS (HVM, SSD Volume Type)' is selected. The 'Architecture' dropdown is set to '64-bit (x86)'. On the right, the 'Summary' panel shows the instance configuration: 1 instance, Canonical, Ubuntu, 24.04, amd64 noble image, t3.micro instance type, and 8 GiB storage. The 'Launch instance' button is highlighted.

## Step 7: select architecture as 64-bit and instance type as t3.micro(i.e., they are free)

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Instance type' section, 't3.micro' is selected. The 'Summary' panel on the right shows the configuration: 1 instance, Canonical, Ubuntu, 24.04, amd64 noble image, t3.micro instance type, and 8 GiB storage. The 'Launch instance' button is highlighted.

Step 8: Create a new keypair and select type as RSA and .pem option and click on create key pair



© 2025, Amazon Web Services, Inc.

Step 9: In network settings select “create security group” and give the security group name

▼ Network settings [Info](#)

VPC - **required** | [Info](#)

vpc-05a9ef3852073b114 (default) ▾ [C](#)

Subnet | [Info](#)

No preference ▾ [C](#) Create new subnet [L](#)

Availability Zone | [Info](#)

No preference ▾ [C](#) Enable additional zones [L](#)

Auto-assign public IP | [Info](#)

Enable ▾

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Security group name - **required**

week-122

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 \_.-/()#,@[]+=;&();\$\*

Description - **required** | [Info](#)

Launch-wizard-1 created 2025-11-11T05:36:49.724Z

Step 10: Click on edit button on the top right corner and select

Type: ssh

Source: anywhere

EC2 > Instances > Launch an instance

week-122

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 \_.-/()#,@[]+=;&();\$\*

Description - **required** | [Info](#)

Launch-wizard-1 created 2025-11-11T05:36:49.724Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | [Info](#) Protocol | [Info](#) Port range | [Info](#)

ssh TCP 22

Remove

Source type | [Info](#) Source | [Info](#) Description - optional | [Info](#)

Anywhere Add CIDR, prefix list or security group e.g. SSH for admin desktop

0.0.0.0/0 X

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

Summary

Number of instances | [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6... [read more](#)

ami-decb62995f68bb549

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

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

## Step 11: In configure storage give 8GB and give number of instances as 2 and click on launch instance

The screenshot shows the AWS EC2 'Launch an Instance' configuration page. In the 'Configure storage' section, there is a warning about security group rules and an 'Add security group rule' button. Below it, the storage configuration shows 1x 8 GiB gp3 volume selected. The 'Number of instances' dropdown is set to 2. In the 'Software Image (AMI)' section, it shows Canonical, Ubuntu, 24.04, amd64. The 'Virtual server type (instance type)' is t3.micro. The 'Launch instance' button is highlighted in orange.

## Step 12: The launching of instance will start and successful message will be shown

The screenshot shows the AWS EC2 'Launch an Instance' progress page. A blue progress bar indicates the process is at 33% completion, with the status 'Creating security group rules'. A message below says 'Please wait while we launch your instance. Do not close your browser while this is loading.'

Below this, the success message 'Successfully initiated launch of instances (E-0f868fed463f89656, i-0a5aa6fe5d0039e34)' is displayed in a green box. The 'Next Steps' section includes links to 'Create billing usage alerts', 'Connect to your instance', 'Connect an RDS database', 'Create EBS snapshot policy', 'Manage detailed monitoring', 'Create Load Balancer', 'Create AWS budget', and 'Manage CloudWatch alarms'.

Step 13: In the instances the created ones will be shown, you can also rename the instance , changed week-1222 to “webapp”

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'EC2' selected. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
24bd5d0503	i-058d48ebcd5b72f91	Terminated	t3.micro	-	View alarms	us-east-1a	-	-
week-12	i-0d69dc97b6767aaaf3	Terminated	t3.micro	-	View alarms	us-east-1a	-	-
week-122	i-0a5aa6fe5d0039e34	Running	t3.micro	Initializing	View alarms	us-east-1c	ec2-13-220-246-239.co...	13.220
webapp	i-0f868f6d463f89656	Running	t3.micro	Initializing	View alarms	us-east-1c	ec2-13-222-21-231.co...	13.222

Below the table, the details for the 'webapp' instance are expanded. The 'Details' tab is active, showing:

- Instance ID: i-0f868f6d463f89656 (webapp)
- Public IPv4 address: 13.222.21.231 | open address
- Private IPv4 addresses: 172.31.9.83
- Public DNS: ec2-13-222-21-231.compute-1.amazonaws.com | open address
- IPv6 address: -
- Instance state: Running

The bottom of the screen shows the AWS navigation bar and a taskbar with various icons.

Step 14: click on connect and select “SSH Client” and copy the ssh command

The screenshot shows the 'Connect' dialog for the 'webapp' instance. The 'SSH client' tab is selected. The dialog contains the following information:

- Instance ID: i-0f868f6d463f89656 (week-122)
- SSH command copied to clipboard: ssh -i "week-122.pem" ubuntu@ec2-13-222-21-231.compute-1.amazonaws.com
- Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

At the bottom right, there's a 'Cancel' button.

Step 15: Navigate to the path where the file with .pem extension is present(week-122.pem) and paste the command

```
PS C:\Users\User\downloads> ssh -i "week-122.pem" ubuntu@ec2-13-222-21-231.compute-1.amazonaws.com
The authenticity of host 'ec2-13-222-21-231.compute-1.amazonaws.com (13.222.21.231)' can't be established.
ED25519 key fingerprint is SHA256:NEGegchQjt8om/1AVLsqfmafnMphv5Ad4A1Mwo8qECo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-222-21-231.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Nov 11 05:50:06 UTC 2025

System load: 0.08 Temperature: -273.1 C
Usage of /: 25.9% of 6.71GB Processes: 118
Memory usage: 24% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.9.83

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-9-83:~$
```

Step 16: check the docker and git version

If they are not present, then go to administrative terminal using command

“sudo su”

Then update using the command “sudo apt-get update”

```

ubuntu@ip-172-31-9-83:~$ docker --version
Command 'docker' not found, but can be installed with:
sudo snap install docker      # version 28.4.0, or
sudo snap install docker      # version 28.1.1+1
sudo apt install docker.io    # version 28.2.2-0ubuntu1~24.04.1
sudo apt install podman-docker # version 4.9.3+ds1-1ubuntu0.2
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-9-83:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-9-83:~$ sudo su
root@ip-172-31-9-83:/home/ubuntu# sudo apt-get update

```

### Step 17: use command “sudo apt-get install docker.io” to install docker

```

Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [27.4 kB]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5708 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
Fetched 38.6 MB in 6s (6197 kB/s)
Reading package lists... Done
root@ip-172-31-9-83:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse
zfs-fuse | zfsutils
The following NEW packages will be installed:
bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 10 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 284 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.2 [8815 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.0 MB in 1s (81.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 71735 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.3.3-0ubuntu1~24.04.2_amd64.deb ...
Unpacking runc (1.3.3-0ubuntu1~24.04.2)

```

Step 18: Clone the git repo that has maven project and change to that directory

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-172-31-9-83:/home/ubuntu# git clone https://github.com/Gayathri2608-hub/maven-practice.git  
Cloning into 'maven-practice'...  
remote: Enumerating objects: 60, done.  
remote: Counting objects: 100% (60/60), done.  
remote: Compressing objects: 100% (40/40), done.  
remote: Total 60 (delta 11), reused 34 (delta 2), pack-reused 0 (from 0)  
Receiving objects: 100% (60/60), 13.39 KiB | 3.35 MiB/s, done.  
Resolving deltas: 100% (11/11), done.  
root@ip-172-31-9-83:/home/ubuntu# ls  
maven-practice  
root@ip-172-31-9-83:/home/ubuntu# cd maven-practice  
root@ip-172-31-9-83:/home/ubuntu/maven-practice# ls  
Dockerfile demo pom.xml readme src target  
root@ip-172-31-9-83:/home/ubuntu/maven-practice#
```

Step 19: change to the project directory and check for Dockerfile, if not present create the Dockerfile – “nano Dockerfile” and then build the image

“sudo docker build -t image\_name .” name of image:img1

```
root@ip-172-31-9-83:/home/ubuntu/maven-practice# ls  
Dockerfile demo pom.xml readme src target  
root@ip-172-31-9-83:/home/ubuntu/maven-practice# sudo docker build -t dep1 .  
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.  
Install the buildx component to build images with BuildKit:  
https://docs.docker.com/go/buildx/  
 Sending build context to Docker daemon 120.8kB  
Step 1/4 : FROM tomcat:9.0  
 9.0: Pulling from library/tomcat  
4b3ffd8ccb52: Pulling fs layer  
b48f960b380d: Pulling fs layer  
58424d8c3e86: Pulling fs layer  
4f4fb700ef54: Pulling fs layer  
37b617836889: Pulling fs layer  
891b6ad931b7: Pulling fs layer  
ac0beccecf50: Pulling fs layer  
4f4fb700ef54: Waiting  
37b617836889: Waiting  
891b6ad931b7: Waiting  
ac0beccecf50: Waiting  
b48f960b380d: Verifying Checksum  
b48f960b380d: Download complete  
4b3ffd8ccb52: Verifying Checksum  
4b3ffd8ccb52: Download complete  
4f4fb700ef54: Verifying Checksum  
4f4fb700ef54: Download complete  
37b617836889: Verifying Checksum  
37b617836889: Download complete  
891b6ad931b7: Verifying Checksum  
891b6ad931b7: Download complete  
ac0beccecf50: Verifying Checksum  
ac0beccecf50: Download complete  
58424d8c3e86: Verifying Checksum  
58424d8c3e86: Download complete  
4b3ffd8ccb52: Pulling fs layer
```

Step 20: Run the image “sudo docker run -d -p 8081:8080 img1”

```
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# sudo docker run -d -p 8081:8080 img1
c5fd91cf9a9b4f0625d4d2c0d896406e8da76929ed75a3f9ccc1699fbbb08535
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice#
```

Step 21: Check the images and the containers

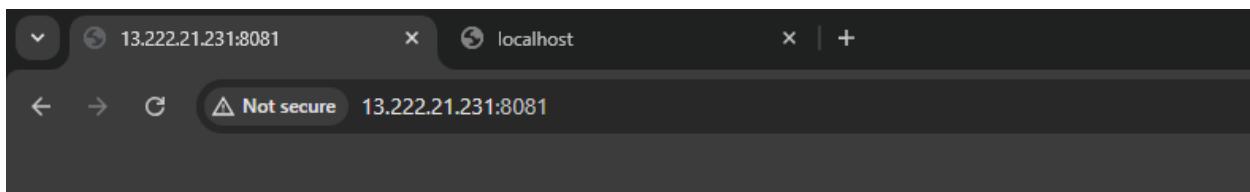
```
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
img1            latest   a67a112ce8ac  2 minutes ago  413MB
dep1            latest   28efbe56e633  29 minutes ago  413MB
tomcat          9.0     2e4887a16e43  12 hours ago   413MB
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice# docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS
              NAMES
c5fd91cf9a9b   img1      "catalina.sh run"  About a minute ago  Up About a minute  0.0.0.0:8081->8080/tcp, [::]:8081->8080/tcp  charming_banzai
84e7f9ce5ec2   dep1      "catalina.sh run"  9 minutes ago    Up 9 minutes   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp  angry_shaw
b62aedc8bb3b   dep1      "catalina.sh run"  27 minutes ago   Up 27 minutes  0.0.0.0:7070->8080/tcp, [::]:7070->8080/tcp  sweet_archimedes
root@ip-172-31-9-83:/home/ubuntu/ar/maven-practice#
```

Step 22: Take the public IP address from the instances in AWS and open it in chrome along with the port number mapped.

Public IP- 13.222.21.231

Port used: 8081

Use: 13.222.21.231:8081, you will find your application that is deployed



**Hello World! practice**