



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION)



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



DEPARTMENT OF INFORMATION TECHNOLOGY

LAB RECORD

SOFTWARE ENGINEERING LAB

B.Tech. III YEAR I SEM (RKR21)

ACADEMIC YEAR 2024-25



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad



NBA
NATIONAL BOARD FOR
TECHNICAL EDUCATION

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



(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad**Daily Laboratory Assessment Sheet**

Name of the Lab:

Name of the Student:

Class:

HT. No:

S.No.	Name of the Experiment	Date	Observation Marks (3M)	Record Marks (4M)	Viva Voice Marks (3M)	Total Marks (10M)	Signature of Faculty
	TOTAL						

INDEX

S.NO	CONTENTS	PAGE NO
I	Vision/Mission /PEOs/POs/PSOs	
II	Syllabus	
III	Course outcomes, C0-PO Mapping	
Exp No:	List of Experiments	
1	INSTALLATION OF STARUML, GIT BASH AND GITHUB ACCOUNT CREATION	
2	IEEE – SRS (SOFTWARE REQUIREMENTS SPECIFICATION) WITH UML DIAGRAMS	
3	INSTALLATION OF ECLIPSE, MAVEN, JDK, TOMCAT, CONFIGURING TOMCAT TO ECLIPSE	
4	A. BASIC GIT COMMANDS - VERSION, CONFIG, INIT, STATUS, ADD, COMMIT, DIFF, HELP B. GIT COMMANDS: WORKING WITH LOCAL AND REMOTE REPOSITORIES - BRANCHES, CHECKOUT, MERGE, REVERT, LOG C. GIT COMMANDS: WORKING WITH REMOTE REPOSITORIES - REMOTE, CLONE, PULL, PUSH, FORK	
5	A. CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. B. CREATING MAVEN WEB PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB. C. INSTALLATION OF JENKINS.	
6	A. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT B. BUILDING THE CI/CD FREESTYLE PIPELINE USING JENKINS FOR MAVEN WEB PROJECT WITH POLL SCM C. BUILDING THE CI/CD SCRIPTED PIPELINE USING JENKINS FOR MAVEN JAVA PROJECT WITH POLL SCM	
7	A. INSTALLATION OF DOCKER, MINIKUBE, ACCOUNT IN DOCKERHUB B. DOCKER CLI COMMANDS C. MODIFY AND PUSH DOCKER IMAGE D. CREATE AND PUSH DOCKERFILE IMAGE E. RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE	

	F. DEPLOYING AND SCALING APPLICATIONS USING MINIKUBE G. DEPLOYING AND MANAGING MONITORING SYSTEMS USING NAGIOS IN DOCKER	
8	A. AWS ACADEMY LEARNING ACCOUNT CREATION B. PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE C. MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE	



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

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



**NBA
NATIONAL BOARD
FOR HIGHER
EDUCATION**

Department of Information Technology

Vision of the Institution:

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

Mission of the Institution:

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

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



Department of Information Technology

Vision of the Department

To produce globally competent graduates to meet the modern challenges through contemporary knowledge and moral values committed to build a vibrant nation.

Mission of the Department

- To create an academic environment, which promotes the intellectual and professional development of students and faculty.
- To impart skills beyond university prescribed to transform students into a well-rounded IT professional.
- To nurture the students to be dynamic, industry ready and to have multidisciplinary skills including e-learning, blended learning and remote testing as an individual and as a team.
- To continuously engage in research and projects development, strategic use of emerging technologies to attain self-sustainability



Department of Information Technology

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the

engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION)

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



Department of Information Technology

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

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



**NBA
ACCREDITED**

Department of Information Technology

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will endeavor to excel in their chosen careers as professionals, researchers and entrepreneurs on a global platform.

PEO2: Graduates will demonstrate the ability to solve challenges in the fields of Engineering and Technology simultaneously catering to societal needs.

PEO3: Graduates will strive to improve their learning curve by practicing Continuing Professional Development (CPD).

PEO4: Graduates will, at all times, adopt a professional demeanor by communicating effectively, working collaboratively, and maintaining the ethics & core values as befitting their education in interdisciplinary and emerging fields.

B. Tech. in INFORMATION TECHNOLOGY**III Year I Semester Syllabus (RKR21)****SOFTWARE ENGINEERING LAB (21CC505PC)****Common to CSE, IT, CSE (AI&ML) and CSE (DS)**

L	T	P	C
0	0	3	1.5

Pre-requisites/ Co-requisites:

1. 21CC502PC – Software Engineering Course
2. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Formulate problem statements and Software Requirement Specifications by comprehensively grasping project requirements.
2. Demonstrate proficiency in designing, developing, and testing diverse project modules.
3. Utilize Git Framework and GitHub while implementing Continuous Integration/Continuous Deployment (CI/CD) pipelines through Jenkins.
4. Implement project deployment using Docker and Kubernetes.
5. Acquire knowledge in AWS cloud infrastructure.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Transform end-user needs into system and software requirements through a structured process.
2. Depict the system's high-level design using CASE tools based on the software requirements.
3. Employ Jenkins CI/CD for project building purposes.
4. Implement project deployment utilizing Docker and Kubernetes.
5. Create a project within the AWS Cloud environment.

Software to be used: The students must use JDK 11/17/21 Version, STAR UML, GIT Bash, Jenkins, Dockers Desktop, Mini KUBE, Eclipse, Tomcat, and Visual Studio Editor.**List of Experiments:**

Do the following exercises for any one project given in the list of sample projects or any other projects?

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Study and usage of any Design phase CASE tool
4. Creating the project and committing using Git and GitHub
5. Creating Maven Java and Maven Web project using Eclipse and Push them to GitHub.
6. Building the CI/CD pipeline using Jenkins for the project in the previous experiment.
7. Local Deployment of project using Docker, Kubernetes and Monitoring using Nagios tool.

8. Cloud Deployment of a project in the AWS Cloud using EC2 instance.

Sample Projects:

1. Book Bank
2. Online course reservation system
3. E-ticketing
4. Recruitment system
5. Hospital Management system
6. Online Banking System

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition, 2015.
2. Software Engineering- Sommerville, 7th edition, Pearson Education, 2017.
3. The unified modeling language user guide Grady Brooch, James Rumbaugh, Ivar Jacobson, Pearson Education, 2016.
4. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2015.

REFERENCE BOOKS:

1. <https://kubernetes.io/docs/tutorials/hello-minikube/>
2. <https://minikube.sigs.k8s.io/docs/start/>
3. <https://www.jenkins.io/doc/>
4. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
5. Introducing Maven by, Balaji Varanasi and Sudha Belida, APRESS publications.



(AN AUTONOMOUS INSTITUTION)

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

Department of Information Technology

Course Outcomes and CO-PO-PSO Mapping

Course Outcomes:

After learning the contents of this course, the student is able to

CO1	Transform end-user needs into system and software requirements through a structured process.
CO2	Depict the system's high-level design using CASE tools based on the software requirements.
CO3	Employ Jenkins CI/CD for project building purposes.
CO4	Implement project deployment utilizing Docker and Kubernetes.
CO5	Create a project within the AWS Cloud environment.

CO-PO-PSO MAPPING:

	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO-1	PSO-2
Software Engineering Lab	CO1	3	3	3	2						2			2	1
	CO2	2	3	3	3	3					2			2	1
	CO3					3				2	2	2		1	2
	CO4				3	3	2	2		2	2	2	2	2	3
	CO5	3				3	2	2				3	3	3	3

GIT COMMANDS

git init: Initializes a Git repository in the current directory.

nano filename.txt: Creates and edits a new text file in the terminal.

git add filename.txt: Stages the specified file for the next commit.

git commit -m "commit message": Commits the staged changes to the repository with the provided commit message.

git log: Displays a log of commits made to the repository.

git status: Shows the current state of the working directory and staging area.

git help: Provides documentation and help for Git commands.

git mv filename1.txt filename2.txt: Renames a file from filename1 to filename2, or moves the file to a different directory.

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB
$ git init
Initialized empty Git repository in D:/SUNNY/SELAB/.git/

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ nano file1.txt

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git commit -m "added file1"
[master (root-commit) ea0f6df] added file1
 1 file changed, 2 insertions(+)
 create mode 100644 file1.txt

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git log
commit ea0f6df95ad6e62bc966224dcbb26bfaeed2bb5a (HEAD -> master)
Author: puppalasagar <sagarpuppala123@gmail.com>
Date:   Sat Nov 16 10:08:54 2024 +0530

    added file1
```

```

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
● $ git status
On branch master
nothing to commit, working tree clean

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
● $ git help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
clone      Clone a repository into a new directory
init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
add        Add file contents to the index
mv         Move or rename a file, a directory, or a symlink
restore    Restore working tree files
rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
bisect    Use binary search to find the commit that introduced a bug
diff      Show changes between commits, commit and working tree, etc
grep      Print lines matching a pattern

```

Week 5: Working with GITHUB and Eclipse

Collaboration

git remote: Adds a new remote repository. This is useful when you want to link your local repository with a remote one, like GitHub.

git remote add <name> <url>

```

sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
● $ git remote add origin https://github.com/Puppala-Sagar/SE.git

```

git push: Used to transfer the commits or pushing the content from the local repository to the remote repository. The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

```
git push -u origin <master|current directory>
```

```
sagar@SAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Puppala-Sagar/SE.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

git clone: used to create a local working copy of an existing remote repository. The command downloads the remote repository to the local computer.

```
git clone <remote url>
```

```
sagar@SAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ git clone https://github.com/Puppala-Sagar/SE.git
Cloning into 'SE'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

git fetch : Downloads changes from a remote repository without applying them to your working directory. You can later merge these changes.

```
git fetch <remote>
```

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
● $ git fetch https://github.com/Puppala-Sagar/SE.git
  From https://github.com/Puppala-Sagar/SE
    * branch           HEAD       -> FETCH_HEAD
```

git pull : Fetches and integrates changes from a remote repository by rebasing instead of merging. This creates a linear history.

```
git pull <remote> <branch>
```

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
● $ git pull
  Already up to date.
```

Using SSH Key for Authentication in Git-Github:

Using SSH keys in GitHub allows you to securely authenticate your GitHub account from your computer without needing to repeatedly enter your username and password. This method is more secure and convenient, especially for frequent interactions with GitHub repositories.

Step 1: Set Up SSH

Check for Existing SSH Keys

Before creating a new SSH key, check if you already have one:

```
ls -al ~/.ssh
```

Look for files named `id_rsa` and `id_rsa.pub`. If these files exist, you already have an SSH key pair.

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ ls -al ~/.ssh
total 42
drwxr-xr-x 1 sagar 197609 0 Nov 16 09:43 .
drwxr-xr-x 1 sagar 197609 0 Nov 16 10:08 ..
-rw-r--r-- 1 sagar 197609 155 Jun 3 14:16 config
-rw-r--r-- 1 sagar 197609 2602 Nov 5 11:02 id_rsa
-rw-r--r-- 1 sagar 197609 572 Nov 5 11:02 id_rsa.pub
-rw-r--r-- 1 sagar 197609 2538 Nov 16 09:43 known_hosts
-rw-r--r-- 1 sagar 197609 1809 Jun 3 14:16 known_hosts.old
```

Generate SSH Key (if you haven't already):

Open a terminal and run the following command to generate an SSH key:

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

Press Enter to accept the default file location and then enter a passphrase
(optional).

Your SSH key will be saved to `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ ssh-keygen -t rsa -b 4096 -C "sagarpuppala123@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/sagar/.ssh/id_rsa): sagar
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in sagar
Your public key has been saved in sagar.pub
The key fingerprint is:
SHA256:7UkkFk11fIqde04Dq94RIYaKCAFmkmsl+jhV+pItFn0 sagarpuppala123@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
| .o.... |
| .o ... |
| . . o..o + + |
| + = o..+. o.= |
| + O o .S o .o. |
| . O . E o . .o.o |
| .+ = . o .. +. |
| oo= o . . . . |
| ..o.o . . . . |
+---[SHA256]---+
```

Add SSH Key to GitHub:

Copy the contents of your SSH key to your clipboard:

```
cat ~/.ssh/id_rsa.pub
```

```
sagar@sAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ cat ./sagar.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAACQDHvS6jgCJMoSxInx0I2rYcfD4c9/0S6IGEEbv2PzAAmdE8f8lqtsnpNwDarJ
BC2/sjZXHCn2FqgzlaLmnThxd/ppOfT+51YA0NpZw3+IcuSnQCrVxjluyLDEwwEESXJtW61j2oYwHrpvA6bXVL07EiuFoqEYZj
Mhfz3cE20vPnTc4QOp64U/mDoK2zM+AvOK096qbW4B35KxDopk5hHsDIFI0xwv7Pgy0P6hkz/PkJlmarcQ60yoZyH0st6V8XJH
6G8xww/kjA87p25JQxNdkqA2P1bUb0mvrcBrekCB/828UPOxEzWi71gA3uTb1LVxp1Sz/CdSZSXET5rQbu0PZCAEMYVP7EJwUp
LRnR6Myrt1n8q1MK1p9RMVEmX3vlumZwxUAZUCO/9IUWOGIR6h5SEBpuuqAih+FIdvVK6G07xYUW8bHFEka2BBNPa0mEaz8Jgd
J74dMMbAI5bzCZ1nifxinQtNWqq9HVPgktNJRi3+VZMJ0+bZhKYUBMNeEpstsgLK0jeYjWPtvCw96CbgANvzYeADSHs5I1Sj5
ZY1riy3LYSpm3WXgCXHciAXwLdR4XjdTsZdw6VnrsgShWfMzlIm3clmerg1Z5yij4nrh0twA2xi4E1Dw4htd7FdokvtIIZCdY
825TY8BWrlaZzY9AFMB20DnAcKpYlu+5nfIQ== sagarpuppala123@gmail.com
```

Copy the entire output.

Now, Go to GitHub, navigate to **Settings > SSH and GPG keys**, and click **New SSH key**.

Paste the SSH key into the "Key" field and give it a title, then click **Add SSH key**.

The screenshot shows the GitHub user interface for adding a new SSH key. On the left, there is a sidebar with navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access (Billing and plans, Emails, Password and authentication, Sessions), SSH and GPG keys (which is currently selected and highlighted in blue), Organizations, Enterprises, and Moderation. Below the sidebar, there is a note: 'Code, planning, and automation'. The main content area has a title 'Add new SSH Key' and a 'Title' input field. Under 'Key type', there is a dropdown menu set to 'Authentication Key'. The 'Key' input field contains the copied SSH key text from the terminal. At the bottom right of the main form, there is a green 'Add SSH key' button.

The screenshot shows the GitHub profile page for 'Puppala-Sagar'. The left sidebar includes links for Public profile, Account, Appearance, Accessibility, Notifications, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (which is selected), Organizations, and Enterprises. The main content area is titled 'SSH keys' and contains two entries:

Key	Type	Added	Last used	Action
sagar@SAGARPUPPALA	SSH	Added on Nov 5, 2024	Within the last week — Read/write	Delete
sagarpuppala123@gmail.com	SSH	Added on Nov 16, 2024	Never used — Read/write	Delete

A note at the bottom suggests connecting to GitHub using SSH keys or troubleshooting common SSH problems.

Fork :

In Git, a "**fork**" refers to a copy of a repository from one user's account to another user's account on a platform like GitHub. Forking is commonly used to contribute to open-source projects or collaborate on projects where you don't have direct write access to the original repository. Forking allows you to work independently.

1. Fork a Repository:

- o Go to a repository you want to contribute to on GitHub (e.g., <https://github.com/octocat/Spoon-Knife>).

Click the **Fork** button in the top-right corner to create a copy of the repository under your account.

The image shows two screenshots of a browser window displaying a GitHub repository and a fork creation dialog.

Screenshot 1: GitHub Repository Page

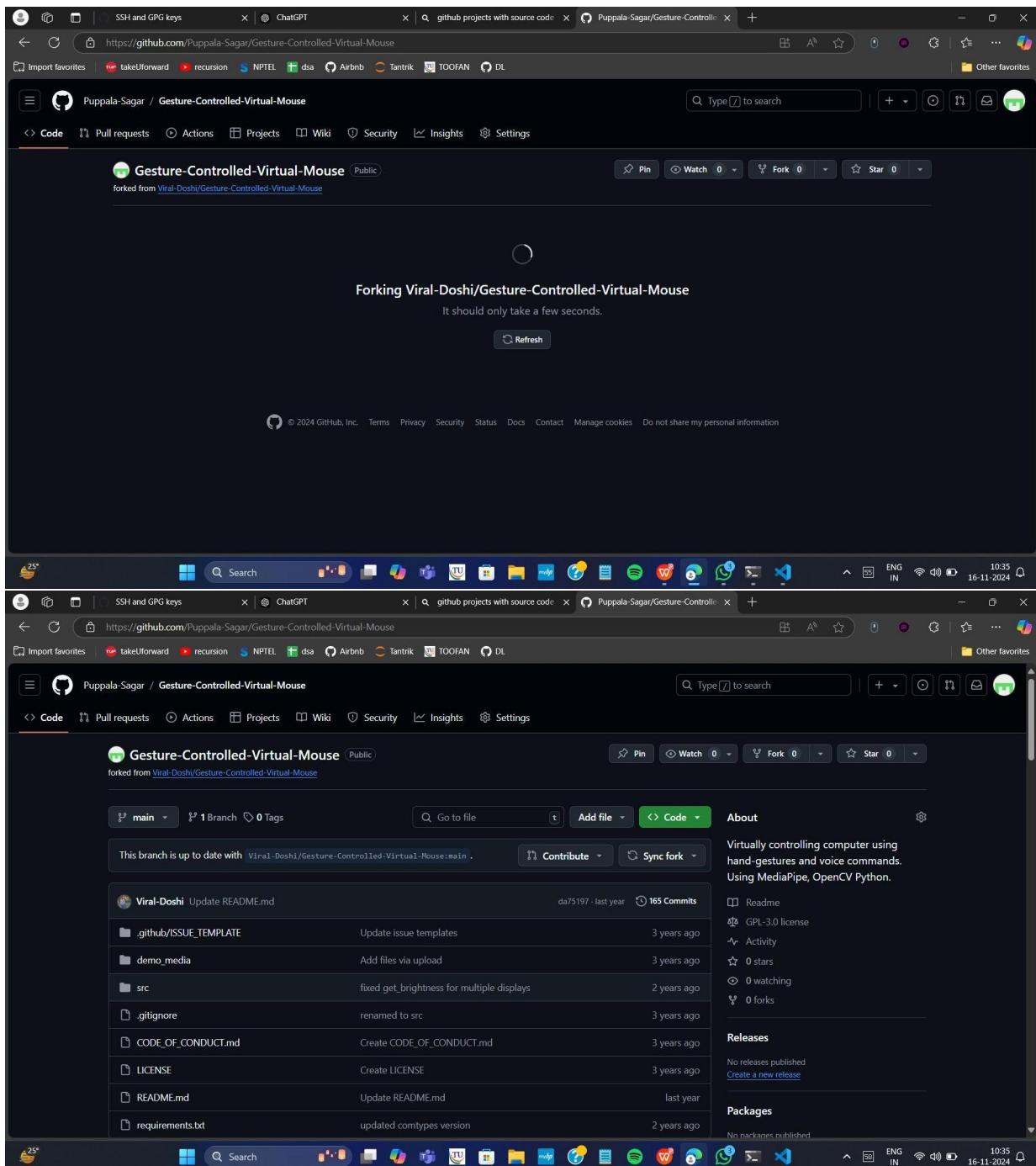
The top screenshot shows the GitHub repository `Viral-Doshi/Gesture-Controlled-Virtual-Mouse`. The repository has 12 watchers, 195 forks, and 503 stars. It contains 1 branch and 0 tags. The repository description states: "Virtually controlling computer using hand-gestures and voice commands. Using MediaPipe, OpenCV Python." The repository has 165 commits. The commit list includes:

- `Update README.md` by `Viral-Doshi` (da75197 · last year)
- `Update issue templates` by `Viral-Doshi` (3 years ago)
- `Add files via upload` by `Viral-Doshi` (3 years ago)
- `fixed get_brightness for multiple displays` by `Viral-Doshi` (2 years ago)
- `renamed to src` by `Viral-Doshi` (3 years ago)
- `Create CODE_OF_CONDUCT.md` by `Viral-Doshi` (3 years ago)
- `Create LICENSE` by `Viral-Doshi` (3 years ago)
- `Update README.md` by `Viral-Doshi` (last year)
- `updated comtypes version` by `Viral-Doshi` (2 years ago)

The repository has a `README`, `Code of conduct`, and `GPL-3.0 license`.

Screenshot 2: Fork Creation Dialog

The bottom screenshot shows the "Create a new fork" dialog for the same repository. The dialog asks for the owner and repository name. The owner is set to `Puppala-Sagar` and the repository name is `Gesture-Controlled-Virtu`. A note says: "By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further." The description field contains: "Virtually controlling computer using hand-gestures and voice commands. Using MediaPipe, OpenCV Python." A checked checkbox says: "Copy the `main` branch only". A note below it says: "Contribute back to `Viral-Doshi/Gesture-Controlled-Virtual-Mouse` by adding your own branch. [Learn more](#)". A note at the bottom says: "You are creating a fork in your personal account." A green "Create fork" button is at the bottom right.



Installing Eclipse

To install Eclipse IDE for Enterprise Java Developers (commonly known as Eclipse Enterprise) on your Windows 10 system, follow these steps:

Step 1: Download Eclipse Installer

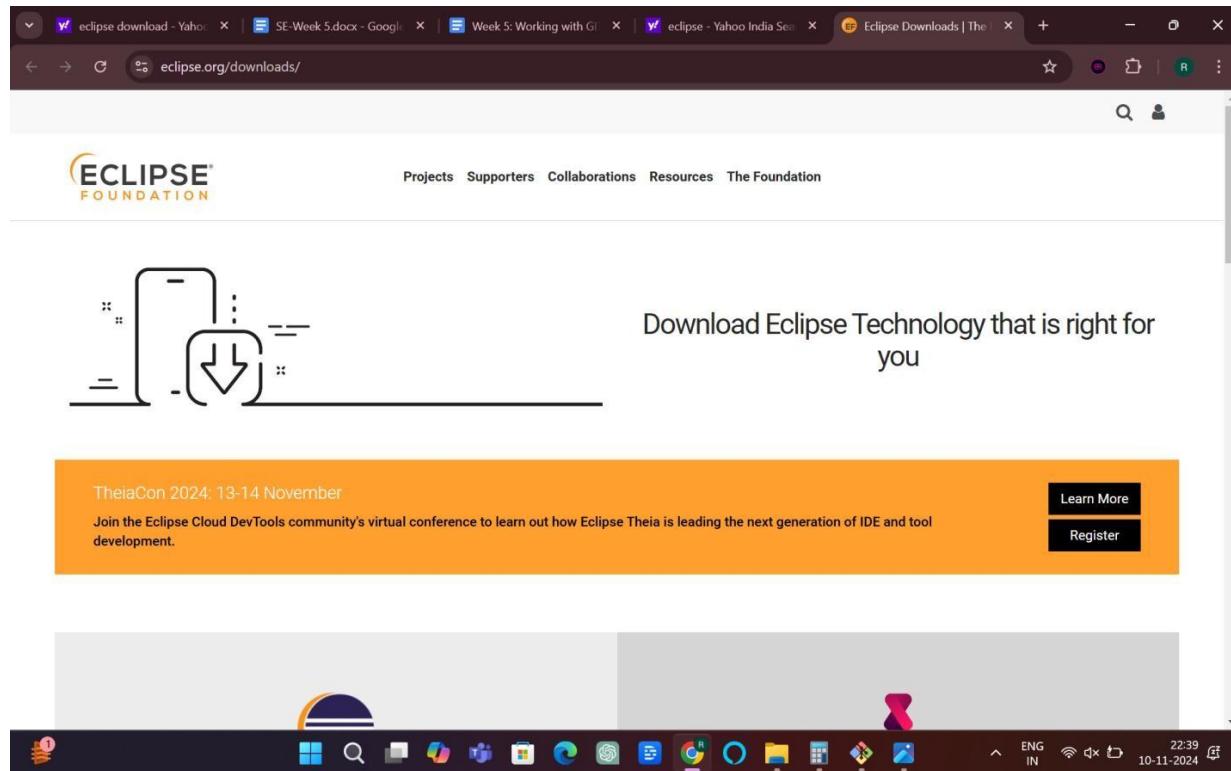
Visit the Eclipse Downloads Page:

Go to the official Eclipse Downloads page.

Download the Eclipse Installer:

Scroll down to find the "Eclipse IDE for Enterprise Java and Web Developers" option.

Click on the "Download x86_64" link to download the installer for Windows.



The Eclipse Installer 2024-09 R now includes a JRE for macOS, Windows and Linux.

Try the Eclipse Installer 2024-09 R

The easiest way to install and update your Eclipse Development Environment.

[Find out more](#)

[613,692 Installer Downloads](#)

[367,145 Package Downloads and Updates](#)

Download

macOS x86_64 | AArch64
Windows x86_64
Linux x86_64 | AArch64

Sponsored Ad

<HDC.Cloud>
HUAWEI DEVELOPER CONFERENCE 2020

Eclipse IDE 2024-09 R Packages

Eclipse IDE for Enterprise Java and Web Developers

530 MB 171,132 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

Windows x86_64
macOS x86_64 | AArch64
Linux x86_64 | AArch64

Install your favorite desktop IDE packages

Learn More [Download x86_64](#)

Download Packages I Need Help

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download

Download from: Korea, Republic Of - Kakao Corp. ([https](https://))

File: [eclipse-jee-2024-09-R-win32-x86_64.zip](#) SHA-512

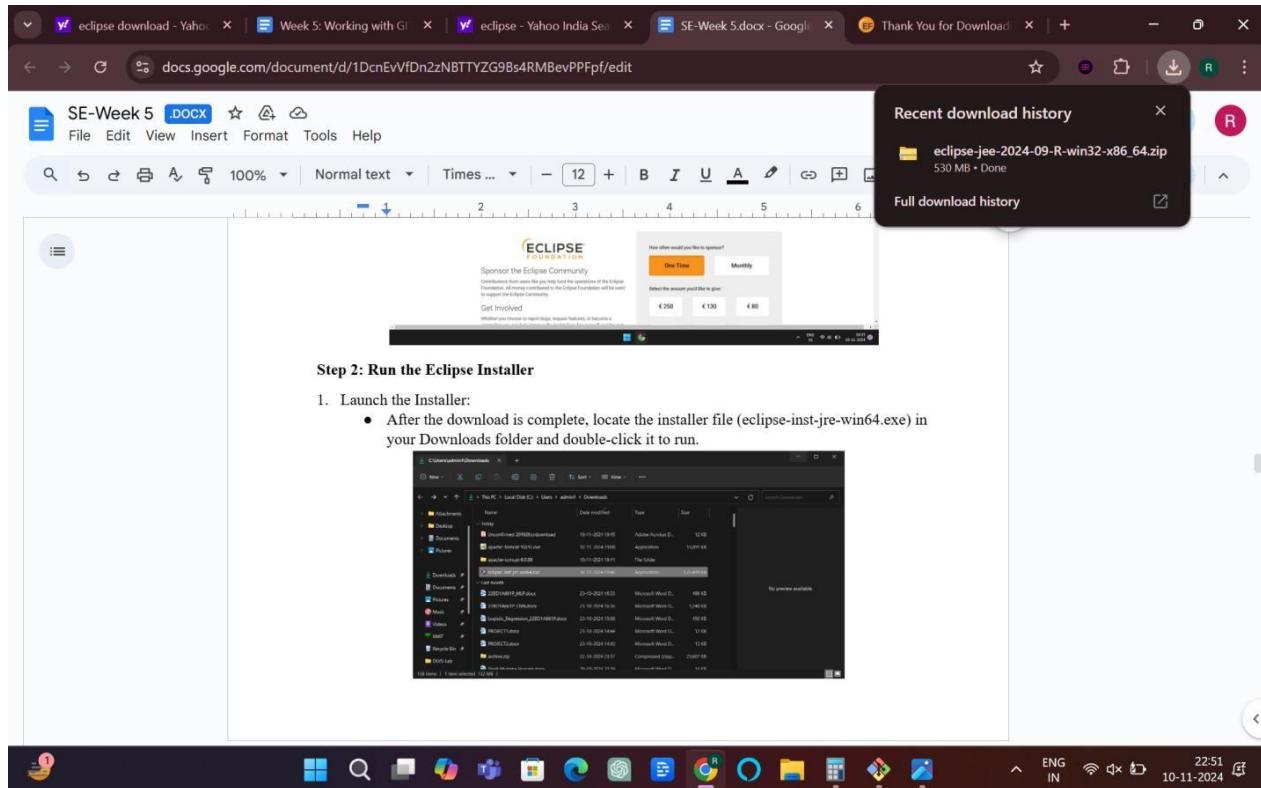
>> Select Another Mirror

Sponsored Ad

<HDC.Cloud>
HUAWEI DEVELOPER CONFERENCE 2020

Other options for this file

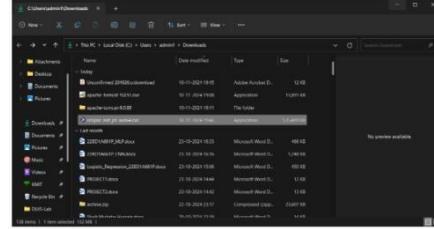
- All mirrors (xml)
- Direct link to file (download starts immediately from best mirror)



Step 2: Run the Eclipse Installer

Launch the Installer:

- After the download is complete, locate the installer file (eclipse-inst-jre-win64.exe) in your Downloads folder and double-click it to run.



Step 2: Run the Eclipse Installer

Launch the Installer:

After the download is complete, locate the installer file (eclipse-inst-jre-win64.exe) in your Downloads folder and double-click it to run.

Choose the IDE Package:

In the Eclipse Installer, you will see various package options. Select "Eclipse IDE for Enterprise Java and Web Developers."

Select Installation Folder:

Choose a destination folder where you want to install Eclipse. The default location is typically fine, but you can change it if needed.

JDK Selection:

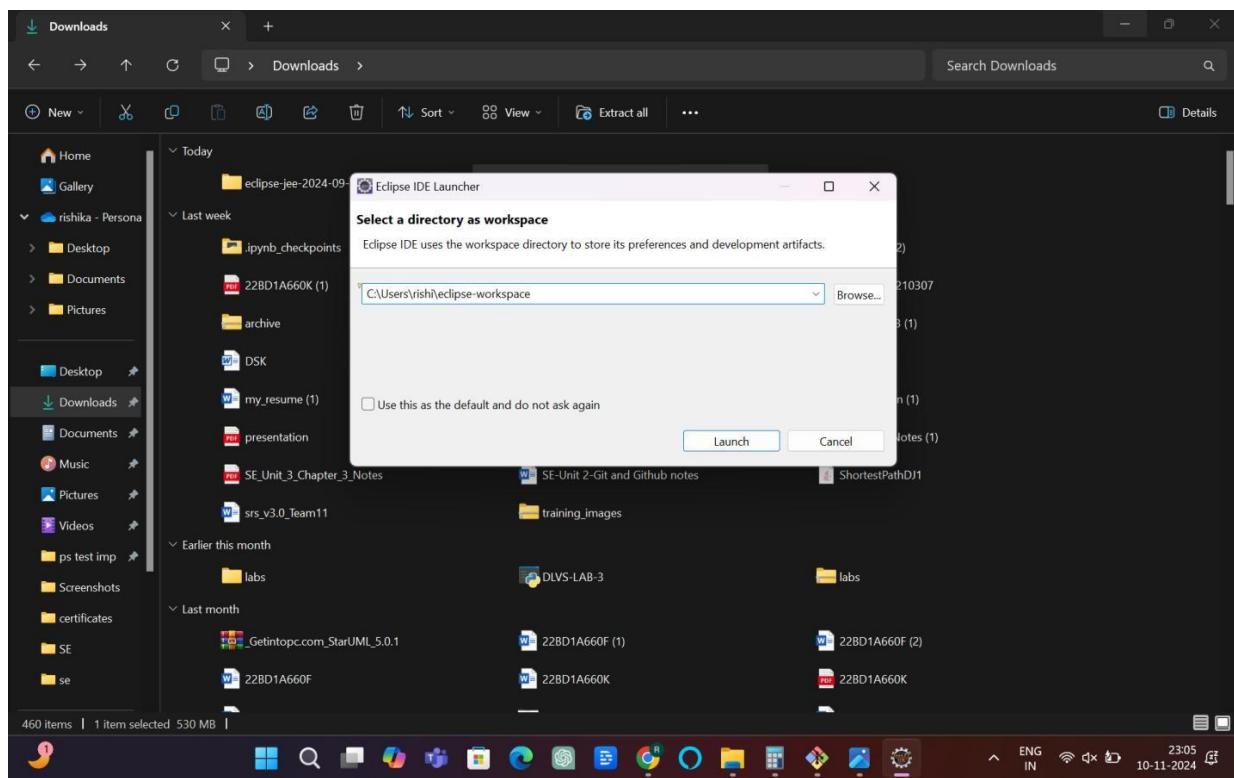
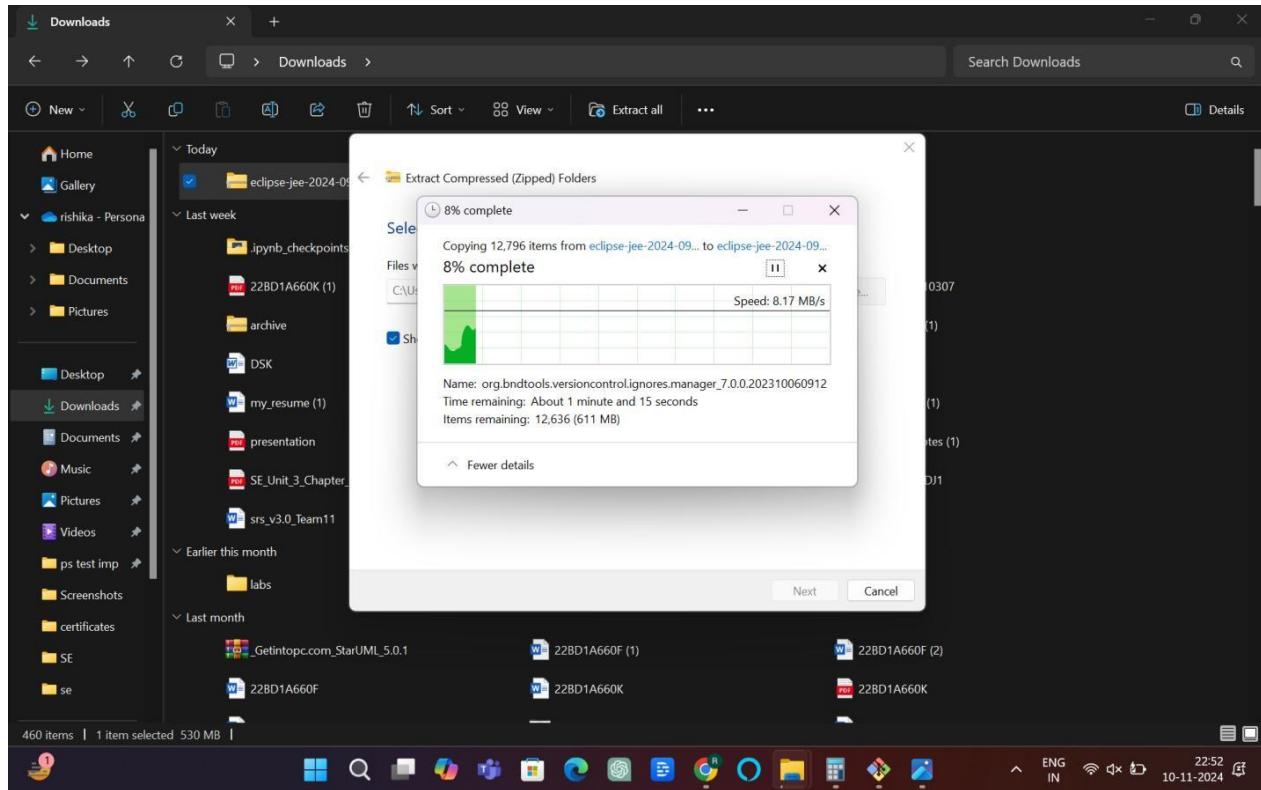
The installer should automatically detect your installed JDK. Ensure that it points to the correct JDK version. If not, you can manually browse and select the correct JDK path.

Start the Installation:

Click on the "INSTALL" button to start the installation process.

Accept the License Agreement:

Read and accept the license agreement to proceed with the installation.



Step 3: Launch Eclipse

Finish Installation:

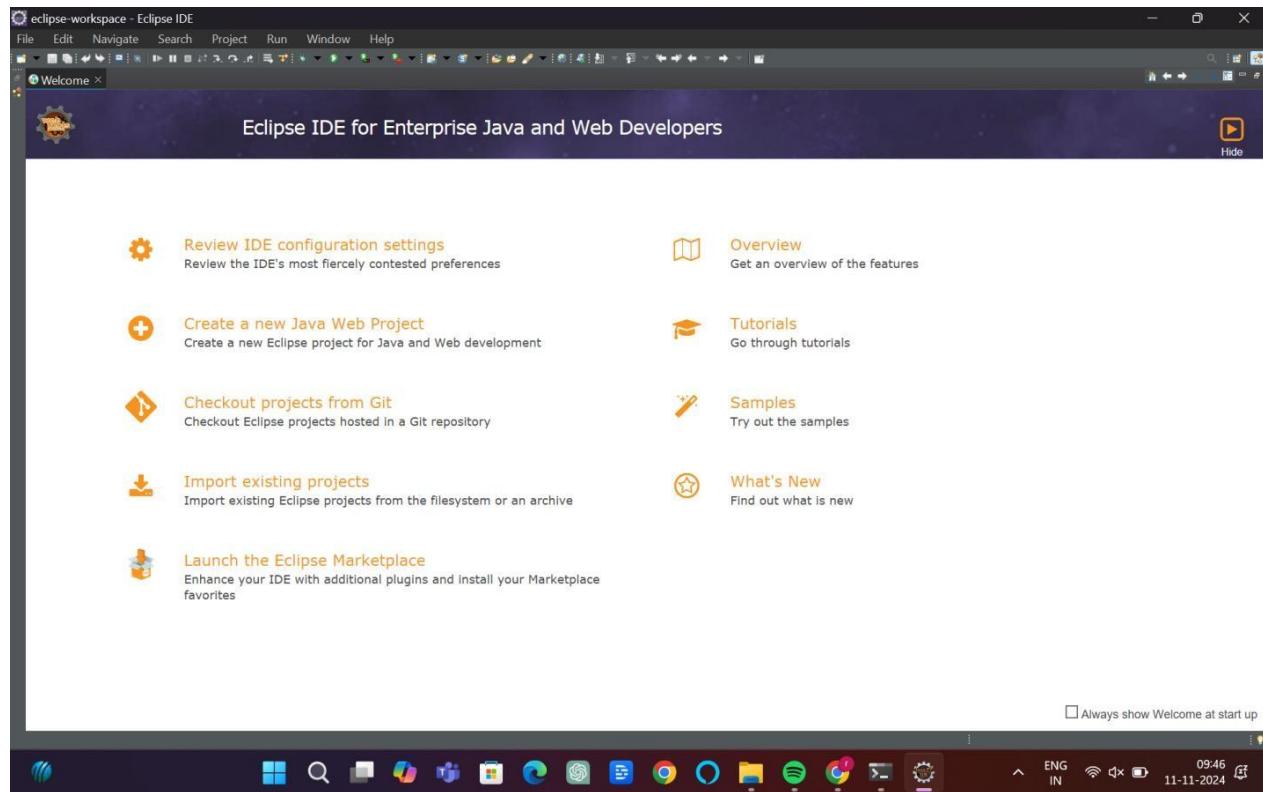
Once the installation is complete, click on the "LAUNCH" button to start Eclipse.

Workspace Selection:

The first time you launch Eclipse, it will ask you to select a workspace. This is the folder where all your projects and settings will be stored. You can either accept the default location or choose a new one.

Start Using Eclipse:

After selecting the workspace, Eclipse will start up, and you can begin creating or importing projects.



Download and install Tomcat :

Apache Tomcat, commonly referred to as Tomcat, is an open-source web server and servlet container developed by the Apache Software Foundation. It is designed to serve Java applications and is widely used to deploy Java-based web applications and services.

Installing Apache Tomcat 9.0 on Windows 10 involves downloading the software, setting up the environment, and configuring Tomcat.

Download Apache Tomcat 9.0

Go to the **Apache Tomcat 9.0 download page**.

The screenshot shows a web browser window with three tabs: "Week 5: Working with GIT HUE!", "SE-Week 5.docx - Google Docs", and "Apache Tomcat® - Apache Tomcat". The main content area displays the Apache Tomcat 9.0 Software Downloads page. The page features a cartoon cat logo, the Apache Tomcat® logo, and the Apache Software Foundation logo. A search bar and a "GO" button are at the top. On the left, there's a sidebar with sections for "Community", "Apache Tomcat", "Download", and "Documentation". The main content includes a "Tomcat 9 Software Downloads" section with a welcome message, a "Quick Navigation" section with links to "KEYS", "9.0.97", "Browse", and "Archives", a "Release Integrity" section with instructions on verifying file integrity using OpenPGP signatures and SHA-512 checksums, a "Mirrors" section listing mirrors and a dropdown for "Other mirrors", and a "9.0.97" section with a note about the README file. The bottom of the screen shows a taskbar with various icons and system status indicators.

Under the **Binary Distributions** section, find the "Core" section and download the **32-bit/64-bit Windows Service Installer (.exe file)**.

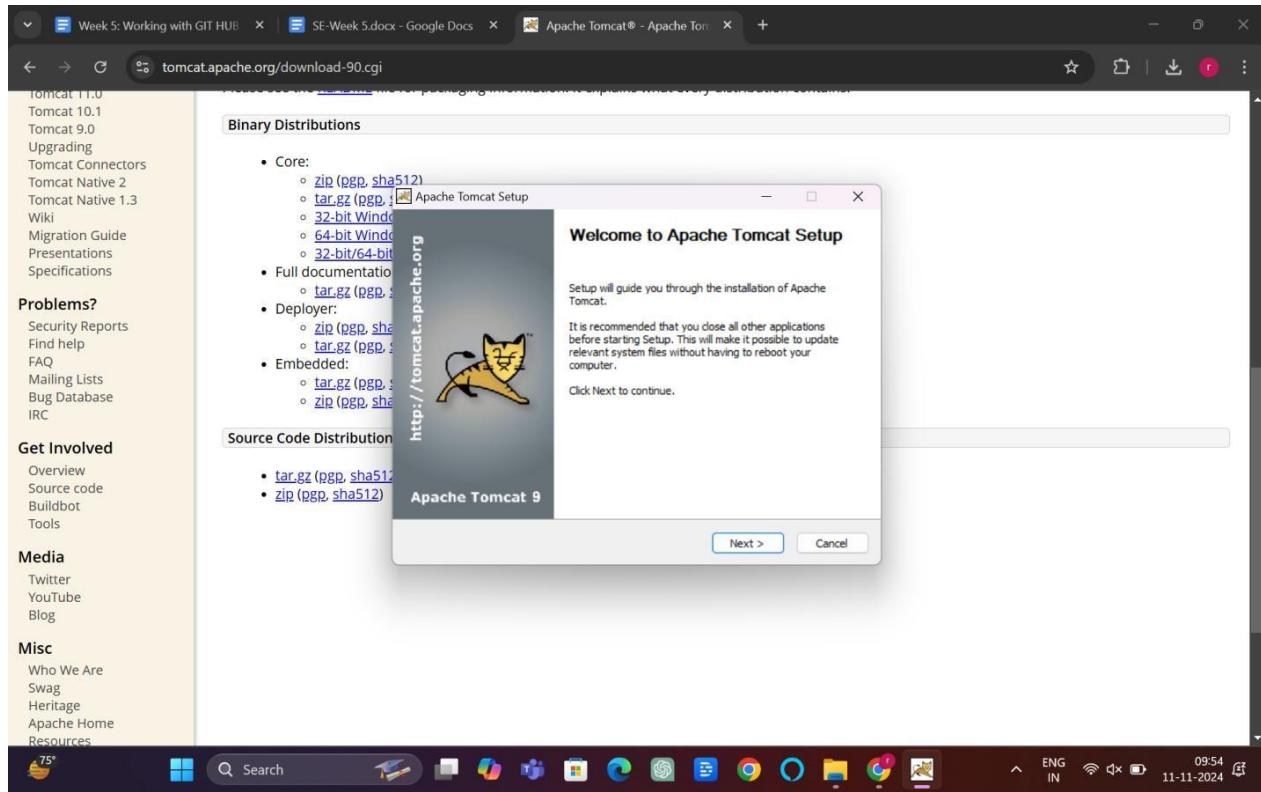
The screenshot shows a web browser window with three tabs open: "Week 5: Working with GIT HU!", "SE-Week 5.docx - Google Docs", and "Apache Tomcat® - Apache Tomcat". The main content area displays the Apache Tomcat download page for version 9.0.97. On the left, there is a sidebar with links for Jakarta EE, Tomcat Connectors, Tomcat Native, Taglibs, Archives, Documentation (Tomcat 11.0, 10.1, 9.0, Upgrading, Connectors, Native 2, Native 1.3, Wiki, Migration Guide, Presentations, Specifications), Problems? (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), Get Involved (Overview, Source code, Buildbot, Tools), and Media (Twitter). The main content area includes a message about mirrors, a dropdown for other mirrors, and sections for "Binary Distributions" and "Source Code Distributions", both listing various download links.

Run the Installer

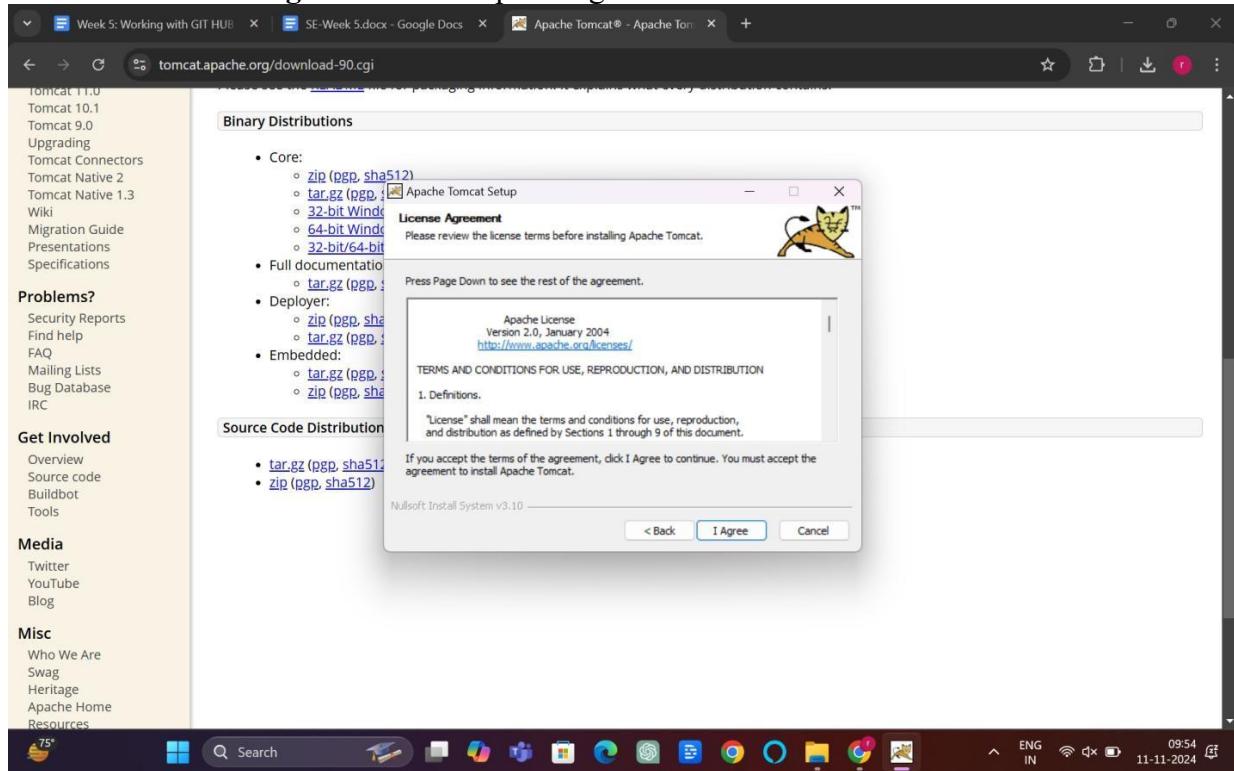
Locate the downloaded .exe file and double-click to start the installation.

During the installation, follow these steps:

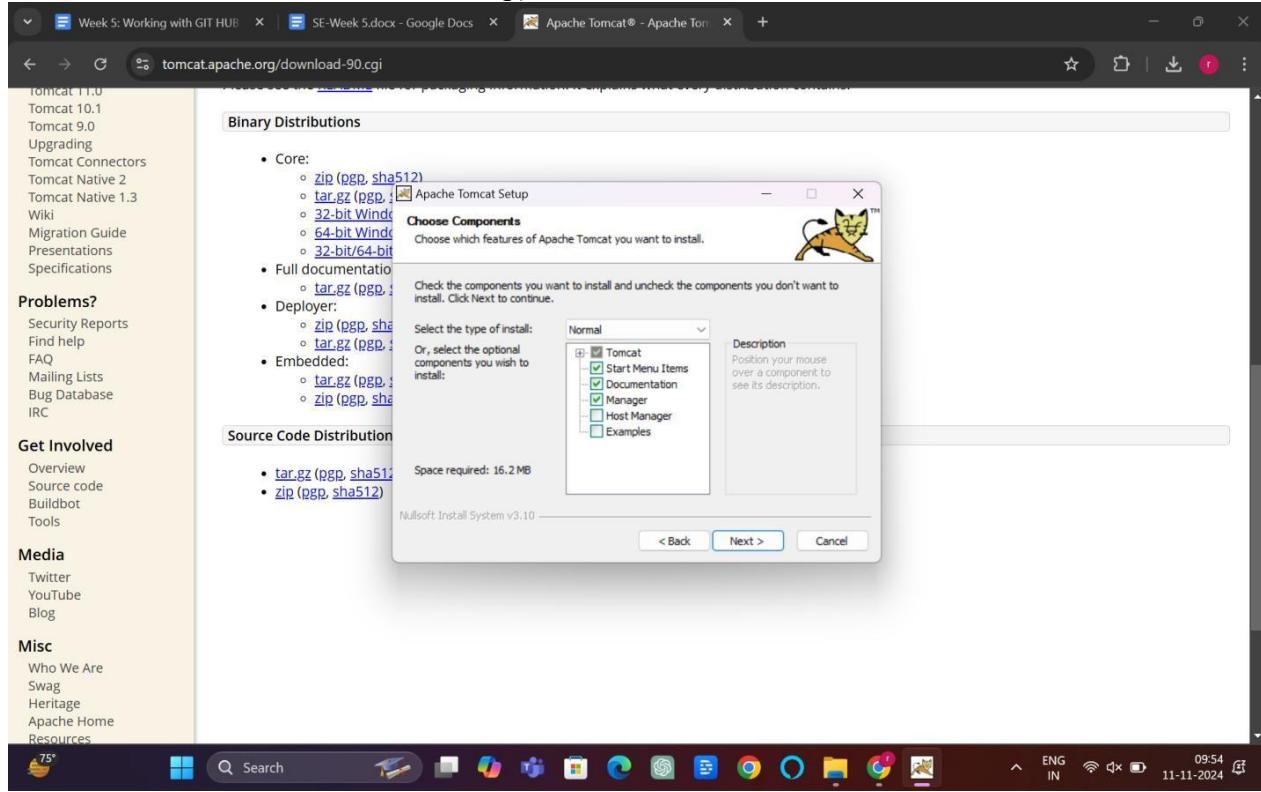
Welcome Screen: Click Next.



License Agreement: Accept the agreement and click Next.



Choose Components: Select the components you need (leave the defaults selected for a basic setup) and click Next.



Configuration:

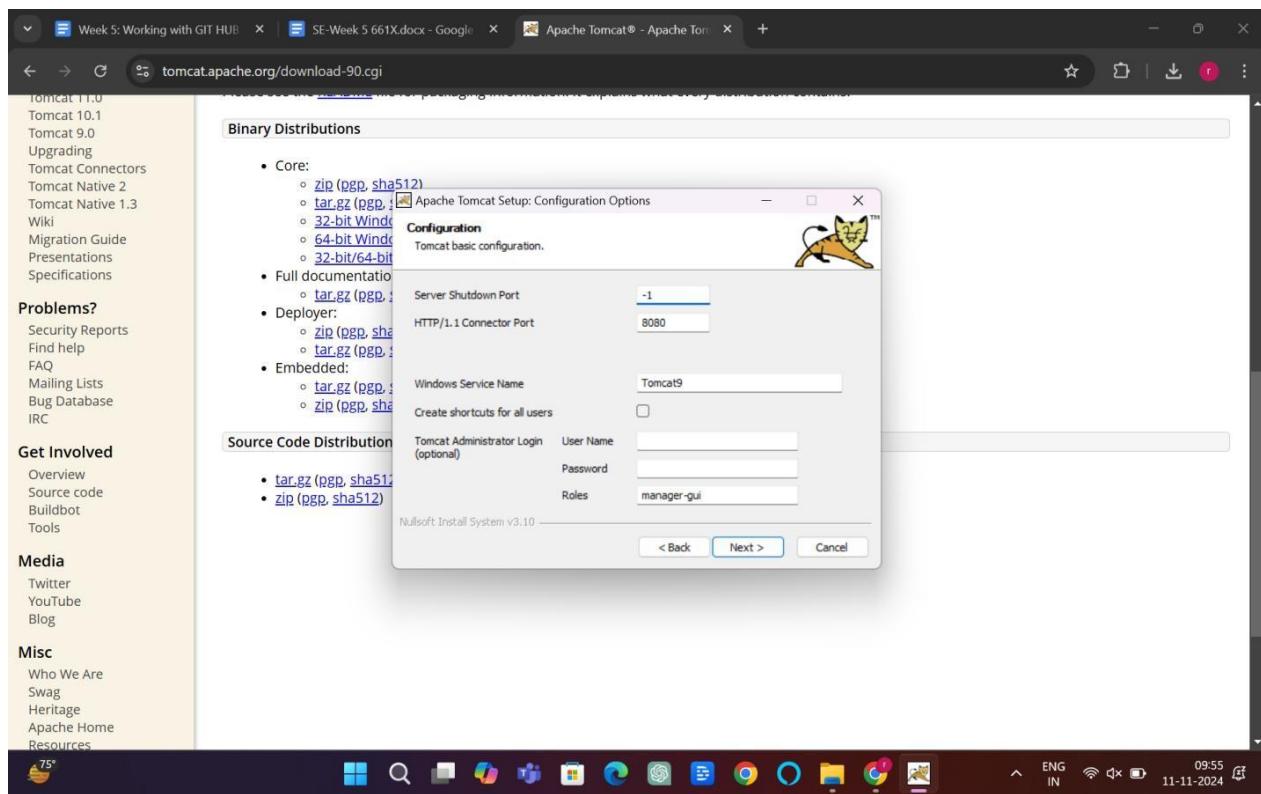
Set the **Tomcat Server** ports:

HTTP/1.1 Connector Port (default is 8080).

If you want to run multiple instances or have another service using 8080, you can change this port.

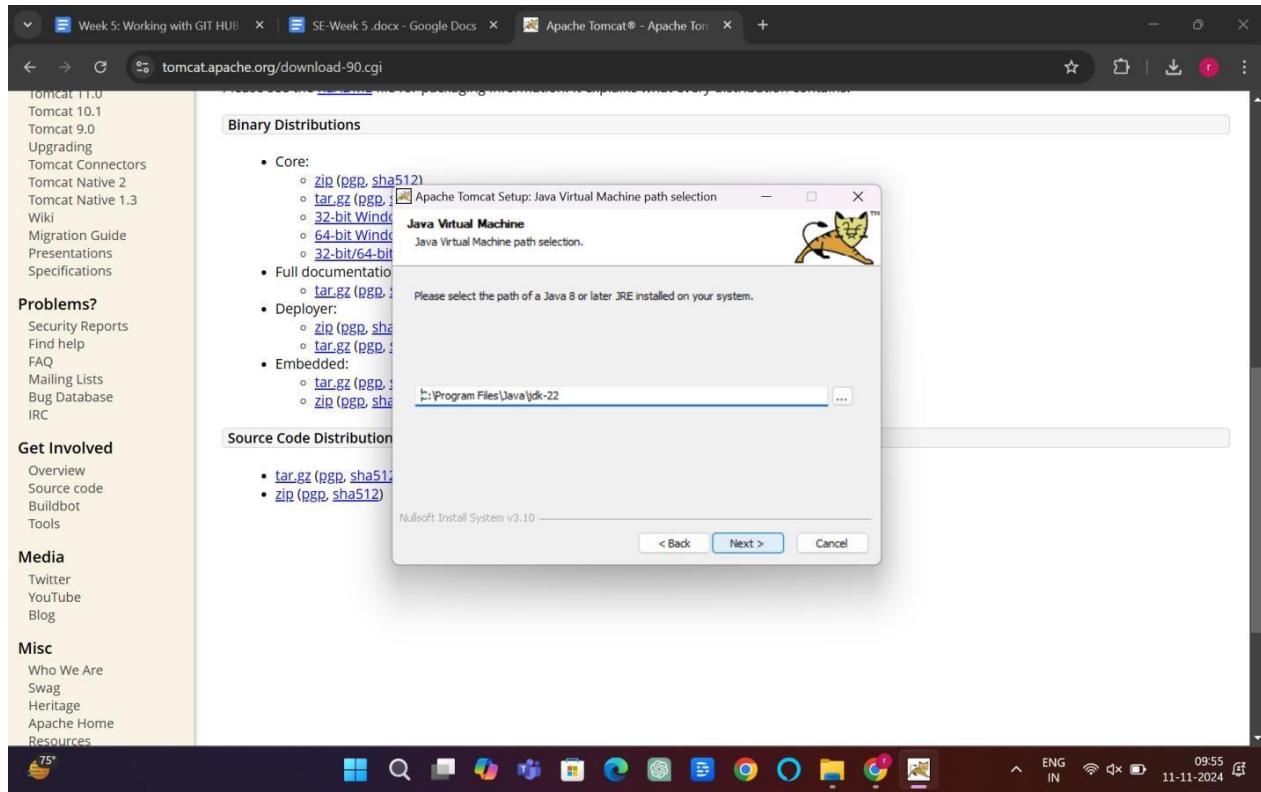
Tomcat Administrator Login: Enter a username and password (this is optional).

Click Next.



Choose Java Virtual Machine:

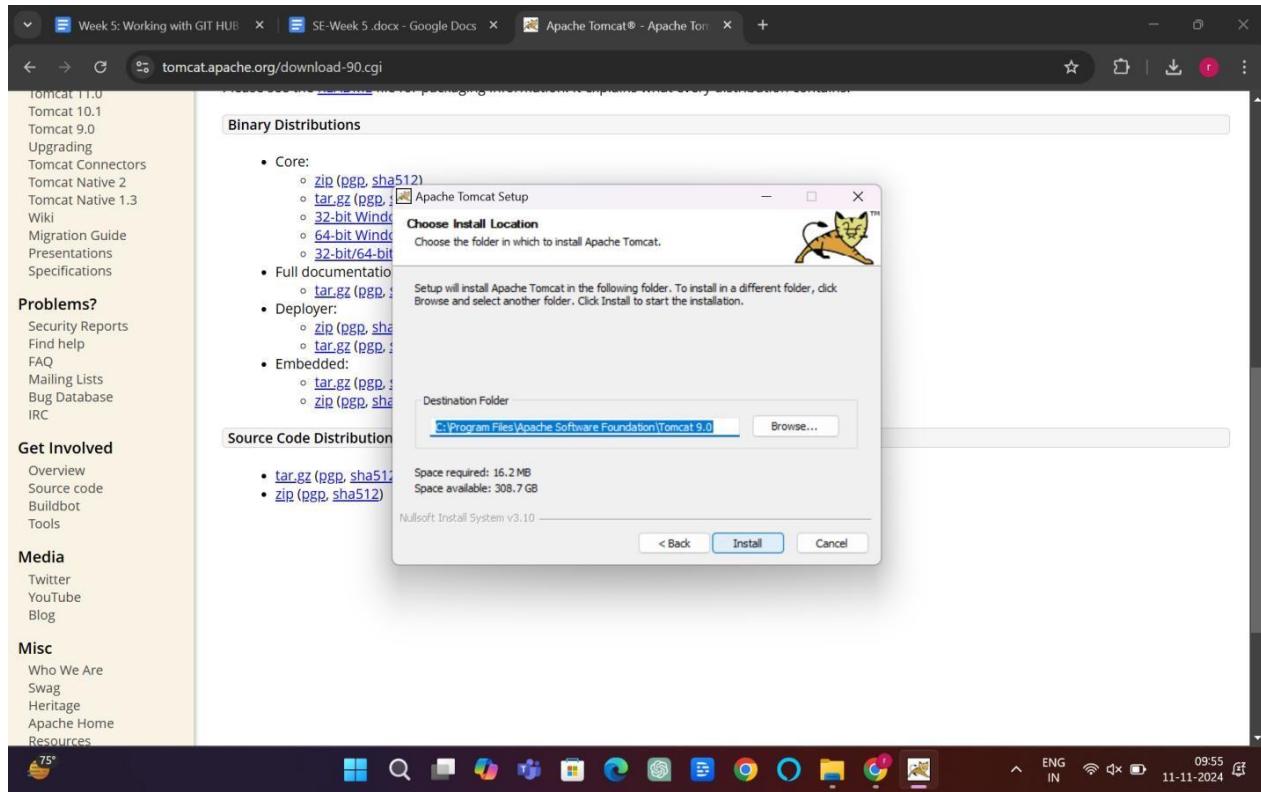
Ensure the correct path to your JDK (Java Development Kit) is set. If you haven't installed the JDK, you'll need to install it first.



Choose Install Location:

Choose the directory where you want to install Tomcat or leave the default location.

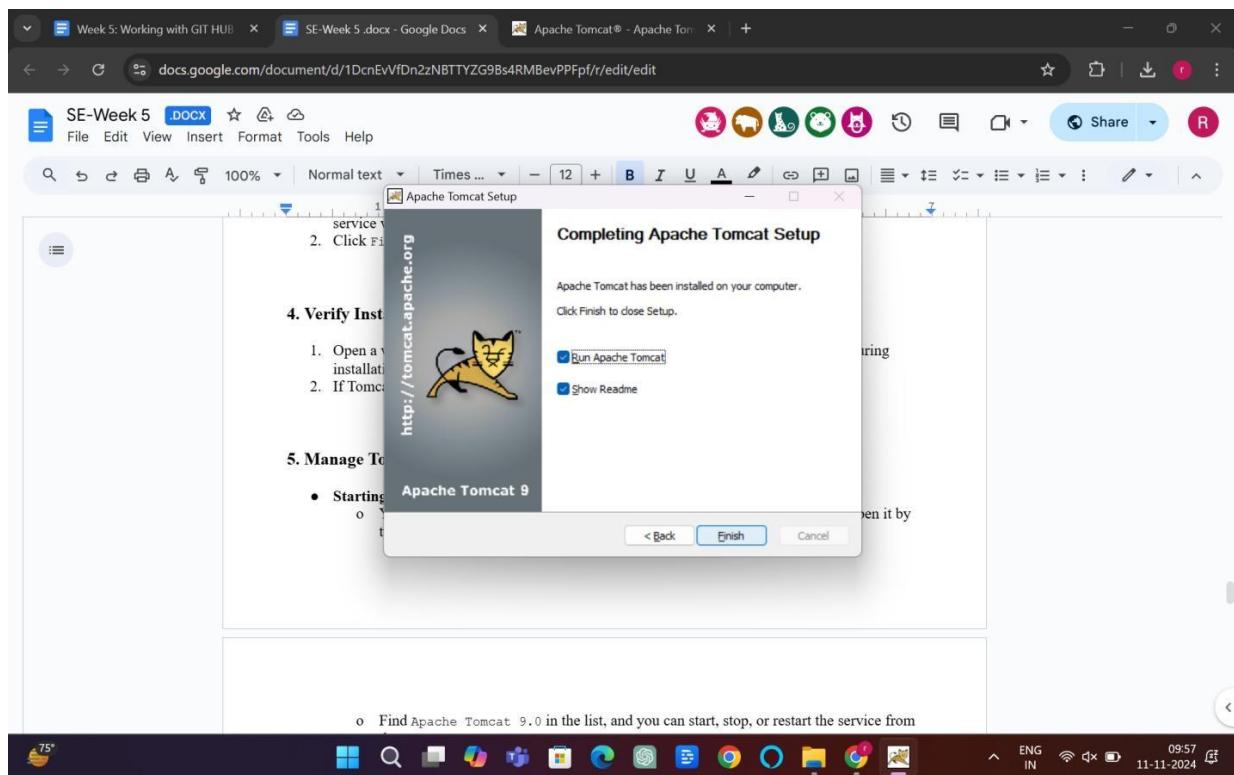
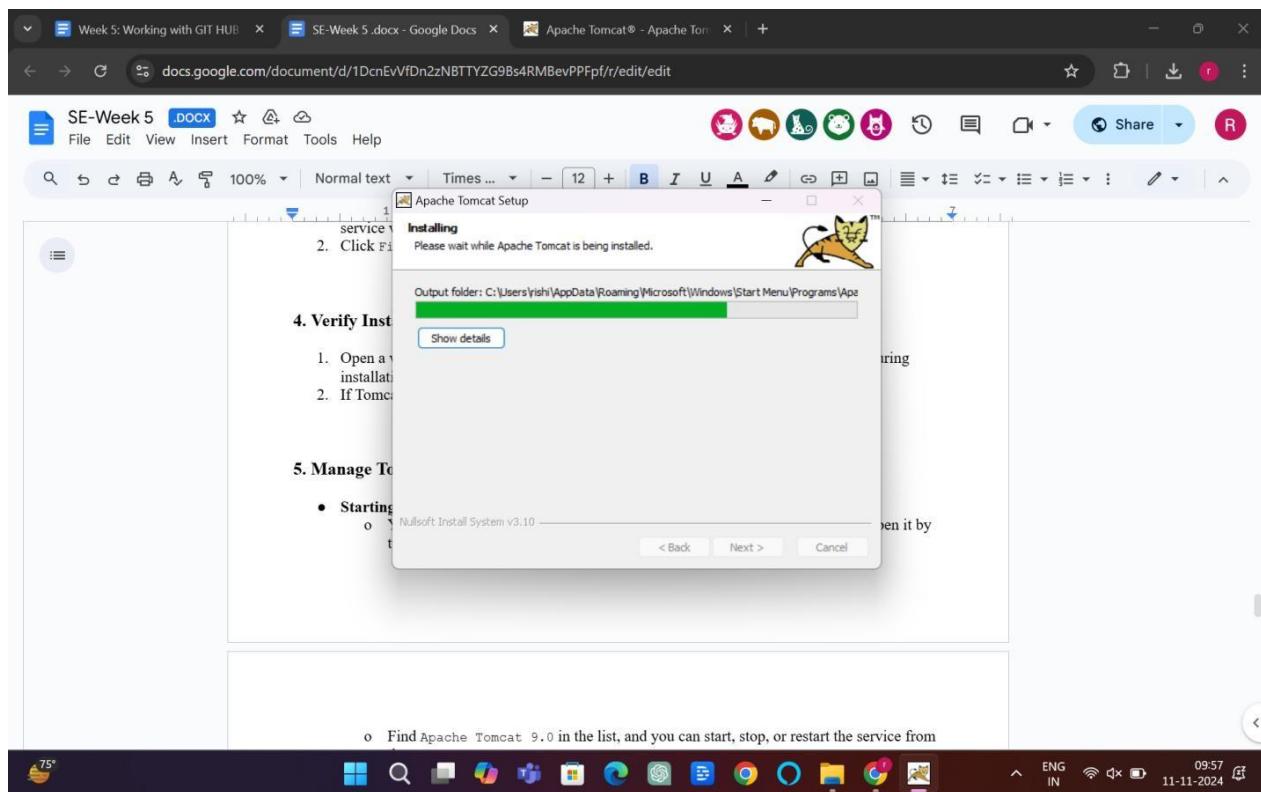
Click **Next >**, then click **Install** to begin the installation.



Complete the Installation

After the installation, you'll have the option to start Tomcat immediately. If selected, the Tomcat service will start.

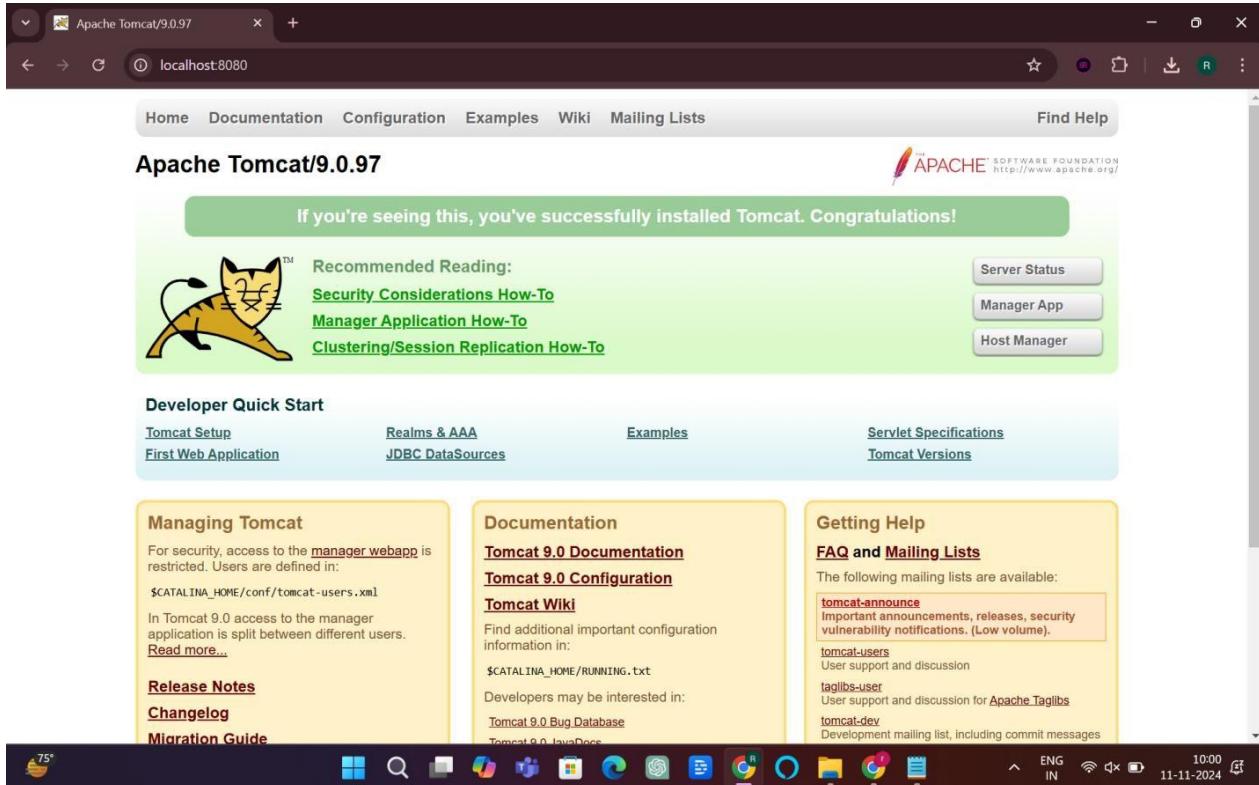
Click **Finish** to exit the installer.



Verify Installation

Open a web browser and go to `http://localhost:8080` (or the port you specified during installation).

If Tomcat is running correctly, you should see the Tomcat homepage.



Manage Tomcat as a Service

Starting/Stopping Tomcat:

You can manage Tomcat from the Windows Services management console. Open it by typing `services.msc` in the Start menu.

Find Apache Tomcat 9.0 in the list, and you can start, stop, or restart the service from there.

Configuring Tomcat:

Tomcat's configuration files are located in the `conf` directory of your Tomcat installation (e.g., `C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf`).

Common files to configure include:

`server.xml`: Configures the main settings of the server.

`web.xml`: Configures web application defaults.

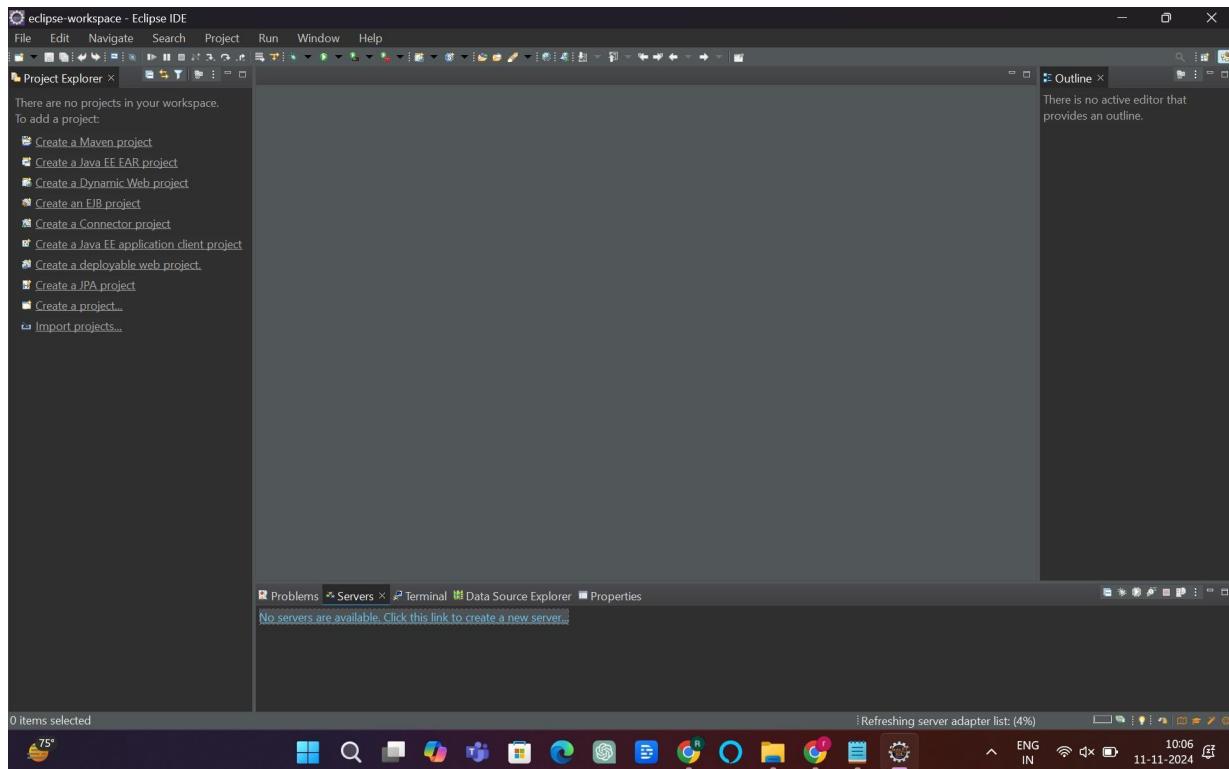
Configure Tomcat Web-Server on Eclipse

Eclipse IDE: Ensure you have Eclipse IDE for Java EE Developers (or any version that supports Java EE). If not, download it from the Eclipse official website.

Apache Tomcat 9.0: Ensure Tomcat 9.0 is installed on your system. If not, download it from the [Apache Tomcat website](#).

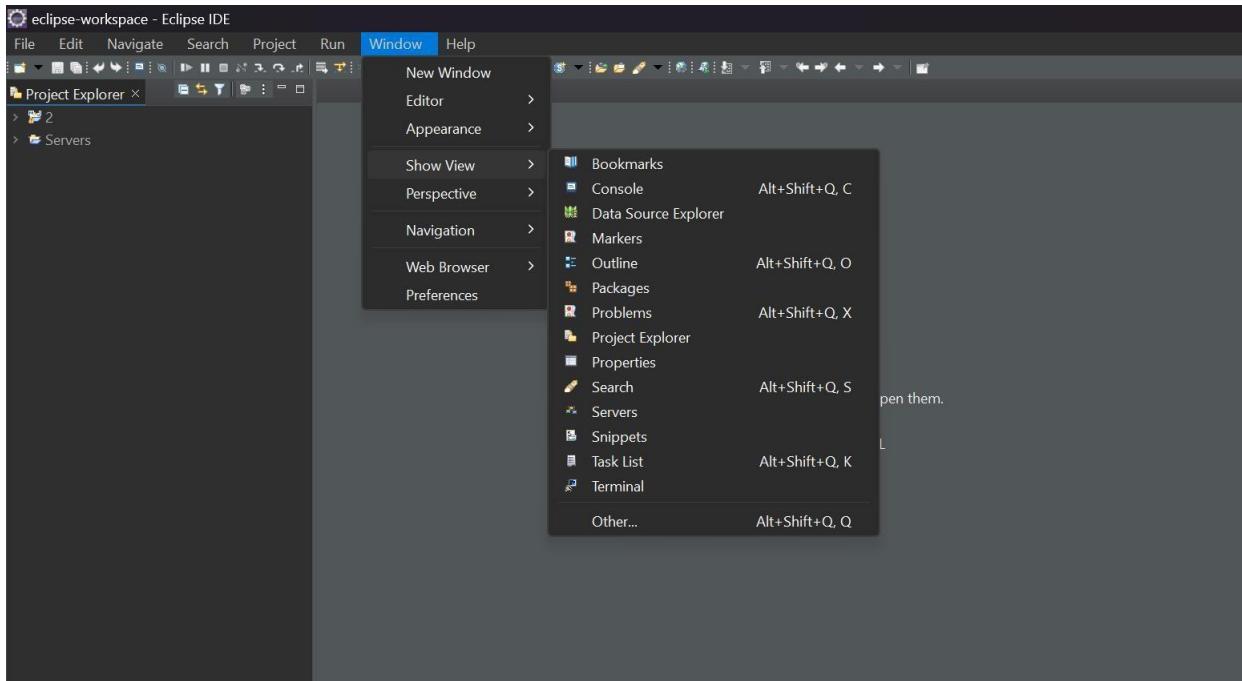
Open Eclipse and Configure the Server

Open Eclipse: Start your Eclipse IDE.

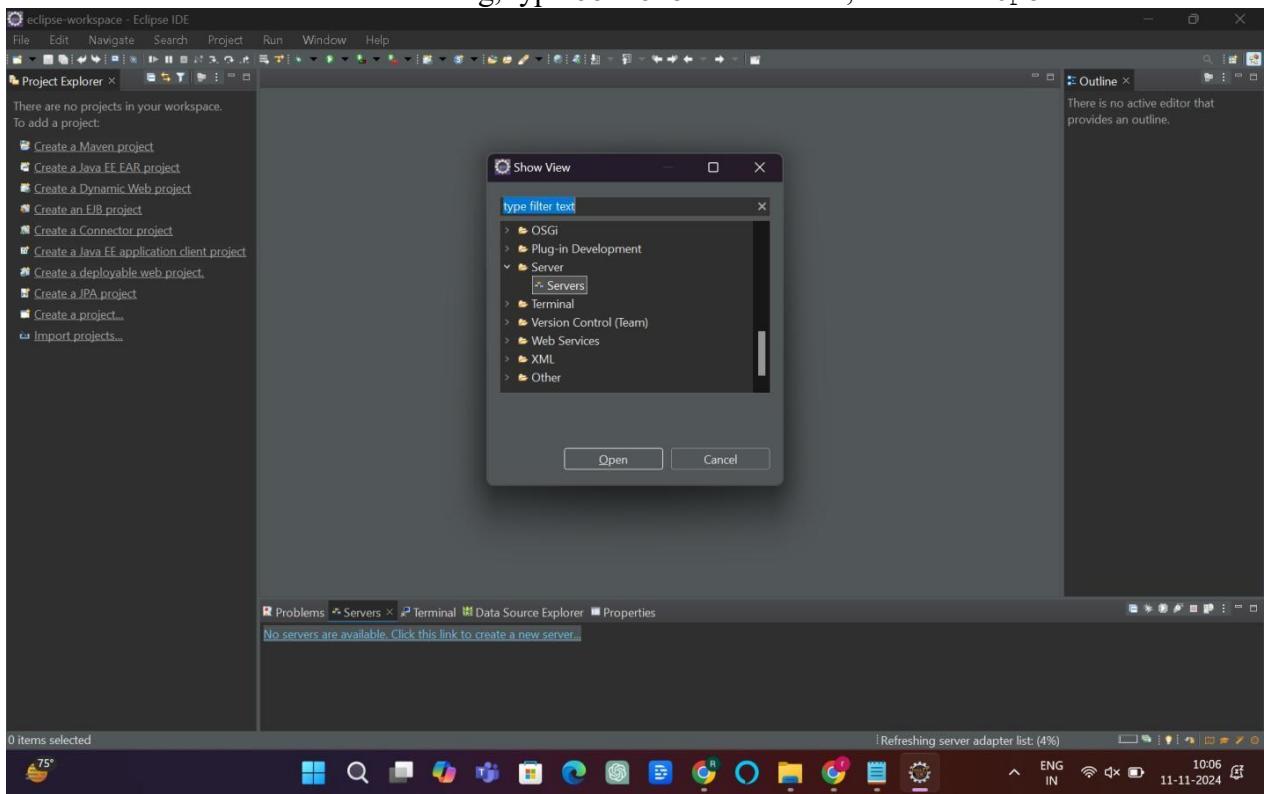


Open the Servers View:

Go to Window > Show View > Other....

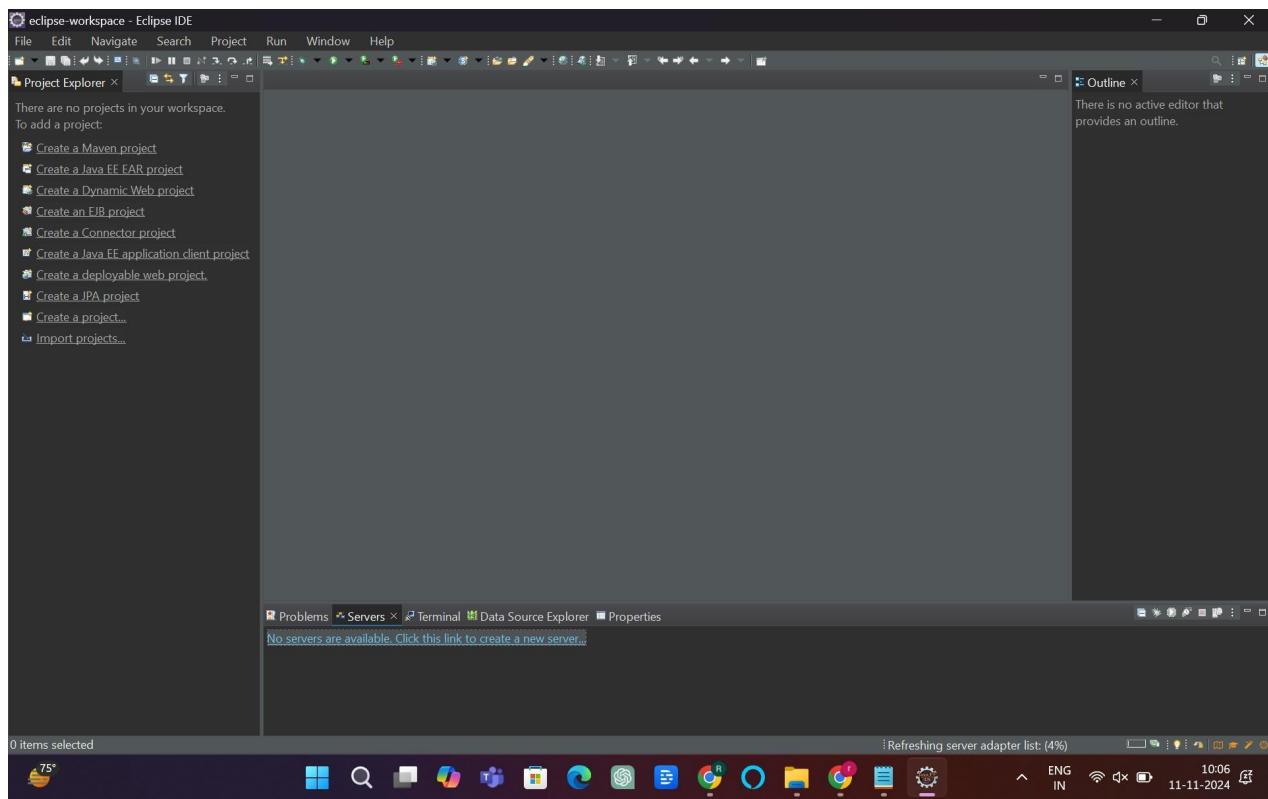


In the "Show View" dialog, type Servers and select it, then click Open.

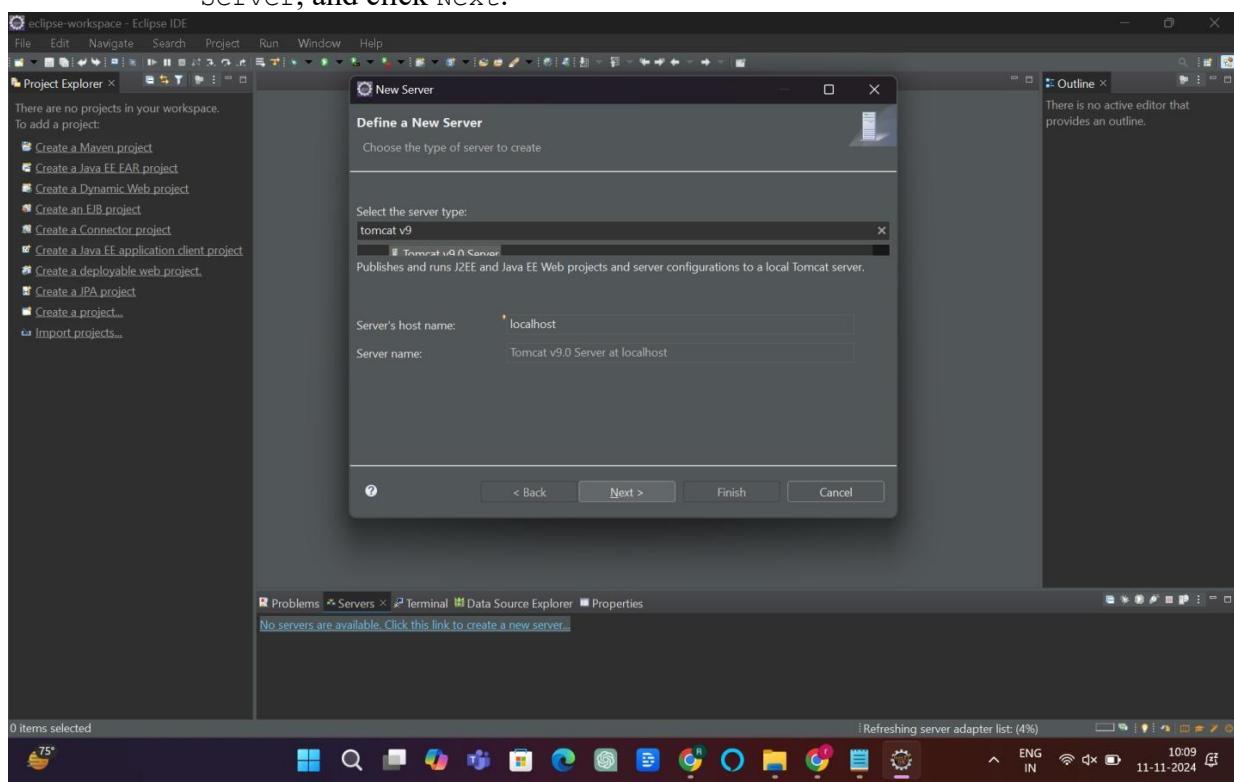


Add a New Server:

In the "Servers" view, right-click and select New > Server.



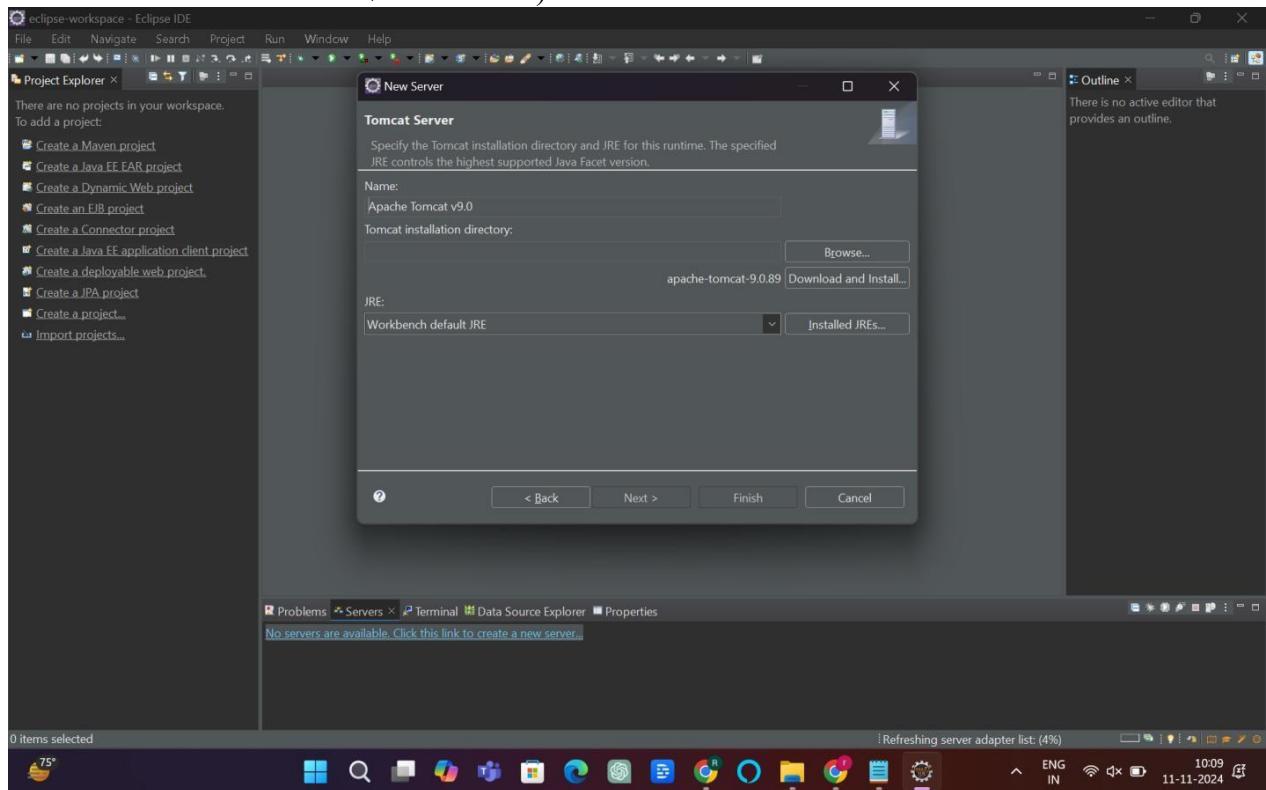
In the "New Server" wizard, expand the "Apache" node, select Tomcat v9.0 Server, and click Next.



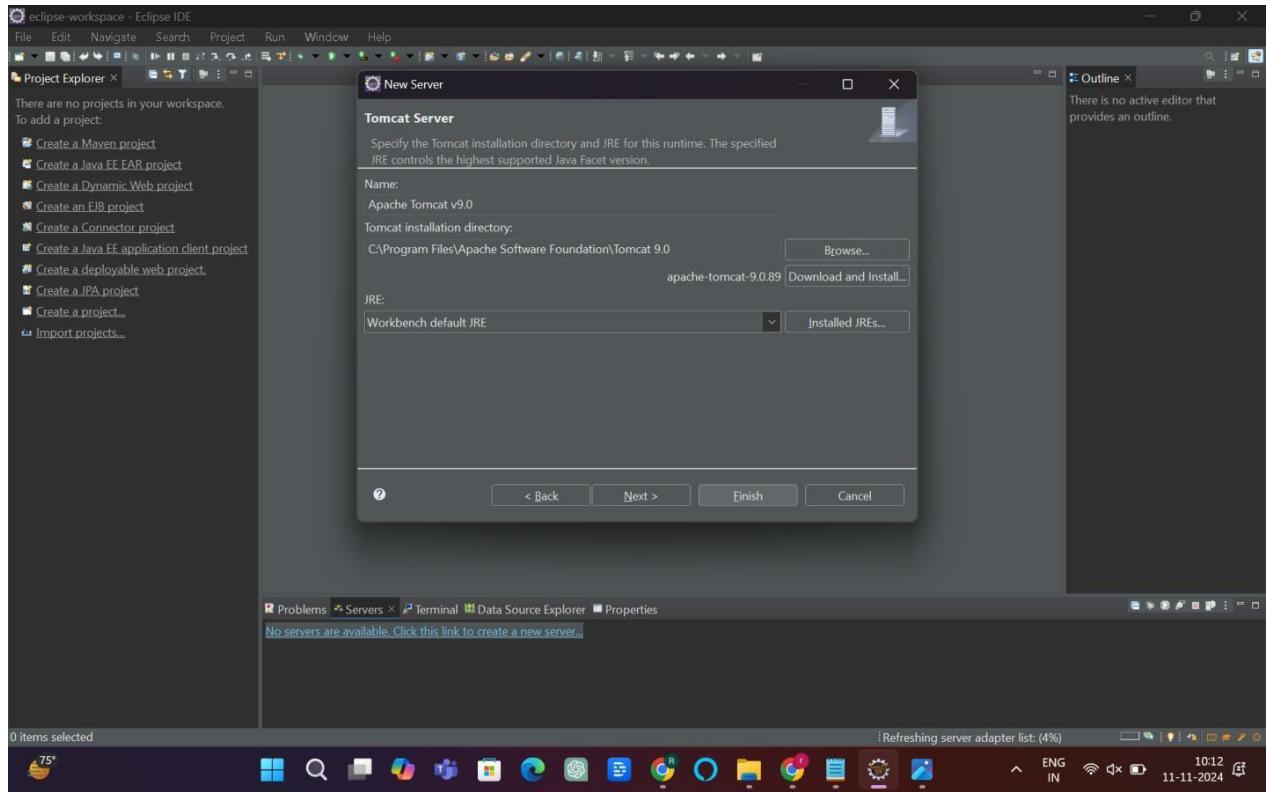
Configure the Tomcat Server:

Server Name: You can leave the default name or provide a custom name.

Tomcat Installation Directory: Click **Browse...** and navigate to the directory where you installed Tomcat (e.g., C:\Program Files\Apache Software Foundation\Tomcat 9.0).



JRE Selection: Ensure that the correct JRE is selected. You can use the default JRE or select another installed JDK.



Add Projects to the Server:

In the next window, you'll have the option to add existing projects to the server. If you have a project you want to run on Tomcat, select it; otherwise, you can add projects later.

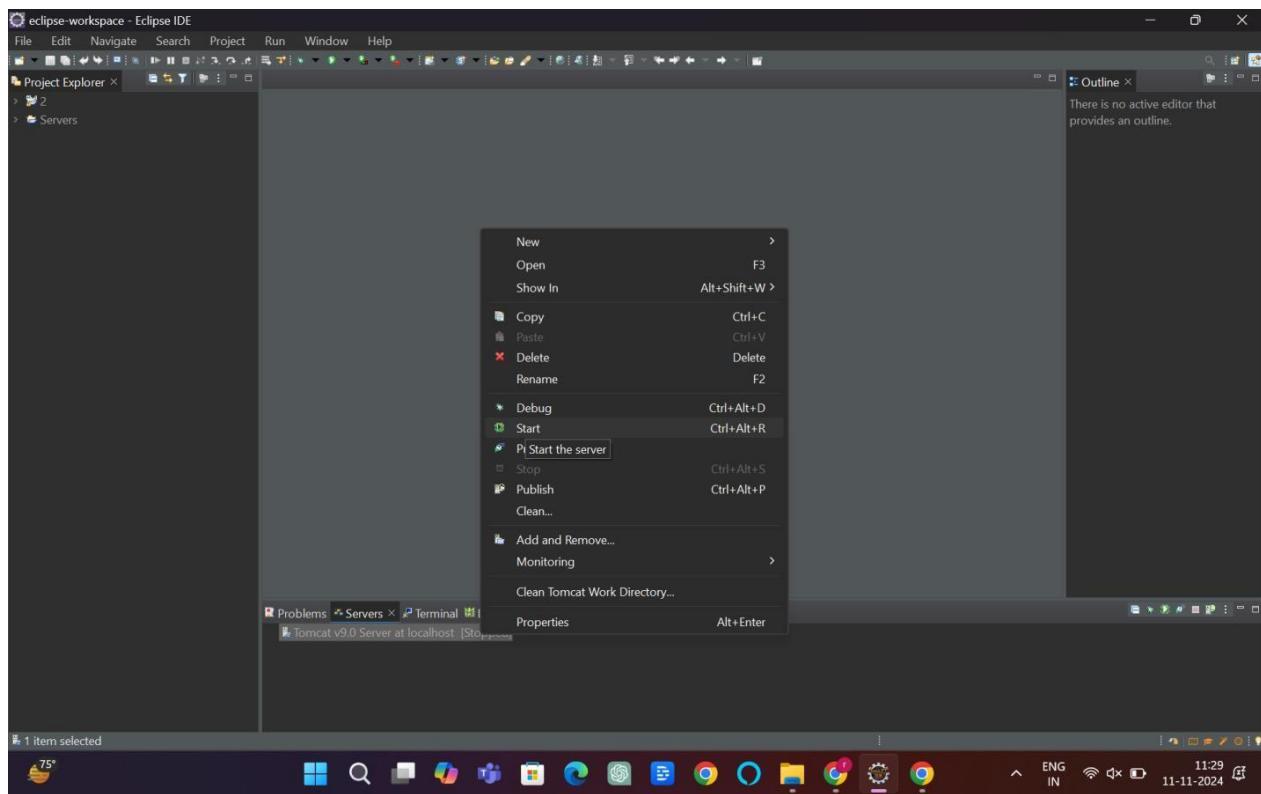
Click **Finish**.

Test the Server Configuration

Start the Server:

In the "Servers" view, right-click on the newly added Tomcat server and select **Start**.

Eclipse will start the Tomcat server. You should see output in the "Console" view indicating that Tomcat is running.



Access Tomcat:

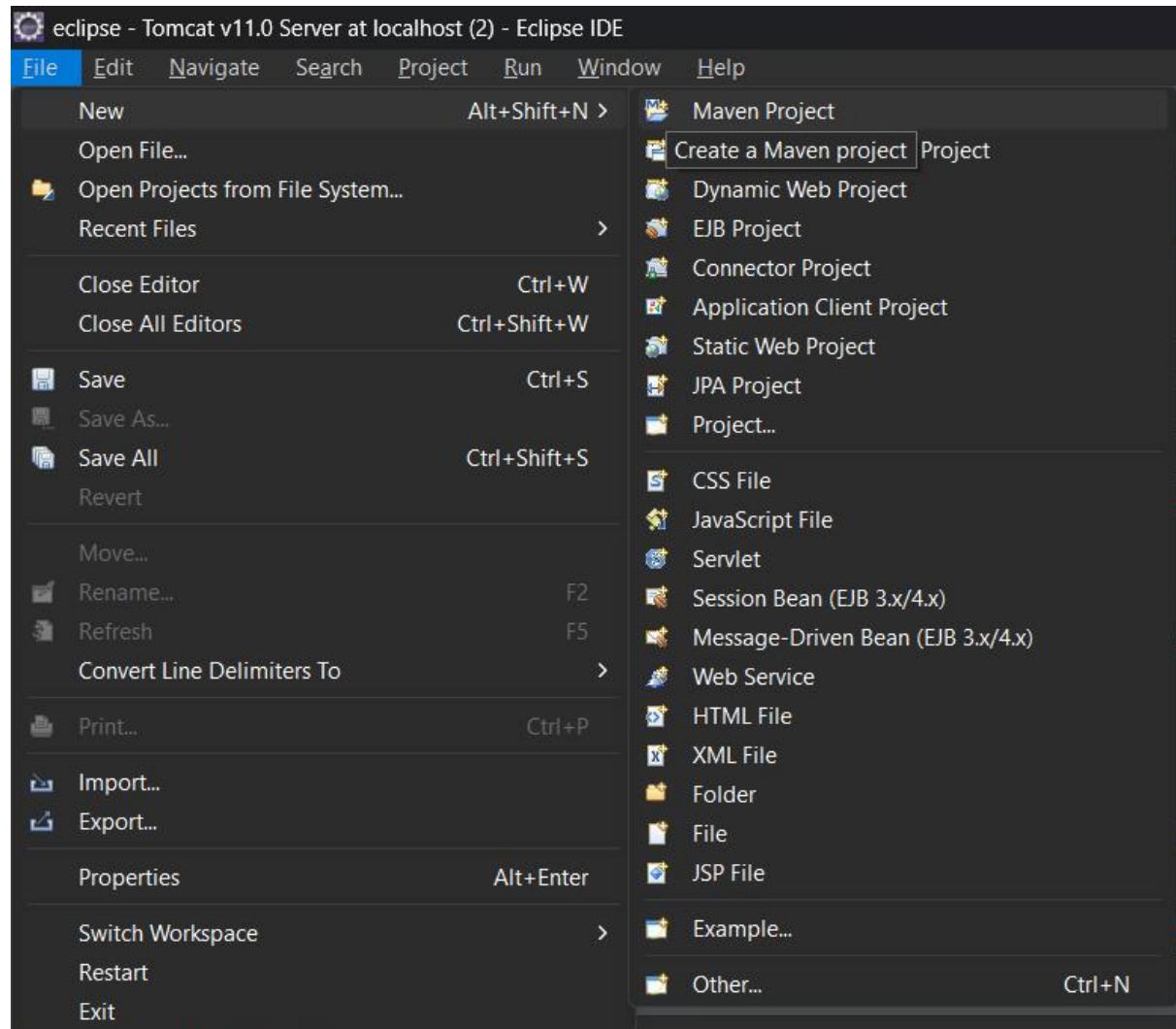
Open a web browser and go to <http://localhost:8080>. You should see the Tomcat default homepage, confirming that the server is running correctly.

The screenshot shows a web browser window with three tabs: 'Week 5: Working with GIT HUB', 'Apache Tomcat® - Apache Tomcat', and 'Apache Tomcat/9.0.97'. The active tab is 'Apache Tomcat/9.0.97', which displays the Tomcat default page. The page features a green header bar with the text 'If you're seeing this, you've successfully installed Tomcat. Congratulations!'. Below this is a cartoon cat icon. To the right of the cat are links for 'Recommended Reading' including 'Security Considerations How-To', 'Manager Application How-To', and 'Clustering/Session Replication How-To'. On the far right are buttons for 'Server Status', 'Manager App', and 'Host Manager'. The main content area includes sections for 'Developer Quick Start' (with links to 'Tomcat Setup', 'First Web Application', 'Realms & AAA', 'JDBC DataSources', 'Examples', 'Servlet Specifications', and 'Tomcat Versions'), 'Managing Tomcat' (with information about restricted access to the manager webapp), '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 available mailing lists like 'tomcat-announce', 'tomcat-users', 'taglibs-user', and 'tomcat-dev'), and 'Release Notes' and 'Changelog' sections. The browser's taskbar at the bottom shows the URL 'localhost:8080/docs/cluster-howto.html'.

CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB.

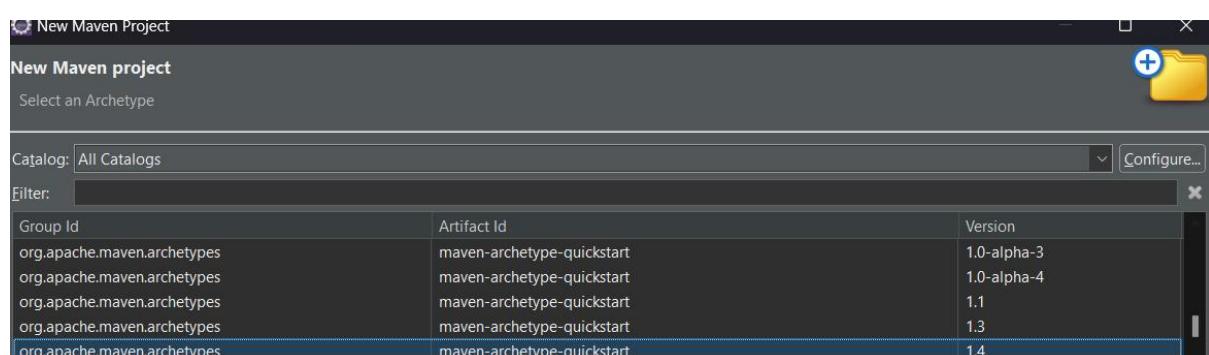
Setting up maven project in eclipse

Step 1: In Eclipse, go to File > New > Maven Project.

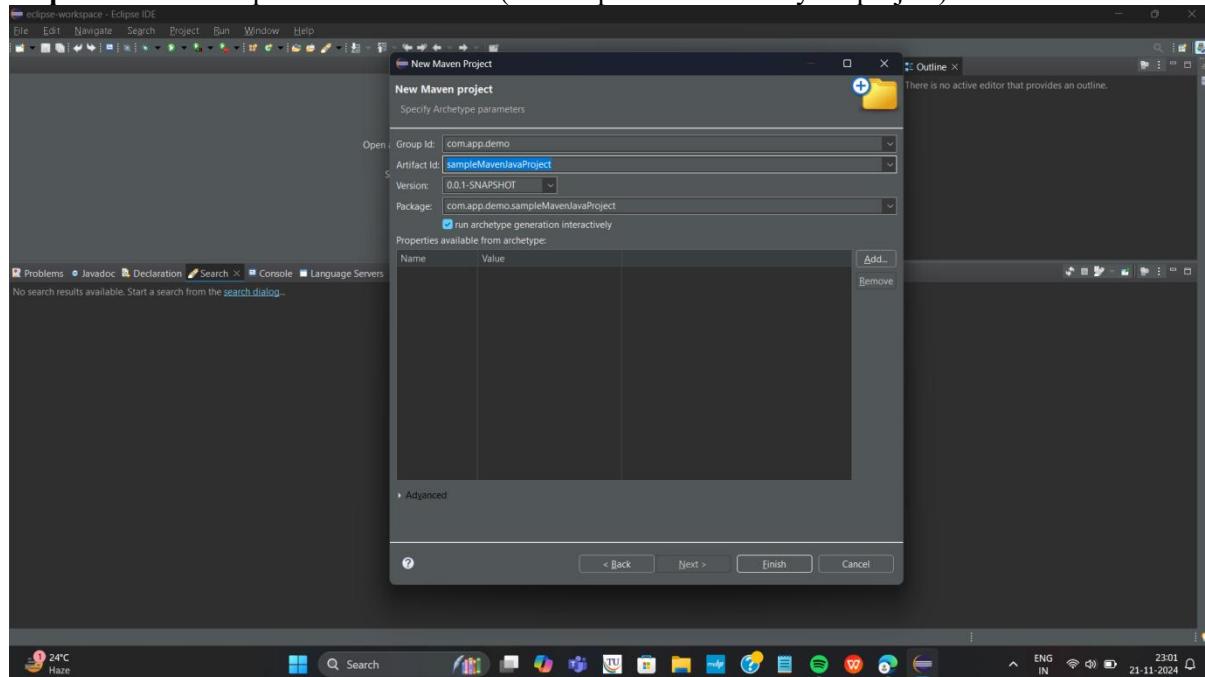


Step 2:

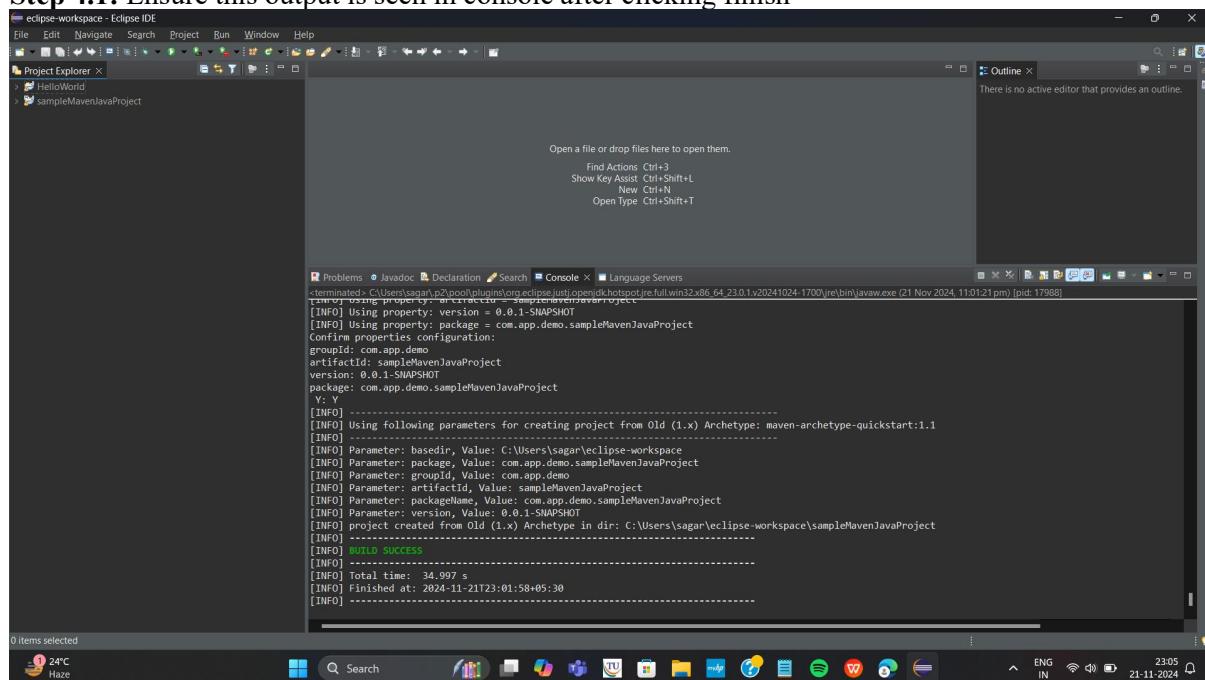
Choose the `quickstart` archetype under `org.apache.maven.archetypes`.



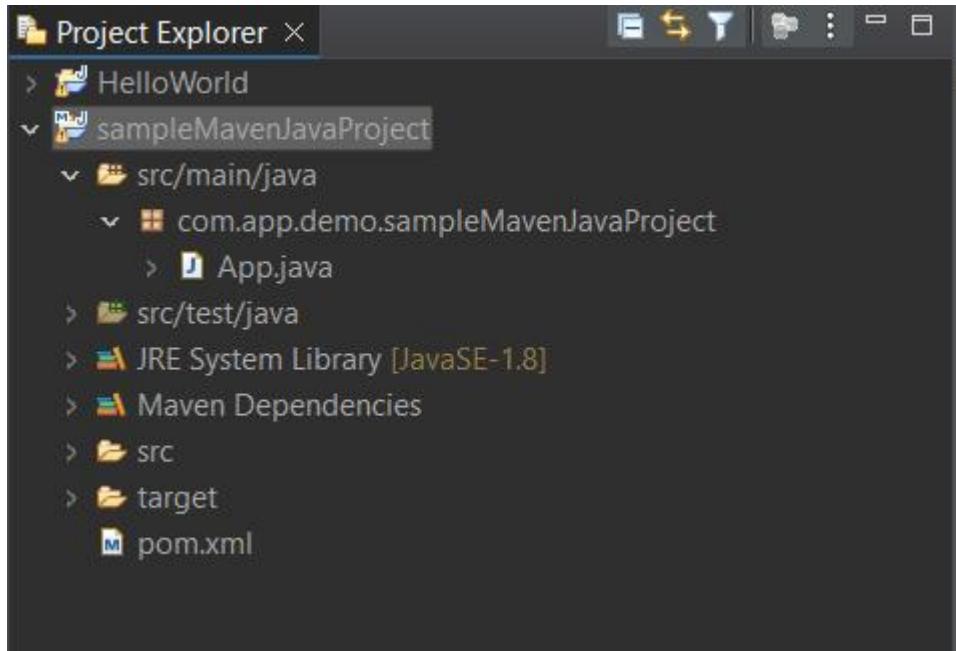
Step 3: Set the Group ID and Artifact ID (the unique identifier for your project).



Step 4.1: Ensure this output is seen in console after clicking finish



Step 4.2: Ensure that the file has been created in *Window>Show View>Project Explorer*



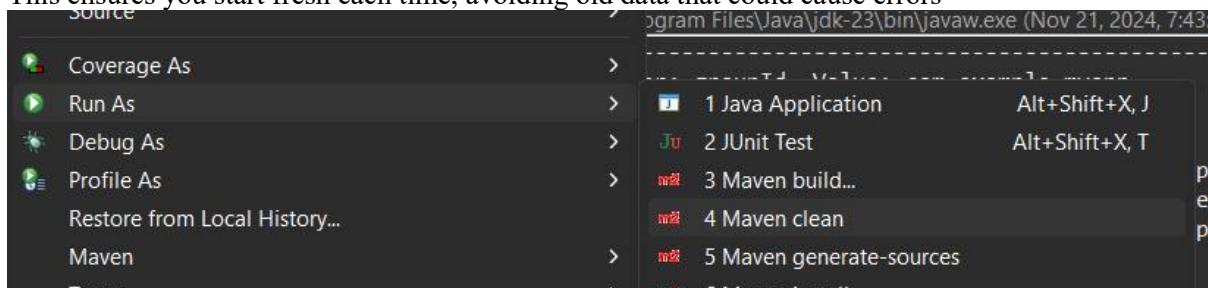
Step 4.3: Ensure you can see the *pom.xml* file and *src/main/java*

```
App.java ×
1 package com.app.demo.sampleMavenJavaProject;
2
3 /**
4  * Hello world!
5  *
6  */
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12    }
13 }
```

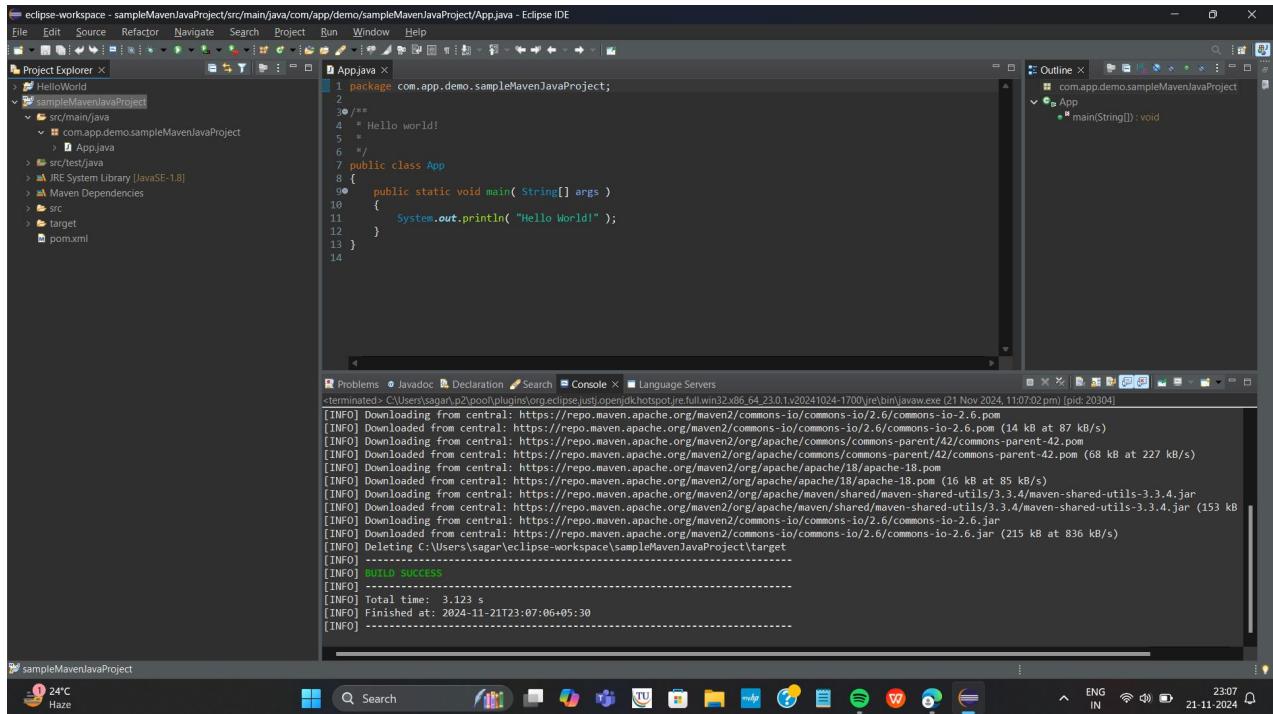
Running Essential Maven Goals

Step 1.1: Right click on folder *my-app* and select *Run As> Maven clean*

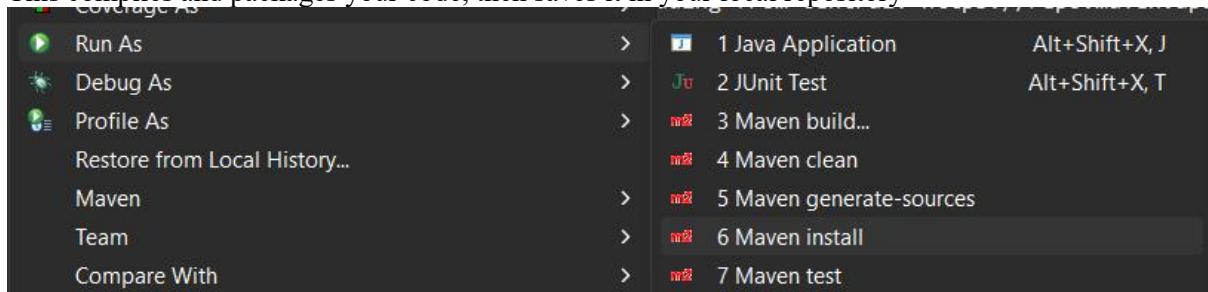
This ensures you start fresh each time, avoiding old data that could cause errors



Step 1.2: Ensure you get “BUILD SUCCESS” in the console



Step 2.1: Right click on project folder and select *Run As>Maven Install*
This compiles and packages your code, then saves it in your local repository

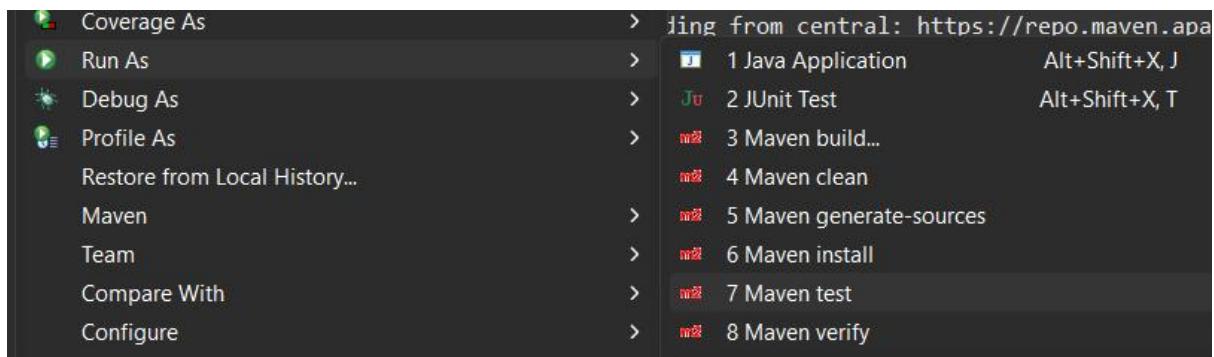


Step 2.2: Ensure you get “BUILD SUCCESS” in the console



Step 3.1:

Step 3.1: Right click on project folder and select *Run As>Maven Test*
This runs tests to ensure your code behaves as expected

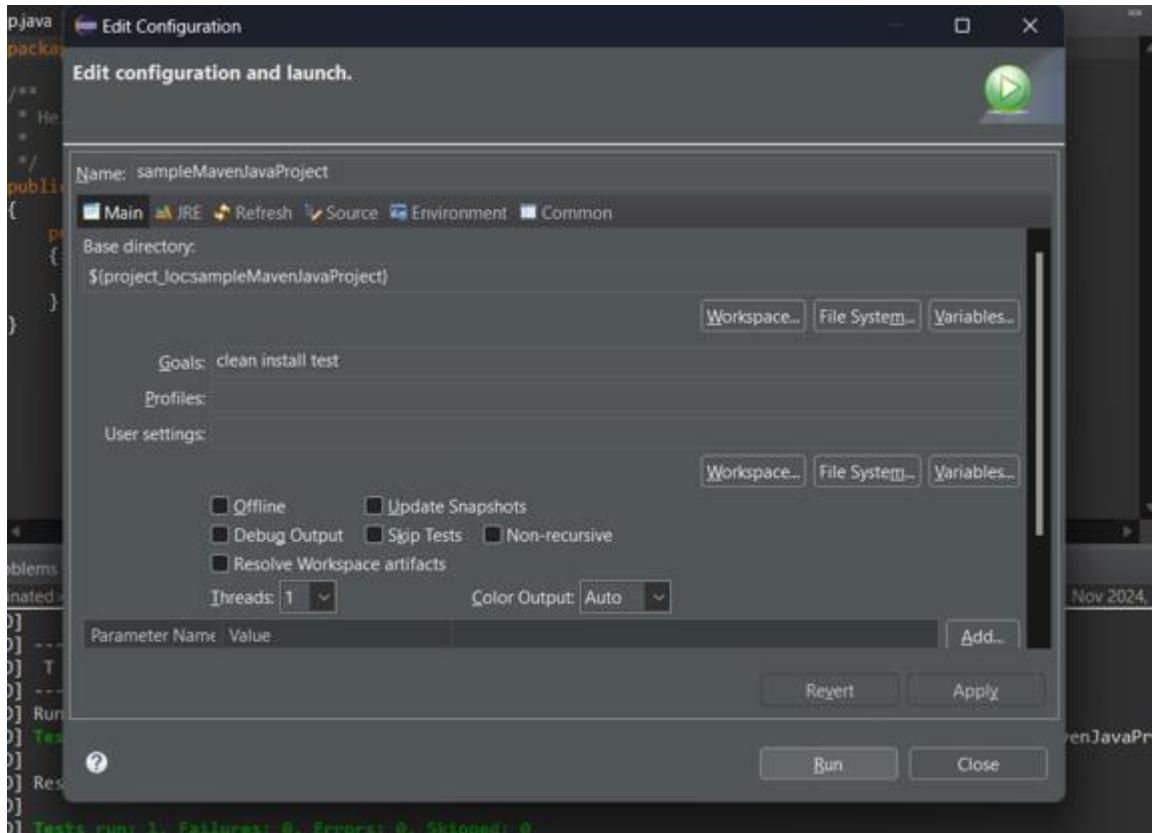


Step 3.2: Ensure you get “BUILD SUCCESS” in the console

```
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.app.demo.sampleMavenJavaProject.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.040 s -- in com.app.demo.sampleMa
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.893 s
[INFO] Finished at: 2024-11-21T23:08:28+05:30
[INFO] -----
```

Step 4.1: Right click on project folder and select *Run As>Maven build...*

Step 4.1: A window opens up and enter Clean Install Test in the Goals field to run all three tasks sequentially.



Step 4.2: Ensure you get “BUILD SUCCESS” in the console

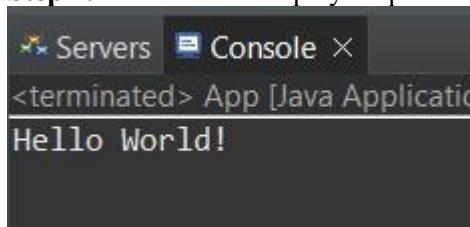
```
[INFO] [INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  3.067 s
[INFO] Finished at: 2024-11-21T20:03:38+05:30
[INFO] -----
```

Running the Project to View Output

Step 1: Right click the project, select *Run As>Java Application*

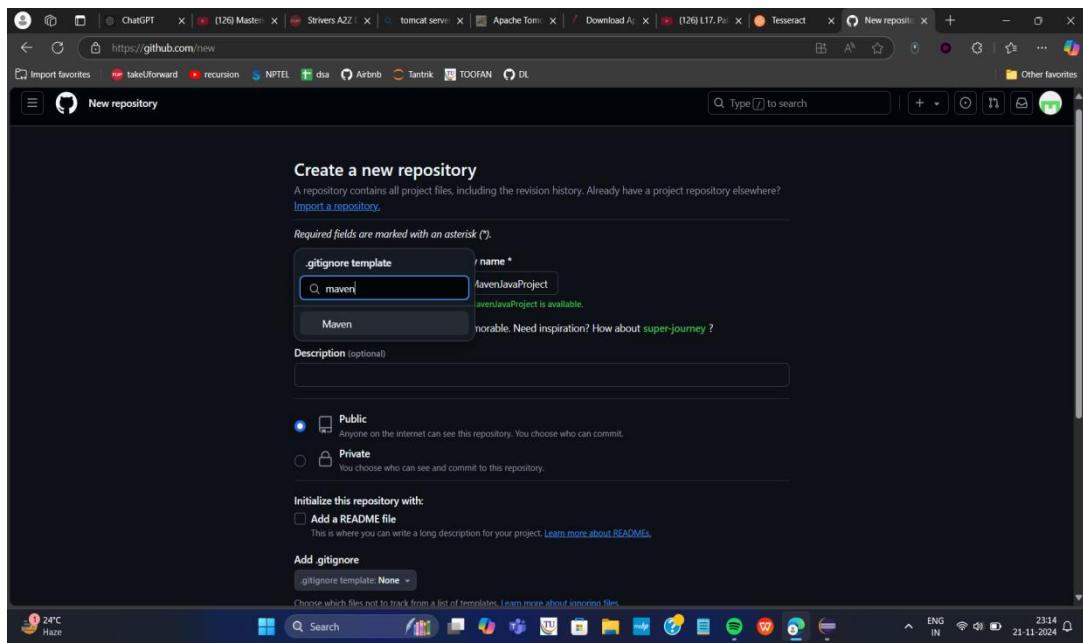


Step 2: Console will display output



Pushing the Project to GitHub

Step 1 & 2: Go to your GitHub account and create new repository, next select Maven under `.gitignore`



Step 3: Copy the HTTP link from GitHub to link your local project and clone it using git bash terminal

```
sagar@sAGARPUPPALA MINGW64 ~ (master)
$ git clone https://github.com/Puppala-Sagar/SampleMavenJavaProject.git
Cloning into 'SampleMavenJavaProject'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Step 4: Switch to the cloned folder

```
sagar@sAGARPUPPALA MINGW64 ~ (master)
$ cd SampleMavenJavaProject

sagar@sAGARPUPPALA MINGW64 ~/SampleMavenJavaProject (main)
$ |
```

Step 5: Locate the folder of your Eclipse Project folder, here named *my-app*

File Explorer				
C:\ > This PC > Windows (C:) > Users > sagar > eclipse-workspace > sampleMavenJavaProject >				
Name	Date modified	Type	Size	...
.settings	21-11-2024 23:02	File folder		
src	21-11-2024 23:01	File folder		
target	21-11-2024 23:09	File folder		
.classpath	21-11-2024 23:02	CLASSPATH File	2 KB	
.project	21-11-2024 23:02	PROJECT File	1 KB	
pom.xml	21-11-2024 23:01	xmlfile	1 KB	

Step 6: Copy files from the project folder (*my-app*) into cloned repository

Name	Date modified	Type	Size
.git	21-11-2024 23:15	File folder	
.gitignore	21-11-2024 23:15	Git Ignore Source ...	1 KB
.settings	21-11-2024 23:16	File folder	
src	21-11-2024 23:16	File folder	
target	21-11-2024 23:16	File folder	
.classpath	21-11-2024 23:02	CLASSPATH File	2 KB
.project	21-11-2024 23:02	PROJECT File	1 KB
pom.xml	21-11-2024 23:01	xmlfile	1 KB

Step 7:

Stage the changes

```
git add .
```

Commit the changes

```
git commit -m "message"
```

Push to your GitHub repo

```
git push
```

```
sagar@SAGARPUPPALA MINGW64 ~/SampleMavenJavaProject (main)
$ git add .

sagar@SAGARPUPPALA MINGW64 ~/SampleMavenJavaProject (main)
$ git commit -m "Sample Maven Java Project"
[main 37a15ce] Sample Maven Java Project
 6 files changed, 92 insertions(+)
 create mode 100644 .settings/org.eclipse.core.resourcesprefs
 create mode 100644 .settings/org.eclipse.jdt.core.preferences
 create mode 100644 .settings/org.eclipse.m2e.core.preferences
 create mode 100644 pom.xml
 create mode 100644 src/main/java/com/app/demo/sampleMavenJavaProject/App.java
 create mode 100644 src/test/java/com/app/demo/sampleMavenJavaProject/AppTest.java

sagar@SAGARPUPPALA MINGW64 ~/SampleMavenJavaProject (main)
$ git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (22/22), 2.17 KiB | 2.17 MiB/s, done.
Total 22 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Puppala-Sagar/SampleMavenJavaProject.git
 25a8ec2..37a15ce main -> main
```

Step 8: Refresh GitHub to verify the files have been uploaded successfully

The screenshot shows a GitHub repository page for 'SampleMavenJavaProject'. The repository is public and has 1 branch and 0 tags. It contains files like .settings, src, .gitignore, and pom.xml. The README file is present. The repository has 2 commits, 0 stars, 1 watcher, and 0 forks. There are sections for Activity, Releases, and Packages.

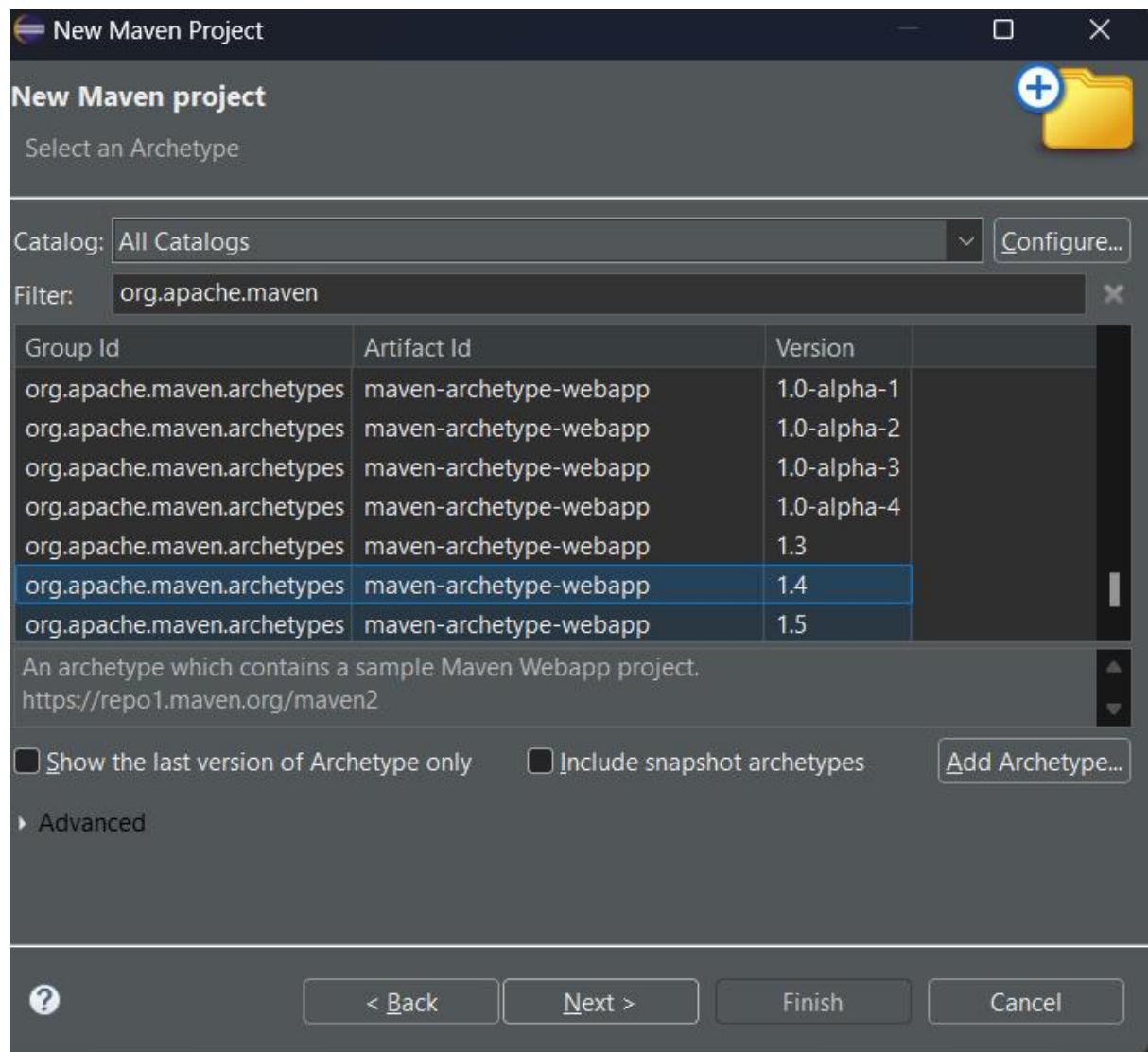
CREATING MAVEN WEB PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB.

CREATE MAVEN WEB PROJECT

The screenshot shows the Eclipse IDE interface with the 'File' menu open. The 'New' submenu is displayed, showing various project and file types. The 'Create a Maven project' option is highlighted.

- New
 - Maven Project
 - Create a Maven project
 - Dynamic Web Project
 - EJB Project
 - Connector Project
 - Application Client Project
 - Static Web Project
 - JPA Project
 - Project...
- CSS File
- JavaScript File
- Servlet
- Session Bean (EJB 3.x/4.x)
- Message-Driven Bean (EJB 3.x/4.x)
- Web Service
- HTML File
- XML File
- Folder
- File
- JSP File
- Example...
- Other...

ChooseFilter



ADD GROUP ID AND ARTIFACT ID AND CLICK ON FINISH

OPEN PROJECT EXPLORER AND OPEN INDEX.JSP and POM.XML

```

1 Appjava index.jsp SampleMavenWebProject/pom.xml
2 <?xml version="1.0" encoding="UTF-8"?>
3
4<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
6   <modelVersion>4.0.0</modelVersion>
7
8   <groupId>com.app.demo</groupId>
9   <artifactId>SampleMavenWebProject</artifactId>
10  <version>0.0.1-SNAPSHOT</version>
11  <packaging>war</packaging>
12
13  <name>SampleMavenWebProject Maven Webapp</name>
14  <!-- FIXME change it to the project's website -->
15  <url>http://www.example.com</url>
16
17<properties>
18   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
19   <maven.compiler.source>1.7</maven.compiler.source>
20   <maven.compiler.target>1.7</maven.compiler.target>
21

```

ADD SERVELET DEPENDENCY

This artifact was moved to:

javax.servlet > javax.servlet-api

Central (7)	Jenkins Releases (1)	Nuxeo (1)	EmergencyPub (3)	ICM (6)
3.0.x	3.0-alpha-1			
2.5.x	2.5			
	2.4			
2.4.x	2.4.public_draft			
2.4.x	2.4-20040521			
2.3.x	2.3			
2.2.x	2.2			

```

25      <artifactId>junit</artifactId>
26      <version>4.11</version>
27      <scope>test</scope>
28  </dependency>
29
30  <!-- https://mvnrepository.com/artifact/javax.servlet/servlet-api -->
31
32<dependency>
33  <groupId>javax.servlet</groupId>
34  <artifactId>servlet-api</artifactId>
35  <version>2.5</version>

```

MAVEN CYCLE

MAVEN CLEAN

Context menu for Maven project:

- Coverage As
- Run As
- Debug As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure

Submenu for Maven:

- 1 Java Application Alt+Shift+X, J
- 2 JUnit Test Alt+Shift+X, T
- 3 Maven build...
- 4 Maven clean**
- 5 Maven generate-sources
- 6 Maven install
- 7 Maven test
- 8 Maven verify

```

[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/19/a
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/19/a
[INFO]
[INFO] --- clean:3.1.0:clean (default-clean) @ SampleMavenWebProject ---
[INFO] Deleting C:\Users\sagar\eclipse-workspace\SampleMavenWebProject\target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  1.639 s
[INFO] Finished at: 2024-11-21T23:45:39+05:30
[INFO]

```

2 MAVEN INSTALL

Context menu for Maven project:

- Run As
- Debug As
- Restore from Local History...
- Maven
- Team
- Compare With
- Configure
- Properties Alt+Enter

Submenu for Maven:

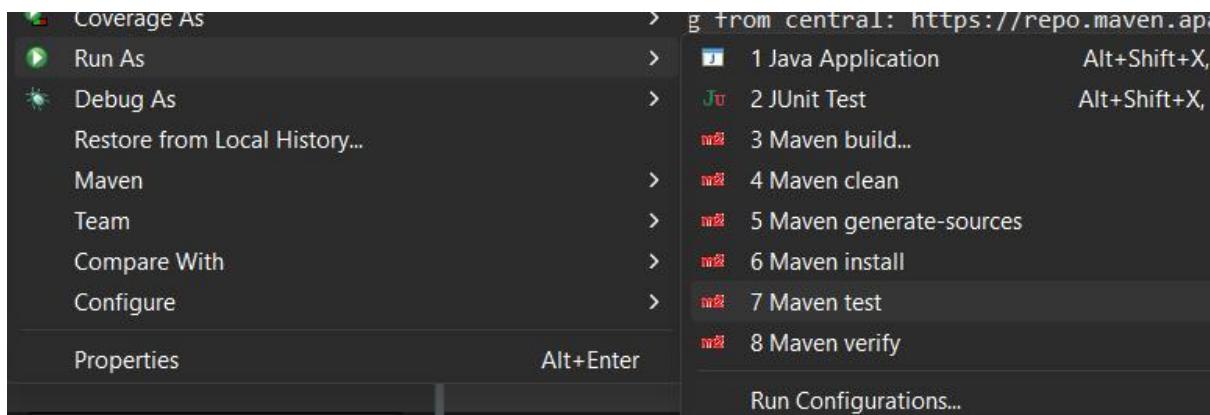
- 1 Java Application Alt+Shift+X, J
- 2 JUnit Test Alt+Shift+X, T
- 3 Maven build...
- 4 Maven clean
- 5 Maven generate-sources
- 6 Maven install**
- 7 Maven test
- 8 Maven verify

```

INFO] Installing C:\Users\sagar\eclipse-workspace\SampleMavenWebProject\target\SampleMavenWeb
INFO] Installing C:\Users\sagar\eclipse-workspace\SampleMavenWebProject\pom.xml to C:\Users\s
INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  14.526 s
[INFO] Finished at: 2024-11-21T23:46:58+05:30
[INFO]

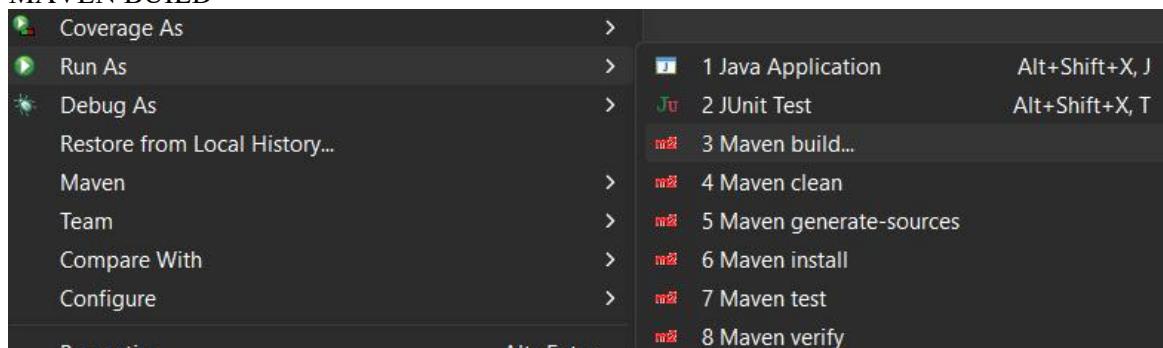
```

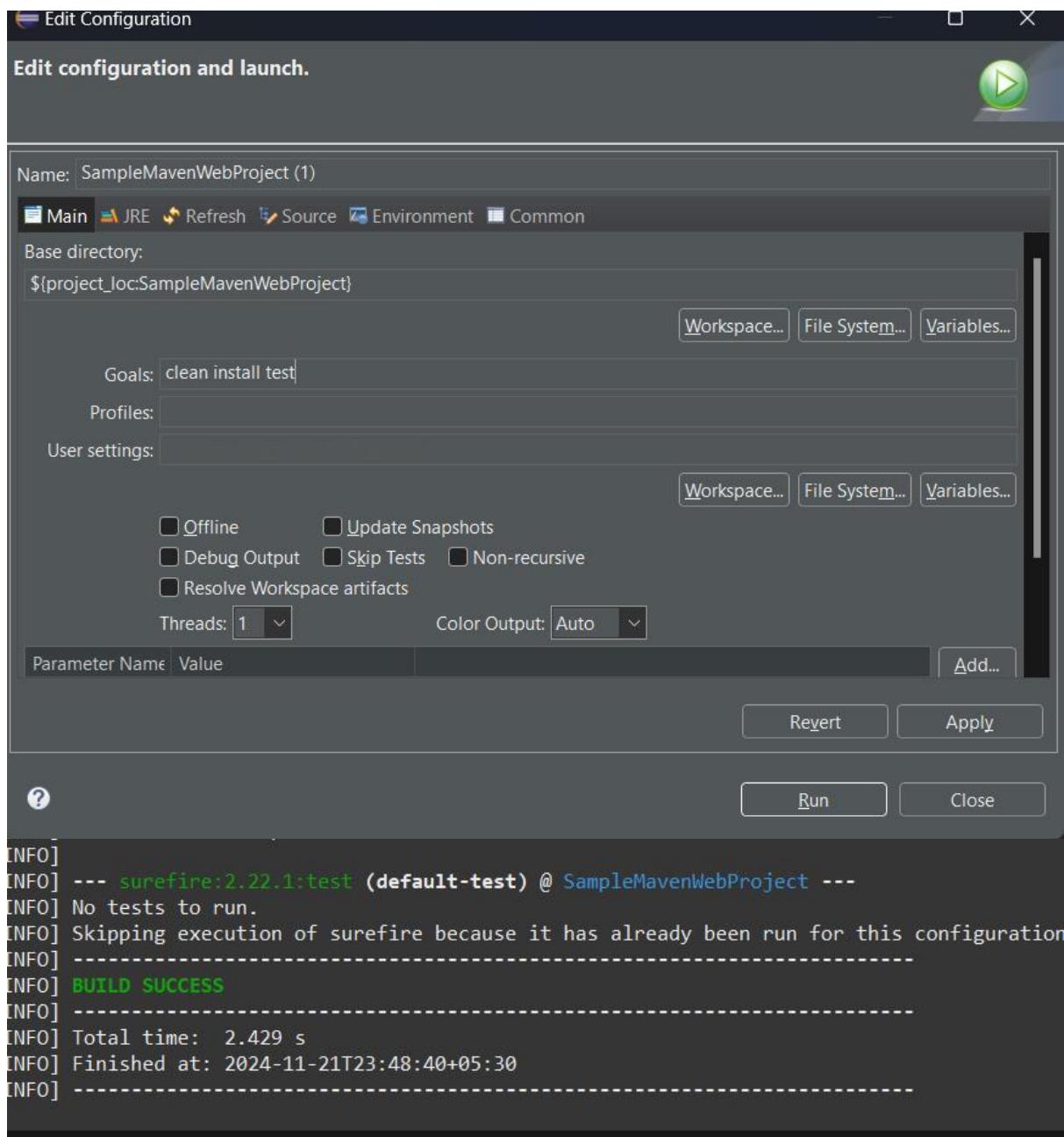
3 MAVEN TEST



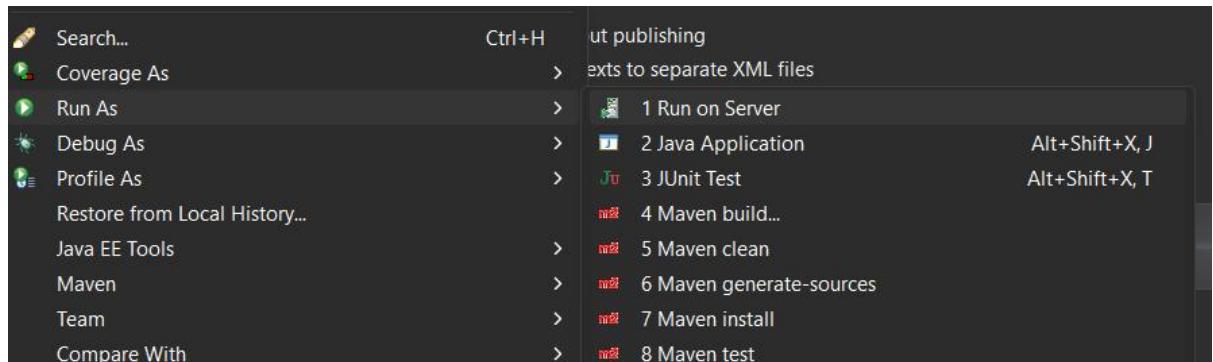
```
[INFO] --- surefire:2.22.1:test (default-test) @ SampleMavenWebProject ---
[INFO] No tests to run.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.480 s
[INFO] Finished at: 2024-11-21T23:47:32+05:30
[INFO] -----
```

MAVEN BUILD





NOW RUN THE SERVER



CHOOSE TOMCAT AS SERVER

Run On Server

Select which server to use

How do you want to select the server?

Choose an existing server
 Manually define a new server

Select the server that you want to use:

type filter text

Server	State
localhost	
Tomcat v9.0 Server at localhost	Started

Apache Tomcat v9.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, and 8 Web modules. [Columns...](#)

Always use this server when running this project

< Back [Next >](#) [Finish](#) [Cancel](#)

YOUR WEBPAGE WILL GET OPEN

← ⌂ ⓘ localhost:8083/SampleMavenWebProject/

Import favorites | [takeUforward](#) [recursion](#) [NPTEL](#) [dsa](#) [Airbnb](#) [Tantrik](#) [TOOFAN](#) ⌂

Hello World!

PUSHING INTO GITHUB
CREATE NEW GIT REPOISTORY

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *	Repository name *
 Puppala-Sagar	SampleMavenWebProject
✓ SampleMavenWebProject is available.	

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

Description (optional)

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

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

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **Maven**

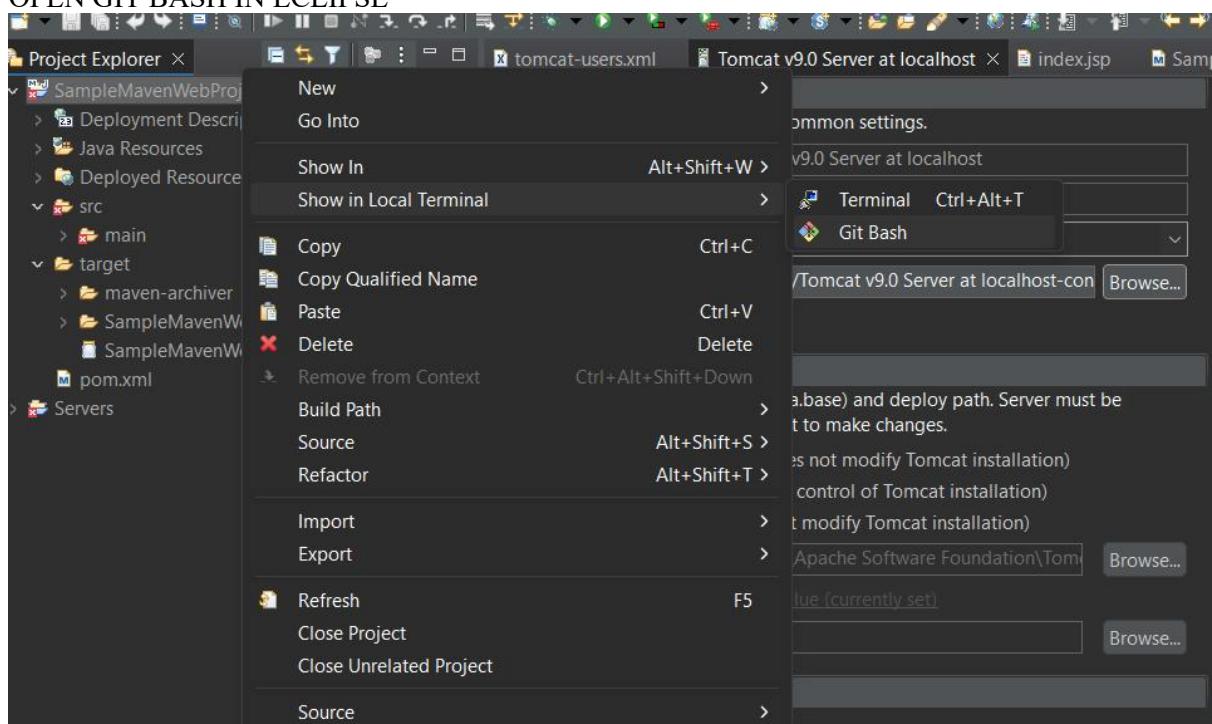
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

OPEN GIT BASH IN ECLIPSE



INITIALIZE THE GIT AND ADD ALL FILES

```
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ git init
Initialized empty Git repository in C:/Users/sagar/eclipse-workspace/SampleMavenWebProject/.git/
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ ls
pom.xml  src/  target/
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ git add .
```

DO COMMIT

```
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ git commit -m "sample Maven Web Project"
[master (root-commit) 7c5595a] sample Maven Web Project
17 files changed, 182 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.common.component
create mode 100644 .settings/org.eclipse.wst.common.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.validation.prefs
create mode 100644 pom.xml
create mode 100644 src/main/webapp/WEB-INF/web.xml
```

PUSH IN TO GITHUB

```
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ git push -u origin master
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 12 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (24/24), 4.62 KiB | 1.54 MiB/s, done.
Total 24 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Puppala-Sagar/SampleMavenWebProject/pull/new/master
remote:
To https://github.com/Puppala-Sagar/SampleMavenWebProject.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$
```

REFRESH YOUR GITHUB

Puppala-Sagar / SampleMavenWebProject

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

SampleMavenWebProject Public

master had recent pushes 1 minute ago

Compare & pull request

master 2 Branches 0 Tags

This branch is 1 commit ahead of, 1 commit behind main.

Contribute

Puppala-Sagar sample Maven Web Project 7c5595a · 3 minutes ago 1 Commit

.settings sample Maven Web Project 3 minutes ago

src/main/webapp sample Maven Web Project 3 minutes ago

target sample Maven Web Project 3 minutes ago

.classpath sample Maven Web Project 3 minutes ago

.project sample Maven Web Project 3 minutes ago

pom.xml sample Maven Web Project 3 minutes ago

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

JENKINS INSTALLATION

GOTO JENKINS WEBISTE

The screenshot shows the Jenkins website's 'Download and deploy' page. At the top, there are two main sections: 'Stable (LTS)' and 'Weekly releases'. Both sections have 'Changelog' and 'Past Releases' buttons. Below these, under 'Stable (LTS)', it says: 'Long-Term Support (LTS) release baselines are chosen every 12 weeks from the stream of regular releases. Every 4 weeks we release stable releases which include bug and security fix backports.' A 'Learn more...' link is provided. Under 'Weekly releases', it says: 'This release line delivers bug fixes and new features rapidly to users and plugin developers who need them. It is generally delivered on a weekly cadence.' Another 'Learn more...' link is provided. At the bottom of the page, there is a list of steps for downloading Jenkins:

1. Before downloading, please take a moment to review the [Hardware and Software requirements](#) section of the User Handbook.

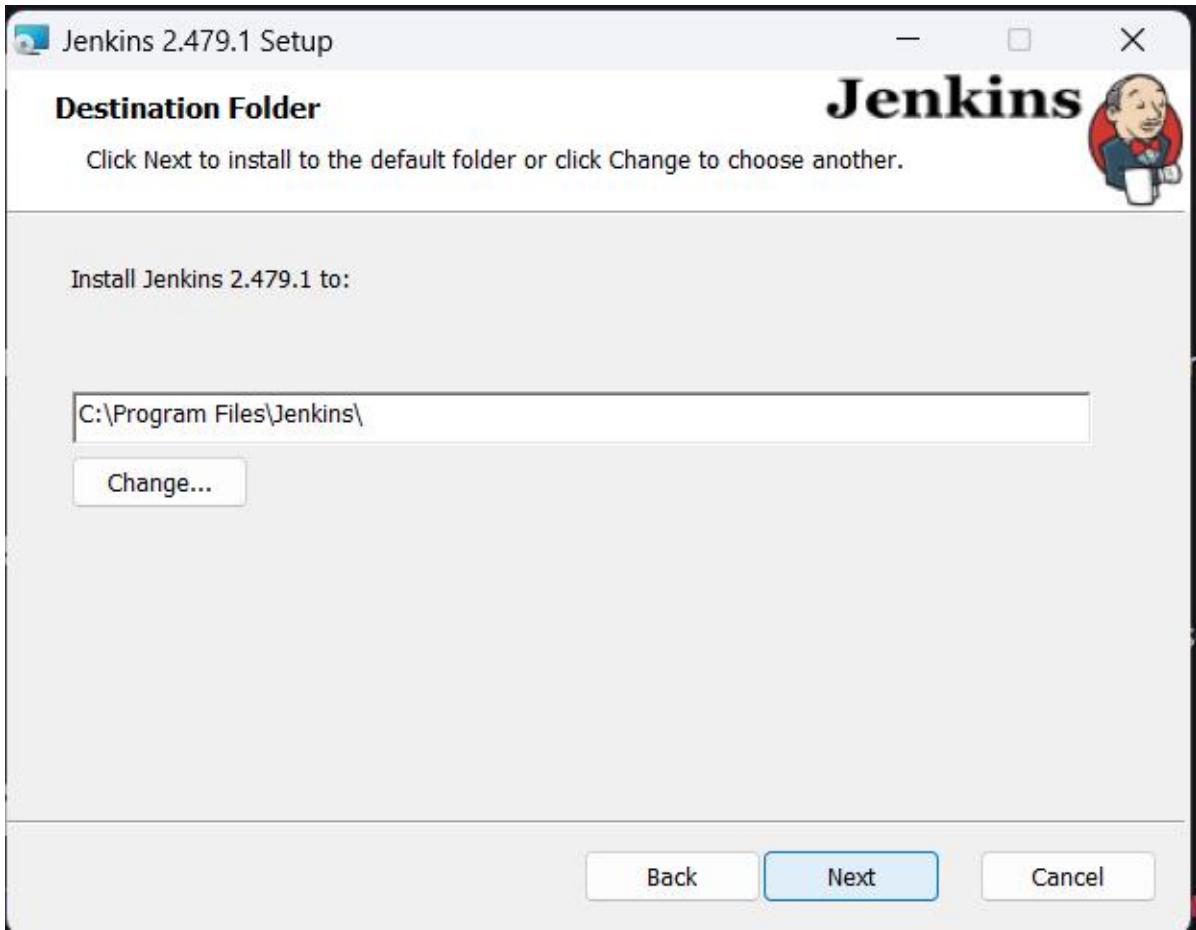
SELECT WINDOWS AND DONWLOAD EXE

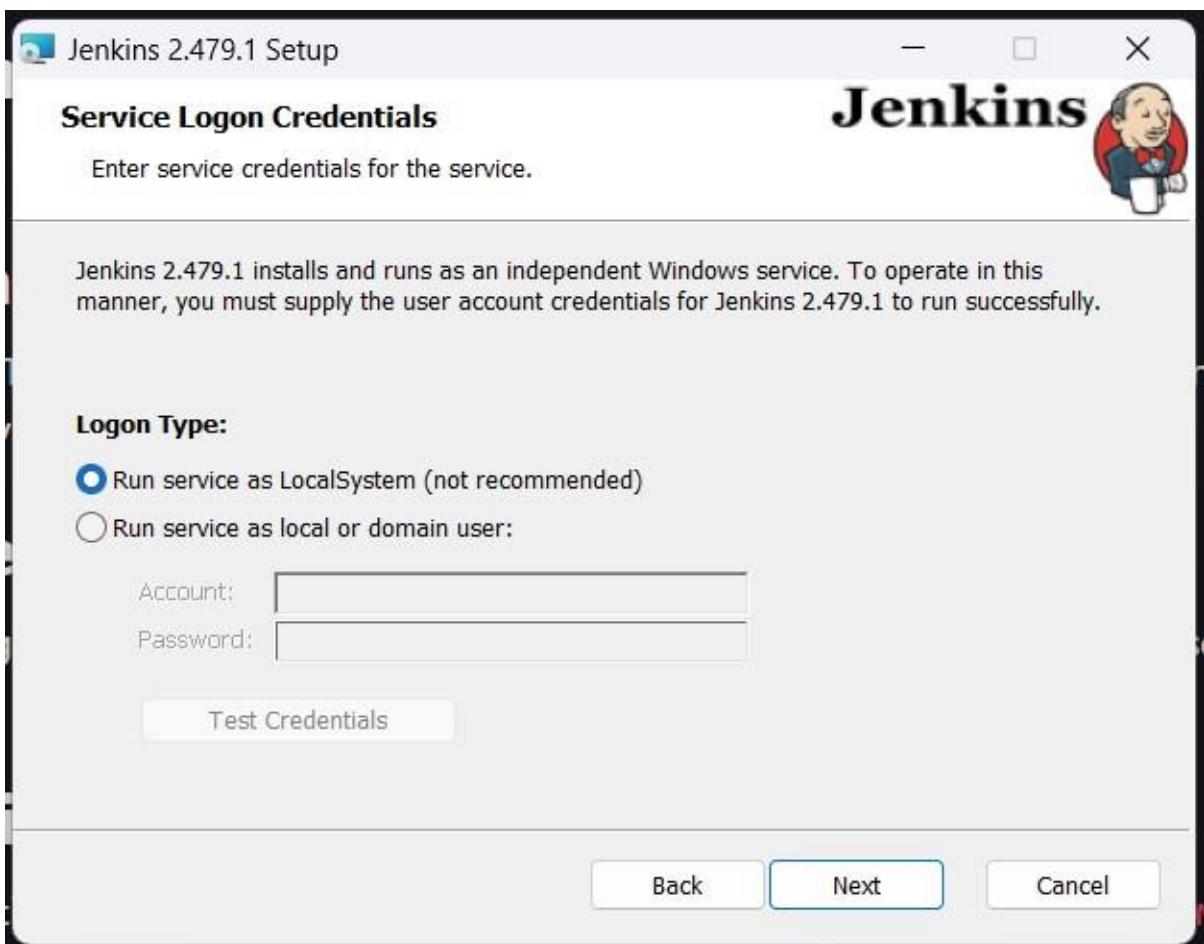
The screenshot shows the Jenkins website's download section for Jenkins 2.479.1 LTS and Jenkins 2.486. Each section has a 'Download Jenkins X.XXX for:' heading and a list of download options:

Download Jenkins X.XXX for:	Options
Download Jenkins 2.479.1 LTS for:	Generic Java package (.war), Docker, Kubernetes, Ubuntu/Debian, Red Hat/Fedora/Alma/Rocky/CentOS, Windows, openSUSE, FreeBSD
Download Jenkins 2.486 for:	Generic Java package (.war), Docker, Ubuntu/Debian, Red Hat/Fedora/Alma/Rocky/CentOS, Windows, openSUSE, Arch Linux, FreeBSD

INSTALL EXE







TEST PORT

Jenkins 2.479.1 Setup

Port Selection

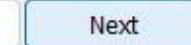
Choose a port for the service.

Please choose a port.

Port Number (1-65535):

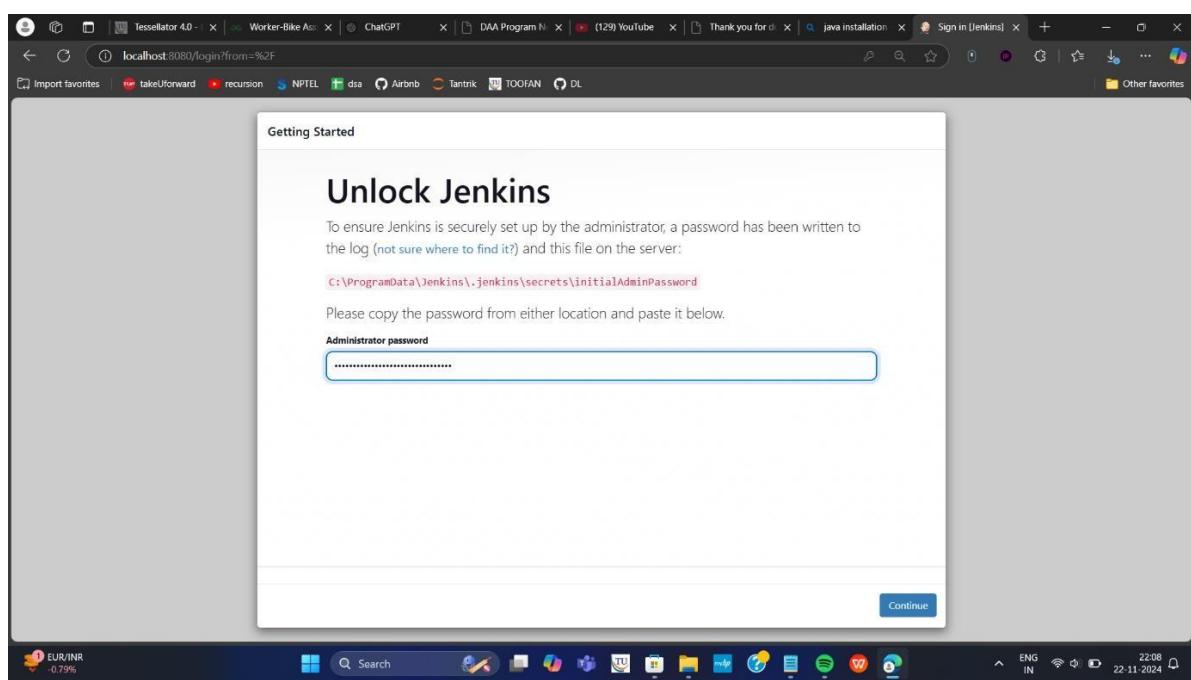
Test Port 

It is recommended that you accept the selected default port.

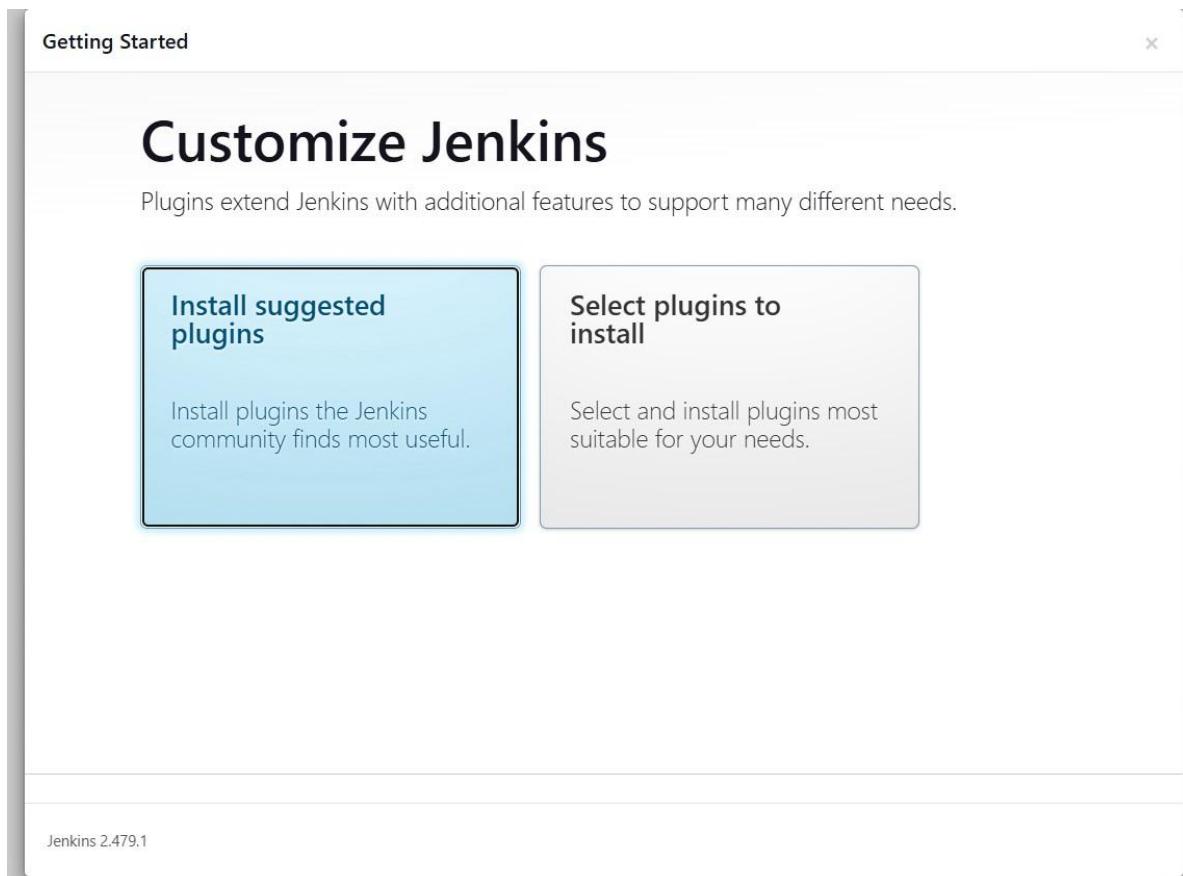
Back  Next Cancel



ADD ADMINISTRATOR PASSWORD



CLICK ON INSTALL SUGGESTED PLUGINS



WAIT FOR INSTALLATION

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Pipeline: Stage Tags Metadata ** Pipeline: Input Step ** Pipeline: Declarative Pipeline ** Java JSON Web Token (JWT) ** GitHub API ** Mina SSHD API :: Common ** Mina SSHD API :: Core ** Gson API ** Git client Git ** GitHub GitHub Branch Source Pipeline: GitHub Groovy Libraries ** Pipeline Graph Analysis ** Metrics Pipeline Graph View Git ** EDDSA API ** Trilead API SSH Build Agents Matrix Authorization Strategy PAM Authentication LDAP Email Extension Mailer ** - required dependency
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline Graph View	
✓ Git	✓ SSH Build Agents	✓ Matrix Authorization Strategy	✓ PAM Authentication	
✓ LDAP	✓ Email Extension	✓ Mailer	⌚ Dark Theme	

Jenkins 2.479.1

Instance Configuration

Jenkins URL:

http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.479.1

Not now

Save and Finish

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

LOGIN INTO YOUR JENKINS ACCOUNT

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Manage Jenkins' (which is highlighted), and 'My Views'. The main content area is titled 'Manage Jenkins' and contains several sections: 'System Configuration' (with 'System', 'Tools', 'Nodes', 'Clouds' sections), 'Security' (with 'Security', 'Credentials', 'Users' sections), and 'Plugins' (with 'Plugins' and 'Credential Providers' sections). A status bar at the bottom indicates 'localhost:8080/manage/configureTools'.

CLICK ON MANAGE JENKINS AND PLUGINS

The screenshot shows the Jenkins Manage Jenkins > Plugins page. The left sidebar has links for 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is highlighted). The main content area is titled 'Download progress' and shows a table of plugin names and their download status. Most items show a green checkmark indicating success, except for 'commons-lang3 v2 v Jenkins API' which shows a red error icon.

Plugin	Status
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
commons-lang3 v2 v Jenkins API	Error

MAVEN JAVA PROJECT PIPELINE

Sign in to Jenkins

Username

admin

Password

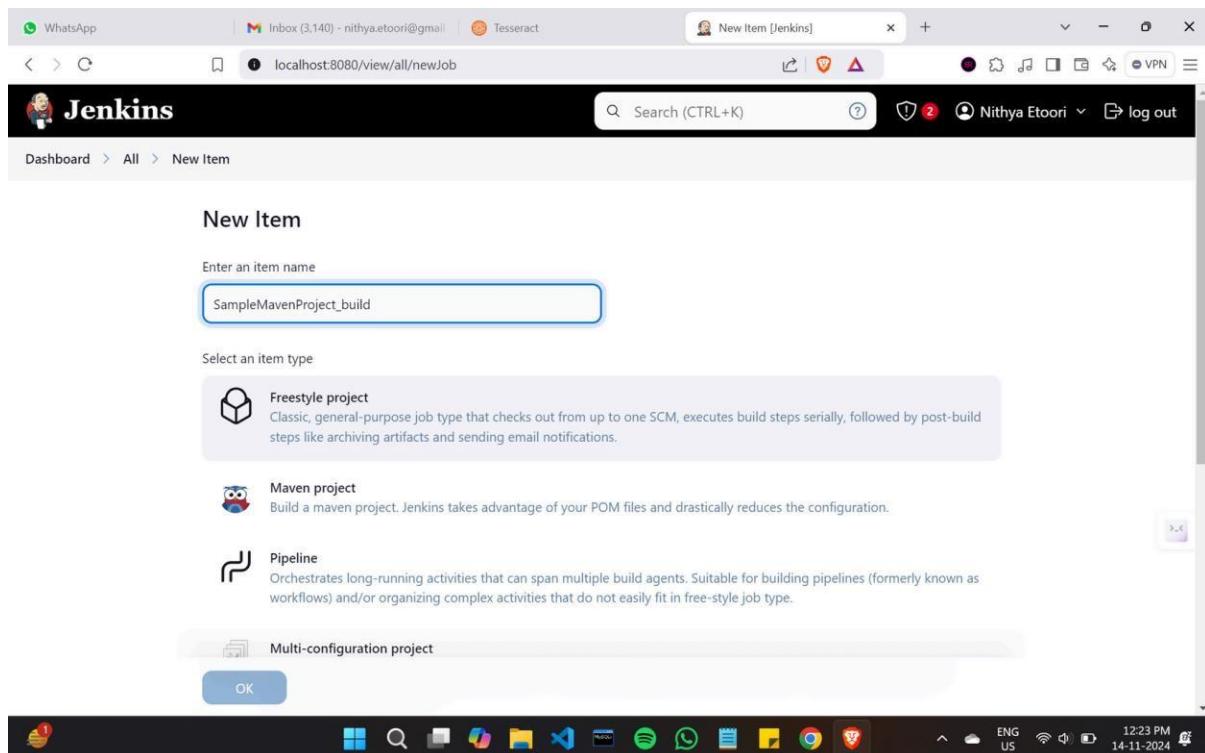
Keep me signed in

Sign in

Dashboard

The screenshot shows the Jenkins dashboard interface. At the top, there's a navigation bar with links for 'All Bookmarks', 'Search (CTRL+K)', a user icon, and 'Sairidhi Sriramula'. On the far right, there are 'log out' and a profile picture. Below the navigation, the title 'Dashboard' is followed by a breadcrumb trail: 'Dashboard >'. To the left, a sidebar contains links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. It also displays 'Build Queue' (empty) and 'Build Executor Status' (empty). The main content area features a 'Welcome to Jenkins!' message and a 'Start building your software project' section. This section includes a 'Create a job' button, a 'Set up a distributed build' section with 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds', and a 'REST API' link at the bottom right.

New freestyle project for build

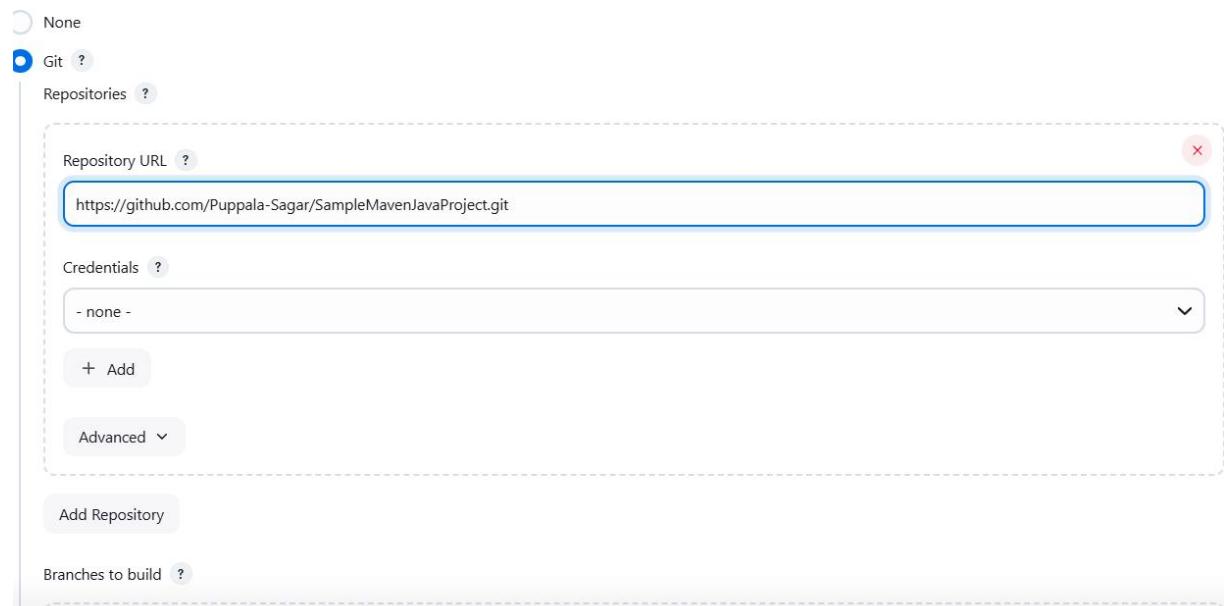


Git repo of existing project

The screenshot shows a GitHub repository page for 'SampleMavenJavaProject'. The repository was created by 'Puppala-Sagar'. It has one branch ('main') and one tag ('0 Tags'). The repository contains files: '.settings', 'src', '.gitignore', and 'pom.xml'. The 'pom.xml' file was committed yesterday. On the right side, there are sections for 'About', 'No releases', 'Releases', and 'Packaging'.

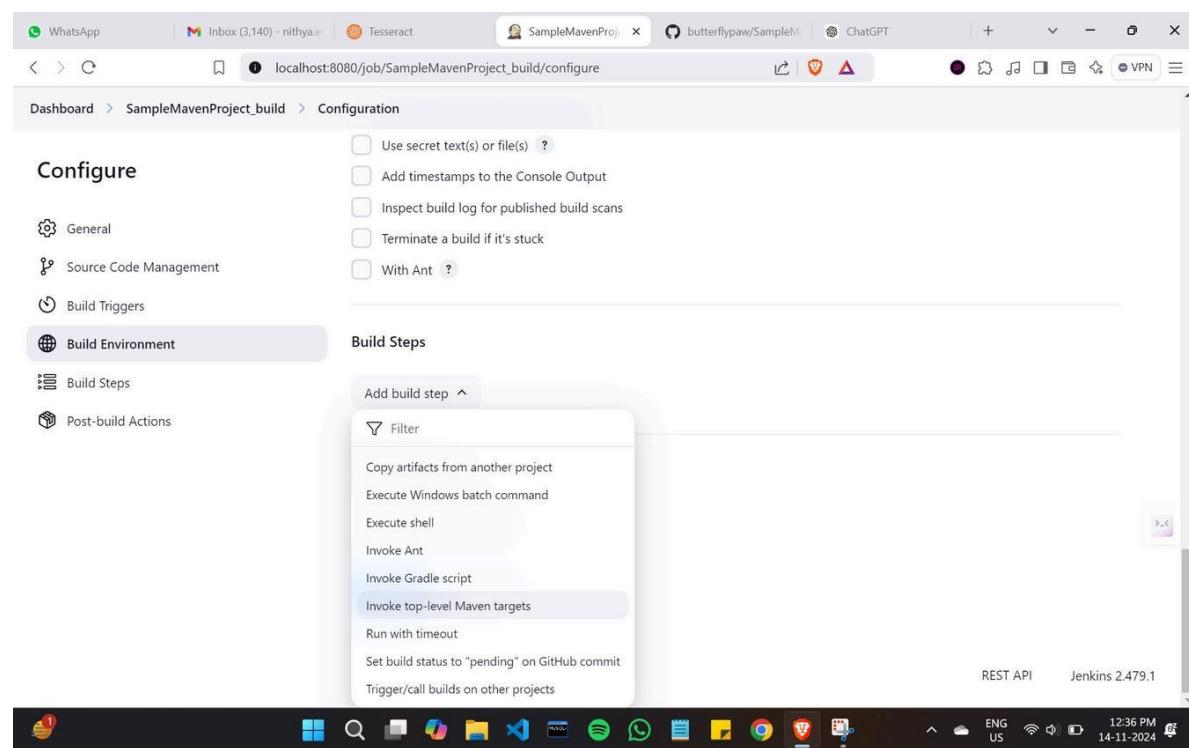
File	Description	Commit Date
.settings	Sample Maven Java Project	yesterday
src	Sample Maven Java Project	yesterday
.gitignore	Initial commit	yesterday
pom.xml	Sample Maven Java Project	yesterday

Add repo url and select the branch



The screenshot shows the Jenkins Git configuration page. At the top, there are two options: "None" and "Git". The "Git" option is selected, indicated by a blue icon. Below it, there is a "Repositories" section with a "Repository URL" input field containing the URL <https://github.com/Puppala-Sagar/SampleMavenJavaProject.git>. Under "Credentials", there is a dropdown menu set to "- none -" and a "+ Add" button. An "Advanced" dropdown menu is also present. At the bottom left, there is a "Add Repository" button.

Build Steps



The screenshot shows the Jenkins job configuration page for "SampleMavenProject_build". The "Configure" section is open, and the "Build Environment" tab is selected. In the "Build Steps" section, a dropdown menu is open, showing various options: "Copy artifacts from another project", "Execute Windows batch command", "Execute shell", "Invoke Ant", "Invoke Gradle script", "Invoke top-level Maven targets" (which is highlighted), "Run with timeout", "Set build status to "pending" on GitHub commit", and "Trigger/call builds on other projects". The Jenkins interface includes a header with multiple browser tabs and a taskbar at the bottom.

Add clean and install

The screenshot shows a web-based build configuration interface. On the left, there is a sidebar with icons for General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is selected), and Post-build Actions. The main area displays two 'Invoke top-level Maven targets' sections. The first section has 'Goals' set to 'clean'. The second section has 'Goals' set to 'install'. Both sections have a 'Maven Version' dropdown set to 'MAVEN_HOME'. At the bottom of the main area are 'Save' and 'Apply' buttons.

Post Build actions and invoking test

The screenshot shows a web-based build configuration interface. On the left, there is a sidebar with icons for General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is selected), and Post-build Actions. The main area displays two sections under 'Post-build Actions'. The first section is 'Archive the artifacts' with 'Files to archive' set to '**/*'. The second section is 'Build other projects' with 'Projects to build' set to 'SampleMavenProject_test'. A red error message says 'No such project 'SampleMavenProject_test'. Did you mean 'SampleMavenProject_build''? Below it are three radio button options: '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 main area are 'Save' and 'Apply' buttons.

New freestyle project for test

The screenshot shows the Jenkins 'New Item' configuration page. The 'Item name' field contains 'SampleMavenProject_test'. Under 'Select an item type', the 'Freestyle project' option is selected, described as a 'Classic, general-purpose job type'. Other options like 'Pipeline' and 'Multi-configuration project' are also listed. A blue 'OK' button is at the bottom.

Build Environment

The screenshot shows the Jenkins 'Configuration' page for the 'SampleMavenProject_test' job. The 'Build Triggers' section has 'Poll SCM' selected. The 'Build Environment' section has 'Delete workspace before build starts' checked. Buttons for 'Save' and 'Apply' are at the bottom.

Add build project

The screenshot shows the Jenkins 'Build Steps' configuration page for a build project named 'SampleMavenProject_test'. The 'Copy artifacts from another project' step is selected. The 'Project name' is set to 'SampleMavenProject_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. At the bottom are 'Save' and 'Apply' buttons.

Add test goal

The screenshot shows the Jenkins 'Post-build Actions' configuration page for a build project named 'SampleMavenProject_test'. The 'Invoke top-level Maven targets' action is selected. The 'Maven Version' is set to 'MAVEN_HOME' and the 'Goals' are set to 'test'. Below this, an 'Advanced' button is visible. The 'Add build step' button is also present. In the 'Post-build Actions' section, the 'Archive the artifacts' action is selected. The 'Files to archive' field contains '**/*'. At the bottom are 'Save' and 'Apply' buttons.

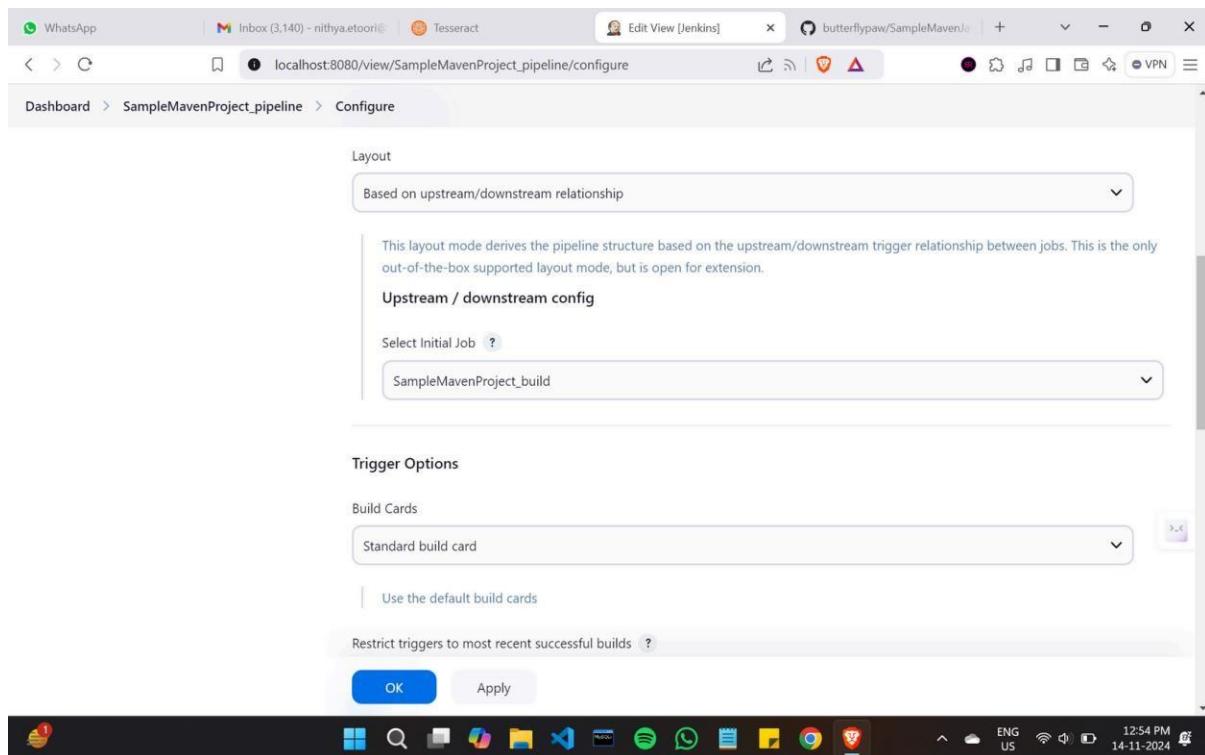
Dashboard after build and test projects

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Under 'Build History', there are two entries: 'SampleMavenProject_build' and 'SampleMavenProject_test', both marked as successful (green). The main area has tabs for 'All' and '+'. Below the tabs is a table with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table contains two rows for the builds mentioned. At the bottom, there's a 'Build Executor Status' section showing 0/2 executors available, with icons for S, M, and L.

Create pipeline

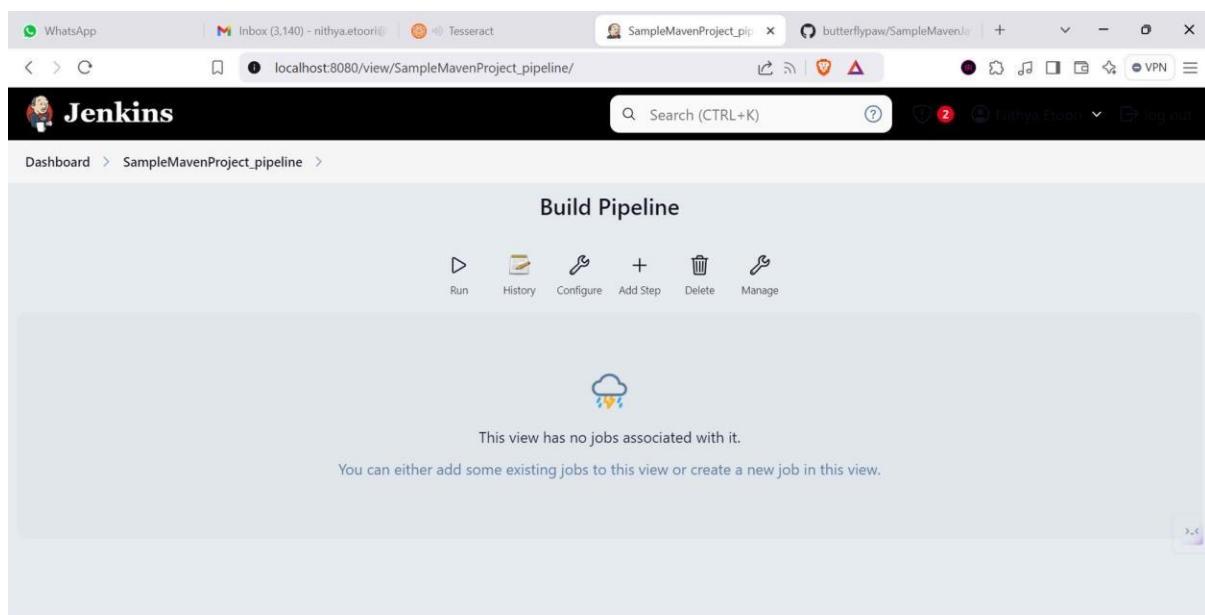
The screenshot shows the 'New view' creation page in Jenkins. The 'Name' field is filled with 'SampleMavenProject_pipeline'. The 'Type' section has three options: 'Build Pipeline View' (selected), 'List View', and 'My View'. A description for 'Build Pipeline View' states: 'Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.' Below the type selection is a 'Create' button. The bottom of the screen shows a taskbar with various icons and system status.

Select build project



The screenshot shows the Jenkins configuration interface for a pipeline named "SampleMavenProject_pipeline". The "Configure" tab is selected. Under the "Layout" section, it is set to "Based on upstream/downstream relationship". A note explains that this mode derives the pipeline structure based on trigger relationships. In the "Upstream / downstream config" section, the "Select Initial Job" dropdown is set to "SampleMavenProject_build". Below this, the "Trigger Options" section includes a "Build Cards" dropdown set to "Standard build card" and a checkbox for "Use the default build cards". There is also a checkbox for "Restrict triggers to most recent successful builds". At the bottom are "OK" and "Apply" buttons.

Build pipeline



The screenshot shows the Jenkins "Build Pipeline" view for the "SampleMavenProject_pipeline". The title bar says "Build Pipeline". Below the title are action buttons: Run, History, Configure, Add Step, Delete, and Manage. A message states "This view has no jobs associated with it." and "You can either add some existing jobs to this view or create a new job in this view." The Jenkins logo is visible in the top left corner of the main content area.

REST API Jenkins 2.479.1



Graphical representation of pipeline

The screenshot shows the Jenkins Build Pipeline interface. At the top, there is a navigation bar with links for WhatsApp, Inbox (3,140) - nithya.etoori@..., Tesseract, SampleMavenProject_pip, and butterflypaw/SampleMavenProject_pipeline/. Below the navigation bar is a search bar with the placeholder "Search (CTRL+K)". On the right side of the header, there are icons for a question mark, a shield with a red number "2", and a user profile for "Nithya Elston". A "log out" link is also present.

The main content area is titled "Build Pipeline". It displays a flow from a "Pipeline #1" step to a "SampleMavenProject_build" step, which then leads to a "SampleMavenProject_test" step. The "SampleMavenProject_build" step is highlighted in yellow, indicating it has been successfully completed. The "SampleMavenProject_test" step is shown in blue. The pipeline interface includes standard Jenkins navigation buttons: Run, History, Configure, Add Step, Delete, and Manage.

At the bottom right of the screen, the Jenkins version "Jenkins 2.479.1" is displayed, along with a "REST API" link. The taskbar at the bottom of the window shows various application icons, including WhatsApp, Mail, Spotify, and a browser icon.

Green color indicate successful pipeline

This screenshot shows the Jenkins Build Pipeline interface again, but this time the entire pipeline run is highlighted in green, indicating all steps have been successfully completed. The pipeline structure is identical to the previous screenshot, with "Pipeline #2" leading to "SampleMavenProject_build" and then "SampleMavenProject_test". The "SampleMavenProject_build" step is now fully green, and the "SampleMavenProject_test" step is also green. The Jenkins interface buttons (Run, History, etc.) are visible at the top, and the Jenkins version "Jenkins 2.479.1" is at the bottom right. The taskbar at the bottom of the screen is also visible.

Log

The screenshot shows the Jenkins interface with the following details:

- Top navigation bar: Dashboard > SampleMaven_pipeline2 > Jenkins
- Sub-navigation: Dashboard > SampleMavenProject_test > #1 > Console Output
- Main title: Console Output
- Left sidebar: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#1', Timings, See Fingerprints.
- Content area:
 - Started by upstream project "SampleMavenProject_build" build number 1 originally caused by: Started by user Sairidhi Sriramula Running as SYSTEM Building in workspace C:\ProgramData\Jenkins\workspace\SampleMavenProject_test [WS-CLEANUP] Deleting project workspace... [WS-CLEANUP] Deferred wipeout is used... [WS-CLEANUP] Done Copied 18 artifacts from "SampleMavenProject_build" build number 1 [SampleMavenProject_test] \$ cmd.exe /C "C:\ProgramData\Jenkins\tools\hudson.tasks.Maven_MavenInstallation\MAVEN_HOME\bin\mvn.cmd test && exit %ERRORLEVEL%" [INFO] Scanning for projects...
- Bottom footer: Console output for SampleMavenProject_test #1

Jenkins Poll SCM

Jenkins Dashboard

The screenshot shows the Jenkins dashboard with the following details:

- Top navigation bar: WhatsApp, Inbox (3,140) - nithya.eotori@..., Tesseract, Dashboard [Jenkins], localhost:8080, butterfypaw/SampleMavenJenkins, log out
- Sub-navigation: Dashboard > Jenkins
- Main title: SampleMavenProject_pipeline
- Left sidebar: + New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views.
- Content area:
 - Build Queue: No builds in the queue.
 - Build Executor Status: 0/2
 - Table view:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	SampleMavenProject_build	15 min #2	N/A	21 sec
✓	☀️	SampleMavenProject_test	15 min #2	N/A	9.7 sec
- Bottom footer: REST API, Jenkins 2.479.1



New build freestyle project

The screenshot shows the Jenkins 'New Item' configuration page. The 'Item name' field contains 'SampleMavenWebProject_build'. The 'Select an item type' section is expanded, showing the 'Freestyle project' option selected. Other options like 'Maven project', 'Pipeline', and 'Multi-configuration project' are also listed. An 'OK' button is visible at the bottom of the configuration panel.

Maven Web Project git repo

The screenshot shows the GitHub repository 'SampleMavenWebProject'. The 'master' branch is selected, showing 2 branches and 0 tags. The commit history for the 'master' branch is displayed, showing the following commits:

Author	Commit Message	Time	Commits
Puppala-Sagar	Update index.jsp	0162f08 · 11 hours ago	2 Commits
	.settings	sample Maven Web Project	yesterday
	src/main/webapp	Update index.jsp	11 hours ago
	target	sample Maven Web Project	yesterday
	.classpath	sample Maven Web Project	yesterday
	.project	sample Maven Web Project	yesterday
	pom.xml	sample Maven Web Project	yesterday

Add git repo and select branch

Repository URL ? X

Credentials ?

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? X

Trigger Poll SCM

WhatsApp | Gmail | Inbox (3,140) - nithya.e.toori@gmail | Tesseract | SampleMavenWebProject_build | +

localhost:8080/job/SampleMavenWebProject_build/configure

Dashboard > SampleMavenWebProject_build > Configuration

Build Triggers

Configure

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

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Schedule ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H * * * *" to poll once per hour
Would last run at Thursday, 14 November, 2024, 1:37:45 pm India Standard Time; would next run at Thursday, 14 November, 2024, 1:37:45 pm India Standard Time.

Ignore post-commit hooks ?

Build Environment

Save Apply

01:38 PM 14-11-2024 ENG US

Build Steps

The screenshot shows the Jenkins configuration interface for the 'SampleMavenProject_build' job. The 'Build Steps' section is currently selected. A dropdown menu is open under 'Add build step ^', showing various options: Copy artifacts from another project, Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets (which is highlighted), Run with timeout, Set build status to "pending" on GitHub commit, and Trigger/call builds on other projects.

Add clean and install

The screenshot shows the Jenkins configuration interface for the 'SampleMavenWebProject_build' job. The 'Build Steps' section is selected. Two 'Invoke top-level Maven targets' steps are defined. The first step has 'Goals' set to 'clean'. The second step has 'Goals' set to 'install'. Both steps have 'Maven Version' set to 'MAVEN_HOME'. At the bottom, there are 'Save' and 'Apply' buttons.

Post build actions and trigger test

The screenshot shows the Jenkins configuration interface for the 'SampleMavenWebProject_build' job. The 'Post-build Actions' section is selected. It contains two main configurations:

- Archive the artifacts**: Set to archive all files (glob: **/*).
- Build other projects**: Set to build 'SampleMavenWebProject_test'. A warning message indicates that no such project exists, suggesting a typo.

Below these are three trigger options for the build project:

- Trigger only if build is stable
- Trigger even if the build is unstable
- Trigger even if the build fails

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

New freestyle project for test

The screenshot shows the 'New Item' creation screen in Jenkins. The 'Freestyle project' option is selected. The form fields are as follows:

- Item name**: SampleMavenWebProject_test
- Item type**: Freestyle project (selected)
- Maven project**: Described as building a Maven project using POM files.
- Pipeline**: Described as orchestrating long-running activities across multiple build agents.

At the bottom are 'OK' and 'Cancel' buttons.

Build Environment

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Environment

- Delete workspace before build starts
- Advanced
- Use secret text(s) or file(s)
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant

Build Steps

Add build step

Save Apply

Add build project

Configure

General

Source Code Management

Build Triggers

Build Steps

Post-build Actions

Copy artifacts from another project

Project name ?
SampleMavenWebProject_build

Which build ?
Latest successful build

Stable build only

Artifacts to copy ?
**/*

Artifacts not to copy ?

Target directory ?

Save Apply

Add test goal

The screenshot shows the Jenkins configuration interface for a job named "SampleMavenWebProject_test". The "Configuration" screen is displayed, specifically the "Post-build Actions" section. A sub-section titled "Invoke top-level Maven targets" is expanded, showing the "Goals" field set to "test". Other options like "Flatten directories", "Optional", "Fingerprint Artifacts", and "Include Build Number" are visible but not selected. The "Advanced" dropdown is open. Below this, there is a "Post-build Actions" summary and "Save" and "Apply" buttons.

Trigger deploy

The screenshot shows the Jenkins configuration interface for the same job. The "Build Triggers" section is expanded, showing a "Build other projects" configuration. It lists "SampleMavenWebProject_deploy" as a target project. A warning message states: "No such project 'SampleMavenWebProject_deploy'. Did you mean 'SampleMavenWebProject_build'?" Below this, three trigger options are listed: "Trigger only if build is stable" (selected), "Trigger even if the build is unstable", and "Trigger even if the build fails". The "Advanced" dropdown is also visible. The "Save" and "Apply" buttons are at the bottom.

New freestyle project for deploy

The screenshot shows the Jenkins 'New Item' creation interface. At the top, there's a search bar with 'Search (CTRL+K)' and a user dropdown for 'Nithya Etoori'. Below the header, the breadcrumb navigation shows 'Dashboard > All > New Item'. The main title is 'New Item'. A text input field is labeled 'Enter an item name' and contains the value 'SampleMavenWebProject_deploy'. Under 'Select an item type', the 'Freestyle project' option is highlighted with a description: '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 are 'Maven project' and 'Pipeline'. At the bottom right of the dialog is a blue 'OK' button.

Build Environment

The screenshot shows the Jenkins 'Configure' screen for the 'SampleMavenWebProject_deploy' job. The URL in the browser is 'localhost:8080/job/SampleMavenWebProject_deploy/configure'. The left sidebar has sections for 'General', 'Source Code Management', 'Build Triggers' (which is selected), 'Build Environment', 'Build Steps', and 'Post-build Actions'. In the 'Build Triggers' section, 'Delete workspace before build starts' is checked. In the 'Build Environment' section, several options are listed: 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. At the bottom of the configuration page are 'Save' and 'Apply' buttons. The system tray at the bottom right shows the date and time as '14-11-2024 01:44 PM'.

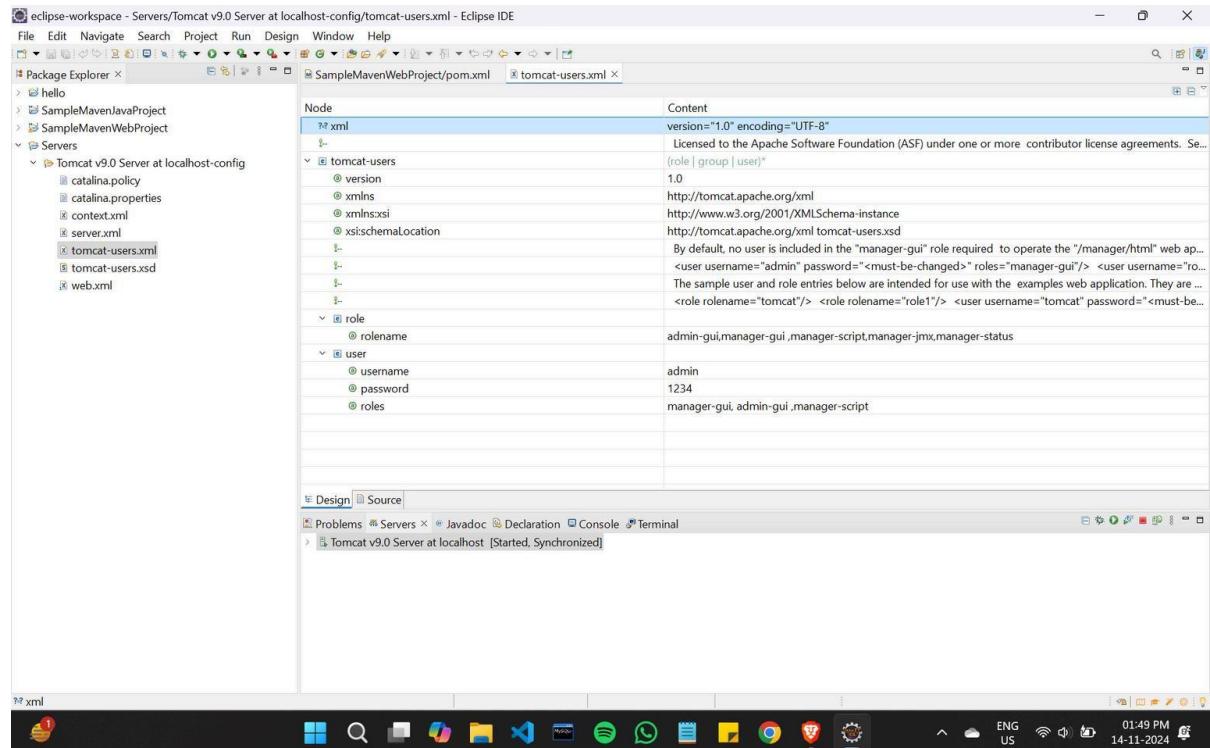
Add test project

The screenshot shows the Jenkins configuration interface for a job named "SampleMavenWebProject_deploy". The left sidebar shows navigation options: General, Source Code Management, Build Triggers, Build Environment, Build Steps (selected), and Post-build Actions. The main panel is titled "Copy artifacts from another project". It includes fields for "Project name" (set to "SampleMavenWebProject_test"), "Which build" (set to "Latest successful build"), and "Stable build only" (checkbox checked). Under "Artifacts to copy", the value is set to "*/". There are also fields for "Artifacts not to copy" and "Target directory". At the bottom are "Save" and "Apply" buttons.

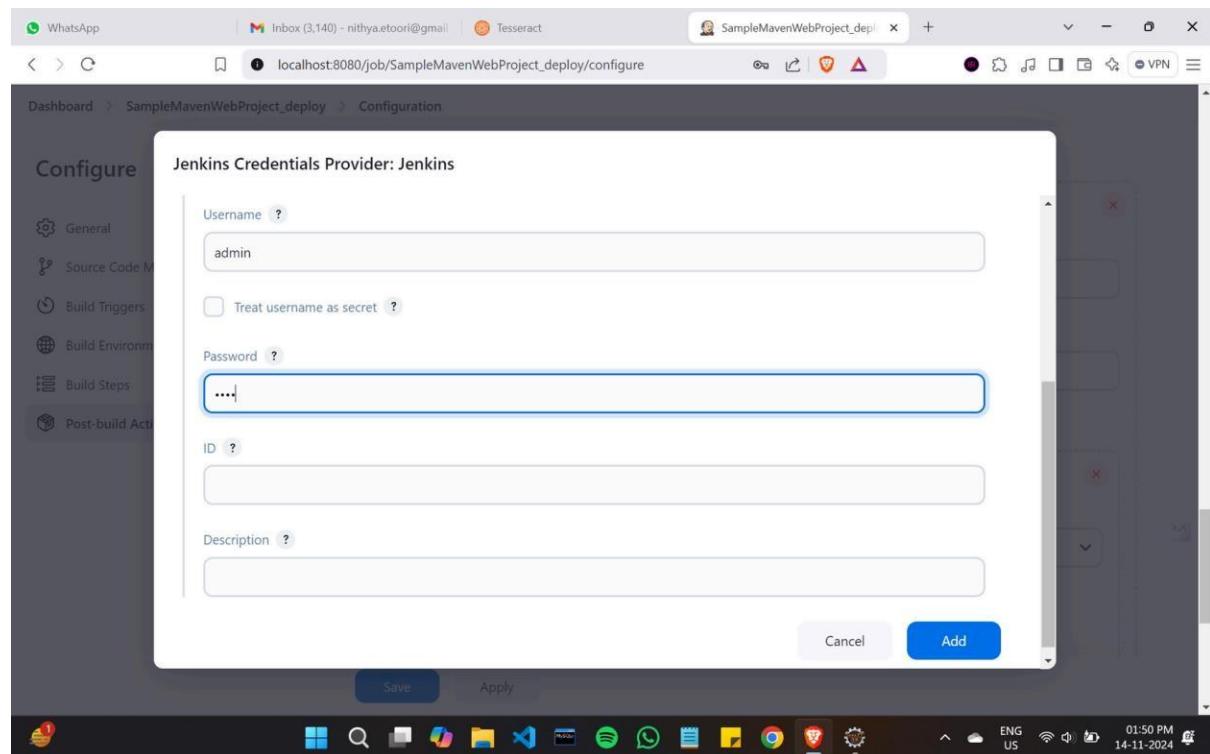
Post Build Actions

The screenshot shows the Jenkins configuration interface for a job named "SampleMavenWebProject_deploy". The left sidebar shows navigation options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions (selected). The main panel is titled "Post-build Actions". It includes a section for "Deploy war/ear to a container". Under "WAR/EAR files", the value is set to "*/*.war". Under "Context path", the value is set to "samplewebprojectmaven". A "Containers" section is expanded, showing "Tomcat 9.x Remote" and "Credentials" dropdowns. The dropdown shows "none" and "Jenkins Credentials Provider" (selected). Below that is a "Jenkins" option. At the bottom are "Save" and "Apply" buttons.

Check tomcat credential s from Eclipse and start server



Add the credentials in jenkins



Tomcat is running on 8082

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

Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Developer Quick Start

Tomcat Setup
First Web Application

Realms & AAA
JDBC DataSources

Examples

Servlet Specifications
Tomcat Versions

Documentation

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

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

- [tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)
User support and discussion
- [taglibs-user](#)
User support and discussion for Apache Taglibs
- [tomcat-dev](#)
Development mailing list, including commit messages

Add tomcat url

Configure

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

Context path ?
samplewebprojectmaven

Containers

Tomcat 9.x Remote

Credentials
admin/*****

+ Add

Tomcat URL ?
http://localhost:8082/

Advanced

Save Apply

Create a pipeline

The screenshot shows the Jenkins dashboard at localhost:8080. A pipeline named "SampleMavenProject_pipeline" is displayed in the center. The pipeline consists of five jobs: "SampleMavenProject_build", "SampleMavenProject_test", "SampleMavenWebProject_build", "SampleMavenWebProject_deploy", and "SampleMavenWebProject_test". Each job has a green checkmark icon and a yellow sun icon. The "SampleMavenProject_build" job is the most recent, with a "Last Success" timestamp of 57 min #2. The "SampleMavenWebProject_build" job is the second most recent, with a "Last Success" timestamp of 12 min #1.

Pipeline for project

The screenshot shows the "New view" configuration page in Jenkins. The "Name" field is set to "SampleWebProject_pipeline". The "Type" section is set to "Build Pipeline View", which is described as showing the complete pipeline of jobs that a version propagates through. There are also options for "List View" and "My View". A "Create" button is visible at the bottom.

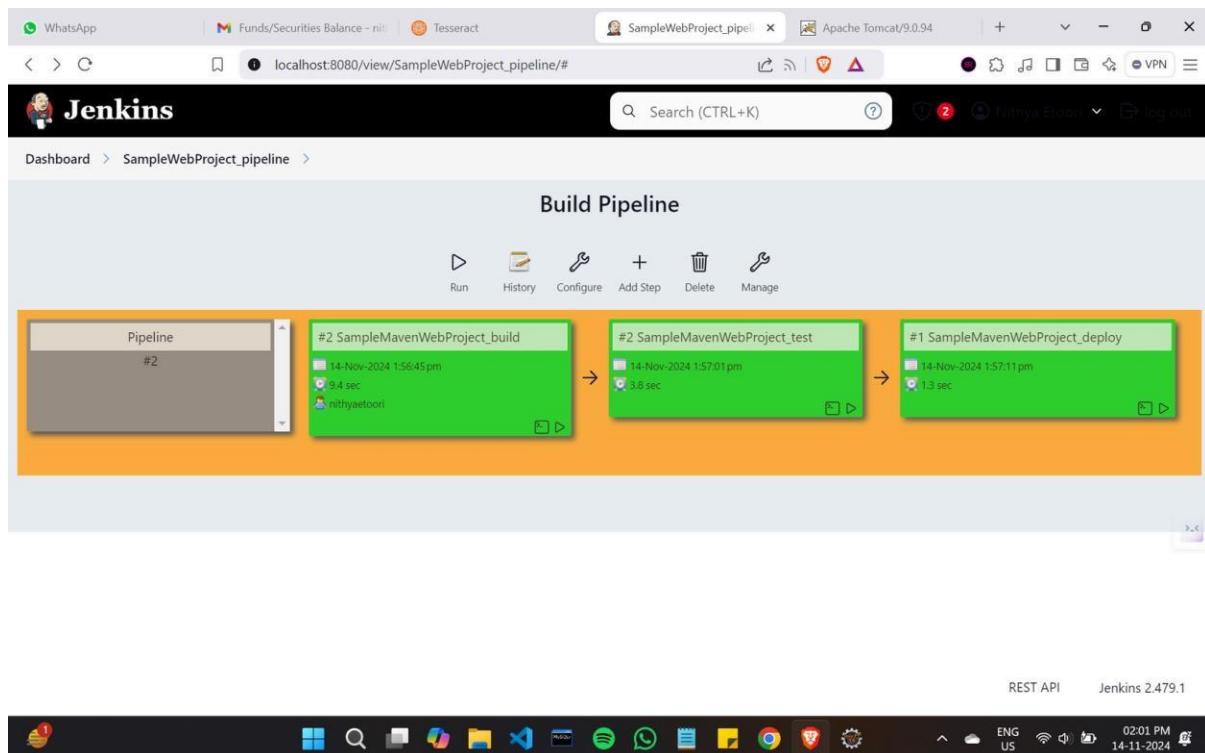
Select the build project

The screenshot shows the Jenkins Pipeline Flow configuration page for a project named "SampleWebProject_pipeline". The "Layout" section is set to "Based on upstream/downstream relationship". The "Upstream / downstream config" section shows a dropdown menu for "Select Initial Job" with the option "SampleMavenWebProject_build" selected. Other options listed include SampleMavenProject_build, SampleMavenProject_test, SampleMavenWebProject_build, SampleMavenWebProject_deploy, and SampleMavenWebProject_test. Below the dropdown are "OK" and "Apply" buttons. The browser's address bar shows the URL `localhost:8080/view/SampleWebProject_pipeline/configure`. The system tray at the bottom right indicates the date as 14-11-2024 and time as 01:55 PM.

Build pipeline

The screenshot shows the Jenkins Build Pipeline visualization for the "SampleWebProject_pipeline". The pipeline consists of four stages: "Pipeline #1" (grey), "#1 SampleMavenWebProject_build" (green, completed at 14-Nov-2024 1:40:59pm, duration 40 sec), "#1 SampleMavenWebProject_test" (green, completed at 14-Nov-2024 1:41:49pm, duration 0.29 sec), and "SampleMavenWebProject_deploy" (cyan). The stages are connected by arrows indicating the flow from build to test to deployment. The Jenkins logo is visible in the top left of the browser window. The system tray at the bottom right indicates the date as 14-11-2024 and time as 01:55 PM.

Successful completion of pipeline



Tomcat Manager App

The screenshot shows the Tomcat Web Application Manager interface. At the top, there is a message box with "Message:" and "OK". Below it is a navigation bar with tabs: Manager, List Applications, HTML Manager Help, Manager Help, and Server Status. The main content area is titled "Tomcat Web Application Manager" and contains two sections:

Applications

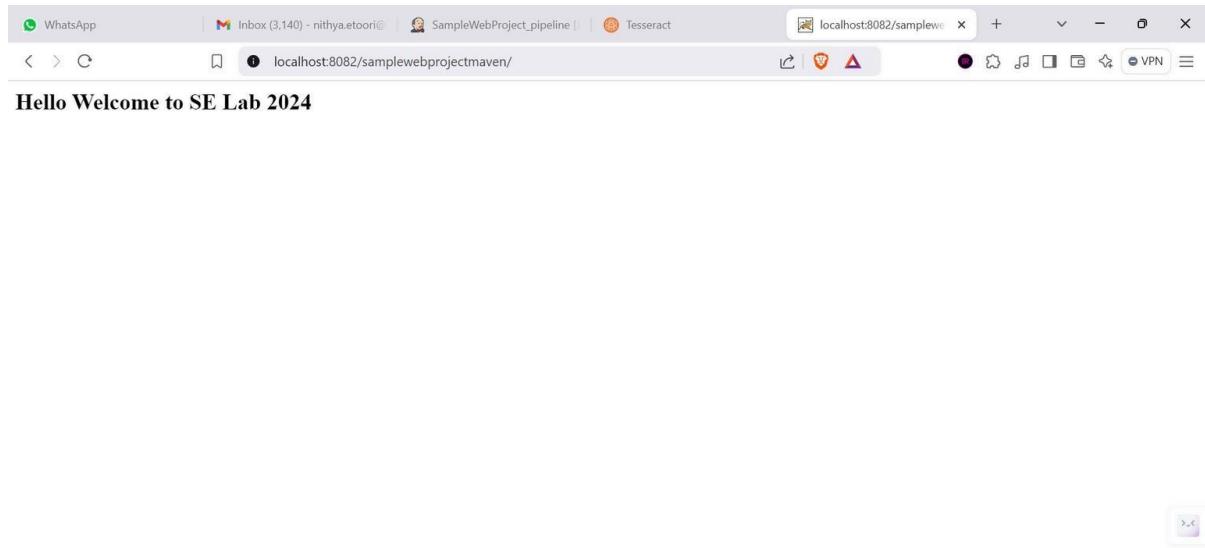
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>
/SampleMavenWebProject2	None specified	Archetype Created Web Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>
/docs	None specified	Tomcat Documentation	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>
/examples	None specified	Archetype Created Web Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>
/manager	None specified	Tomcat Manager Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions with idle ≥ 30 minutes</button>

Deploy

Deploy directory or WAR file located on server

Context Path:

Project working on Sever



Git repo

A screenshot of a GitHub repository page for 'SampleMavenWebProject'. The repository is public and has 2 branches and 0 tags. The master branch is currently selected. The page shows a commit from 'Puppala-Sagar' titled 'Update index.jsp' made 11 hours ago. Below the commit, a list of files is shown, all updated yesterday: '.settings', 'src/main/webapp', 'target', '.classpath', '.project', and 'pom.xml'. Each file entry includes a sample commit message and the date it was last modified.

Select index.jsp

SampleMavenWebProject / src / main / webapp /

Puppala-Sagar Update index.jsp

This branch is 2 commits ahead of, 1 commit behind main.

Name	Last commit message
...	
WEB-INF	sample Maven Web
index.jsp	Update index.jsp

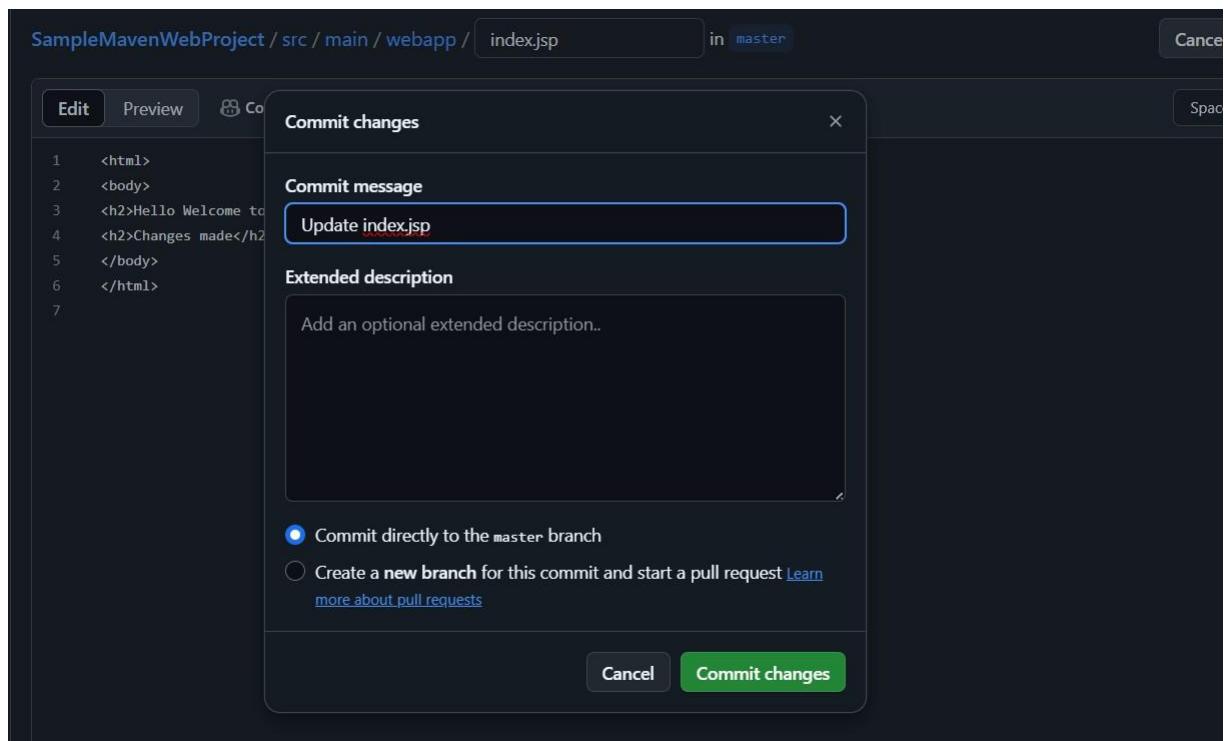
Make changes to the jsp file

SampleMavenWebProject / src / main / webapp / index.jsp in master

Edit Preview Code 55% faster with GitHub Copilot

```
1 <html>
2 <body>
3 <h2>Hello Welcome to SE Lab 2024</h2>
4 <h2>Changes made</h2>
5 </body>
6 </html>
7
```

Commit changes



Check for changes in build project

A screenshot of the Jenkins Dashboard. On the left, there is a sidebar with links like "Dashboard", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". Under "Build History", it says "No builds in the queue.". In the center, there are tabs for "All", "SampleMavenProject_pipeline", and "SampleWebProject_pipeline". The "All" tab is selected, showing a table of build status:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	SampleMavenProject_build	1 hr 15 min #2	N/A	21 sec
✓	☀️	SampleMavenProject_test	1 hr 14 min #2	N/A	9.7 sec
✓	☀️	SampleMavenWebProject_build	14 min #2	N/A	9.4 sec
✓	☀️	SampleMavenWebProject_deploy	14 min #1	N/A	1.3 sec
✓	☀️	SampleMavenWebProject_test	14 min #2	N/A	3.8 sec

At the bottom, there is a toolbar with icons for "Icon: S M L" and a status bar showing "localhost:8080/job/SampleMavenWebProject_build/" and "Jenkins 2.479.1".

Check builds for changes

The screenshot shows a Jenkins job page for 'SampleMavenWebProject_build'. On the left, there's a sidebar with options like 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'Git Polling Log', and 'Rename'. Below that is a 'Builds' section listing three recent builds: #3 (2:11PM), #2 (1:56PM), and #1 (1:40PM). To the right, under 'Last Successful Artifacts', a list of files is shown with their sizes and 'view' links:

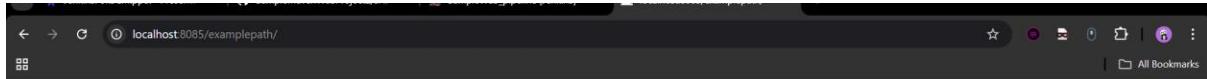
Artifact	Size	Action
.classpath	1.65 KiB	view
project	1.07 KiB	view
jsdtscope	639 B	view
org.eclipse.core.resources_prefs	57 B	view
org.eclipse.jdt.core_prefs	552 B	view
org.eclipse.m2e.core_prefs	90 B	view
org.eclipse.wst.common.component	704 B	view
org.eclipse.wst.common.project.facet.core.xml	252 B	view
org.eclipse.wst.jsdt.ui.superType.container	49 B	view
org.eclipse.wst.validation_prefs	6 B	view
pom.xml	2.15 KiB	view
webapp/index.jsp	96 B	view
webapp/WEB-INF/web.xml	222 B	view
pom.properties	112 B	view
SampleMavenWebProject/index.jsp	96 B	view
servlet-api-2.5.jar	102.65 KiB	view
SampleMavenWebProject/WEB-INF/web.xml	222 B	view
SampleMavenWebProject.war	98.10 KiB	view

Below the artifacts, there's a 'Downstream Projects' section with a link to 'SampleMavenWebProject_test'. At the bottom, there's a 'Permalinks' section and a Windows taskbar at the very bottom.

Build pipeline after changes

The screenshot shows the Jenkins pipeline interface for 'SampleWebProject_pipeline'. It displays a sequence of three pipeline stages: #3 SampleMavenWebProject_build, #3 SampleMavenWebProject_test, and #2 SampleMavenWebProject_deploy. Each stage is represented by a green box with its name, build number, timestamp, and duration. The stages are connected by arrows indicating the flow from build to test to deployment. The Jenkins logo is visible at the top left, and the bottom of the screen shows a Windows taskbar with various icons and system status.

Result on server after changes

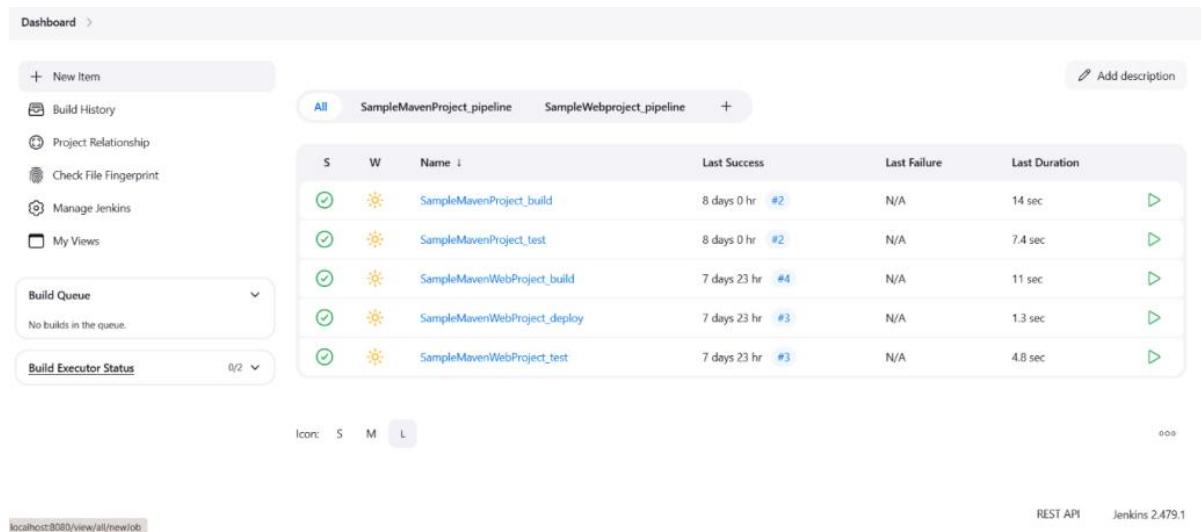


Hello World!

Changes made

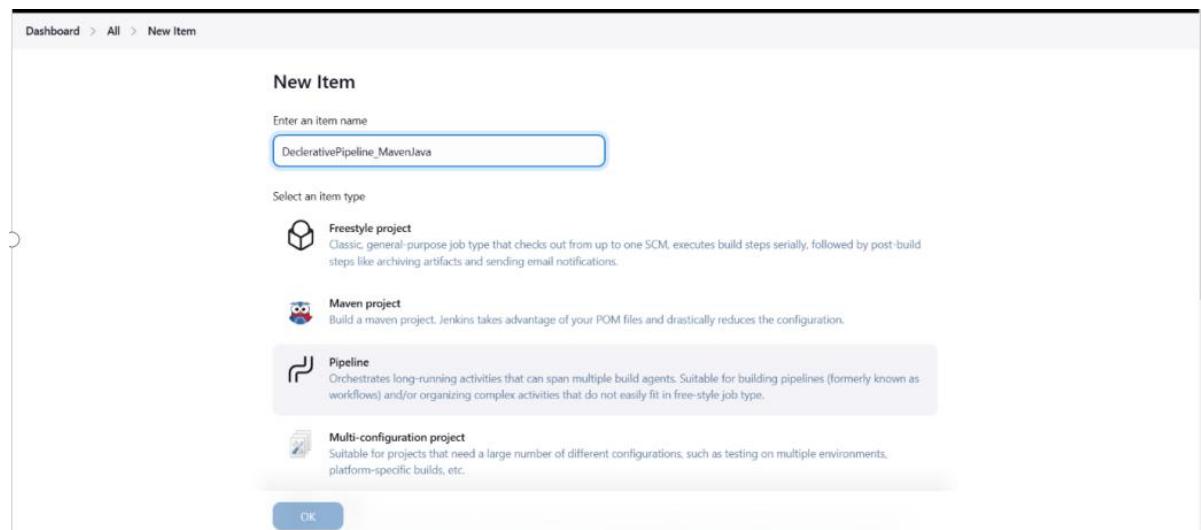
CI/CD Scripted Pipelines for Maven Java Project

Step 1: Open “<http://localhost:8080>” and sign in. Click on New Item to create a pipeline.

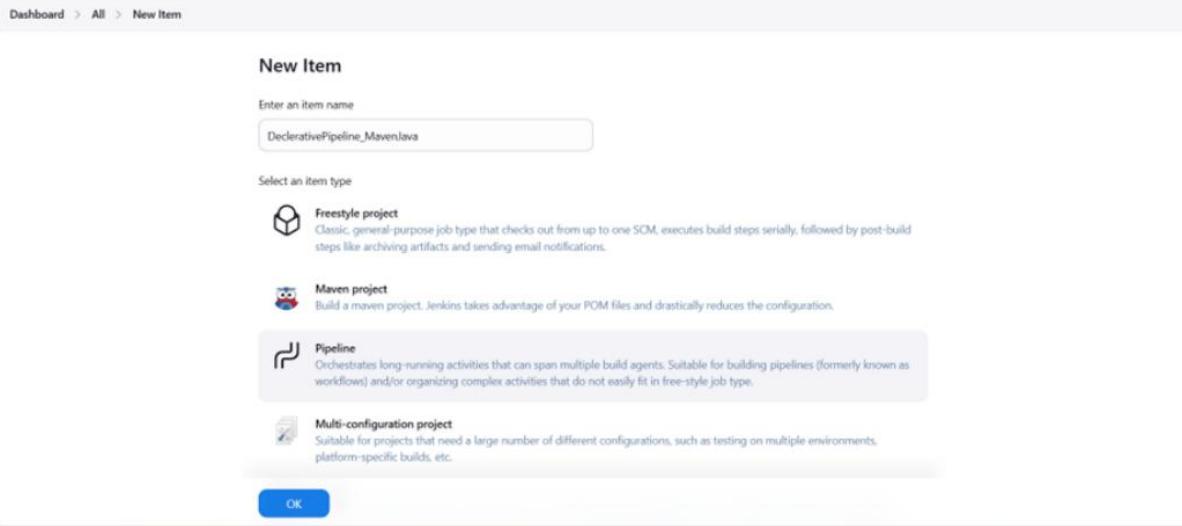


The screenshot shows the Jenkins dashboard with the 'New Item' button highlighted. The dashboard includes a sidebar with links like Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. A 'Build Queue' section indicates 'No builds in the queue'. A 'Build Executor Status' section shows 0/2 executors available. The main area displays a table of existing pipelines: SampleMavenProject_pipeline, SampleWebproject_pipeline, SampleMavenProject_build, SampleMavenProject_test, SampleMavenWebProject_build, SampleMavenWebProject_deploy, SampleMavenWebProject_test. Each row shows the status (green checkmark), last success time, last failure, and duration. The URL in the address bar is localhost:8080/view/all/newjob.

Step 2: Enter a name like DeclarativePipeline_MavenJava and select Pipeline and click ok.



The screenshot shows the 'New Item' dialog. In the 'Enter an item name' field, 'DeclarativePipeline_MavenJava' is typed. Below it, under 'Select an item type', the 'Pipeline' option is selected, shown with a highlighted background. Other options include 'Freestyle project', 'Maven project', and 'Multi-configuration project'. At the bottom right is an 'OK' button.



Step 3: Paste the script in the Pipeline script and click Apply and then Save.

```

1 - pipeline {
2   agent any
3   tools {
4     maven 'MAVEN_HOME'
5     // git 'GIT_HOME'
6   }
7   stages {
8     stage('git repo & clean') {
9       steps {
10        bat "rmdir /s /q SampleMavenJavaProject"
11        bat "git clone https://github.com/Puppia-Sagar/SampleMavenJavaProject.git"
12        bat "mvn clean -f SampleMavenJavaProject"
13      }
14    }
15    stage("Install") {
16      steps {
17        bat "mvn install -f SampleMavenJavaProject" //project name
18      }
19    }
}

```

Use Groovy Sandbox ?

Save Apply

Step 4: Click on Build Now.

The screenshot shows the Jenkins job configuration page for 'DeclarativePipeline_MavenJava'. The top navigation bar includes links for 'Dashboard', 'DeclarativePipeline_MavenJava', 'Status' (which is currently selected), 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Stages', 'Rename', and 'Pipeline Syntax'. A 'Permalinks' section is also present. On the left, there's a 'Builds' summary card showing one build from today at 11:25 pm, which is marked as successful (green). The URL in the address bar is 'localhost:8080/jobs/DeclarativePipeline_MavenJava/build?delay=0sec'. The bottom right corner indicates 'REST API' and 'Jenkins 2.479.1'.



DeclarativePipeline_MavenJava

Permalinks

- [Last build \(#15\), 1 min 2 sec ago](#)
- [Last stable build \(#15\), 1 min 2 sec ago](#)
- [Last successful build \(#15\), 1 min 2 sec ago](#)
- [Last failed build \(#14\), 3 min 1 sec ago](#)
- [Last unsuccessful build \(#14\), 3 min 1 sec ago](#)
- [Last completed build \(#15\), 1 min 2 sec ago](#)

Step 5: If it turns green then the build is successful.

```

[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.036 s - in com.app.demo.SampleMavenJavaProject.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.0.2:jar (default-jar) @ SampleMavenJavaProject ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  3.648 s
[INFO] Finished at: 2024-12-06T00:49:54+05:30
[INFO] -----
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Step 6: Click on configuration and check the Build periodically in Build Triggers. Enter “H/3* * * * *” in the Schedule. This indicates the build is automatically triggered every three minute.

Dashboard > DeclarativePipeline_MavenJava > Configuration

Build after other projects are built ?

Build periodically ?

General

Provides a cron-like feature to periodically execute this project. This feature is primarily for using Jenkins as a cron replacement, and it is **not ideal for continuously building software projects**. When people first start continuous integration, they are often so used to the idea of regularly scheduled builds like nightly/weekly that they use this feature. However, the point of continuous integration is to start a build as soon as a change is made, to provide a quick feedback to the change. To do that you need to [hook up SCM change notification to Jenkins](#).

So, before using this feature, stop and ask yourself if this is really what you want.

Schedule ?
H/15 * * * *

Would last have run at Thursday, December 5, 2024, 11:28:42 PM India Standard Time; would next run at Thursday, December 5, 2024, 11:43:42 PM India Standard Time.

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:

MINUTE	HOUR	DOM	MONTH	DOW
--------	------	-----	-------	-----

MINUTE Minutes within the hour (0-59)

Save **Apply**

The screenshot shows the Jenkins job configuration page for 'DeclarativePipeline_MavenJava'. In the 'General' section, the 'Build periodically' checkbox is checked. The cron schedule is set to 'H/15 * * * *'. A note below explains that this follows cron syntax and will run every 15 minutes. A green 'Saved' button is visible at the bottom left.

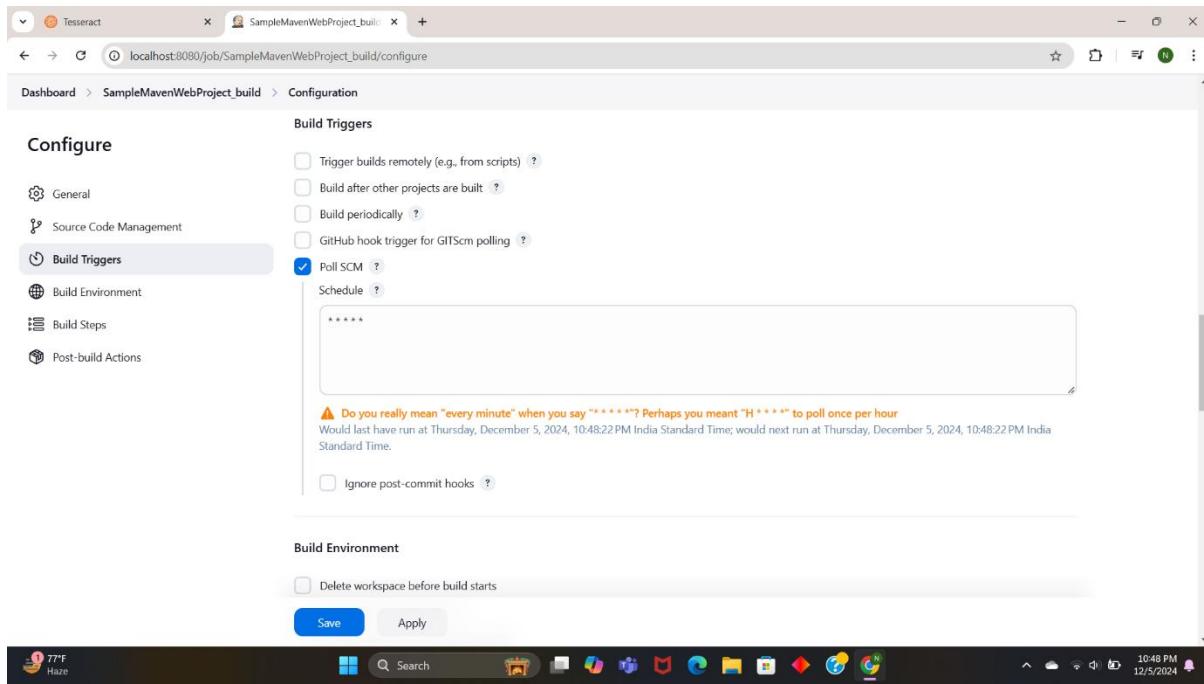
Step 7: Uncheck the Build Periodically in Build Trigger to stop the automation of build every minute.

The screenshot shows the Jenkins job configuration page for 'DeclarativePipeline_MavenJava'. In the 'Build triggers' section, the 'Poll SCM' checkbox is checked. Other options like 'Build after other projects are built' and 'Trigger builds remotely' are also present. The 'Pipeline' section shows a 'Pipeline script' field with the value 'Pipeline script'. A blue 'Save' button is visible at the bottom left.

Step-09 : enable poll SCM

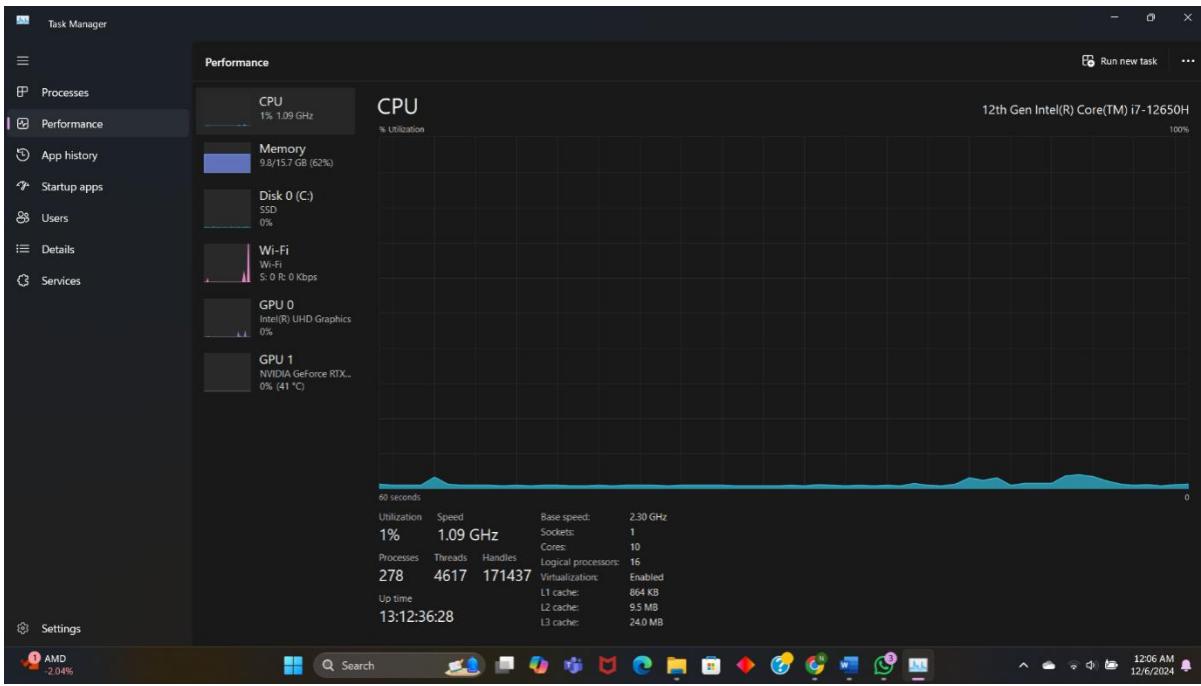
The screenshot shows the Jenkins configuration page for the job 'DeclarativePipeline_MavenJava'. The 'Configure' section is open, specifically the 'General' tab. Under the 'Poll SCM' section, the checkbox labeled 'Poll SCM' is checked. The schedule is set to 'H/3 * * * *'. A note indicates that this is an expensive operation for CVS and suggests using a 'push' trigger. Below the schedule, there are options for 'Ignore post-commit hooks', 'Quiet period', and 'Trigger builds remotely'. At the bottom of the configuration page, there are 'Save' and 'Apply' buttons, and a green 'Saved' message is displayed.

Freestyle pipeline for java web project



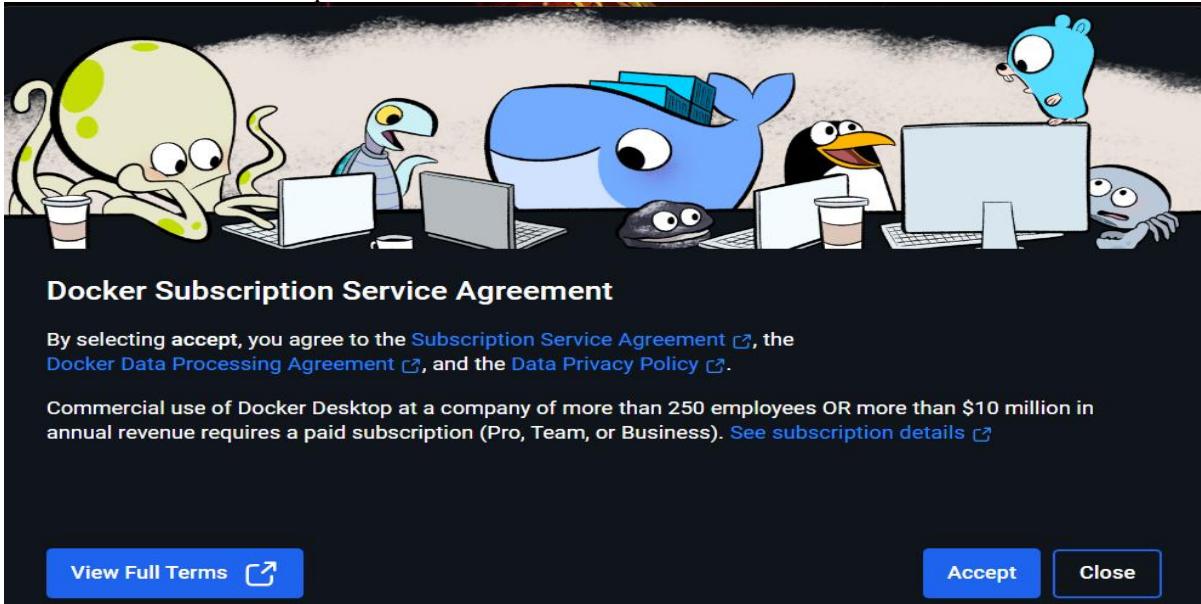
Installation of Docker

Step 1 : prepare your computer
open task manager
go to performance tab
check if virtualization is enabled .

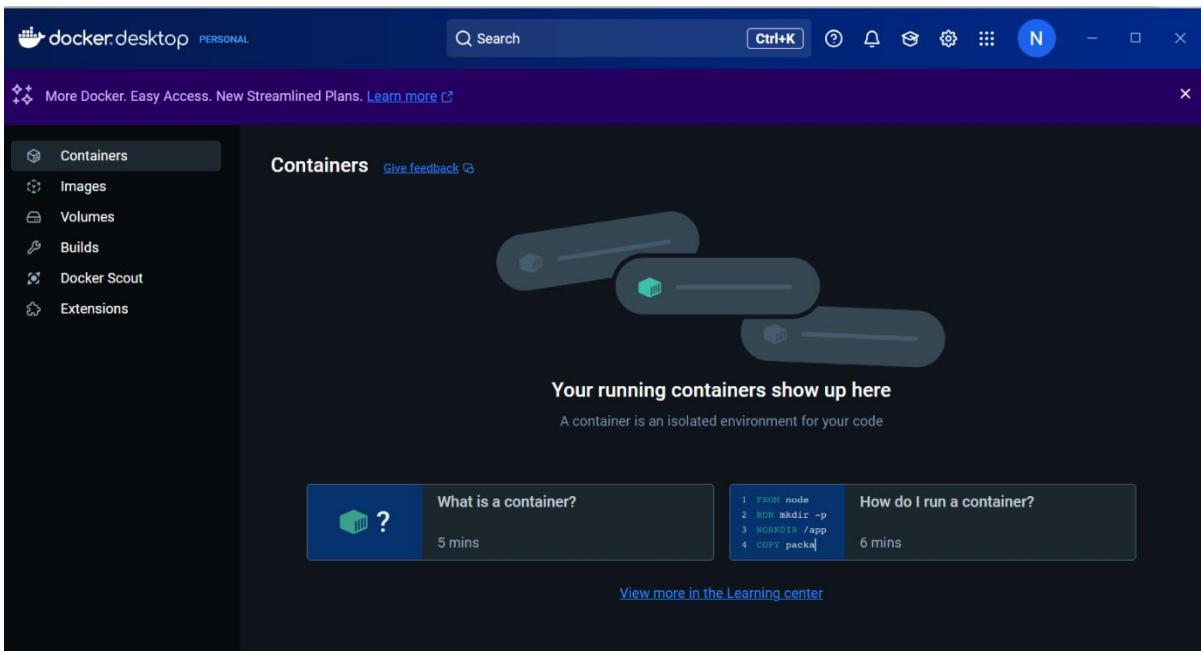


Step 2 : install Docker Desktop

- 2.1 go to Dckers website and download Docker Desktop for windows
- 2.2 Double click the downloaded file to start the installations
- 2.3 follow the steps in the installer
 - 2.3.1 accept the terms



- 2.3.2 choose to enable WSL2 instead of Hyper-V
- 2.3.3 install and wait



Docker version check

A screenshot of a Windows Command Prompt window titled 'C:\windows\system32\cmd.exe'. The window shows the following text:

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sride>docker -v
Docker version 27.3.1, build ce12230

C:\Users\sride>
```

The text is white on a black background, with command keywords in blue.

Dokcer run helloworld

```
PS C:\Users\sagar> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fc966
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

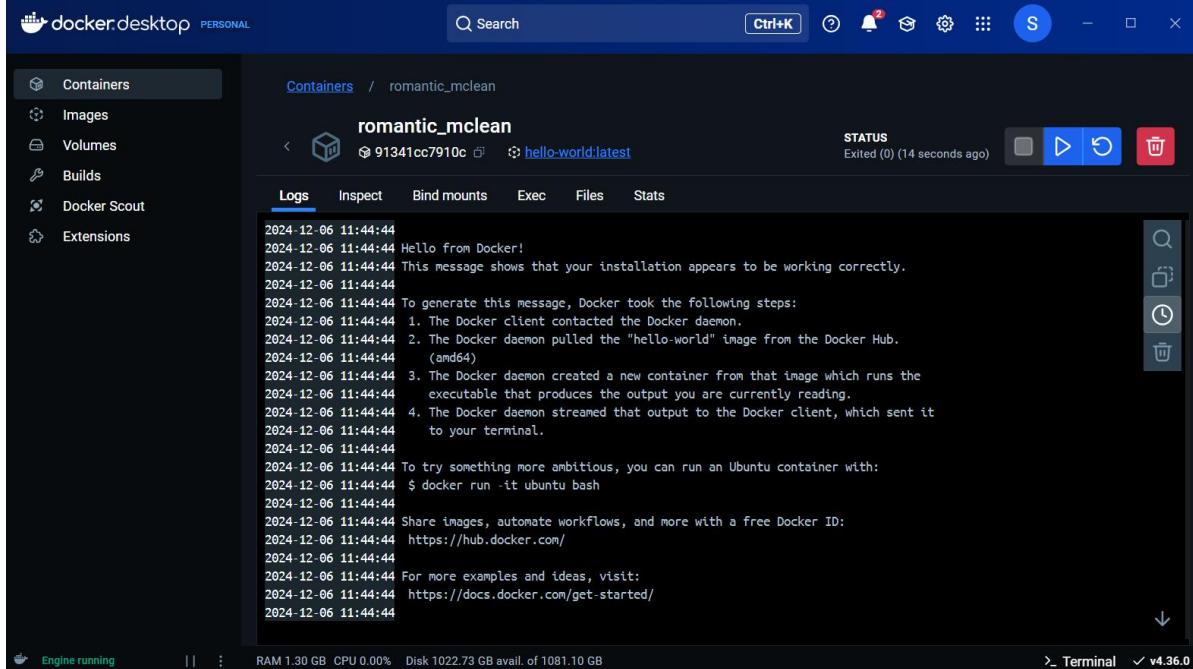
Share images, automate workflows, and more with a free Docker ID:

```
https://hub.docker.com/
```

For more examples and ideas, visit:

```
https://docs.docker.com/get-started/
```

```
PS C:\Users\sagar> |
```



Minikube Installation\

Step 1: Go to

<https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86->

[64%2Fstable%2F.exe+download](#) and select the appropriate installer based on your Operating System.

The screenshot shows the official minikube documentation website. The header features the minikube logo and the word "minikube". A search bar is present. On the left sidebar, there are several sections: "Documentation", "Get Started!", "Handbook", "Addons", and a long list of troubleshooting topics. The main content area is titled "1 Installation". It contains a note about selecting target platform buttons for "Operating system" (Linux, macOS, Windows), "Architecture" (x86-64), "Release type" (Stable), and "Installer type" (.exe download, Windows Pack). Below this, instructions are given for installing the latest stable release on x86-64 Windows via PowerShell, including a command snippet and a note to run as Administrator. Another command snippet for setting the PATH environment variable is also shown.

Search this site...

1 Installation

Click on the buttons that describe your target platform. For minikube binaries.

Operating system

Architecture

Release type

Installer type

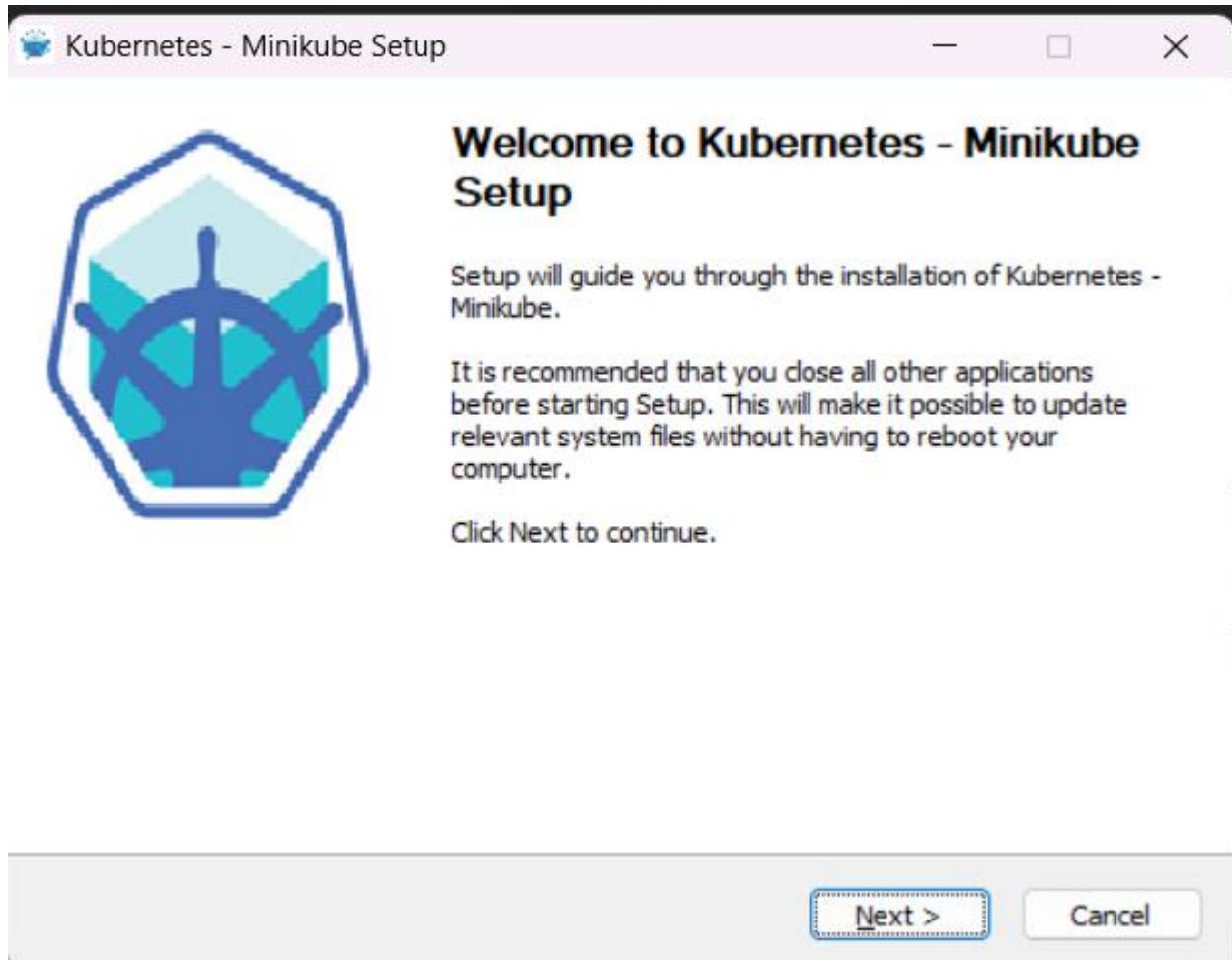
To install the latest minikube **stable** release on **x86-64 Windows**:

1. Download and run the installer for the [latest release](#).
Or if using `PowerShell`, use this command:

```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory  
Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe'
```
2. Add the `minikube.exe` binary to your `PATH`.
Make sure to run PowerShell as Administrator.

```
$oldPath = [Environment]::GetEnvironmentVariable('Path')  
if ($oldPath.Split(';') -inotcontains 'C:\minikube') {  
    [Environment]::SetEnvironmentVariable('Path', '$({0};{1})', 1)
```

Step 2: Now run the minikube installer and follow the steps below





License Agreement

Please review the license terms before installing Kubernetes - Minikube.



Press Page Down to see the rest of the agreement.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Kubernetes - Minikube.

< Back

I Agree

Cancel



Kubernetes - Minikube Setup



Choose Install Location

Choose the folder in which to install Kubernetes - Minikube.



Setup will install Kubernetes - Minikube in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\Program Files\Kubernetes\Minikube

[Browse...](#)

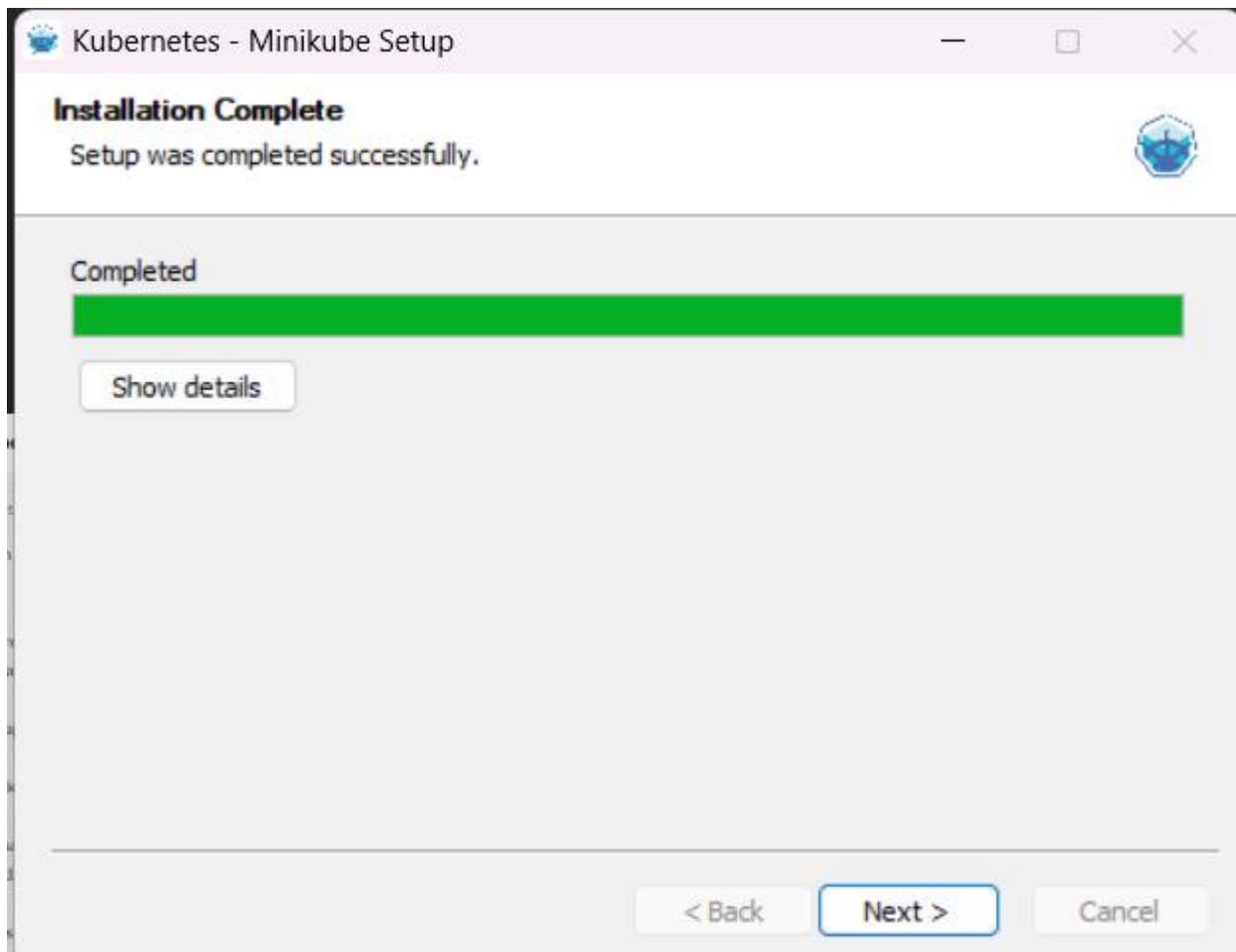
Space required: 102.0 MB

Space available: 55.8 GB

[< Back](#)

[Install](#)

[Cancel](#)



Step 3: Now open command prompt and run “minikube start”.

```
PS C:\Users\sagar> minikube start
🕒 minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4460 Build 22631.4460
💡 Using the docker driver based on existing profile
🔥 Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.45 ...
🕒 Restarting existing docker container for "minikube" ...
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🌐 Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
🌐 Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 4: Now, run “minikube status”. The following output shows that minikube is working properly

```
PS C:\Users\sagar> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

WEEK-10 Software Engineering Lab

DOCKER COMMANDS:

```
PS C:\windows\system32> docker --version
Docker version 27.3.1, build ce12230
PS C:\windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:5b3cc85e16e3058003c13b7821318369dad01dac3dbb877aac3c28182255c724
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
PS C:\windows\system32> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
kicbase/stable  v0.0.45  aeed0e1d4642  3 months ago  1.28GB
hello-world     latest    d2c94e258dcb  19 months ago  13.3kB
PS C:\windows\system32> docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:
<https://hub.docker.com/>

For more examples and ideas, visit:

```
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS        PORTS
1a43b939d308  hello-world   "/hello"
               sleepy_chandrasekhar
91341cc7910c  hello-world   "/hello"
               romantic_mclean
1879b82aa084  kicbase/stable:v0.0.45  "/usr/local/bin/entr..."
tcp, 127.0.0.1:32777->32443/tcp  minikube
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
bc0965b23a04: Pull complete
9501a6ec095f: Pull complete
98e7597530ef: Pull complete
75dfffa679c9b: Pull complete
8912a88e73c8: Pull complete
141f00d6fee8: Pull complete
4f4fb700ef54: Pull complete
8242f9d5b464: Pull complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
```

```

tcp, 127.0.0.1:32777->32443/tcp  minikube
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
bc0965b23a04: Pull complete
9501a6ec095f: Pull complete
98e7597530ef: Pull complete
75dfffa679c9b: Pull complete
8912a88e73c8: Pull complete
141f00d6fee8: Pull complete
4f4fb700ef54: Pull complete
8242f9d5b464: Pull complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\windows\system32> docker run --name newredis -d redis
548501d4395cef576059fa98e5bb1ed075a5bc73e327bc7c282861a78e89bf20
PS C:\windows\system32> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
              NAMES
548501d4395c   redis          "docker-entrypoint.s..."  32 seconds ago Up 31 seconds
                  newredis
1a43b939d308   hello-world    "/hello"
                  sleepy_chandrasekhar
91341cc7910c   hello-world    "/hello"
                  romantic_mclean
1879b82aa084   kicbase/stable:v0.0.45  "/usr/local/bin/entr..."
cp, 127.0.0.1:32777->32443/tcp  minikube
PS C:\windows\system32> docker exec -it newredis redis-cli
127.0.0.1:6379> SET name "ABCDEFG"
OK
127.0.0.1:6379>
PS C:\windows\system32> docker exec -it newredis redis-cli
127.0.0.1:6379> SET name "ABCDEFG"
OK
127.0.0.1:6379> GET name
"ABCDEFG"
127.0.0.1:6379> exit
PS C:\windows\system32>

PS C:\windows\system32> docker stop 548501d4395c
548501d4395c
PS C:\windows\system32> docker rm 548501d4395c
548501d4395c
PS C:\windows\system32> docker rmi redis
Untagged: redis:latest
Untagged: redis@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Deleted: sha256:b5e874b32a794f96b6119e5478f93165d4110cdd8c056d470b3d48b27114f716
Deleted: sha256:b29f5016befac03b5cde2788c31508900647ab8d456136768cb4743862739d1b
Deleted: sha256:6ae211a59cc93f2ff8a2d23914420f49632b93256cb925143a4b99ed07b8a1e6
Deleted: sha256:9138b812320895356a59879ae78ba94089af1f32ad80177967363bde5cac10d1
Deleted: sha256:36c5c6a38ba8a8cf7c71b8035ab30704bb905d54980a2528025dc90d8513543
Deleted: sha256:aea9e1fe4e796252e3fa7c8f6400b11717c601c8b23061aa702bcc0d61ad8108
Deleted: sha256:75850a6af24741b7465394e4b26811475e867fc793aa9cfb5cca7c27e1ed788a
Deleted: sha256:d734075709827a4cbe434847a33d25f14642a8b9f5589d80aa9b38ed82f3126f
Deleted: sha256:c0f1022b22a9b36851b358f44e5475e39d166e71a8073cf53c894a299239b1c5
PS C:\windows\system32>
```

PULLING AN IMAGE, RUNNING IT AS CONTAINER, MODIFYING IT, COMMITTING THE CHANGES WITH NEW IMAGE AND PUSHING INTO

DOKCER HUB

```
PS C:\windows\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
de44b265507a: Pull complete
Digest: sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
PS C:\windows\system32> docker run -it --name newubuntu -d ubuntu
9f6e2c1386f74af5940fdfd49781f5d2931d007988e67c2edbdb302cde7c6251
PS C:\windows\system32> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
9f6e2c1386f7        ubuntu              "/bin/bash"         4 seconds ago      Up 4 seconds       0.0.0.0:6379->6379/tcp   newubuntu
40e6fa8bb244        sagarpuppala/redisnewimage   "docker-entrypoint.s..."  2 minutes ago     Up 2 minutes      0.0.0.0:6379->6379/tcp   myredisnew
PS C:\windows\system32> docker exec -it 9f6e2c1386f7 bash
root@9f6e2c1386f7:/# git --version
bash: git: command not found
root@9f6e2c1386f7:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [67 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [19.3 MB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [19.3 MB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [19.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [117 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [117 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922 kB]
All packages are up to date.
root@9f6e2c1386f7:/# apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adduser ca-certificates git-man krb5-locales less libbrotli1 libbsas1
  libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 libl
  libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch p
Suggested packages:
  liblocale-gettext-perl cron quota ecryptfs-utils gettext-base git-d
  libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libsa
  libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap
The following NEW packages will be installed:
root@9f6e2c1386f7:/# git --version
git version 2.43.0
root@9f6e2c1386f7:/#
```

```

PS C:\windows\system32> docker login -u sagarpuppala
Password:
Login Succeeded
PS C:\windows\system32> docker push sagarpuppala/newubuntu2024
Using default tag: latest
The push refers to repository [docker.io/sagarpuppala/newubuntu2024]
0fefbf6876af: Pushed
687d50f2f6a6: Mounted from library/ubuntu
latest: digest: sha256:5ea369cb5fb98685715a7dff23be5aedd6a9cb8dbd6888225d33fabe648d6373 size: 741
PS C:\windows\system32> docker rm 9f6e2c1386f7
9f6e2c1386f7

PS C:\windows\system32> docker rmi sagarpuppala/newubuntu2024
Untagged: sagarpuppala/newubuntu2024:latest
Untagged: sagarpuppala/newubuntu2024@sha256:5ea369cb5fb98685715a7dff23be5aedd6a9cb8dbd6888225d33fabe648d6373
Deleted: sha256:5fbc1116a6d10f3d979889942fa8625326baa2267c809d9c7e65404801f695f9
Deleted: sha256:f226cb12155882ee006fd400a72aaaf804ade85ff37158cfb08a549710da85fa0
PS C:\windows\system32> docker logout
Removing login credentials for https://index.docker.io/v1/
PS C:\windows\system32> docker pull sagarpuppala/newubuntu2024
Using default tag: latest
latest: Pulling from sagarpuppala/newubuntu2024
de44b265507a: Already exists
a76fd1b9b633: Pull complete
Digest: sha256:5ea369cb5fb98685715a7dff23be5aedd6a9cb8dbd6888225d33fabe648d6373
Status: Downloaded newer image for sagarpuppala/newubuntu2024:latest
docker.io/sagarpuppala/newubuntu2024:latest
PS C:\windows\system32> docker run --name newubuntu2024 -it sagarpuppala/newubuntu2024
root@a5d9326eaba7:/# git --version
git version 2.43.0
root@a5d9326eaba7:/# exit
exit
PS C:\windows\system32> docker ps -a
CONTAINER ID IMAGE COMMAND NAMES CREATED STATUS
a5d9326eaba7 sagarpuppala/newubuntu2024 "/bin/bash" 31 seconds ago Exited (0) 3 seconds ago
40e6fa8bb244 sagarpuppala/redisnewimage "docker-entrypoint.s..." 29 minutes ago Up 29 minutes
1a43b939d308 hello-world "/hello" sleepy_chandrasekhar About an hour ago Exited (0) About an hour ago
91341cc7910c hello-world "/hello" romantic_mclean 6 days ago Exited (0) 6 days ago
1879b82aa084 kicbase/stable:v0.0.45 "/usr/local/bin/entr..." 8 days ago Exited (255) 6 days ago
776->8443/tcp, 127.0.0.1:32777->32443/tcp minikube
PS C:\windows\system32> dokcer rm a5d9326eaba7

PS C:\windows\system32> docker rm a5d9326eaba7
a5d9326eaba7
PS C:\windows\system32> docker rmi sagarpuppala/newubuntu2024
Untagged: sagarpuppala/newubuntu2024:latest
Untagged: sagarpuppala/newubuntu2024@sha256:5ea369cb5fb98685715a7dff23be5aedd6a9cb8dbd6888225d33fabe6
Deleted: sha256:5fbc1116a6d10f3d979889942fa8625326baa2267c809d9c7e65404801f695f9
Deleted: sha256:f226cb12155882ee006fd400a72aaaf804ade85ff37158cfb08a549710da85fa0
PS C:\windows\system32>

```

CREATING A NEW IMAGE USING DOCKER FILE, PUSHING INTO DOCKER HUB AND RUNNING IT

C:\> This PC > New Volume (D:) > SUNNY > SELAB > DOCKER_PRO > Redis

Name	Date modified	Type	Size
Dockerfile	13-12-2024 10:23	File	1 KB

```
PS C:\Windows\system32> cd D:\SUNNY\SELAB\DOCKER_PRO\Redis
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> ls

Directory: D:\SUNNY\SELAB\DOCKER_PRO\Redis

Mode                LastWriteTime         Length Name
----                - - - - -           - - - - -
-a--- 13-12-2024      10:10          39 Dockerfile.txt

PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker build -t myredis .
>>
[+] Building 12.7s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 76B
=> [internal] load metadata for docker.io/library/redis:latest
=> [auth] library/redis:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/1] FROM docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
=> => resolving docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
=> => sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83 10.26kB / 10.26kB
=> => sha256:b5e874b32a794f96b6119e5478f93165d4110cd8c056d470b3d48b27114f716 8.59kB / 8.59kB
=> => sha256:fa84a527b1808ee5ad2778e5d89aae791e7040b5c303f1d45569a9ca8d5b61ee 2.48kB / 2.48kB
=> => sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f 28.23MB / 28.23MB
=> => sha256:9501a6ec095f4bd242b8ceefdf119d098b45e9237c53cedb022a20ce9c1fa 1.10kB / 1.10kB
=> => sha256:98e7597530ef912e6a2cce5e6e0cd3f155b611a5d6b6ea7823640b6c4f2c4a30 8748 / 8748
=> => sha256:75dfffa679c9bb360fd358682b0842b9f32a732e6f80d47e8a7f3efde5f05dc07 1.44MB / 1.44MB
=> => sha256:8912a88e73c83f622d19fd2dac8a055ca96342628d29d56b81fde09475ef7007 15.33MB / 15.33MB
=> => sha256:141f00d6fee86bf601ac2eca3db312a24d3e4147bc166bbecb09428be54794b5 978 / 978
=> => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B
=> => sha256:8242f9d5b464f375c010545626f3107bcfe0f68e7a2a78e95c8d397427a12c0e 572B / 572B
=> => extracting sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f
=> => extracting sha256:9501a6ec095f4bd242b8ceefdf119d098b45c6e9237c53cedb022a20ce9c1fa
=> => extracting sha256:98e7597530ef912e6a2cce5e6e0cd3f155b611a5d6b6ea7823640b6c4f2c4a30
=> => extracting sha256:75dfffa679c9bb360fd358682b0842b9f32a732e6f80d47e8a7f3efde5f05dc07
=> => extracting sha256:8912a88e73c83f622d19fd2dac8a055ca96342628d29d56b81fde09475ef7007
=> => extracting sha256:141f00d6fee86bf601ac2eca3db312a24d3e4147bc166bbecb09428be54794b5
=> => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1
=> => extracting sha256:8242f9d5b464f375c010545626f3107bcfe0f68e7a2a78e95c8d397427a12c0e
=> exporting to image
=> => exporting layers
=> => writing image sha256:b811c987af0831dd5a1712170ea61a95f065f13a4948c8af664a8b547c4d1928
=> => naming to docker.io/library/myredis
```

```

PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
myredis         latest   b811c987af08  2 months ago  117MB
kicbase/stable v0.0.45  aeed0e1d4642  3 months ago  1.28GB
hello-world     latest   d2c94e258dcf  19 months ago  13.3kB
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker run --name newmyredis -d myredis
2b71e60cebb48e4753e4bb278959d6b4b72524f0bf96d2eb71be37da3e865ad9
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS        PORTS
NAMES
2b71e60cebb4  myredis        "docker-entrypoint.s..."  8 seconds ago  Up 7 seconds  6379/tcp
               newmyredis
1a43b939d308  hello-world    "/hello"      27 minutes ago  Exited (0) 27 minutes ago
91341cc7910c  hello-world    "sleepy_chandrasekhar"  6 days ago   Exited (0) 6 days ago
1879bb82aa084  kicbase/stable:v0.0.45  "/usr/local/bin/entr..."  8 days ago   Exited (255) 6 days ago  127.0.0.1:32773->22/tcp, 127.0.0.1:32777->32443/tcp
2b71e60cebb4
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker stop 2b71e60cebb4
2b71e60cebb4
Start a build
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker commit 2b71e60cebb4 sagarpuppala/redisnewimage
sha256:2aab01517818ea1b66070e56b9810ebf03e98deb8117315f9468e1553b53fea8
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker images
REPOSITORY          TAG      IMAGE ID      CREATED      SIZE
sagarpuppala/redisnewimage  latest   2aab01517818  52 seconds ago  117MB
myredis             latest   b811c987af08  2 months ago  117MB
kicbase/stable      v0.0.45  aeed0e1d4642  3 months ago  1.28GB
hello-world         latest   d2c94e258dcf  19 months ago  13.3kB
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis>
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker login
Authenticating with existing credentials...
Login Succeeded
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker push sagarpuppala/redisnewimage
Using default tag: latest
The push refers to repository [docker.io/sagarpuppala/redisnewimage]
4c3a66986d54: Mounted from library/redis
5f70bf18a086: Mounted from library/redis
59858d305cb4: Mounted from library/redis
3c008636f68d: Mounted from library/redis
d3ce9d926f3b: Mounted from library/redis
d575106a8e12: Mounted from library/redis
74264edca52d: Mounted from library/redis
c0f1022b22a9: Mounted from library/redis
latest: digest: sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2c491b0e size: 1986

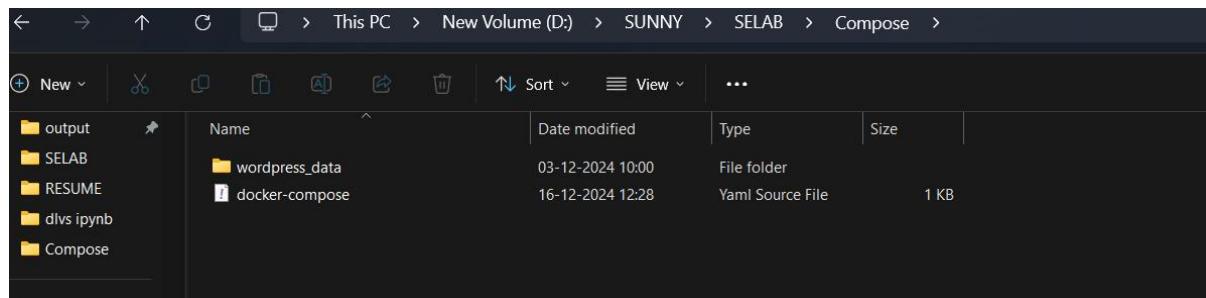
```

```
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker rm 2b71e60cebb4
2b71e60cebb4
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker rmi sagarpuppala/redisnewimage
Untagged: sagarpuppala/redisnewimage:latest
Untagged: sagarpuppala/redisnewimage@sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2
Deleted: sha256:2aab01517818ea1b66070e56b9810ebf03e98deb8117315f9468e1553b53fea8
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker pull sagarpuppala/redisnewimage
Using default tag: latest
latest: Pulling from sagarpuppala/redisnewimage
bc0965b23a04: Already exists
9501a6ec095f: Already exists
98e7597530ef: Already exists
75dfffa679c9b: Already exists
8912a88e73c8: Already exists
141f00d6fee8: Already exists
4f4fb700ef54: Already exists
8242f9d5b464: Already exists
Digest: sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2c491b0e
Status: Downloaded newer image for sagarpuppala/redisnewimage:latest
docker.io/sagarpuppala/redisnewimage:latest
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker run --name myredisnew -d sagarpuppala/redisnewimage
40e6fa8bb244f3ec9ad7326fe845de5af5bd09d1a6384bba3ab097fae118b187
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker exec -it myredisnew redis-cli
127.0.0.1:6379> exit
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker logout
Removing login credentials for https://index.docker.io/v1/
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> ■
```

```
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker rm 2b71e60cebb4
2b71e60cebb4
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker rmi sagarpuppala/redisnewimage
Untagged: sagarpuppala/redisnewimage:latest
Untagged: sagarpuppala/redisnewimage@sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2c491b0e
Deleted: sha256:2aab01517818ea1b66070e56b9810ebf03e98deb8117315f9468e1553b53fea8
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker pull sagarpuppala/redisnewimage
Using default tag: latest
latest: Pulling from sagarpuppala/redisnewimage
bc0965b23a04: Already exists
9501a6ec095f: Already exists
98e7597530ef: Already exists
75dfffa679c9b: Already exists
8912a88e73c8: Already exists
141f00d6fee8: Already exists
4f4fb700ef54: Already exists
8242f9d5b464: Already exists
Digest: sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2c491b0e
Status: Downloaded newer image for sagarpuppala/redisnewimage:latest
docker.io/sagarpuppala/redisnewimage:latest
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker run --name myredisnew -d sagarpuppala/redisnewimage
40e6fa8bb244f3ec9ad7326fe845de5af5bd09d1a6384bba3ab097fae118b187
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker exec -it myredisnew redis-cli
127.0.0.1:6379> exit
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker logout
Removing login credentials for https://index.docker.io/v1/
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis>
```

Docker Compose and installation of Minikube and working with it

1 DOKCER COMPOSE:



dokcer-compose.yaml file

Welcome docker-compose.yaml

```
docker-compose.yaml
1  version: '3'
2
3  services:
4    # Database
5    db:
6      image: mysql:5.7
7      volumes:
8        - db_data:/var/lib/mysql
9      restart: always
10     environment:
11       MYSQL_ROOT_PASSWORD: password
12       MYSQL_DATABASE: wordpress
13       MYSQL_USER: wordpress
14       MYSQL_PASSWORD: wordpress
15     networks:
16       - wpsite
17
18  # WordPress
19  wordpress:
20    depends_on:
21      - db
22    image: wordpress:latest
23    ports:
24      - '8000:80'
25    restart: always
26    volumes:
27      - ./wordpress_data:/var/www/html
28    environment:
29      WORDPRESS_DB_HOST: db:3306
30      WORDPRESS_DB_USER: wordpress
31      WORDPRESS_DB_PASSWORD: wordpress
32    networks:
33      - wpsite
34
35  networks:
36    wpsite:
37
38  volumes:
39    db_data:
```

```
PS D:\SUNNY\SELAB\Compose> docker-compose up -d
time="2024-12-16T12:30:06+05:30" level=warning msg="D:\\SUNNY\\SELAB\\Compose\\ will be ignored, please remove it to avoid potential confusion"
[+] Running 35/13
  ✓ wordpress Pulled
  ✓ db Pulled

[+] Running 4/4
  ✓ Network compose_wpsite          Created
  ✓ Volume "compose_db_data"        Created
  ✓ Container compose-db-1          Started
  ✓ Container compose-wordpress-1   Started
PS D:\SUNNY\SELAB\Compose>
```

localhost:8000

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

Username

SagarPuppala

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

Password

Sagar@3101

 Hide

Medium

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

Your Email

sagarpuppala123@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](#)



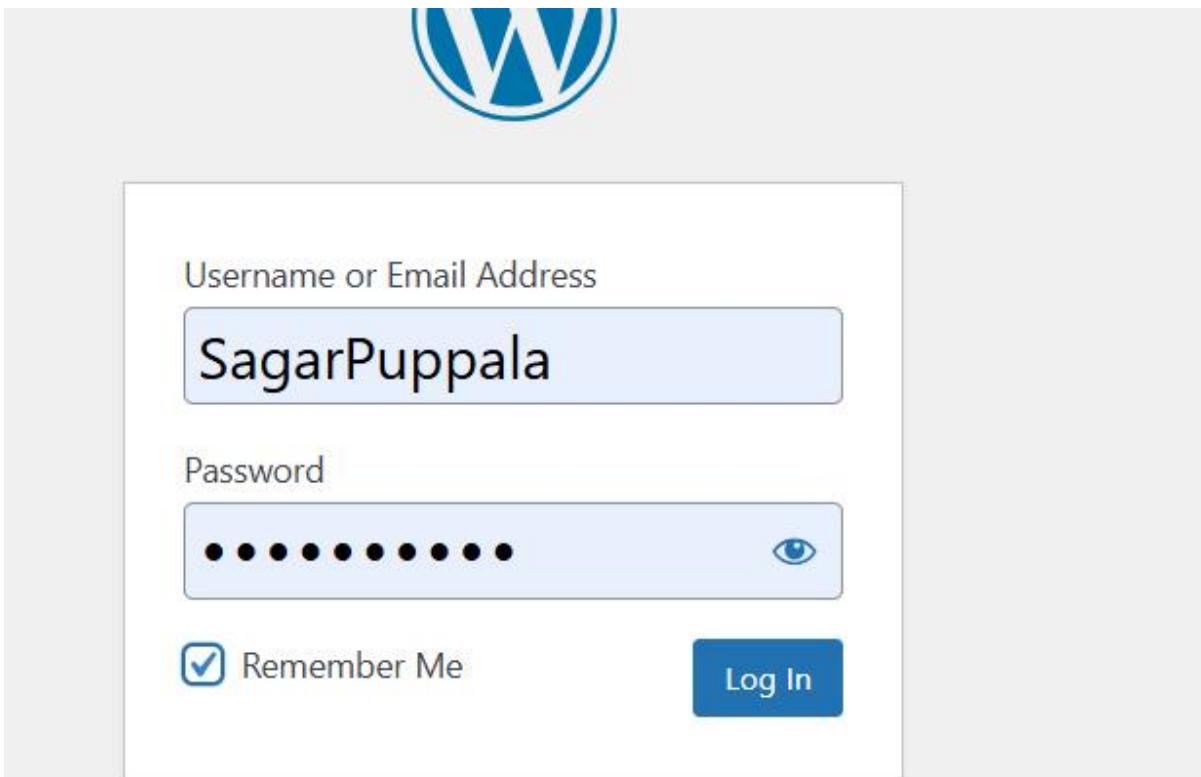
Success!

WordPress has been installed. Thank you, and enjoy!

Username SagarPuppala

Password *Your chosen password.*

[Log In](#)



[Lost your password?](#)

[← Go to](#)

A screenshot of the WordPress dashboard. The top navigation bar shows 'localhost:8000' and the user 'Howdy, SagarPuppala'. The dashboard sidebar includes links for Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main content area features a large 'Welcome to WordPress!' message and a 'Learn more about the 6.7.1 version.' link. Below this are three cards: one for 'Author rich content with blocks and patterns', one for 'Customize your entire site with block themes', and one for 'Switch up your site's look & feel with Styles'. At the bottom, there are sections for 'Site Health Status' and 'Quick Draft'.

```

✓ Container compose-wordpress-1 Started
● PS D:\SUNNY\SELAB\Compose> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6ceee4d4a1c4 wordpress:latest "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 0.0.0.0:8000->80/tcp compose-wordpress-1
f50a87dcac33 mysql:5.7 "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 3306/tcp, 33060/tcp compose-db-1
● PS D:\SUNNY\SELAB\Compose> docker-compose stop
time="2024-12-16T12:45:25+05:30" level=warning msg="D:\\SUNNY\\SELAB\\Compose\\docker-compose.yaml: the attribute `version` is obsolete, please remove it to avoid potential confusion"
[+] Stopping 2/2
✓ Container compose-wordpress-1 Stopped
✓ Container compose-db-1 Stopped
PS D:\SUNNY\SELAB\Compose> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS D:\SUNNY\SELAB\Compose> docker-compose start
time="2024-12-16T12:45:56+05:30" level=warning msg="D:\\SUNNY\\SELAB\\Compose\\docker-compose.yaml: the attribute `version` is obsolete, please remove it to avoid potential confusion"
[+] Running 2/2
✓ Container compose-db-1 Started
✓ Container compose-wordpress-1 Started
PS D:\SUNNY\SELAB\Compose> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
6ceee4d4a1c4 wordpress:latest "docker-entrypoint.s..." 5 minutes ago Up 2 seconds 0.0.0.0:8000->80/tcp
f50a87dcac33 mysql:5.7 "docker-entrypoint.s..." 5 minutes ago Up 3 seconds 3306/tcp, 33060/tcp
PS D:\SUNNY\SELAB\Compose> docker-compose stop
time="2024-12-16T12:46:10+05:30" level=warning msg="D:\\SUNNY\\SELAB\\Compose\\docker-compose.yaml: the attribute `version` is obsolete, please remove it to avoid potential confusion"
[+] Stopping 2/2
✓ Container compose-wordpress-1 Stopped
✓ Container compose-db-1 Stopped
PS D:\SUNNY\SELAB\Compose>

```

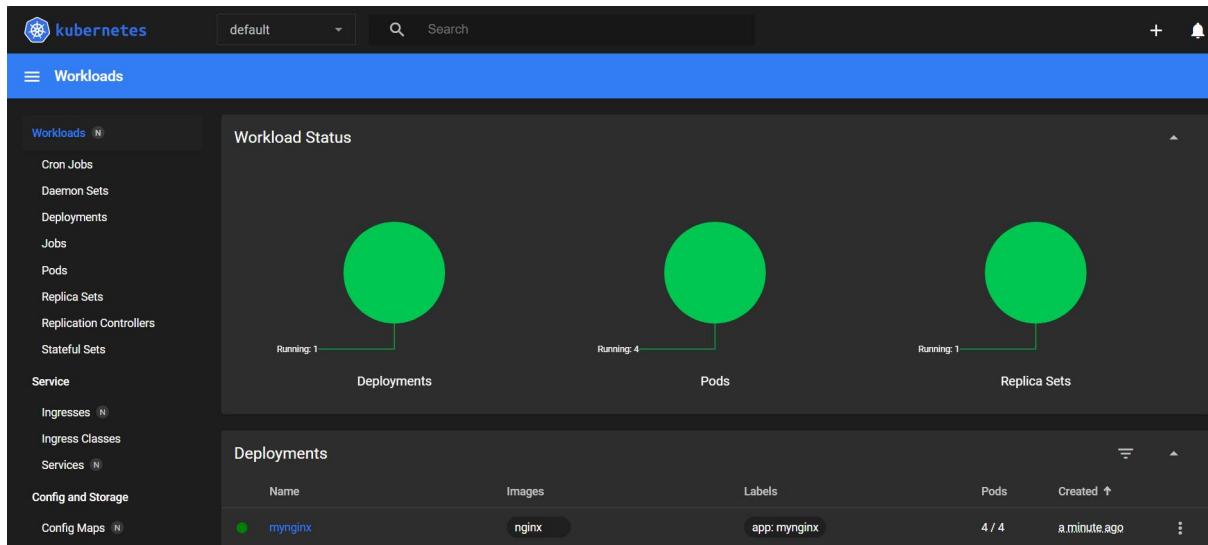
2 MINIKUBE

```

PS C:\WINDOWS\system32> minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.26100.2605 Build 26100.2605
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/tutorials/config/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

```

```
PS C:\WINDOWS\system32> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\WINDOWS\system32> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   1/1     1           1           75s
PS C:\WINDOWS\system32> kubectl expose deployment mynginx --type=NodePort --port=80
service/mynginx exposed
PS C:\WINDOWS\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\WINDOWS\system32> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   4/4     4           4           4m48s
PS C:\WINDOWS\system32> kubectl get pods
NAME                           READY   STATUS    RESTARTS   AGE
mynginx-79bb8756c7-6x898     1/1     Running   0          26s
mynginx-79bb8756c7-bgmrj     1/1     Running   0          26s
mynginx-79bb8756c7-mh2nq     1/1     Running   0          4m57s
mynginx-79bb8756c7-x7fpn     1/1     Running   0          26s
PS C:\WINDOWS\system32> 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
PS C:\WINDOWS\system32> 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
PS C:\WINDOWS\system32> kubectl delete deployment mynginx
deployment.apps "mynginx" deleted
PS C:\WINDOWS\system32> kubectl delete service mynginx
service "mynginx" deleted
PS C:\WINDOWS\system32> minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.26100.2605 Build 26100.2605
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Updating the running docker "minikube" container ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/
PS C:\Users\sagar> minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:54520/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard.default:443/proxy
ur default browser...
|
```



NAGIOS INSTALLATION:

1. Pull nagios latest from the jasonrivers in docker

```
PS C:\WINDOWS\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
ff65ddf9395b: Pull complete
785b9873bdf4: Extracting [=====] 42.34MB/2
9ef9446ba5cc: Download complete
53afff88babc4: Download complete
d72f92e29533: Download complete
706ed7d4ce0a: Download complete
d3245570f968: Download complete
e58e184b986a: Download complete
eeb77e6dde3e: Download complete
9bd0f5795eeb: Download complete
71fbfb306f8cb: Download complete
738fc7520889: Download complete
fe8a6b2cf4e3: Download complete
e6f8fab512d1: Download complete
15f36d0b0439: Download complete
a2fc4187e3b4: Download complete
3d5785144815: Download complete
566cdc02555d: Download complete
c700be87d617: Download complete
4f4fb700ef54: Download complete
b69c76bd2b6b: Download complete
d5aa2a3a6539: Download complete
3fb30af17153: Download complete
9ffe54c5c139: Download complete
279b28aefa10: Download complete
a900dfcceeb38: Download complete
9a90645e352c: Download complete
3e911c59da28: Download complete
c219d58cc3f9: Download complete
b0e280e9aa8c: Download complete
```

Run It by giving any custom name

```

PS C:\WINDOWS\system32> docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest
Adding password for user nagiosadmin
chown: warning: '.' should be ':' 'nagios.nagios'
Started runsvdir, PID is 13
checking permissions for nagios & nagiosgraph
rsyslogd: [origin software="rsyslogd" swVersion="8.2312.0" x-pid="22" x-info="https://www.rsyslog.com"] start

Nagios Core 4.5.7
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2024-10-24
License: GPL

Website: https://www.nagios.org
Nagios 4.5.7 starting... (PID=25)
Local time is Mon Dec 16 10:10:35 UTC 2024
nagios: Nagios 4.5.7 starting... (PID=25)
nagios: Local time is Mon Dec 16 10:10:35 UTC 2024
nagios: LOG VERSION: 2.0
wproc: Successfully registered manager as @wproc with query handler
nagios: qh: Socket '/opt/nagios/var/rw/nagios.qh' successfully initialized
nagios: qh: core query handler registered
nagios: qh: echo service query handler registered
nagios: qh: help for the query handler registered
nagios: wproc: Successfully registered manager as @wproc with query handler

```

3 NAGIOS WEB PAGE

The screenshot shows the Nagios Core 4.5.7 web interface. On the left, there's a sidebar with navigation links like General, Home, Documentation, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, Reports, Availability, Trends, and Alerts. The main content area displays a dashboard with several cards: 'System Status' showing a grid of host and service icons, 'Server Statistics' with CPU and RAM usage, 'Status Grid' showing hosts and services, and 'Latest Alerts' with a single entry for a memory usage alert. A blue banner at the top right says 'A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.5.8.'

4 CHECK WHETHER THE NAGIOS IS RUNNING OR NOT

5 STOP NAGIOS DEMO

```

PS C:\Users\sagar> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
1665c9941fa4        jasonrivers/nagios:latest   "/usr/local/bin/star..."   5 minutes ago      Up 5 min          nagiosdemo
6ceee4d4a1c4        wordpress:latest       "docker-entrypoint.s..."   3 hours ago       Up 2 hours        compose-wordpr
f50a87dcac33        mysql:5.7                 "docker-entrypoint.s..."   3 hours ago       Up 2 hours        compose-db-1
1879b82aa084        kicbase/stable:v0.0.45     "/usr/local/bin/entr..."   12 days ago       Up 2 hours        minikube
PS C:\Users\sagar> docker stop nagiosdemo
nagiosdemo
PS C:\Users\sagar>

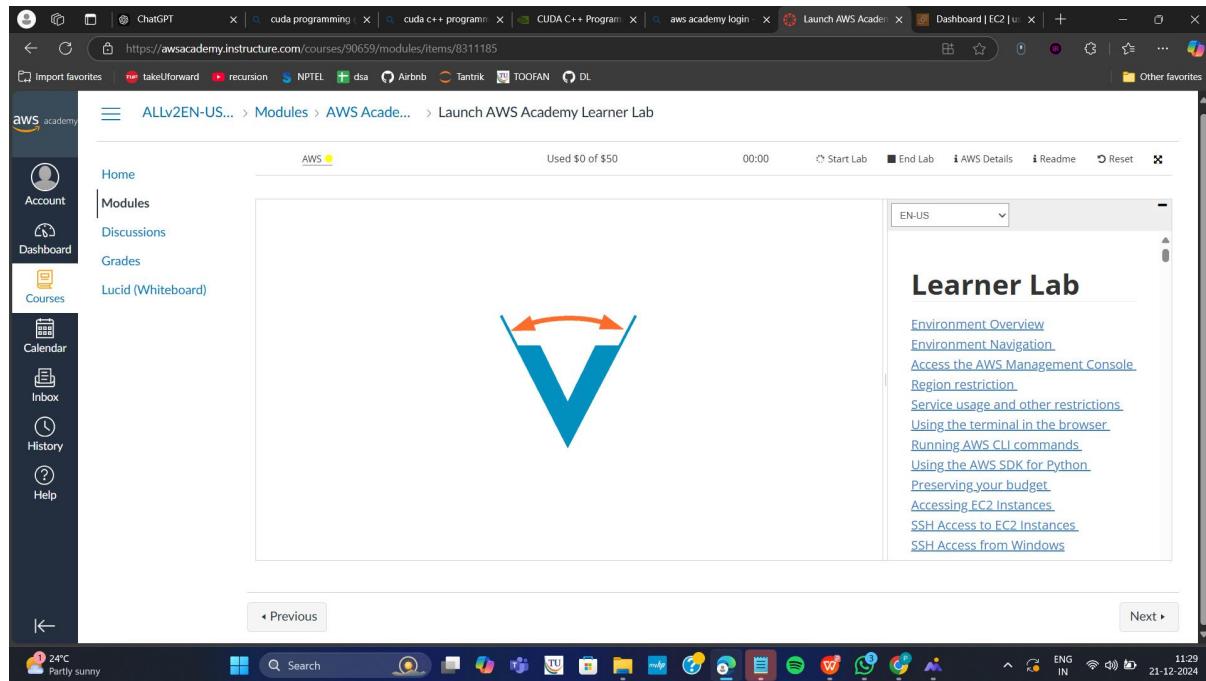
```

```
PS C:\Users\sagar> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
6ceee4d4a1c4        wordpress:latest   "docker-entrypoint.s..." 3 hours ago       Up 2 hours        0.0.0.0:8000->80/tcp
f50a87dcac33        mysql:5.7         "docker-entrypoint.s..." 3 hours ago       Up 2 hours        3306/tcp, 33060/tcp
1879b82aa084        kicbase/stable:v0.0.45 "/usr/local/bin/entr..." 12 days ago      Up 2 hours        127.0.0.1:50577->22/tcp, 127.0
575->5000/tcp, 127.0.0.1:50576->8443/tcp, 127.0.0.1:50574->32443/tcp   minikube
PS C:\Users\sagar> |
```

WEEK-12

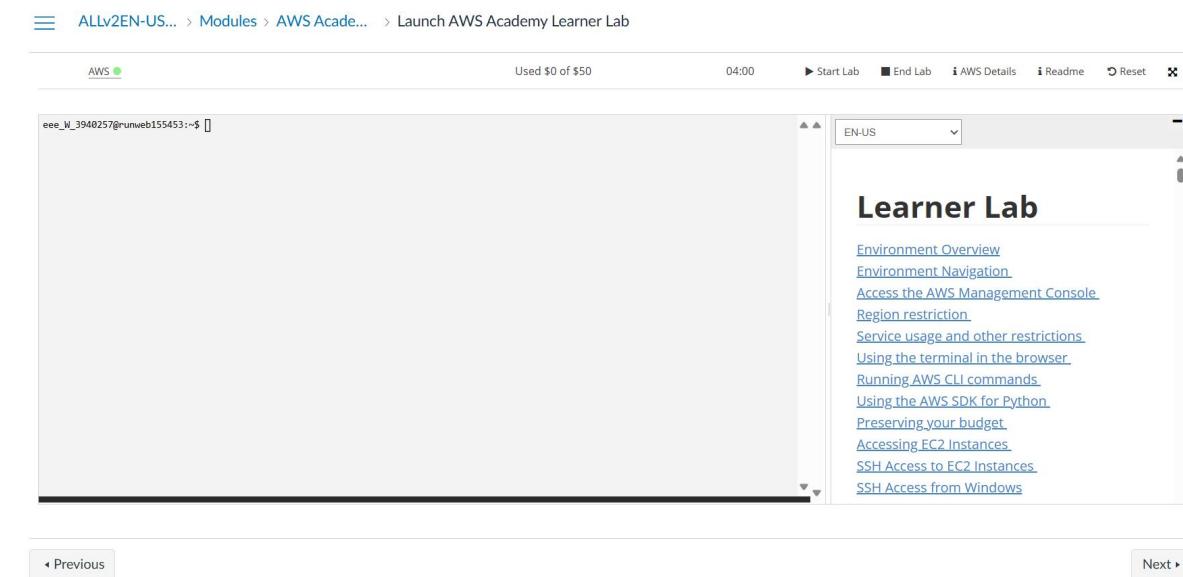
1 DEPLOYMENT OF INDEX.HTML USING EC2 INSTANCE in AWS

Click on start lab



The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with navigation links: Home, Modules, Discussions, Grades, Courses, Calendar, Inbox, History, and Help. The main area is titled "Learner Lab" and contains a large blue downward-pointing arrow with a red curved arrow at the top. At the bottom of the main area, there are "Previous" and "Next" buttons. A toolbar at the top includes "Start Lab", "End Lab", "AWS Details", "Readme", and "Reset". The status bar at the bottom shows the date and time as 21-12-2024.

Click on AWS



The screenshot shows the AWS Academy Learner Lab interface. The terminal window displays the command "eee_w_3940257@runweb155453:~\$". The right panel is identical to the previous screenshot, showing the "Learner Lab" section with various links. Navigation buttons "Previous" and "Next" are at the bottom.

Click on EC2

The screenshot shows the AWS Console Home page. On the left, there's a 'Recently visited' section with a single entry for 'EC2'. To the right, there's a 'Applications' section showing 0 applications in the US East (N. Virginia) region. Other visible cards include 'Welcome to AWS', 'AWS Health', and 'Cost and usage'.

Click on Launch Instance

This screenshot shows the EC2 Global View dashboard. It displays various EC2 resources: 0 instances (running), 0 auto scaling groups, 0 capacity reservations, 0 dedicated hosts, 0 elastic IPs, 0 instances, 1 key pair, 0 load balancers, 0 placement groups, 1 security group, 0 snapshots, and 0 volumes. Below this, there's a 'Launch instance' section with a 'Launch instance' button and a note about launching instances in the US East (N. Virginia) region. The 'Service health' section shows the status of the AWS Health Dashboard. The 'Account attributes' section includes settings for the Default VPC and other EC2-related configurations. A sidebar on the left lists navigation links for Dashboard, Instances, Images, and Elastic Block Store.

Give name and select ubuntu under application

This screenshot shows the 'Launch an instance' wizard. Step 1, 'Name and tags', has a 'Name' field containing 'MyExampleWebServer'. Step 2, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a list of AMIs including Amazon Linux, macOS, Ubuntu (selected), Windows, Red Hat, SUSE Linux, and Debian. Step 3, 'Quick Start', shows a grid of icons for different AMI types. The right side of the screen shows a sidebar with a 'Browse more AMIs' link.

Make sure AMI and Architecture are there as shown

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-0e2c8caa4b6378d8c (64-bit (x86)) / ami-0932ffb346ea84d48 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible ▾
--	----------------------

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture 64-bit (x86) ▾	AMI ID ami-0e2c8caa4b6378d8c	Username ⓘ ubuntu	Verified provider
---------------------------------------	--	-------------------------------	--------------------------

Click on create-new key pair

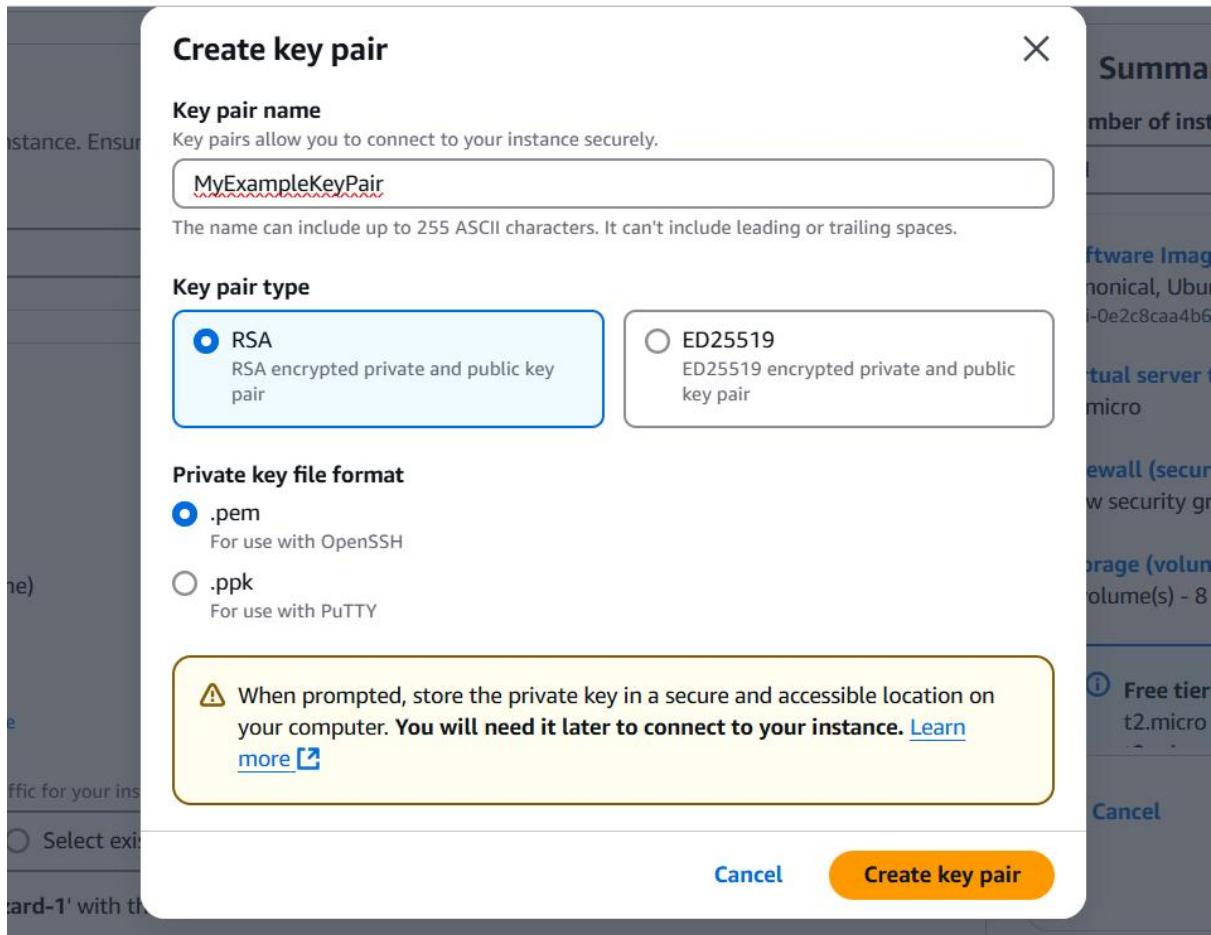
▼ **Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

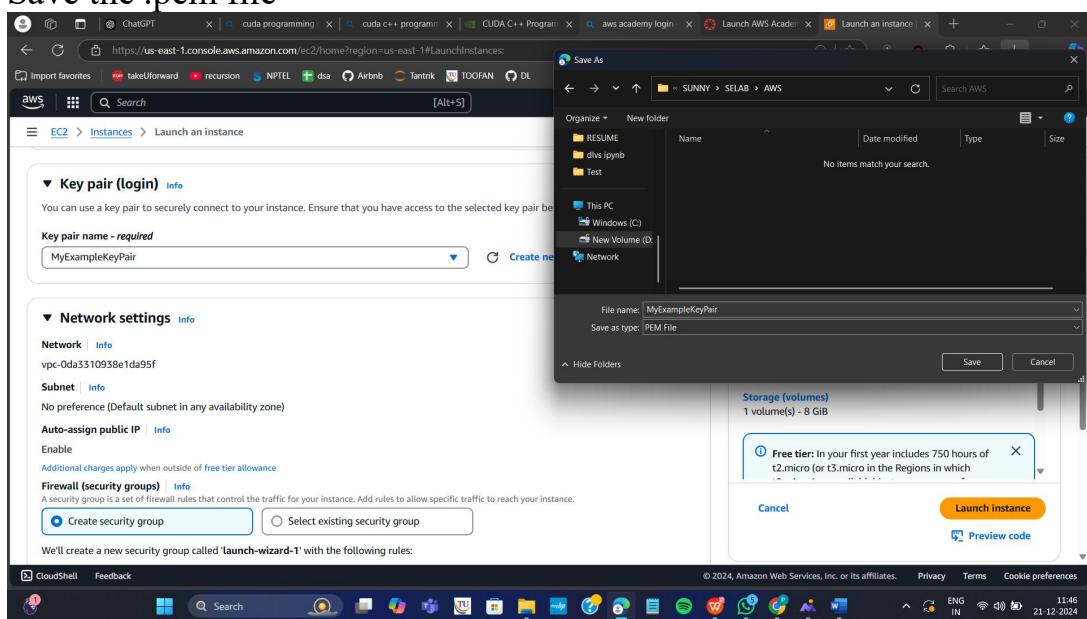
Key pair name - required

Select ▾ ⟳ Create new key pair

Give KeyPair name and click on create key pair



Save the .pem file



In network setting check all the checkboxes and click on launch instance which is at the bottom right to the page

▼ Network settings [Info](#)

[Edit](#)

Network | [Info](#)
vpc-0da3310938e1da95f

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable
Additional charges apply when outside of free tier allowance

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

We'll create a new security group called '**launch-wizard-1**' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ 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 [X](#)

Wait for the Success message.

Now click on the instances

[aws](#) | [Search](#) [Alt+S] N. Virginia v vociabs/user3722153=sagarpuppala123@gmail.com @ 4891-1941-8681 ▾

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-038830a6ac9f80be1)

[Launch log](#)

Next Steps
Q: What would you like to do next with this instance, for example "create alarm" or "create backup" 1 2 3 4 5 6 >

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
[Create billing alerts](#)

Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#) [Learn more](#)

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
[Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots.
[Create EBS snapshot policy](#)

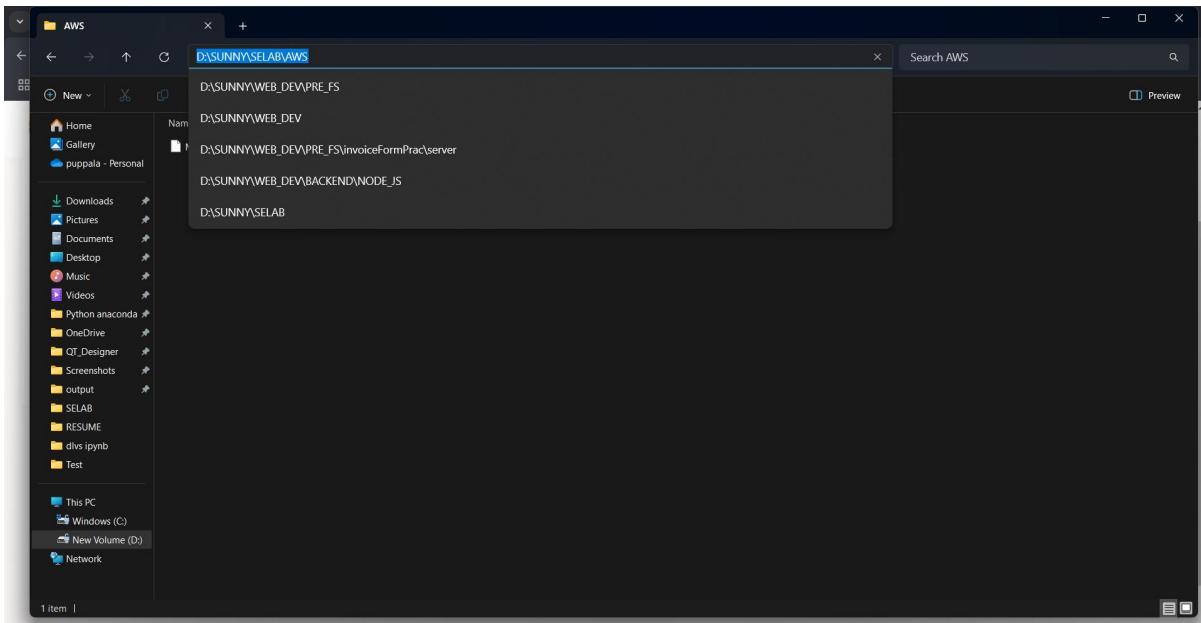
You can see MyExampleWebServer is Running and wait for it to initialize

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main content area displays 'Instances (1) Info' with a table. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. One row is shown: MyExampleWe... (Instance ID i-038830a6ac9f80be1), Running, t2.micro, Initializing, View alarms +, us-east-1c, ec2-3-80-1. Below the table, a section titled 'Select an instance' is visible. At the bottom of the page, there's a footer with links for CloudShell and Feedback, along with system status icons.

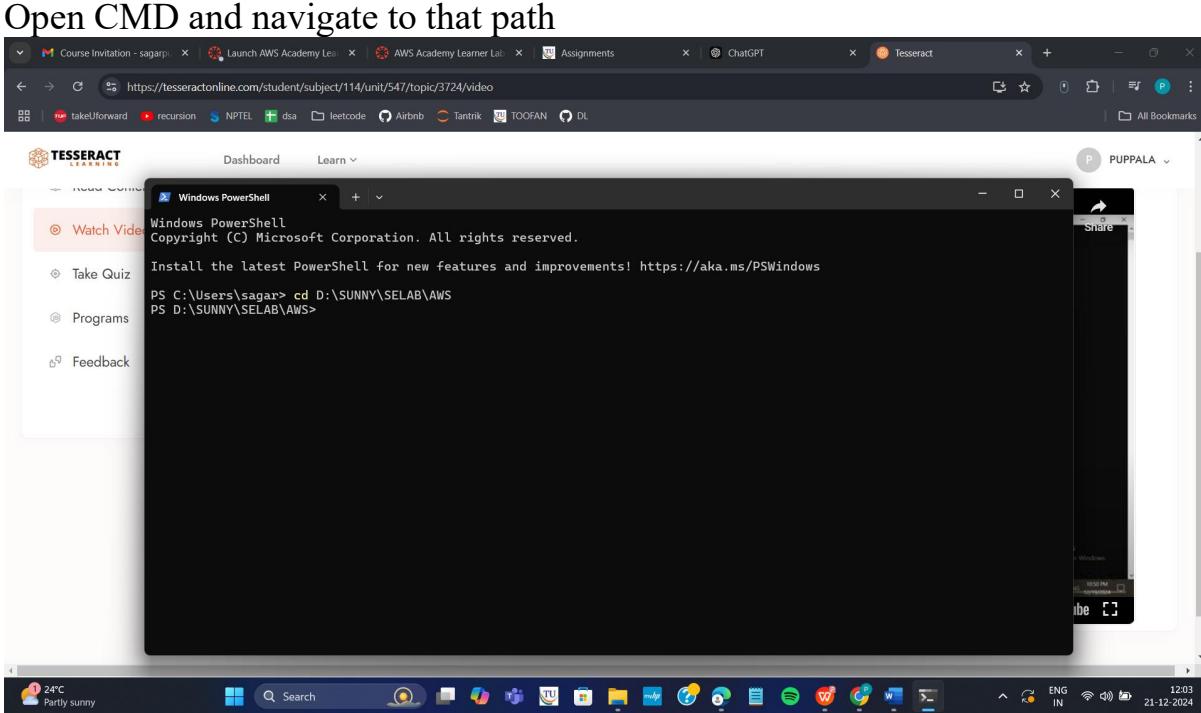
You have get 2 tests passes. Now check the box and click on connect.

This screenshot is from the same EC2 Instances page as the previous one, but with a key difference: the checkbox next to the instance name 'MyExampleWebServer' is checked. This triggers the 'Connect' button, which is now highlighted in orange. The rest of the interface remains the same, showing the instance details and the 'Details' tab selected in the instance summary.

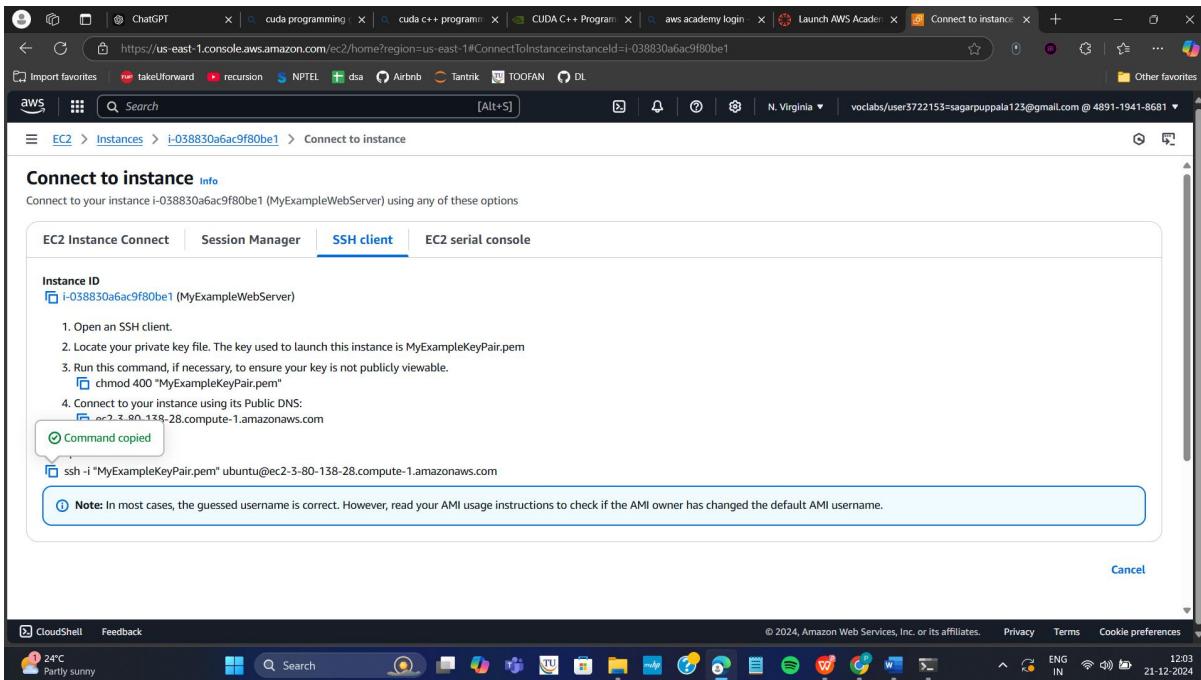
Copy the path of .pem file that you saved earlier



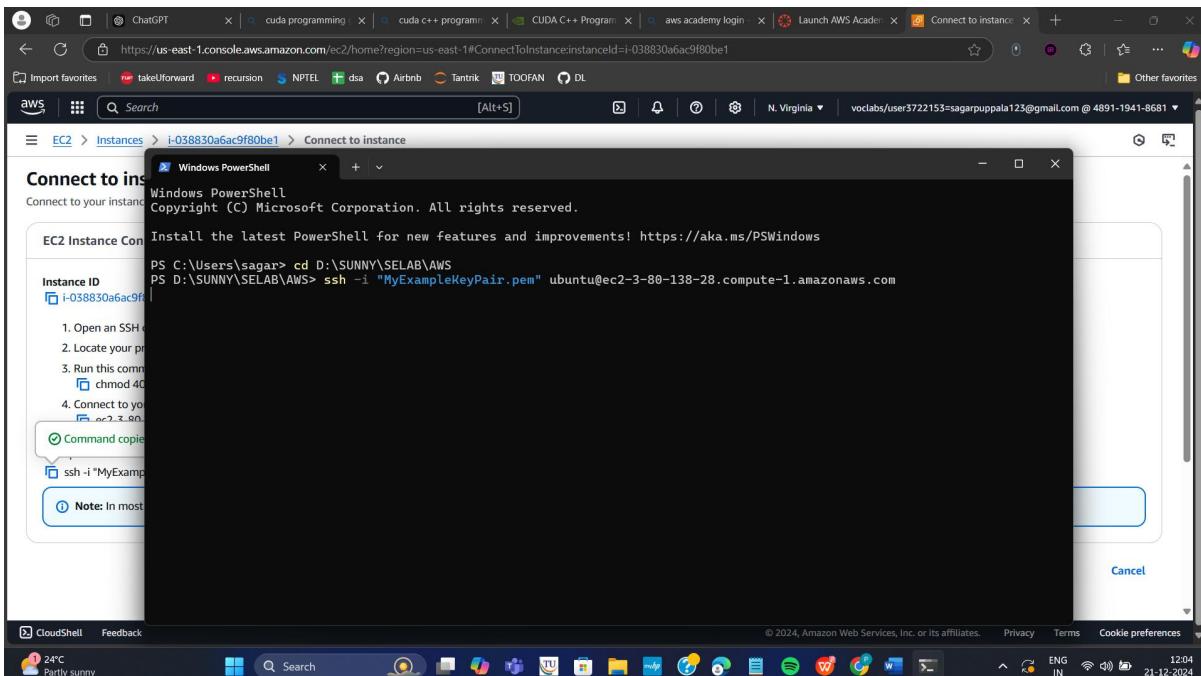
Open CMD and navigate to that path



Go to SSh and copy the command which is present at the below



Run the command in the terminal



Run the following commands

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.

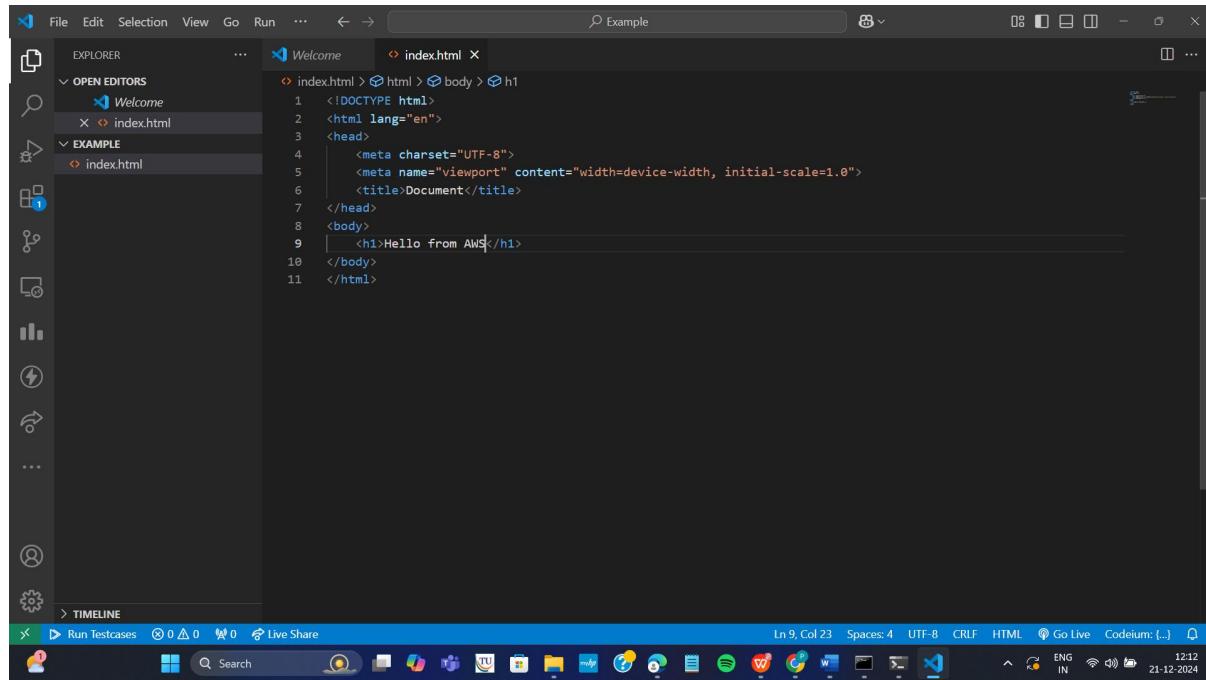
```
ubuntu@ip-172-31-93-141:~$ sudo apt update
```

```

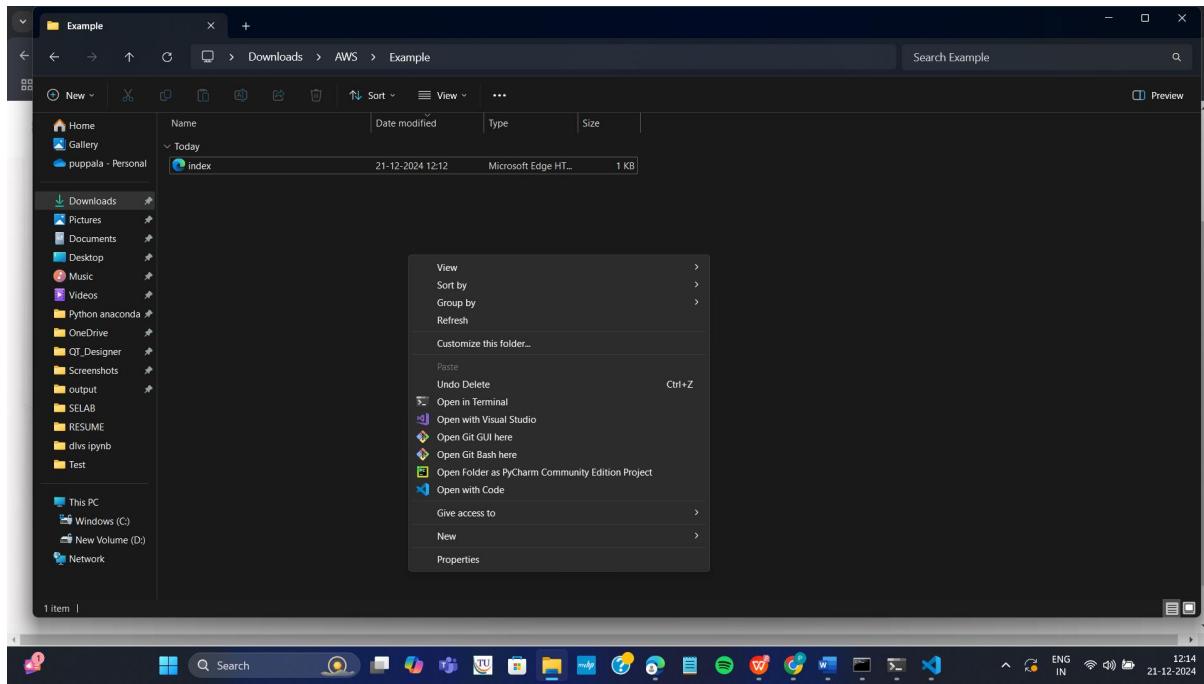
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-93-141:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd 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-
  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 58 not upgraded.
Need to get 0 B/29.2 MB of archives.
After this operation, 0 B of additional disk space will be used.
ubuntu@ip-172-31-93-141:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-93-141:~$ sudo apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
nano set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.

```

Create basic index.html file



Open git Bash



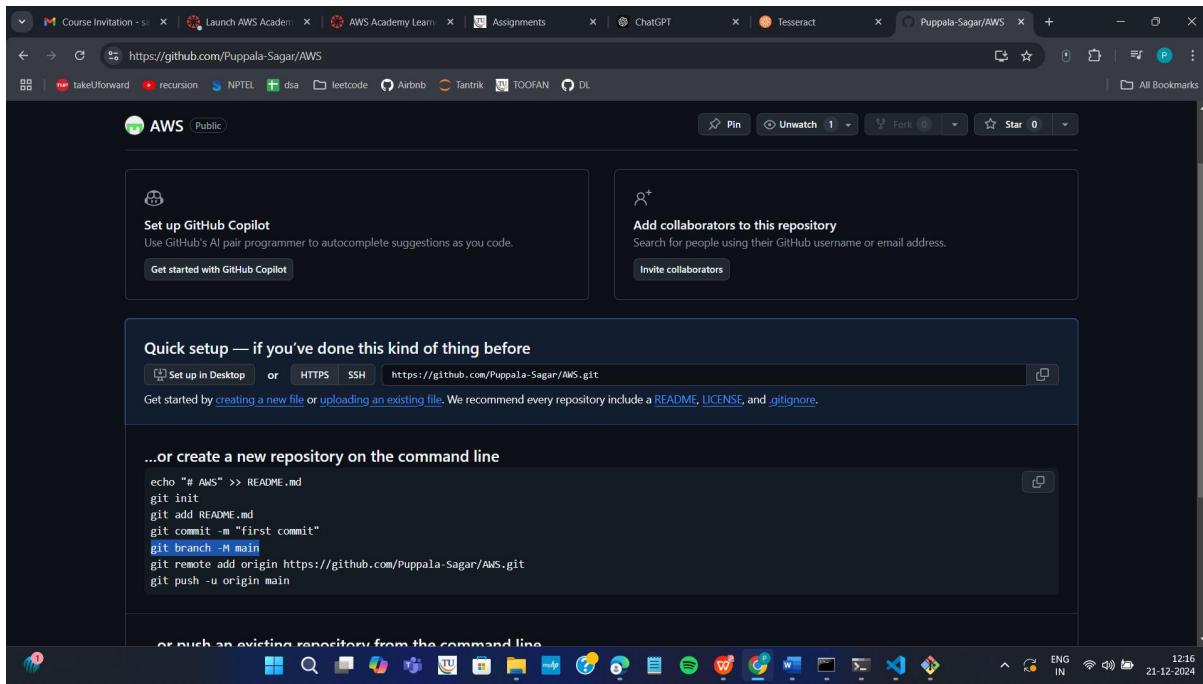
Run the following commands

```
MINGW64:/c/Users/sagar/Downloads/AWS/Example
sagar@SAGARPUPPALA MINGW64 ~/Downloads/AWS/Example (master)
$ git init
Initialized empty Git repository in C:/Users/sagar/Downloads/AWS/Example/.git/
sagar@SAGARPUPPALA MINGW64 ~/Downloads/AWS/Example (master)
$ git add .

sagar@SAGARPUPPALA MINGW64 ~/Downloads/AWS/Example (master)
$ git commit -m "first commit"
[master (root-commit) f064c10] first commit
 1 file changed, 11 insertions(+)
 create mode 100644 index.html

sagar@SAGARPUPPALA MINGW64 ~/Downloads/AWS/Example (master)
$ |
```

Create git repo



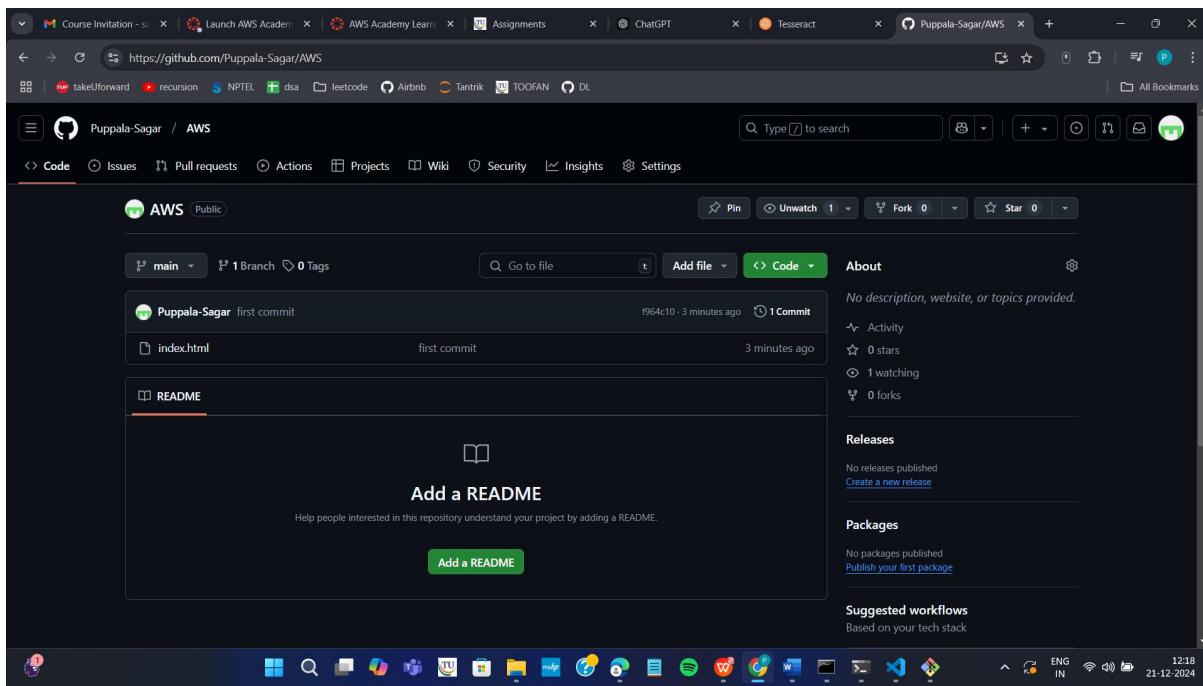
Now run the following commands

```
sagar@sagarpuppala MINGW64 ~/Downloads/AWS/Example (master)
$ git branch -M main

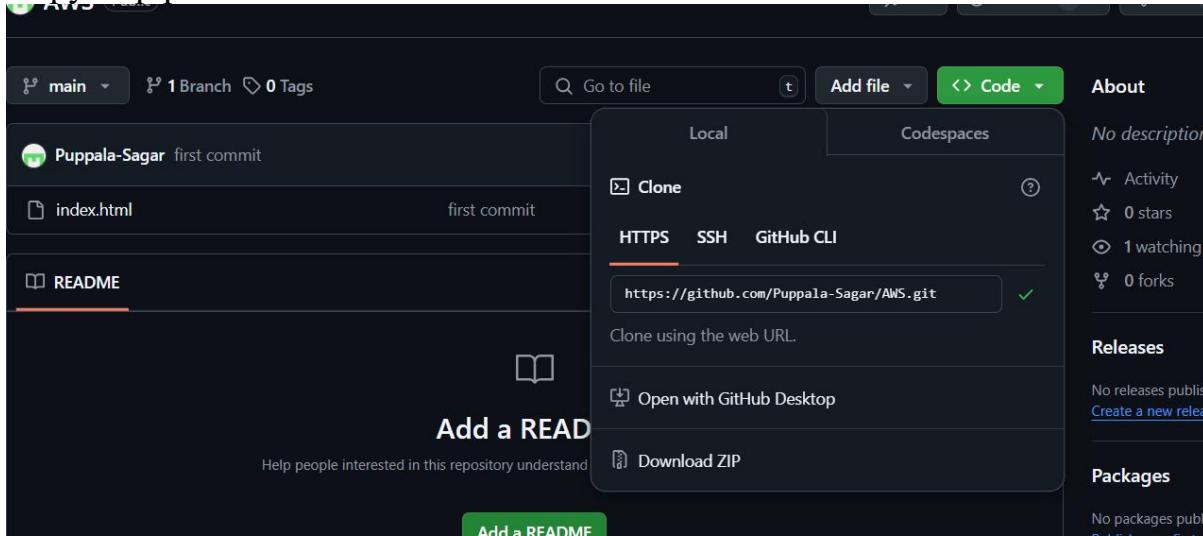
sagar@sagarpuppala MINGW64 ~/Downloads/AWS/Example (main)
$ git remote add origin https://github.com/Puppala-Sagar/AWS.git

sagar@sagarpuppala MINGW64 ~/Downloads/AWS/Example (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 380 bytes | 380.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Puppala-Sagar/AWS.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

You can now see index.html in github



Copy http path



Clone the repository

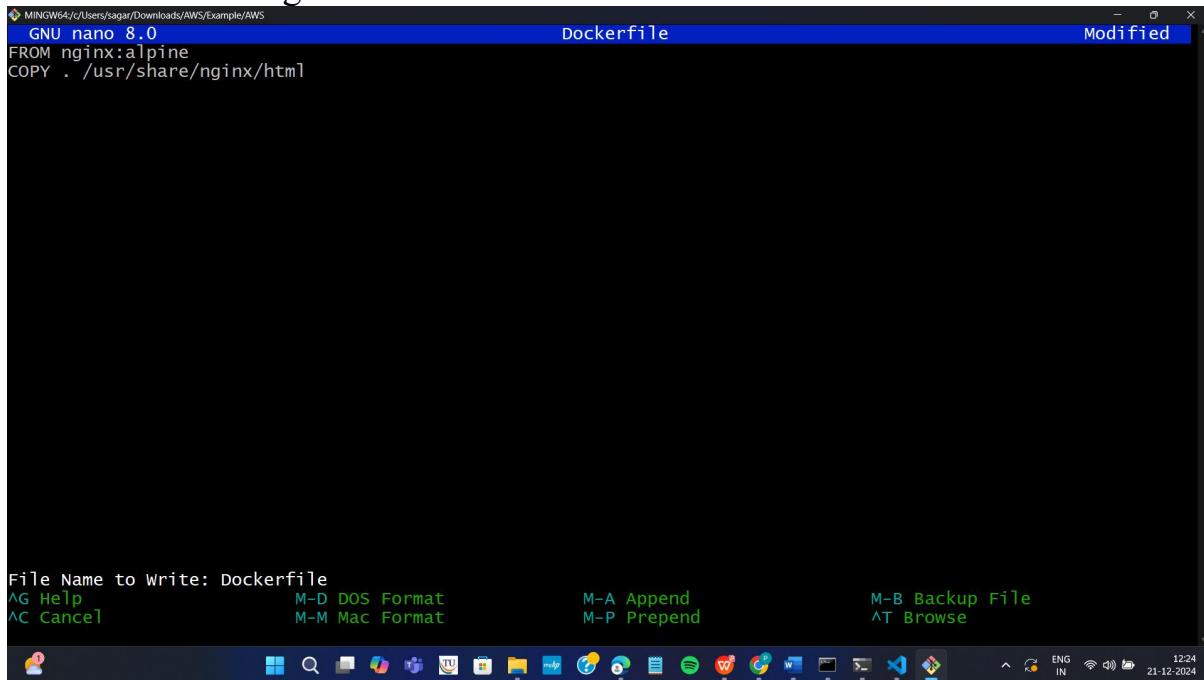
```
sagar@sagarpuppala MINGW64 ~/Downloads/AWS/Example (main)
$ git clone https://github.com/Puppala-Sagar/AWS.git
Cloning into 'AWS'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Navigate to the cloned folder

```
ubuntu@ip-172-31-93-141:~$ git clone https://github.com/Puppala-Sagar/AWS.git
Cloning into 'AWS'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-93-141:~$ ls
AWS
ubuntu@ip-172-31-93-141:~$ cd AWS
ubuntu@ip-172-31-93-141:~/AWS$ ls
index.html
ubuntu@ip-172-31-93-141:~/AWS$ ■

sagar@SAGARPUPPALA MINGW64 ~/Downloads/AWS/Example/AWS (main)
$ nano Dockerfile|
```

Write the following data in Dockerfile and click **ctrl+o** and **ctrl+x**



```
MINGW64/c/Users/sagar/Downloads/AWS/Example/AWS
  GNU nano 8.0
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

The screenshot shows a terminal window titled "Dockerfile" with the file content displayed. The content is a Dockerfile for an Nginx container, specifying the base image as "nginx:alpine" and copying the current directory's files to the "/usr/share/nginx/html" path. The terminal window has a dark theme and includes standard Linux-style navigation keys at the bottom.

Build docker image by executing the following command

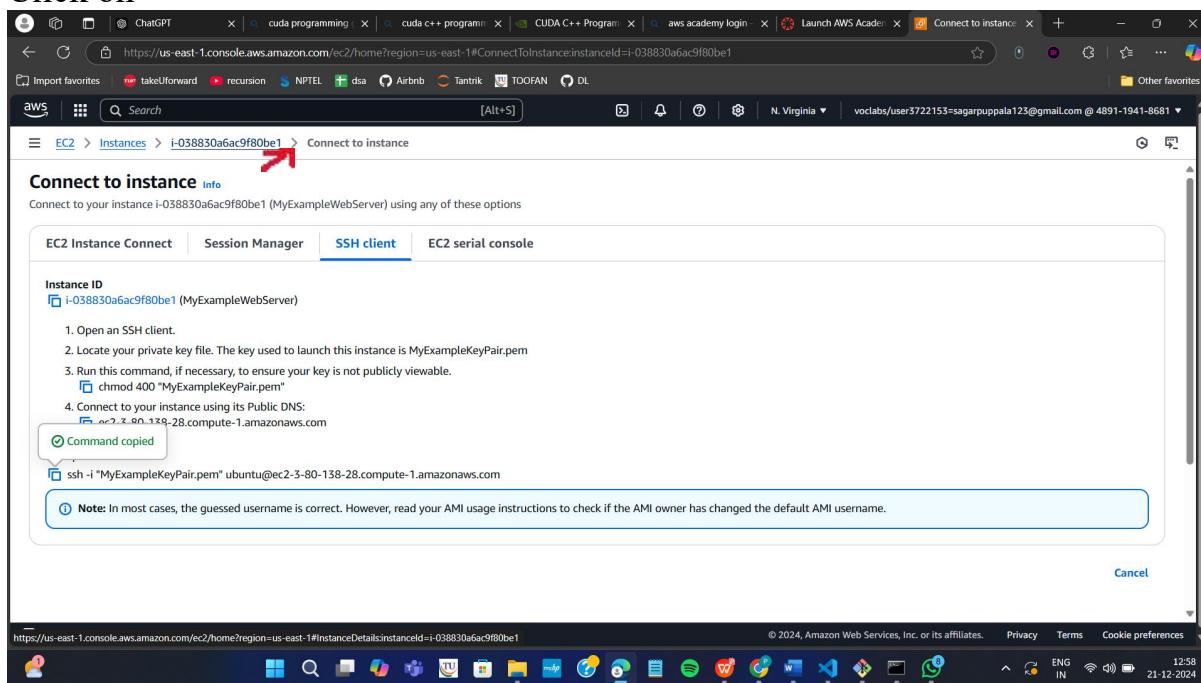
```
ubuntu@ip-172-31-93-141:~/AWS$ sudo docker build -t mywebapp .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx

Sending build context to Docker daemon 65.02kB
Step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
da9db072f522: Pull complete
e10e486de1ab: Pull complete
af9c0e53c5a4: Pull complete
b2eb2b8af93a: Pull complete
e351ee5ec3d4: Pull complete
fbbbf7d28be71: Pull complete
471412c08d15: Pull complete
a2eb5282fbec: Pull complete
Digest: sha256:41523187cf7d7a2f2677a80609d9caa14388bf5c1fbca9c410ba3de602aaaab4
Status: Downloaded newer image for nginx:alpine
--> 91ca84b4f577
Step 2/2 : COPY . /usr/share/nginx/html
--> a3737b372d23
Successfully built a3737b372d23
Successfully tagged mywebapp:latest
```

Run the container

```
ubuntu@ip-172-31-93-141:~/AWS$ sudo docker run -d -p 80:80 mywebapp
12429f3ea7b7b0750424000e6404b7c34ba8213745f54708e5f1e9e716177b58
```

Click on



Copu public ip

The screenshot shows the AWS EC2 Instances details page. The instance summary for i-038830a6ac9f80be1 is displayed. Key details include:

- Public IPv4 address:** 3.80.138.28
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-93-141.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0da3310938e1da95f
- Subnet ID:** subnet-03dfd34535cf23e91
- Instance ARN:** arn:aws:ec2:us-east-1:489119418681:instance/i-038830a6ac9f80be1

Paste it in the browser

The browser window displays the URL <http://3.80.138.28>. The page content is "Hello from AWS".

Hello from AWS

The terminal window shows the following commands and output:

```
ubuntu@ip-172-31-93-141:~/AWS$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
12429f3ea7b7        mywebapp           "/docker-entrypoint..."   3 minutes ago    Up 3 minutes     0.0.0.0:80->80/tcp, ::80->80/tcp   laughing_allen
12429f3ea7b7
ubuntu@ip-172-31-93-141:~/AWS$ sudo docker stop 12429f3ea7b7
12429f3ea7b7
ubuntu@ip-172-31-93-141:~/AWS$
```

Click on terminate instances

The screenshot shows the AWS EC2 Instances page with a single instance listed:

- Instances (1/1) Info**
- Name:** MyExampleWebServer
- Instance ID:** i-038830a6ac9f80be1
- Instance state:** Running
- Instance type:** t2.micro
- Actions:** Stop instance, Start instance, Reboot instance, Hibernate instance, Terminate (delete) instance

i-038830a6ac9f80be1 (MyExampleWebServer)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-038830a6ac9f80be1	3.80.138.28 [open address]	172.31.93.141
IPv6 address	-	Public IPv4 DNS ec2-3-80-138-28.compute-1.amazonaws.com [open address]
Instance state	Running	-

Terminate (delete) instance?

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

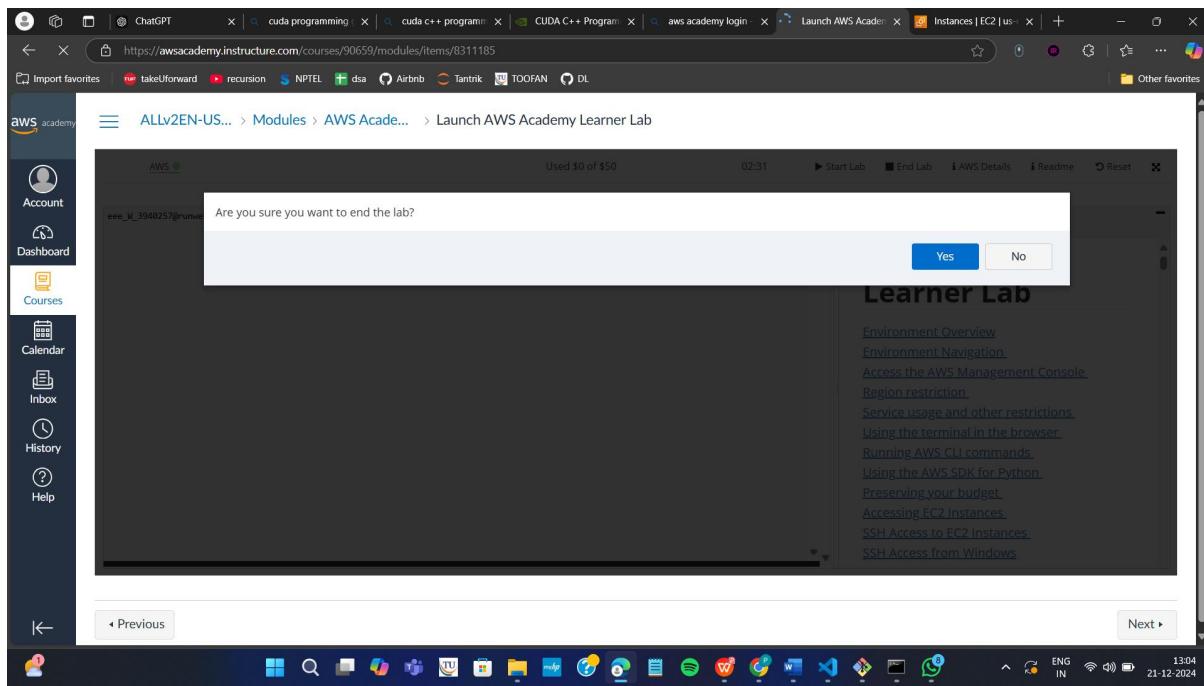
Are you sure you want to terminate these instances?

Instance ID	Termination protection
i-038830a6ac9f80be1 (MyExampleWebServer)	Disabled

To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

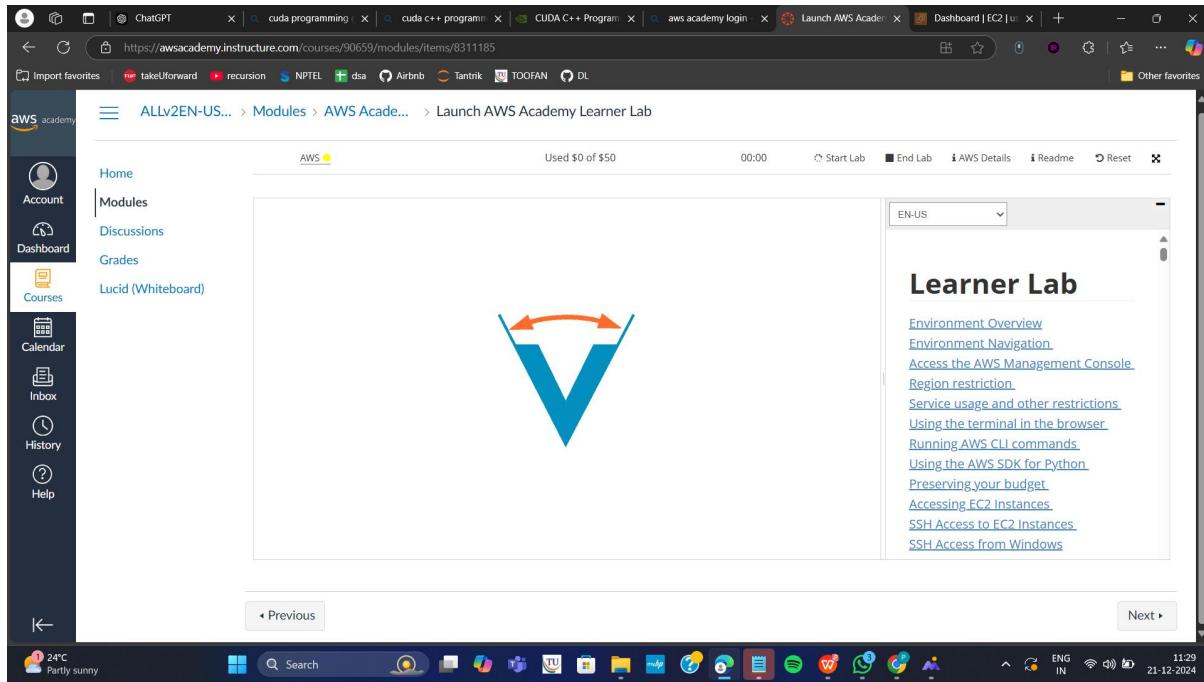
Cancel **Terminate (delete)**

Now click on end lab



2 MAVEN WEB PROJECT DEPLOYMENT IN AWS

Click on start lab



Click on AWS

AWS

Used \$0 of \$50

04:00

Start Lab End Lab AWS Details Readme Reset

eee_W_3940257@runweb155453:~\$

EN-US

Learner Lab

- [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](#)
- [SSH Access from Windows](#)

◀ Previous Next ▶

Click on EC2

Console Home

Recently visited: EC2

Applications (0) info

No applications. Get started by creating an application.

Create application

Welcome to AWS

AWS Health

Cost and usage

Click on launch instance

Dashboard

Instances

Images

Elastic Block Store

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance Migrate a server

Note: Your instances will launch in the US East (N. Virginia) Region

Service health

AWS Health Dashboard

Region: US East (N. Virginia)

Status: This service is operating normally.

Zones

Zone name Zone ID

Account attributes

Default VPC vpc-0da310938e1da95f

Settings

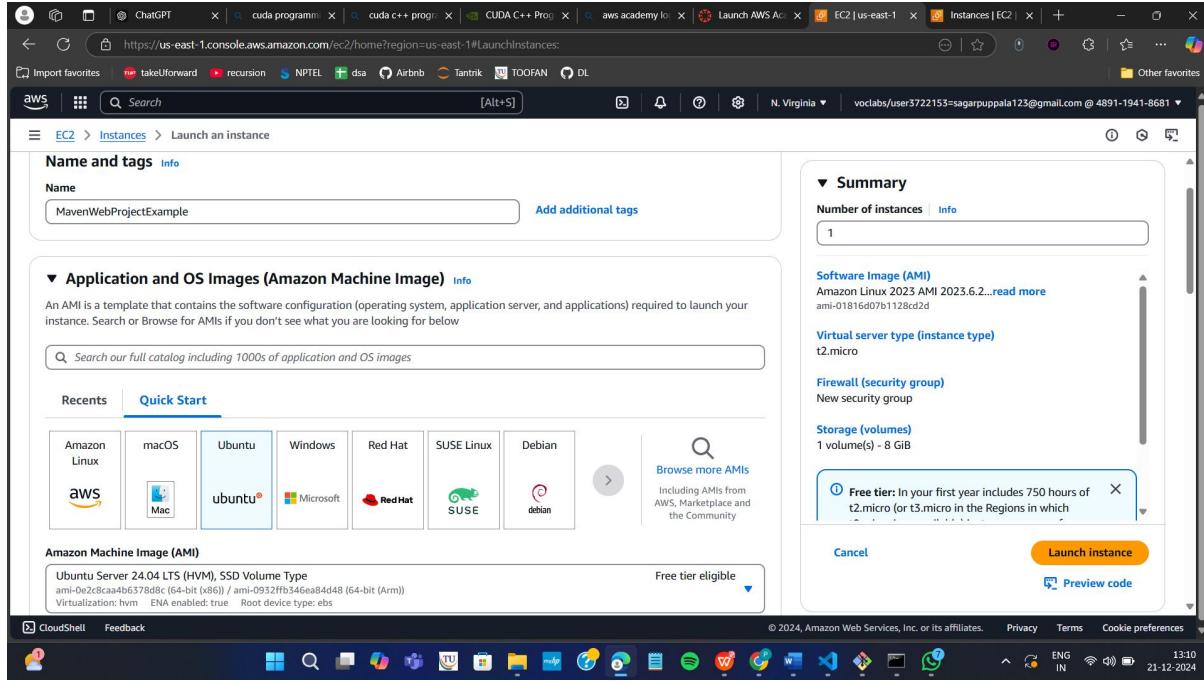
Data protection and security Allowed AMIs Zones EC2 Serial Console Default credit specification EC2 console preferences

Explore AWS

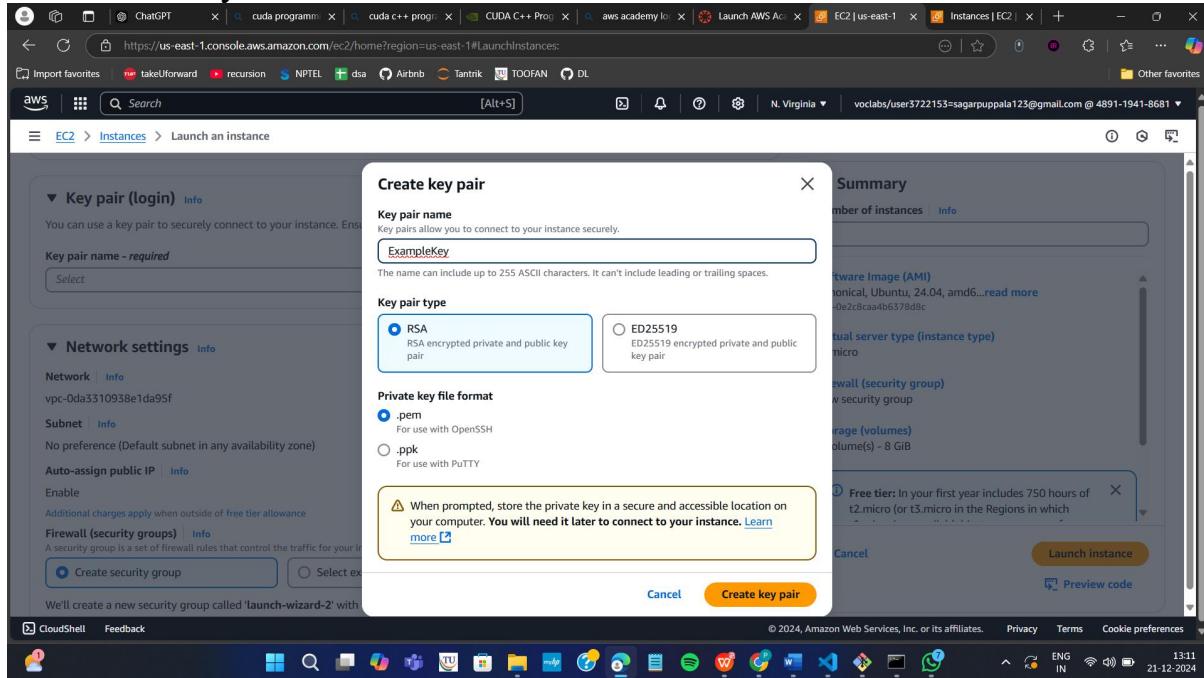
Get Up to 40% Better Price Performance T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

10 Things You Can Do Today to Reduce AWS Costs Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more

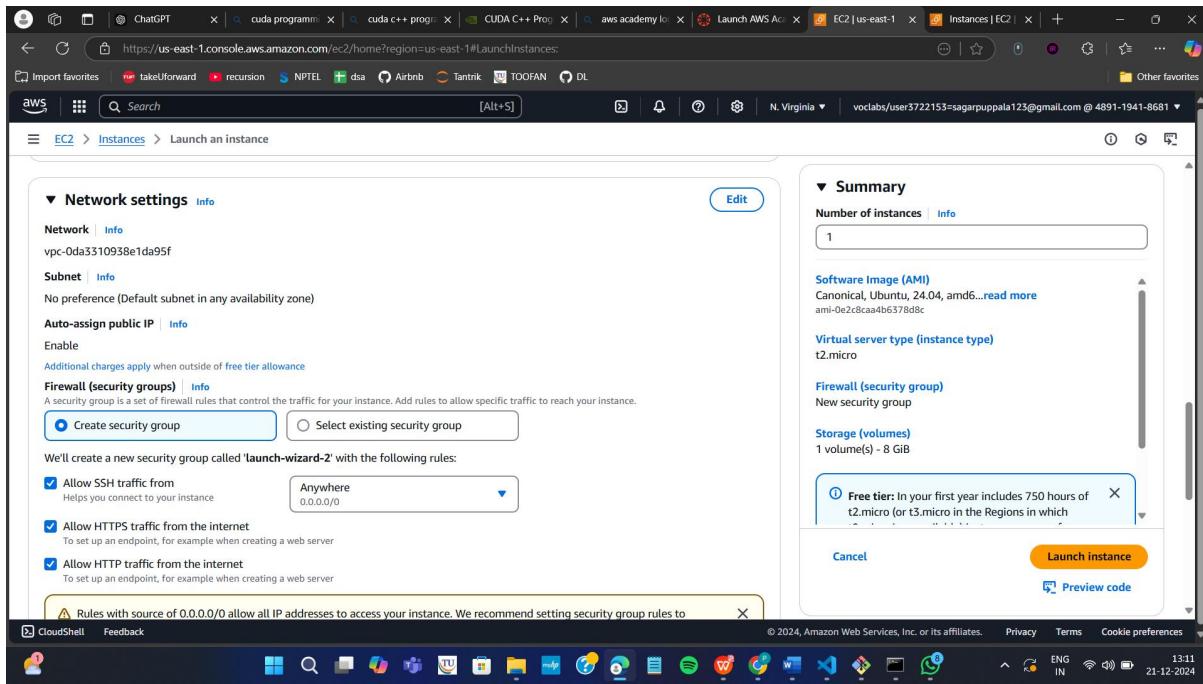
Give name



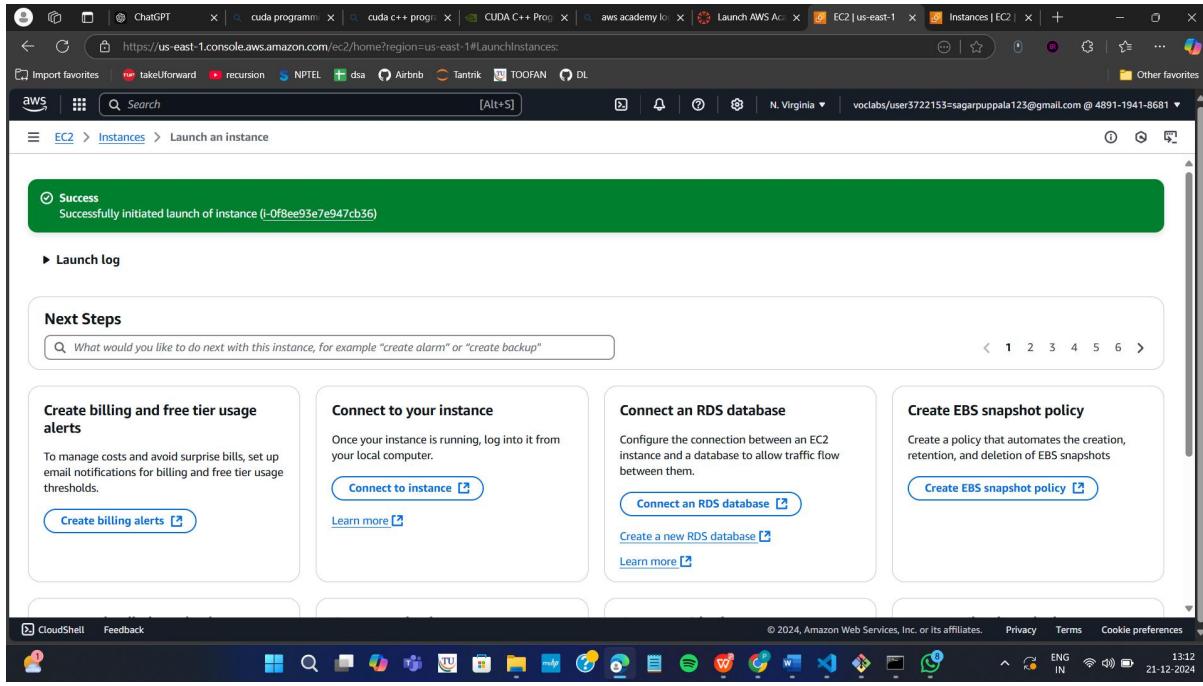
Create new key



Check all the boxes under network and click on launch instance



Click on instances



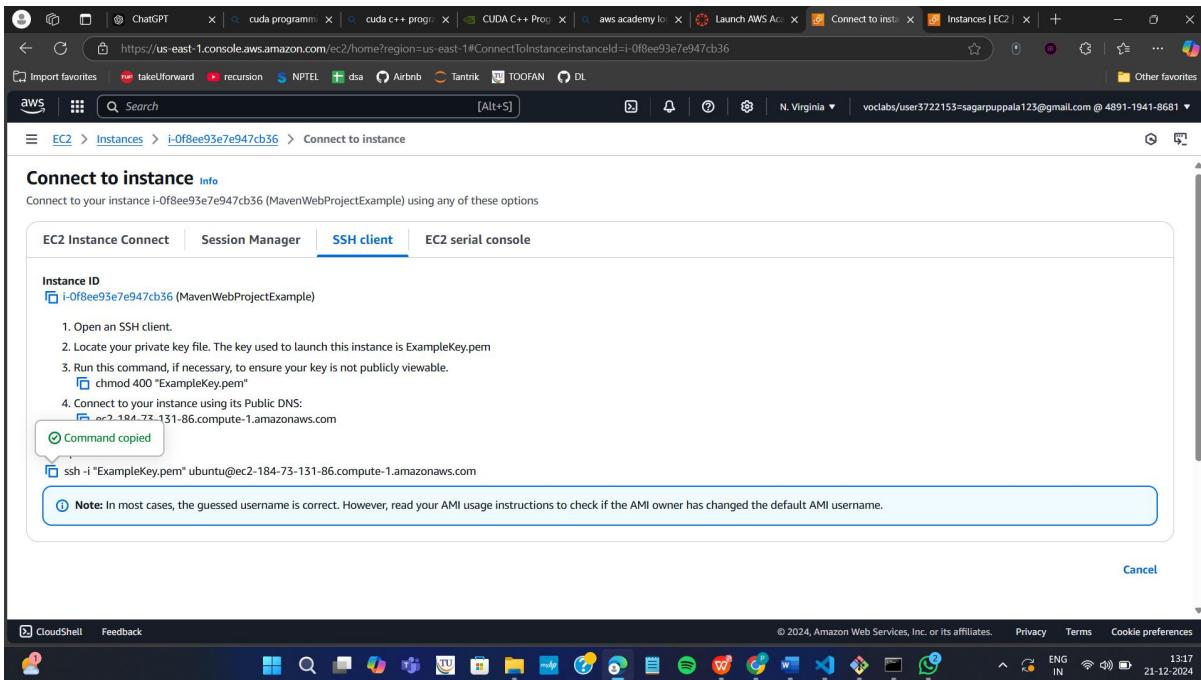
Wait until you get 2 tests passes

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main area displays 'Instances (2) Info' with a table. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. The first row, 'MyExampleWeb...', has a terminated state. The second row, 'MavenWebPro...', is running. A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and an 'Actions' dropdown is visible.

Click on connect after checking the box

This screenshot is similar to the previous one but focuses on the 'MavenWebPro...' instance. The 'Connect' button is highlighted in blue. The instance details page for 'i-0f8ee93e7e947cb36 (MavenWebProjectExample)' is shown. It includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under the Details tab, there's an 'Instance summary' section with fields for Instance ID, IPv6 address, Public IPv4 address (184.73.131.86), Instance state (Running), Private IPv4 addresses (172.31.22.244), and Public IPv4 DNS (ec2-184-73-131-86.compute-1.amazonaws.com). A note at the bottom says 'Copy ssh command'.

Copy ssh command



Open the terminal, Navigate to the path

```
C:\Users\sagar>cd C:\Users\sagar\Downloads\AWS
```

Run the ssh command

```
C:\Users\sagar\Downloads\AWS>ssh -i "ExampleKey.pem" ubuntu@ec2-184-73-131-86.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Sat Dec 21 07:48:56 UTC 2024

 System load: 0.0          Processes:           104
 Usage of /: 24.7% of 6.71GB   Users logged in:     0
 Memory usage: 21%           IPv4 address for enX0: 172.31.22.244
 Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

Run the following commands

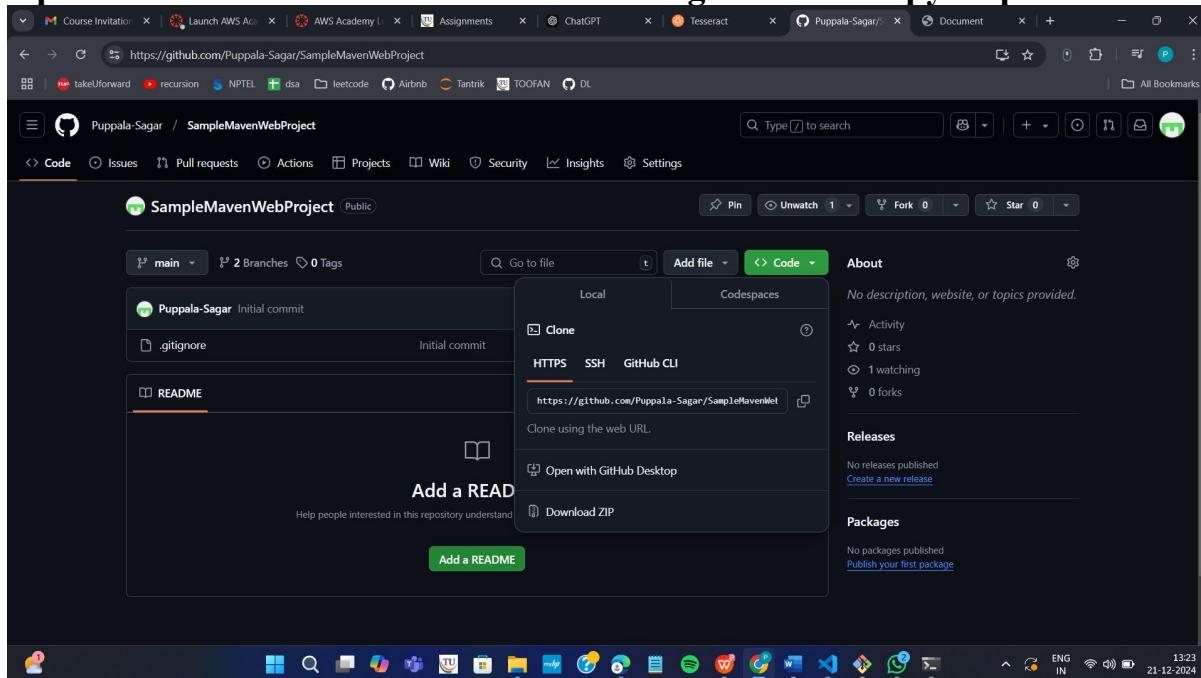
```
ubuntu@ip-172-31-22-244:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [159 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [572 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [59 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [111 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [111 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [111 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [111 kB]
```

```

ubuntu@ip-172-31-22-244:~$ 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
  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 58 not upgraded.
ubuntu@ip-172-31-22-244:~$ sudo apt install git
Command 'sudo' not found, did you mean:
  command 'ssdp' from snap ssdp (0.0.1)
  command 'sfdf' from deb graphviz (2.42.2-9ubuntu0.1)
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
  command 'sup' from deb sup (20100519-3)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-22-244:~$ sudo apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nano is already the newest version (7.2-2ubuntu0.1).
nano set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
ubuntu@ip-172-31-22-244:~$ 

```

Open MAVEN WEB PROJECT REPO in github and copy http link



Clone the repository

```
ubuntu@ip-172-31-22-244:~$ git clone https://github.com/Puppala-Sagar/SampleMavenWebProject.git
Cloning into 'SampleMavenWebProject'...
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 39 (delta 3), reused 23 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (39/39), 7.82 KiB | 2.60 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

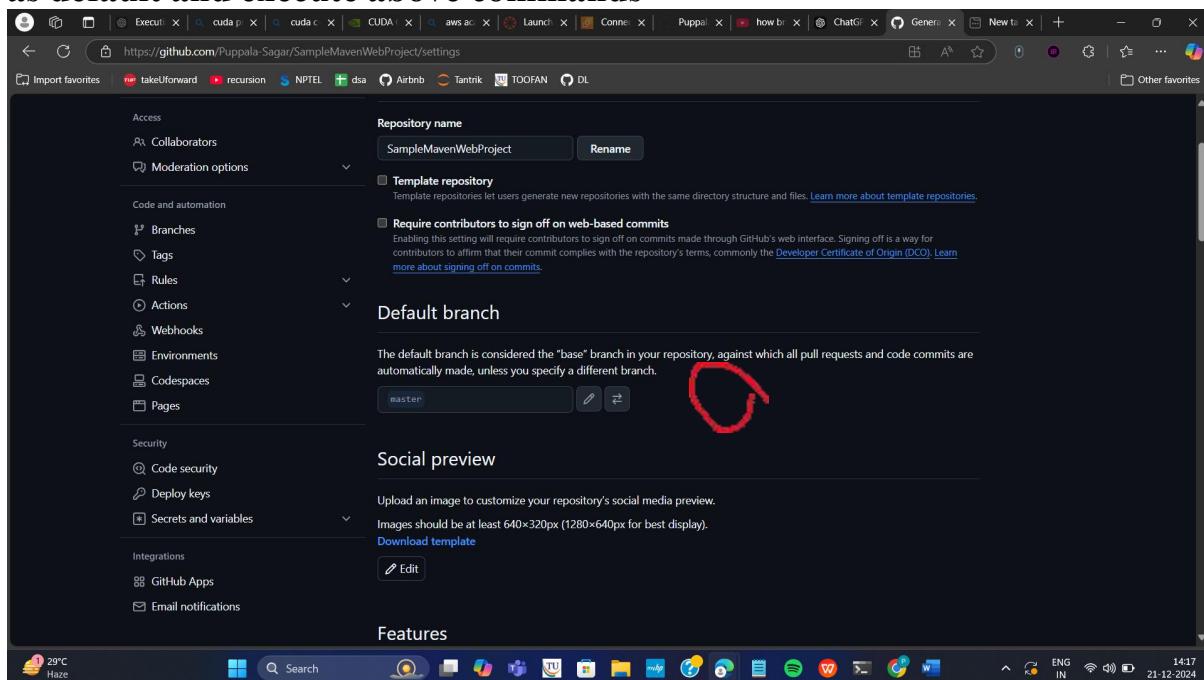
If your repository is in main run following

```
ubuntu@ip-172-31-22-244:~$ ls
SampleMavenWebProject
ubuntu@ip-172-31-22-244:~/SampleMavenWebProject$ cd SampleMavenWebProject/
ubuntu@ip-172-31-22-244:~/SampleMavenWebProject$ ls
```

If not goto your github repo

Click on settings

Under default branch section click on the icon shown and you're your master branch as default and execute above commands



```
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ nano Dockerfile
```

```
GNU nano 7.2                               Dockerfile *
FROM tomcat:9-jdk11
COPY target/*.war /usr/local/tomcat/webapps
```

Build docker image

```
Build an image from a Dockerfile
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ sudo docker build -t mavenwebproject .
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 101.4kB
Step 1/2 : FROM tomcat:9-jdk11
9-jdk11: Pulling from library/tomcat
de44b265507a: Extracting [=====] 15.73MB/29.75MB
4d0025a6d227: Download complete
5a7ece70ec66: Downloading [=====] 111MB/145.6MB
623a4ff914ca: Download complete
6d3bf2e30222: Download complete
9ab2f23fa0e1: Download complete
4f4fb700ef54: Download complete
cdbea10bf012: Download complete
```

Run the container

```
Successfully tagged mavenwebproject:latest
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ sudo docker run -d -p 9090:8080 mavenwebproject
e83304610c89a12dd5eab6379ea697e0f445baaf8f45d85edb009d356beda8e7
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ |
```

Copy public ipv4

The screenshot shows the AWS EC2 Instances page. The instance summary for i-090ce930c7cc2155d is displayed. The Public IPv4 address is highlighted as 54.87.134.240. Other details shown include Instance ID (i-090ce930c7cc2155d), IPv6 address (none), Hostname type (IP name: ip-172-31-92-238.ec2.internal), Answer private resource DNS name (IPv4 (A)), Auto-assigned IP address (54.87.134.240 [Public IP]), IAM Role (none), IMDSv2 (Required), Instance state (Running), Private IP DNS name (ip-172-31-92-238.ec2.internal), Instance type (t2.micro), VPC ID (vpc-0da3310938e1da95f), Subnet ID (subnet-03dfd34535cf23e91), Instance ARN (arn:aws:ec2:us-east-1:489119418681:instance/i-090ce930c7cc2155d), Private IPv4 addresses (172.31.92.238), Public IPv4 DNS (ec2-54-87-134-240.compute-1.amazonaws.com), and Elastic IP addresses (none). The AWS Compute Optimizer finding section is also visible.

Enter this url in browser and click enter

The screenshot shows a browser window with two tabs. The first tab is GitHub, and the second tab is Google Account. Below the tabs, the URL 54.87.134.240:9090/SampleMavenWebProject is visible in the address bar of the second tab. The browser interface includes a search bar, a toolbar with various icons, and a status bar at the bottom.

If your app is not running goto security and Click on security groups

The screenshot shows the AWS EC2 Instances page with the instance summary for i-090ce930c7cc2155d. The Security tab is selected. The Security groups section shows sg-030d1847efe1b36cd (launch-wizard-3). Other tabs visible include Details, Status and alarms, Monitoring, Networking, Storage, and Tags. The Inbound rules section is partially visible at the bottom.

Click on edit inbound rules

> sg-030d1847efe1b36cd - launch-wizard-3

Details	
Security group name launch-wizard-3	Security group ID sg-030d1847efe1b36cd
Description launch-wizard-3 created 2024-12-21T08:31:06.033Z	VPC ID vpc-0da3310938e1da95f
Owner 489119418681	Inbound rules count 3 Permission entries
	Outbound rules count 1 Permission entry

[Inbound rules](#) [Outbound rules](#) [Sharing - new](#) [VPC associations - new](#) [Tags](#)

Inbound rules (3)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0a364df2380486ba1	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-07d05f05d7474f15d	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0d0fb3233a424968	IPv4	HTTP	TCP	80

Click on add rule give port as 9090 and 0.0.0.0/0 click on save

EC2 > Security Groups > sg-030d1847efe1b36cd > Edit inbound rules

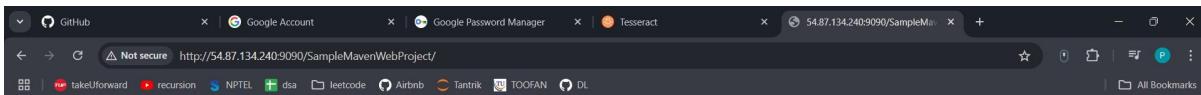
Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0e3eea5774341ae8	Custom TCP	TCP	9090	Cu... ▾	<input type="text"/> 0.0.0.0/0 Delete
sgr-0a364df2380486ba1	HTTPS	TCP	443	Cu... ▾	<input type="text"/> 0.0.0.0/0 Delete
sgr-07d05f05d7474f15d	SSH	TCP	22	Cu... ▾	<input type="text"/> 0.0.0.0/0 Delete
sgr-0d0fb3233a424968	HTTP	TCP	80	Cu... ▾	<input type="text"/> 0.0.0.0/0 Delete

[Add rule](#)

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

[Cancel](#) [Preview changes](#) [Save rules](#)

And refresh the page to check whether the web page has loaded or not



Hello World!



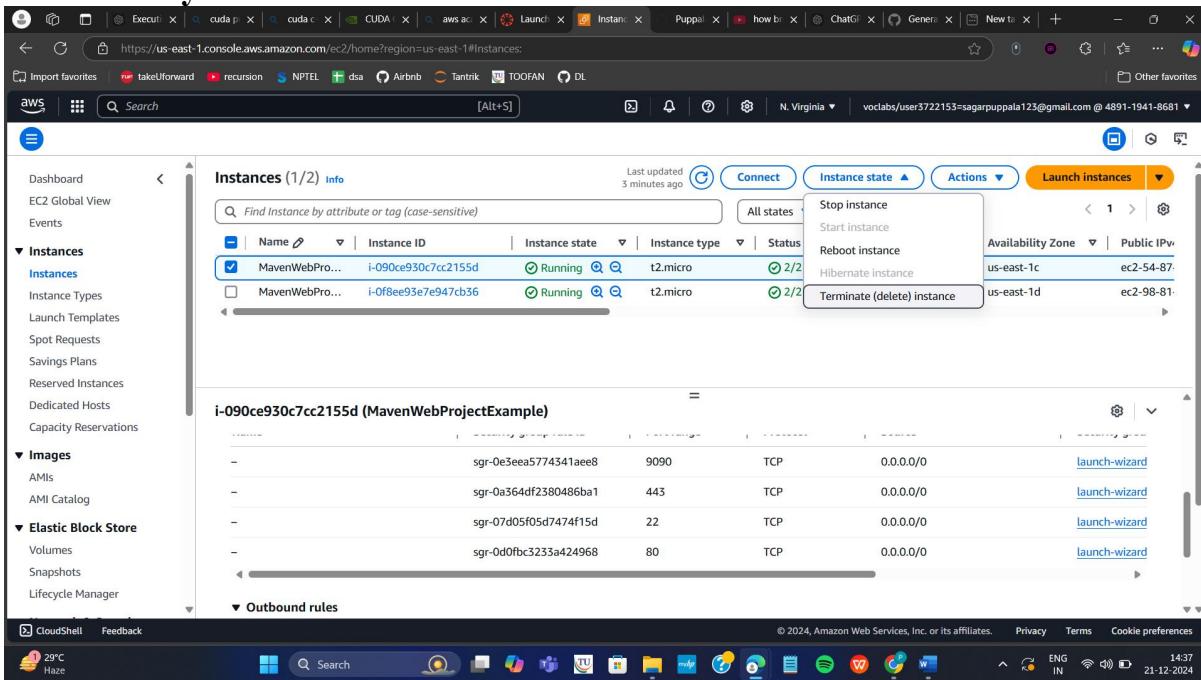
If it is loaded you have successfully deployed your maven web project In your ec2 instance

Run the following commands to stop the

```
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e83304610c89 mavenwebproject "catalina.sh run" 9 minutes ago Up 9 minutes 0.0.0.0:9090->8080/tcp, :::9090->8080/tcp keen_elbakyan
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ sudo docker stop e83304610c89
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$ |
```

container

Terminate your instance



End your lab

The screenshot shows a web browser window for the AWS Academy Learner Lab. The URL is <https://awsacademy.instructure.com/courses/90659/modules/items/8311185>. The page title is "Launch AWS Academy Learner Lab". The left sidebar has sections for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main content area has tabs for Home, Modules, Discussions, Grades, and Lucid (Whiteboard). The Home tab is selected. It shows a terminal window with the command "eee_l_3940257@runweb155440:~\$". To the right of the terminal is a "Learner Lab" section with a "EN-US" dropdown menu. Below the dropdown is a list of 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 of the main content area are "Previous" and "Next" buttons. The status bar at the bottom shows system information: 29°C Haze, Search, Taskbar icons, ENG IN, 14:38, 21-12-2024.