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

Certificate

This is to certify that following is a Bonafide Record of the workbook task done by _____bearing Roll No_____ of _____ Branch of _____ year B.Tech Course in the_____ Subject during the Academic year _____ & _____ under our supervision.

Number of week tasks completed: _____

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

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

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. MODIFYAND PUSH DOCKER IMAGE D. CREATE AND PUSH DOCKER FILE 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	

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

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

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.

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.

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.

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: puppala-sagar <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:

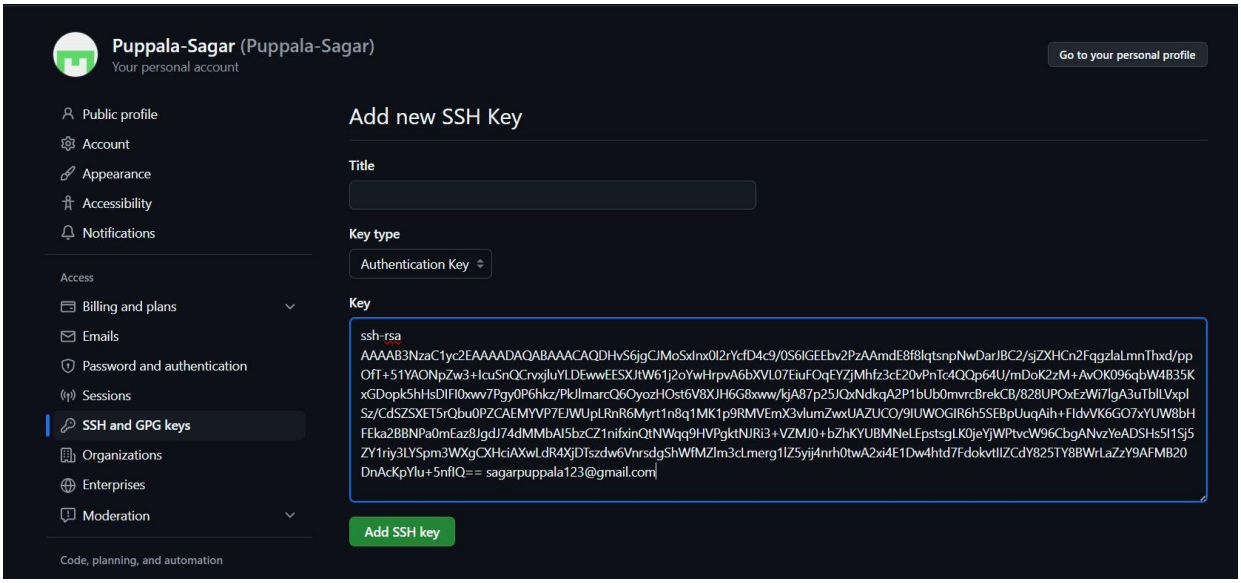
```
cat ~/.ssh/id_rsa.pub
```

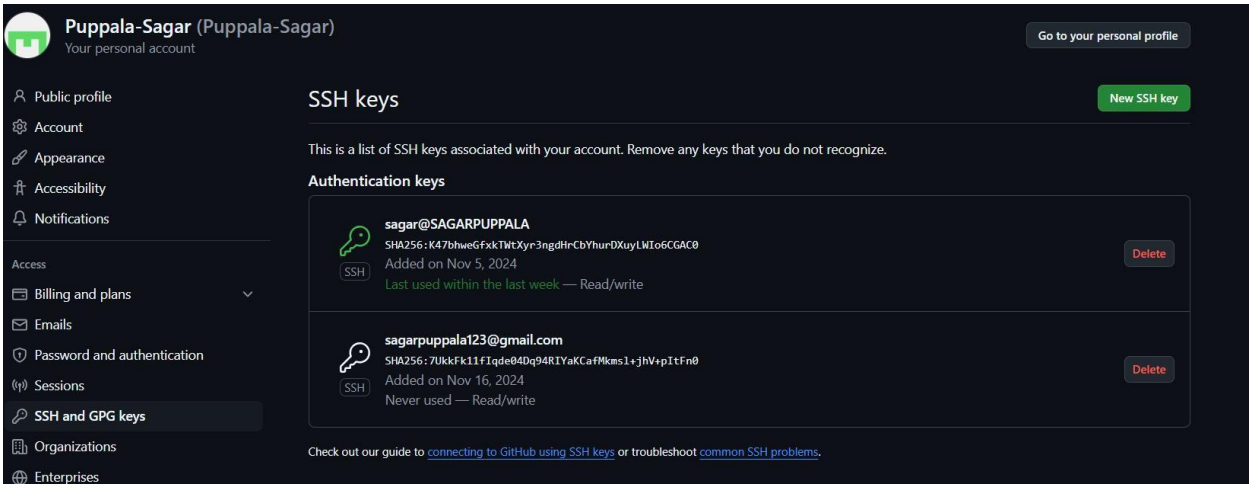
```
sagar@SAGARPUPPALA MINGW64 /d/SUNNY/SELAB (master)
$ cat ./sagar.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDHvS6jgCJMoSxInx0I2rYcfD4c9/0S6IGEEbv2PzAAmdE8f8lqtsnpNwDarJ
BC2/sjZXHCn2FqgzlaLmnThxd/ppOfT+51YAONpZw3+IcuSnQCrvxjluYLDEwwEESXJtW61j2oYwHrpvA6bXVL07EiuFOqEYZj
Mhfz3cE20vPnTc4QQp64U/mDoK2zM+AvOK096qbW4B35KxGDopk5hHsDIFI0xwv7Pgy0P6hkz/PkJlmarcQ6OyozHOst6V8XJH
6G8xww/kjA87p25JQxNdkqA2P1bUb0mvrcBrekCB/828UPOxEzWi7lgA3uTblLVxplSz/CdSZSXET5rQbu0PZCAEMYVP7EJWUp
LRnR6Myrt1n8q1MK1p9RMVEmX3vlumZwxUAZUCO/9IUWOGIR6h5SEBpUuqAih+FIdvVK6GO7xYUW8bHFEka2BBNPa0mEaz8Jgd
J74dMMbAI5bzCZ1nifxinQtNWqq9HVPgktNJRi3+VZMJ0+bZhKYUBMNeLEpstsgLK0jeYjWPtvcW96CbgANvzYeADSHs5I1Sj5
ZY1riy3LYSpm3WXgCXHciAXwLdR4XjDTszdw6VnrsdgShWfMZlm3cLmerg1lZ5yij4nrh0twA2xi4E1Dw4htd7FdokvtIIZCdY
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**.





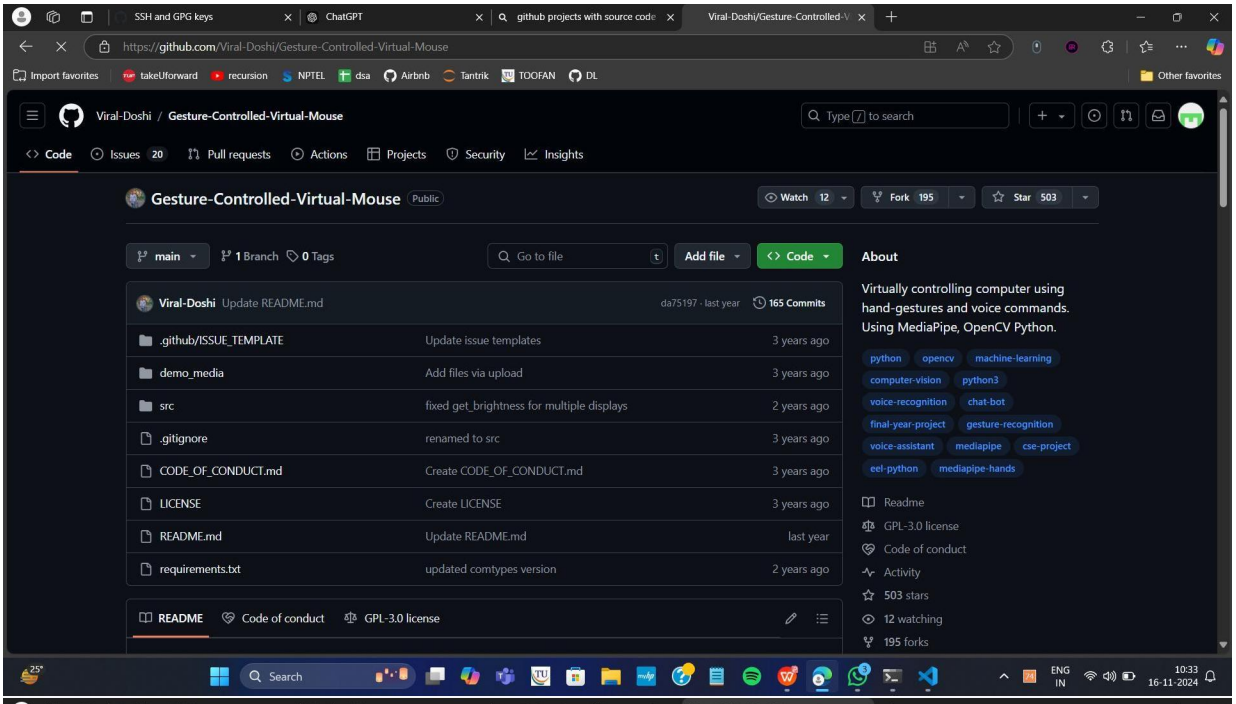
Fork :

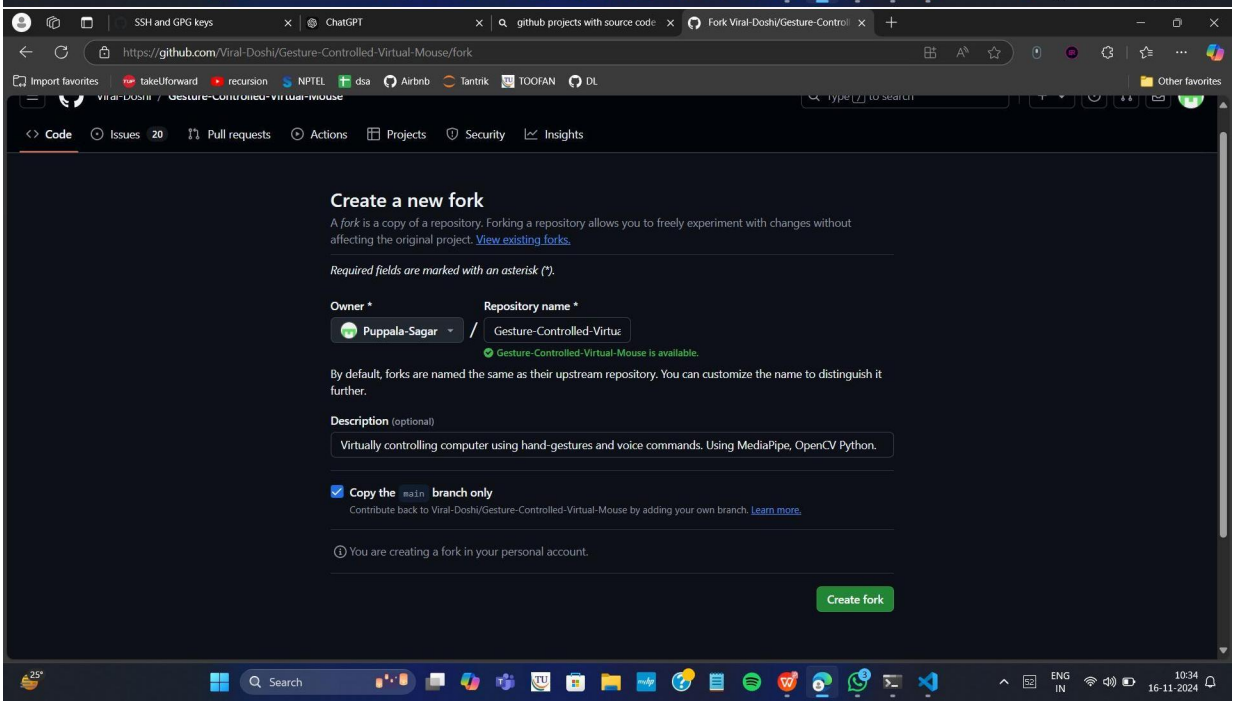
In Git, a **"fork"** refers to a copy of a repository from one user's account to another user's account on a platform like GitHub. Forking is commonly used to contribute to open-source projects or collaborate on projects where you don't have direct write access to the original repository. Forking allows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

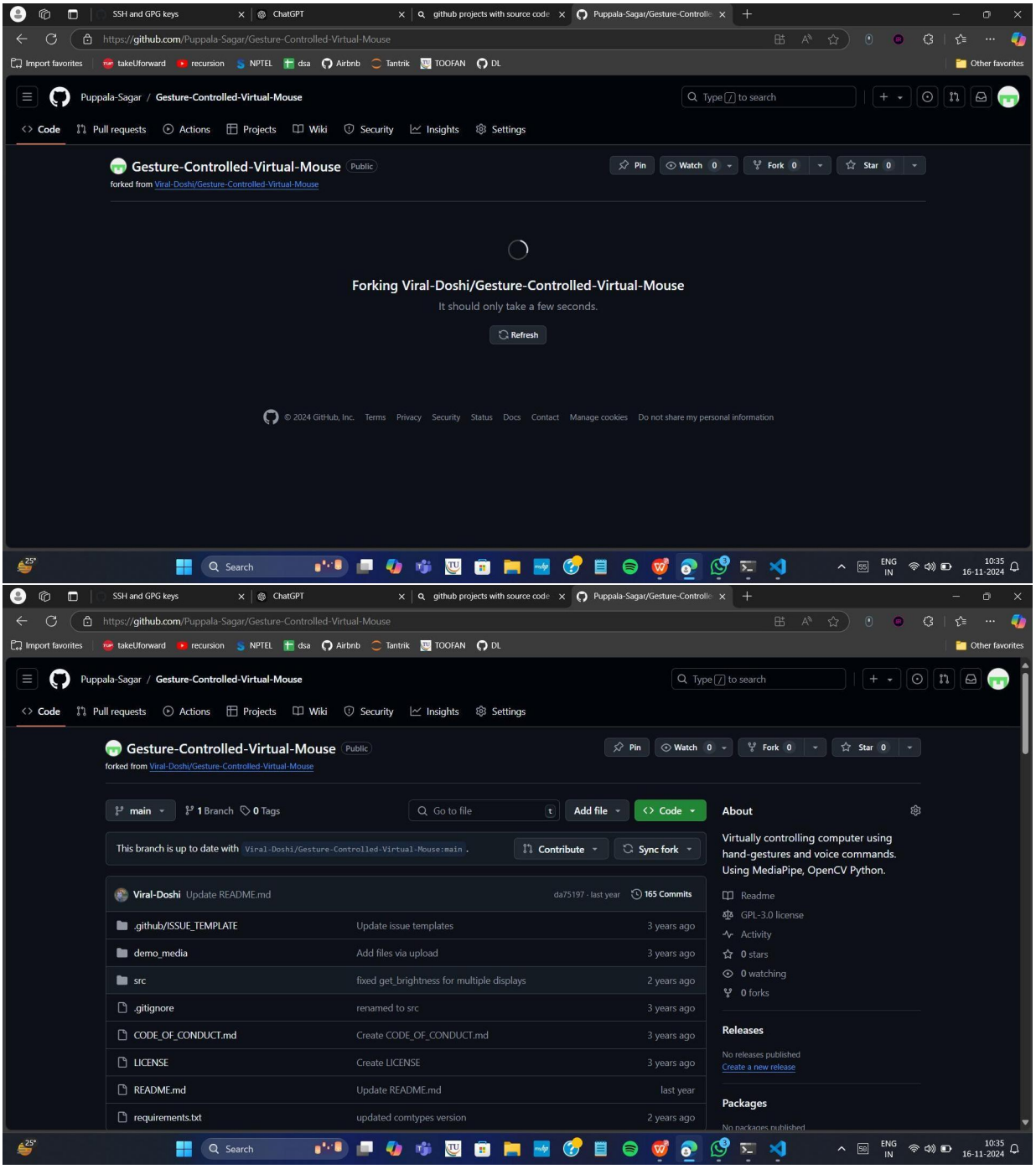
1. **Fork a Repository:**

- 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.


SSH and GPG keys
ChatGPT
github projects with source code
Viral-Doshi/Gesture-Controlled-V
https://github.com/Viral-Doshi/Gesture-Controlled-Virtual-Mouse
Import favorites
takeUforward
recursion
NPTEL
dsa
Airbnb
Tantrik
TOOFAN
DL
Other favorites
Viral-Doshi / Gesture-Controlled-Virtual-Mouse
Type / to search
Code
Issues 20
Pull requests
Actions
Projects
Security
Insights
Gesture-Controlled-Virtual-Mouse
Public
Watch 12
Fork 195
Star 503
main
1 Branch
0 Tags
Go to file
t
Add file
Code
Viral-Doshi Update README.md
da75197 · last year
165 Commits
.github/ISSUE_TEMPLATE
Update issue templates
3 years ago
demo_media
Add files via upload
3 years ago
src
fixed get_brightness for multiple displays
2 years ago
.gitignore
renamed to src
3 years ago
CODE_OF_CONDUCT.md
Create CODE_OF_CONDUCT.md
3 years ago
LICENSE
Create LICENSE
3 years ago
README.md
Update README.md
last year
requirements.txt
updated comtypes version
2 years ago
README
Code of conduct
GPL-3.0 license
About
Virtually controlling computer using hand-gestures and voice commands. Using MediaPipe, OpenCV Python.
python
opencv
machine-learning
computer-vision
python3
voice-recognition
chat-bot
final-year-project
gesture-recognition
voice-assistant
mediapipe
cse-project
eel-python
mediapipe-hands
Readme
GPL-3.0 license
Code of conduct
Activity
503 stars
12 watching
195 forks
Search
ENG
IN
10:33
16-11-2024


SSH and GPG keys
ChatGPT
github projects with source code
Fork Viral-Doshi/Gesture-Control
https://github.com/Viral-Doshi/Gesture-Controlled-Virtual-Mouse/fork
Import favorites
takeUforward
recursion
NPTEL
dsa
Airbnb
Tantrik
TOOFAN
DL
Other favorites
Code
Issues 20
Pull requests
Actions
Projects
Security
Insights
Create a new fork
A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.
Required fields are marked with an asterisk (*).
Owner *
Puppala-Sagar
/
Repository name *
Gesture-Controlled-Virtua
Gesture-Controlled-Virtual-Mouse is available.
By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.
Description (optional)
Virtually controlling computer using hand-gestures and voice commands. Using MediaPipe, OpenCV Python.
Copy the main branch only
Contribute back to Viral-Doshi/Gesture-Controlled-Virtual-Mouse by adding your own branch. Learn more.
You are creating a fork in your personal account.
Create fork
Search
ENG
IN
10:34
16-11-2024



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

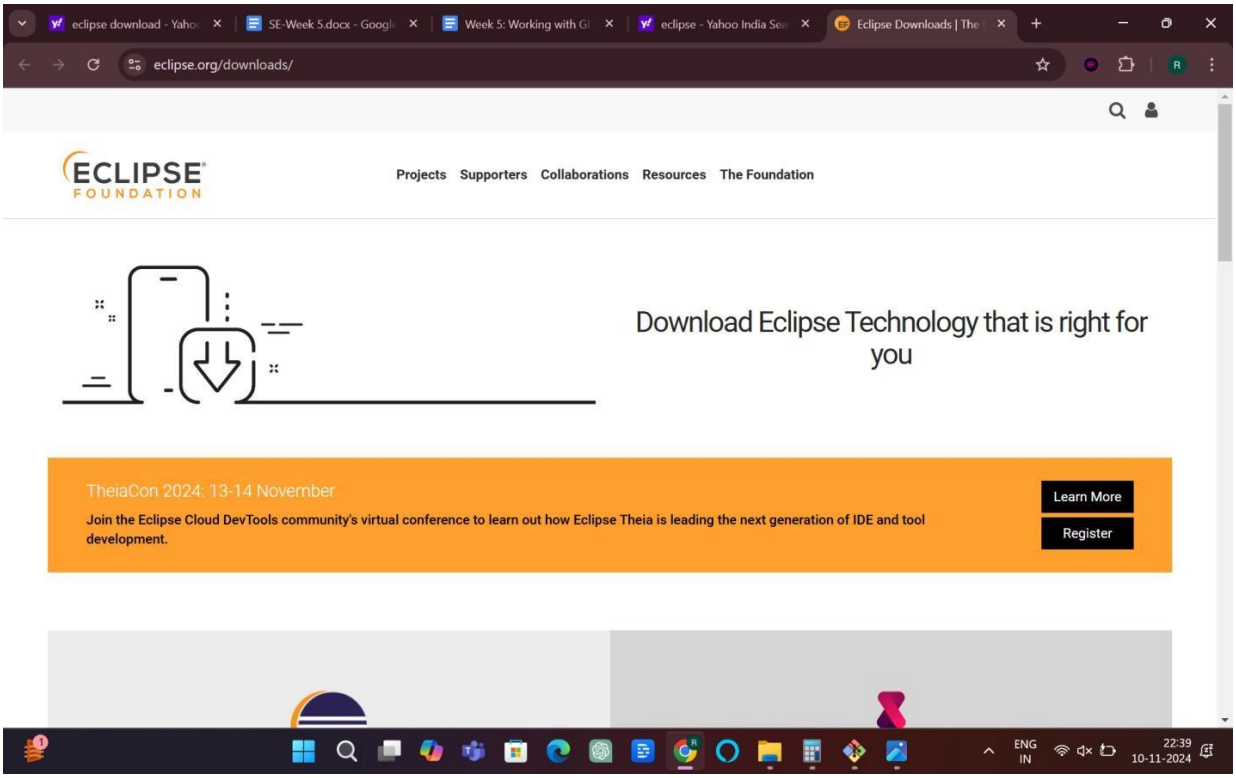
Visit the Eclipse Downloads Page:

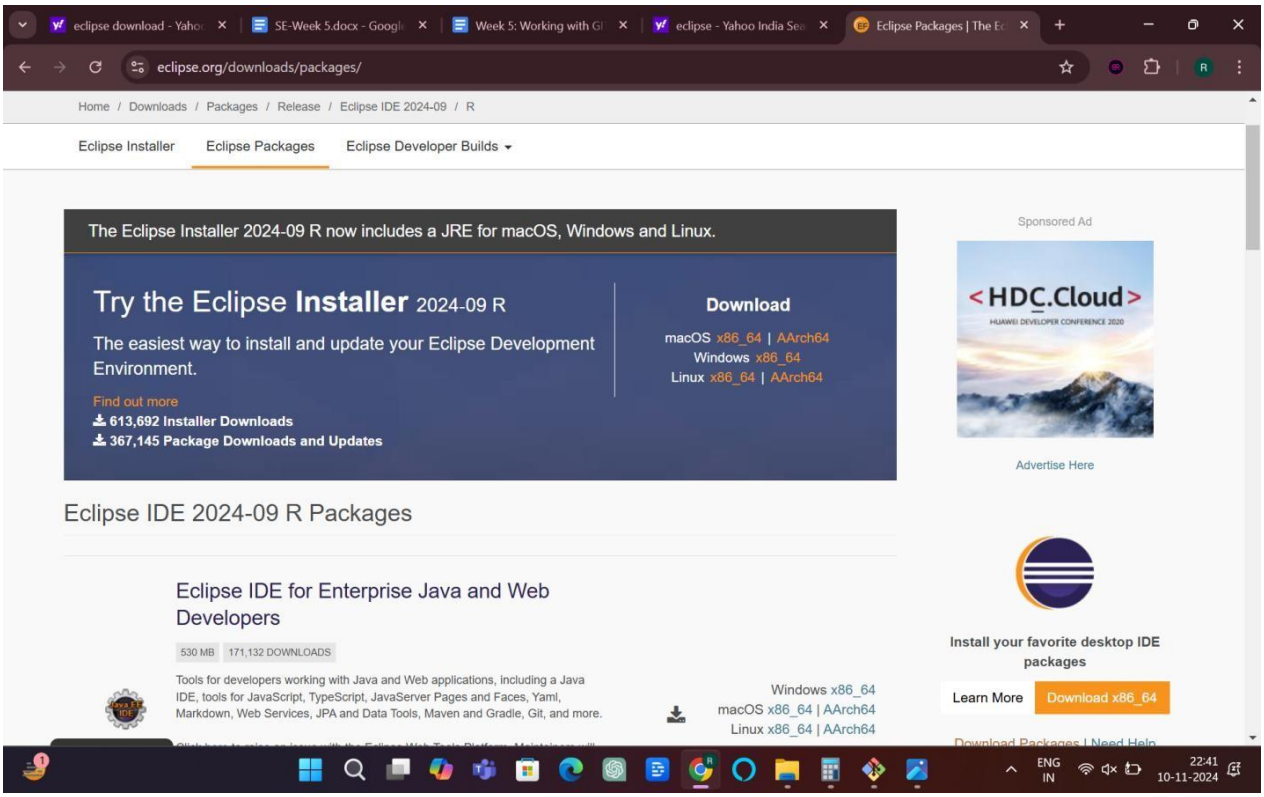
Go to the official Eclipse Downloads page.

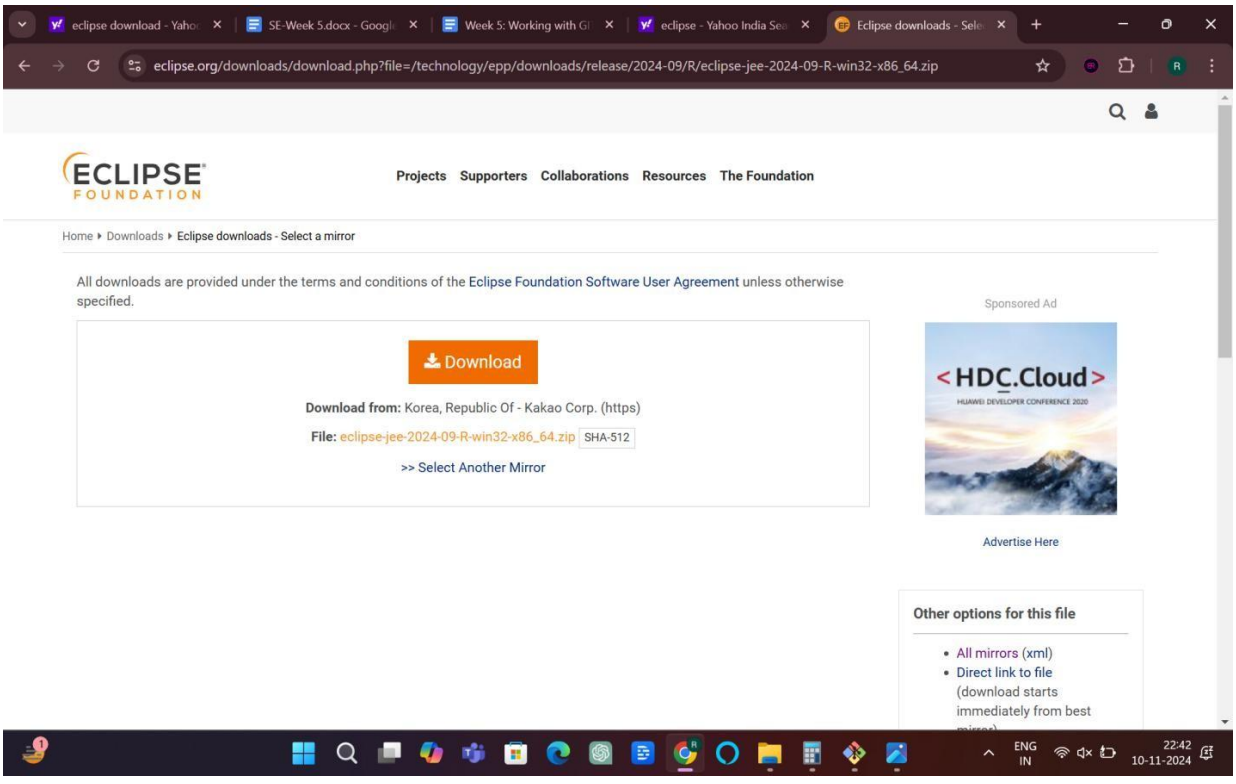
Download the Eclipse Installer:

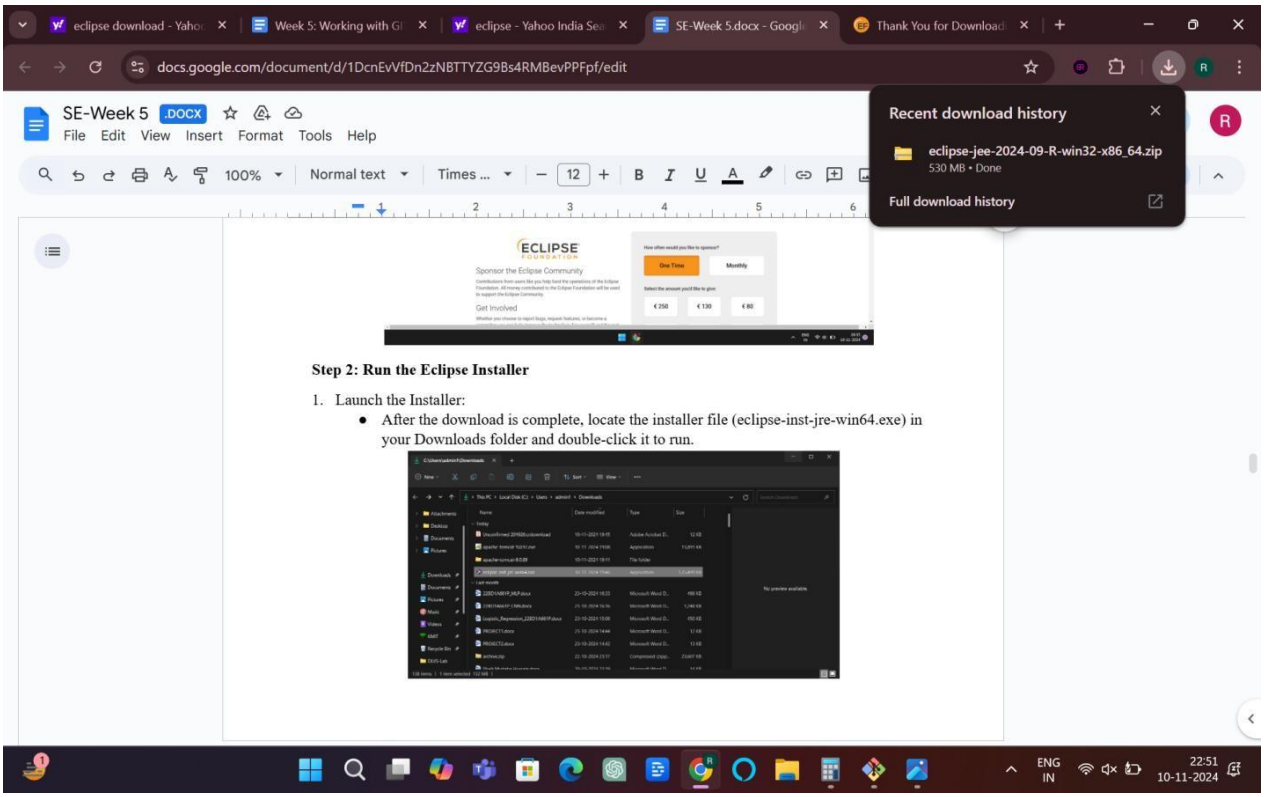
Scroll down to find the "Eclipse IDE for Enterprise Java and Web Developers" option.

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




eclipse download - Yaho
SE-Week 5.docx - Google
Week 5: Working with Gi
eclipse - Yahoo India Sea
Eclipse Packages | The Ec
eclipse.org/downloads/packages/
Home / Downloads / Packages / Release / Eclipse IDE 2024-09 / R
Eclipse Installer
Eclipse Packages
Eclipse Developer Builds
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
Advertise Here
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
ENG
IN
22:41
10-11-2024


eclipse download - Yaho
SE-Week 5.docx - Google
Week 5: Working with Gi
eclipse - Yahoo India Sea
Eclipse downloads - Sele
eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2024-09/R/eclipse-jee-2024-09-R-win32-x86_64.zip
ECLIPSE FOUNDATION
Projects
Supporters
Collaborations
Resources
The Foundation
Home ▸ Downloads ▸ Eclipse downloads - Select a mirror
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)
File: eclipse-jee-2024-09-R-win32-x86_64.zip SHA-512
>> Select Another Mirror
Sponsored Ad
<HDC.Cloud>
HUAWEI DEVELOPER CONFERENCE 2020
Advertise Here
Other options for this file
All mirrors (xml)
Direct link to file
(download starts immediately from best
ENG
IN
22:42
10-11-2024



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:

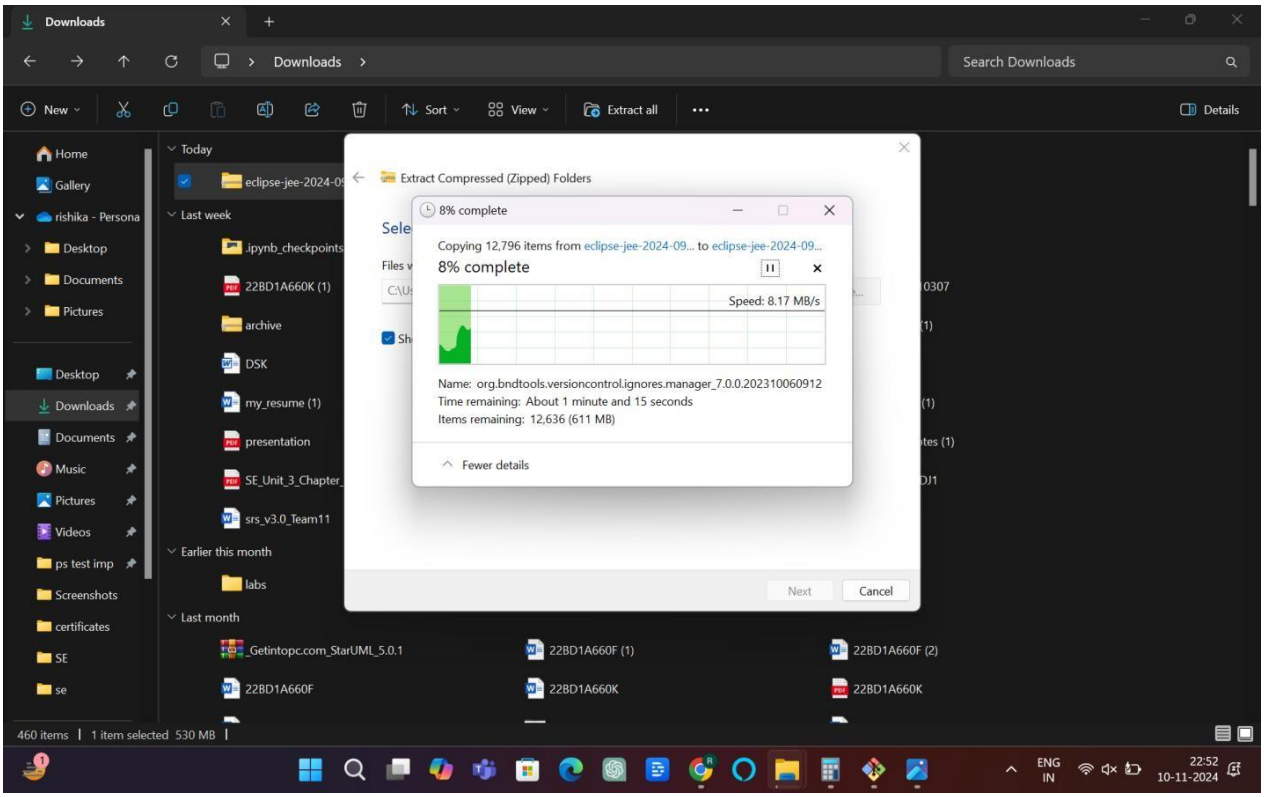
The installer should automatically detect your installed JDK. Ensure that it points to the correct JDK version. If not, you can manually browse and select the correct JDK path.

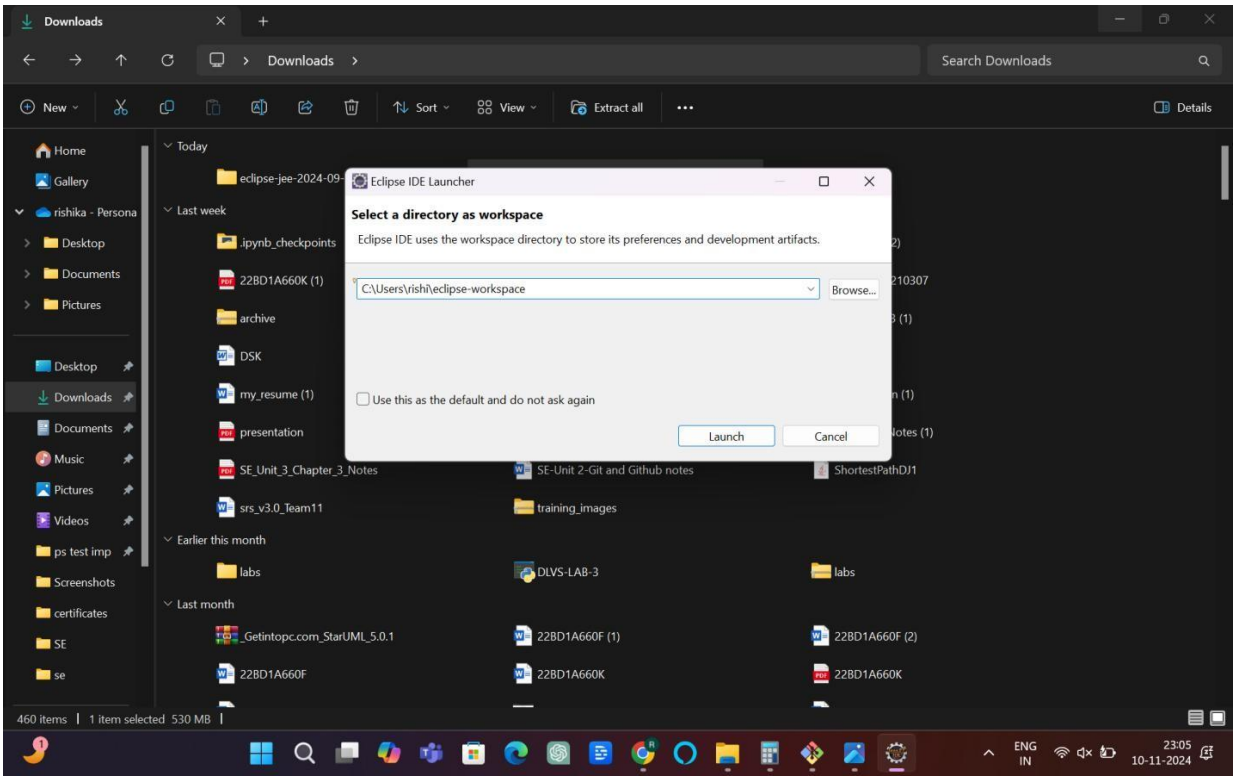
Start the Installation:

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:

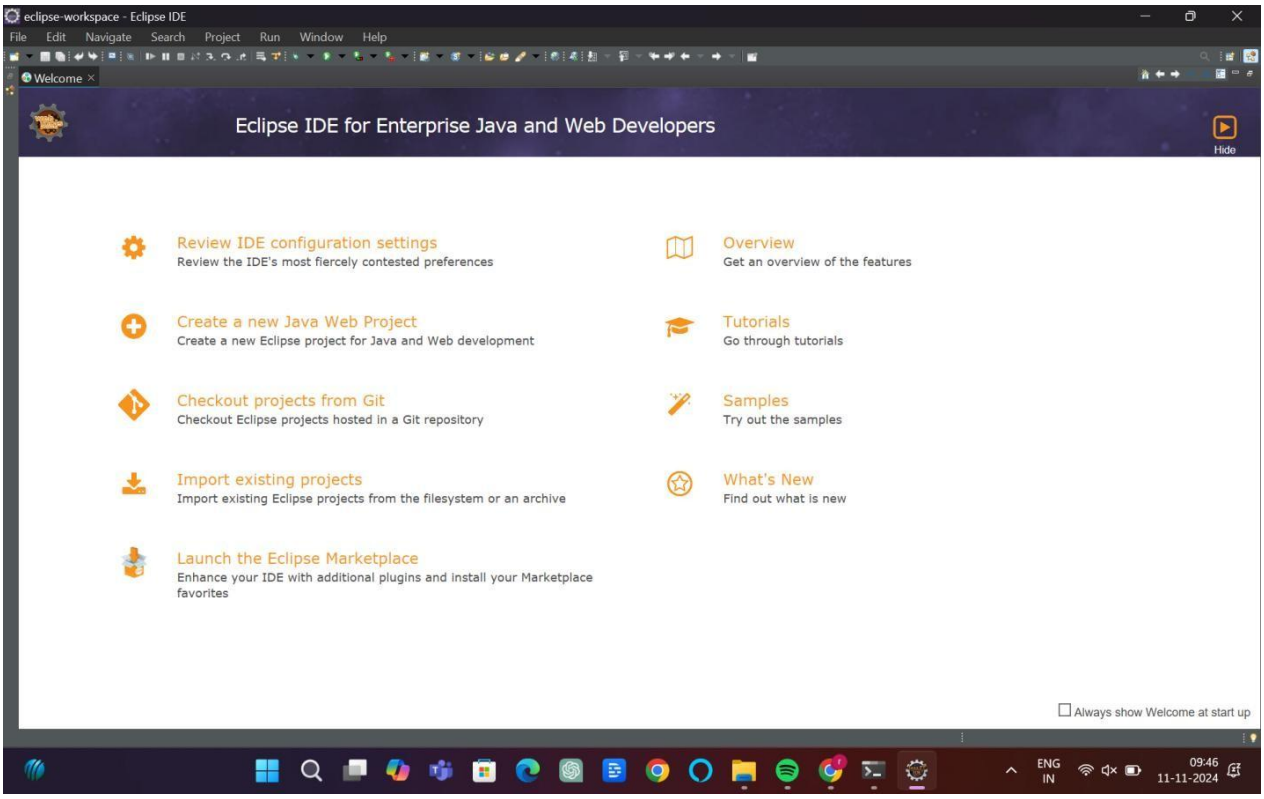
Once the installation is complete, click on the "LAUNCH" button to start Eclipse.

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.



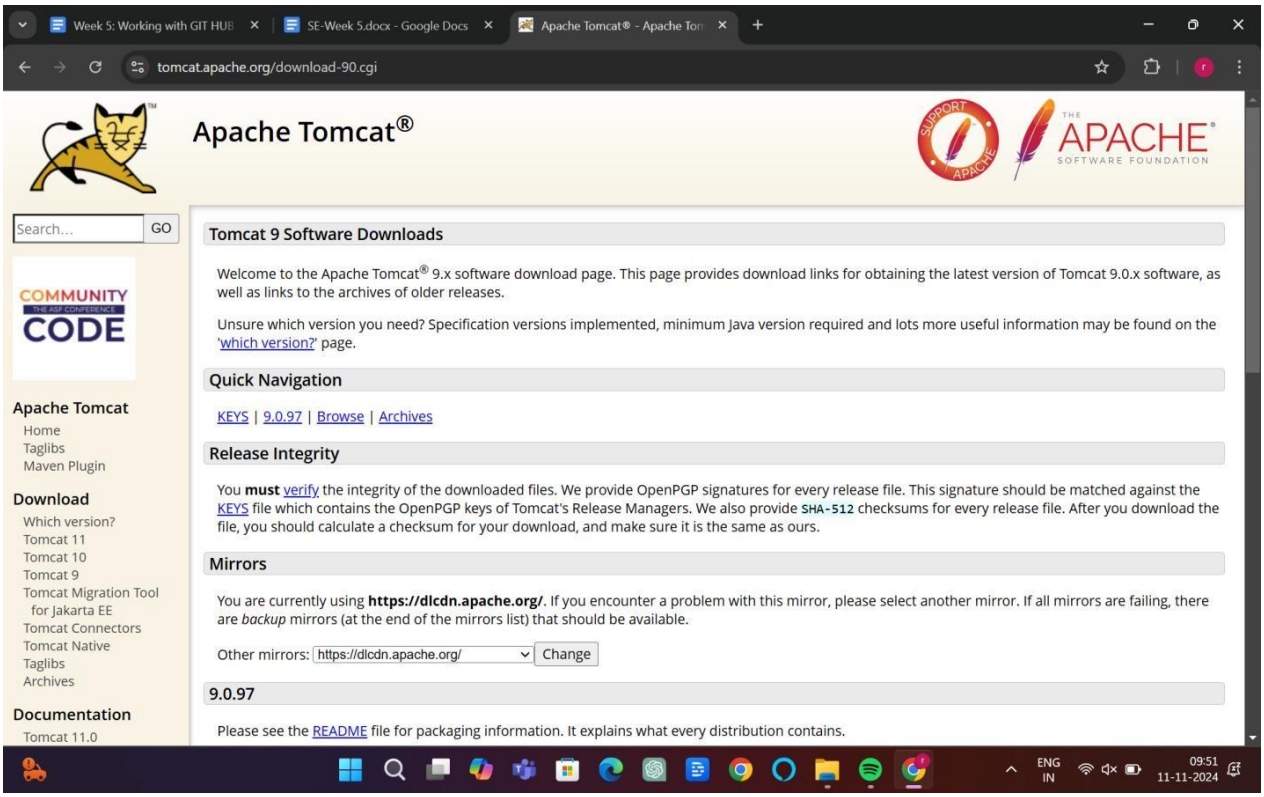
Download and install Tomcat :

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.

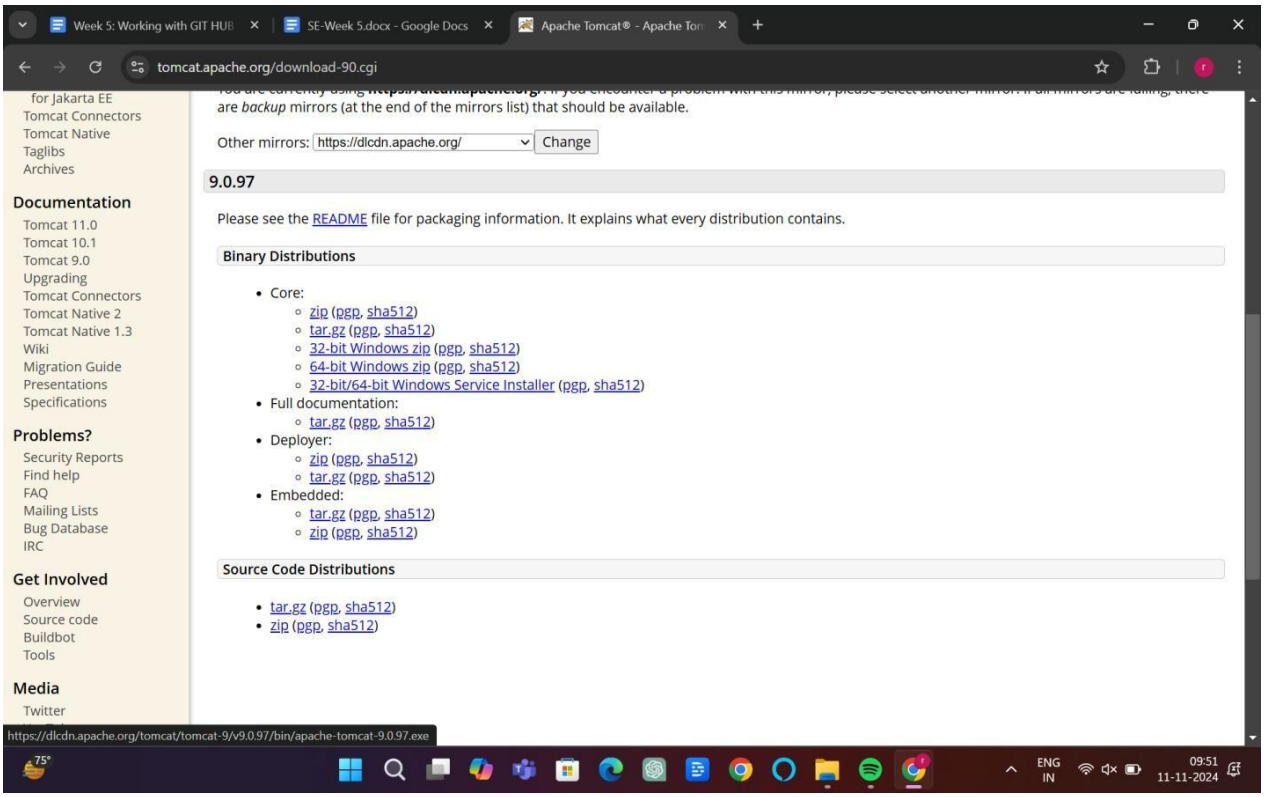
Installing Apache Tomcat 9.0 on Windows 10 involves downloading the software, setting up the environment, and configuring Tomcat.

Download Apache Tomcat 9.0

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



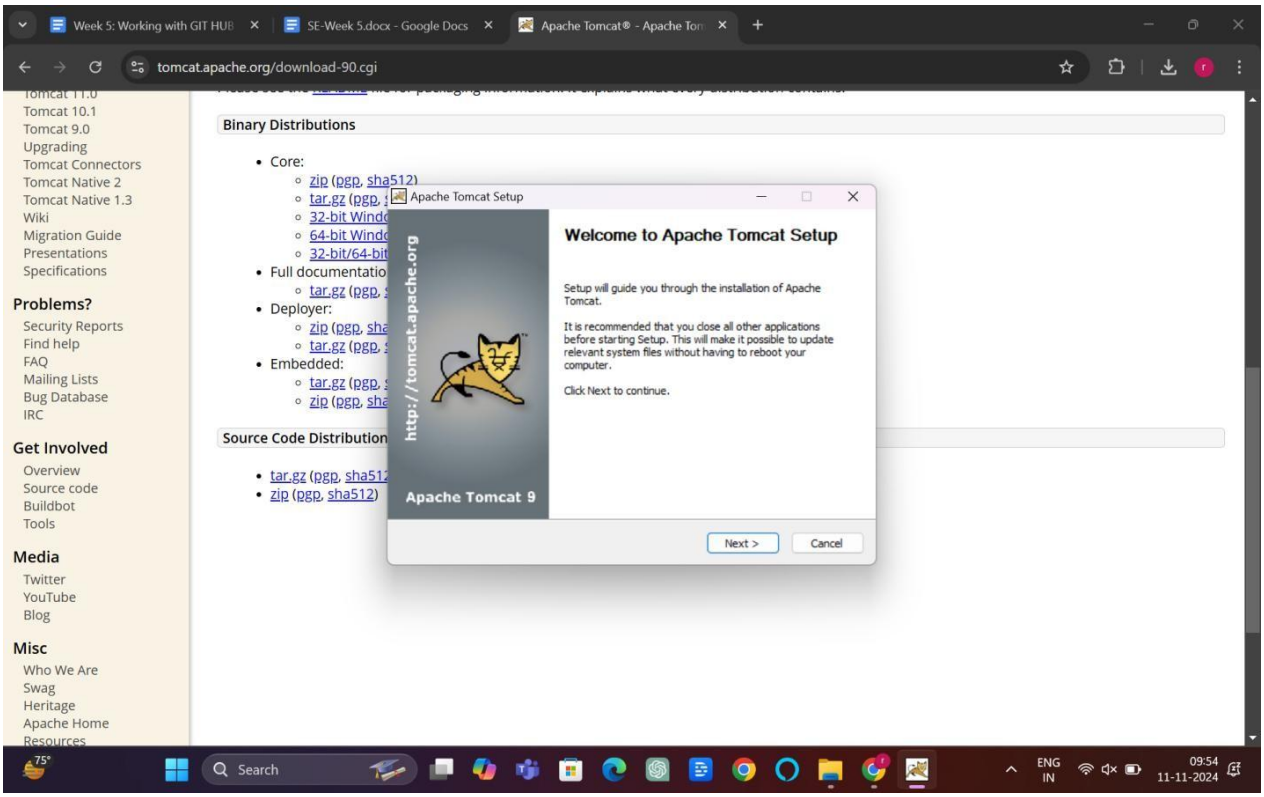
Under the **Binary Distributions** section, find the "Core" section and download the **32-bit/64-bit Windows Service Installer** (`.exe` file).



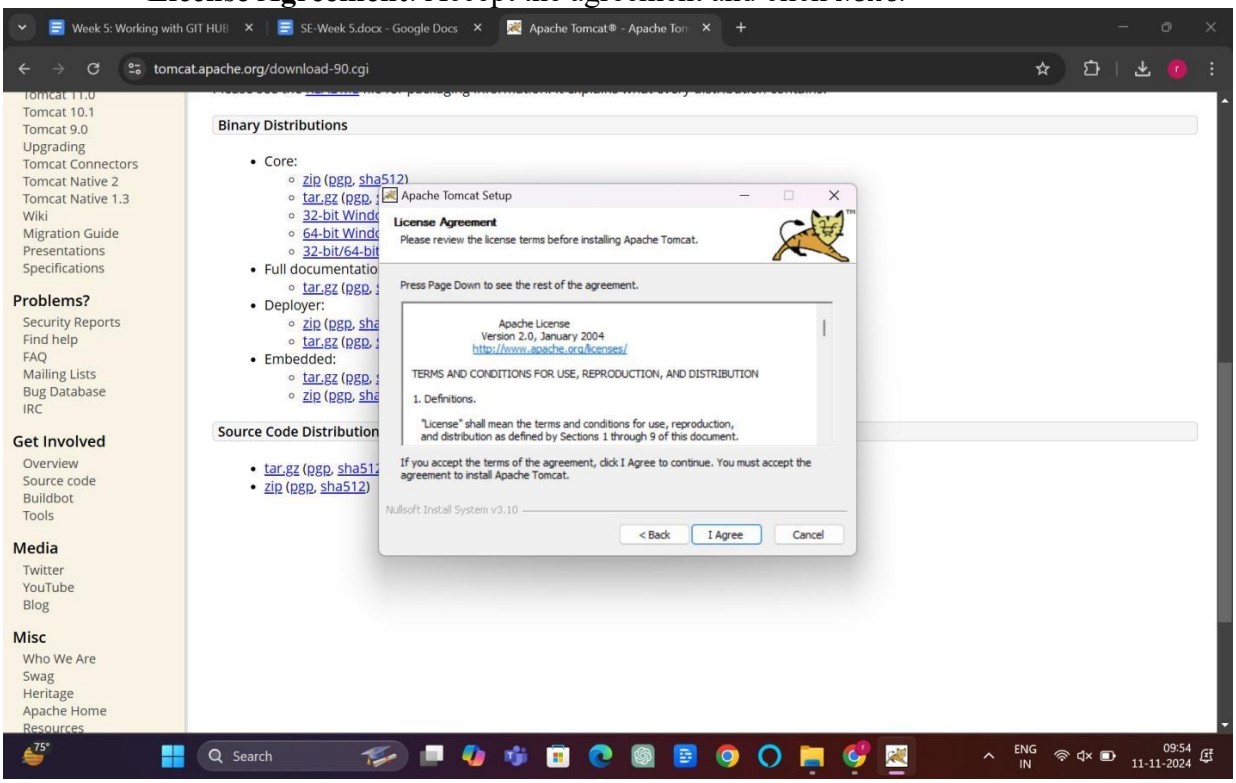
Run the Installer

Locate the downloaded `.exe` file and double-click to start the installation.
During the installation, follow these steps: o

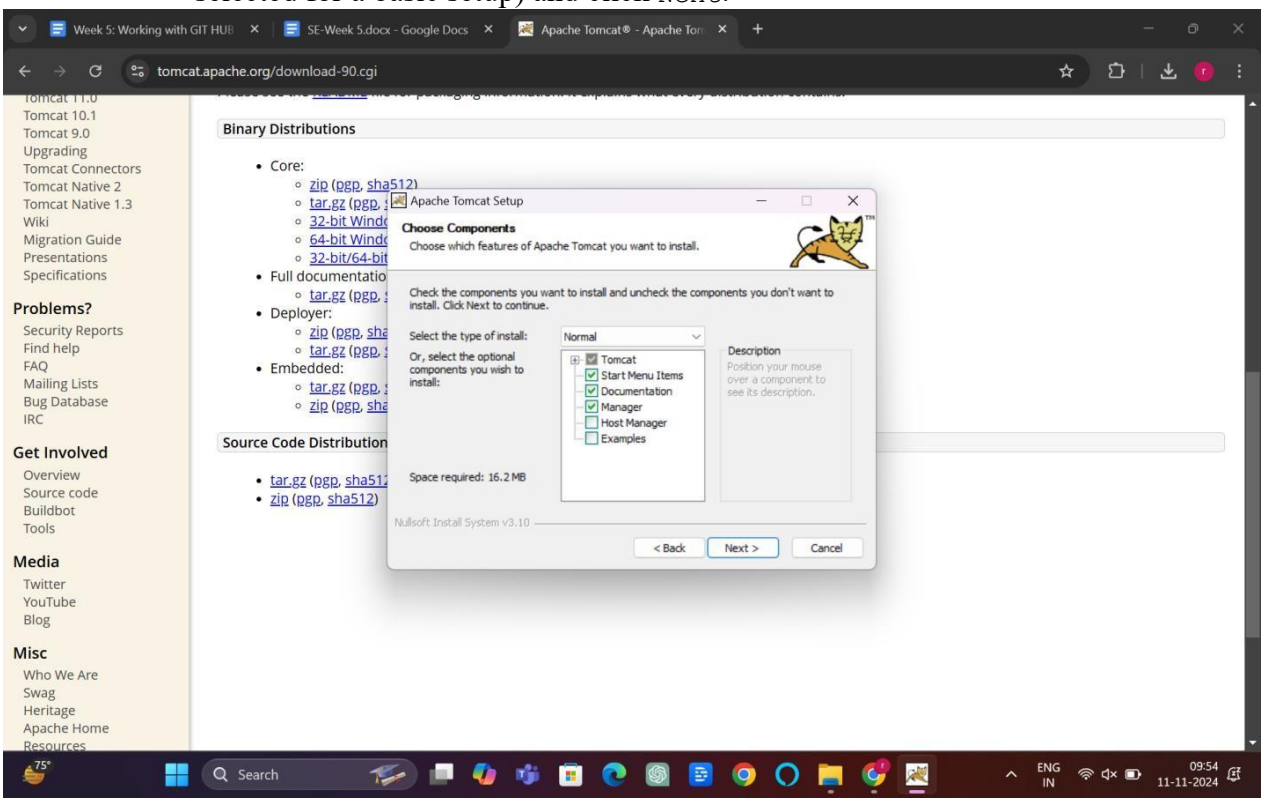
Welcome Screen: Click `Next`.



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:

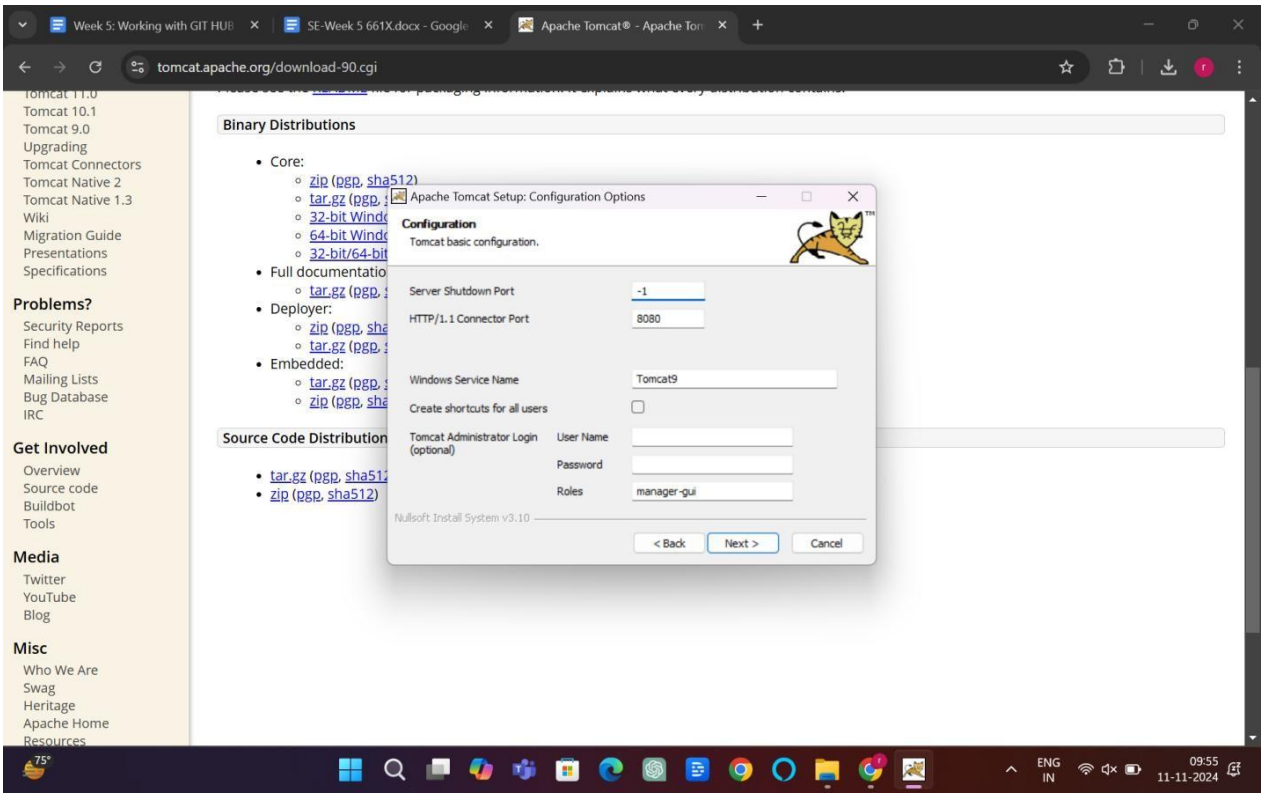
Set the **Tomcat Server** ports:

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.

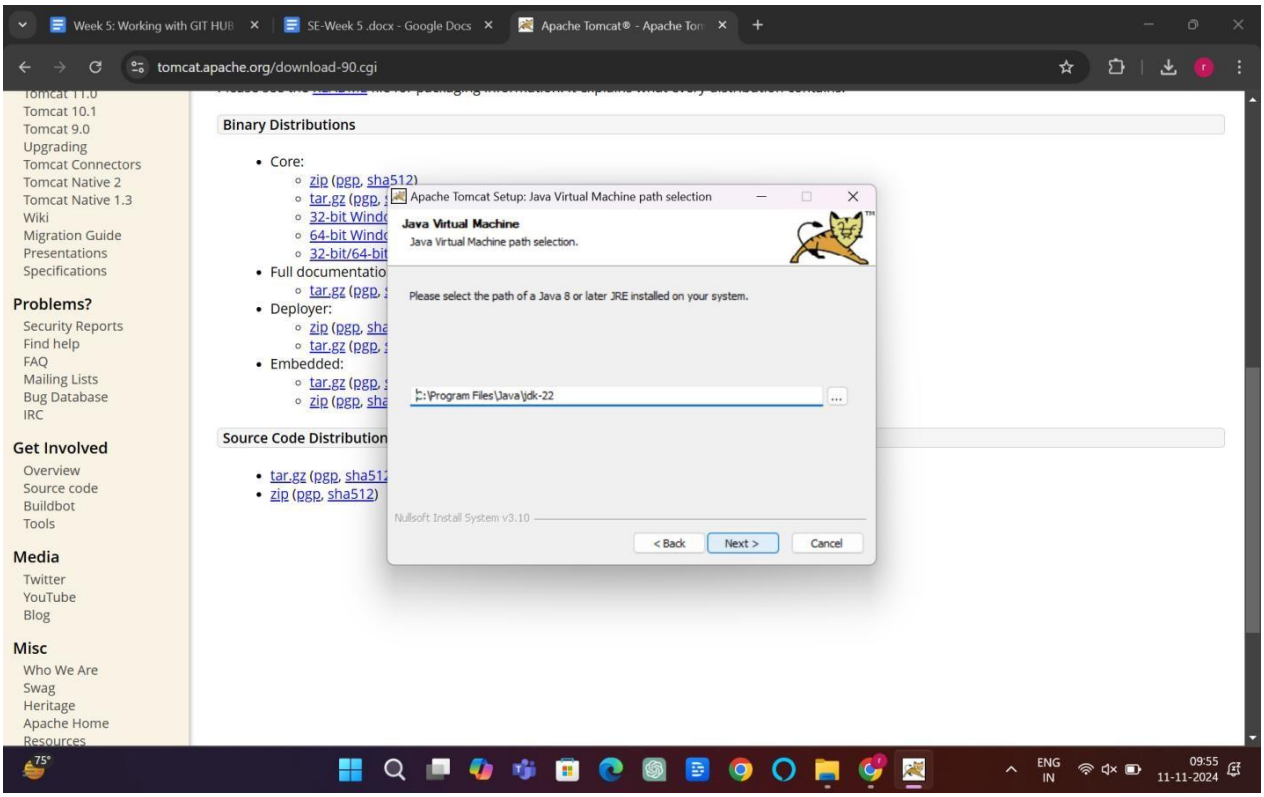
Tomcat Administrator Login: Enter a username and password (this is optional).

Click `Next`.



Choose Java Virtual Machine:

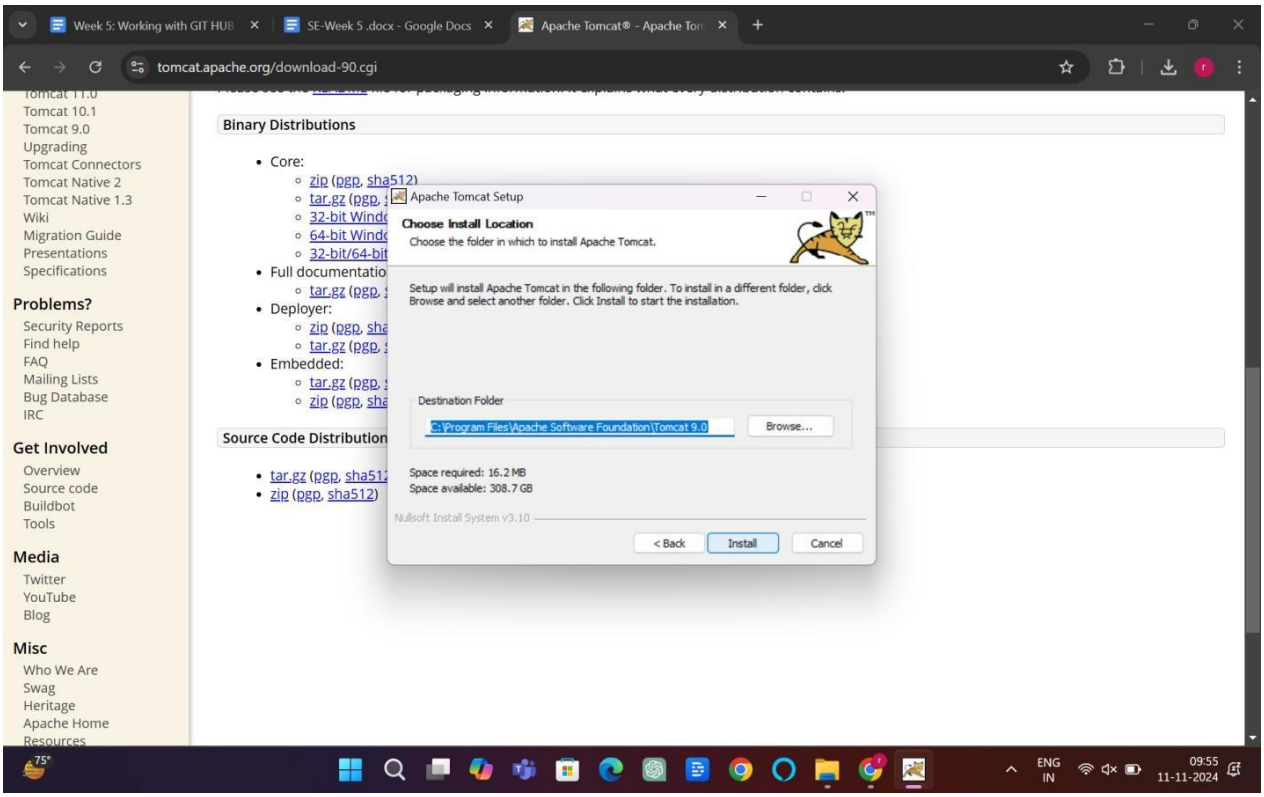
Ensure the correct path to your JDK (Java Development Kit) is set. If you haven't installed the JDK, you'll need to install it first.



Choose Install Location:

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

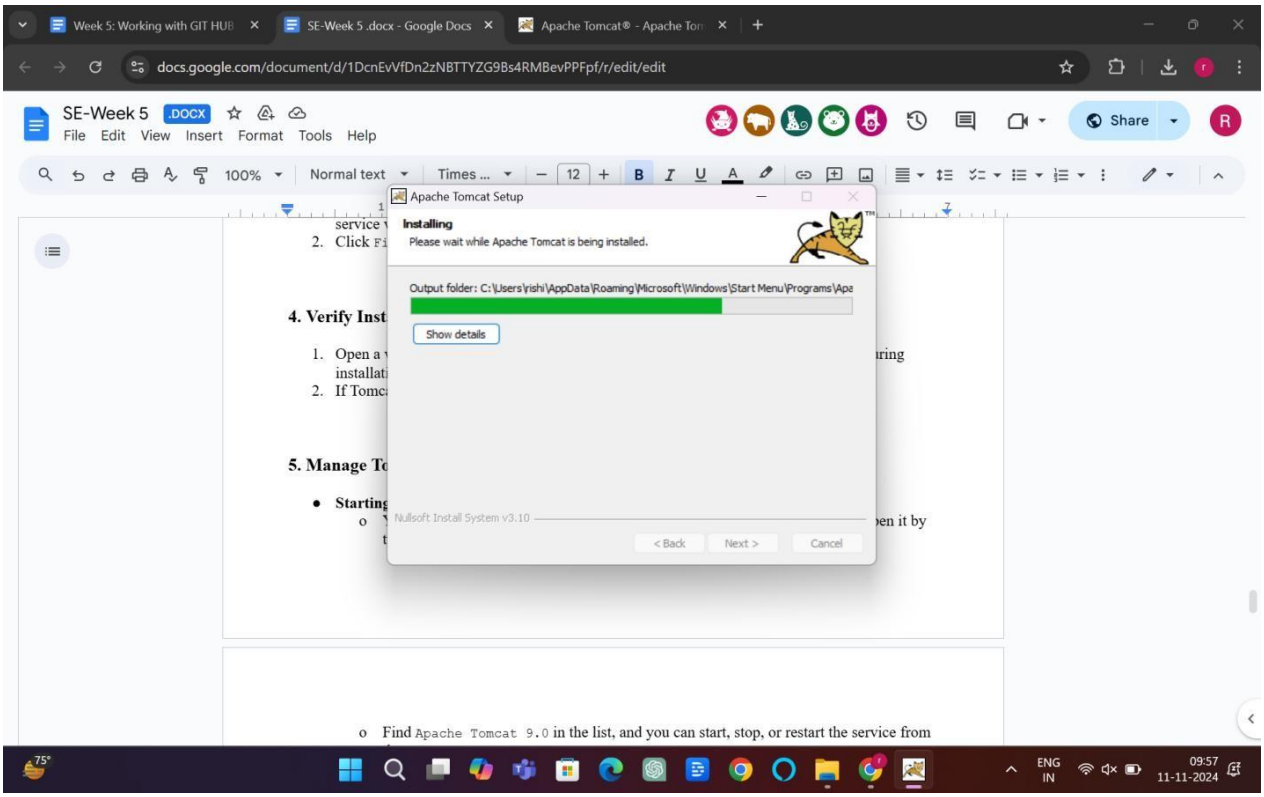
Click `Next`, then click `Install` to begin the installation.

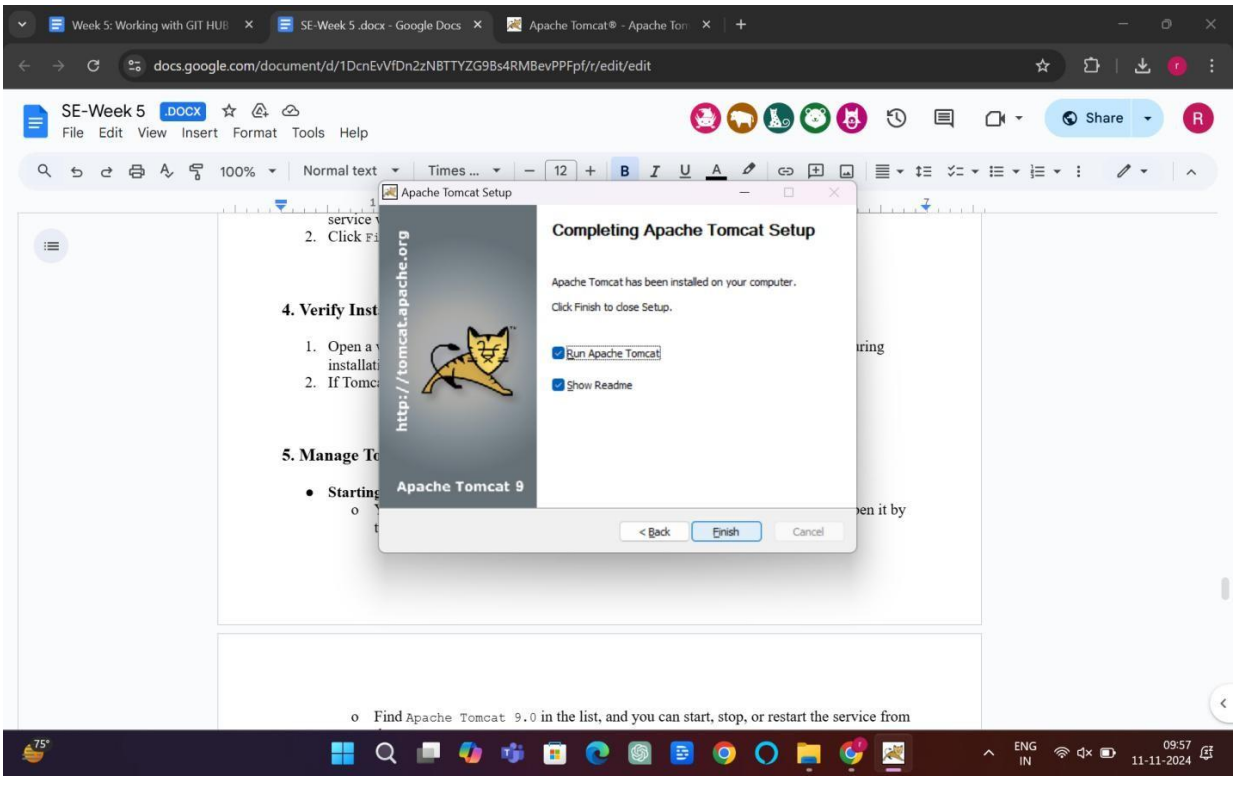


Complete the Installation

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

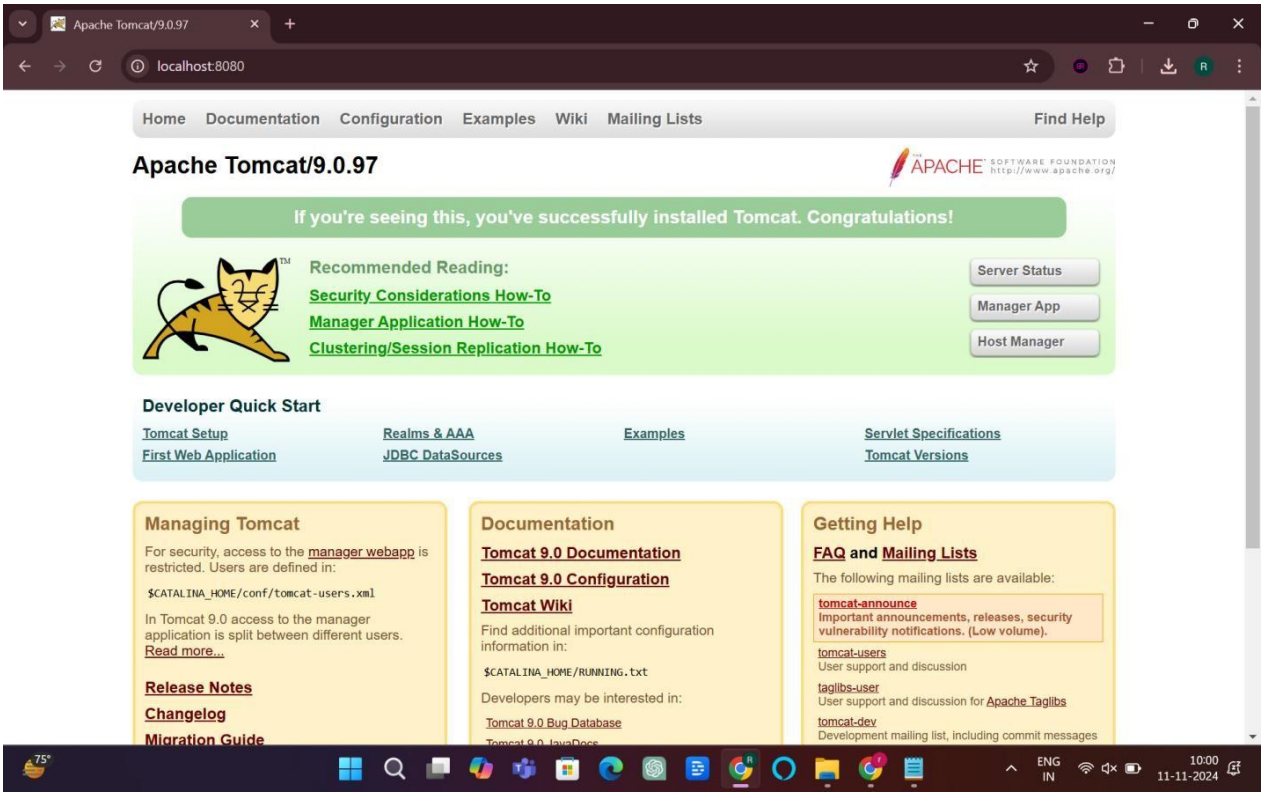
Click `Finish` to exit the installer.


Week 5: Working with GIT HUB
SE-Week 5 .docx - Google Docs
Apache Tomcat® - Apache Tom
docs.google.com/document/d/1DcnEvVfDn2zNBTTYZG9Bs4RMBevPPFpf/r/edit/edit
SE-Week 5 .DOCX
File Edit View Insert Format Tools Help
Share
Apache Tomcat Setup
Installing
Please wait while Apache Tomcat is being installed.
Output folder: C:\Users\rishi\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Apa
Show details
Nullsoft Install System v3.10
< Back
Next >
Cancel
4. Verify Inst
5. Manage To
Starting
Find Apache Tomcat 9.0 in the list, and you can start, stop, or restart the service from
ENG
IN
09:57
11-11-2024


Week 5: Working with GIT HUB
SE-Week 5 .docx - Google Docs
Apache Tomcat® - Apache Tom
docs.google.com/document/d/1DcnEvVfDn2zNBTTYZG9Bs4RMBevPPFpf/r/edit/edit
SE-Week 5 .DOCX
File Edit View Insert Format Tools Help
Share
Apache Tomcat Setup
Completing Apache Tomcat Setup
Apache Tomcat has been installed on your computer.
Click Finish to close Setup.
Run Apache Tomcat
Show Readme
http://tomcat.apache.org
Apache Tomcat 9
< Back
Finish
Cancel
4. Verify Inst
5. Manage To
Starting
Find Apache Tomcat 9.0 in the list, and you can start, stop, or restart the service from
ENG
IN
09:57
11-11-2024

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.

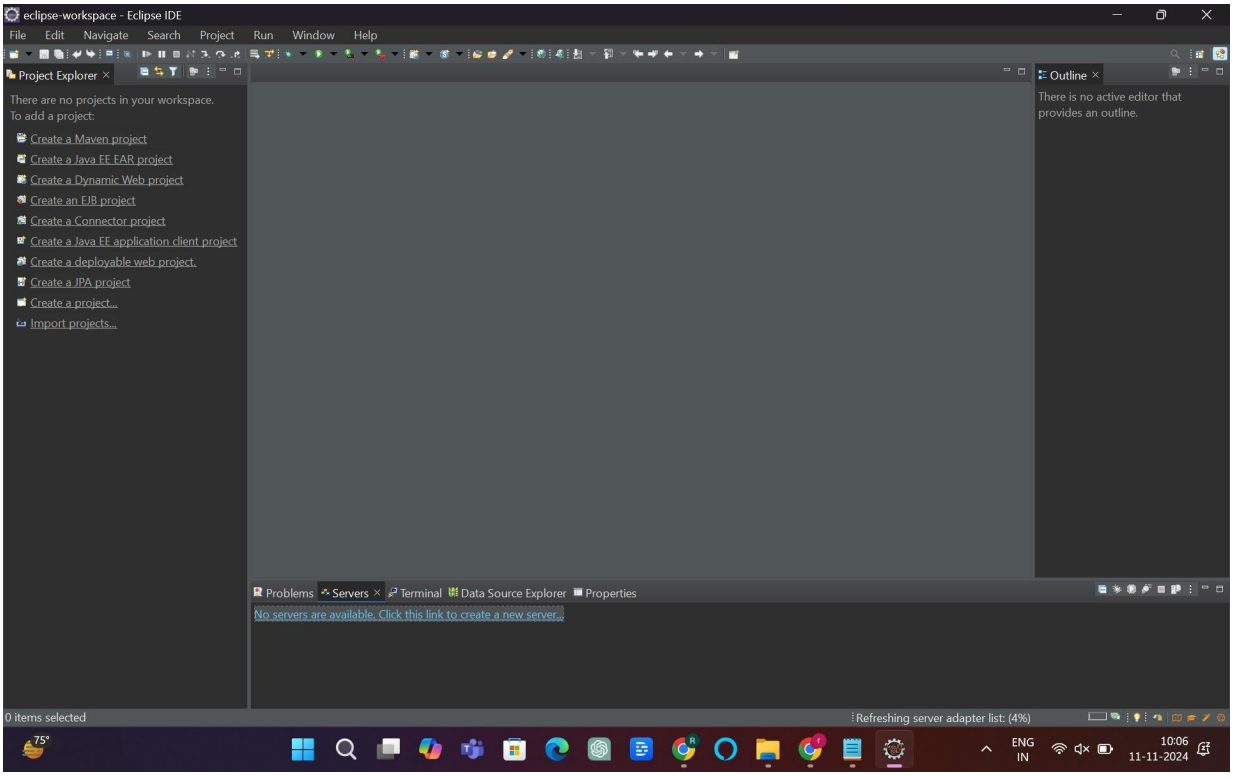
Configure Tomcat Web-Server on Eclipse

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.

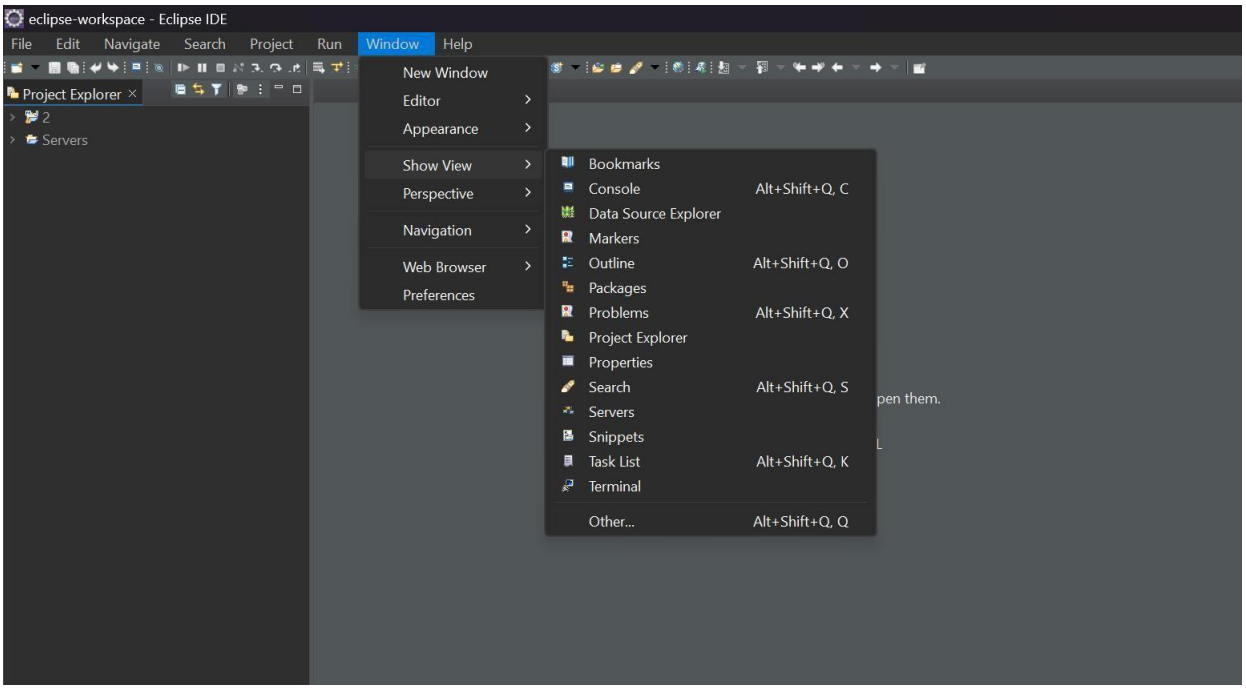
Open Eclipse and Configure the Server

Open Eclipse: Start your Eclipse IDE.

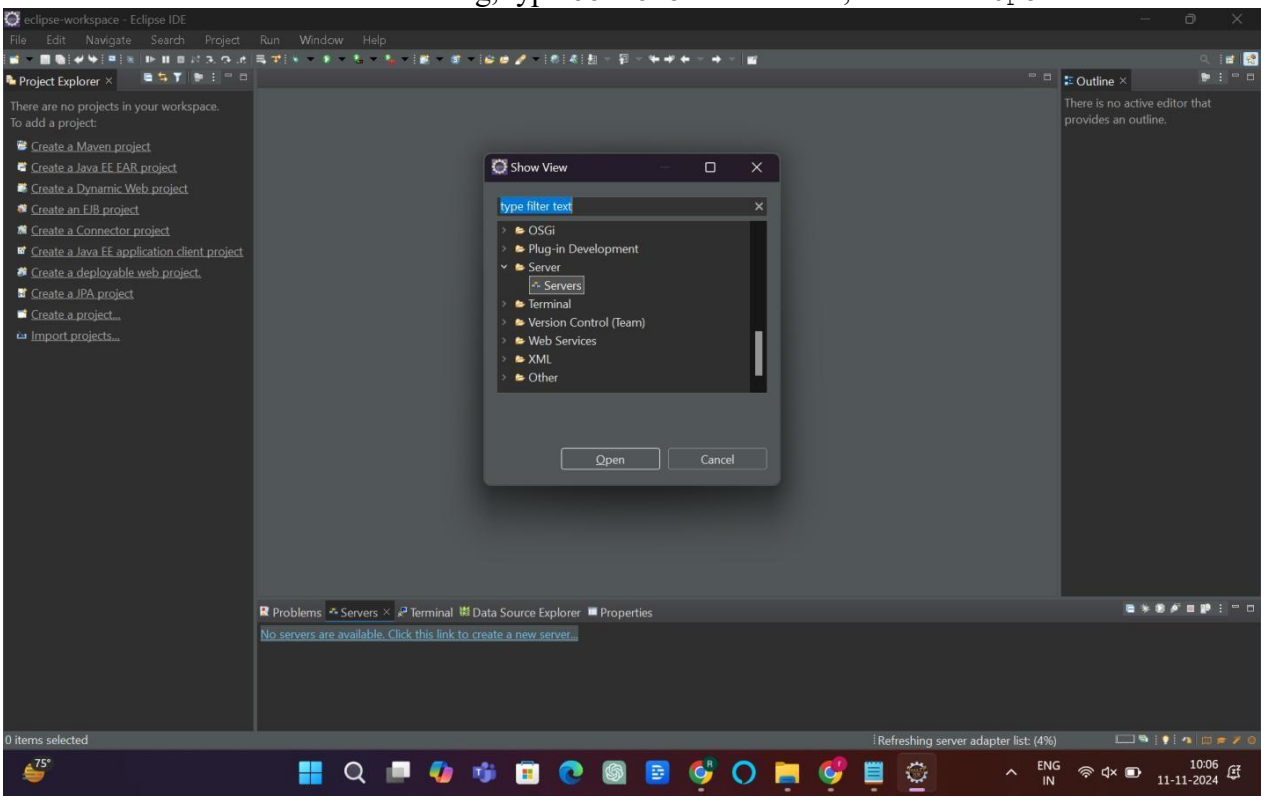


Open the Servers View:

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

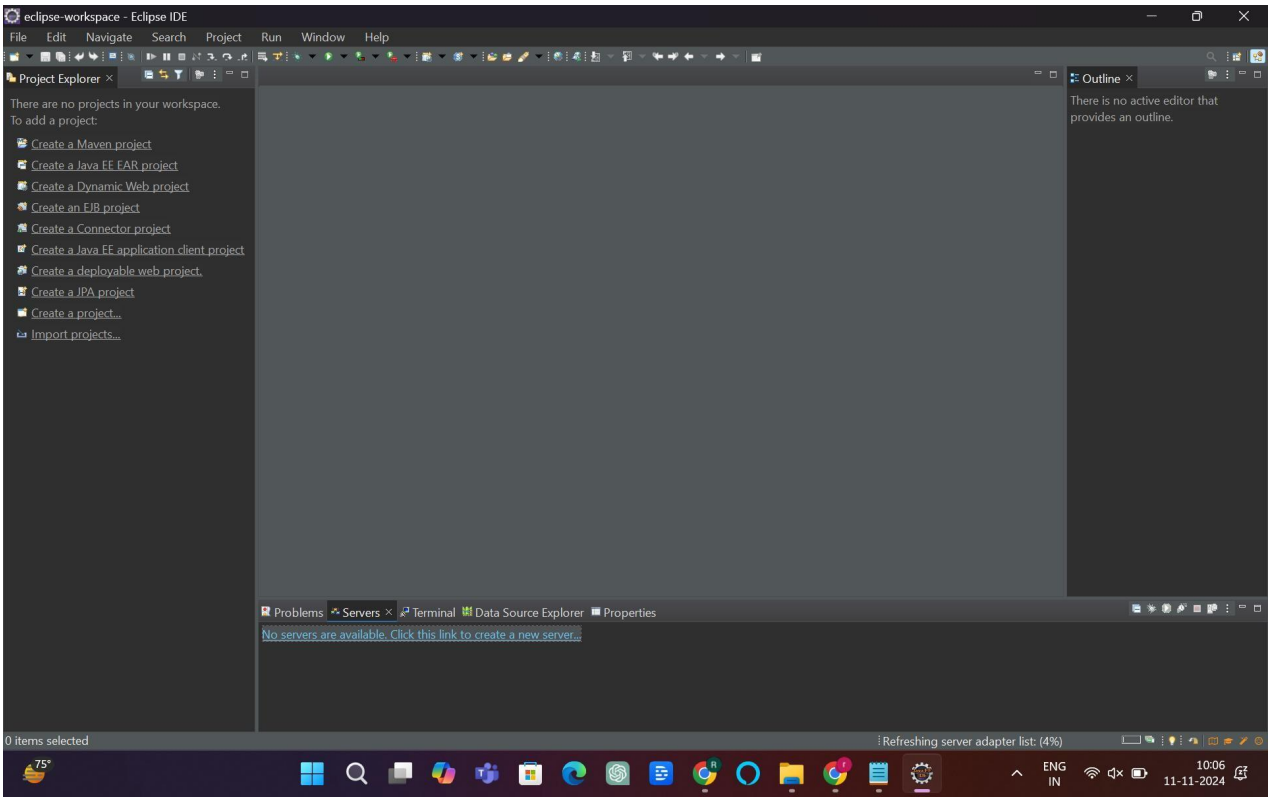


In the "Show View" dialog, type `Servers` and select it, then click `Open`.

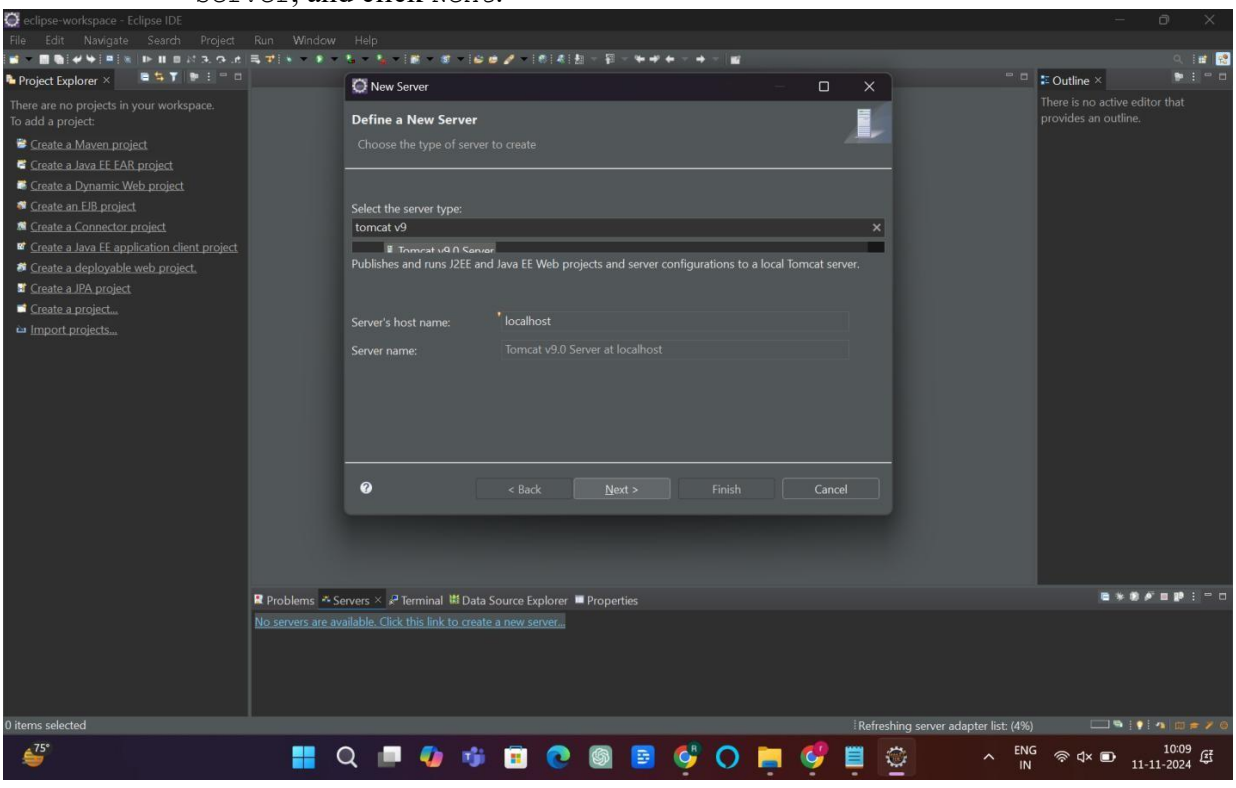


Add a New Server:

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



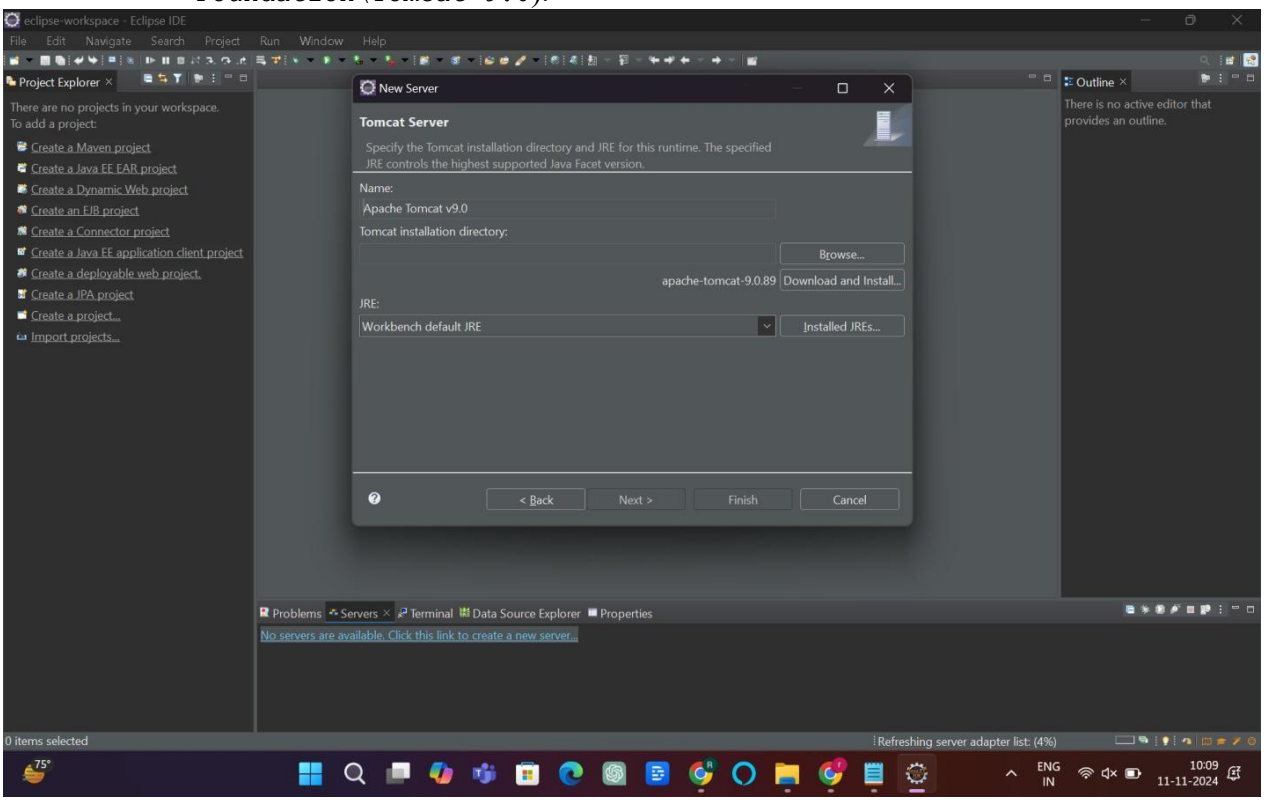
In the "New Server" wizard, expand the "Apache" node, select `Tomcat v9.0 Server`, and click `Next`.



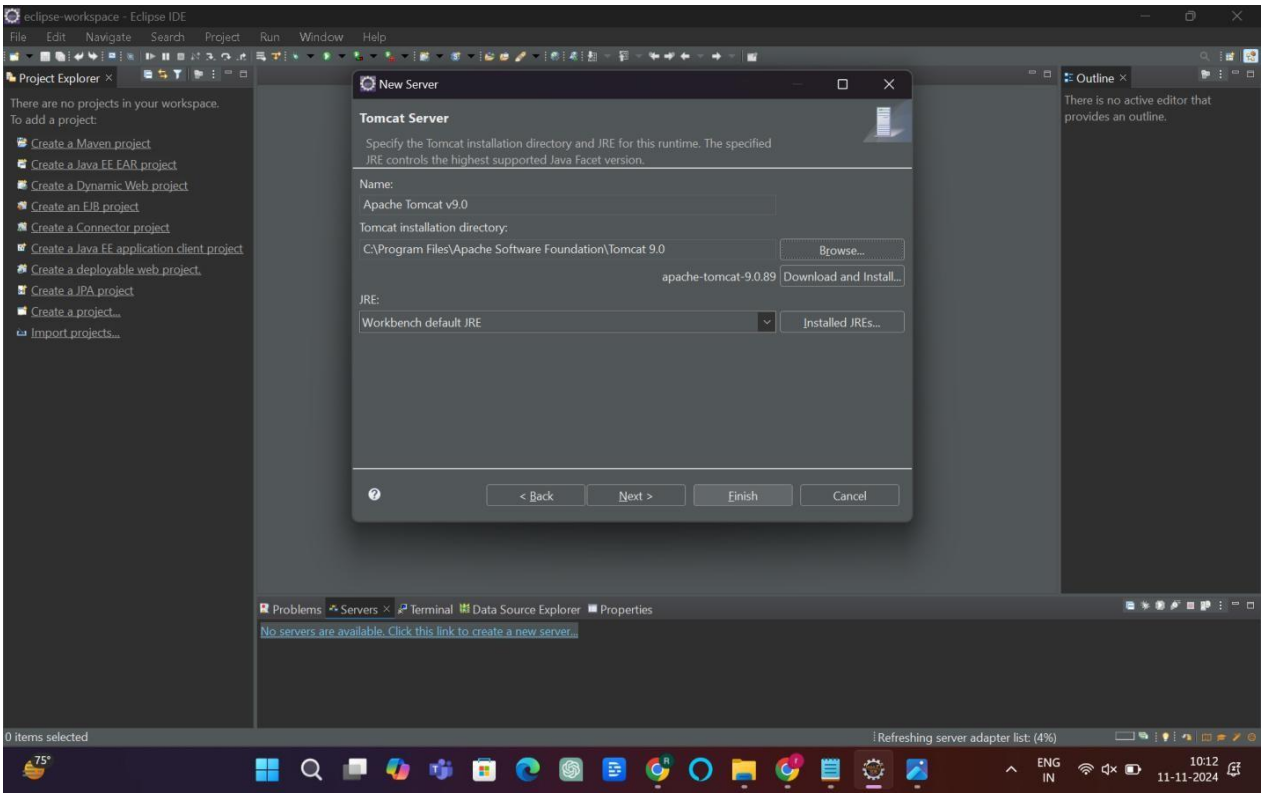
Configure the Tomcat Server:

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.

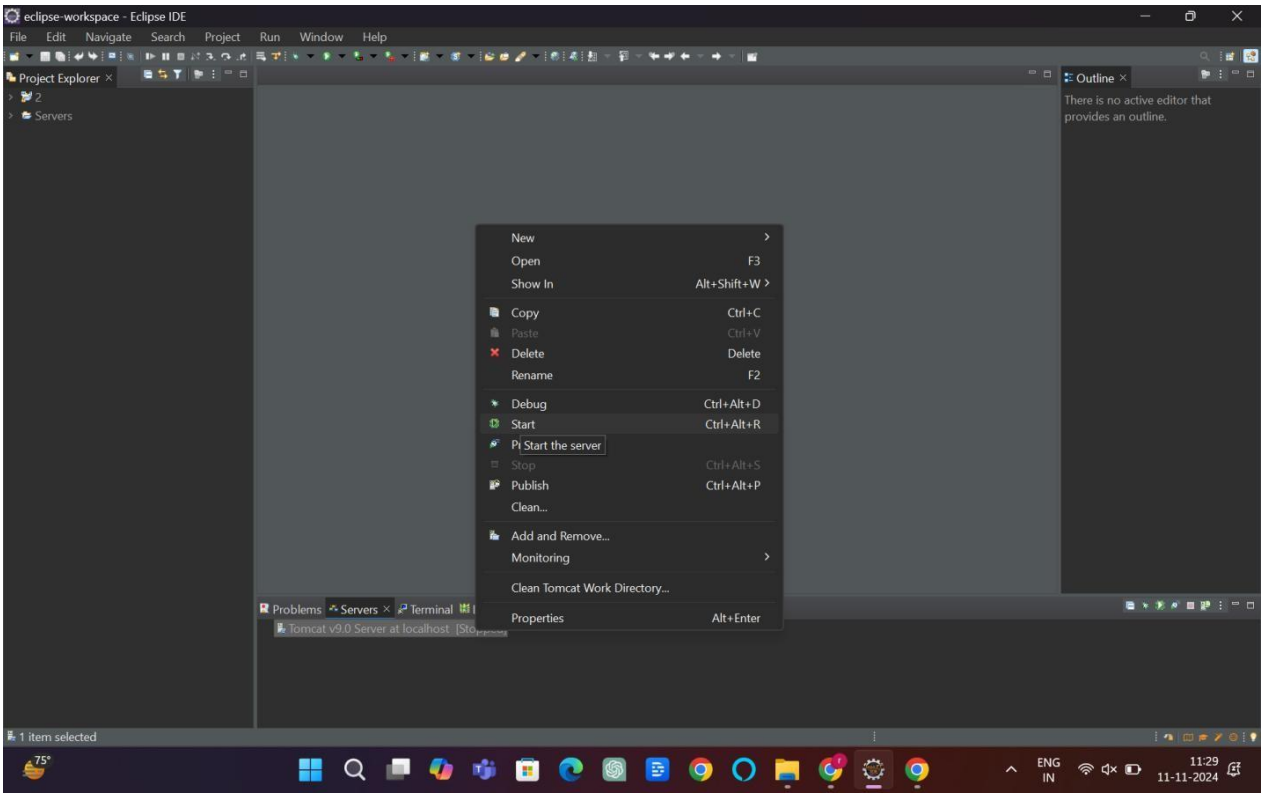
Click `Finish`.

Test the Server Configuration

Start the Server:

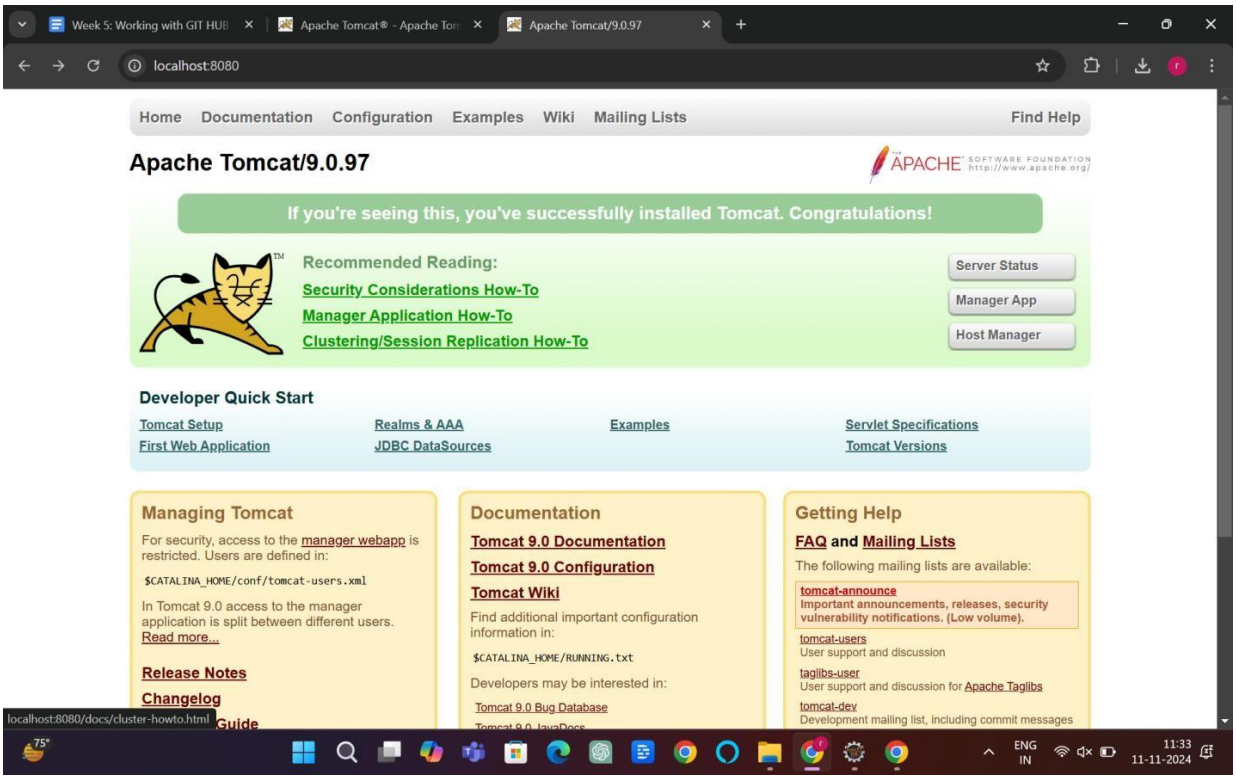
In the "Servers" view, right-click on the newly added Tomcat server and select `Start`.

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



Access Tomcat:

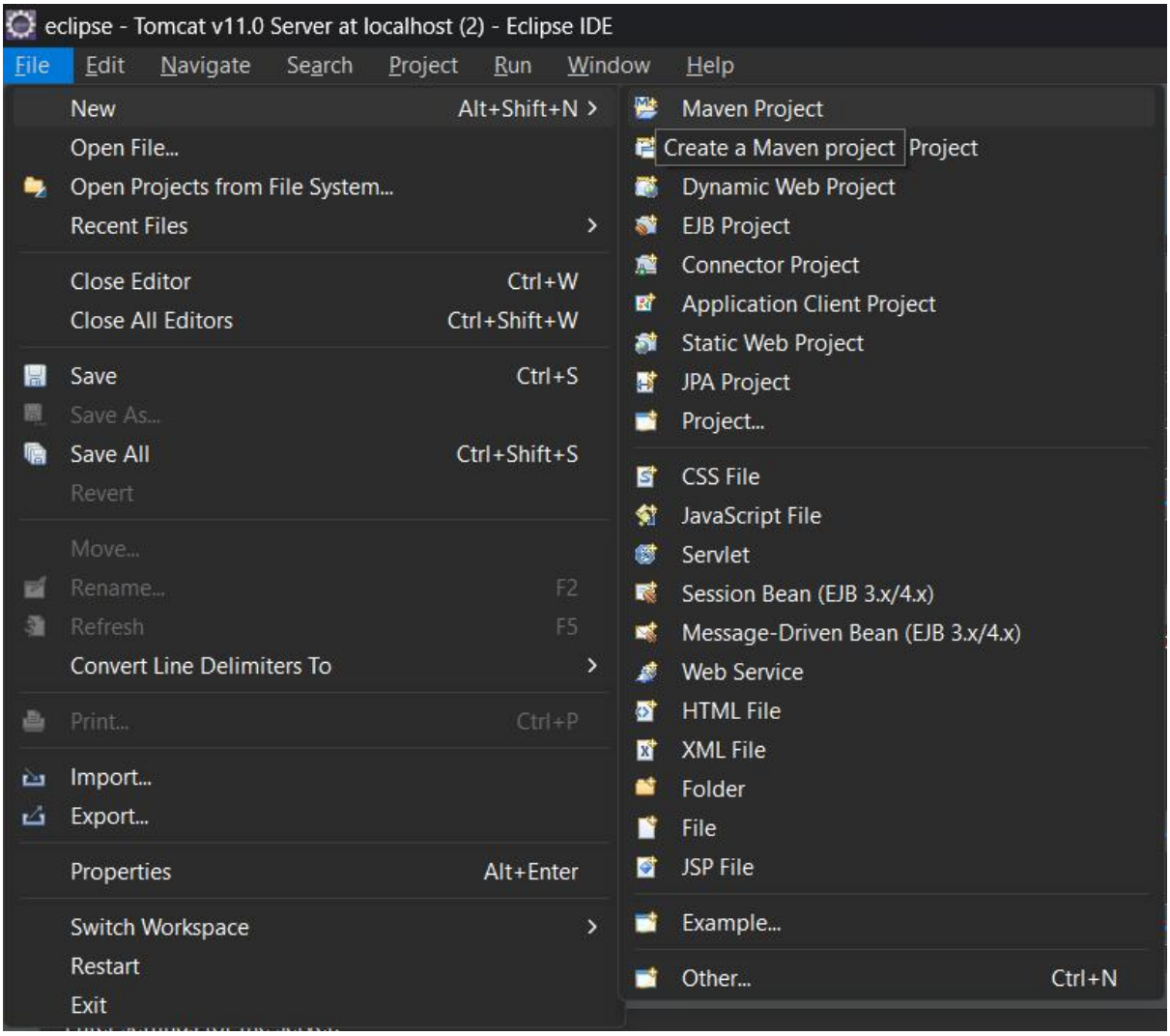
Open a web browser and go to `http://localhost:8080`. You should see the Tomcat default homepage, confirming that the server is running correctly.



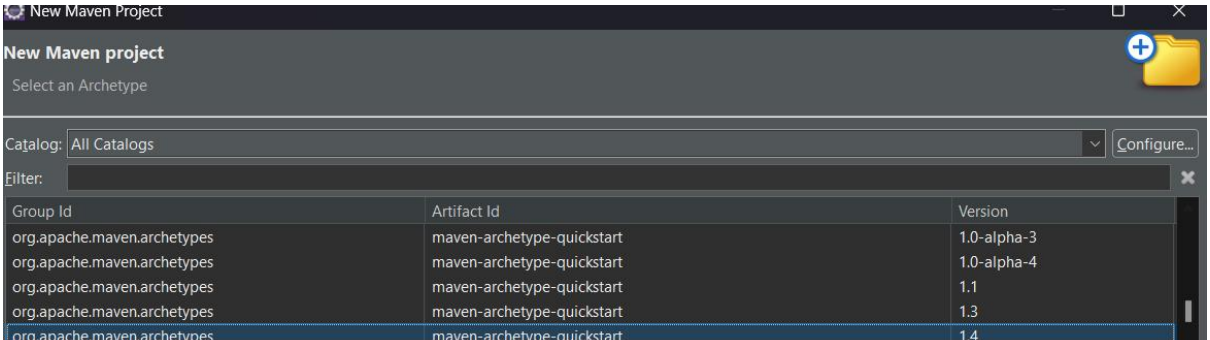
CREATING MAVEN JAVA PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB.

Setting up maven project in eclipse

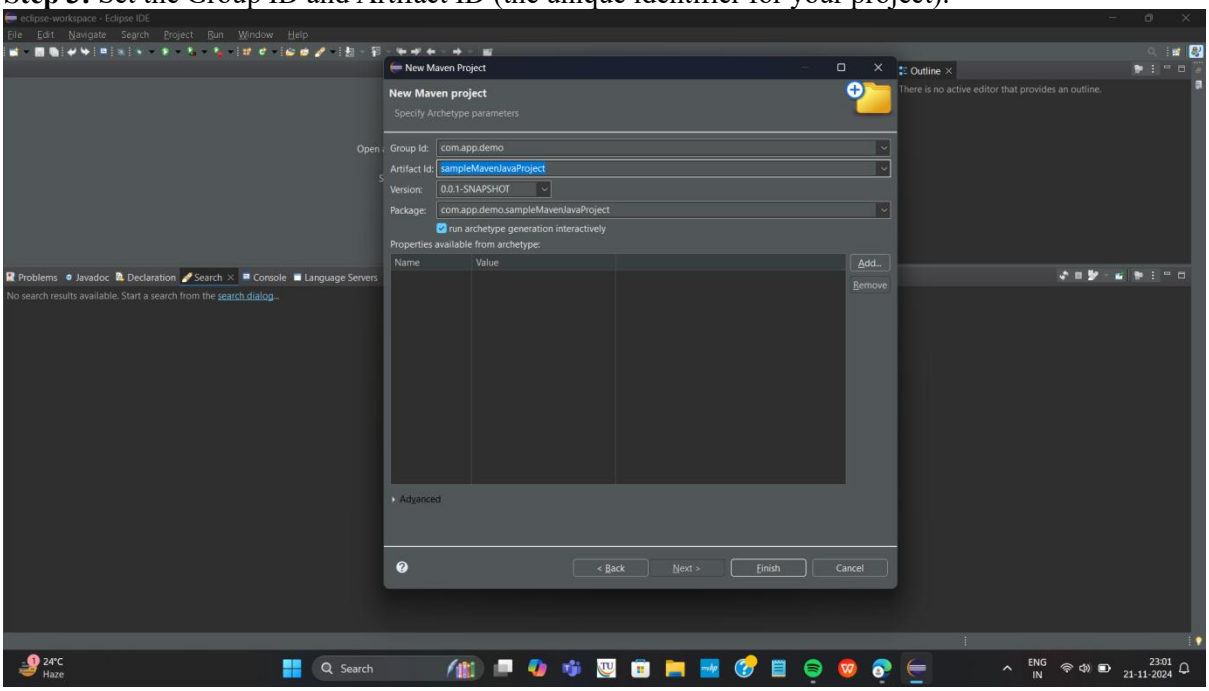
Step 1: In Eclipse, go to File > New > Maven Project.



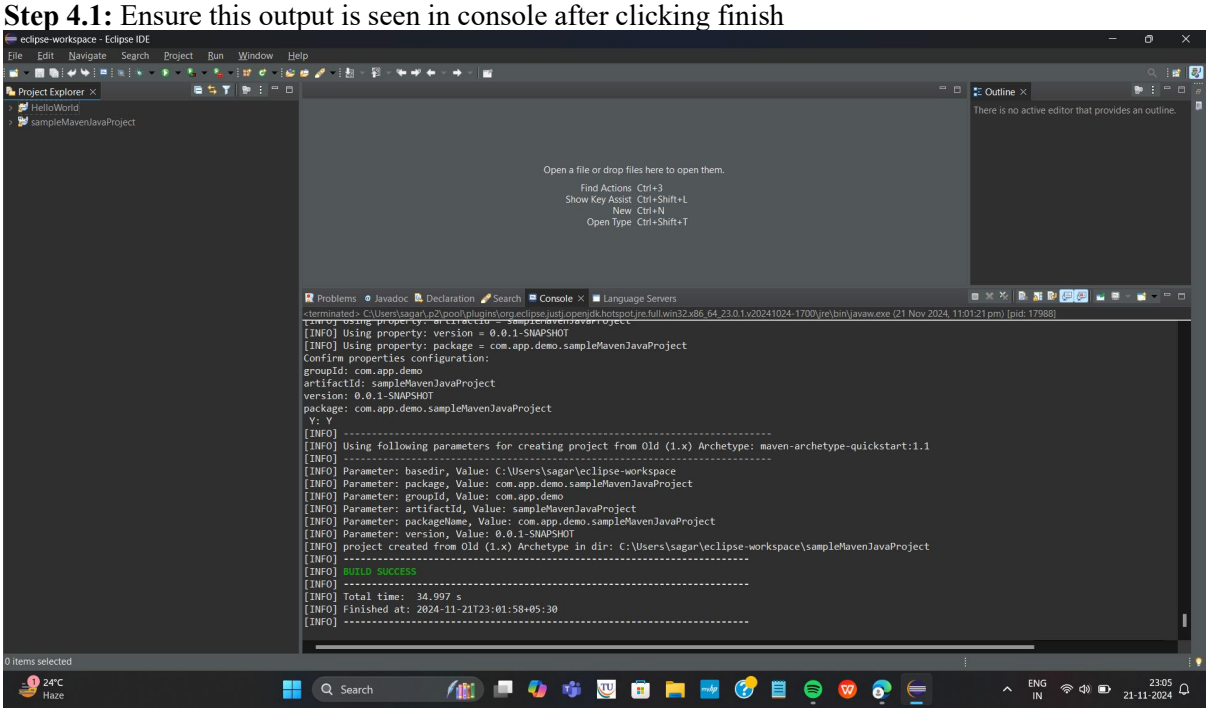
Step 2:
Choose the `quickstart` archtype under `org.apache.maven.archetypes.`



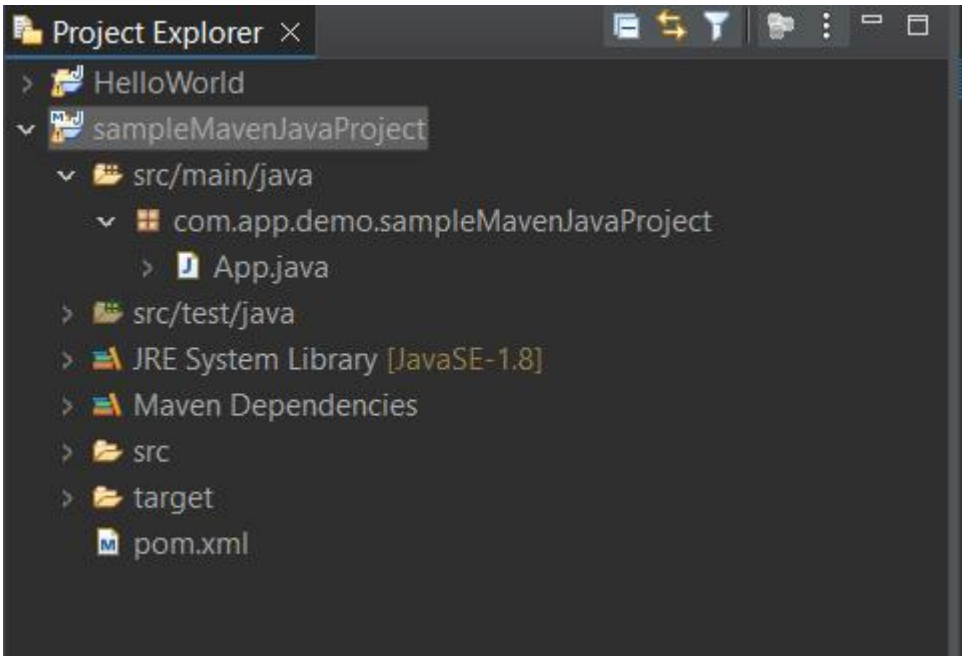
Step 3: Set the Group ID and Artifact ID (the unique identifier for your project).



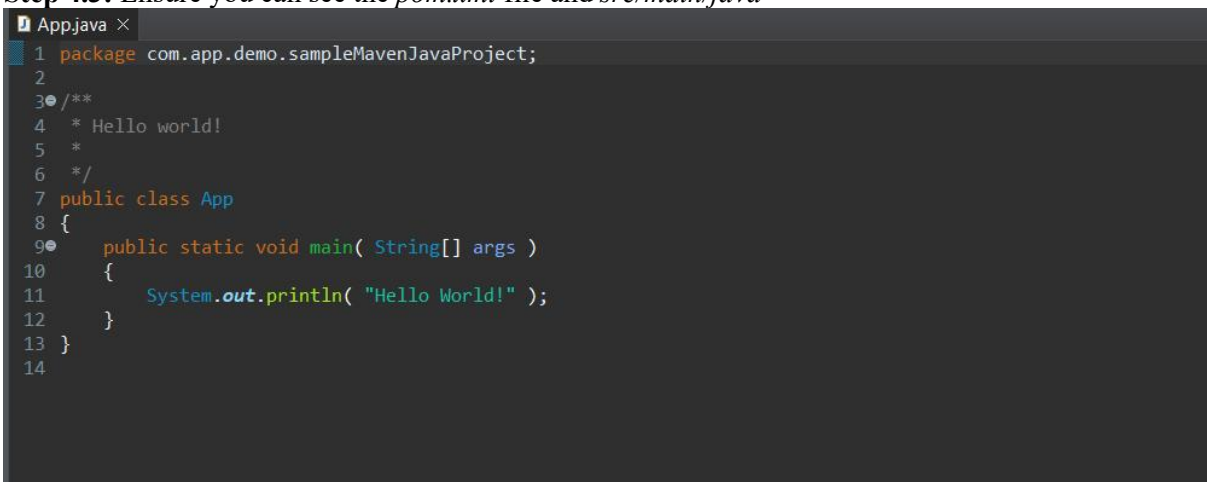
Step 4.1: Ensure this output is seen in console after clicking finish



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

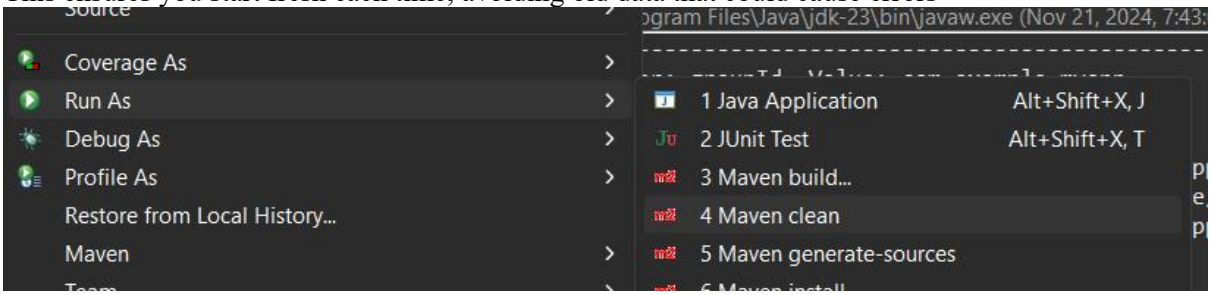


Step 4.3: Ensure you can see the *pom.xml* file and *src/main/java*

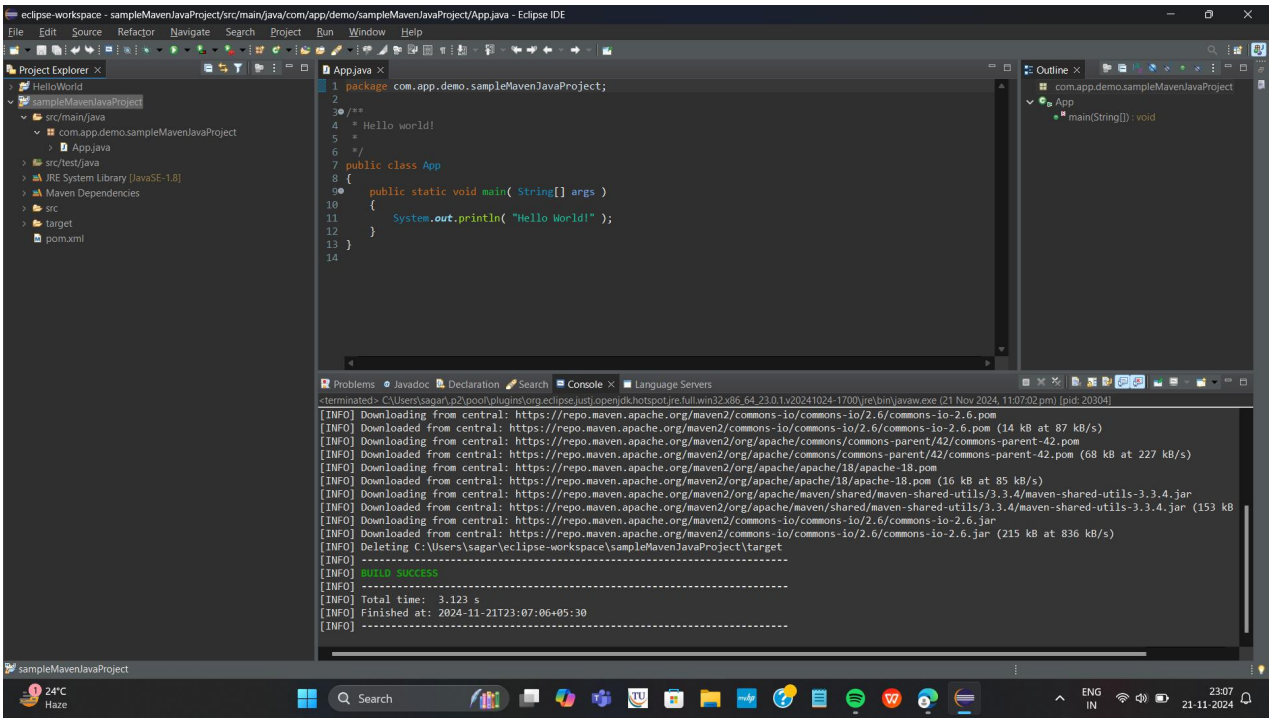


Running Essential Maven Goals

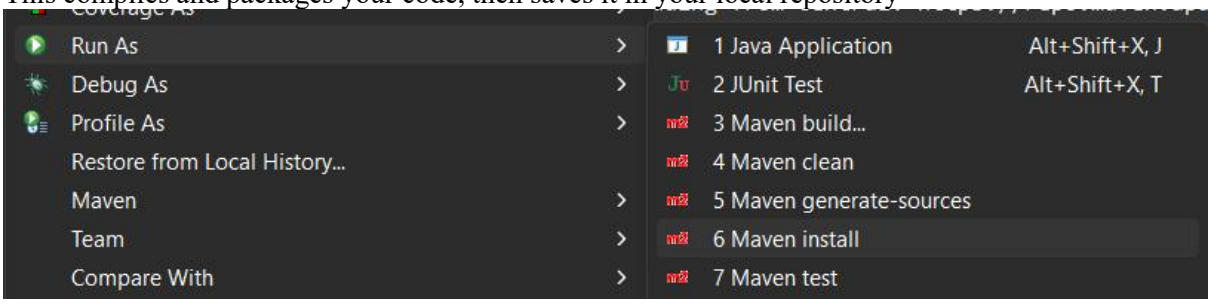
Step 1.1: Right click on folder *my-app* and select *Run As>Maven clean*
This ensures you start fresh each time, avoiding old data that could cause errors



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

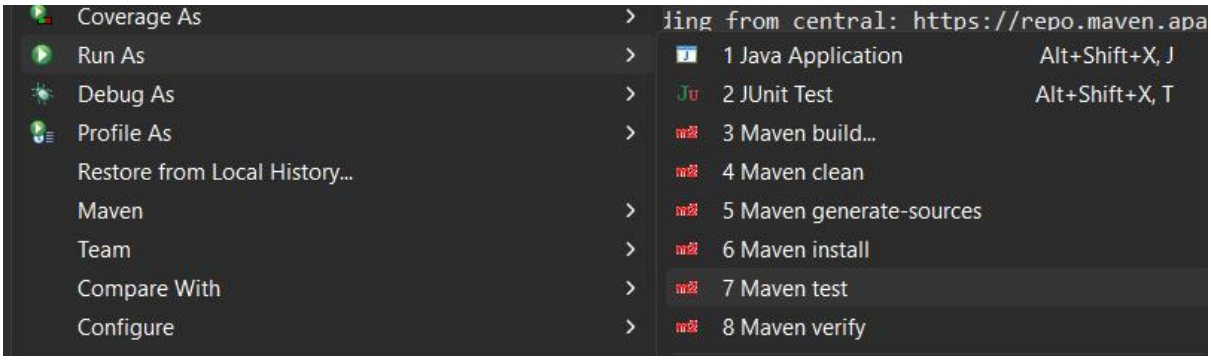


Step 2.1: Right click on project folder and select *Run As>Maven Install*
This compiles and packages your code, then saves it in your local repository

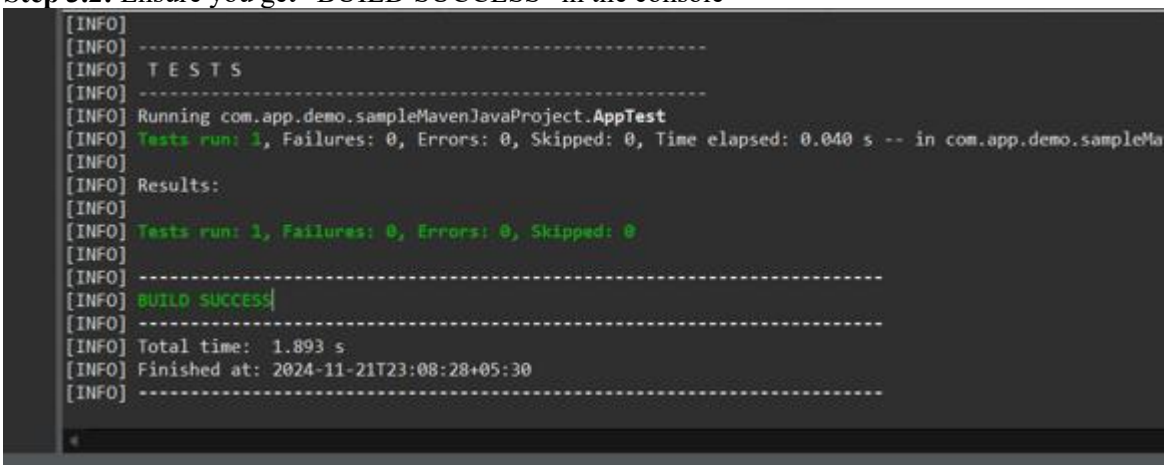


Step 2.2: Ensure you get "BUILD SUCCESS" in the console

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

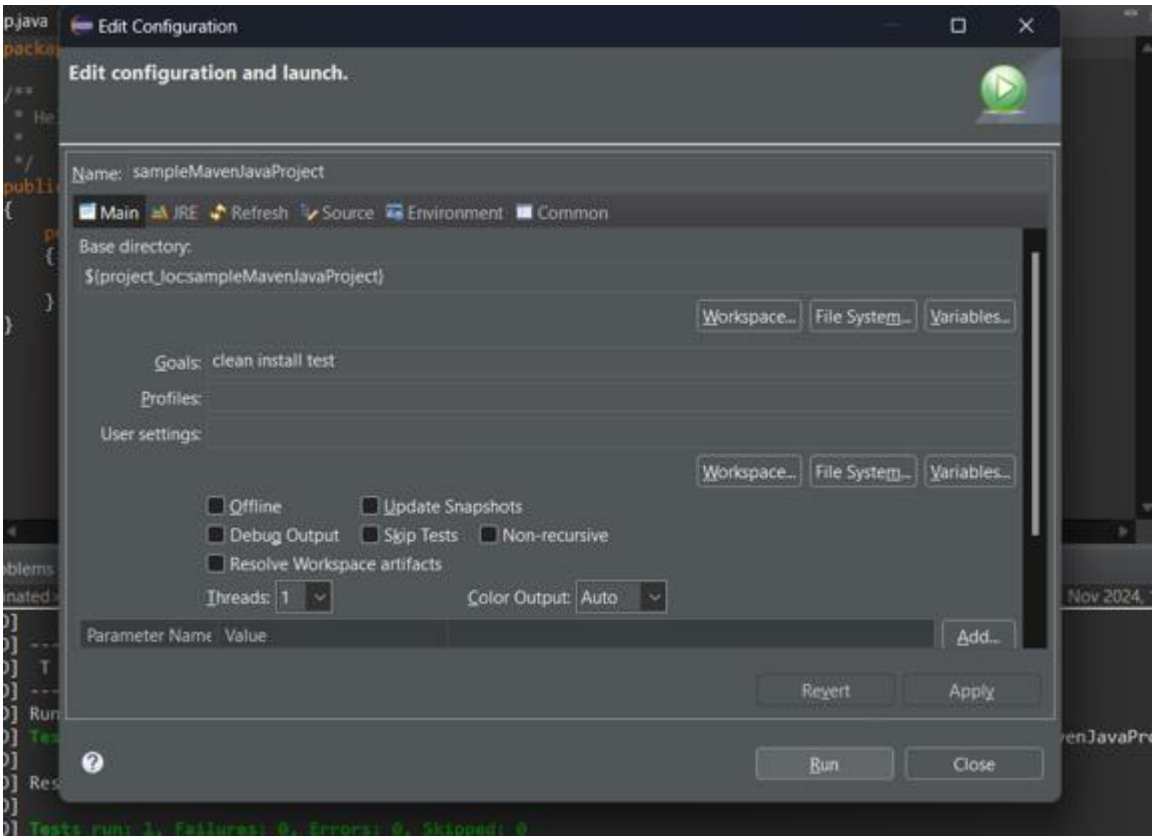


Step 3.2: Ensure you get "BUILD SUCCESS" in the console



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.

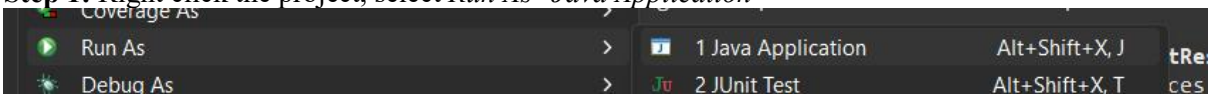


Step 4.2: Ensure you get "BUILD SUCCESS" in the console

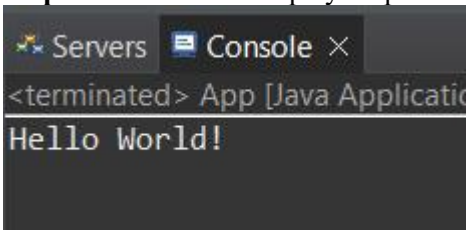
```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.067 s
[INFO] Finished at: 2024-11-21T20:03:38+05:30
[INFO] ------------------------------------------------------------------------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

Running the Project to View Output

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

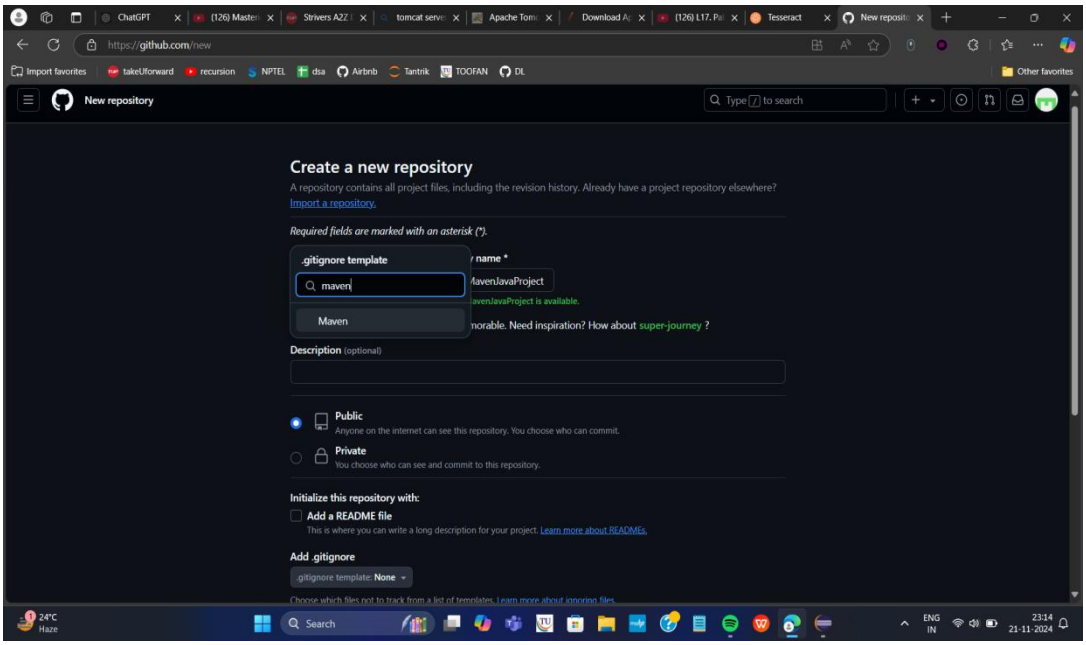


Step 2: Console will display output



Pushing the Project to GitHub

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



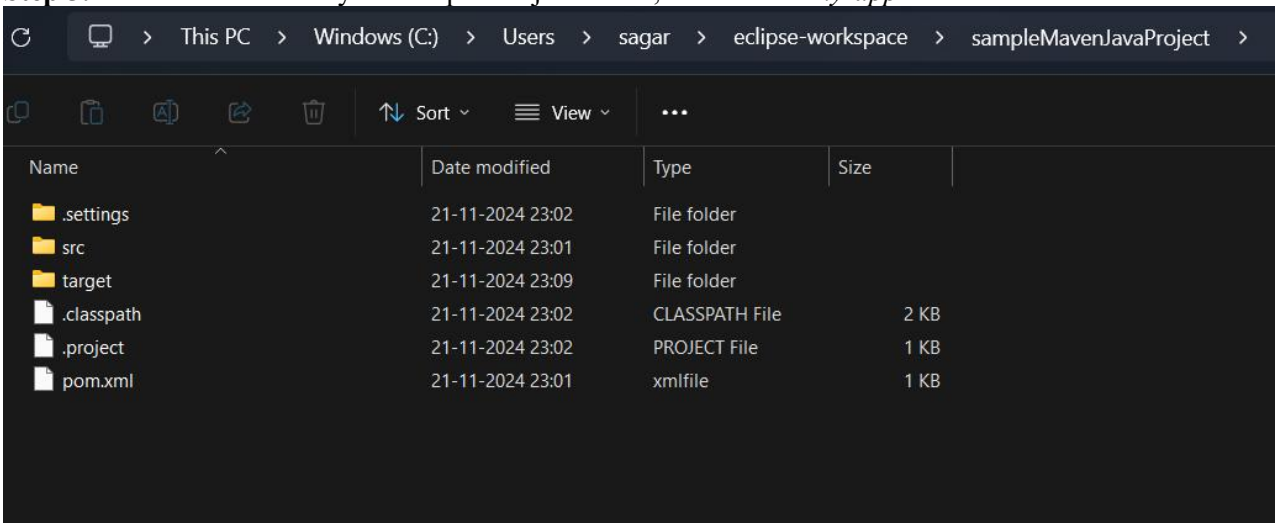
Step 3: Copy the HTTP link from GitHub to link your local project and clone it using git bash terminal

```
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.
```

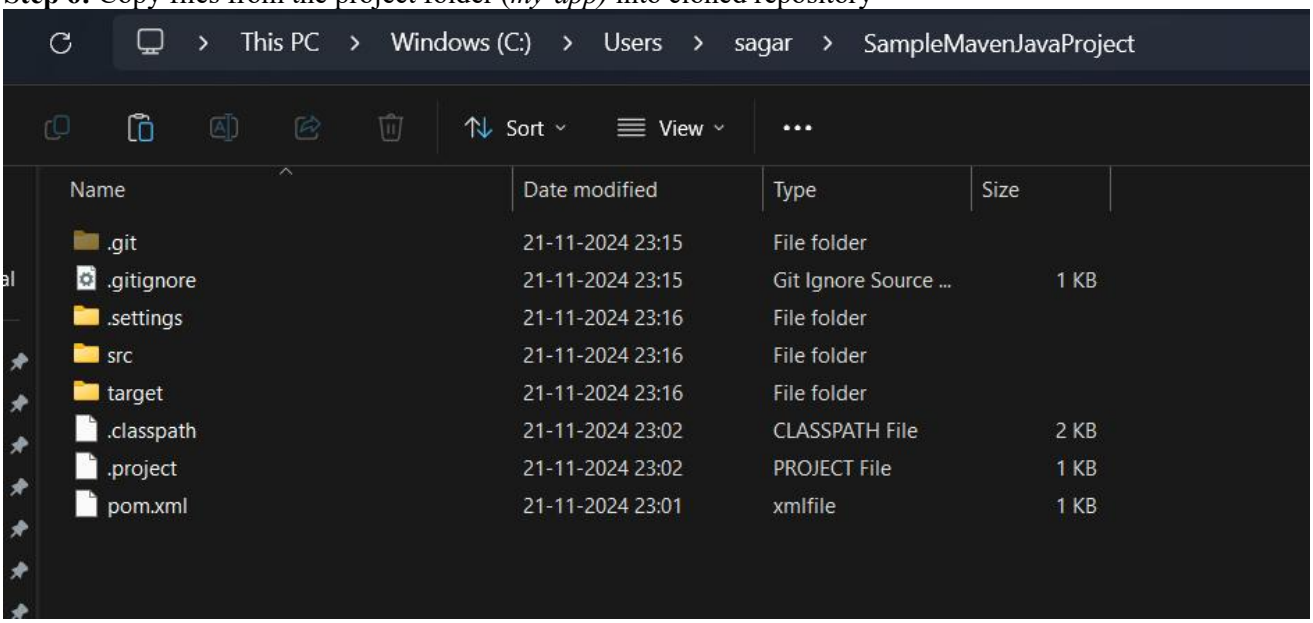
Step 4: Switch to the cloned folder

```
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*



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



Step 7:

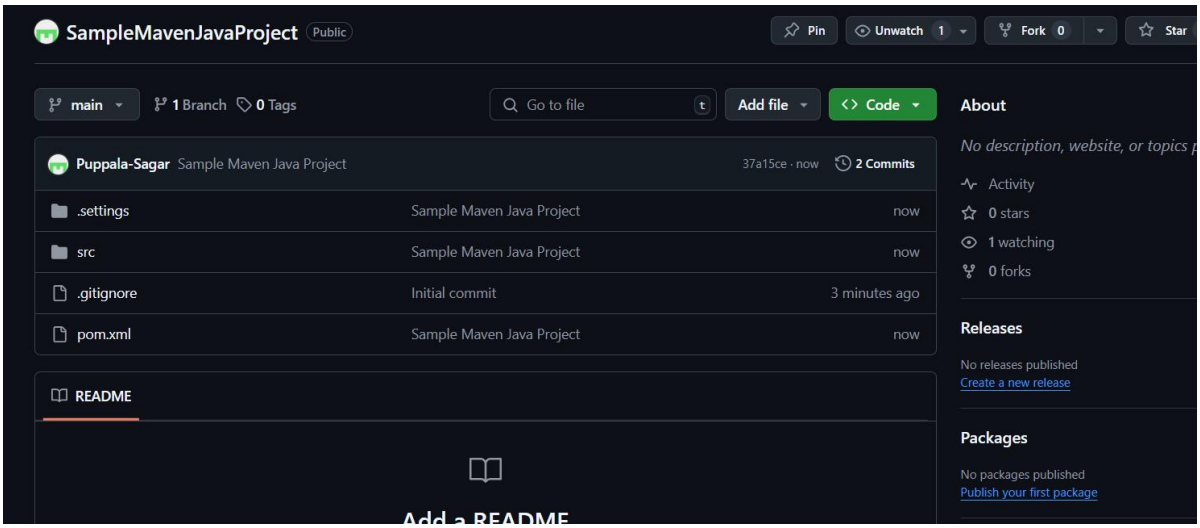
Stage the changes `git add .`

Commit the changes `git commit -m "message"`

Push to your GitHub repo `git push`

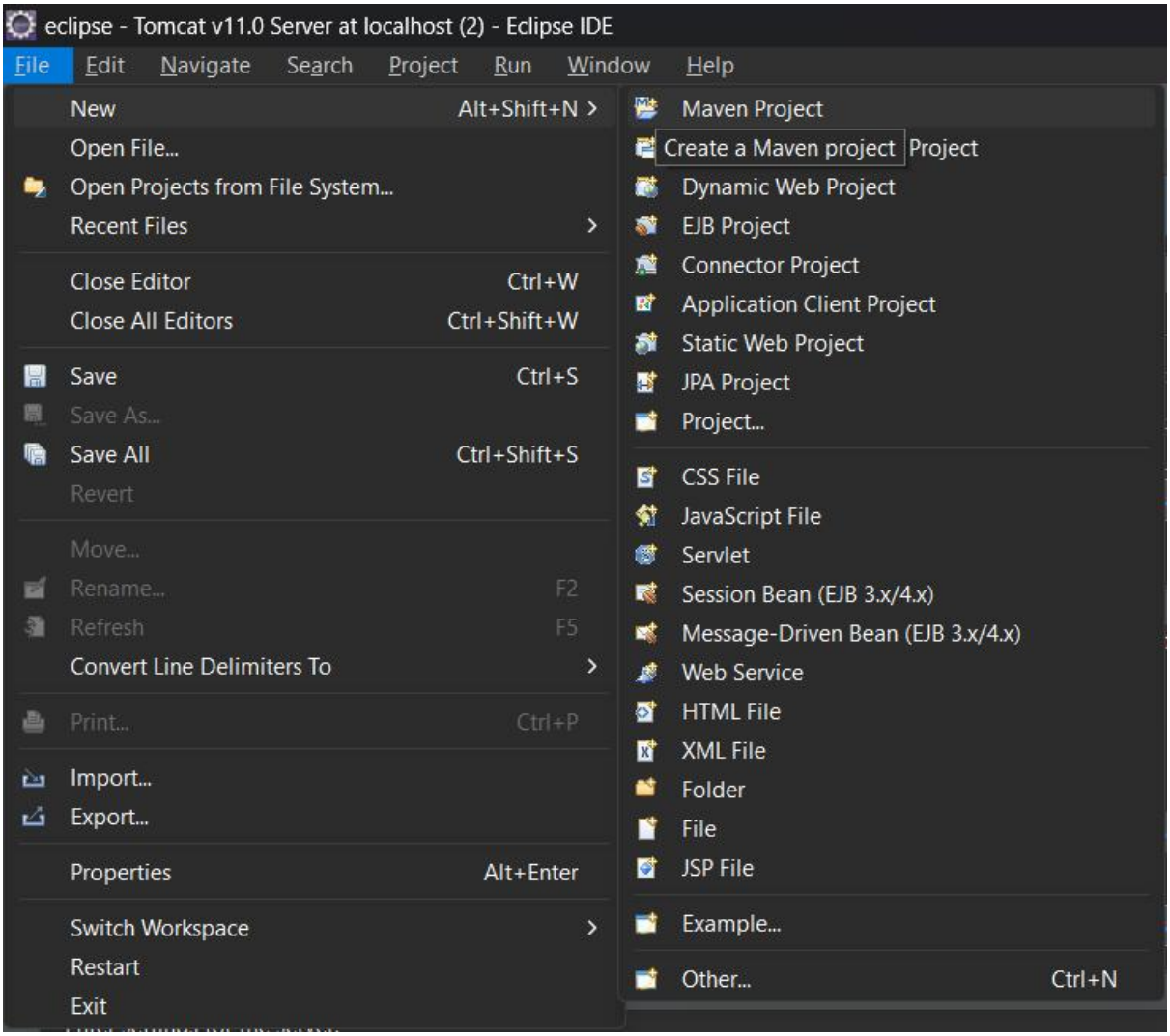
```
sagar@SAGARPUPPALA MINGW64 ~/SampleMavenJavaProject (main)
$ git add .

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.resources.prefs
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 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

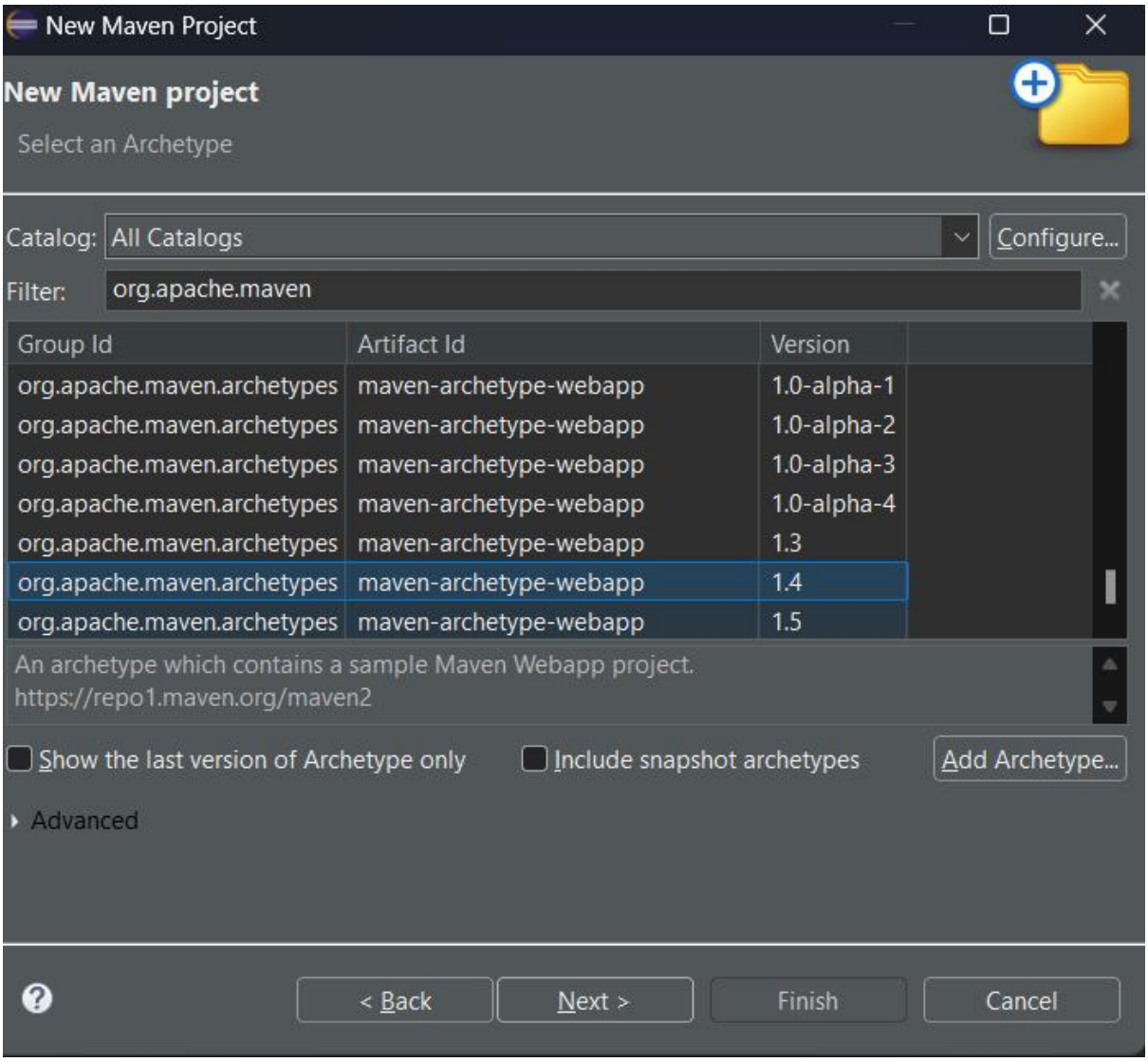


CREATING MAVEN WEB PROJECT USING ECLIPSE AND PUSH INTO TO GITHUB.

CREATE MAVEN WEB PROJECT



ChooseFilter



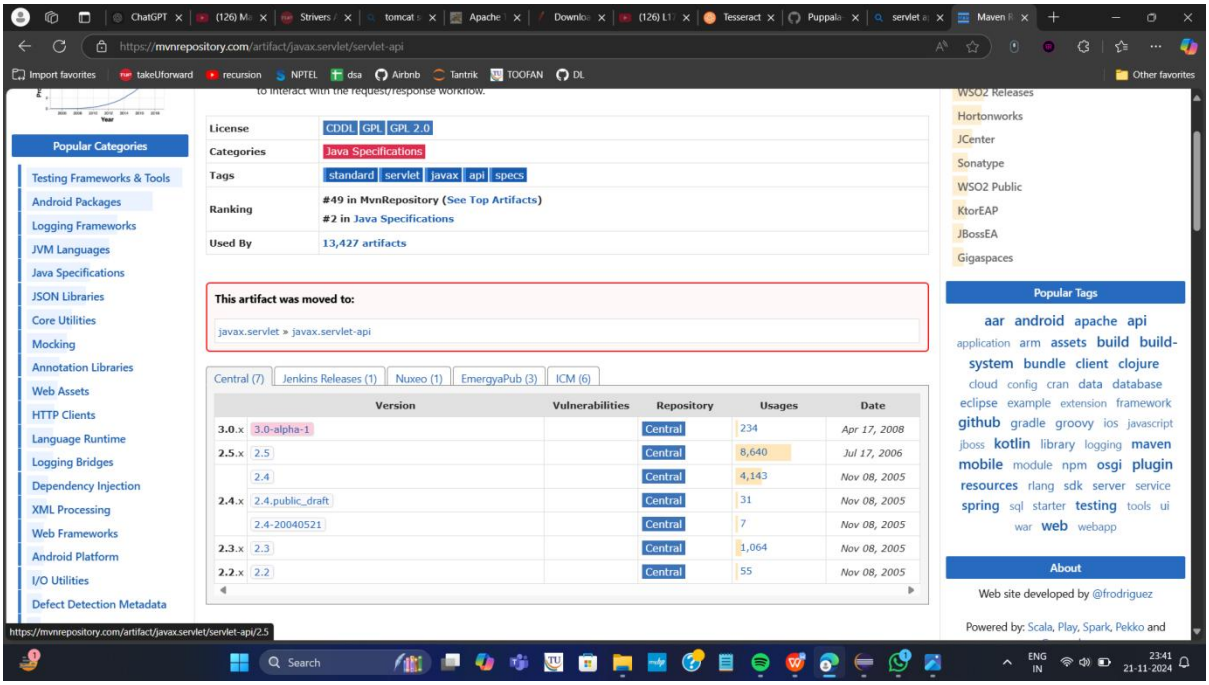
ADD GROUP ID AND ARTIFACT ID AND CLICK ON FINISH

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

App.java | index.jsp | SampleMavenWebProject/pom.xml

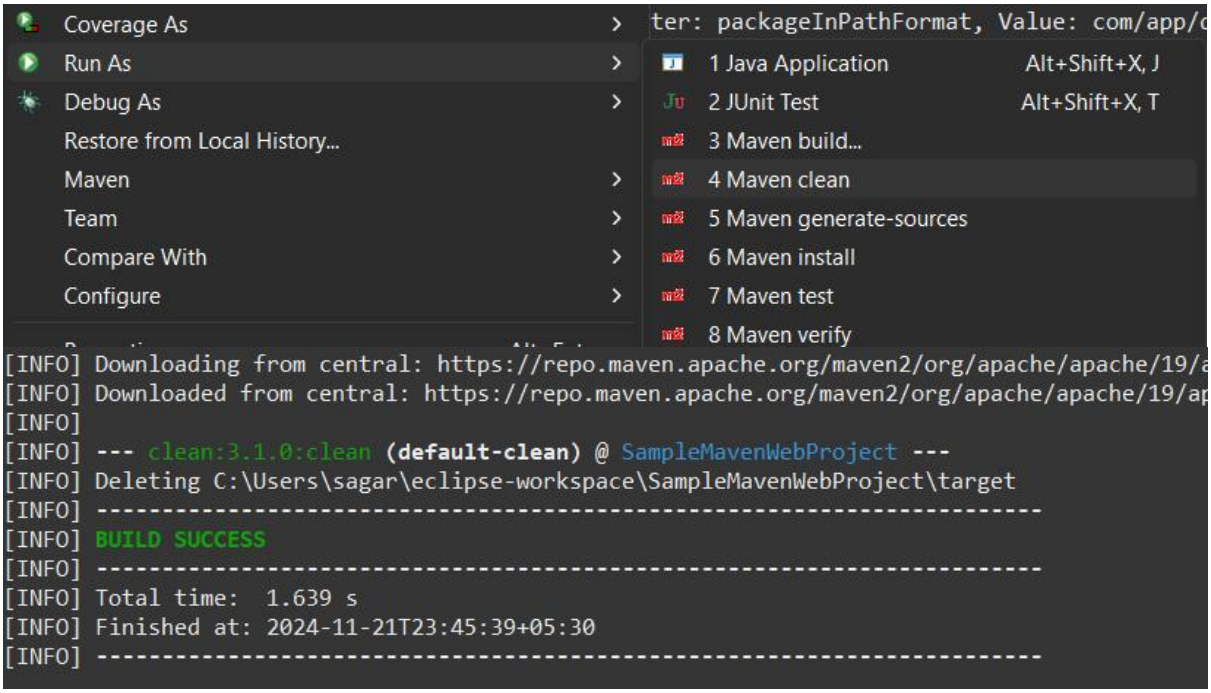
```xml
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.app.demo</groupId>
  <artifactId>SampleMavenWebProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>SampleMavenWebProject Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
```

ADD SERVELET DEPENDENCY

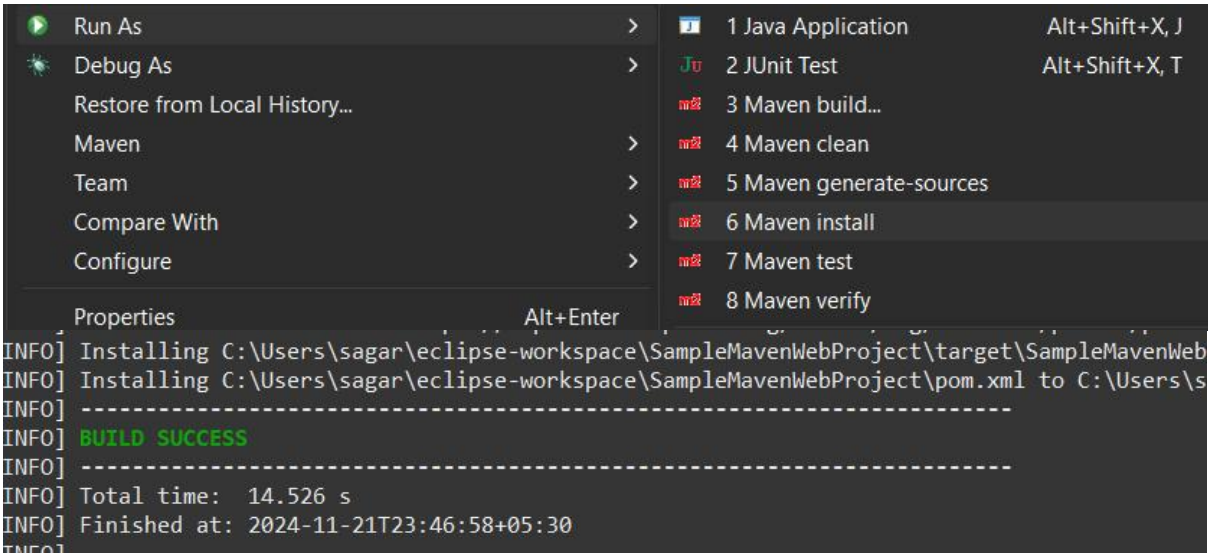


```
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/javax.servlet/servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
```

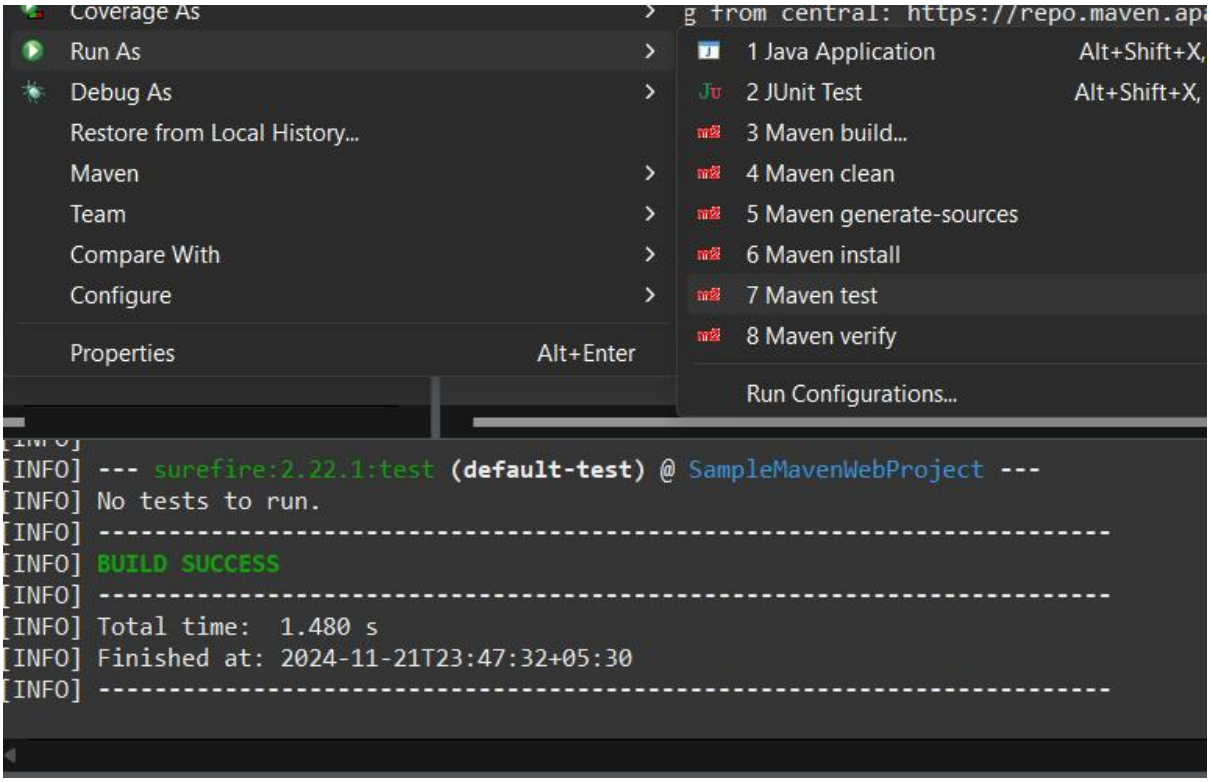
MAVEN CYCYLE
MAVEN CLEAN



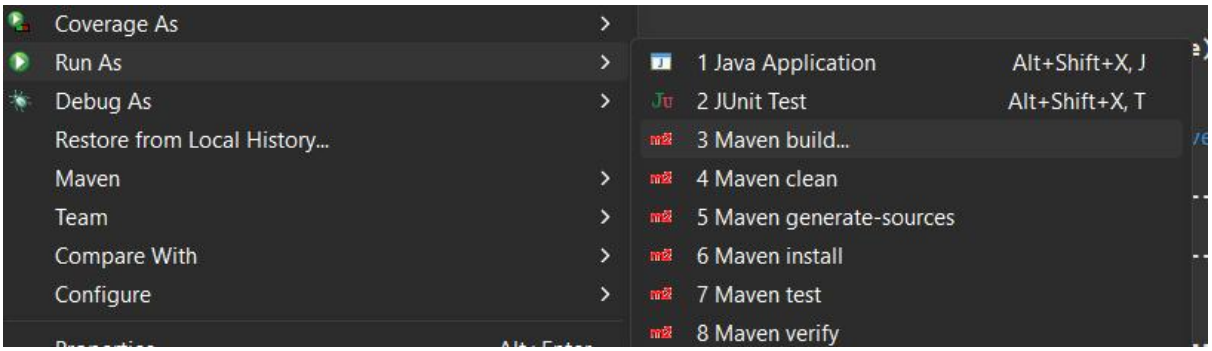
2 MAVEN INSTALL

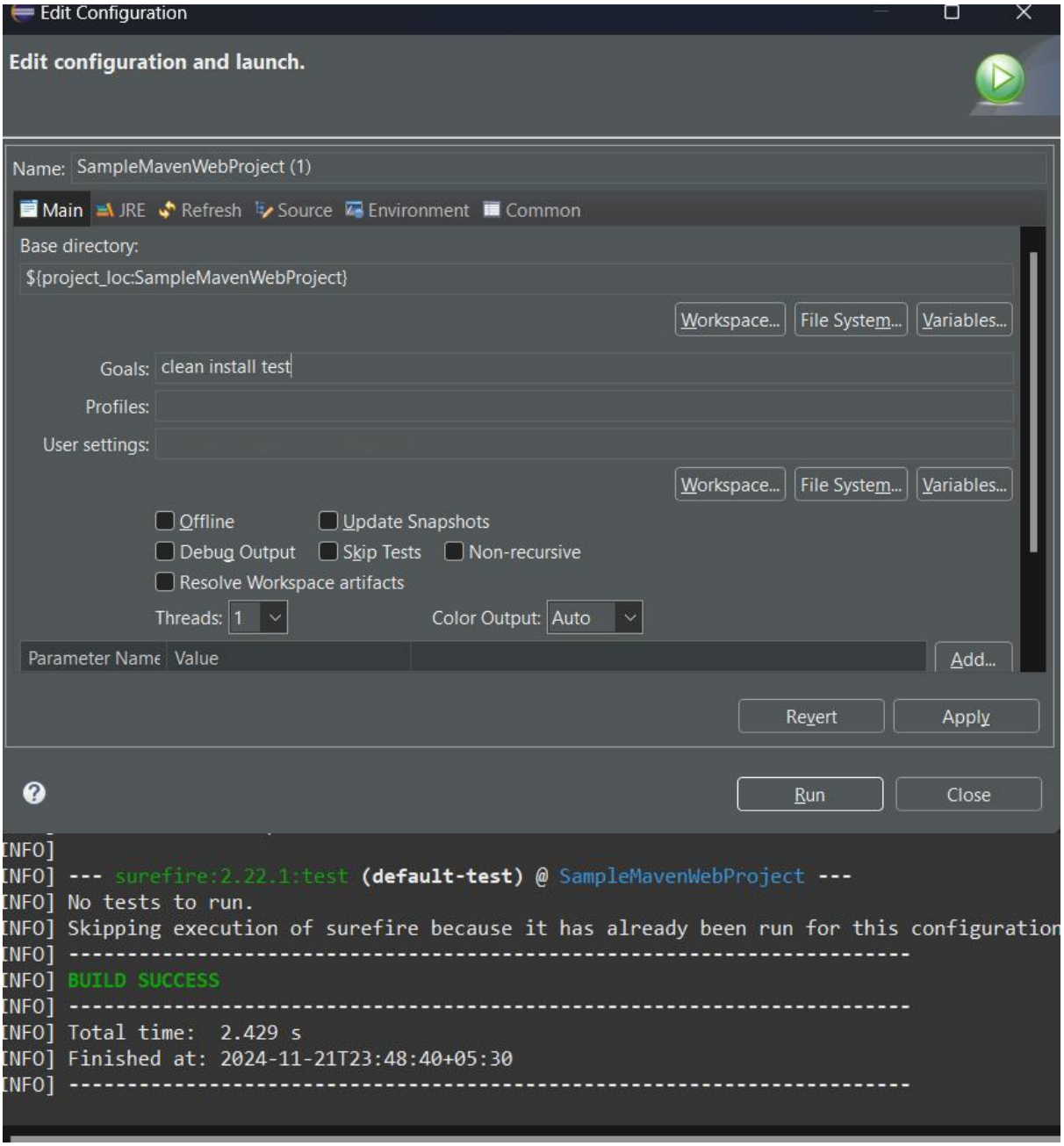


3 MAVEN TEST

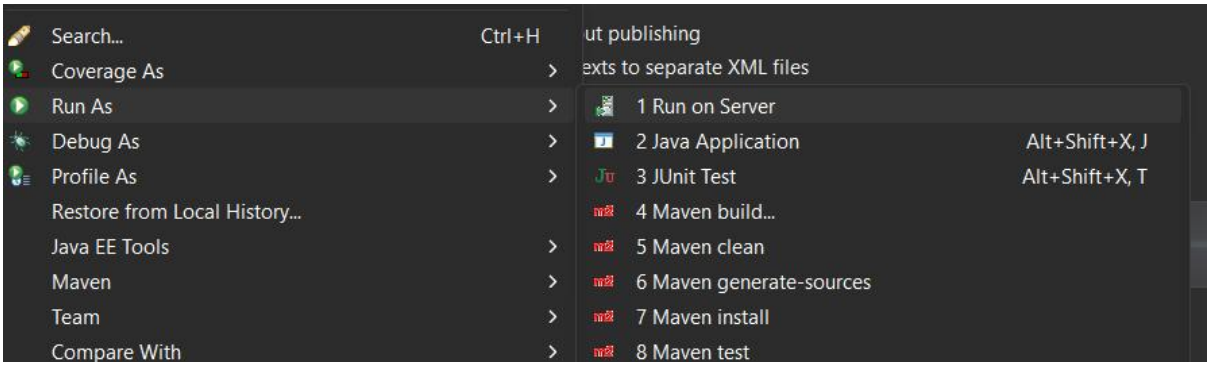

Coverage As
Run As
Debug As
Restore from Local History...
Maven
Team
Compare With
Configure
Properties Alt+Enter
1 Java Application Alt+Shift+X,
2 JUnit Test Alt+Shift+X,
3 Maven build...
4 Maven clean
5 Maven generate-sources
6 Maven install
7 Maven test
8 Maven verify
Run Configurations...
[INFO] --- 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

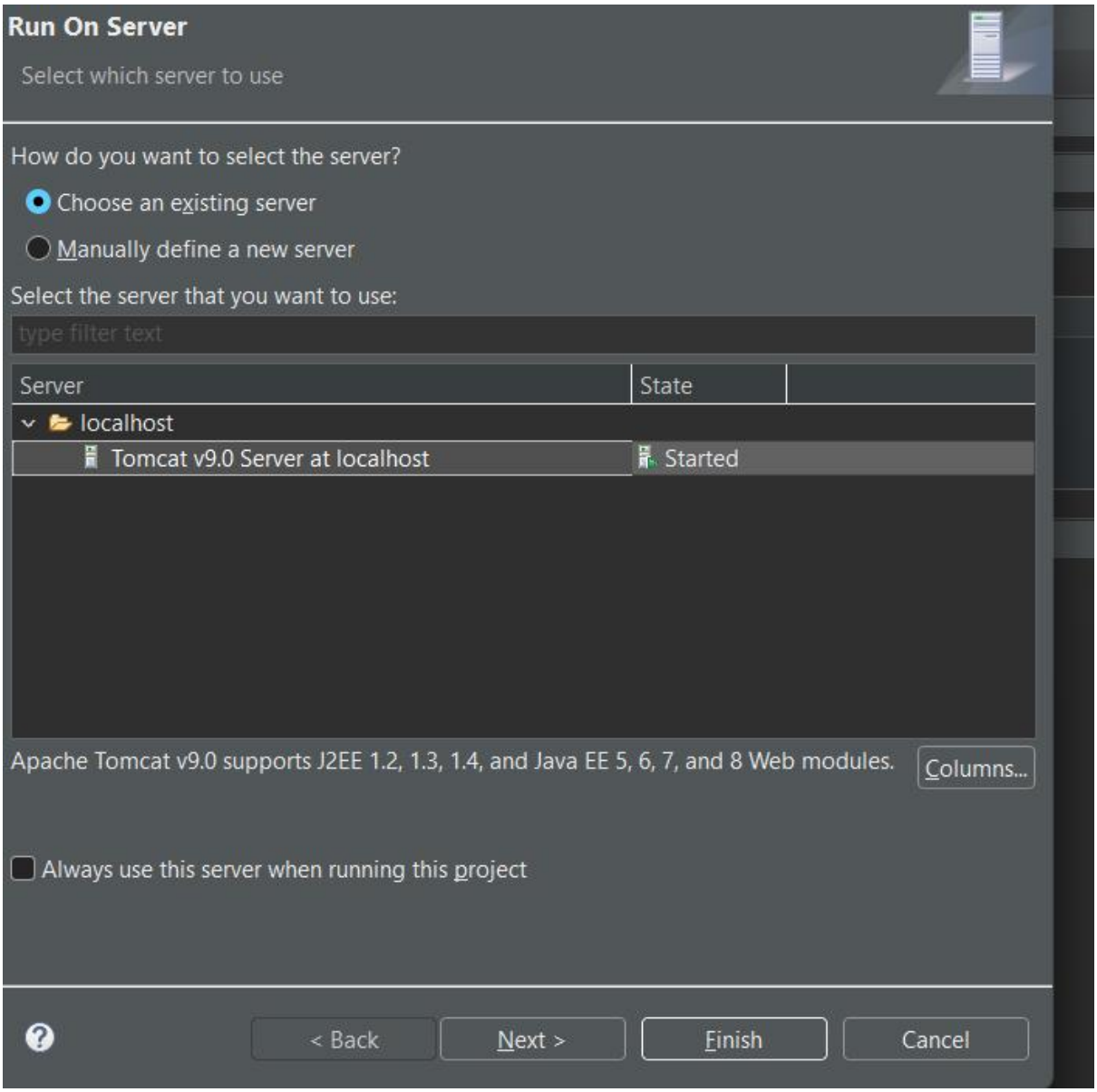

Coverage As
Run As
Debug As
Restore from Local History...
Maven
Team
Compare With
Configure
1 Java Application Alt+Shift+X, J
2 JUnit Test Alt+Shift+X, T
3 Maven build...
4 Maven clean
5 Maven generate-sources
6 Maven install
7 Maven test
8 Maven verify



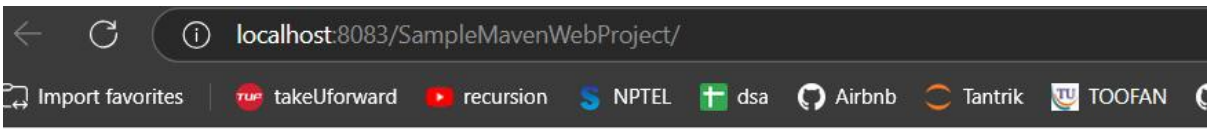
NOW RUN THE SERVER



CHOOSE TOMCAT AS SERVER



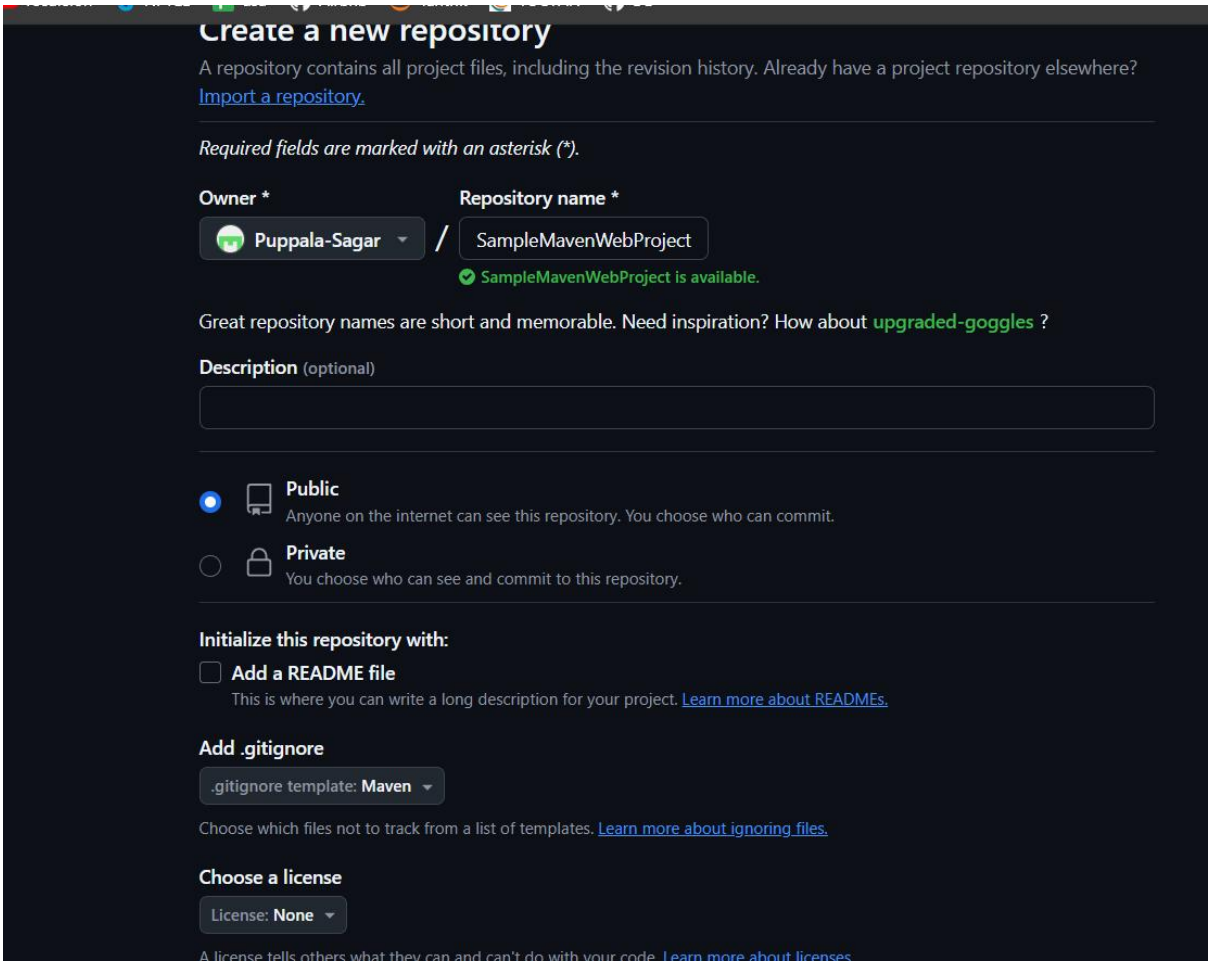
YOUR WEBPAGE WILL GET OPEN



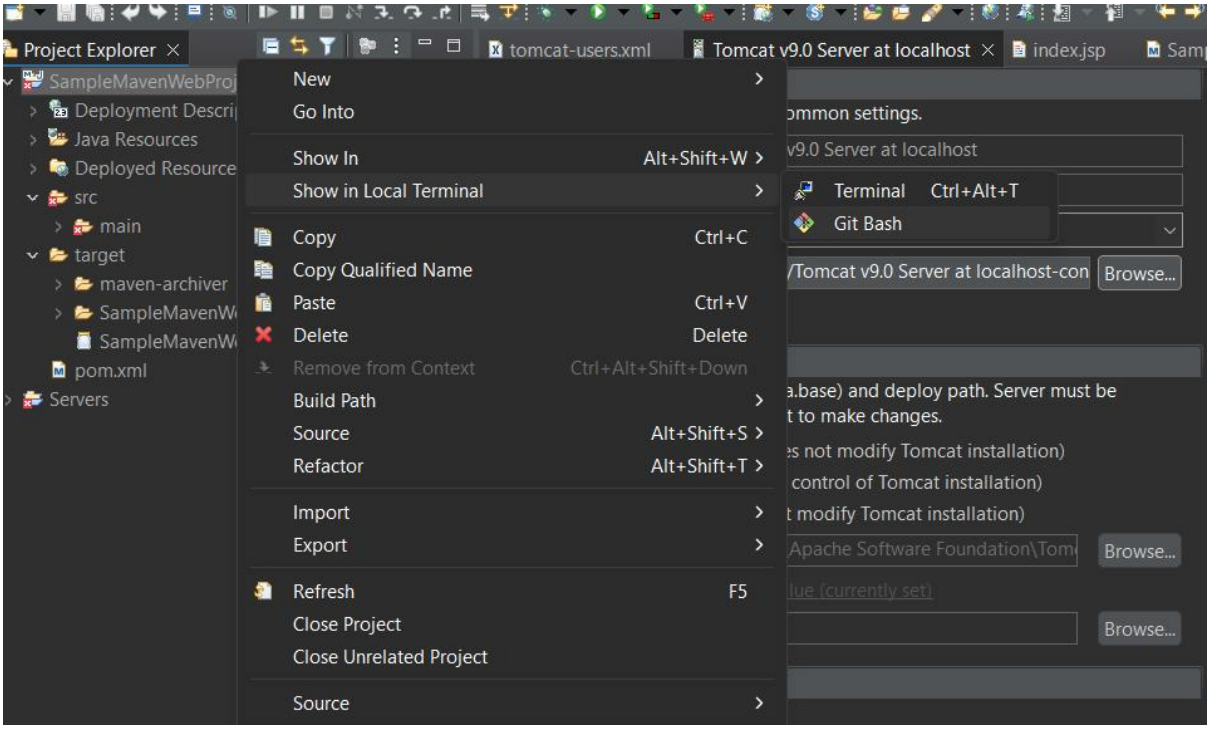
Hello World!

PUSHING INTO GITHUB

CREATE NEW GIT REPOISTORY



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 .
```

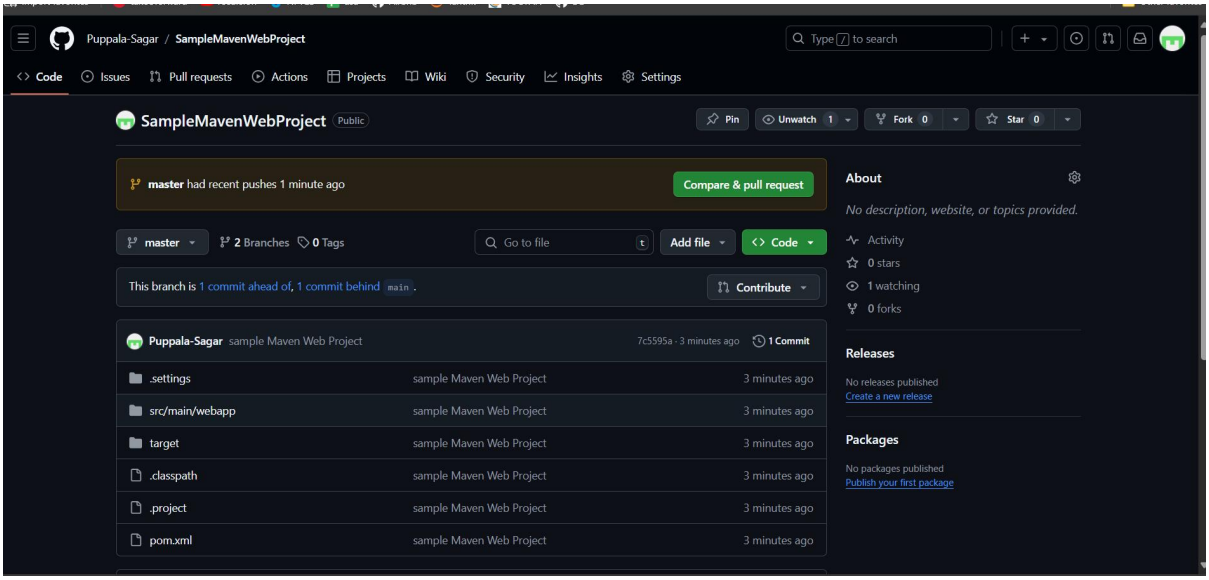
DO COMMIT

```
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

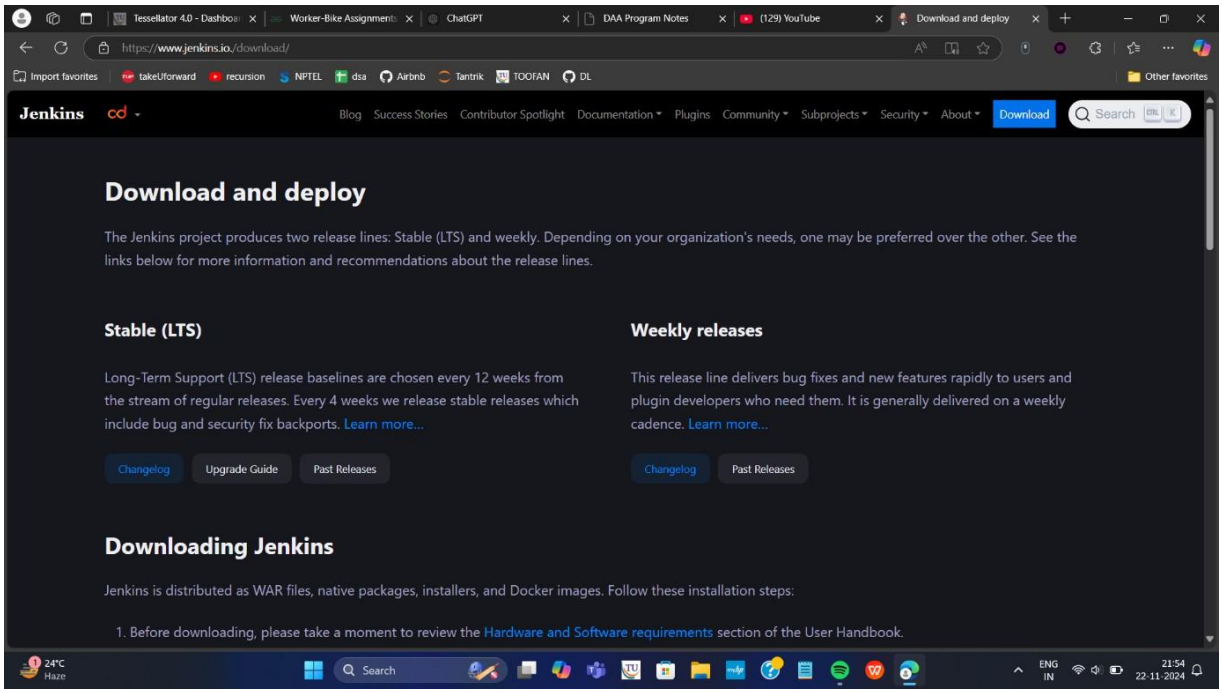
```
sagar@SAGARPUPPALA MINGW64 ~/eclipse-workspace/SampleMavenWebProject (master)
$ git push -u origin master
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 12 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (24/24), 4.62 KiB | 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

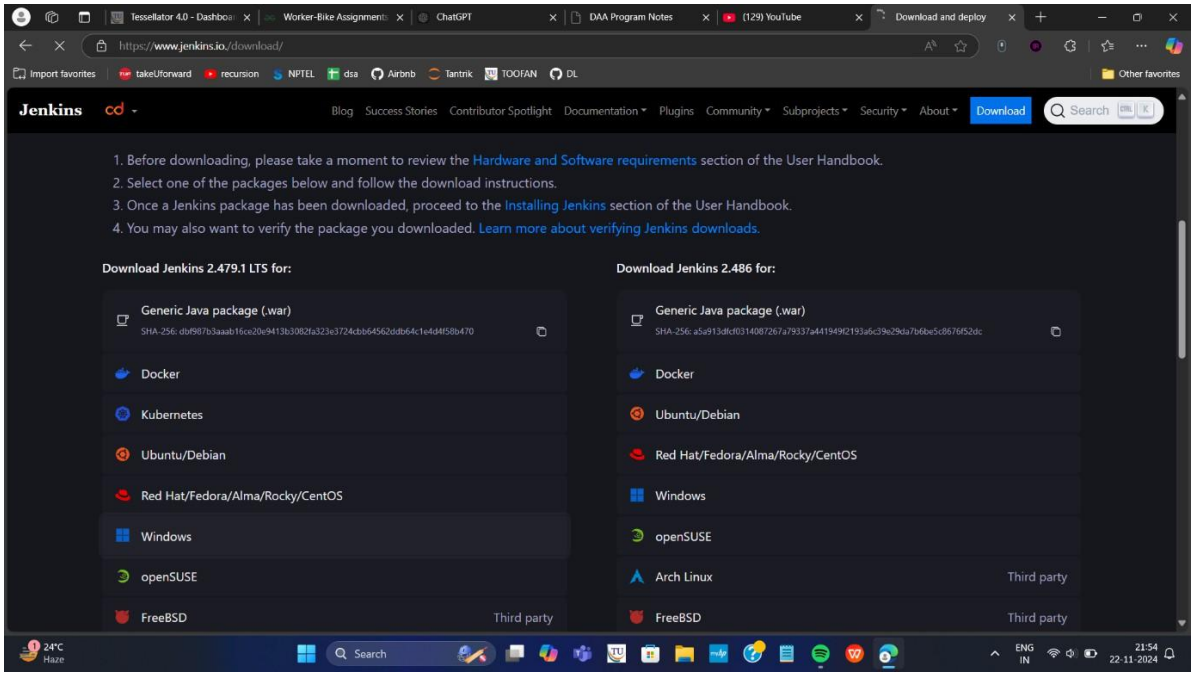


JENKINS INSTALLATION

GOTO JENKINS WEBISTE

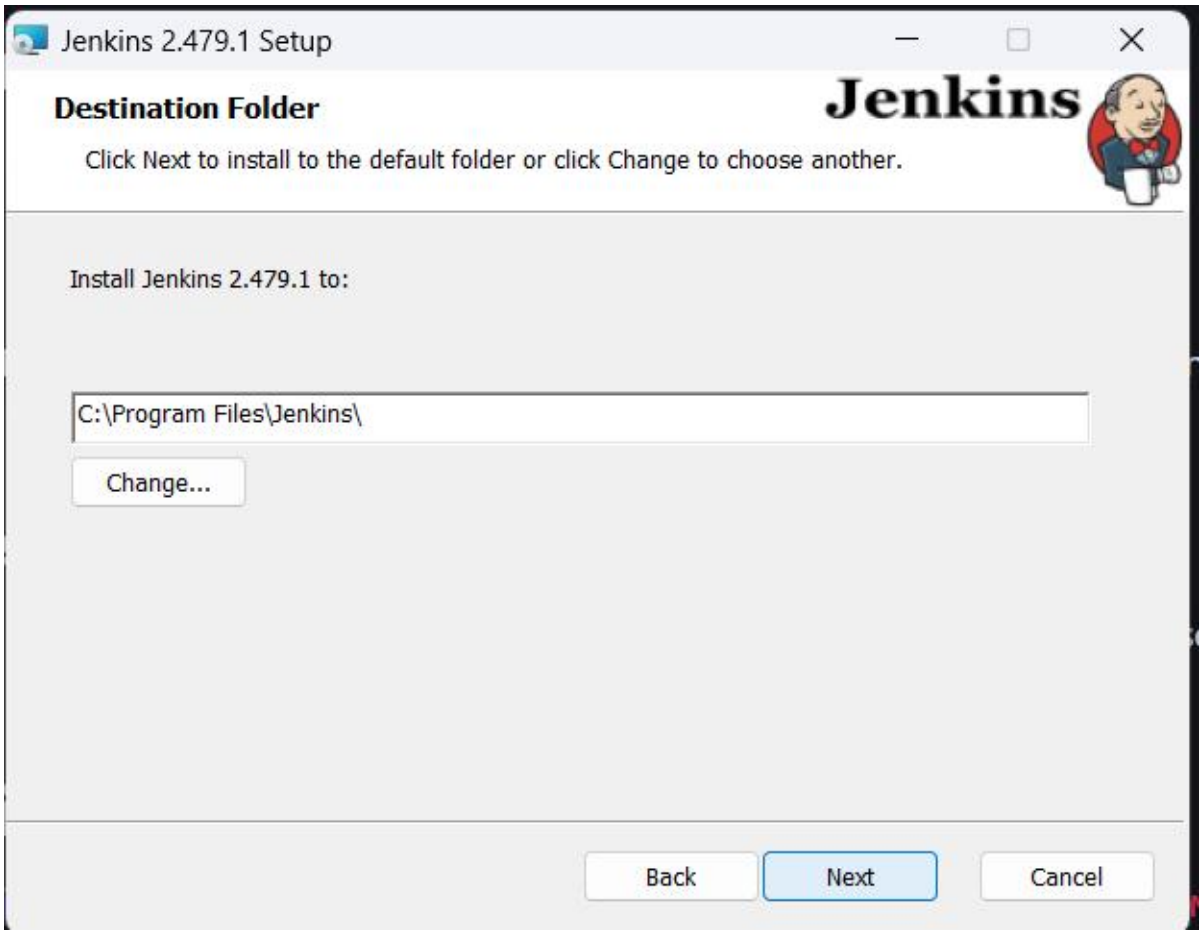


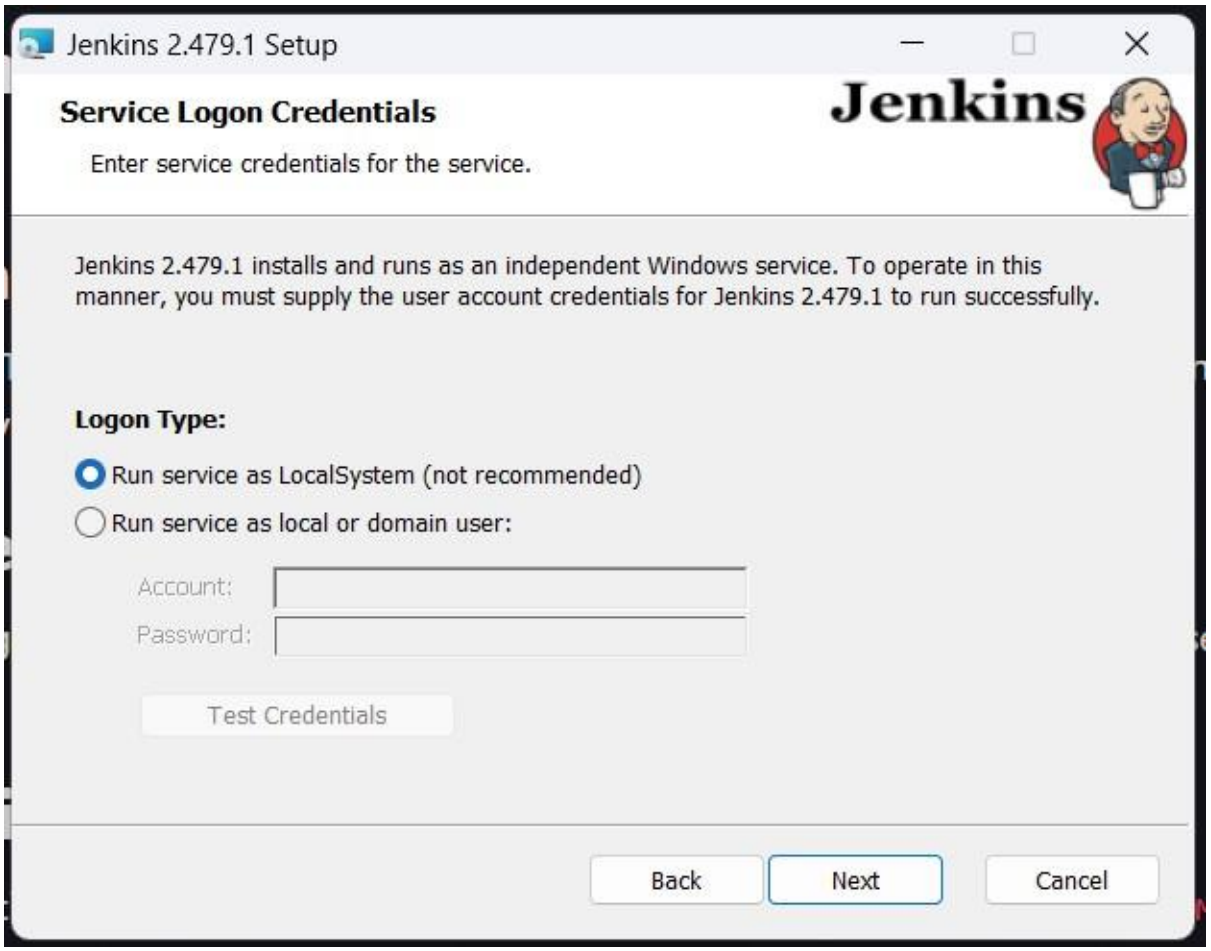
SELECT WINDOWS AND DONWLOAD EXE



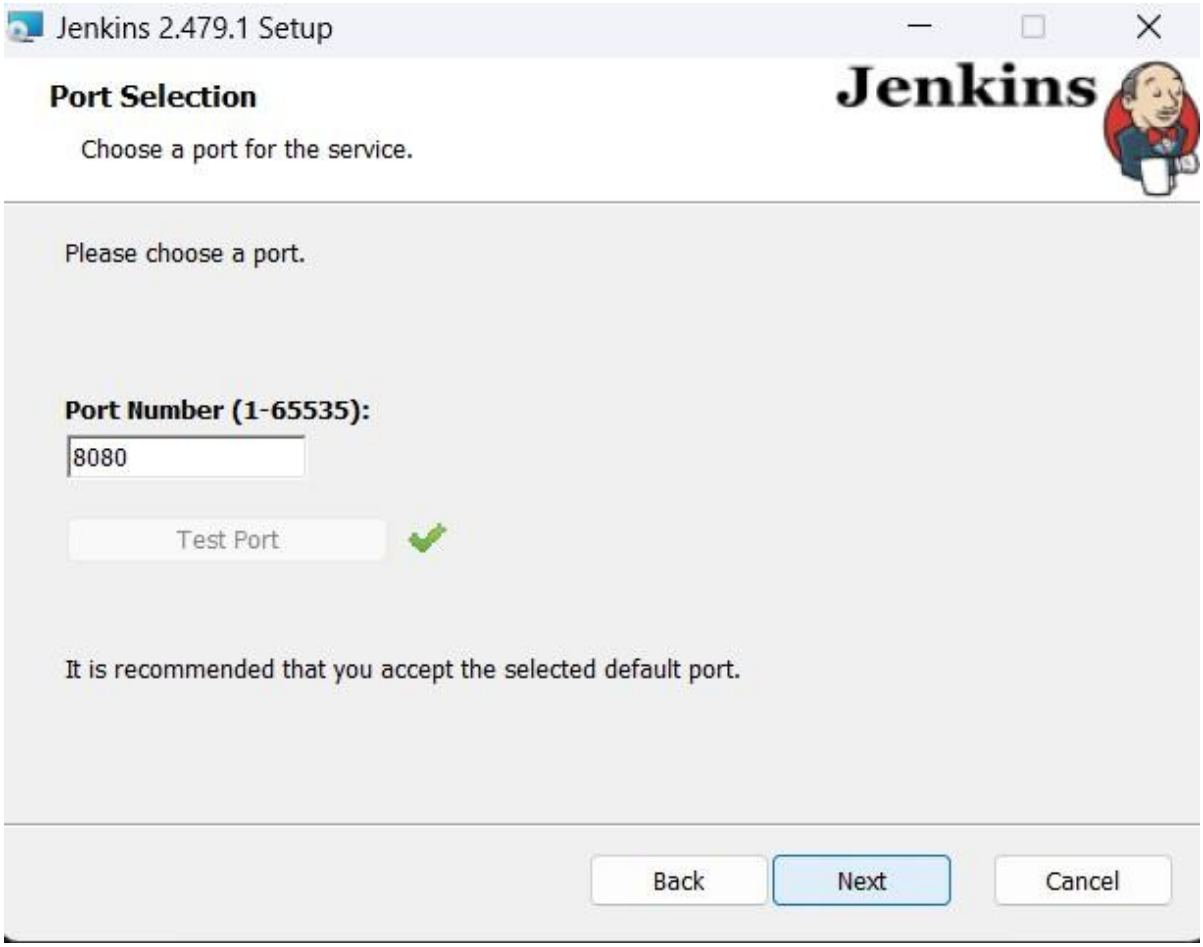
INSTALL EXE

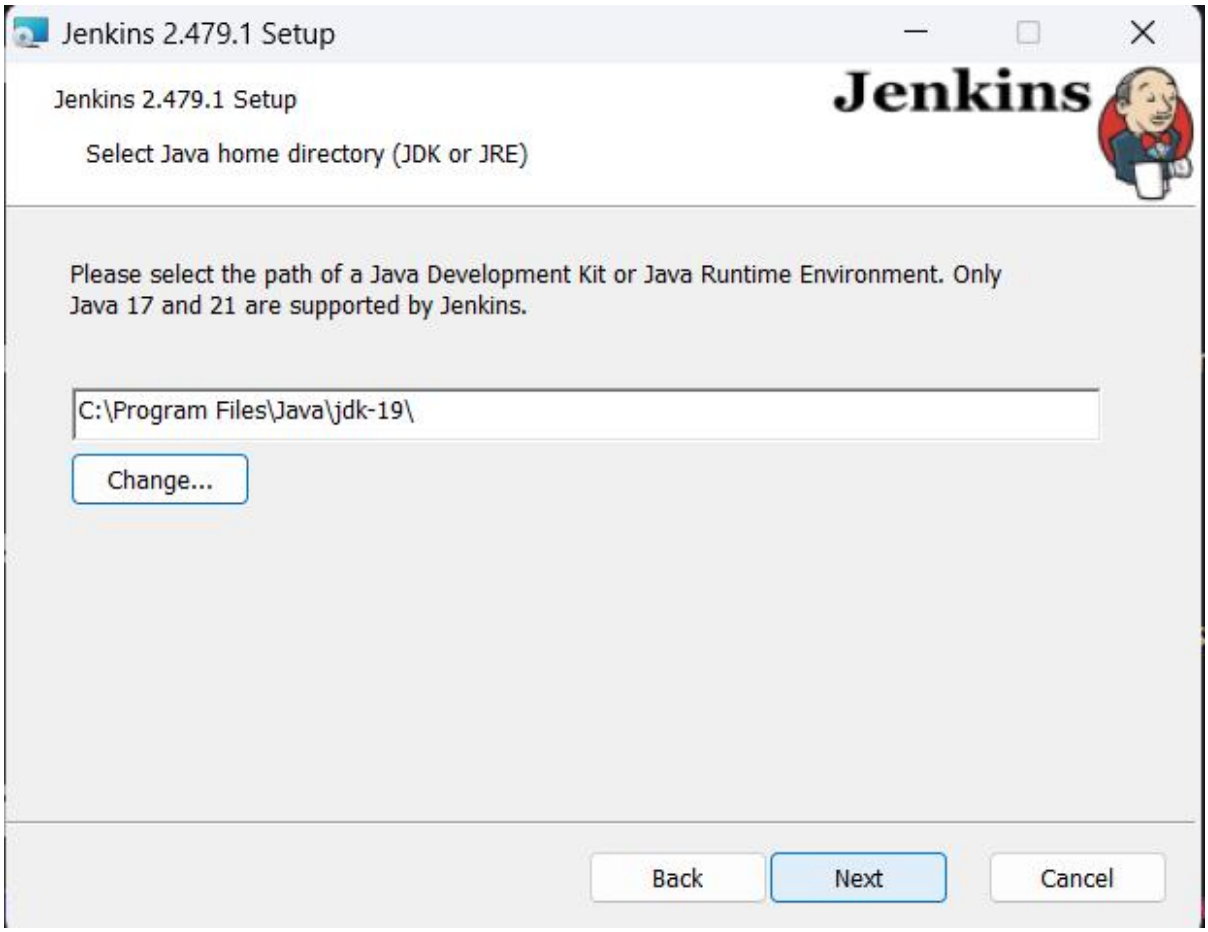

Jenkins 2.479.1 Setup
Welcome to the Jenkins 2.479.1 Setup Wizard
The Setup Wizard will install Jenkins 2.479.1 on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel


Jenkins 2.479.1 Setup
Destination Folder
Click Next to install to the default folder or click Change to choose another.
Jenkins
Install Jenkins 2.479.1 to:
C:\Program Files\Jenkins\
Change...
Back
Next
Cancel

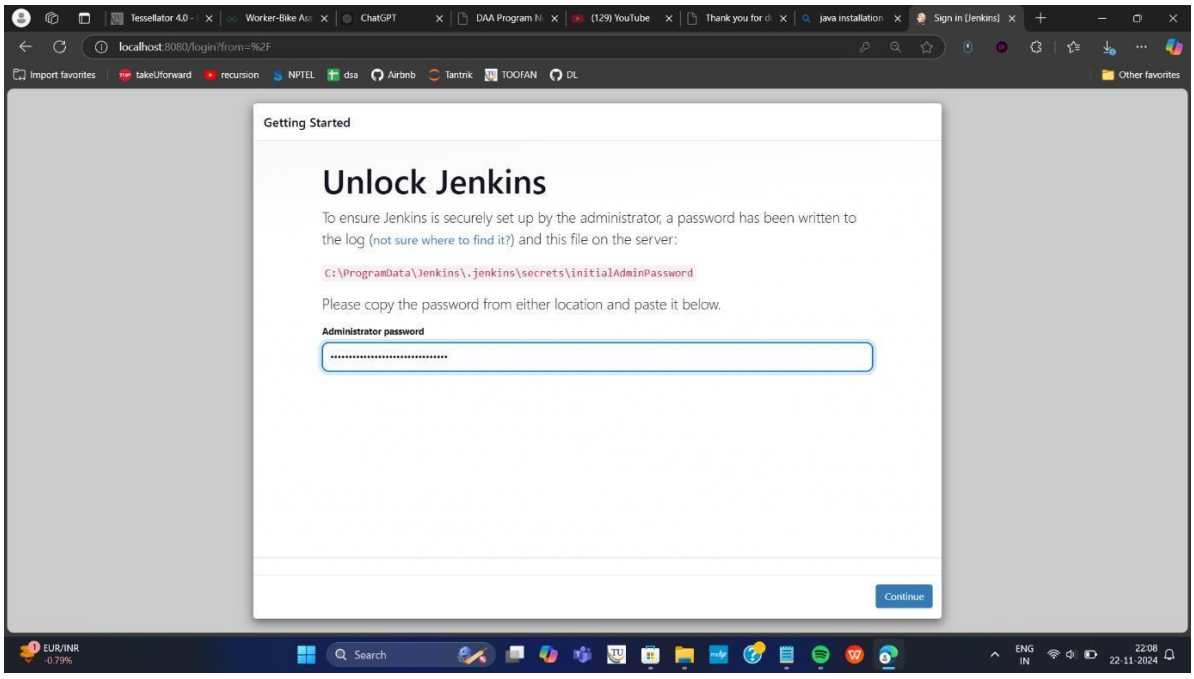

Jenkins 2.479.1 Setup
Service Logon Credentials
Enter service credentials for the service.
Jenkins
Jenkins 2.479.1 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.479.1 to run successfully.
Logon Type:
Run service as LocalSystem (not recommended)
Run service as local or domain user:
Account:
Password:
Test Credentials
Back
Next
Cancel

TEST PORT

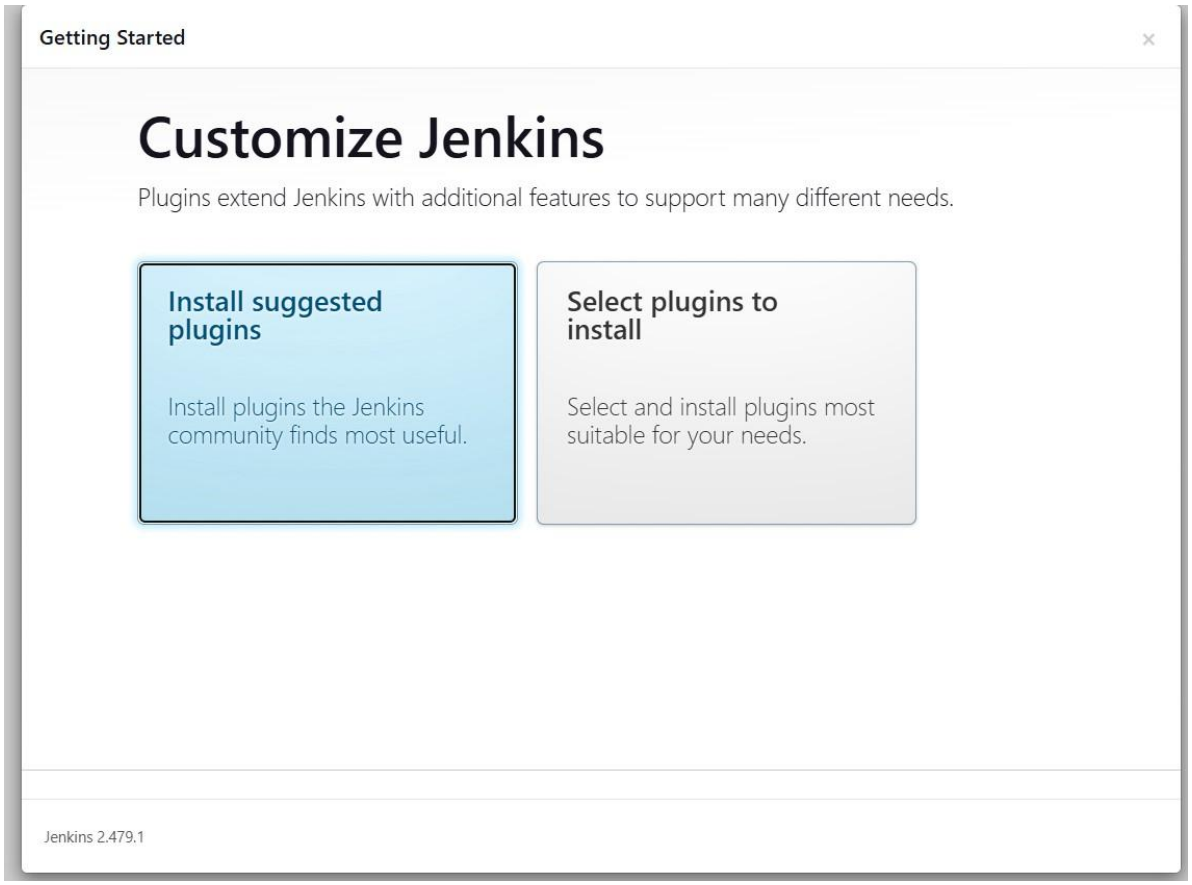

Jenkins 2.479.1 Setup
Port Selection
Choose a port for the service.
Jenkins
Please choose a port.
Port Number (1-65535):
8080
Test Port
It is recommended that you accept the selected default port.
Back
Next
Cancel



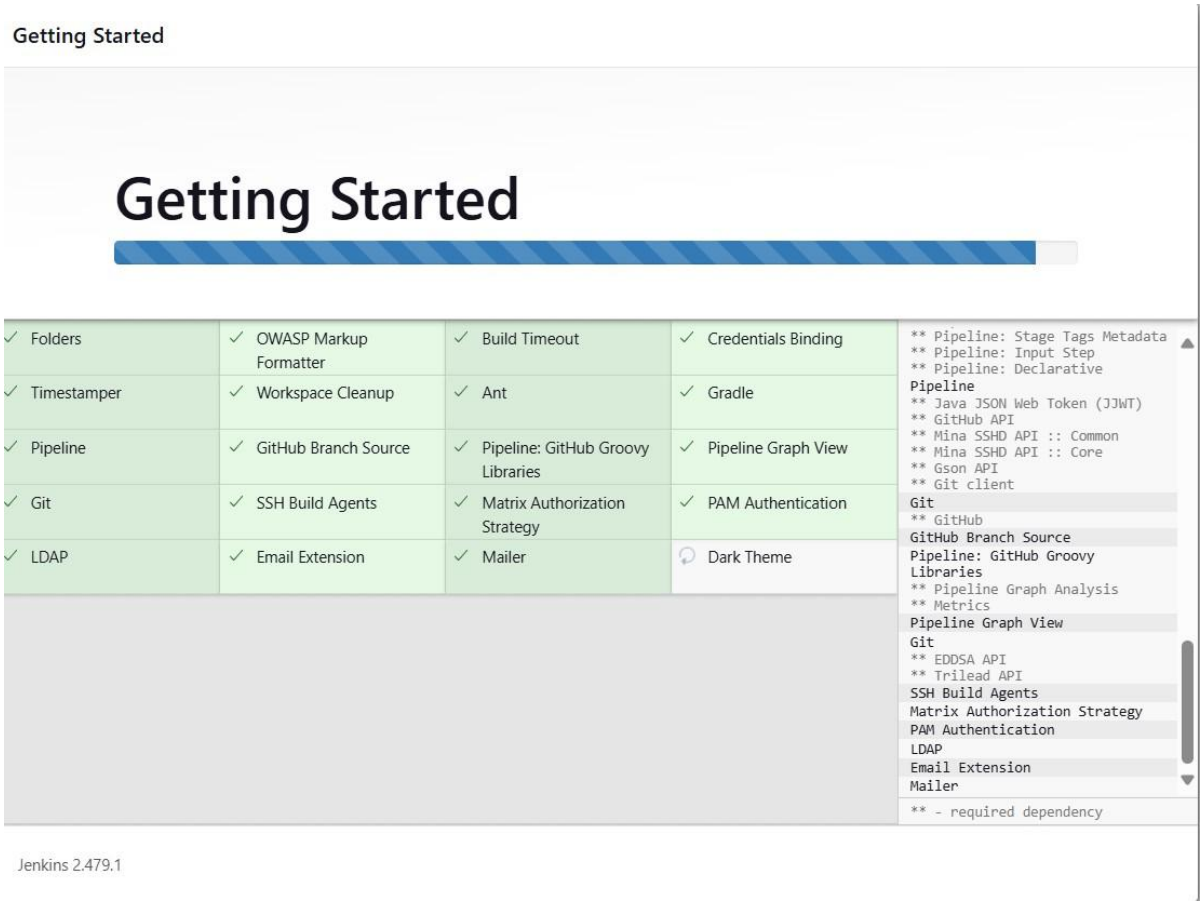
ADD ADMINISTRATOR PASSWORD



CLICK ON INSTALL SUGGESTED PLUGINS



WAIT FOR INSTALLATION



Getting Started

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

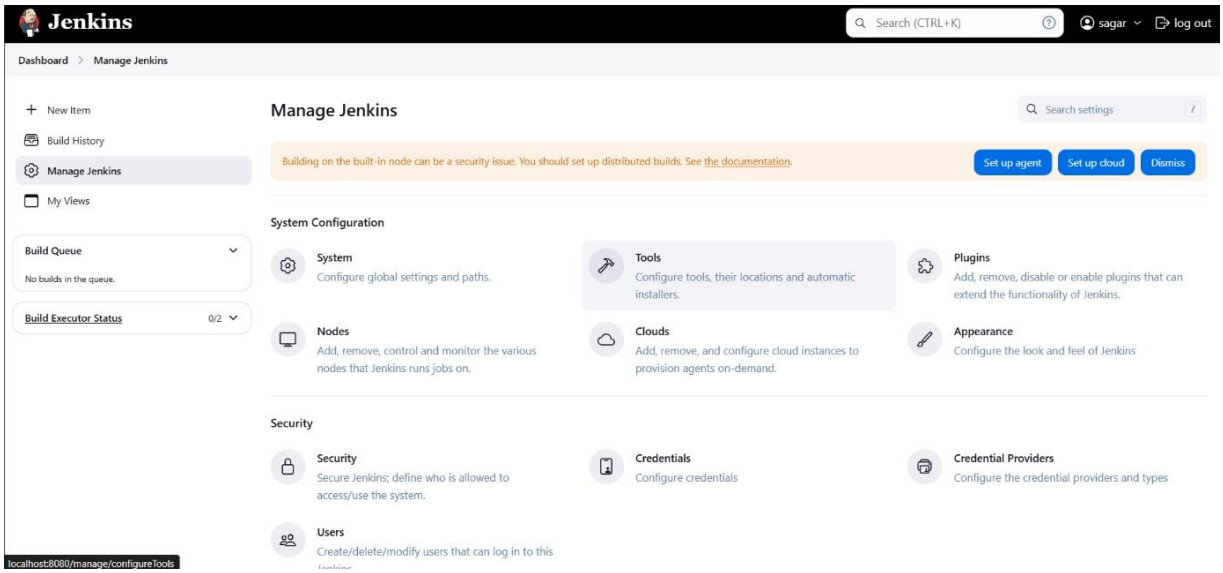
Getting Started

Jenkins is ready!

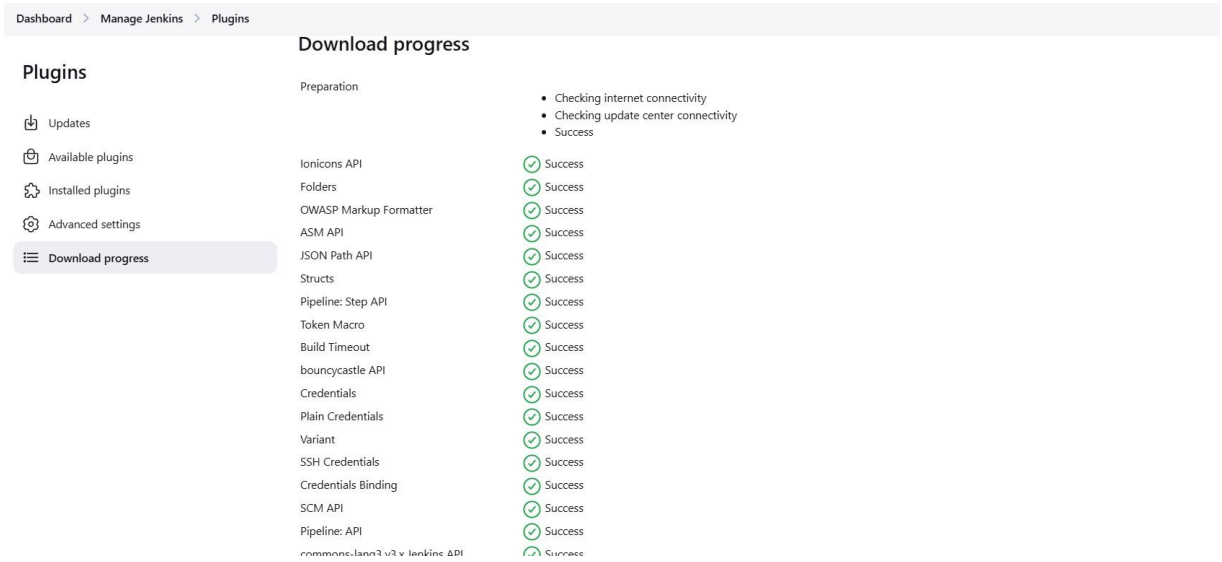
Your Jenkins setup is complete.

Start using Jenkins

LOGIN INTO YOUT JENKINS ACCOUNT



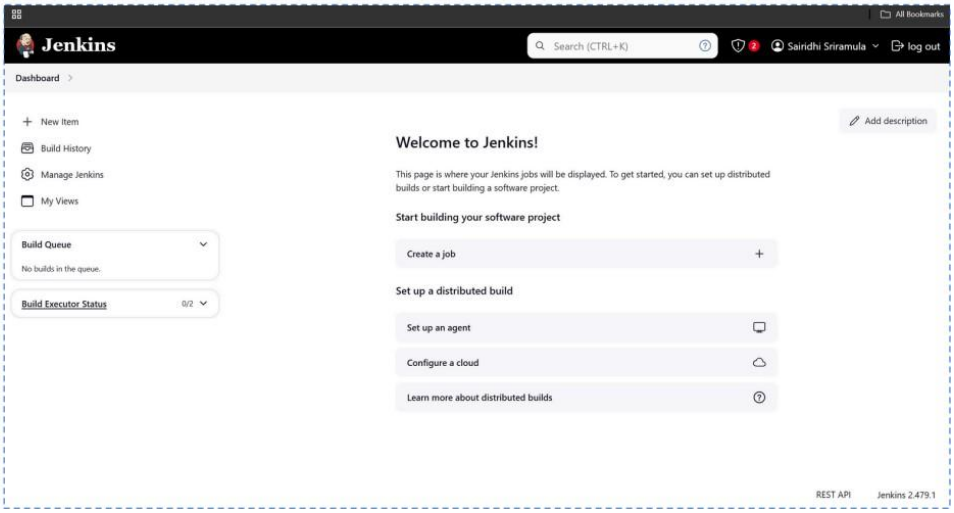
CLICK ON MANAGE JENKINS AND PLUGINS



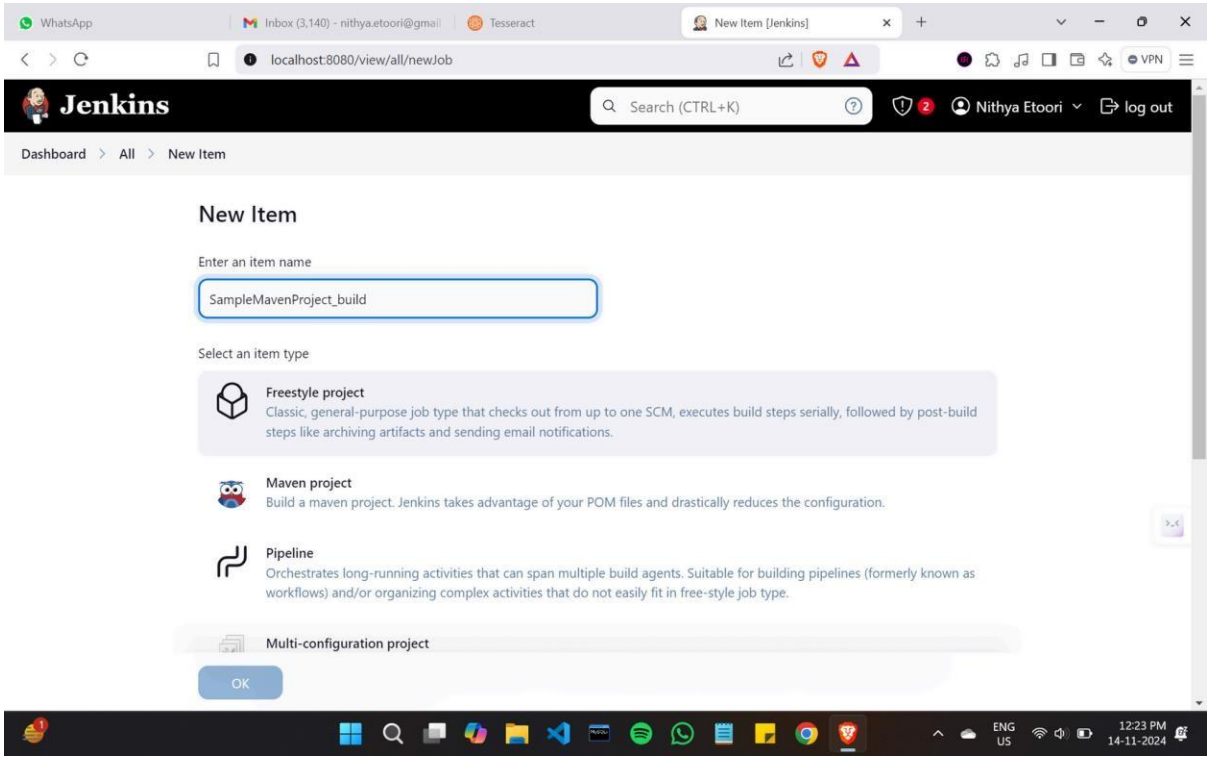
MAVEN JAVA PROJECT PIPELINE


Sign in to Jenkins
Username
admin
Password
Keep me signed in
Sign in

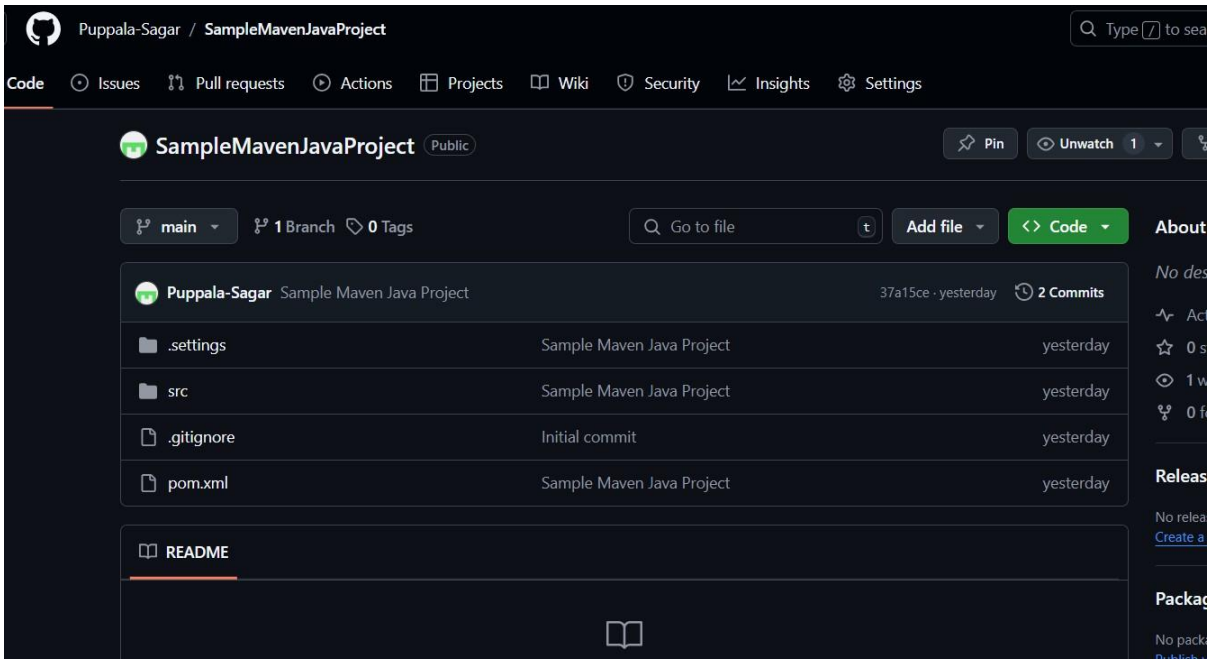
Dashboard


Jenkins
Search (CTRL+K)
Sairidhi Sriramula
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds
REST API
Jenkins 2.479.1

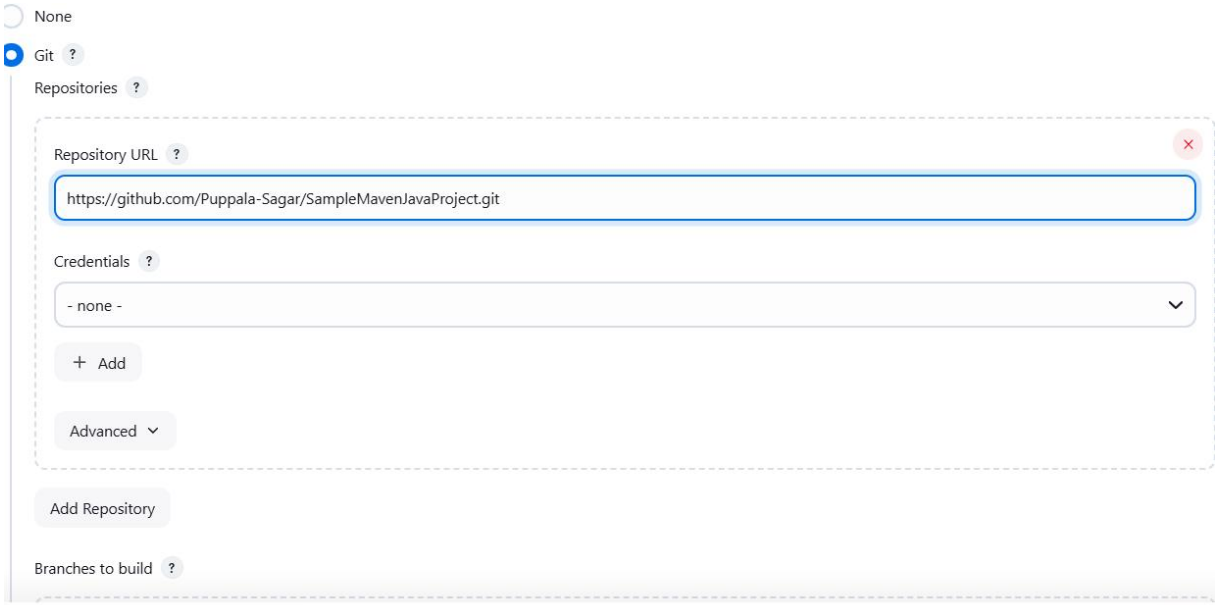
New freestyle project for build



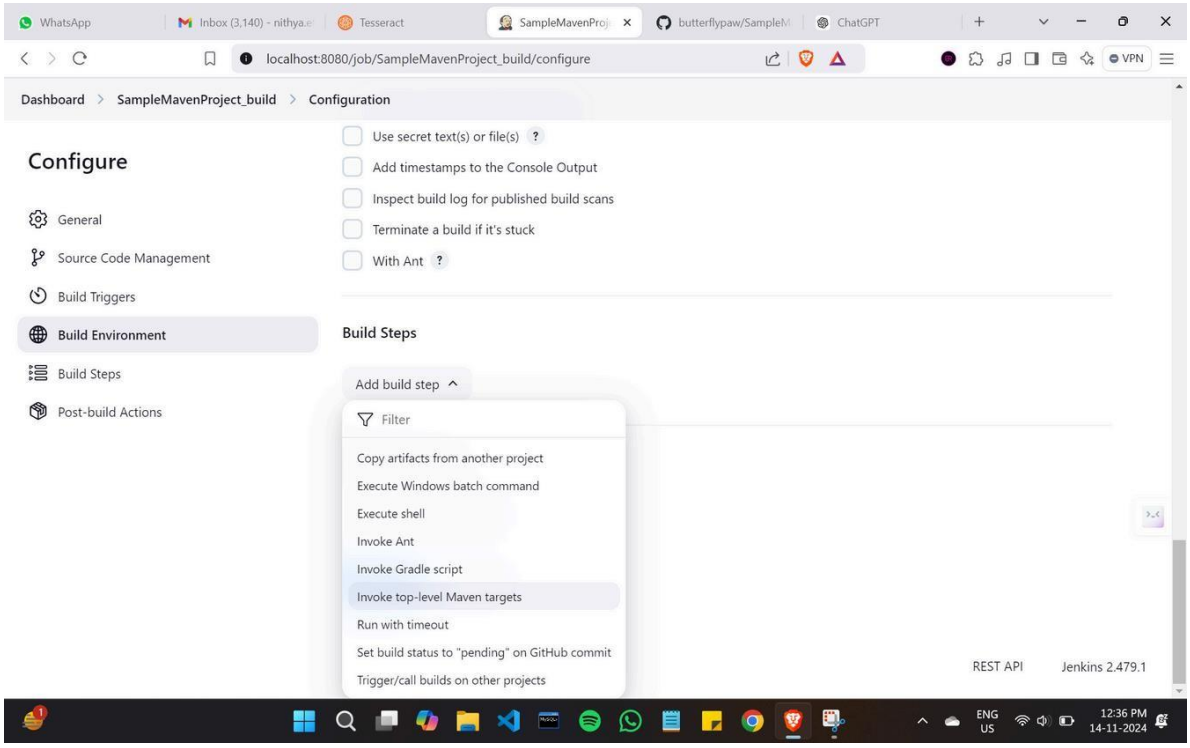
Git repo of existing project



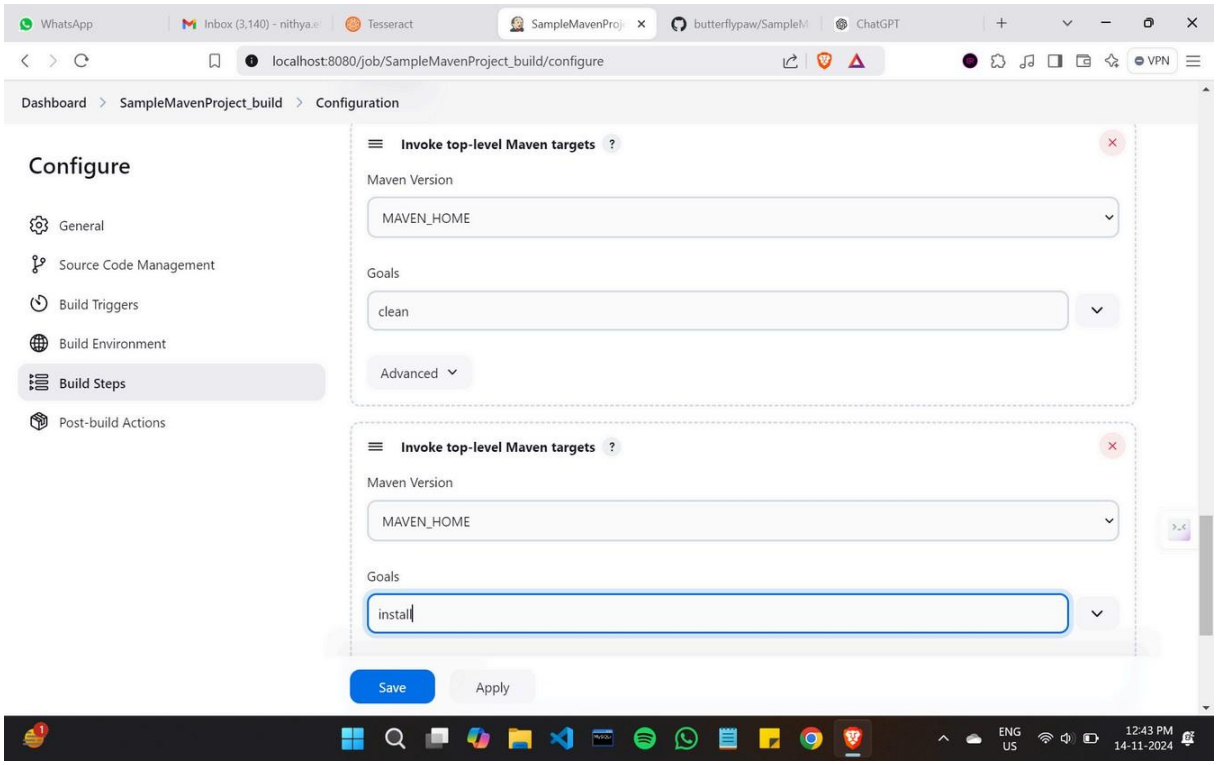
Add repo url an d select the branch



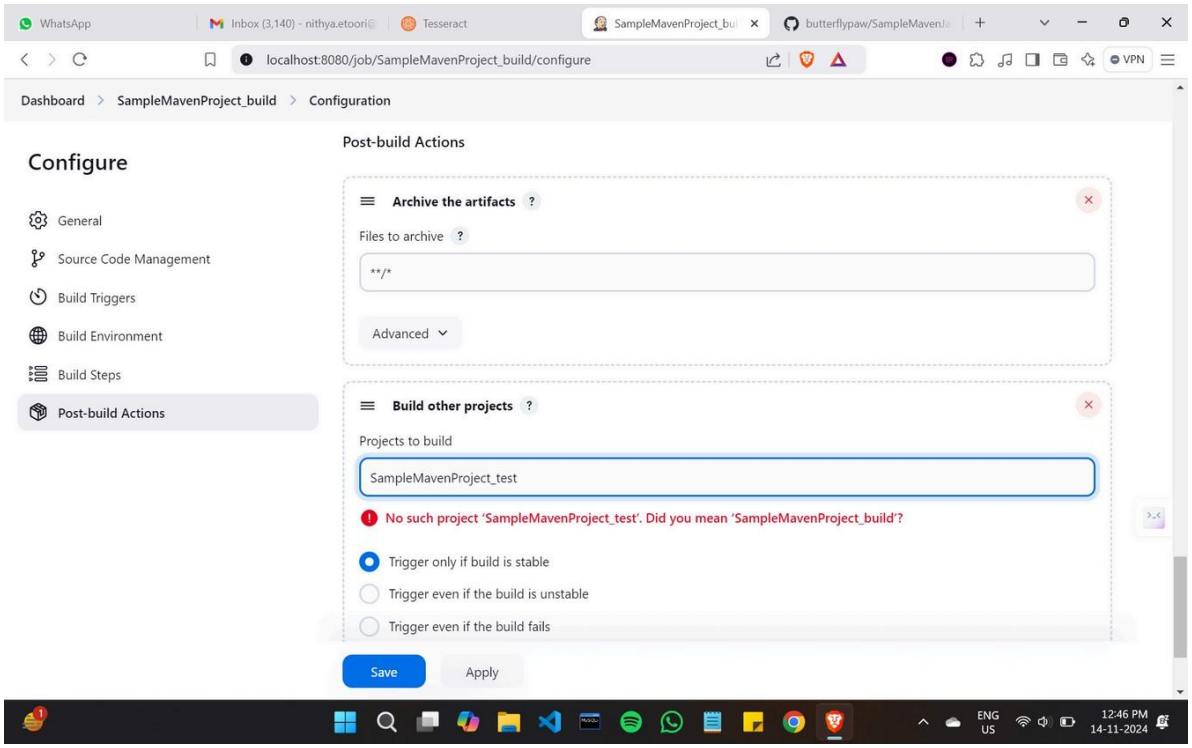
Build Steps



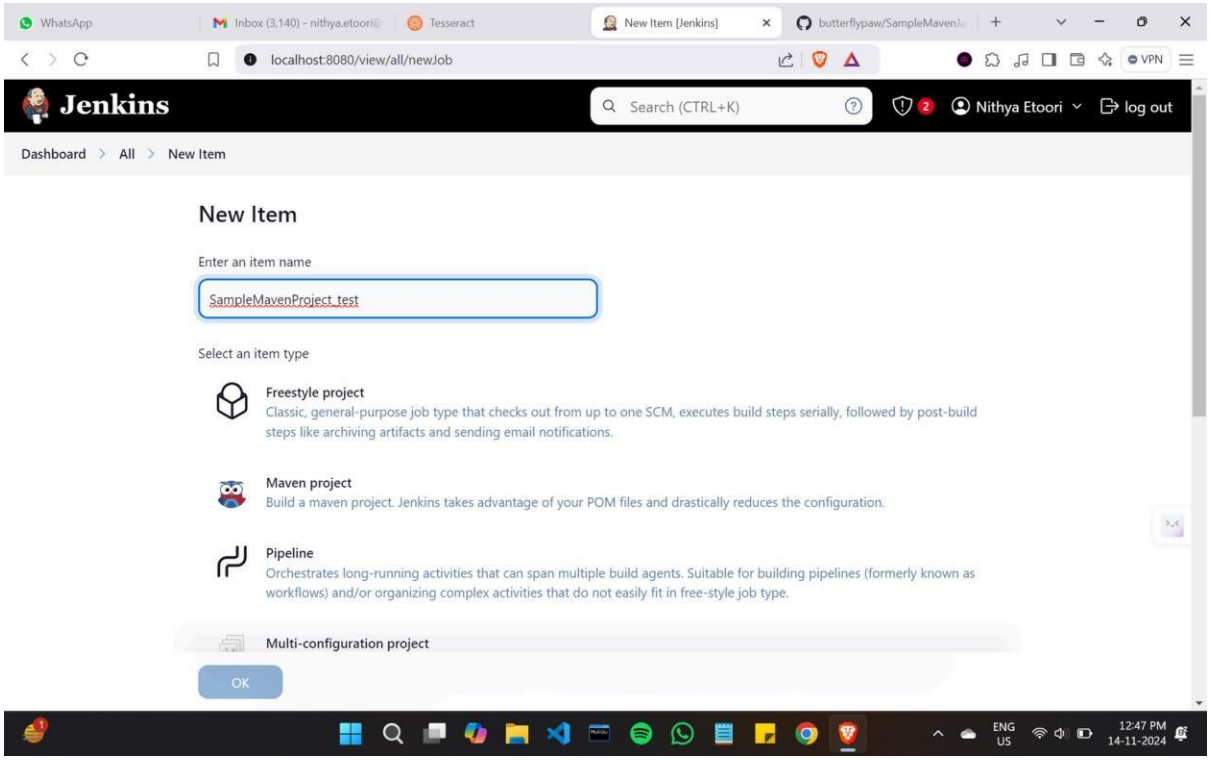
Add clean and install



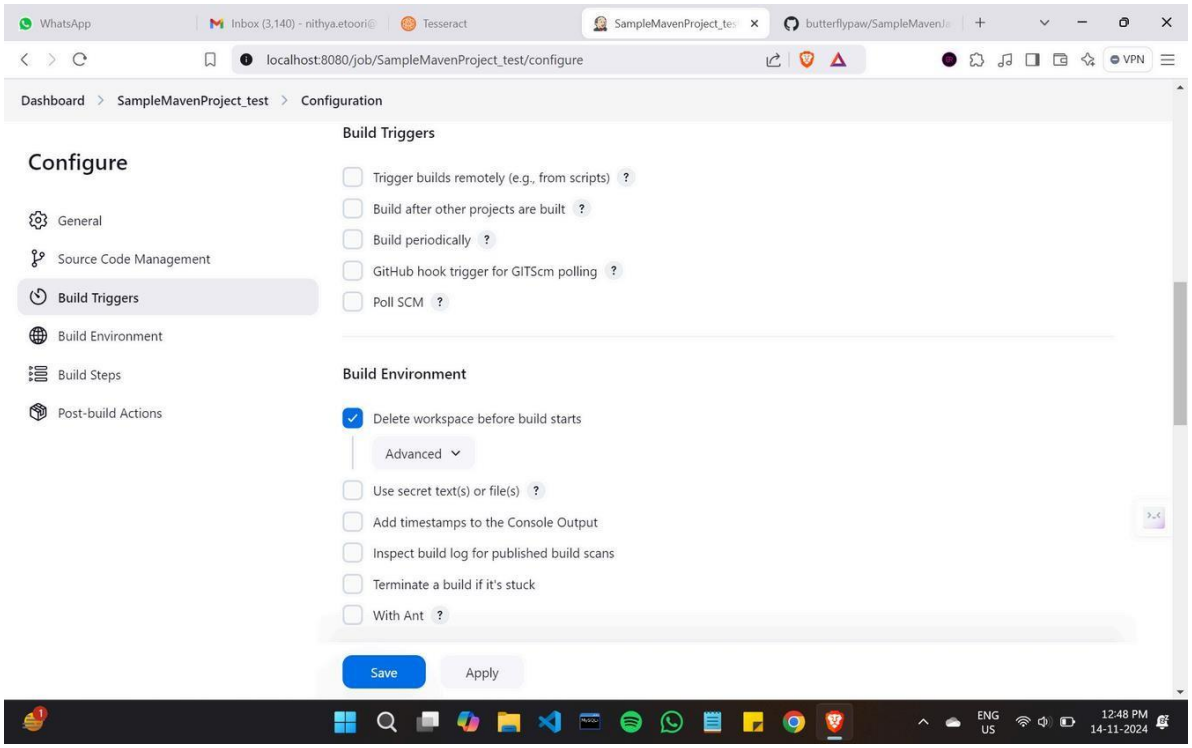
Post Build actions and invoking test



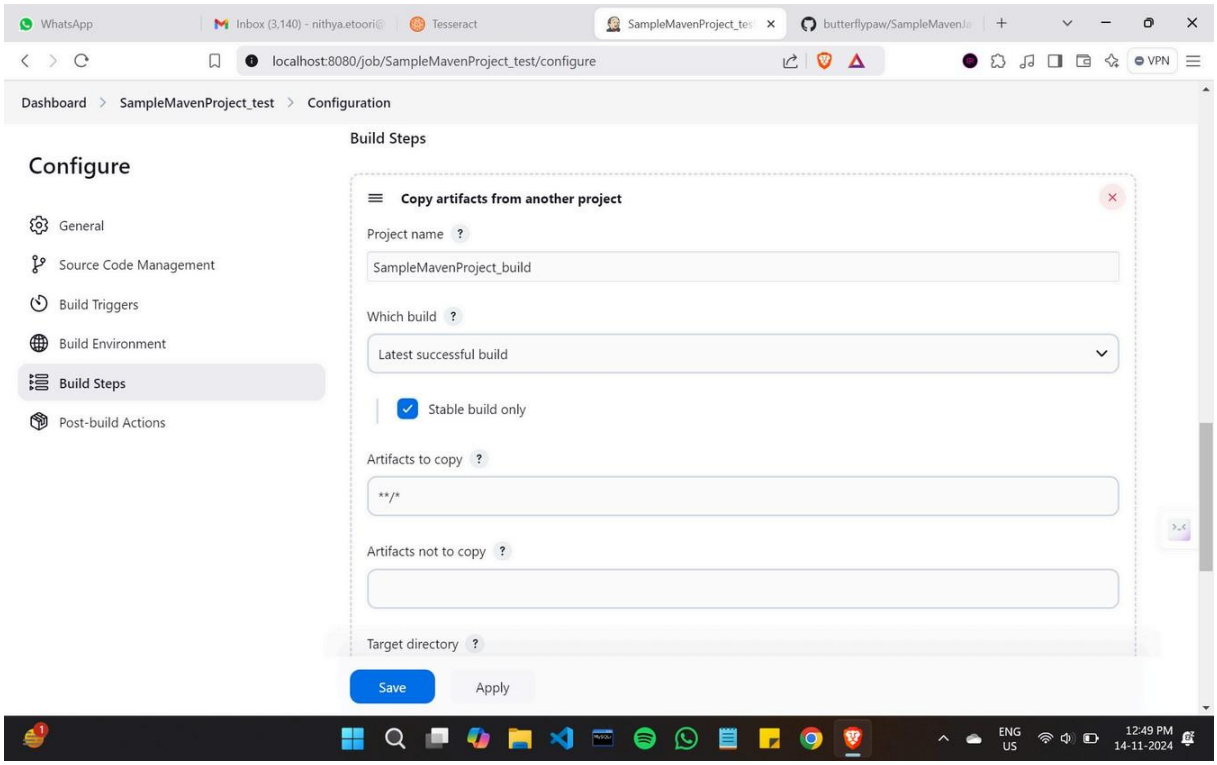
New freestyle project for test



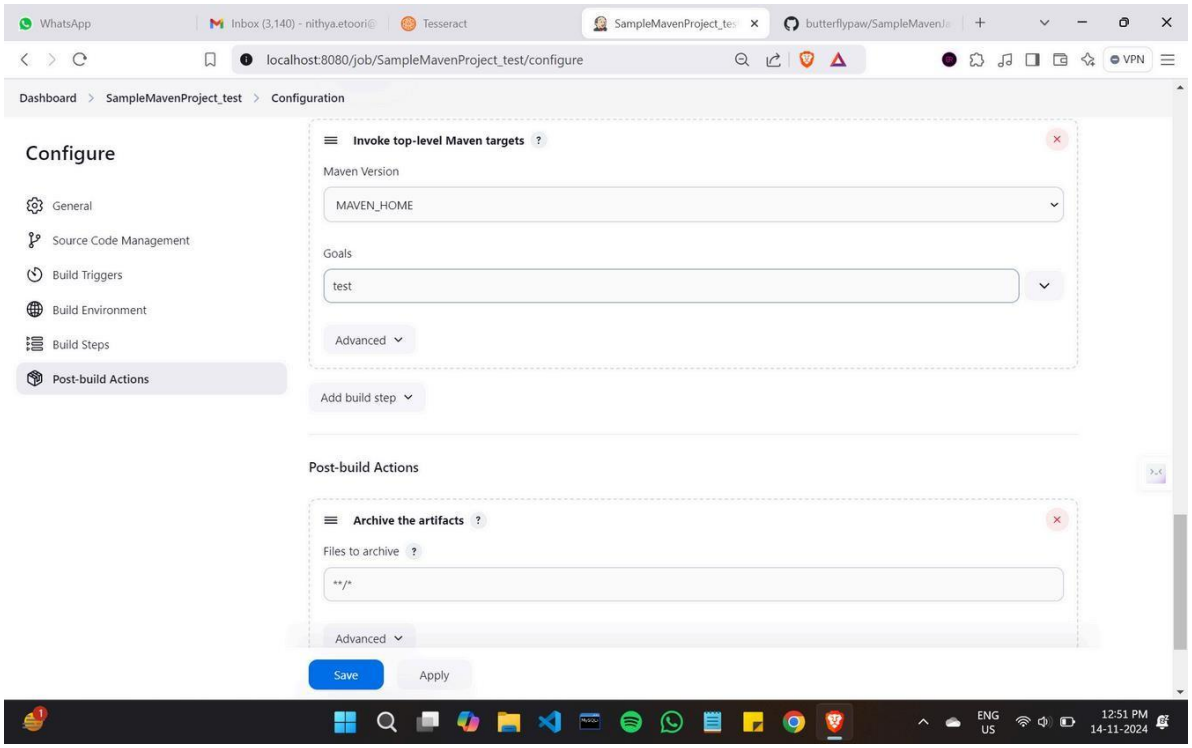
Build Environment



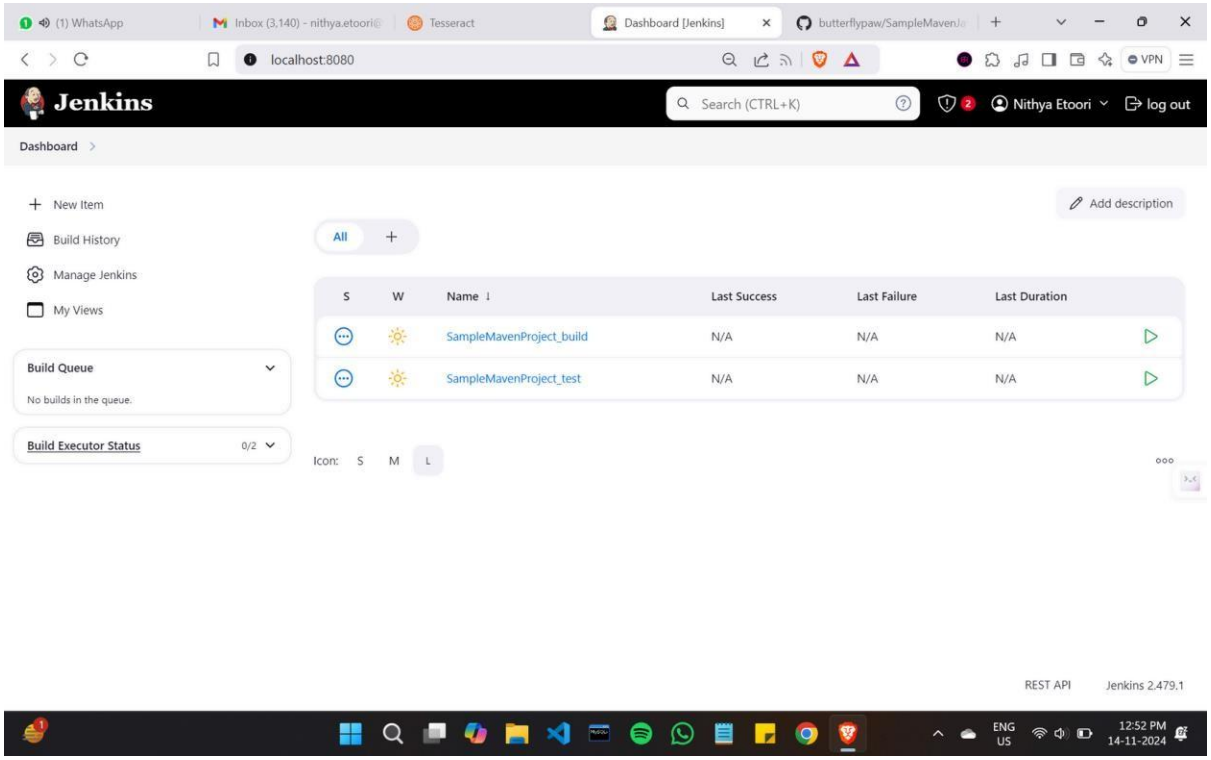
Add build project



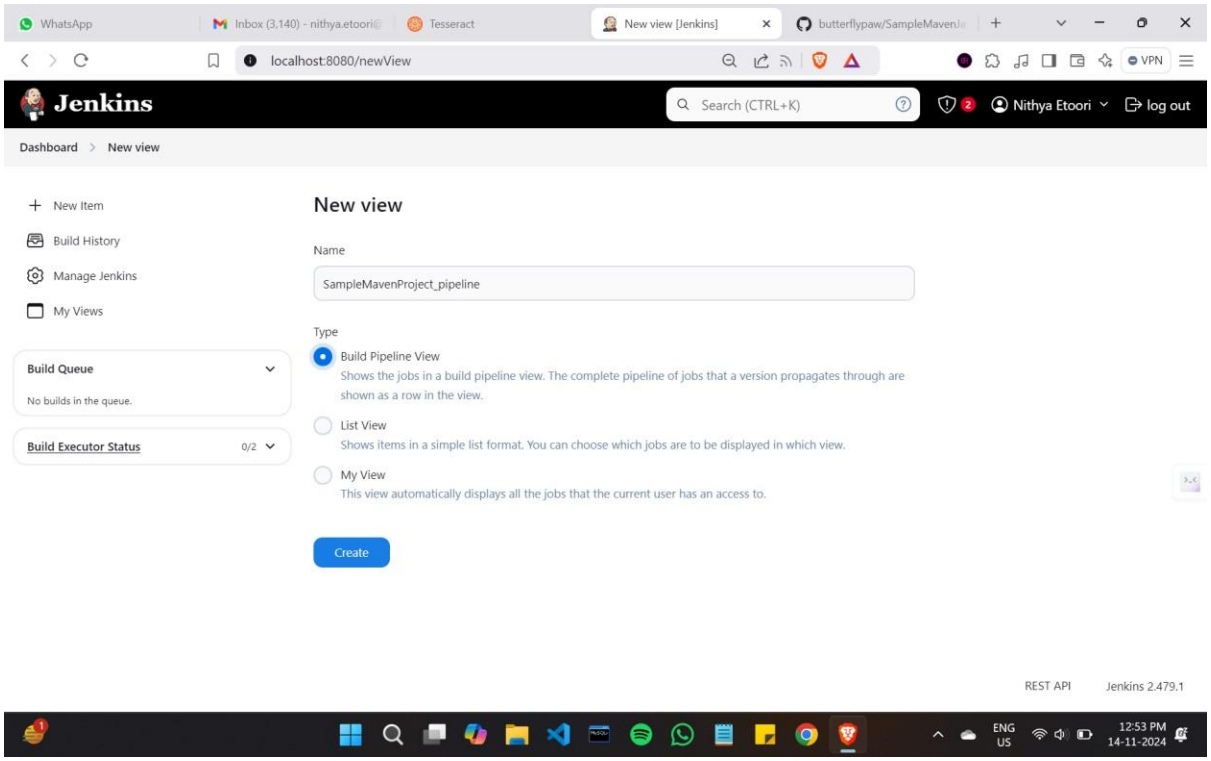
Add test goal



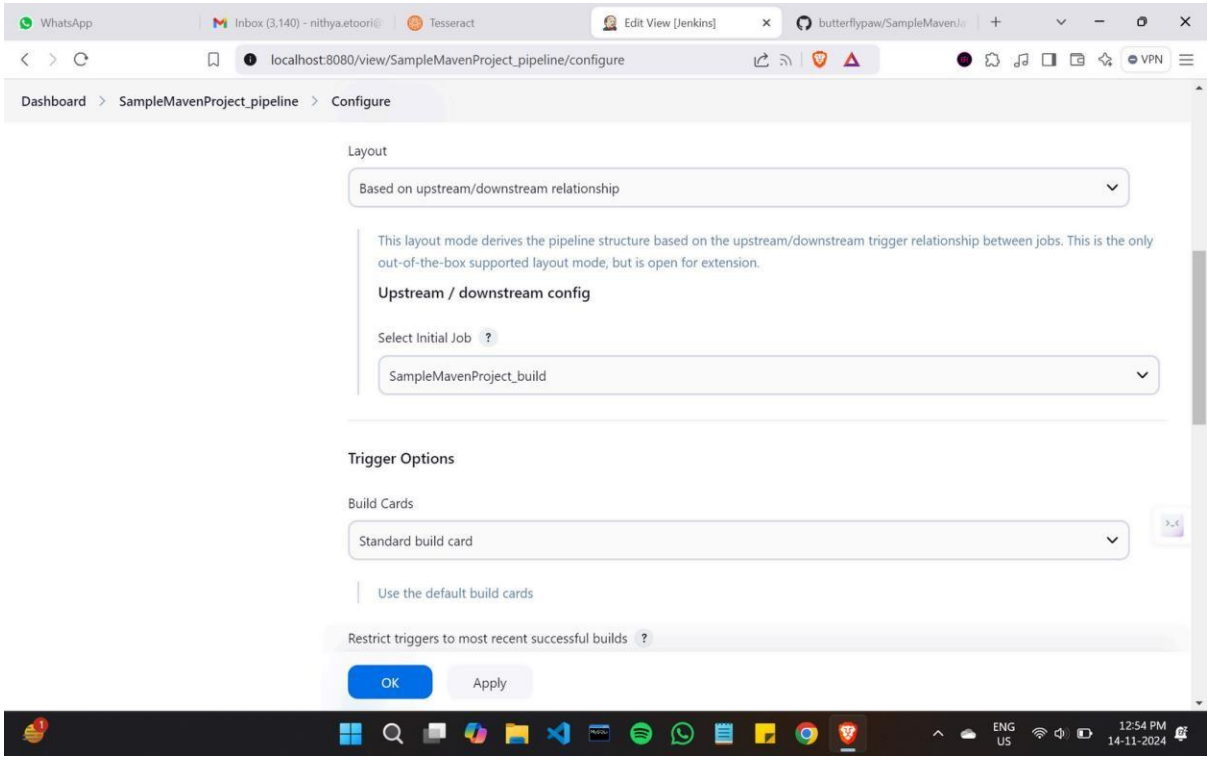
Dashboard after build and test projects



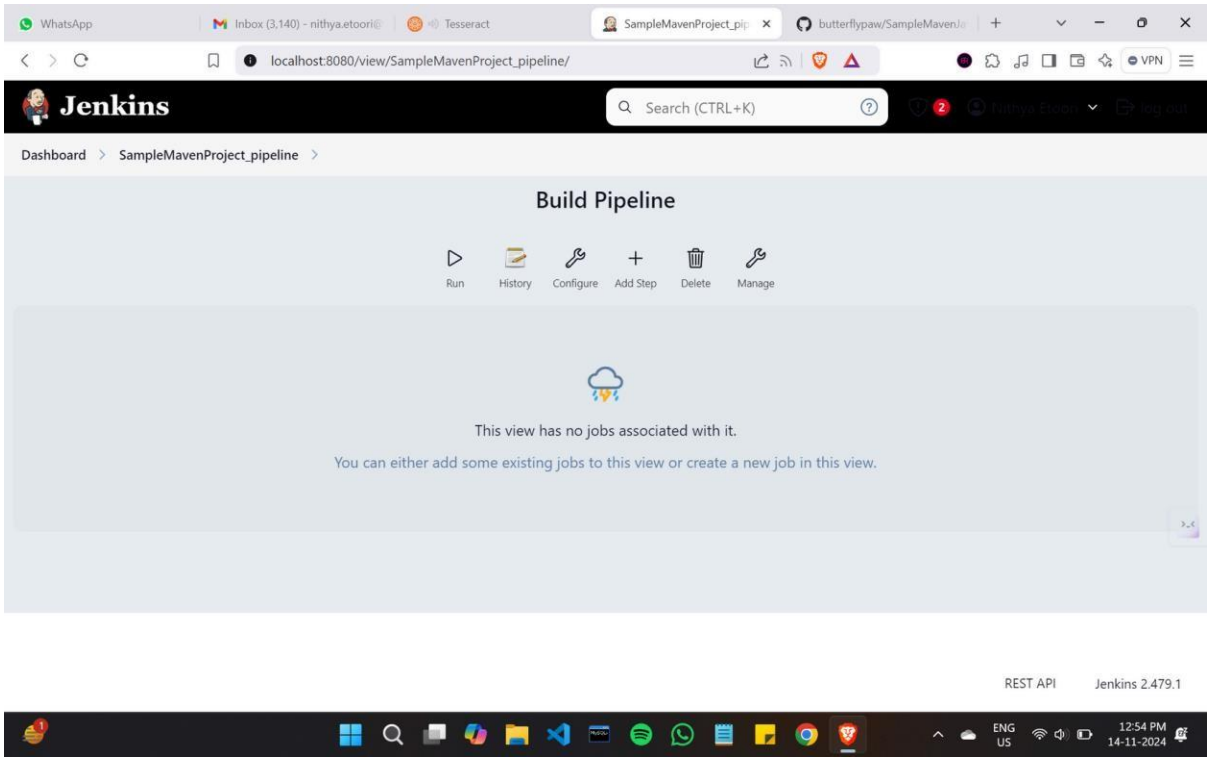
Create pipeline



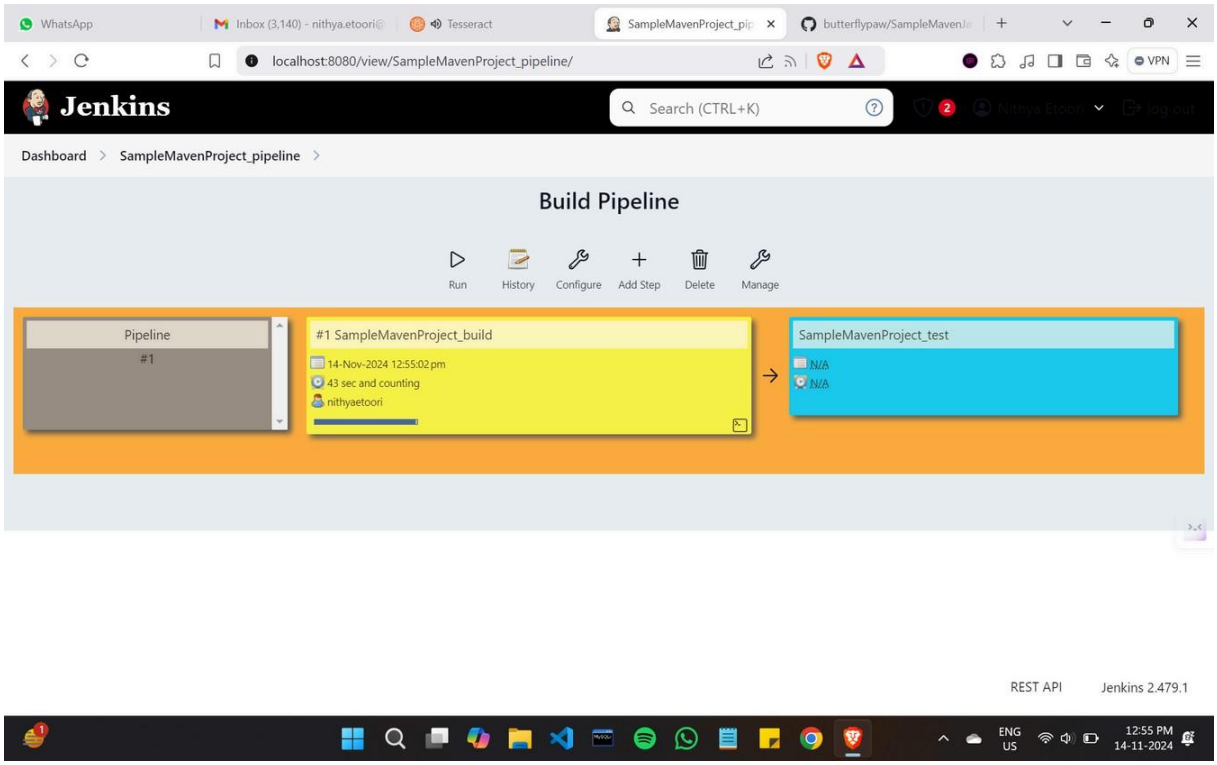
Select build project



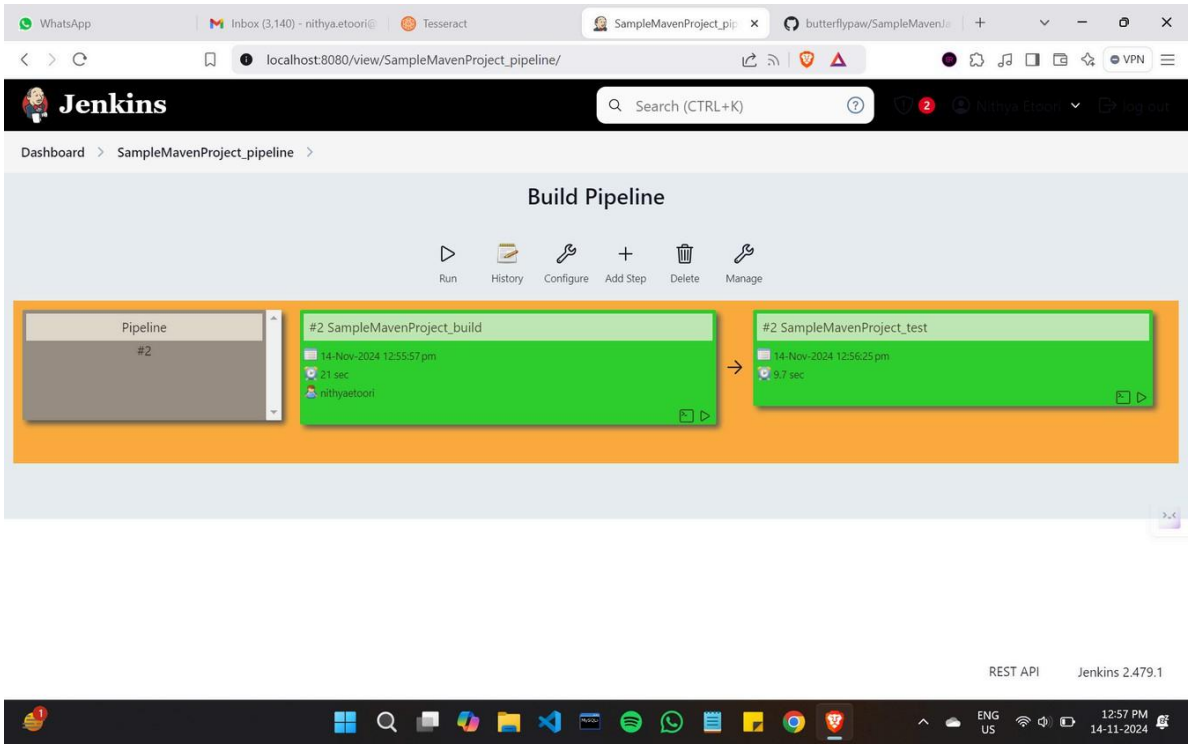
Build pipeline



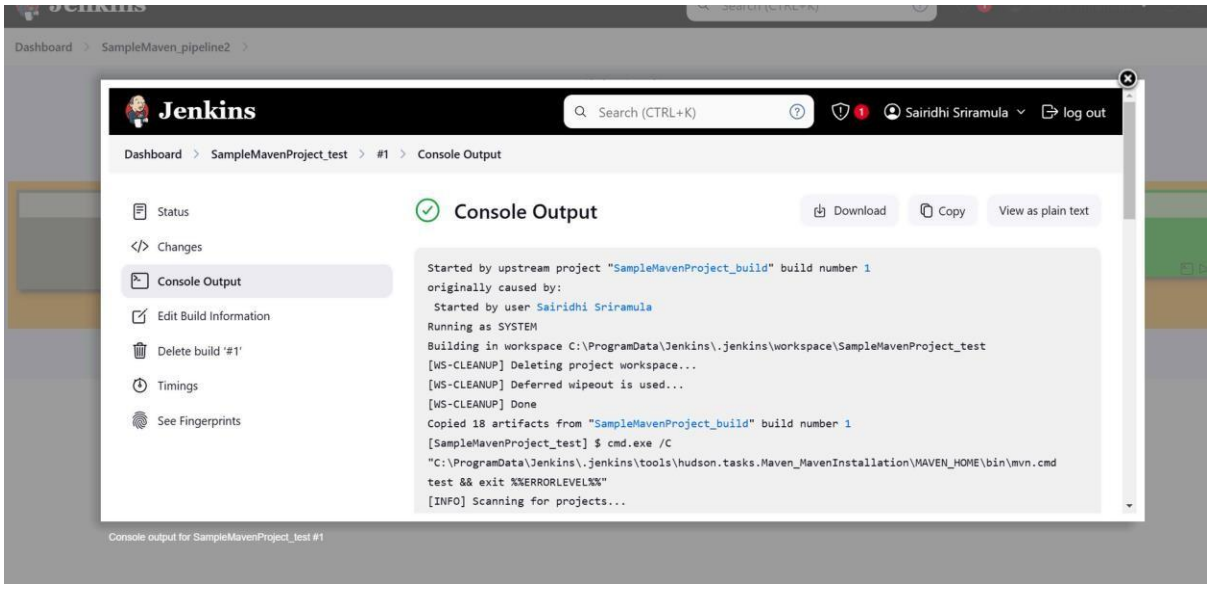
Graphical representation of pipeline



Green color indicate successful pipleline

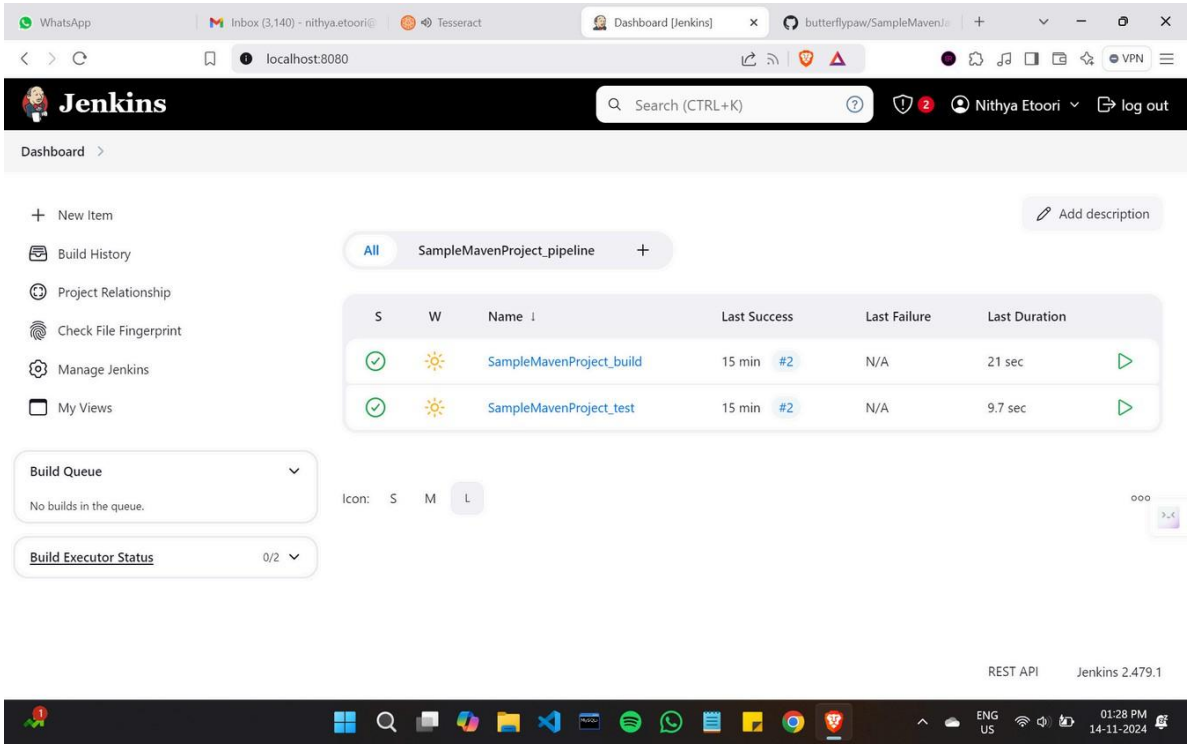


Log

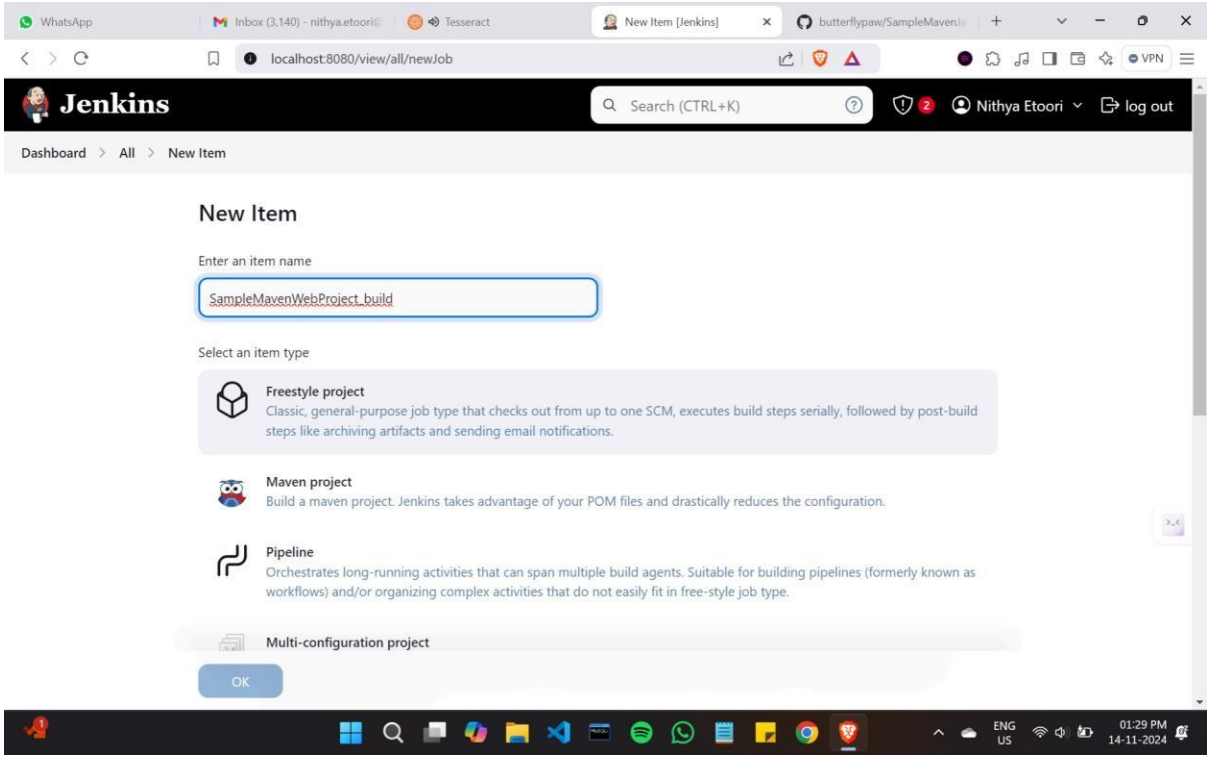


Jenkins Poll SCM

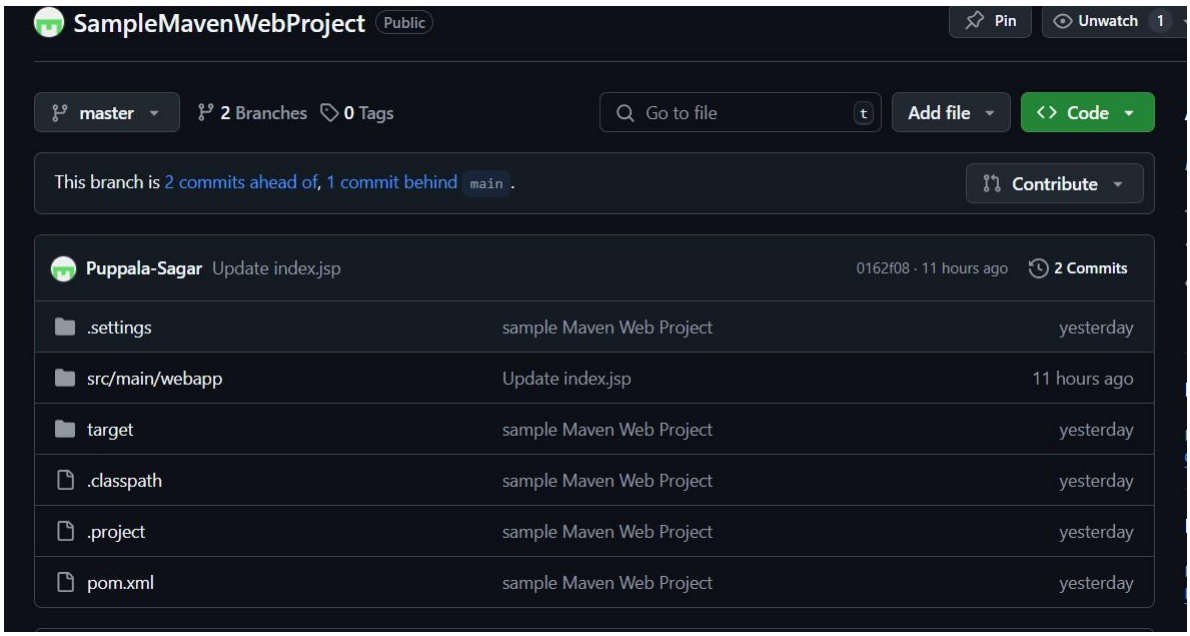
Jenkins Dashboard



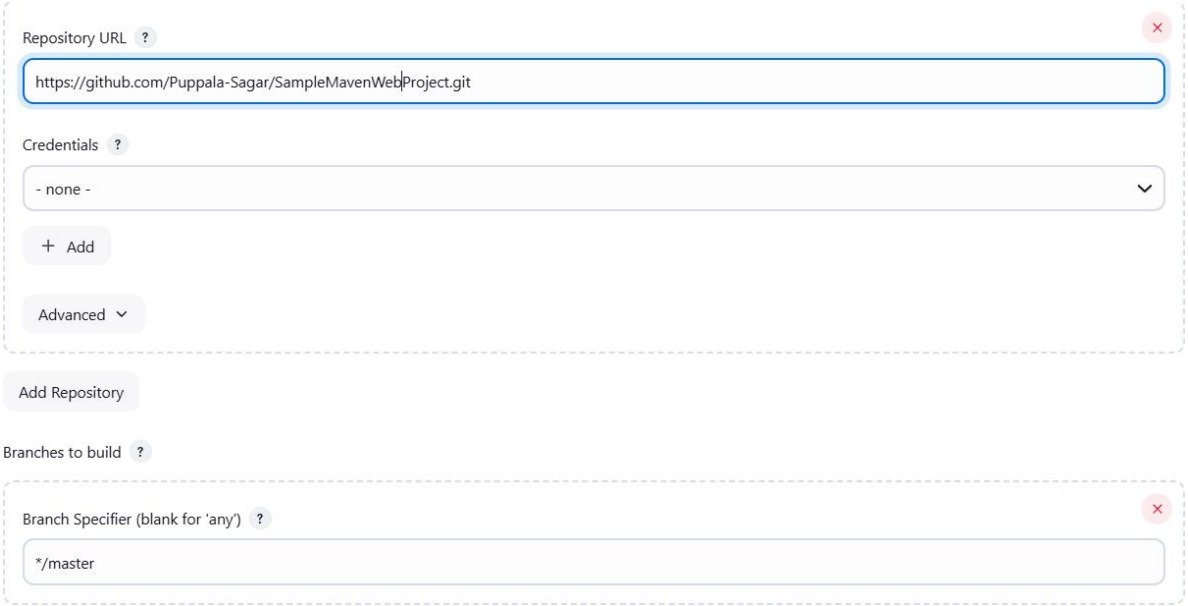
New build freestyle project



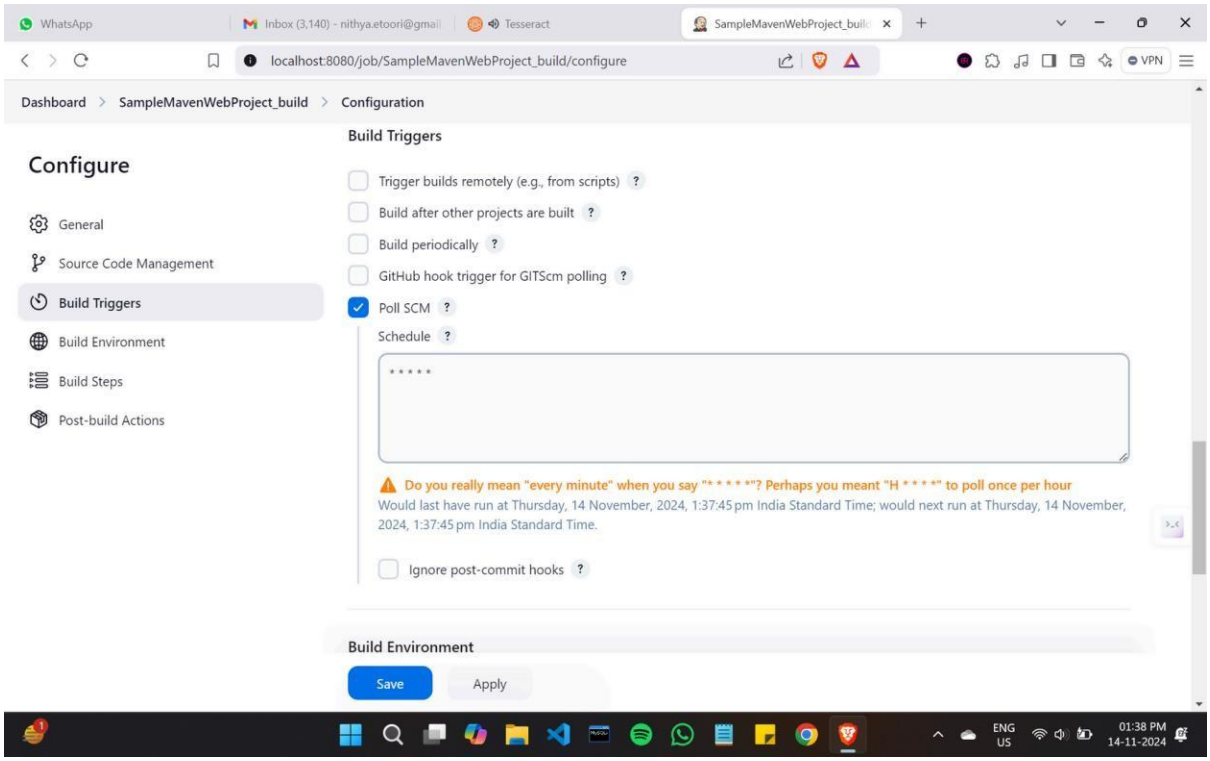
Maven Web Project git repo



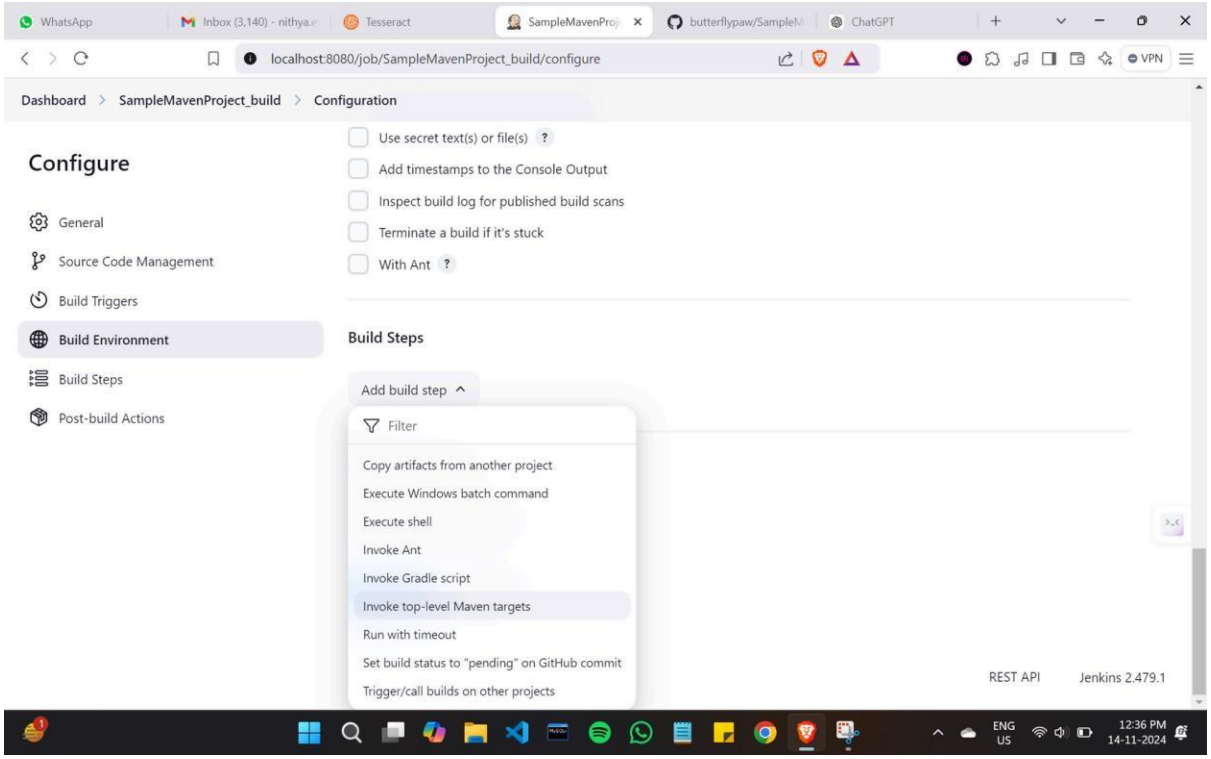
Add git repo and select branch



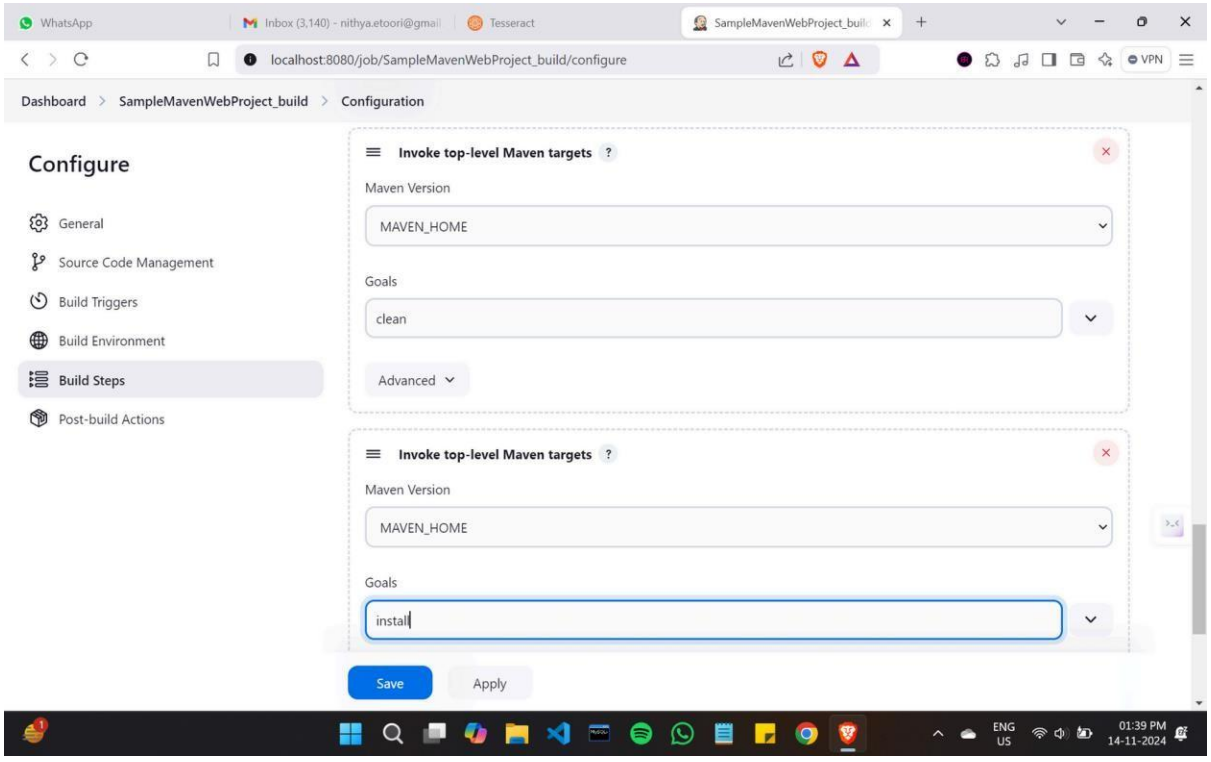
Trigger Poll SCM



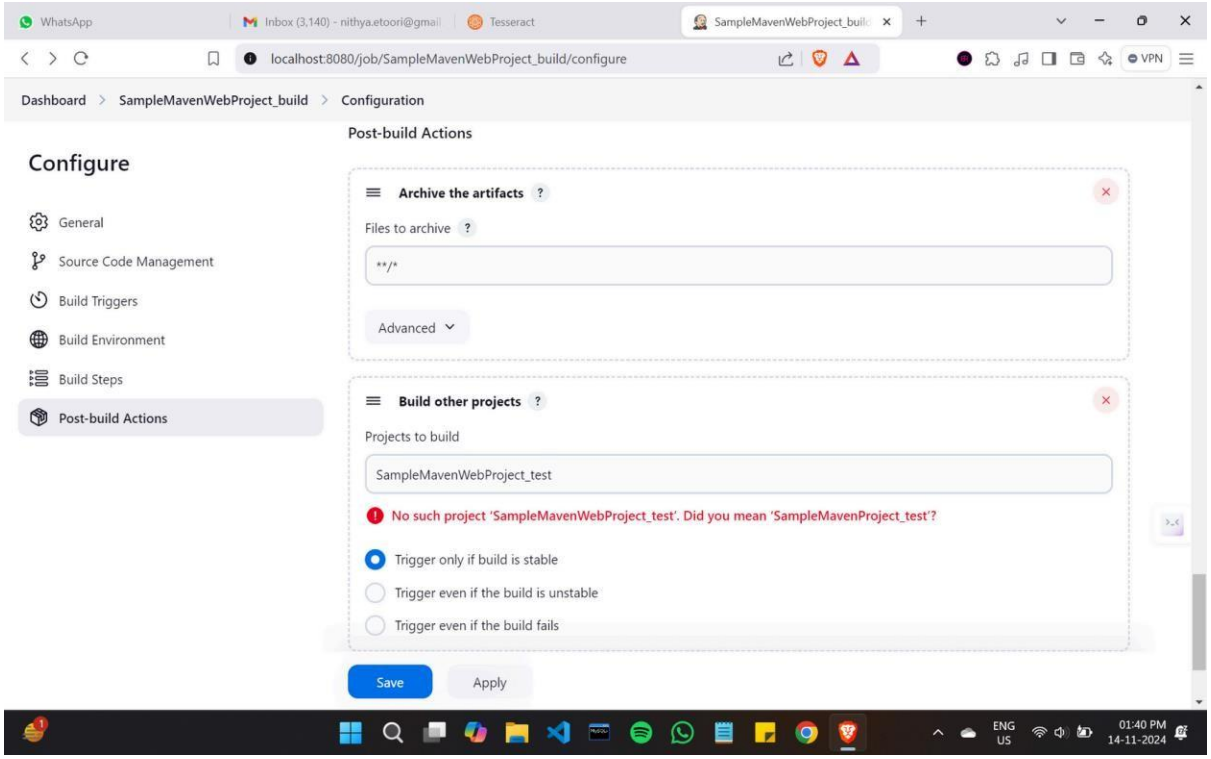
Build Steps



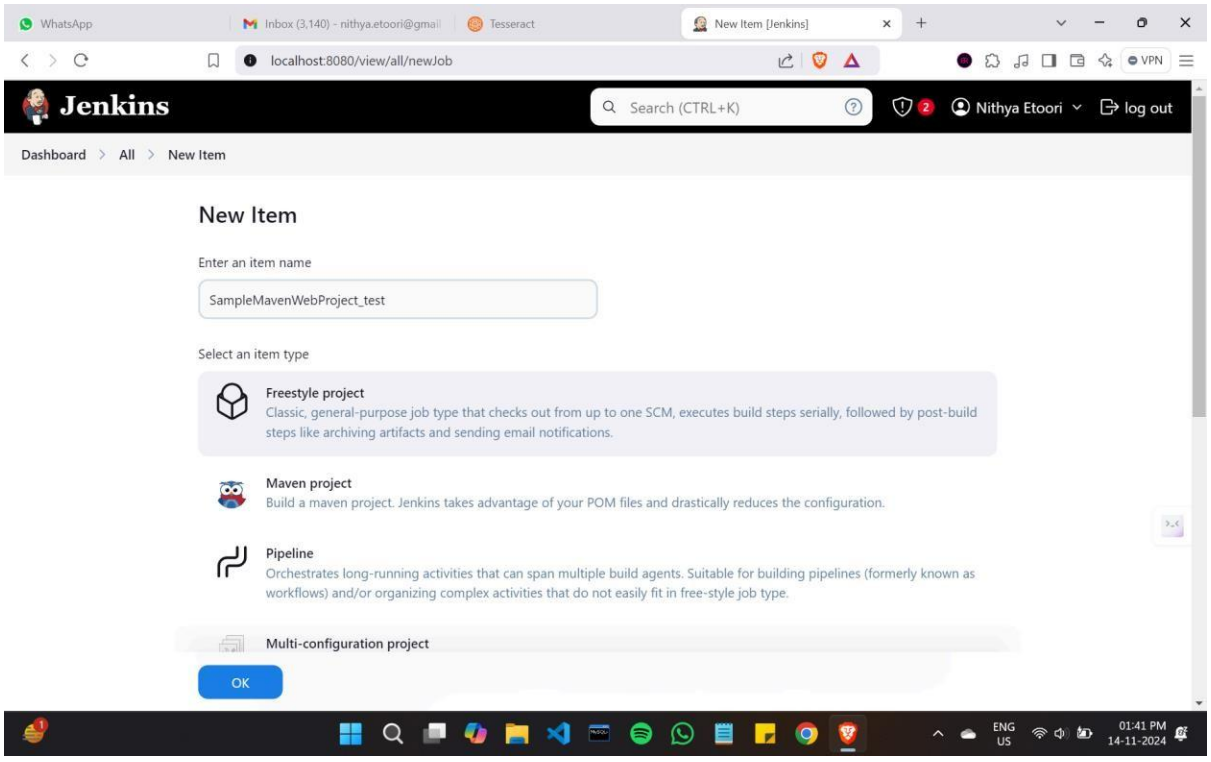
Add clean an d instal l



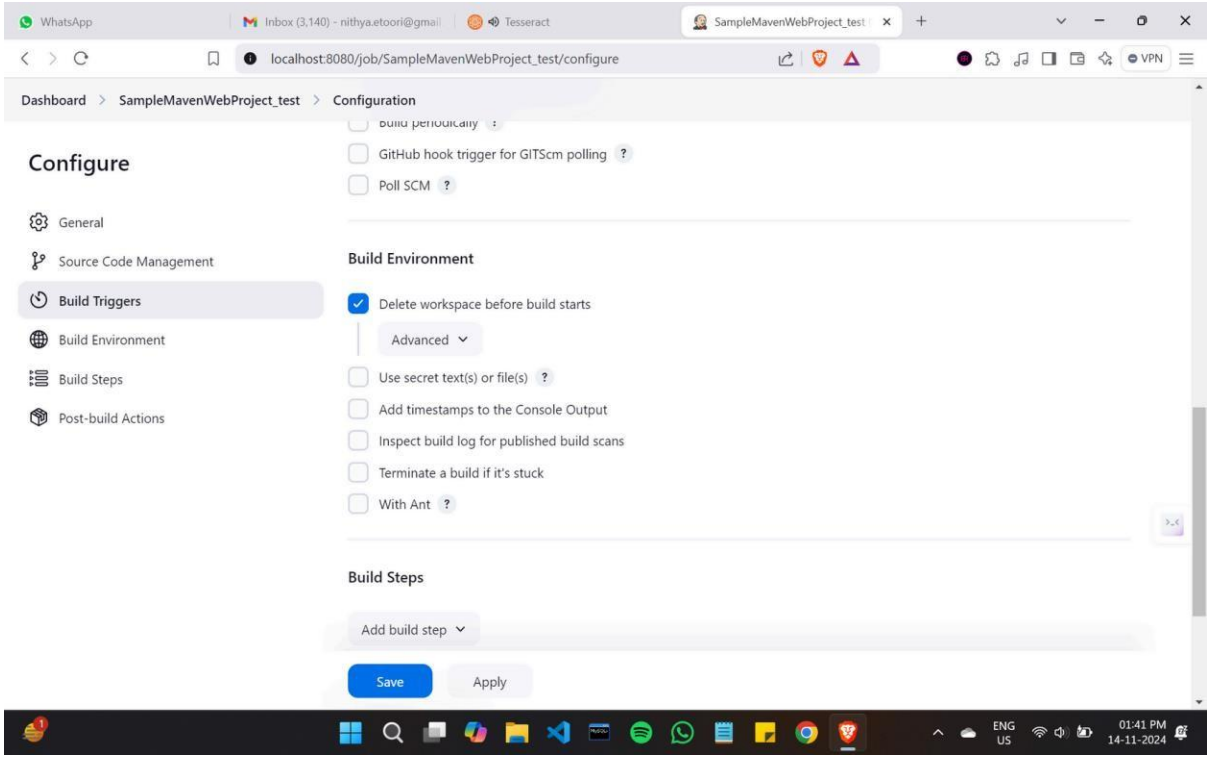
Post build actions and trigger test



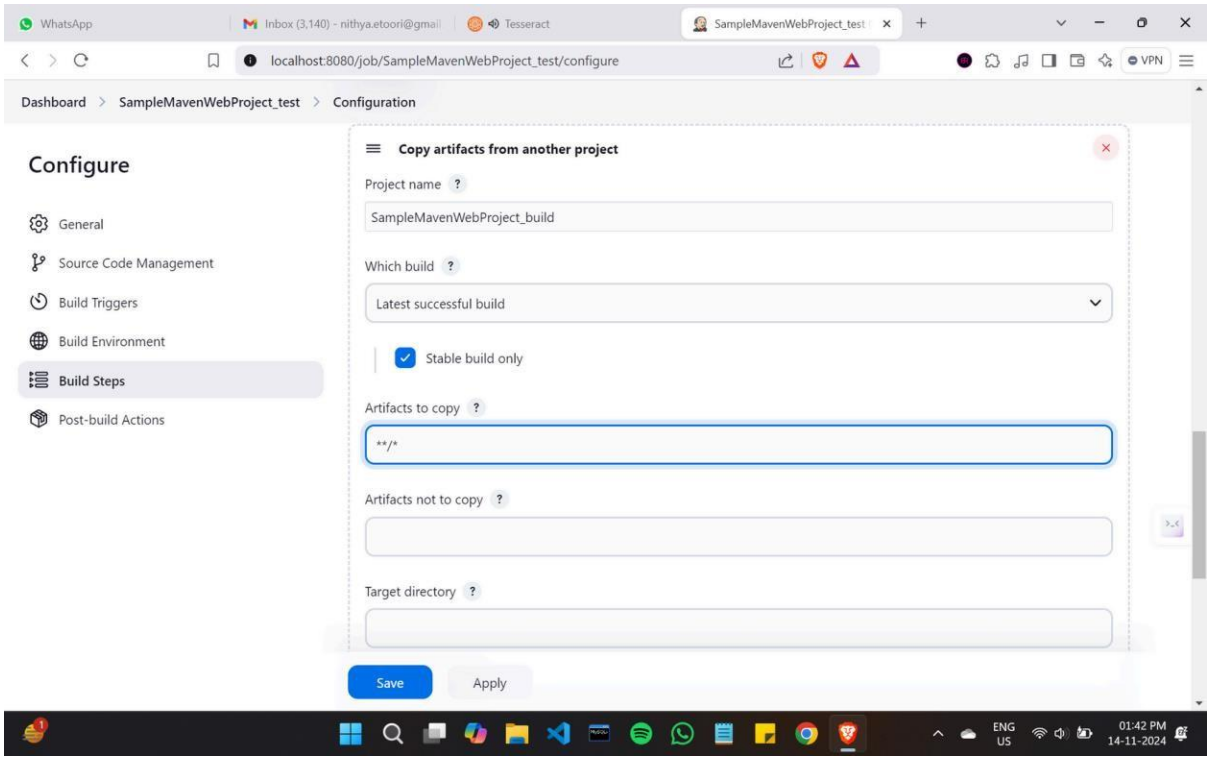
New freestyle project for test



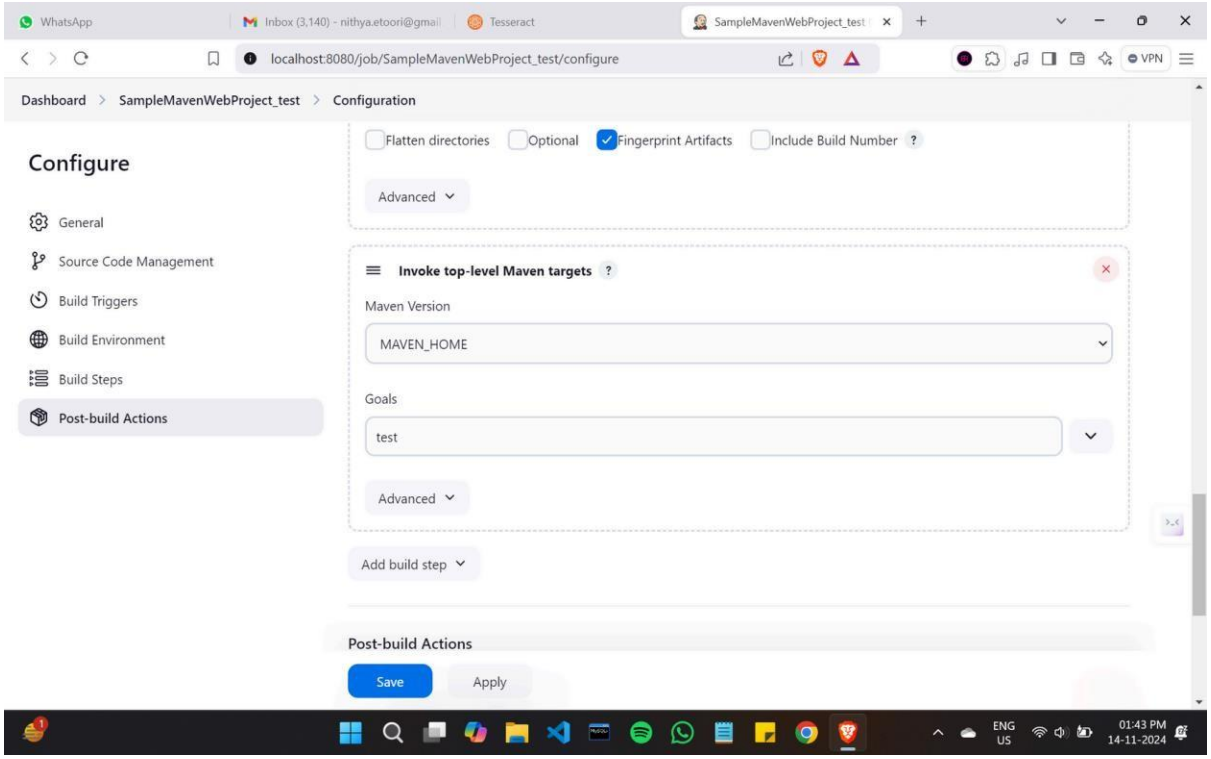
Build Environment



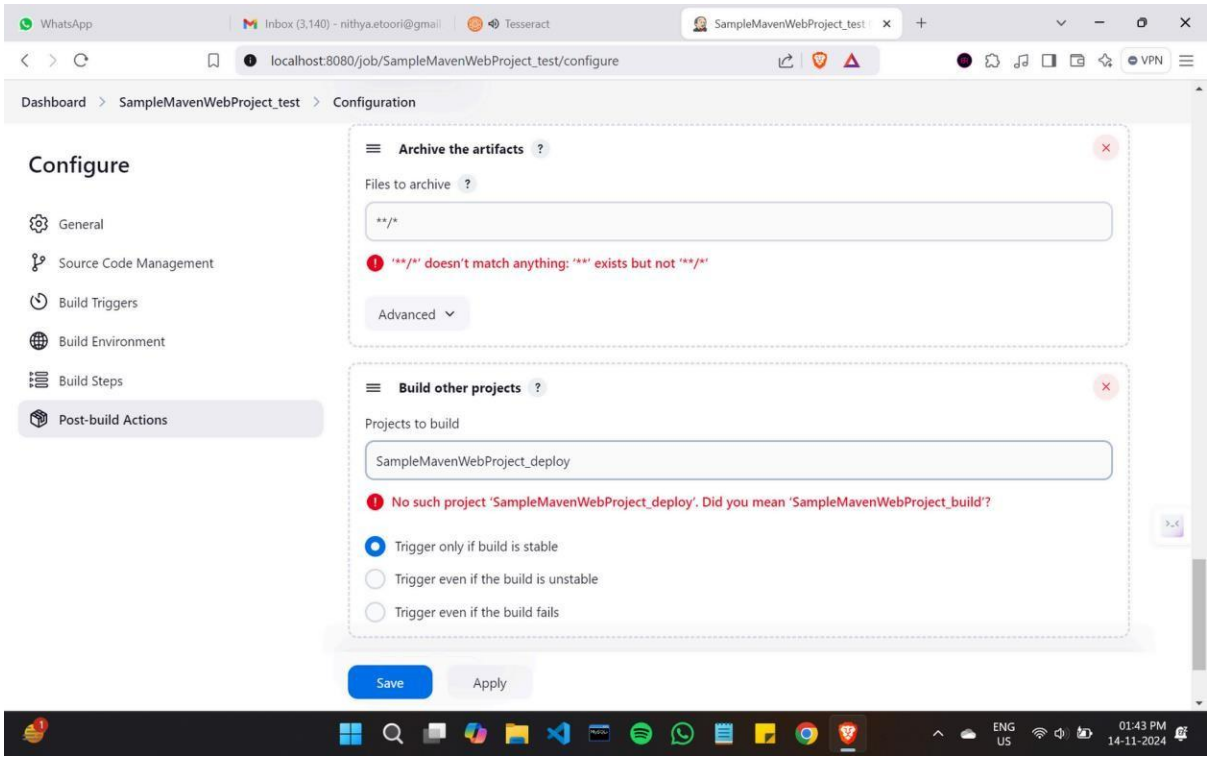
Add build project



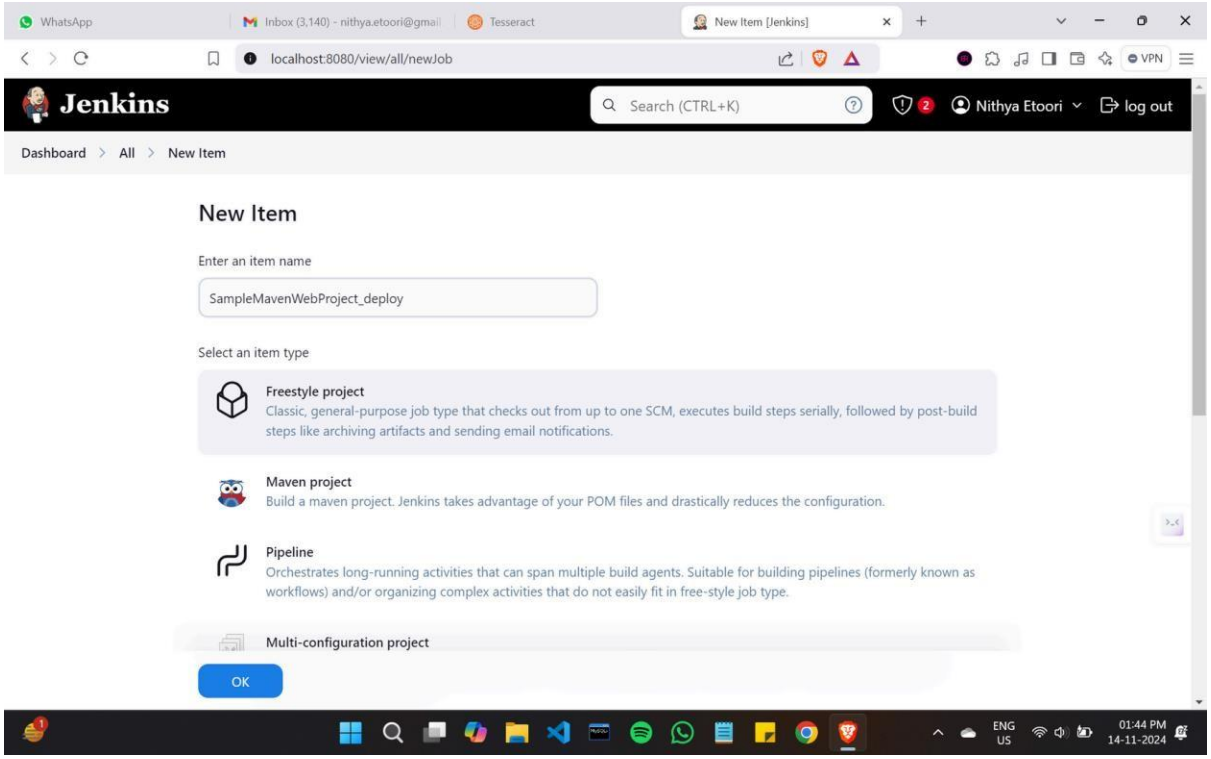
Add test goal



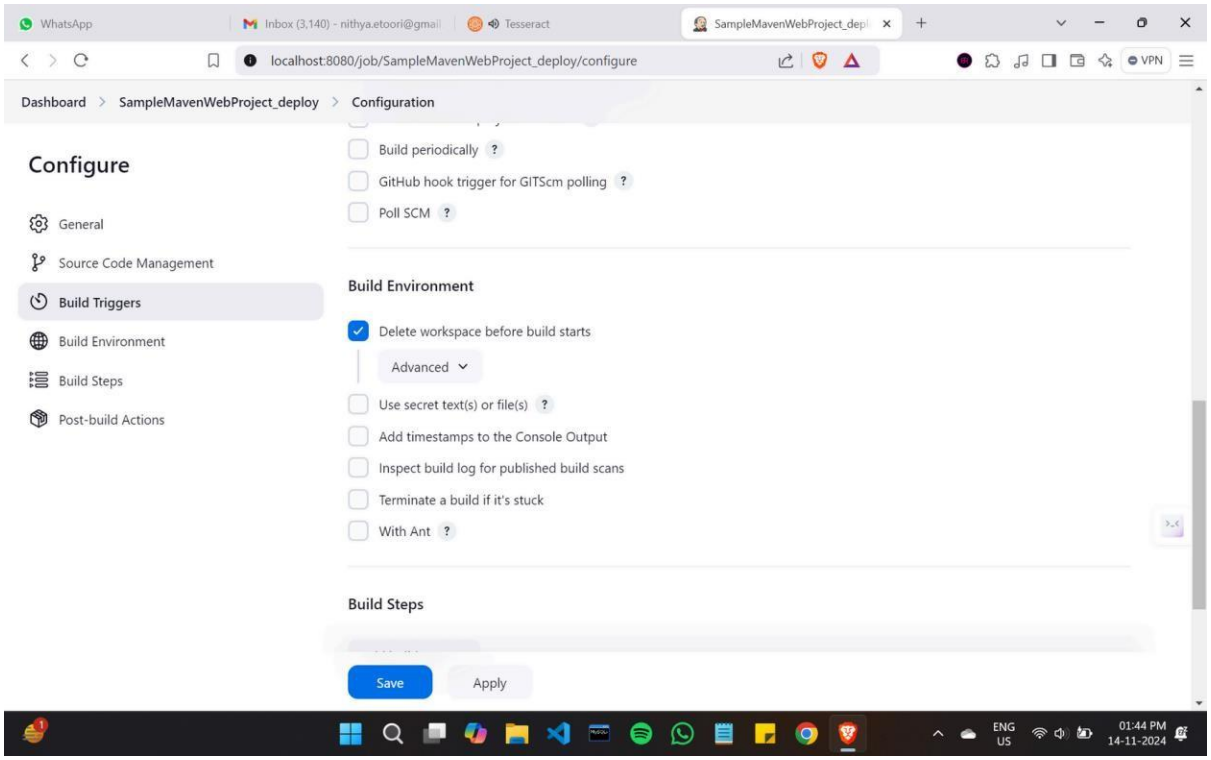
Trigger deploy



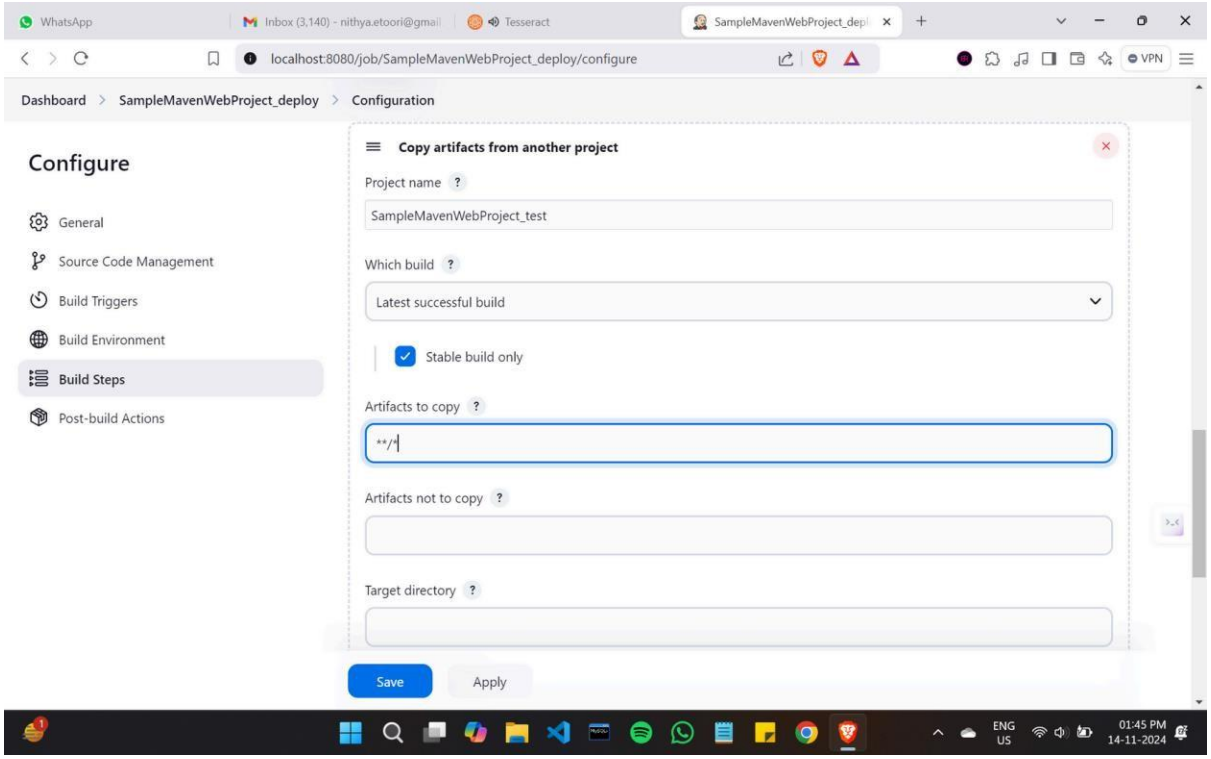
New freestyle project for deploy



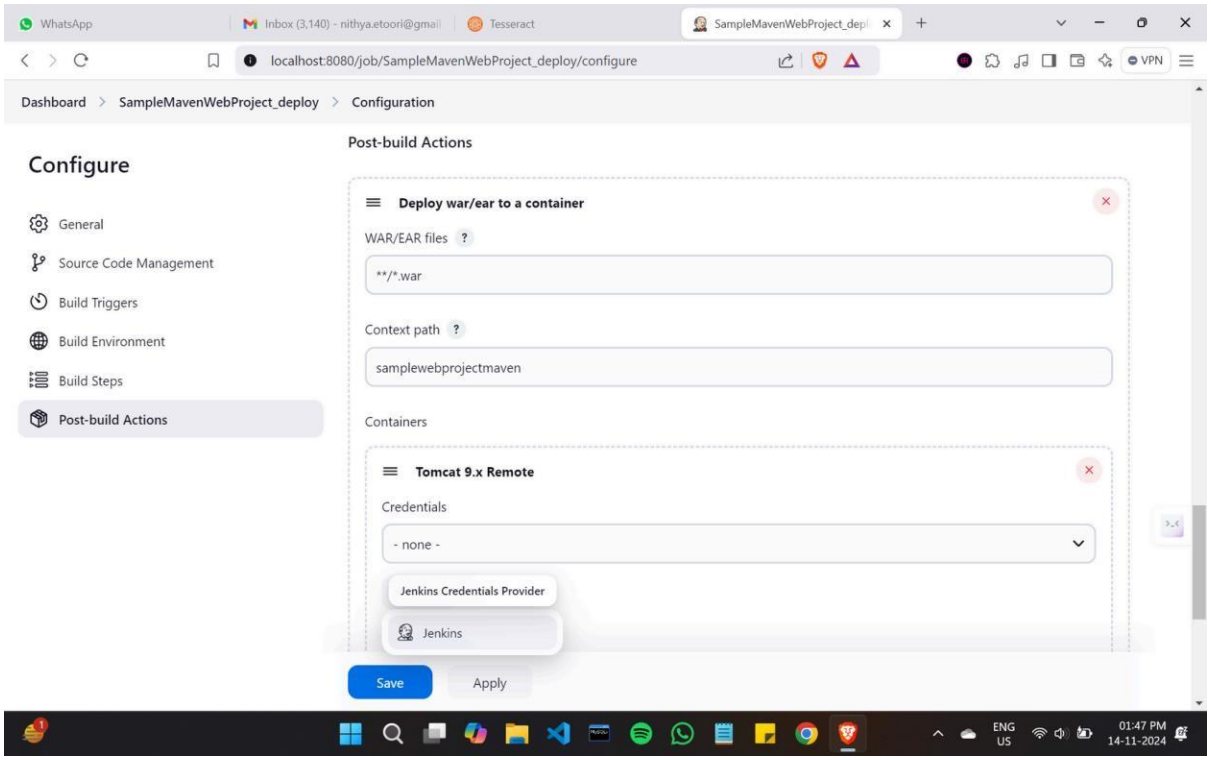
Build Environment



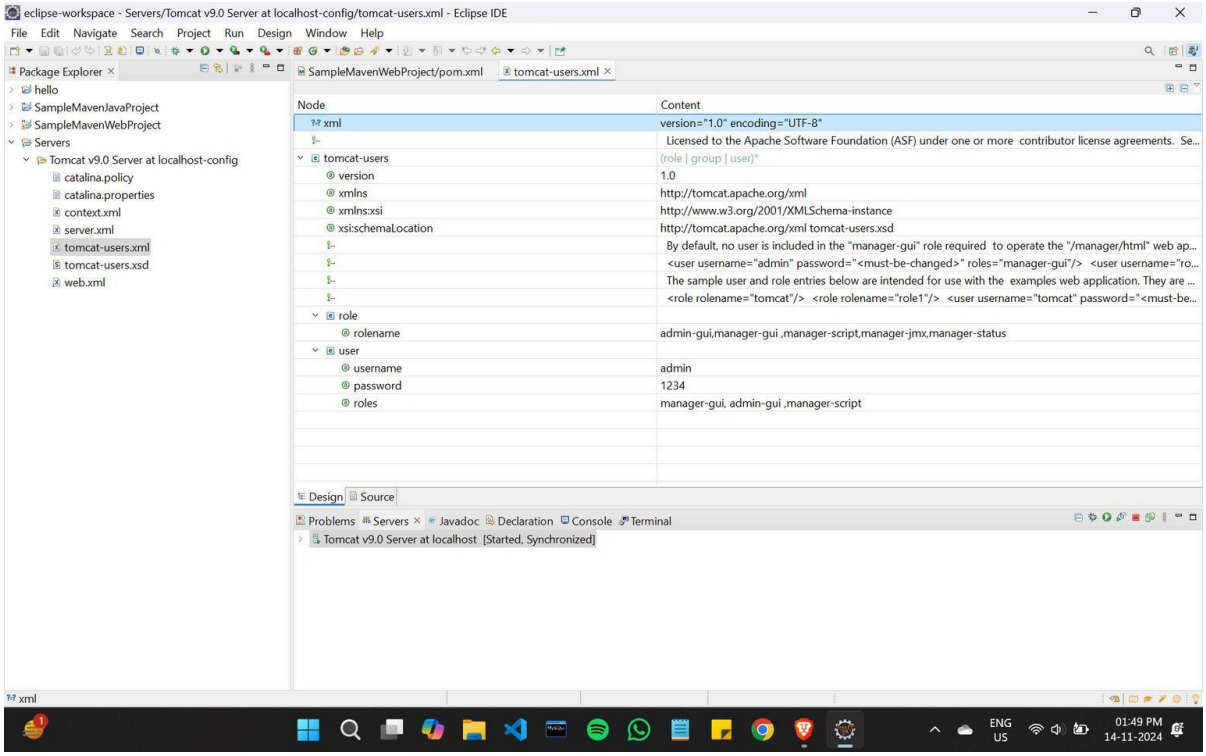
Add test project



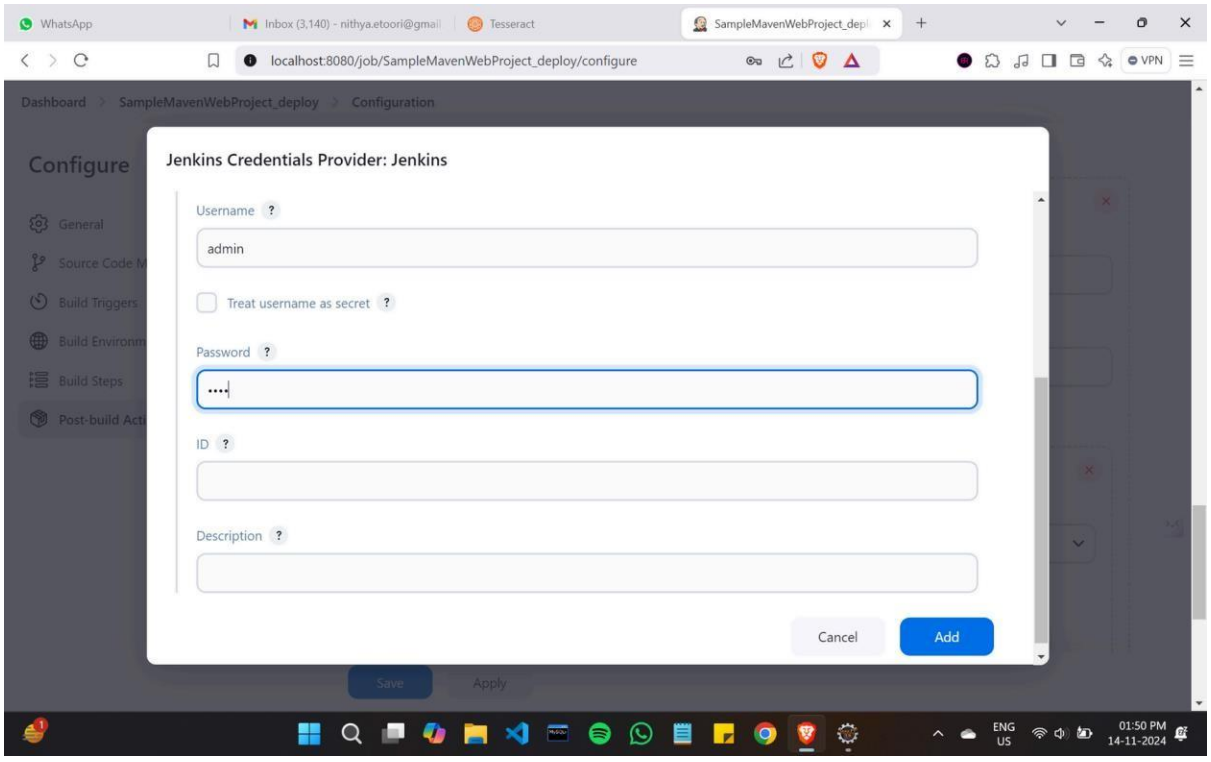
Post Build Actions



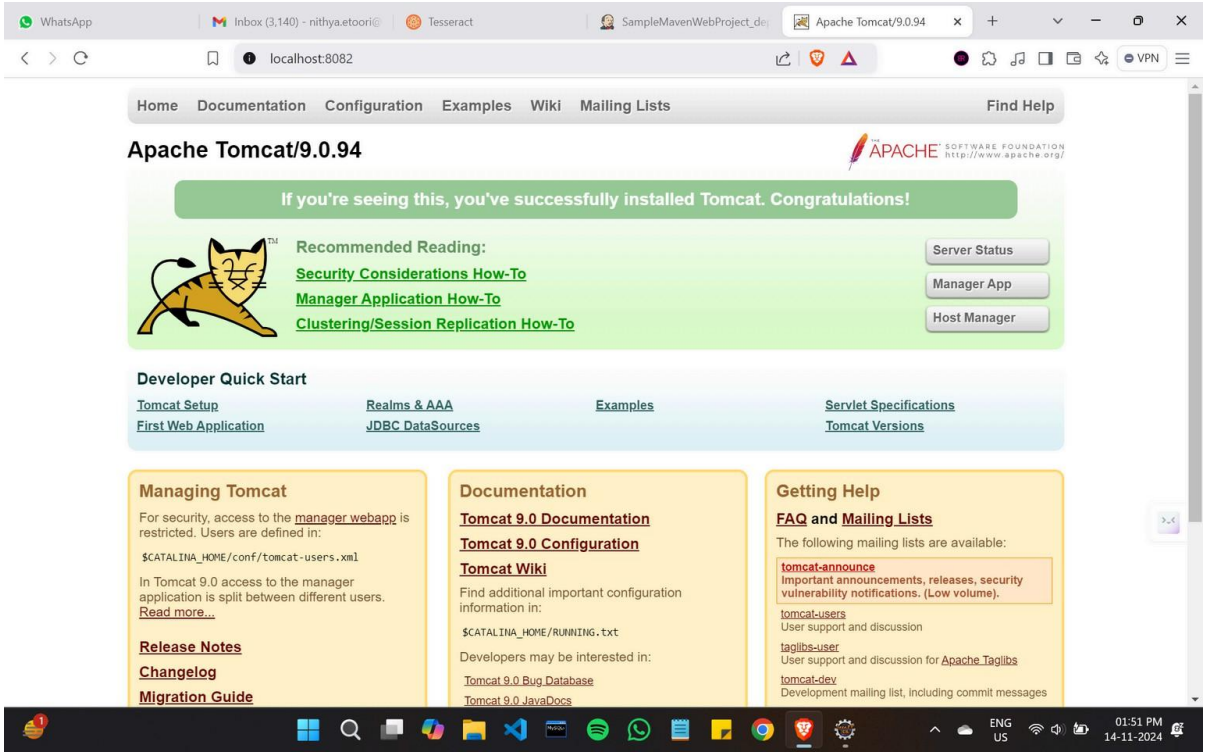
Check tomcat credential s from Eclipse and start server



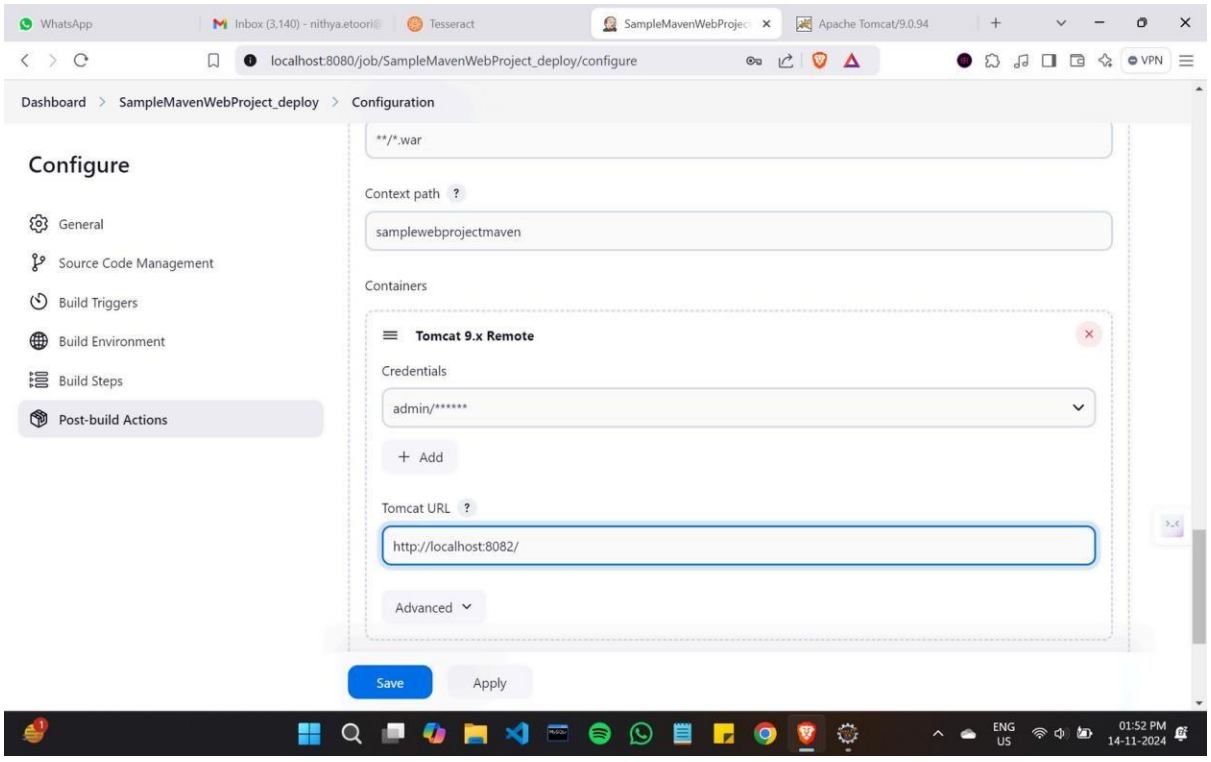
Add the credentials in jenkins



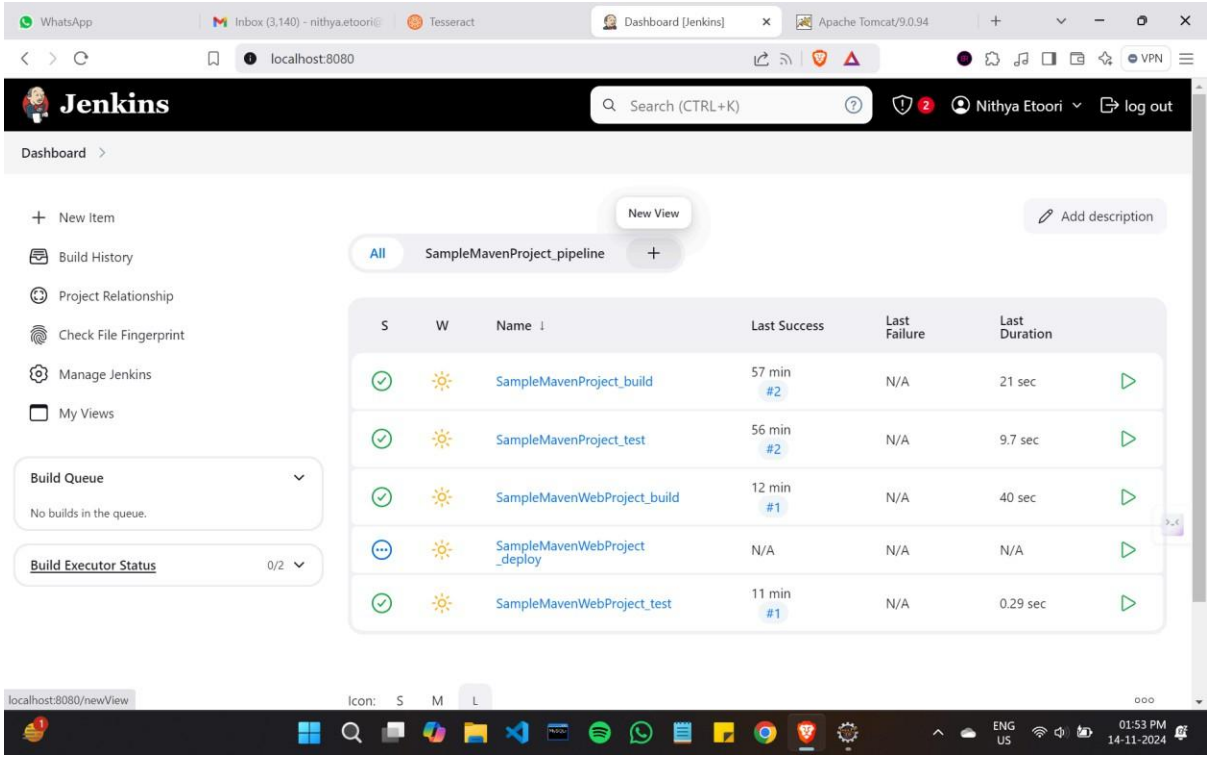
Tomcat is running on 8082



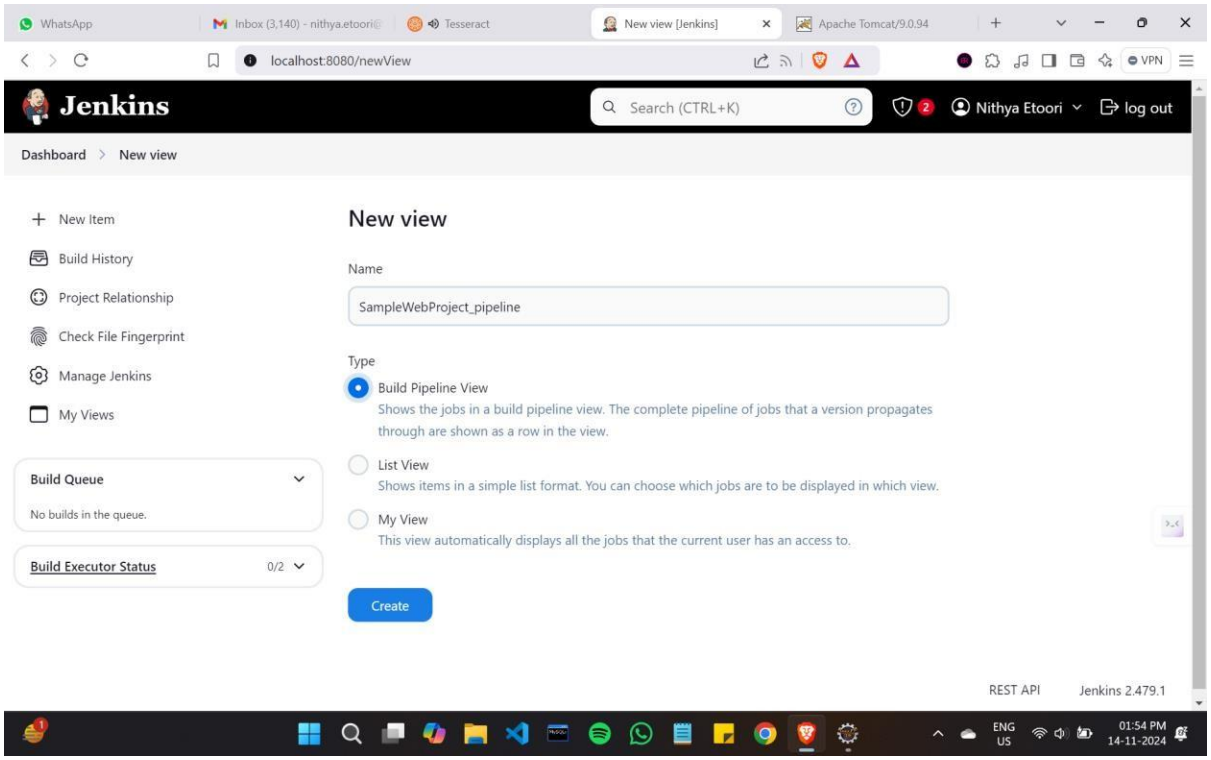
Add tomcat url



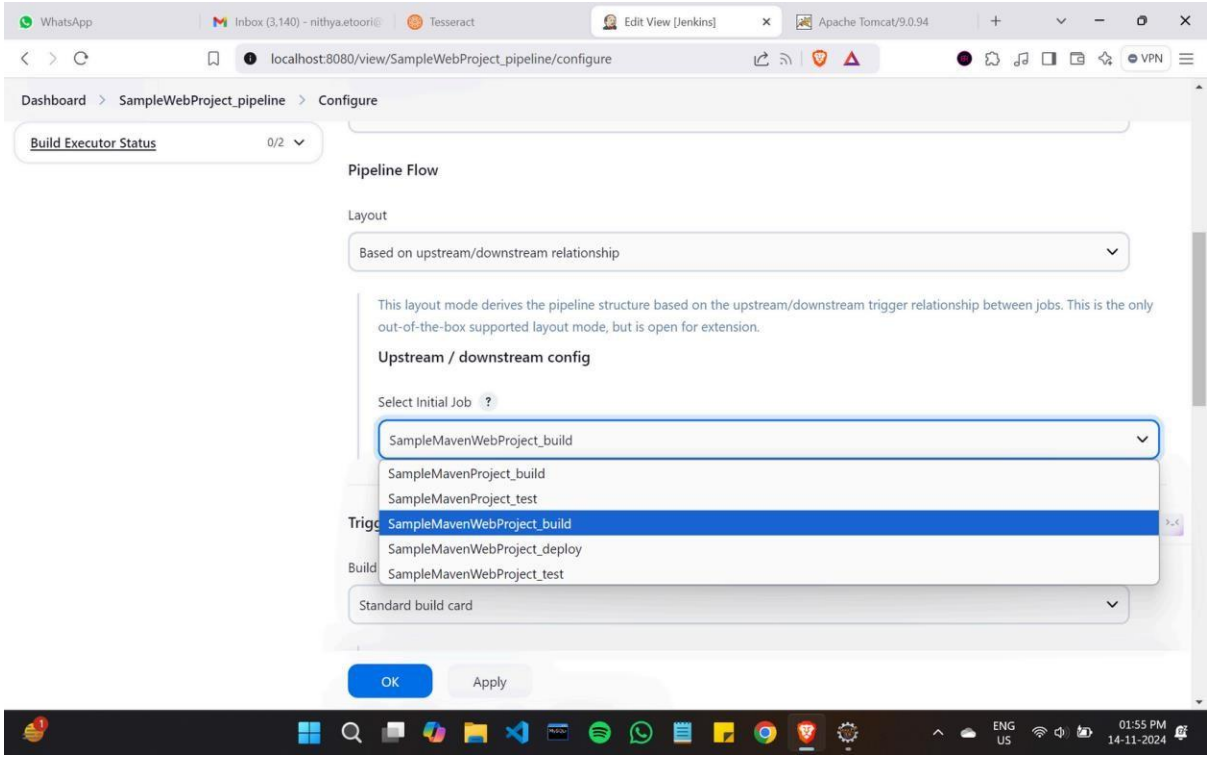
Create a pipeline



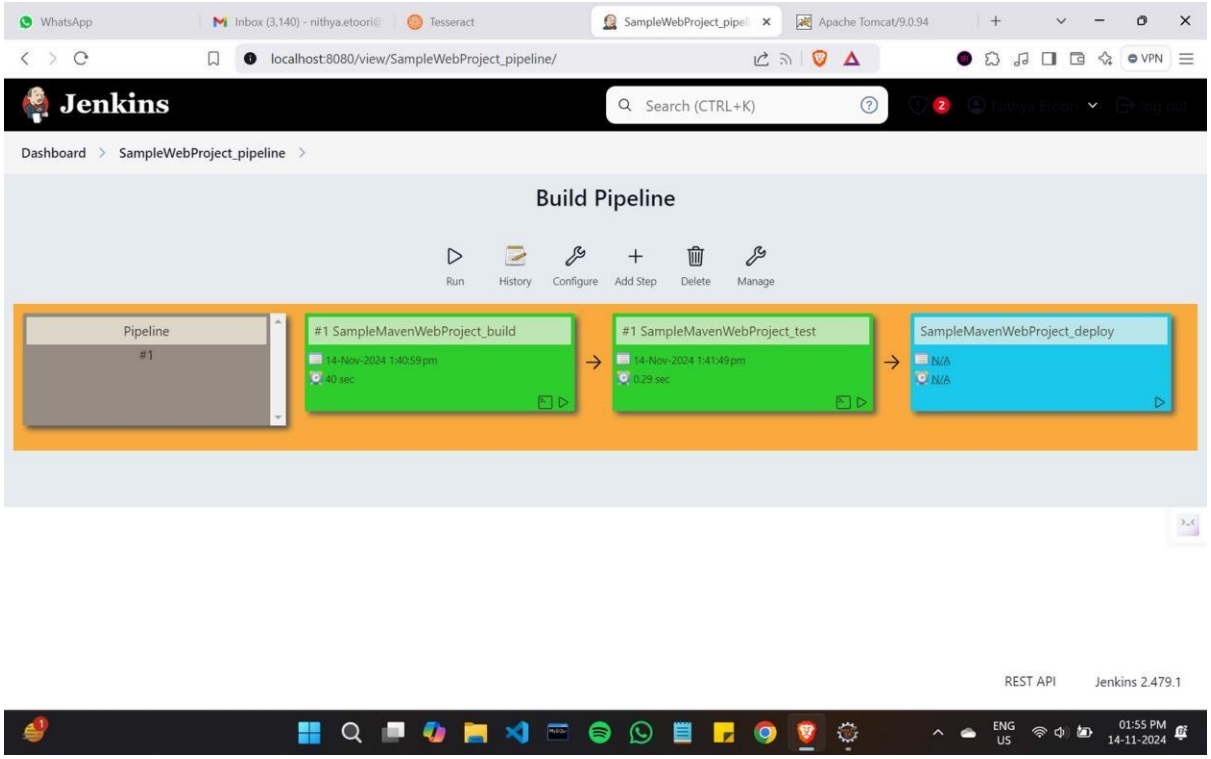
Pipeline for project



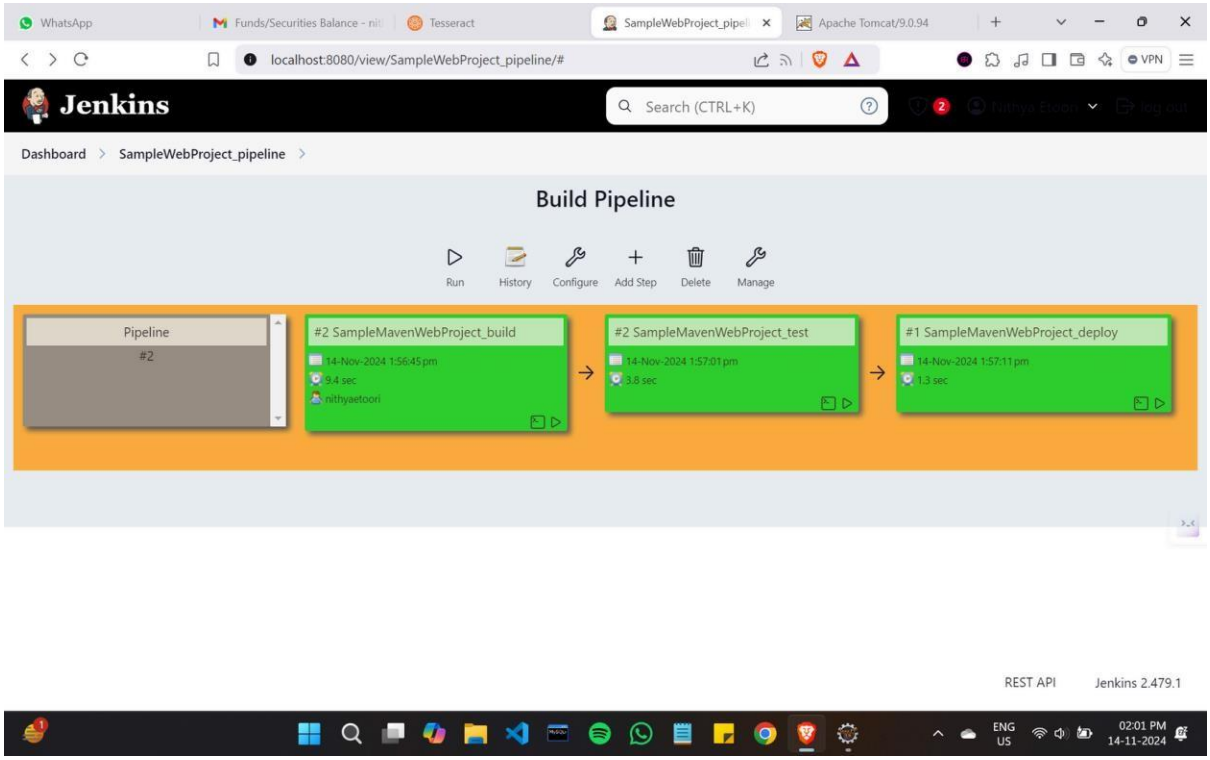
Select the build project



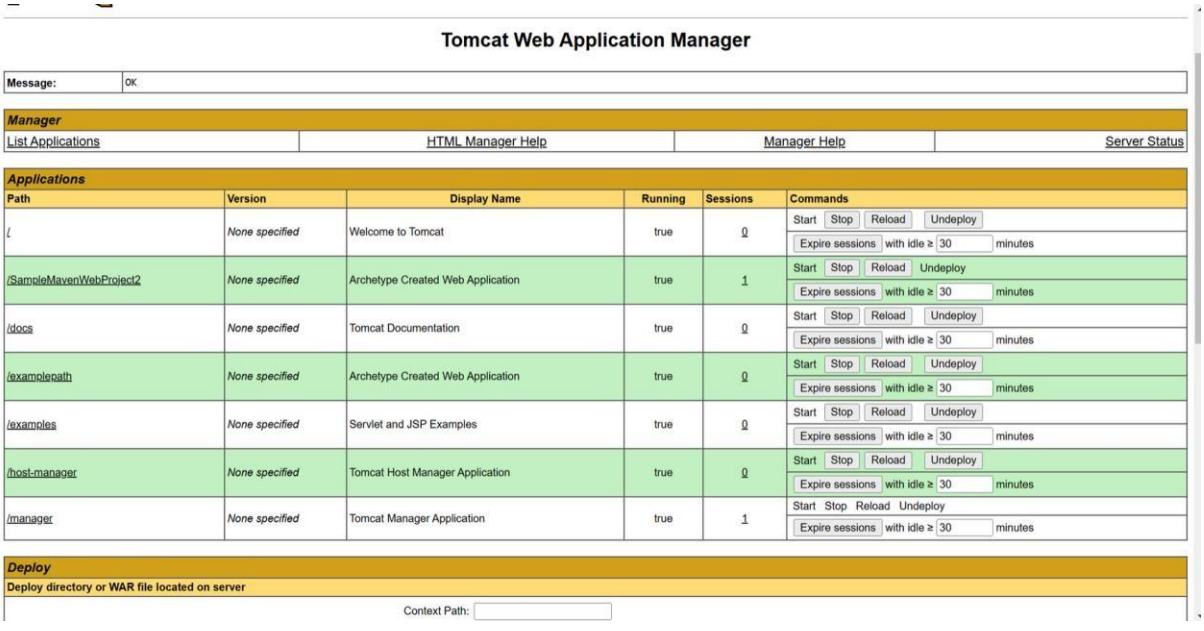
Build pipeline



Successful completion of pipeline



Tomcat Manager App



Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

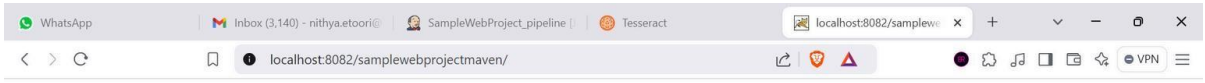
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SampleMavenWebProject2	None specified	Archetype Created Web Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes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

Deploy

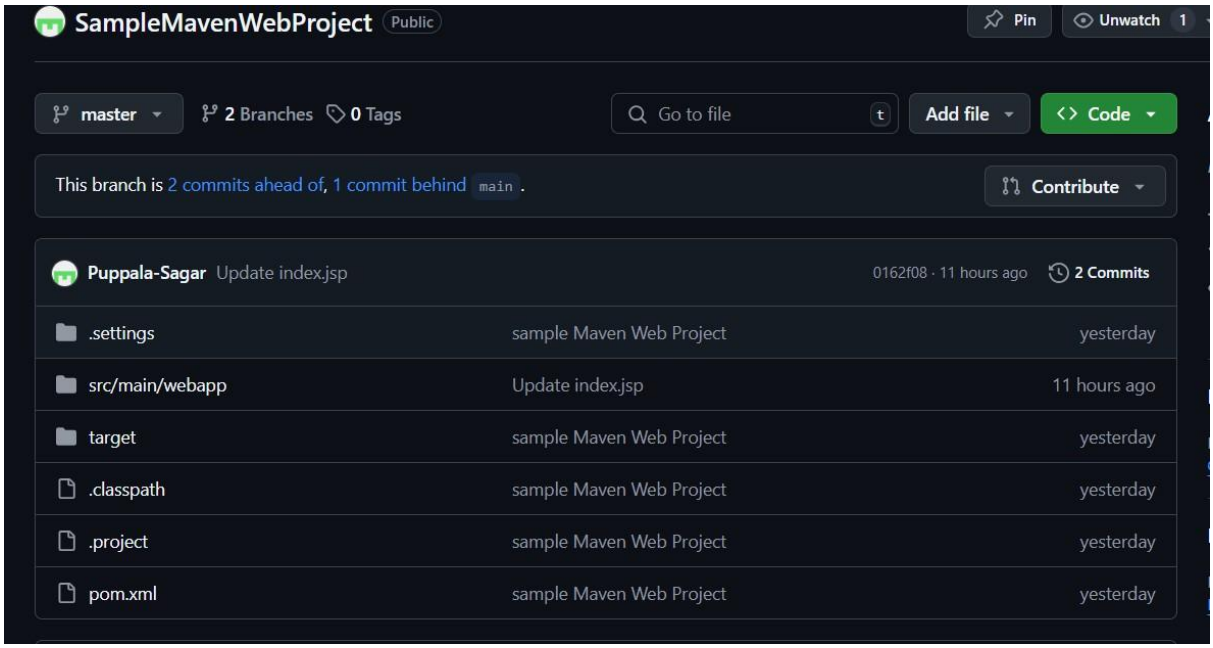
Deploy directory or WAR file located on server

Context Path:

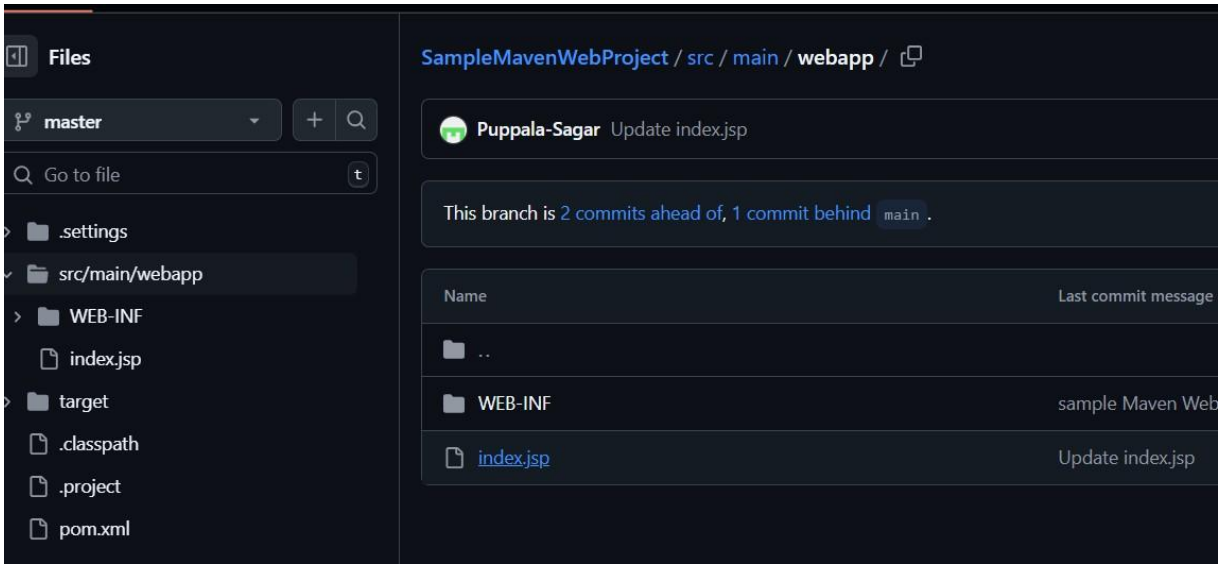
Project working on Sever



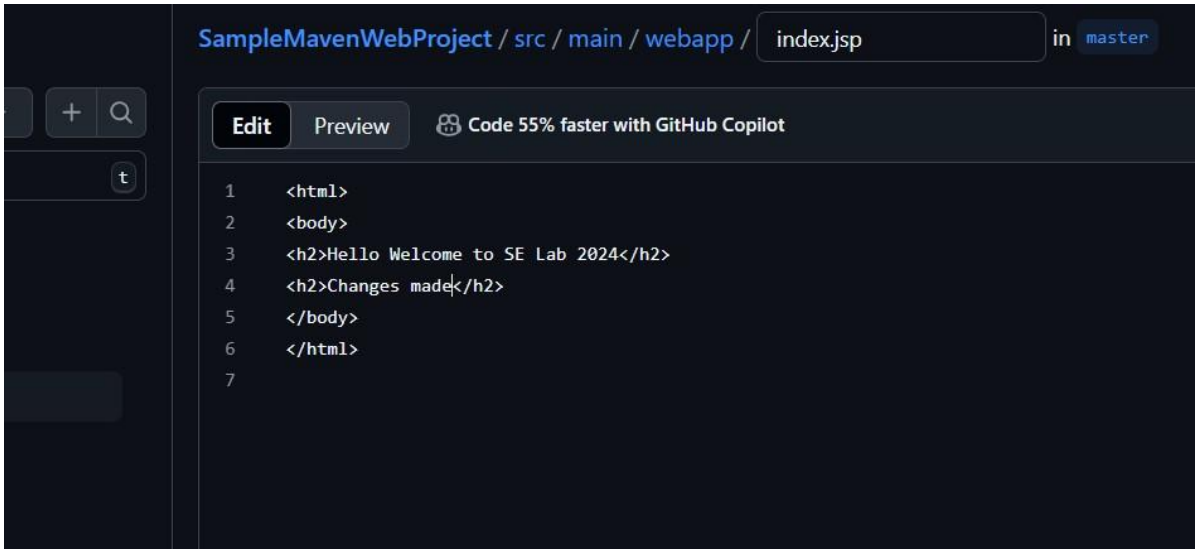
Git repo



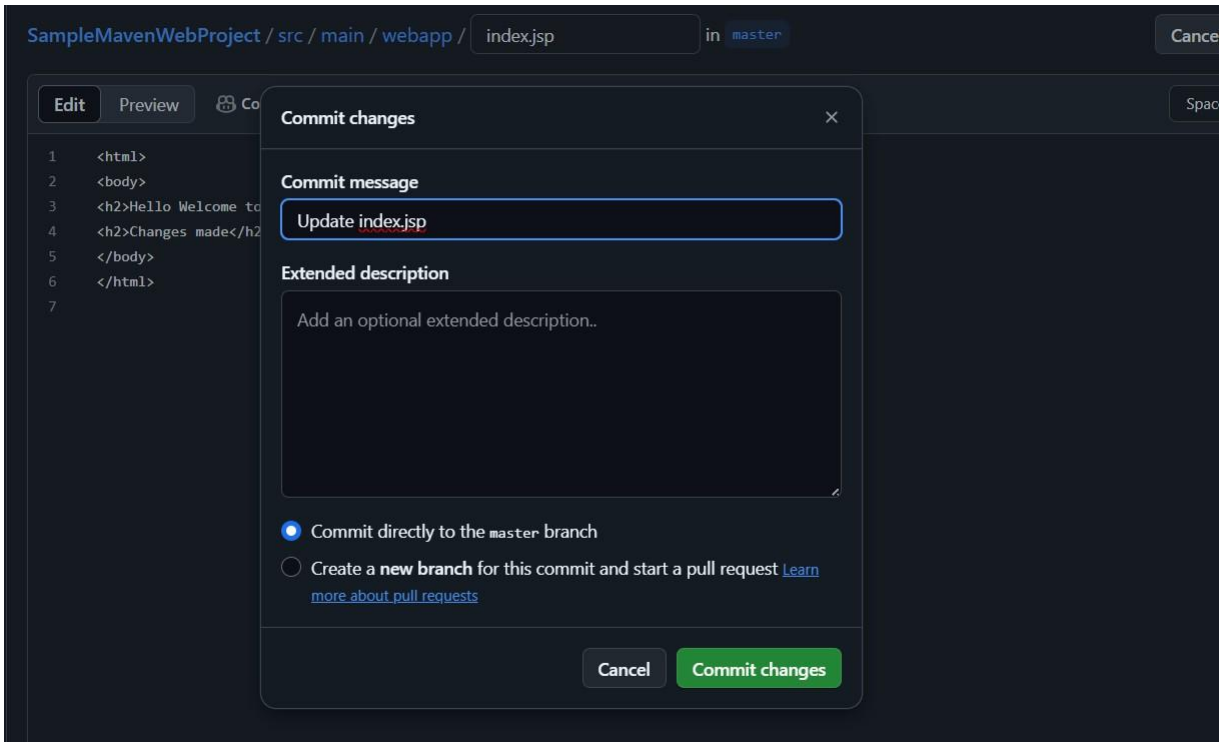
Select index.jsp



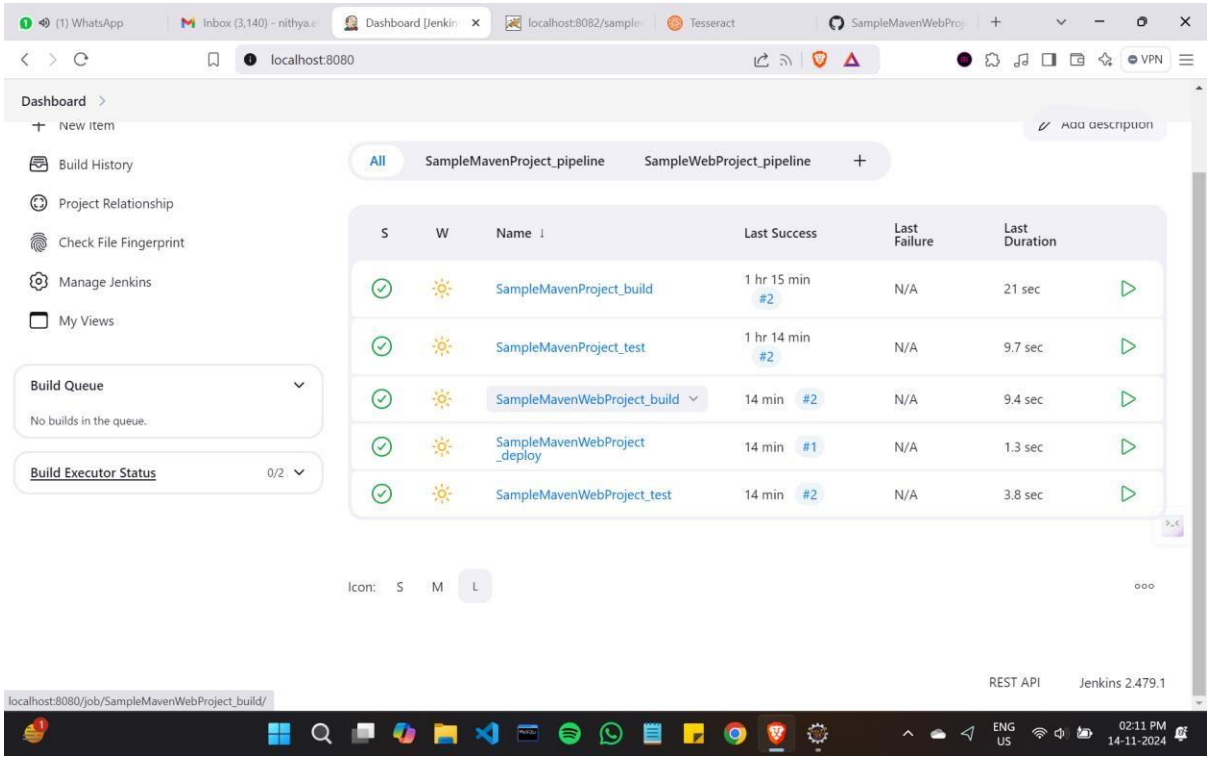
Make changes to the jsp file



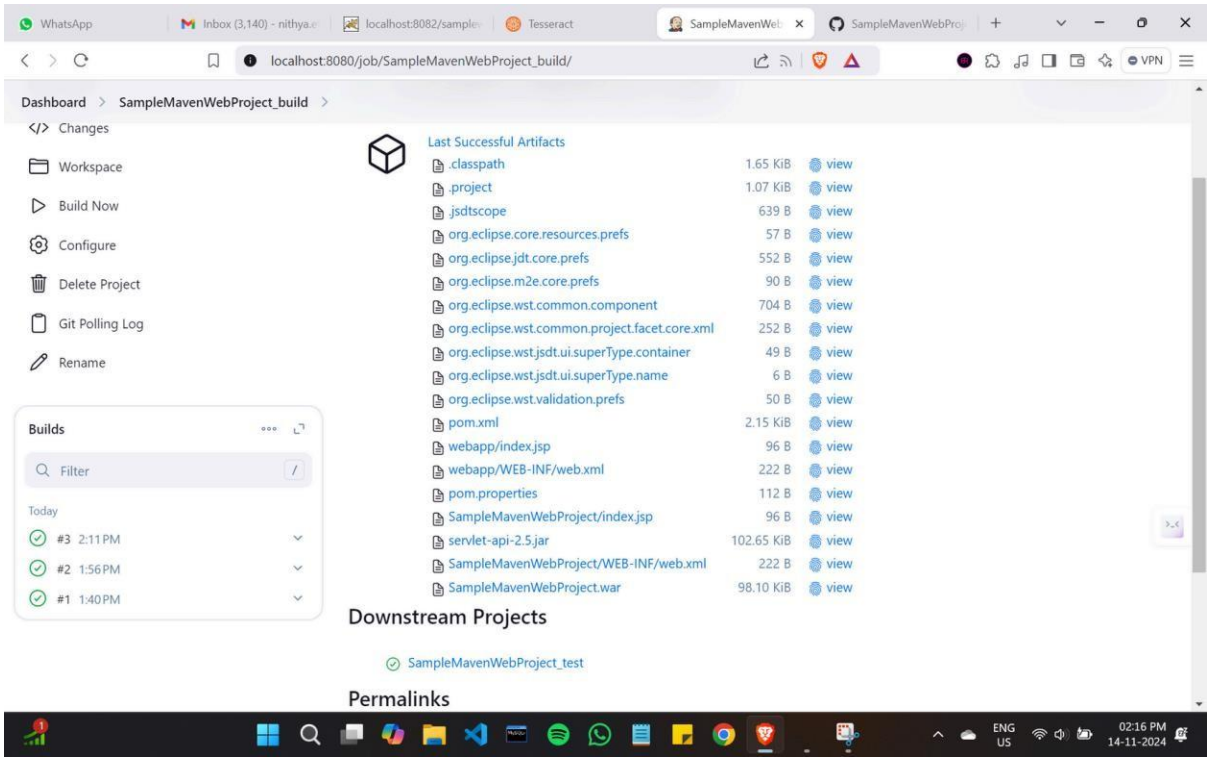
Commit changes



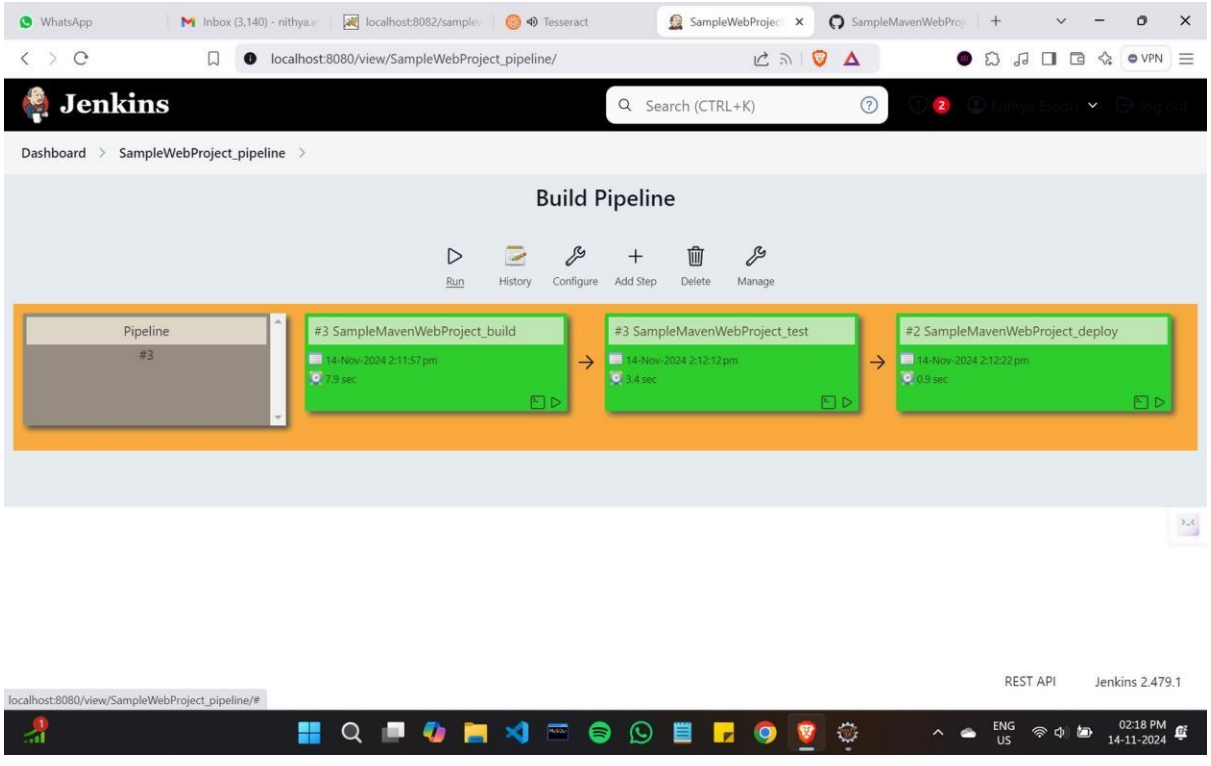
Check for changes in build project



Check builds for changes



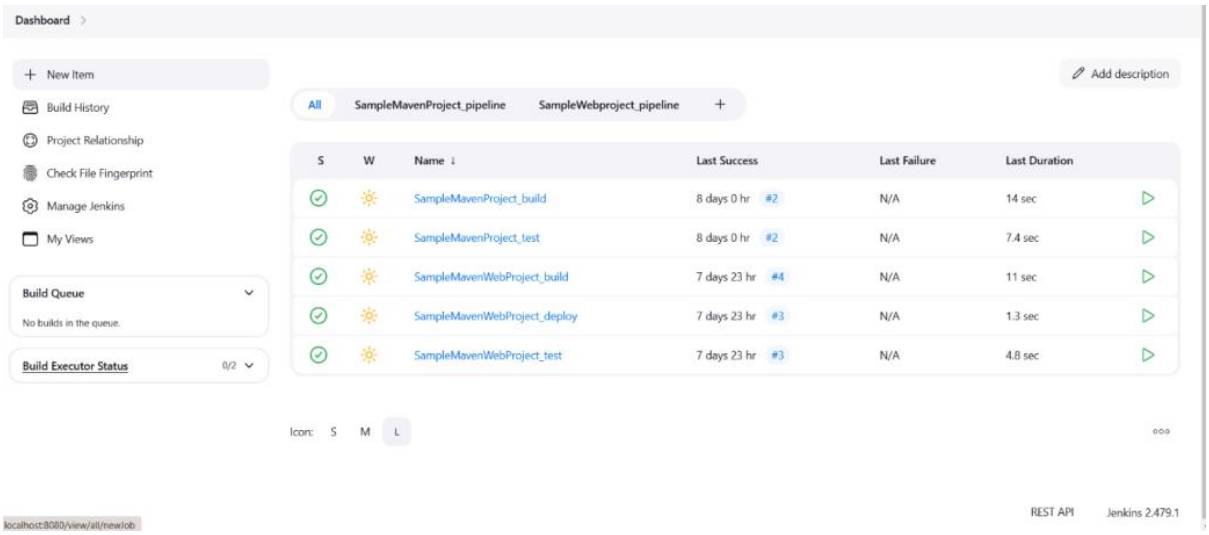
Build pipeline after changes



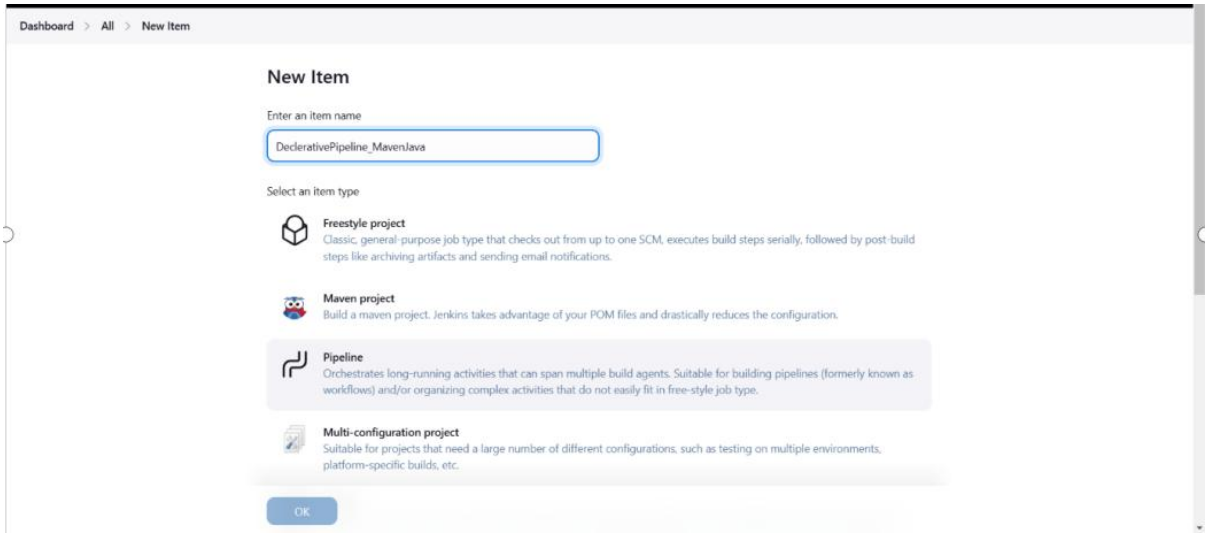
Result on server after changes

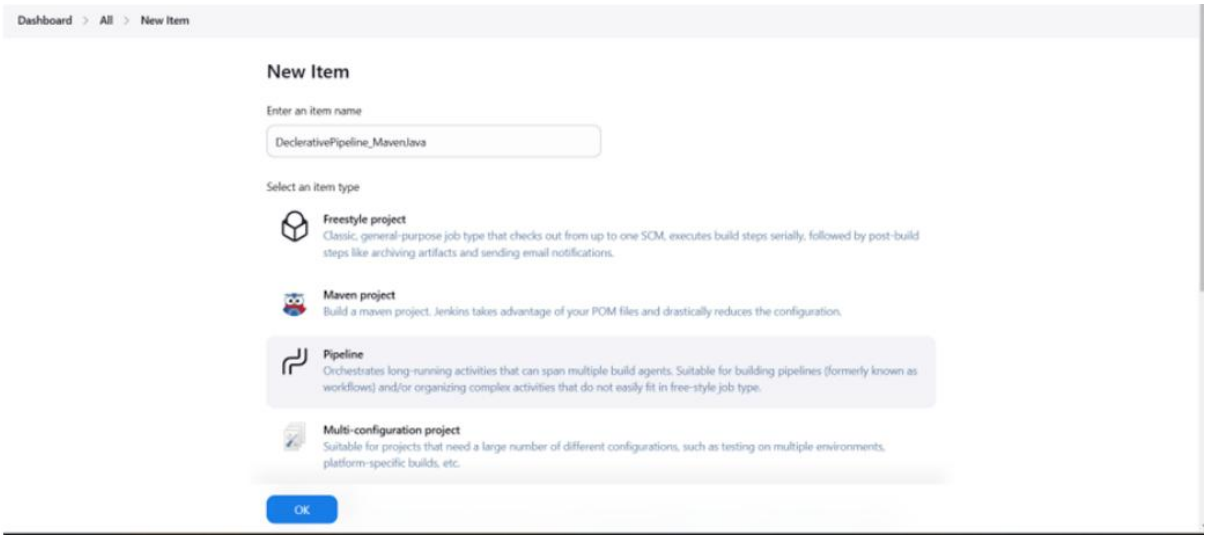
CI/CD Scripted Pipelines for Maven Java Project

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

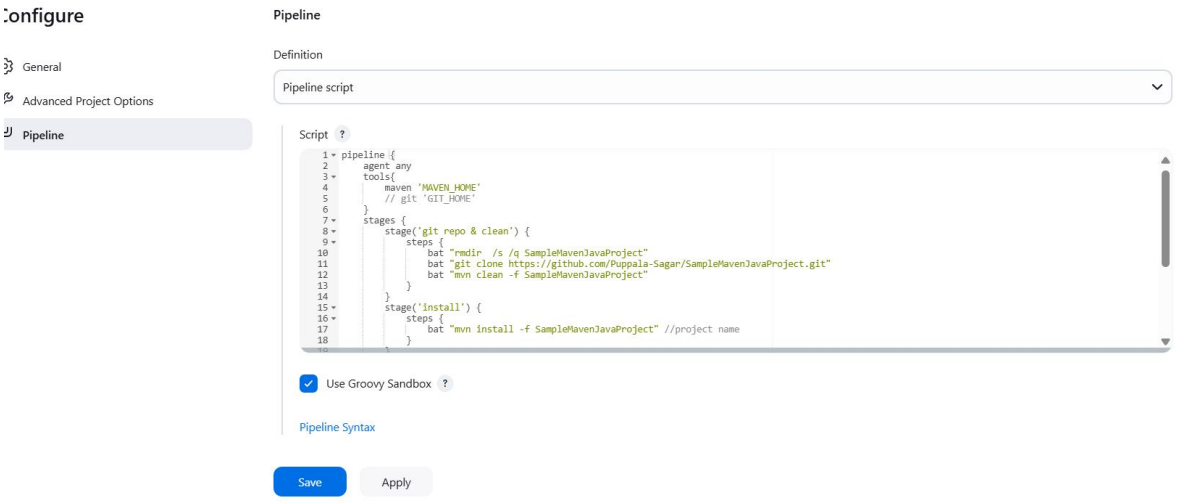


Step 2: Enter a name like DeclarativePipeline_MavenJava and select Pipeline and click ok.

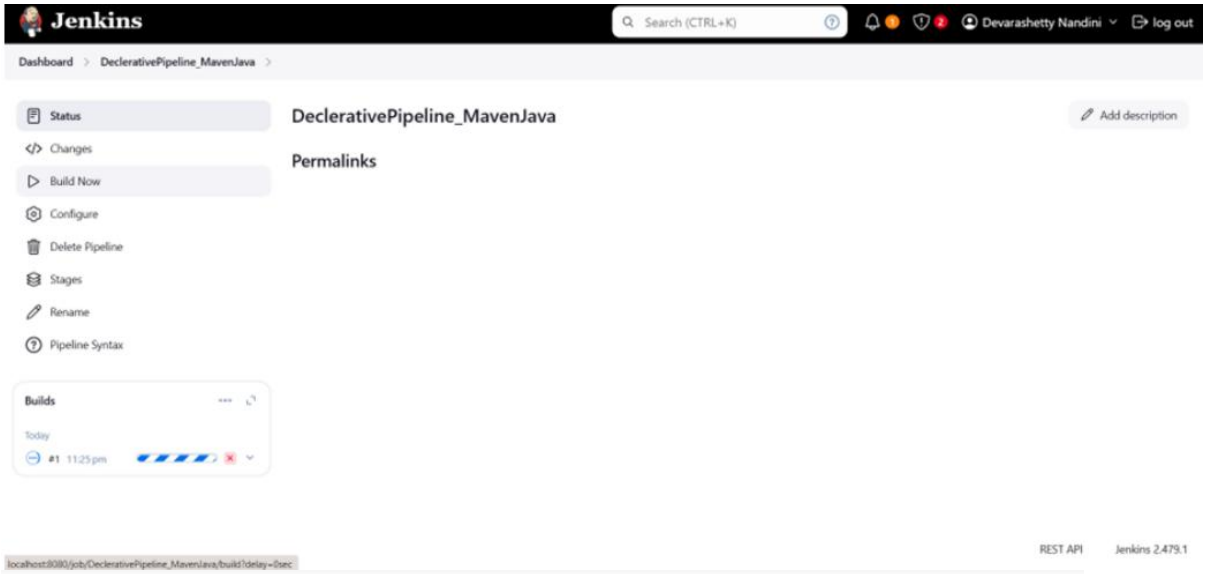


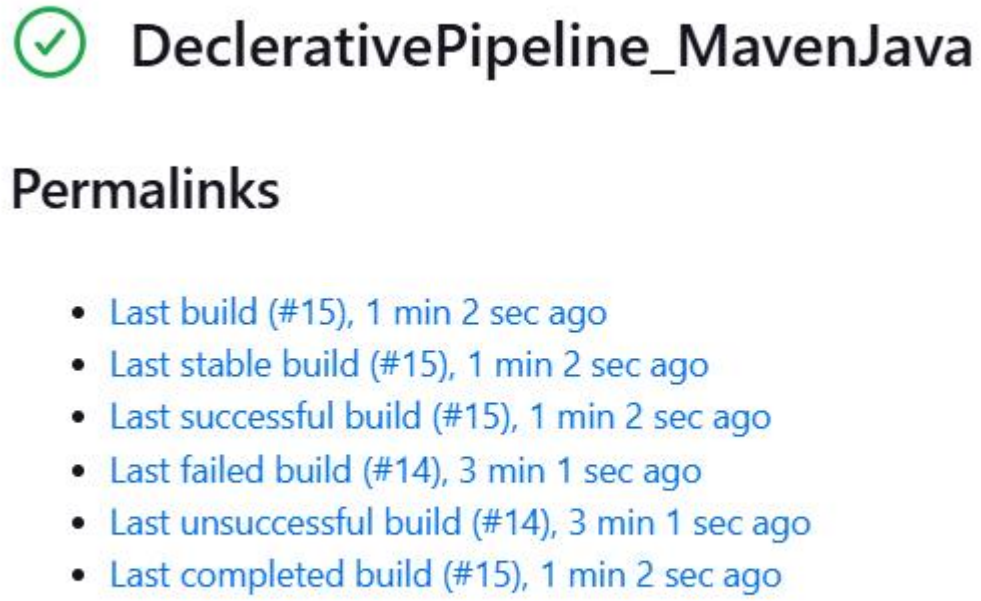


Step 3: Paste the script in the Pipeline script and click Apply and then Save.

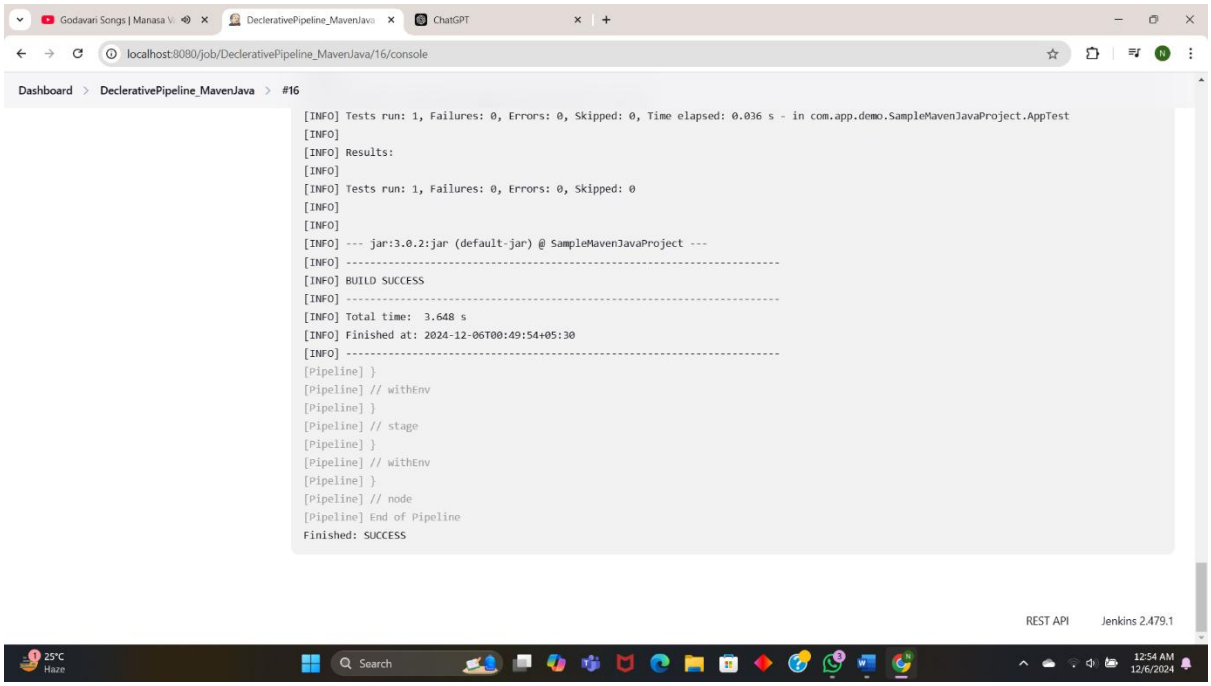


Step 4: Click on Build Now.

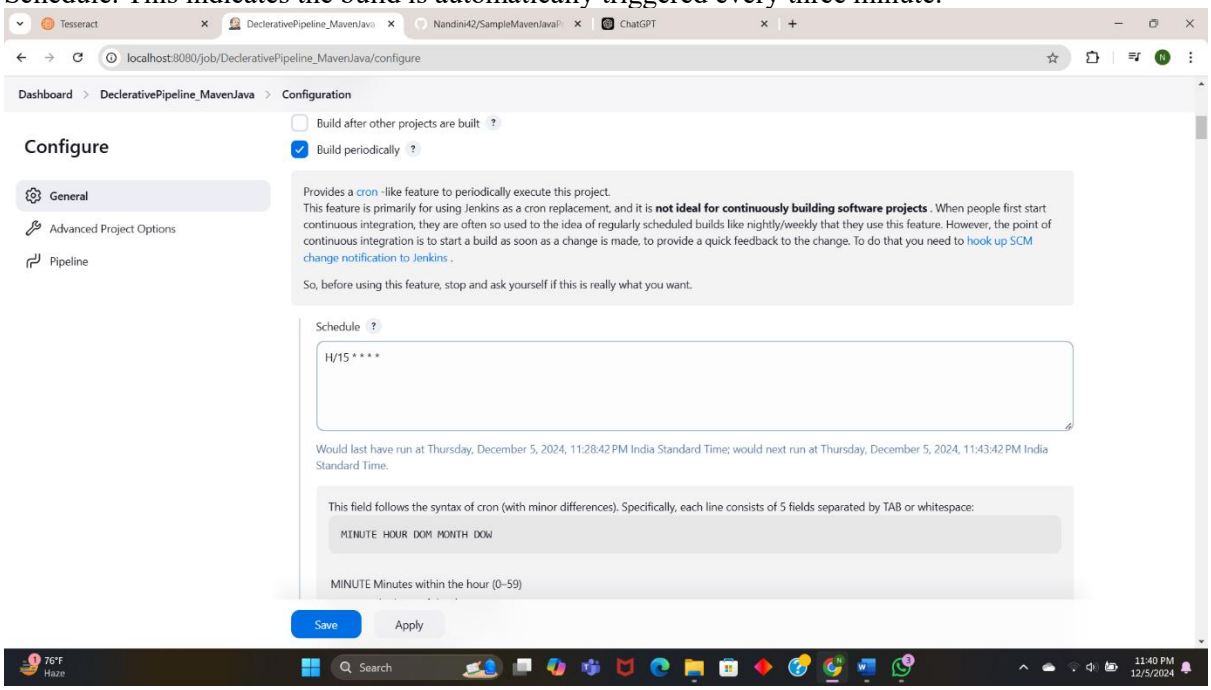


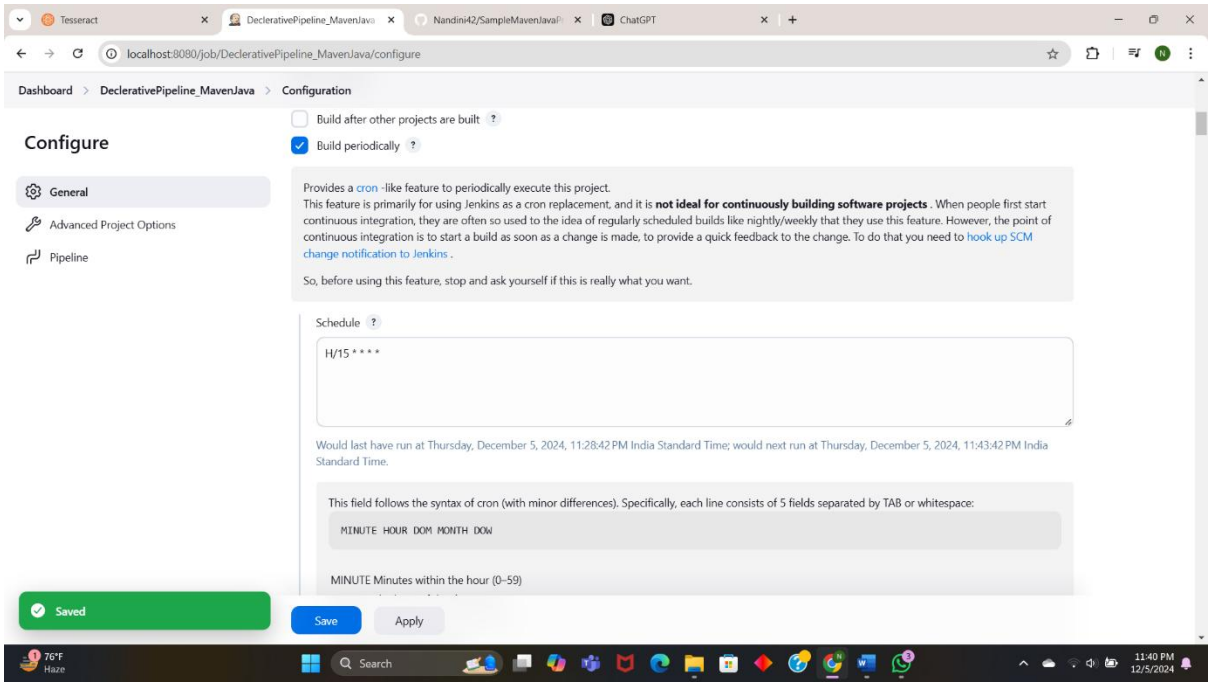


Step 5: If it turns green then the build is successful.

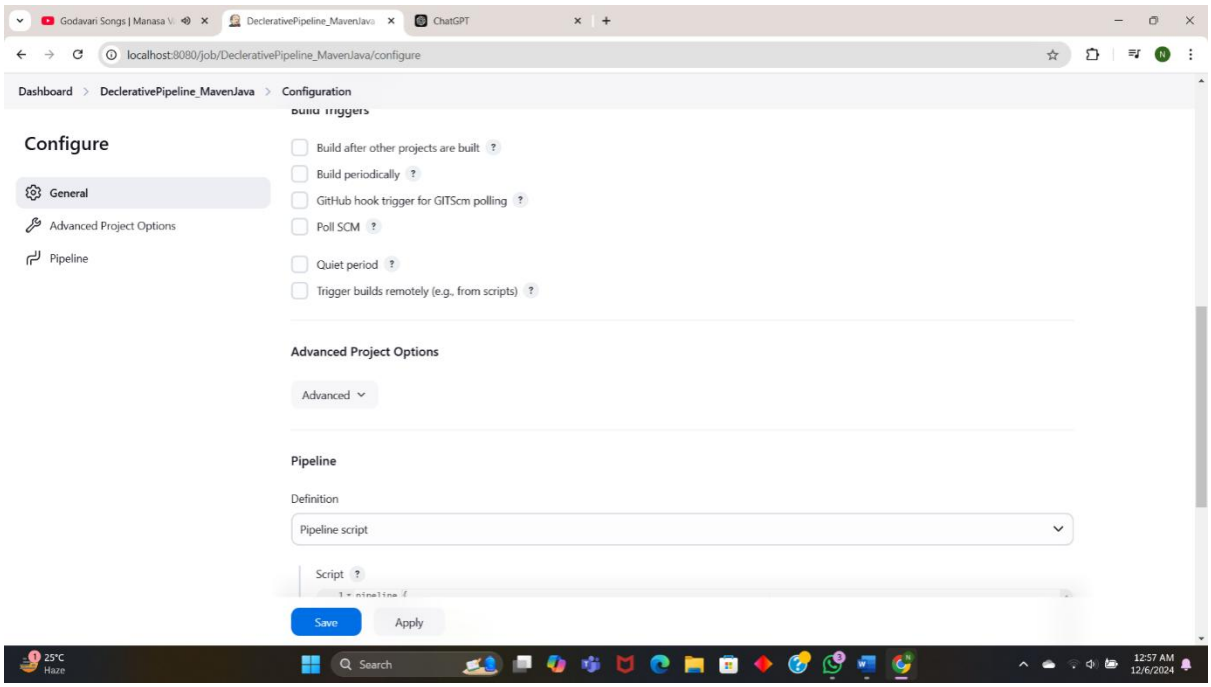


Step 6: Click on configuration and check the Build periodically in Build Triggers. Enter "H/3* * * * *" in the Schedule. This indicates the build is automatically triggered every three minute.

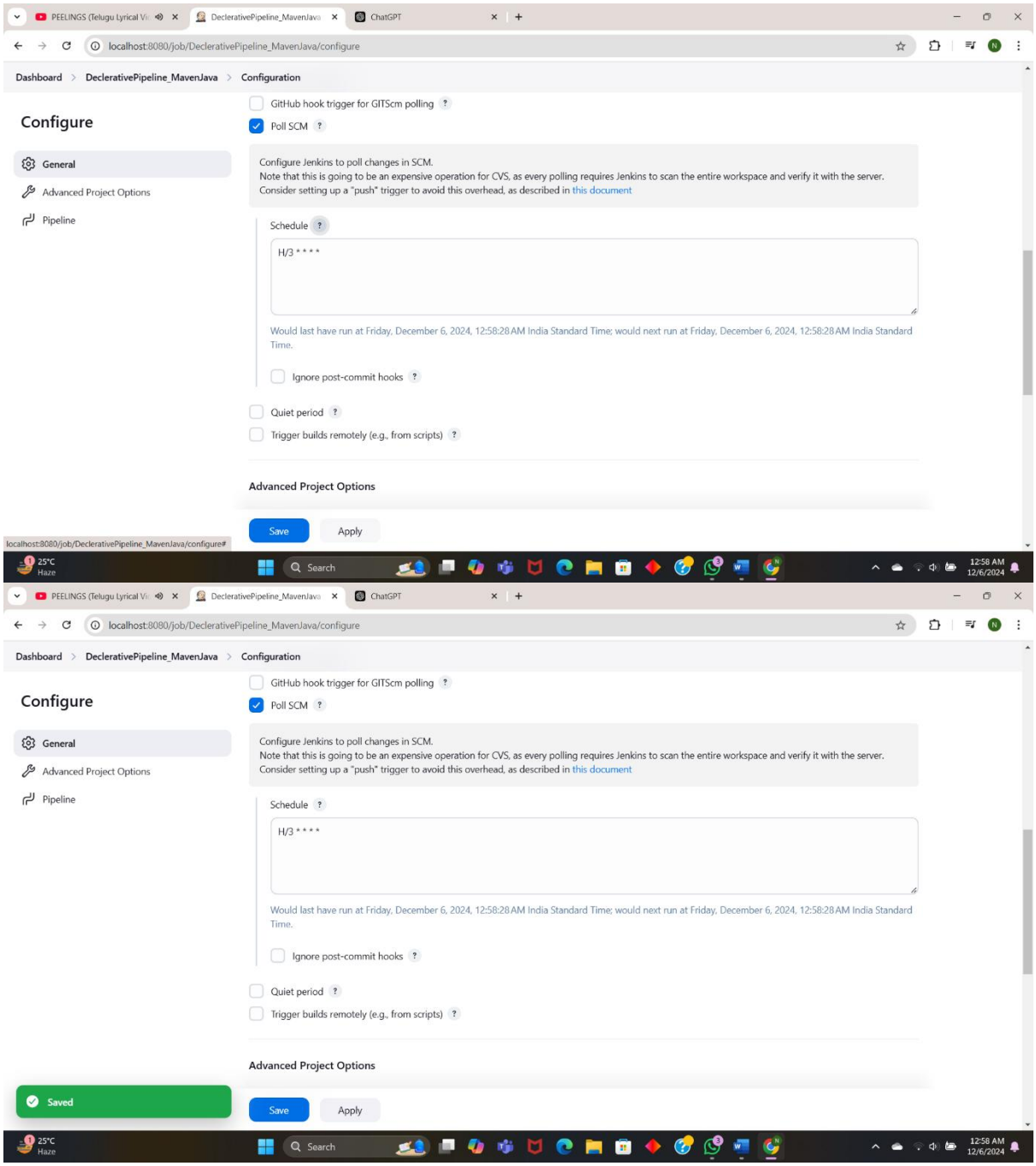




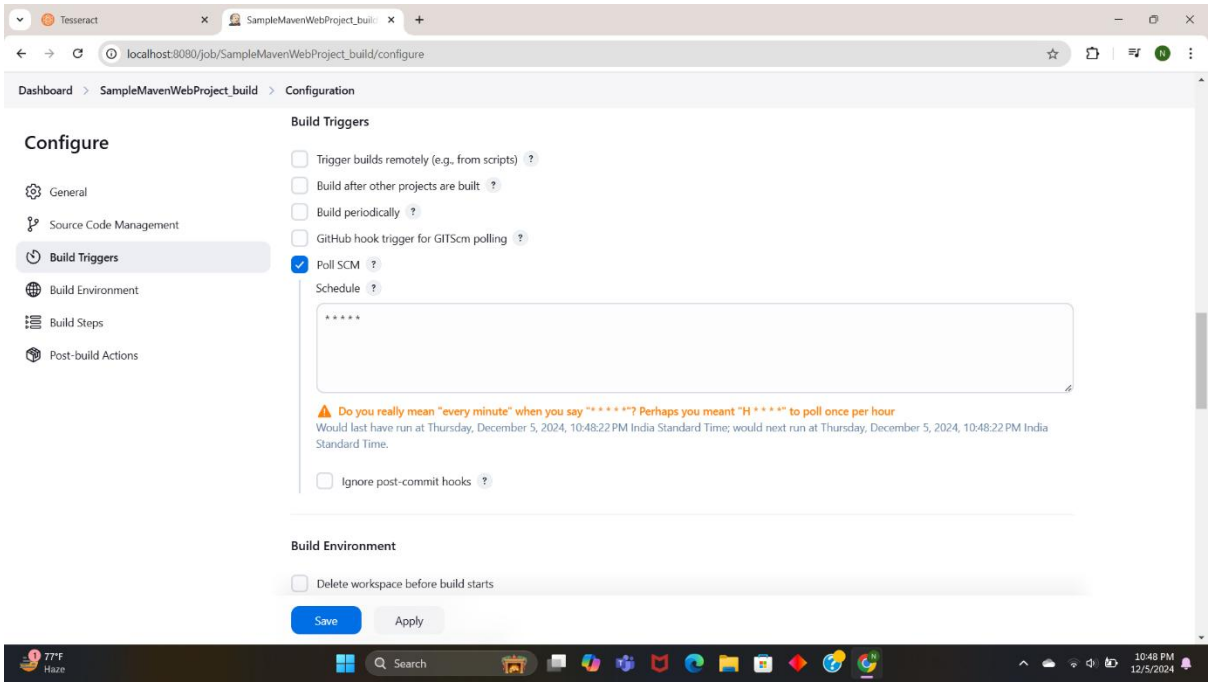
Step 7: Uncheck the Build Periodically in Build Trigger to stop the automation of build every minute.



Step-09 : enable poll SCM

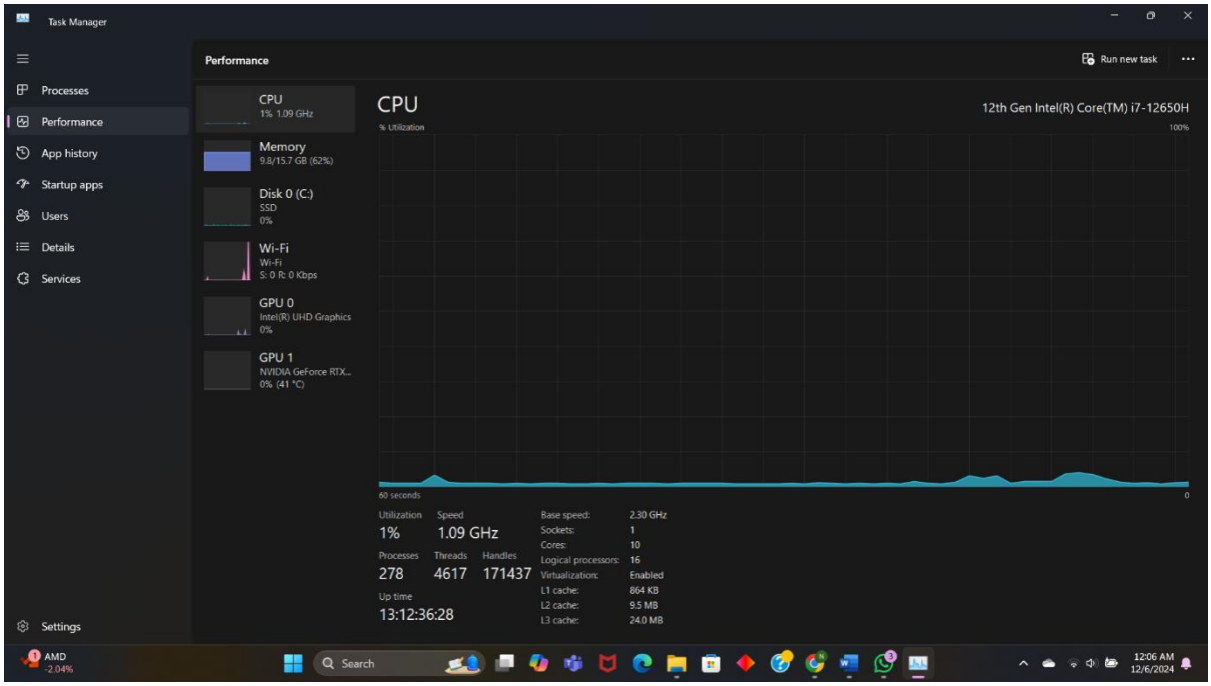


Freestyle pipeline for java web project



Installation of Docker

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



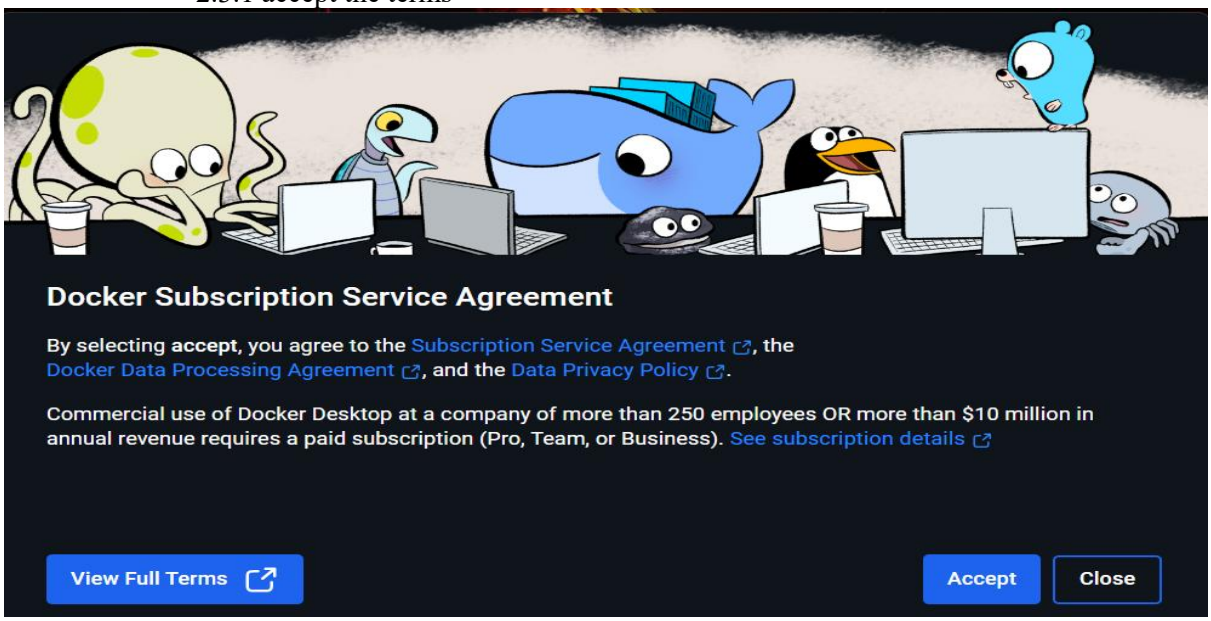
Step 2 : install Docker Desktop

2.1 go to Dcokers website and download Docker Desktop for windows

2.2 Double click the downloaded file to start the installations

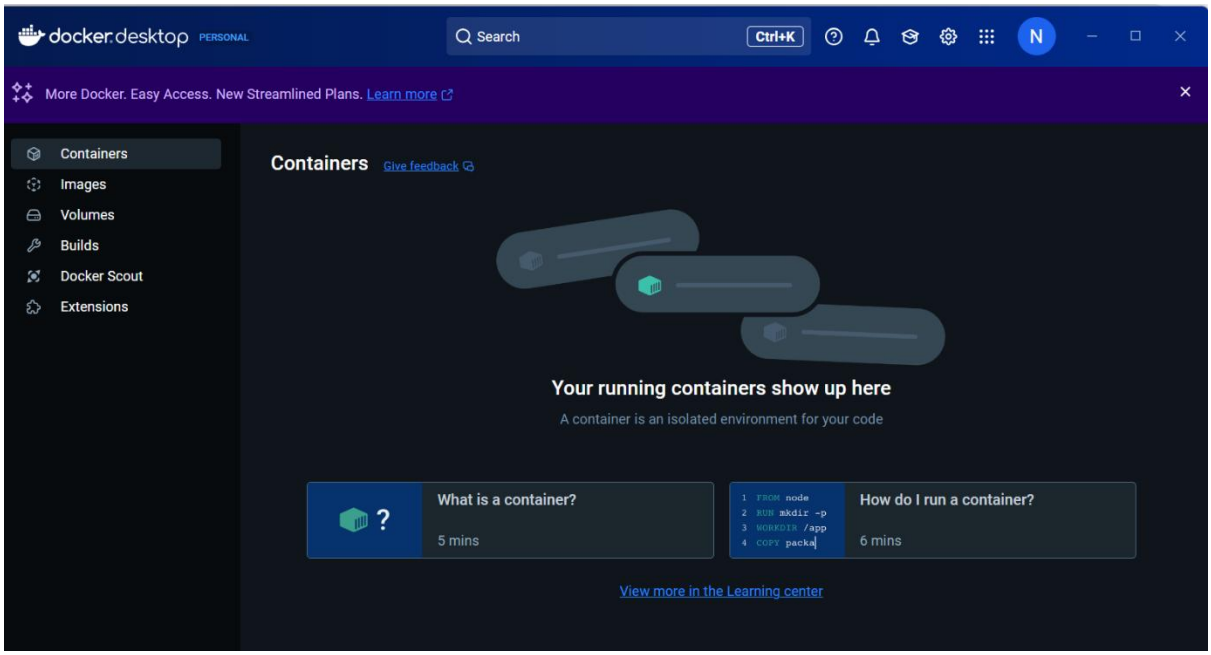
2.3 follow the steps in the installer

2.3.1 accept the terms

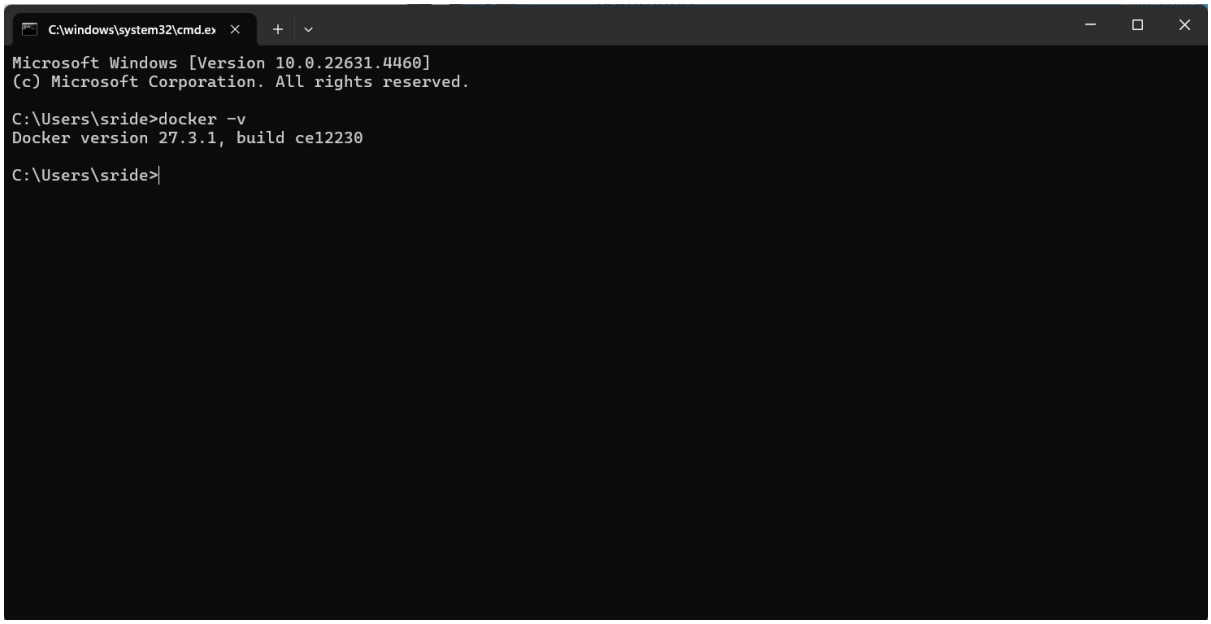


2.3.2 choose to enable WSL2 instead of Hyper-V

2.3.3 install and wait

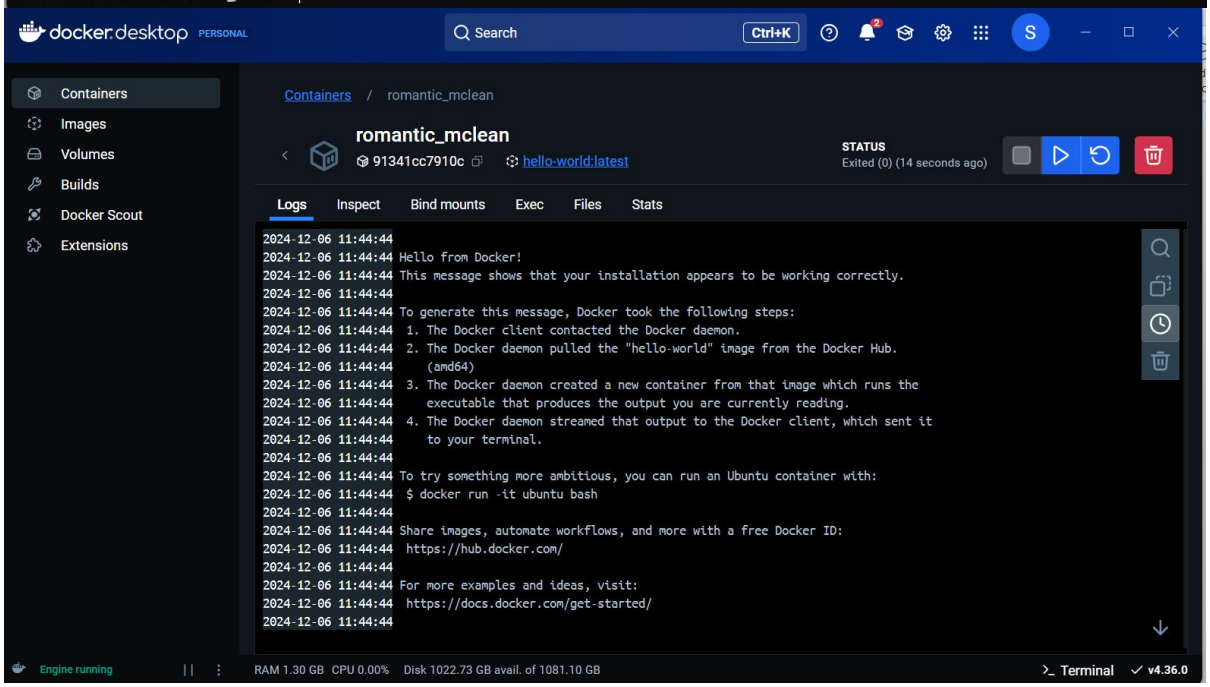


Docker version check



Dokcer run helloworld

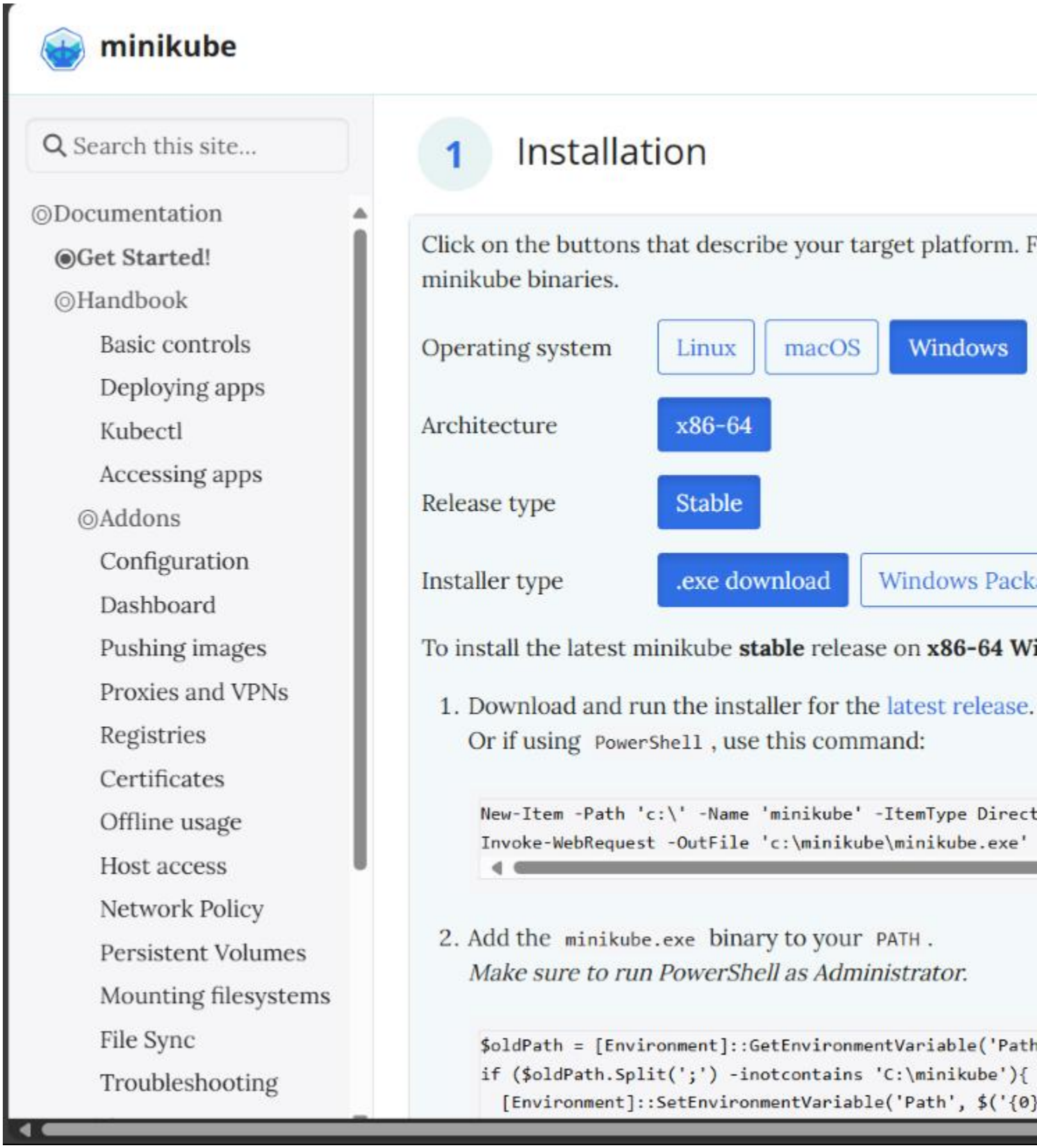
```
PS C:\Users\sagar> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

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>
```



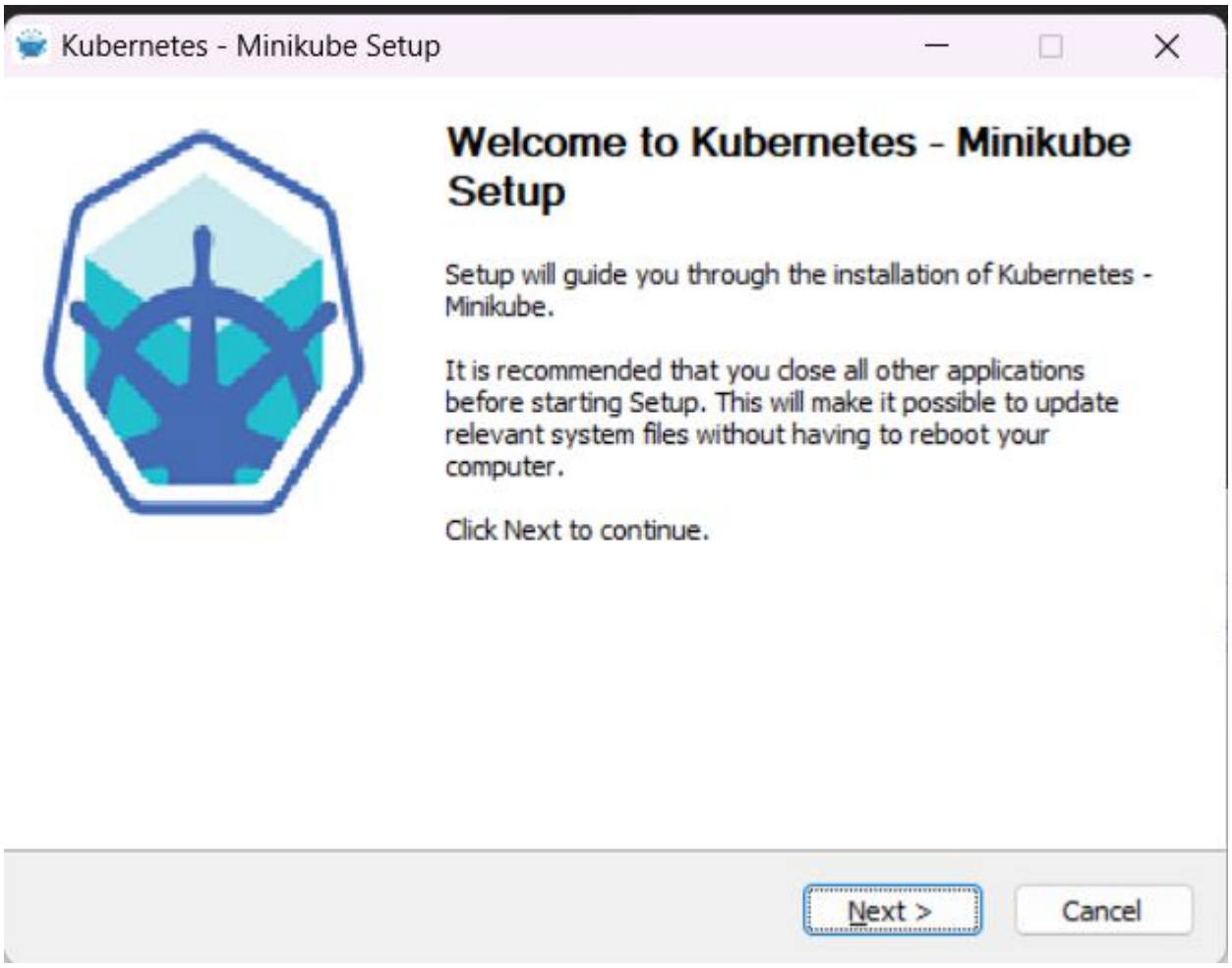
Minikube Installation\

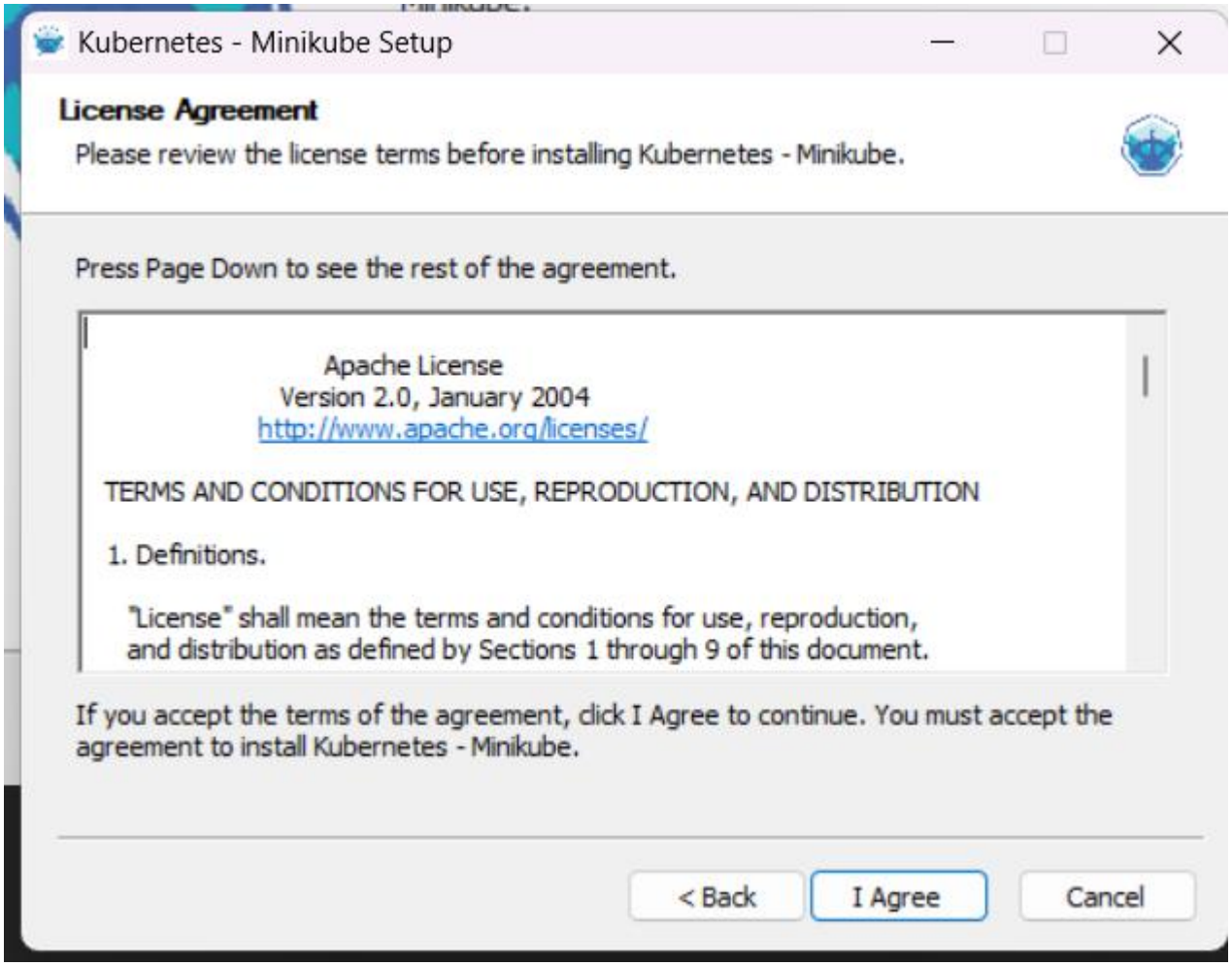
Step 1: Go to https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-

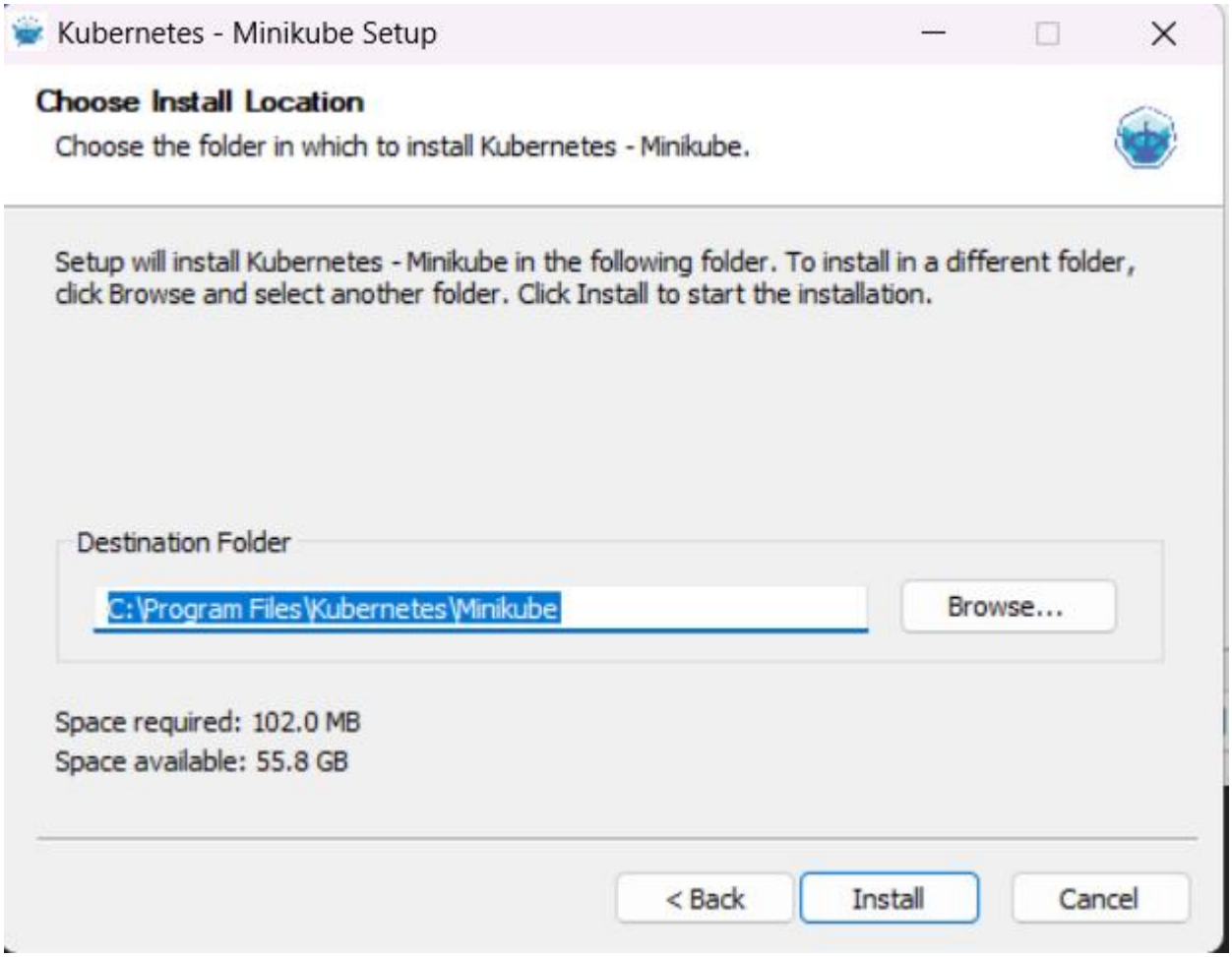
64%2Fstable%2F.exe+download and select the appropriate installer based on your Operating System.

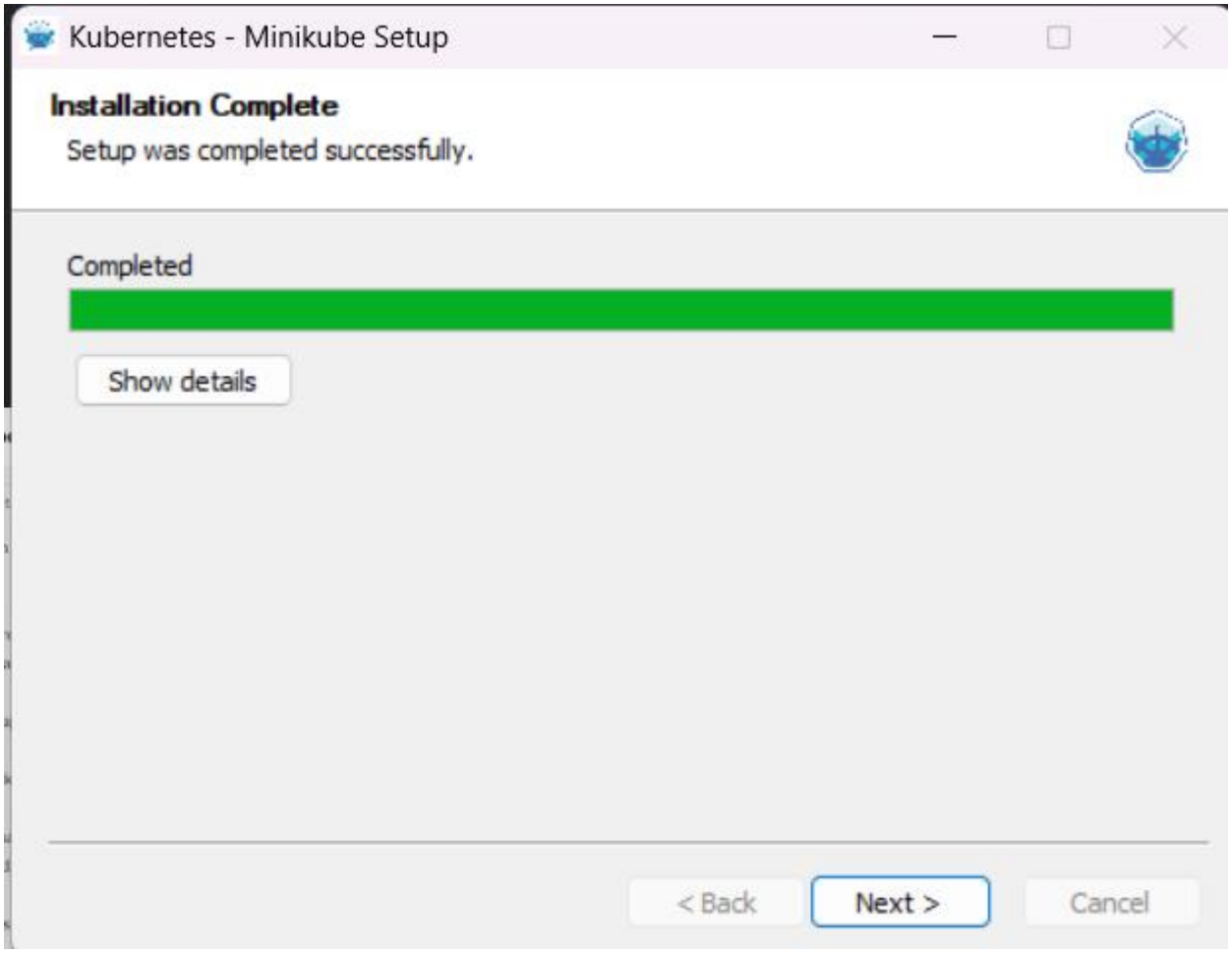


Step 2: Now run the minikube installer and follow the steps below




Kubernetes - Minikube Setup
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
                          NAMES
1a43b939d308   hello-world            "/hello"                 30 seconds ago   Exited (0) 29 seconds ago
                          sleepy_chandrasekhar
91341cc7910c   hello-world            "/hello"                 6 days ago       Exited (0) 6 days ago
                          romantic_mclean
1879b82aa084   kicbase/stable:v0.0.45   "/usr/local/bin/entr…"   8 days ago       Exited (255) 6 days ago    127.0.0.1:32773->22/tc
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
75dffa679c9b: 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
```

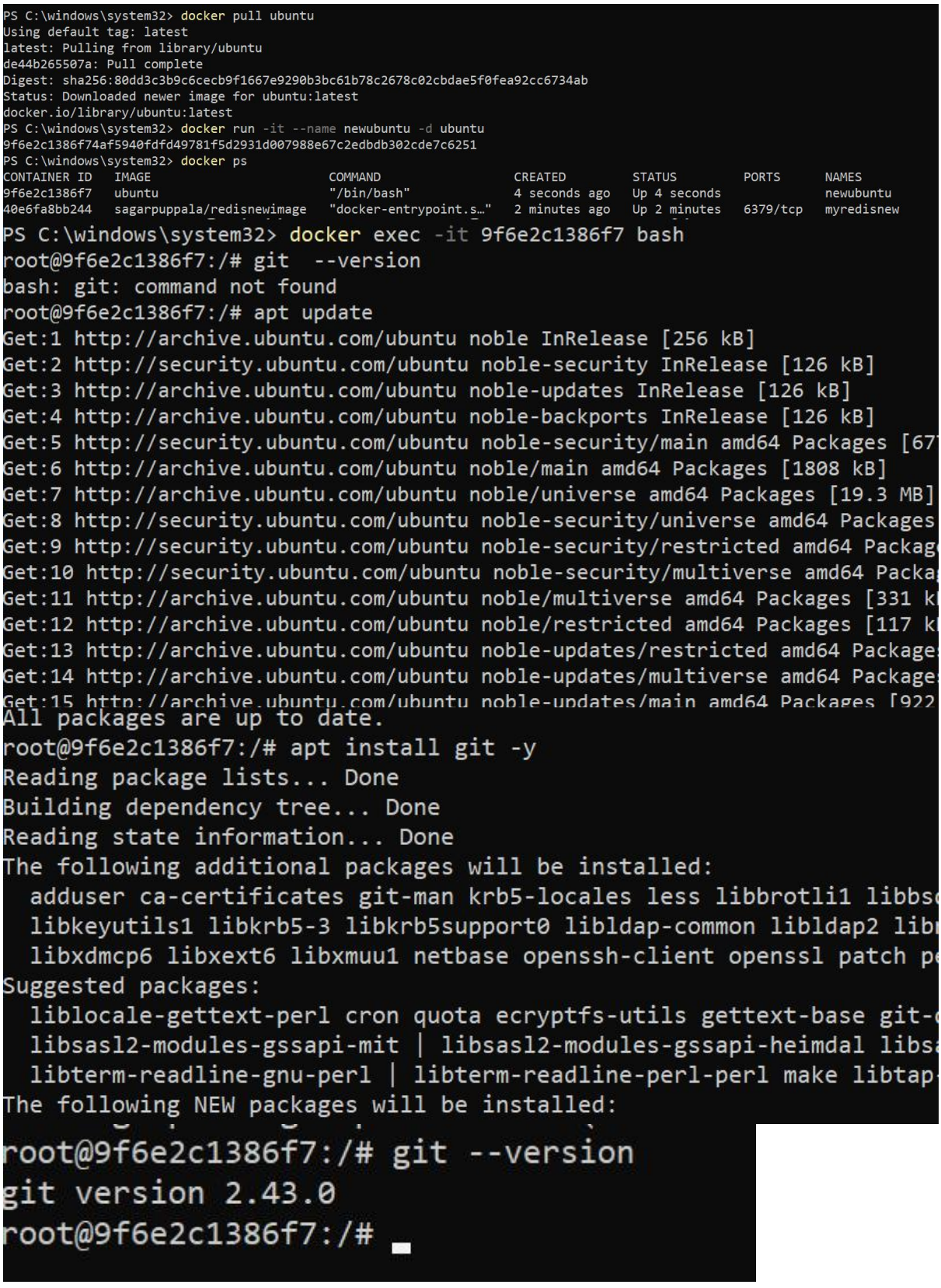
```
tcp, 127.0.0.1:32777->32443/tcp   minikube
PS C:\windows\system32> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
bc0965b23a04: Pull complete
9501a6ec095f: Pull complete
98e7597530ef: Pull complete
75dffa679c9b: Pull complete
8912a88e73c8: Pull complete
141f00d6fee8: 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"                 3 minutes ago    Exited (0) 3 minutes ag
                          sleepy_chandrasekhar
91341cc7910c   hello-world            "/hello"                 6 days ago       Exited (0) 6 days ago
                          romantic_mclean
1879b82aa084   kicbase/stable:v0.0.45   "/usr/local/bin/entr…"   8 days ago       Exited (255) 6 days ago
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: sha256:36c5c6a38ba8a8cf7c71b8035ab30704bb905d54980a2528025dcb90d8513543
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, COMMITING THE CHNAGES WITH NEW IMAGE AND PUSHING INTO

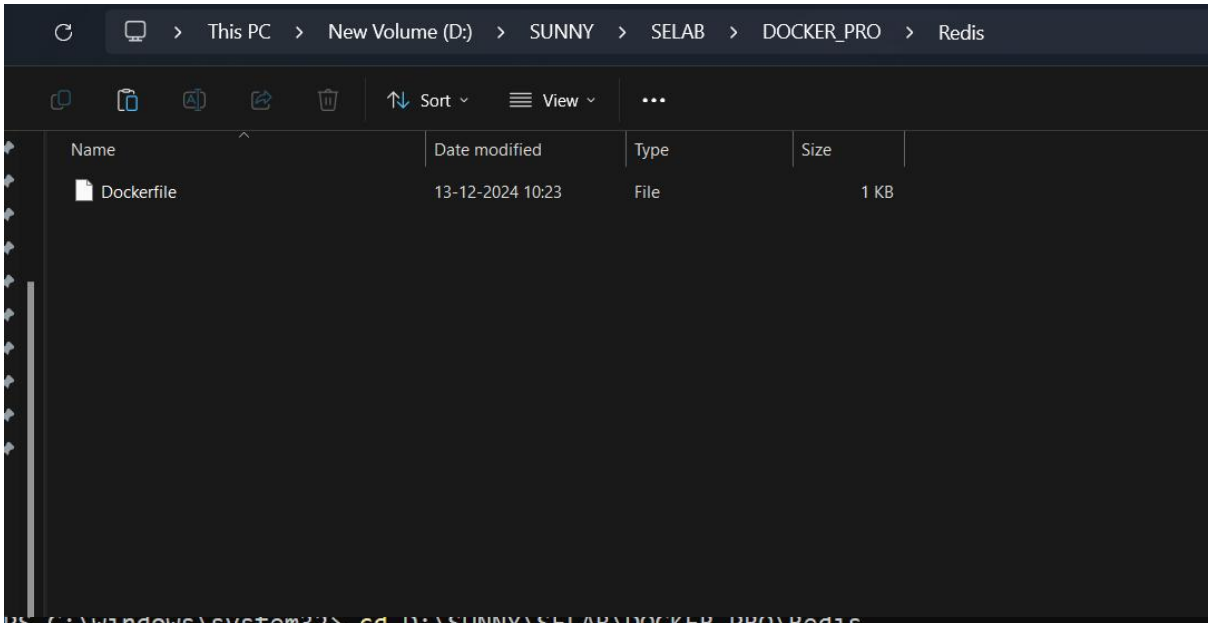
DOKCER HUB


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 newubuntu
40e6fa8bb244 sagarpuppala/redisnewimage "docker-entrypoint.s…" 2 minutes ago Up 2 minutes 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
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
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packag
Get:10 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packa
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 k
Get:12 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 k
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Package
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Package
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922
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 libbs
libkeyutils1 libkrb5-3 libkrb5support0 libldap-common libldap2 lib
libxdmcp6 libxext6 libxmuu1 netbase openssh-client openssl patch p
Suggested packages:
liblocale-gettext-perl cron quota ecryptfs-utils gettext-base git-
libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal libs
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:f226cb12155882ee006fd400a72aaf804ade85ff37158cfb08a549710da85fa0
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                  CREATED             STATUS
                                              NAMES
a5d9326eaba7   sagarpuppala/newubuntu2024     "/bin/bash"              31 seconds ago      Exited (0) 3 seconds ago
                                              newubuntu2024
40e6fa8bb244   sagarpuppala/redisnewimage     "docker-entrypoint.s…"   29 minutes ago      Up 29 minutes
                                              myredisnew
1a43b939d308   hello-world                    "/hello"                 About an hour ago   Exited (0) About an hour ag
                                              sleepy_chandrasekhar
91341cc7910c   hello-world                    "/hello"                 6 days ago          Exited (0) 6 days ago
                                              romantic_mclean
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@sha256:5ea369cb5fb98685715a7dff23be5aedd6a9cb8dbd6888225d33fabe
Deleted: sha256:5fbc1116a6d10f3d979889942fa8625326baa2267c809d9c7e65404801f695f9
Deleted: sha256:f226cb12155882ee006fd400a72aaf804ade85ff37158cfb08a549710da85fa0
PS C:\windows\system32>
```

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




```
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 .dockerignore
 => => transferring context: 2B
 => [1/1] FROM docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
 => => resolve docker.io/library/redis:latest@sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
 => => sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83 10.26kB / 10.26kB
 => => sha256:b5e874b32a794f96b6119e5478f93165d4110cdd8c056d470b3d48b27114f716 8.59kB / 8.59kB
 => => sha256:fa84a527b1808ee5ad2778e5d89aae791e7040b5c303f1d45569a9ca8d5b61ee 2.48kB / 2.48kB
 => => sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f 28.23MB / 28.23MB
 => => sha256:9501a6ec095f4bd5242b8ceedf1119d098b45c6e9237c53cedb022a20ce9c1fa 1.10kB / 1.10kB
 => => sha256:98e7597530ef912e6a2cce5e6e0cd3f155b611a5d6b6ea7823640b6c4f2c4a30 874B / 874B
 => => sha256:75dffa679c9bb360fd358682b0842b9f32a732e6f80d47e8a7f3efde5f05dc07 1.44MB / 1.44MB
 => => sha256:8912a88e73c83f622d19fd2dac8a055ca96342628d29d56b81fde09475ef7007 15.33MB / 15.33MB
 => => sha256:141f00d6fee86bf601ac2eca3db312a24d3e4147bc166bbecb09428be54794b5 97B / 97B
 => => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B
 => => sha256:8242f9d5b464f375c010545626f3107bcfe0f68e7a2a78e95c8d397427a12c0e 572B / 572B
 => => extracting sha256:bc0965b23a04fe7f2d9fb20f597008fcf89891de1c705ffc1c80483a1f098e4f
 => => extracting sha256:9501a6ec095f4bd5242b8ceedf1119d098b45c6e9237c53cedb022a20ce9c1fa
 => => extracting sha256:98e7597530ef912e6a2cce5e6e0cd3f155b611a5d6b6ea7823640b6c4f2c4a30
 => => extracting sha256:75dffa679c9bb360fd358682b0842b9f32a732e6f80d47e8a7f3efde5f05dc07
 => => 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    d2c94e258dcb   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
                          sleepy_chandrasekhar
91341cc7910c   hello-world            "/hello"                 6 days ago       Exited (0) 6 days ago
                          romantic_mclean
1879b82aa084   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   minikube
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker stop 2b71e60cebb4
2b71e60cebb4
```

Start a build

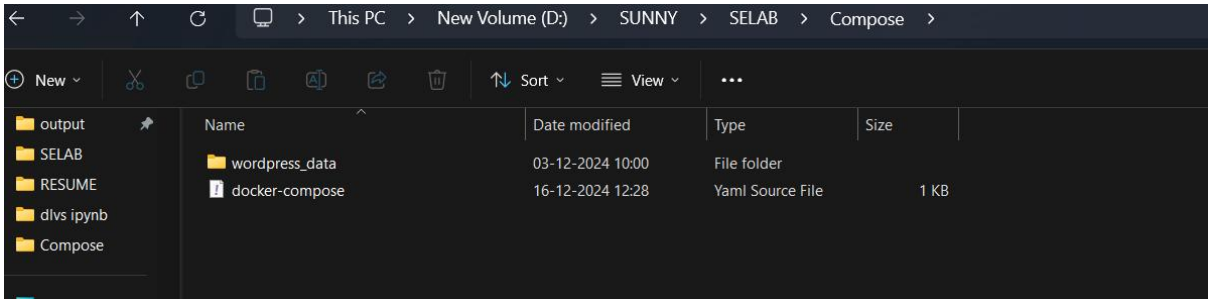
```
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker commit 2b71e60cebb4 sagarpuppala/redisnewimage
sha256:2aab01517818ea1b66070e56b9810ebf03e98deb8117315f9468e1553b53fea8
PS D:\SUNNY\SELAB\DOCKER_PRO\Redis> docker images
REPOSITORY                     TAG       IMAGE ID       CREATED          SIZE
sagarpuppala/redisnewimage     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    d2c94e258dcb   19 months ago    13.3kB
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@sha256:728591b3a66fe127041fed7b77dbe8957a44c4df86de7ec7346d190a2c
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
75dffa679c9b: 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
75dffa679c9b: 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:

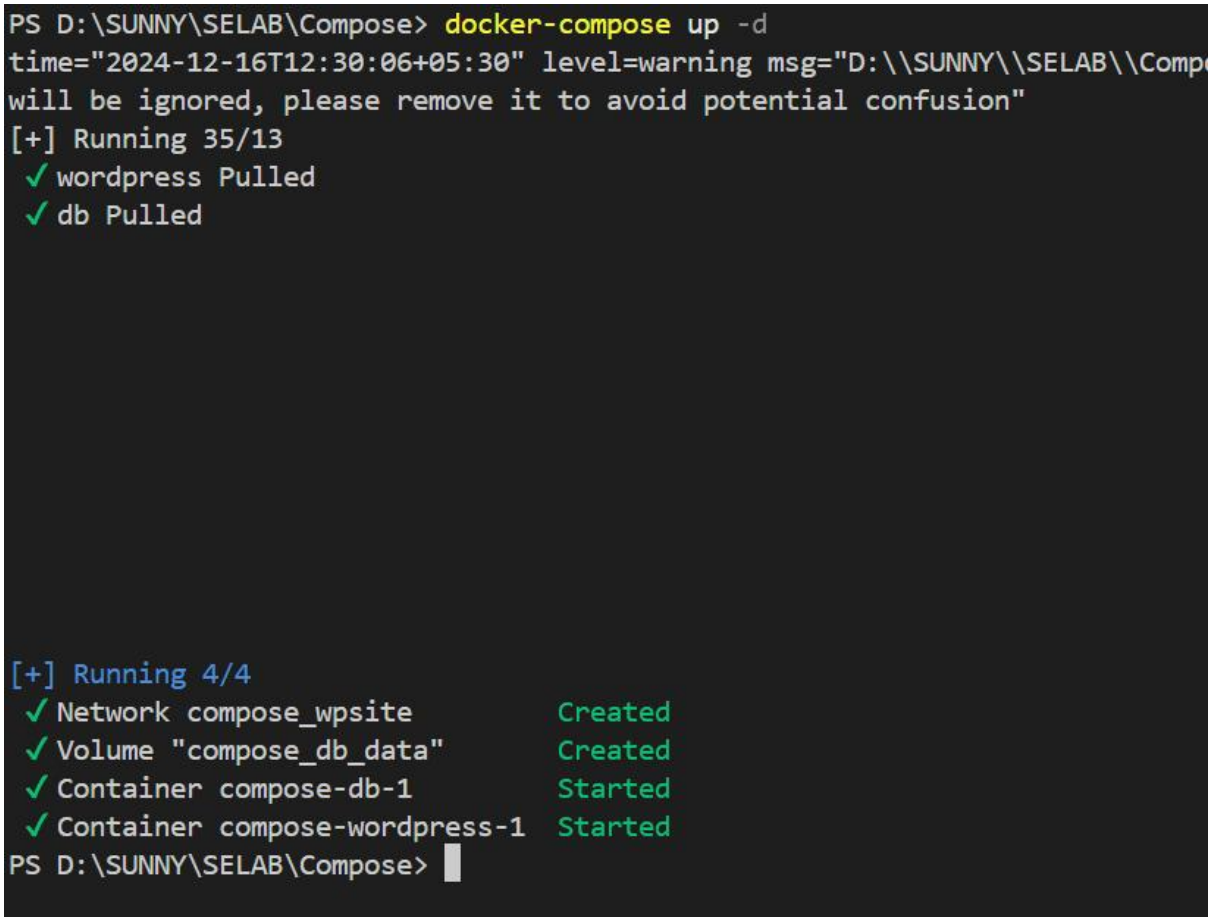


dokcer-compose.yaml file

Welcome docker-compose.yaml

docker-compose.yaml

```yaml
version: '3'

services:
  # Database
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
    networks:
      - wpsite

  # WordPress
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - '8000:80'
    restart: always
    volumes:
      - ./wordpress_data:/var/www/html
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
    networks:
      - wpsite

networks:
  wpsite:

volumes:
  db_data:
```



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.

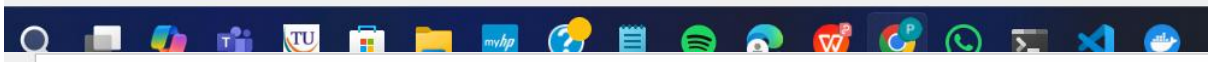
Your Email sagarpuppala123@gmail.com

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



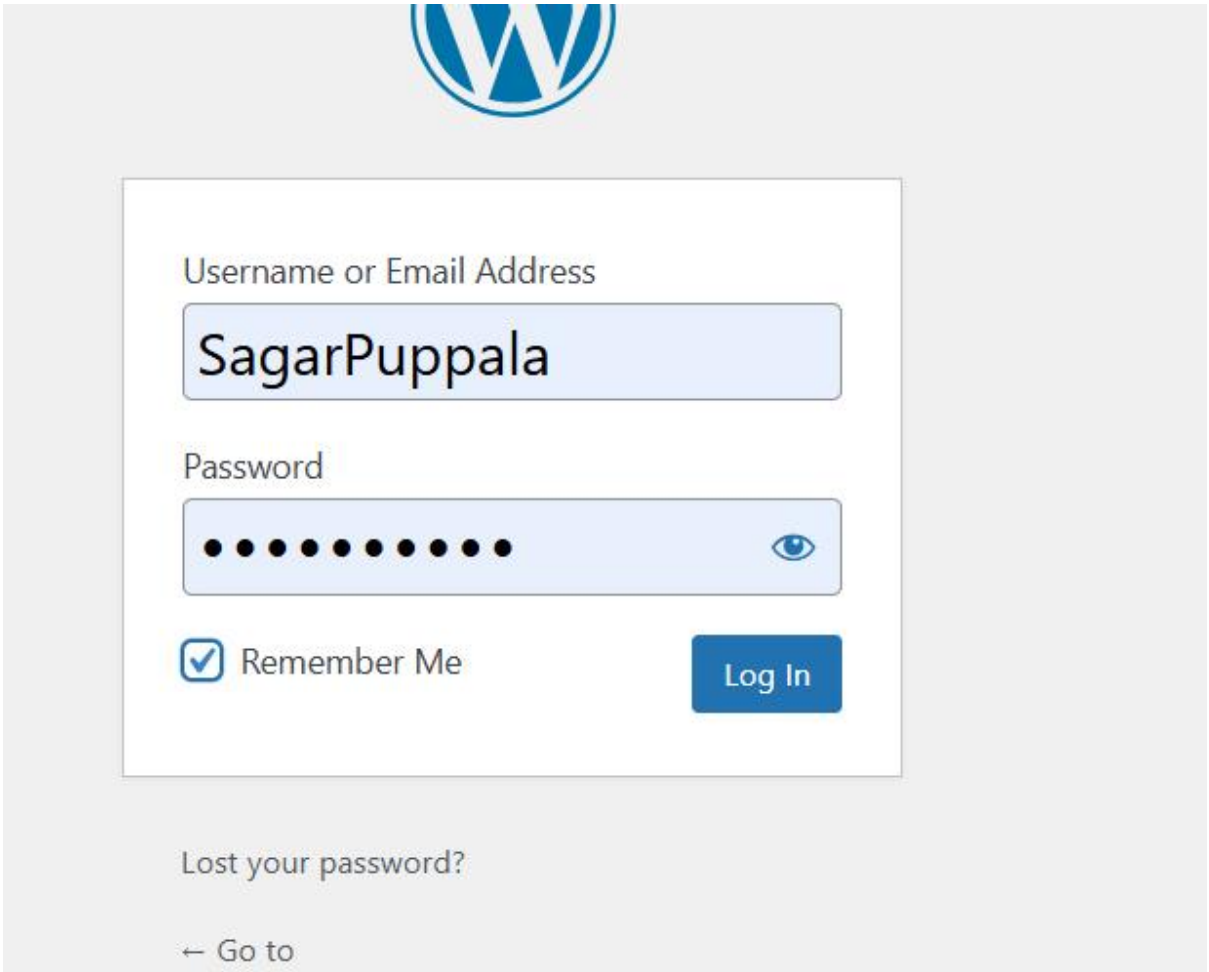
Success!

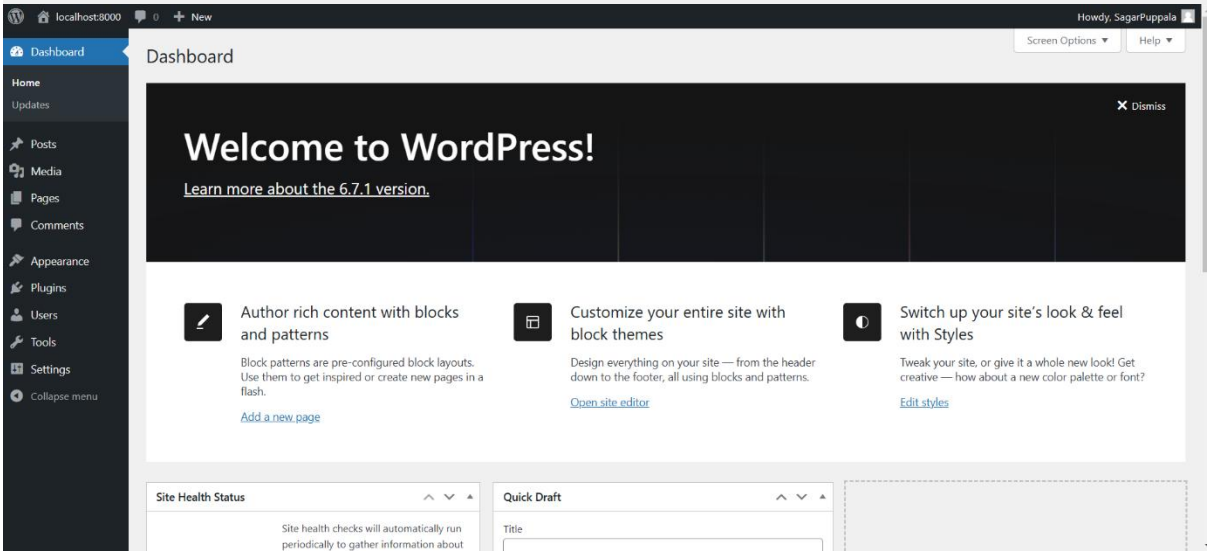
WordPress has been installed. Thank you, and enjoy!

Username SagarPuppala

Password *Your chosen password.*

Log In


Username or Email Address
SagarPuppala
Password
Remember Me
Log In
Lost your password?
← Go to

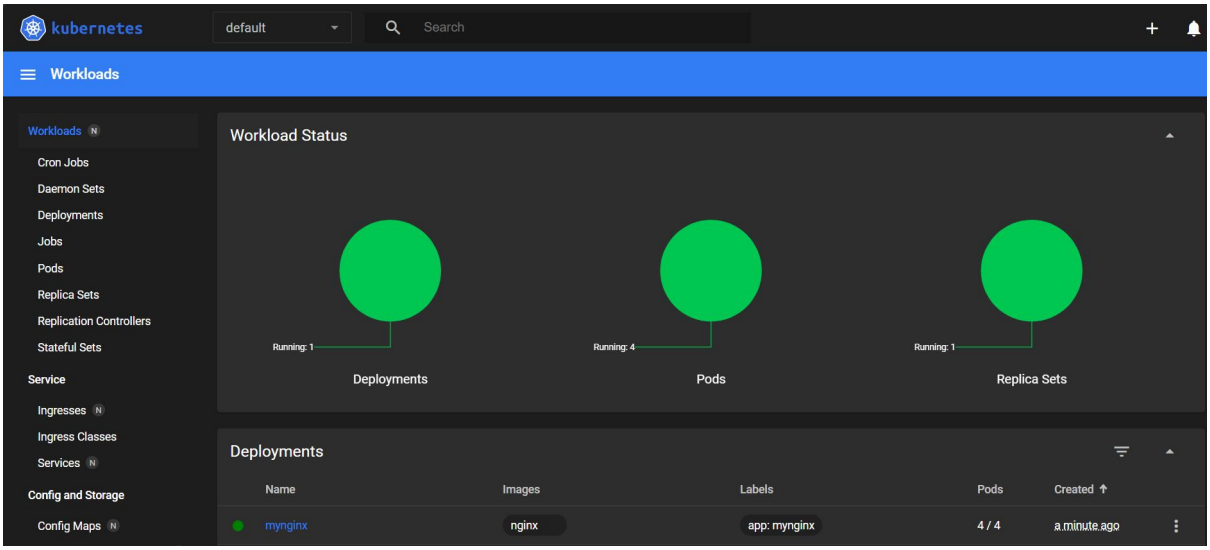

localhost:8000
New
Howdy, SagarPuppala
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Screen Options
Help
Dismiss
Welcome to WordPress!
Learn more about the 6.7.1 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
Site health checks will automatically run periodically to gather information about
Quick Draft
Title

```
 ✓ 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 obsol
will be ignored, 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>
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 attribu
will be ignored, 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 attribu
will be ignored, 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
* 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 --po
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/networki
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:kub
ur default browser...
```



NAGIOS INSTALLATION:

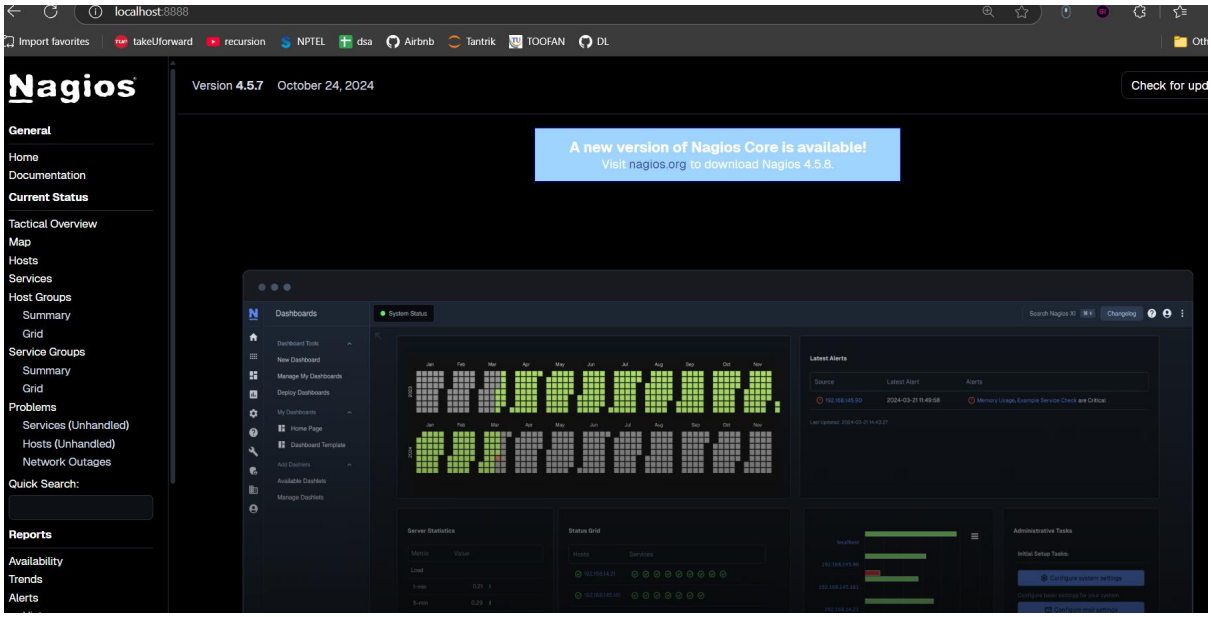
1. Pull nagios latest from the jasonrivers in docker

```
PS C:\WINDOWS\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
ff65ddf9395b: Pull complete
785b9873bdf4: Extracting [=======>                                           ]  42.34MB/2
0ef9446ba5cc: Download complete
53aff88babc4: Download complete
d72f92e29533: Download complete
706ed7d4ce0a: Download complete
d3245570f968: Download complete
e58e184b986a: Download complete
eeb77e6dde3e: Download complete
0bd0f5795eeb: Download complete
71bfb306f8cb: 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
8fb30af17153: Download complete
9ffe54c5c139: Download complete
279b28aefa10: Download complete
a900dfcceb38: Download complete
9a90645e352c: Download complete
8e911c59da28: 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
```

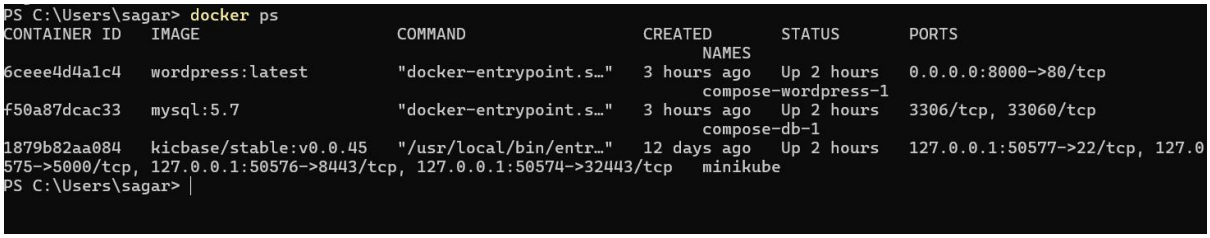
3 NAGIOS WEB PAGE



4 CHECK WHETHER THE NAGIOS IS RUNNING OR NOT

5 STOP NAGIOS DEMO

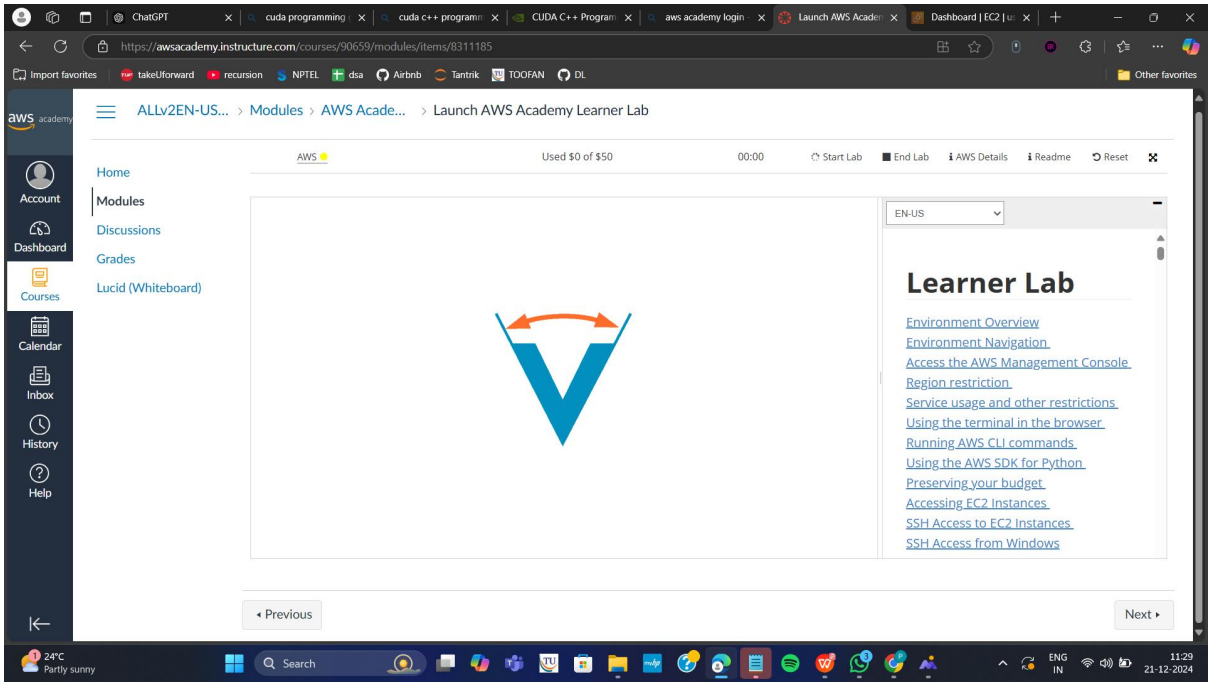
```
PS C:\Users\sagar> docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS
                                                                                  NAMES
1665c9941fa4   jasonrivers/nagios:latest      "/usr/local/bin/star…"   5 minutes ago    Up 5 min
                                                                                  nagiosdemo
6ceee4d4a1c4   wordpress:latest               "docker-entrypoint.s…"   3 hours ago      Up 2 hou
                                                                                  compose-wordpr
f50a87dcac33   mysql:5.7                      "docker-entrypoint.s…"   3 hours ago      Up 2 hou
                                                                                  compose-db-1
1879b82aa084   kicbase/stable:v0.0.45         "/usr/local/bin/entr…"   12 days ago      Up 2 hou
.0.1:50575->5000/tcp, 127.0.0.1:50576->8443/tcp, 127.0.0.1:50574->32443/tcp   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
NAMES
6ceee4d4a1c4 wordpress:latest "docker-entrypoint.s…" 3 hours ago Up 2 hours 0.0.0.0:8000->80/tcp
compose-wordpress-1
f50a87dcac33 mysql:5.7 "docker-entrypoint.s…" 3 hours ago Up 2 hours 3306/tcp, 33060/tcp
compose-db-1
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>

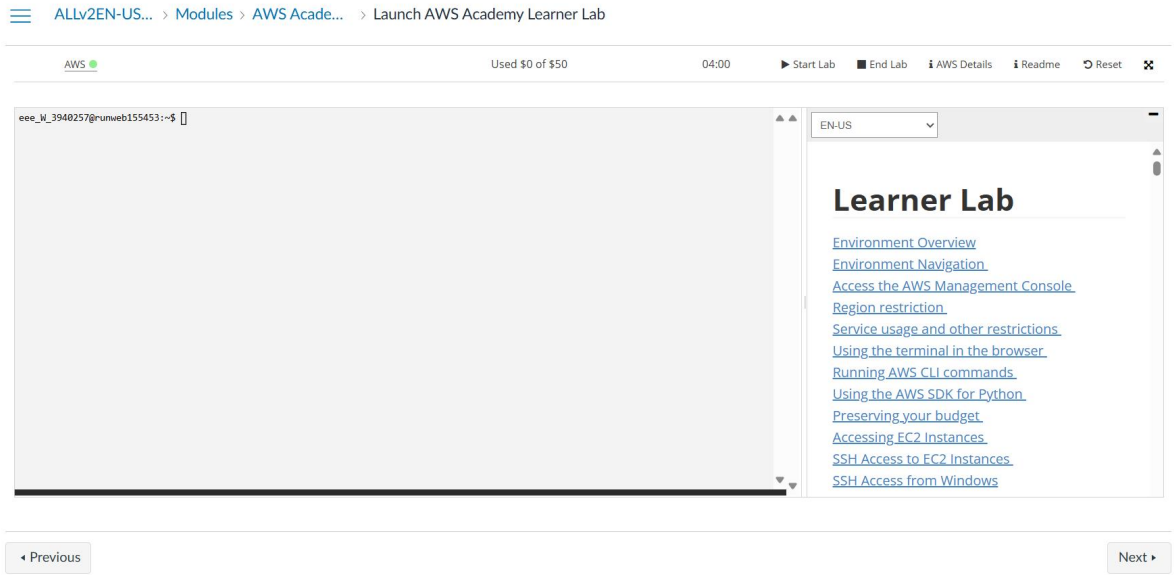
WEEK-12

1 DEPLOYMENT OF INDEX.HTML USING EC2 INSTANCE in AWS

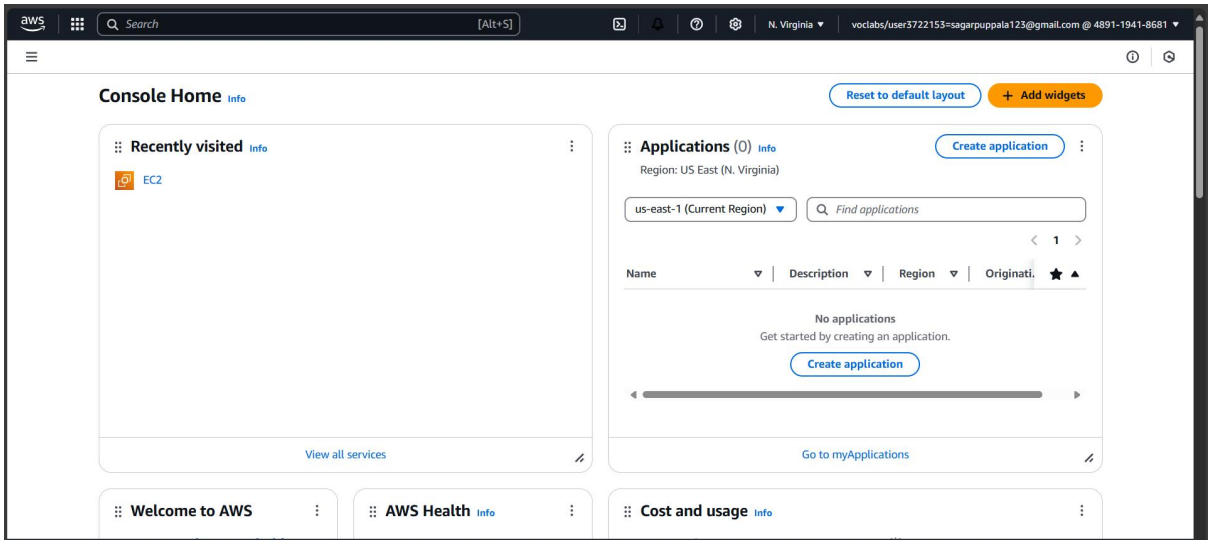
Click on start lab



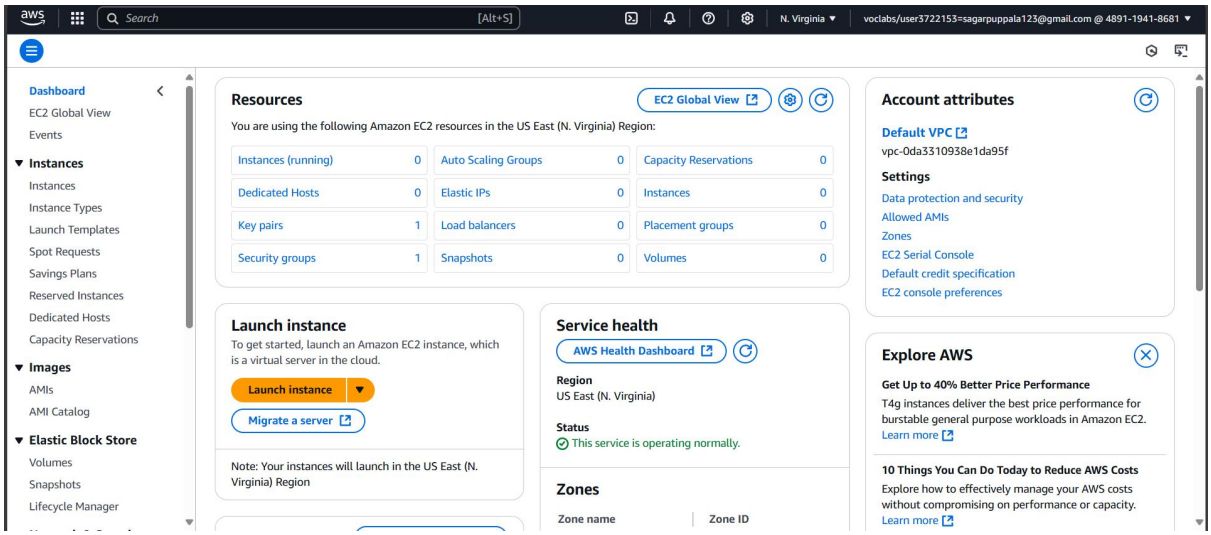
Click on AWS



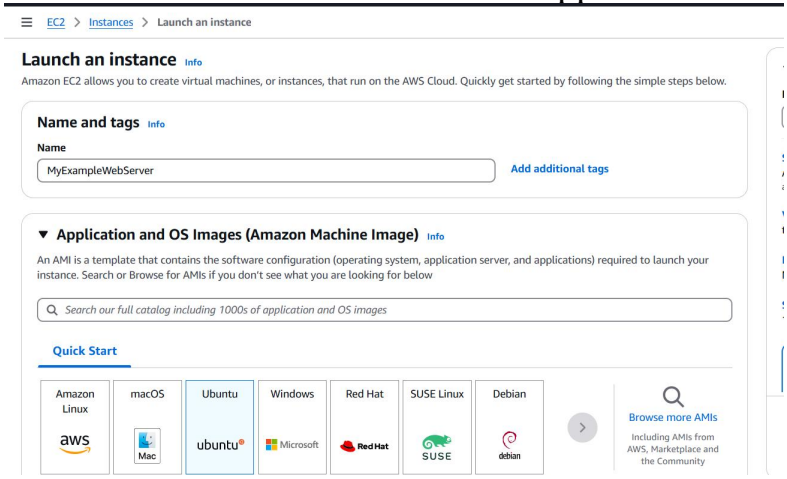
Click on EC2



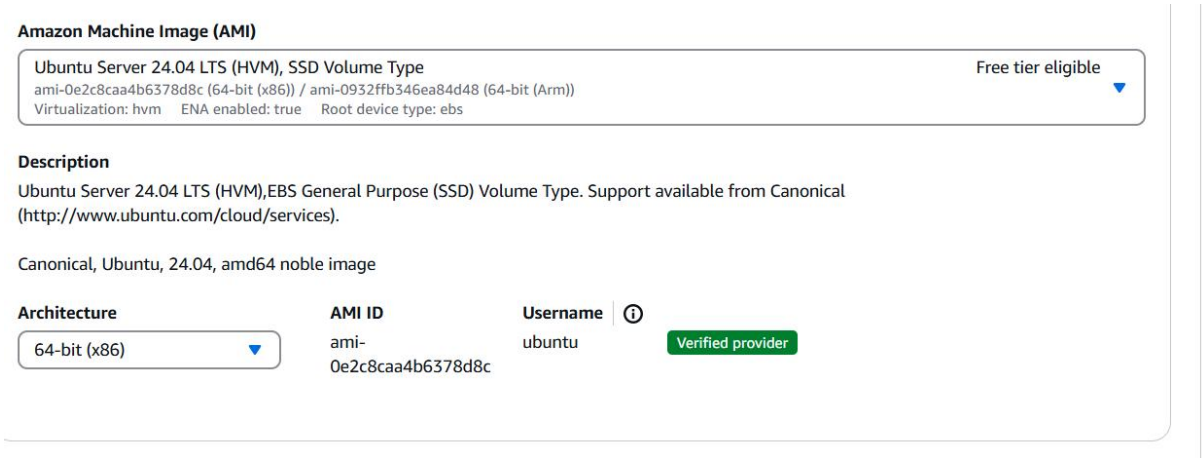
Click on Launch Instance



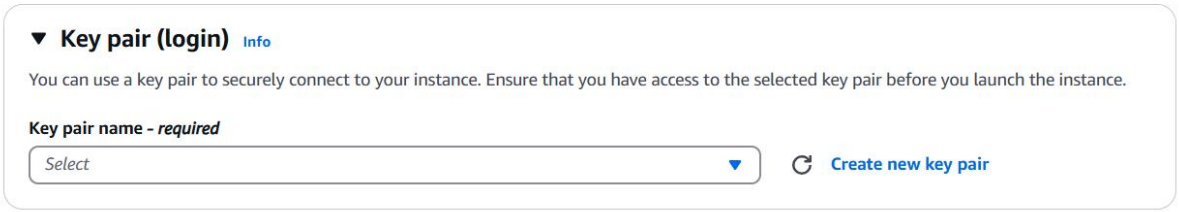
Give name and select ubuntu under application



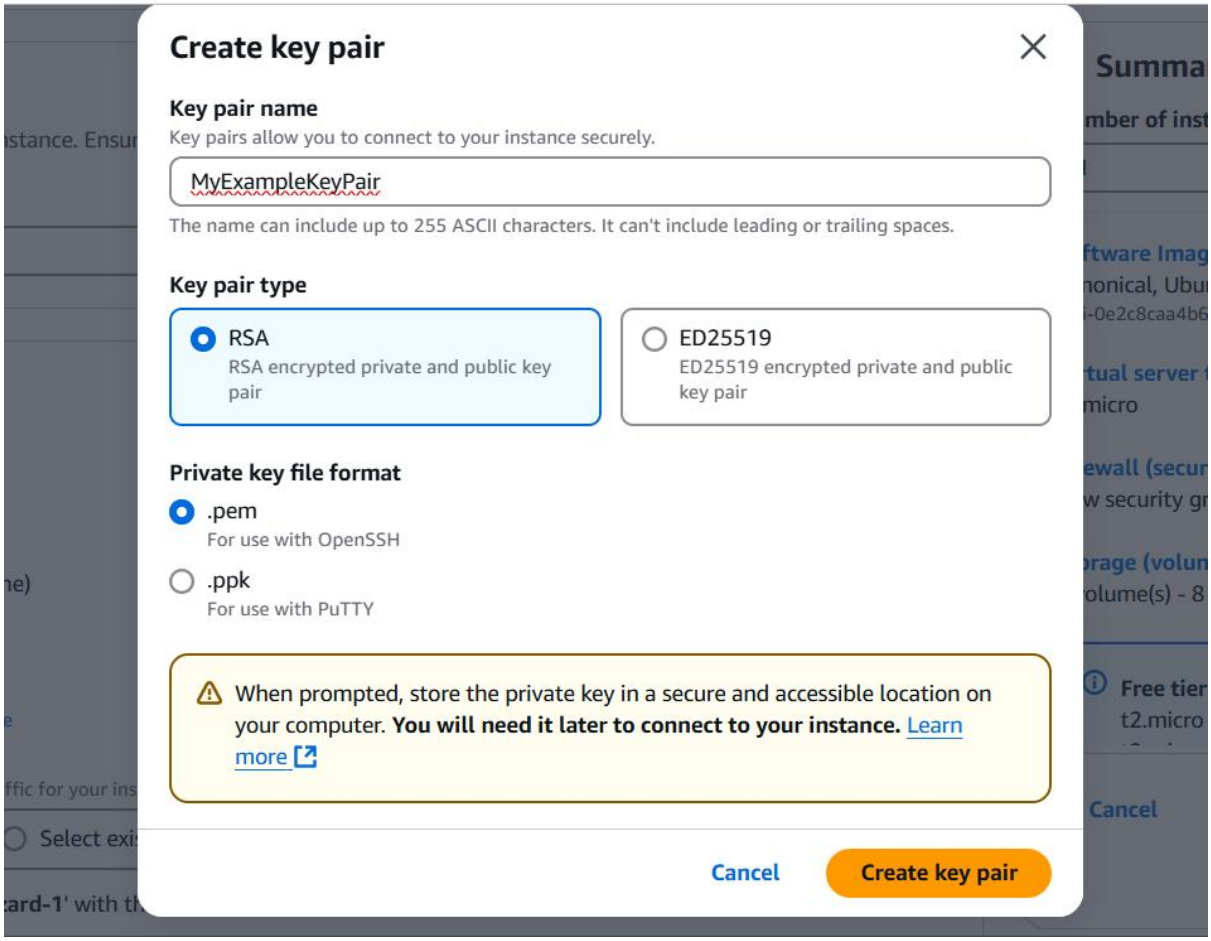
Make sure AMI and Architecture are there as shown



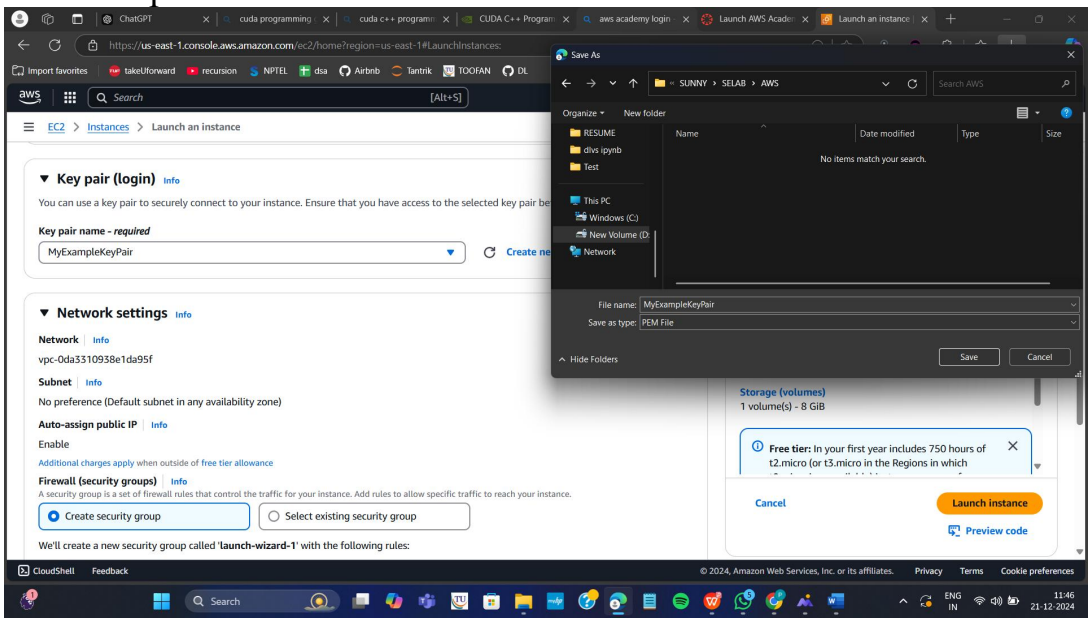
Click on create-new key pair



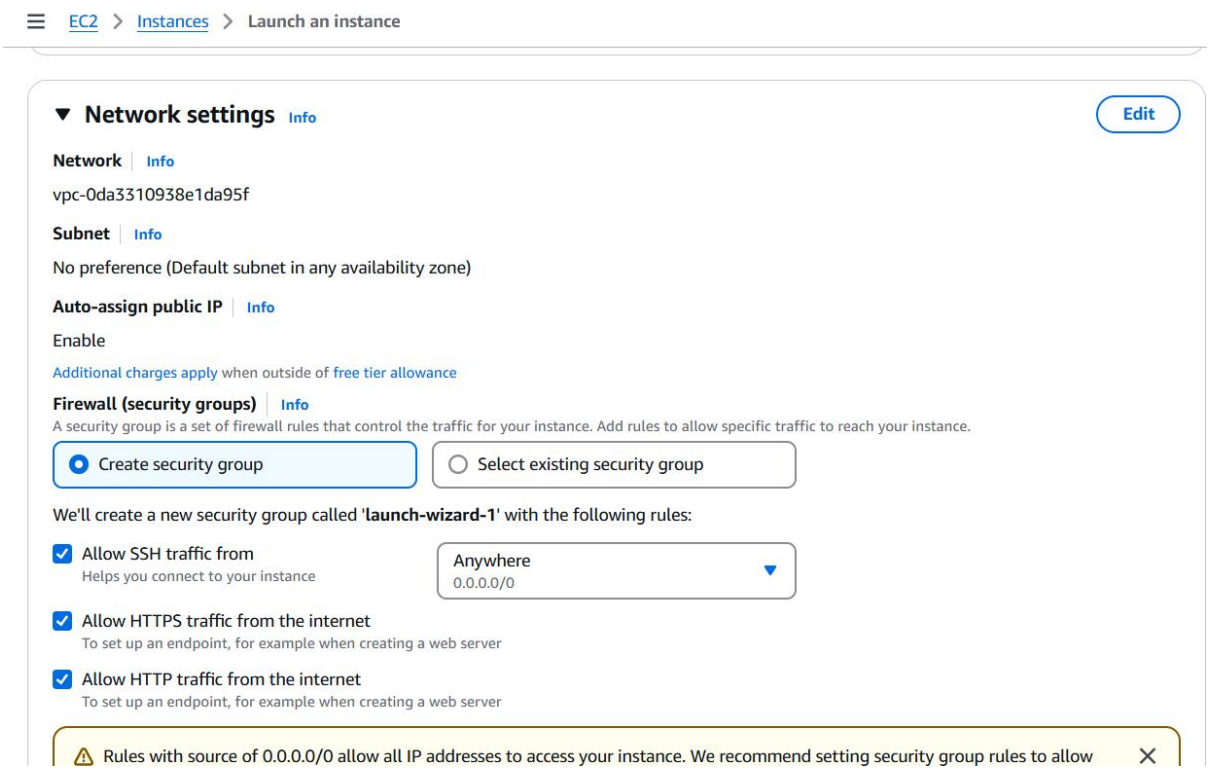
Give KeyPair name and click on create key pair



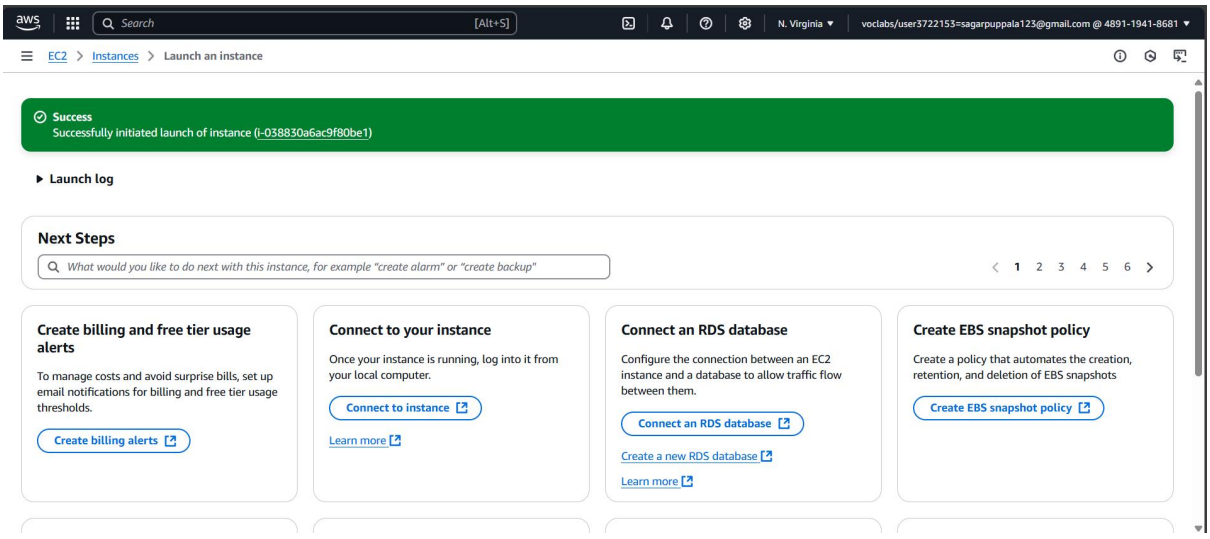
Save the .pem file



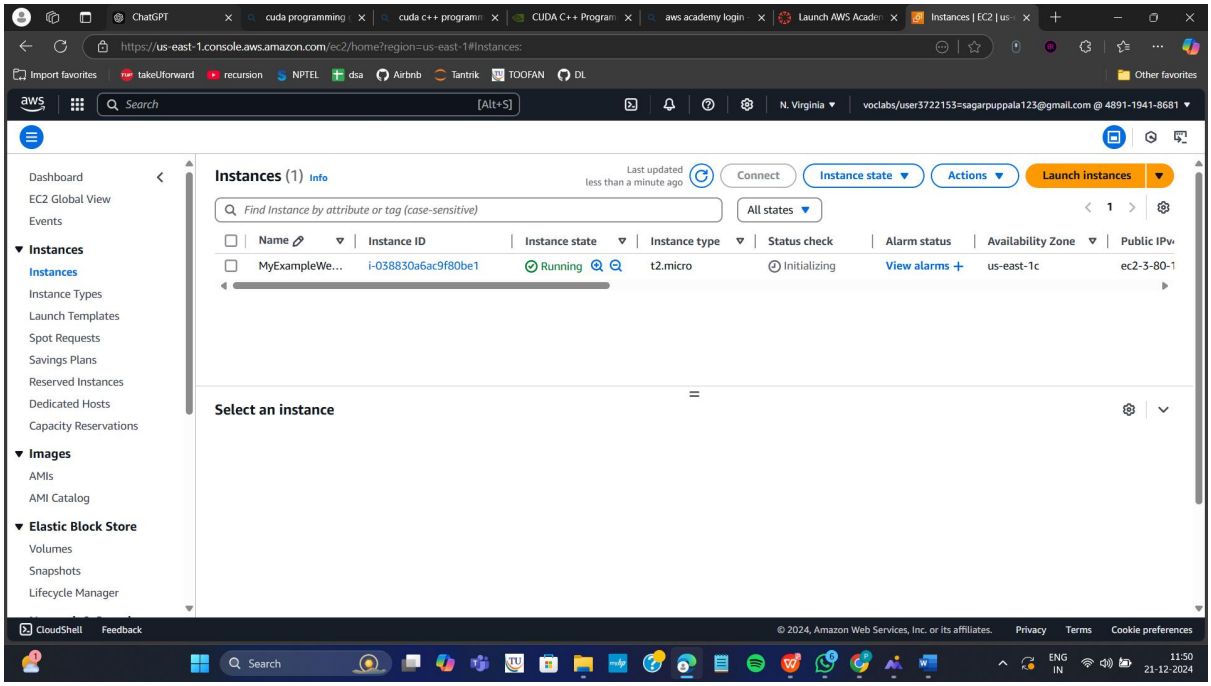
In network setting check all the checkboxes and click on launch instance which is at the bottom right to the page



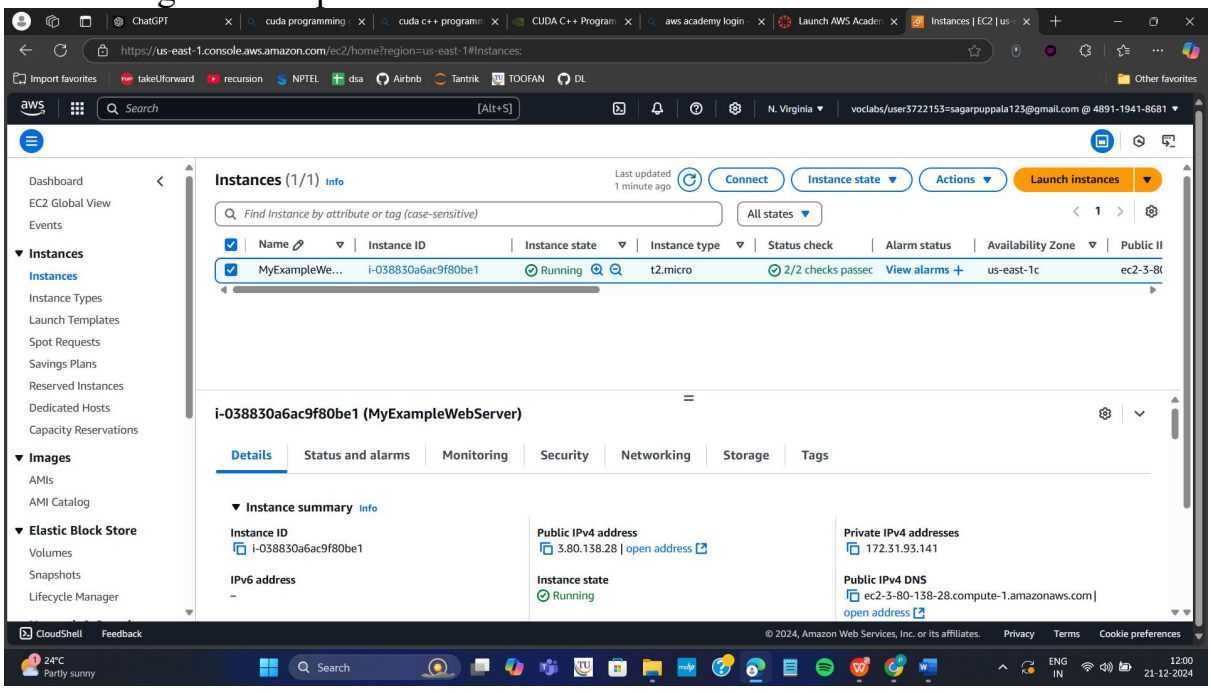
Wait for the Success message.
Now click on the instances



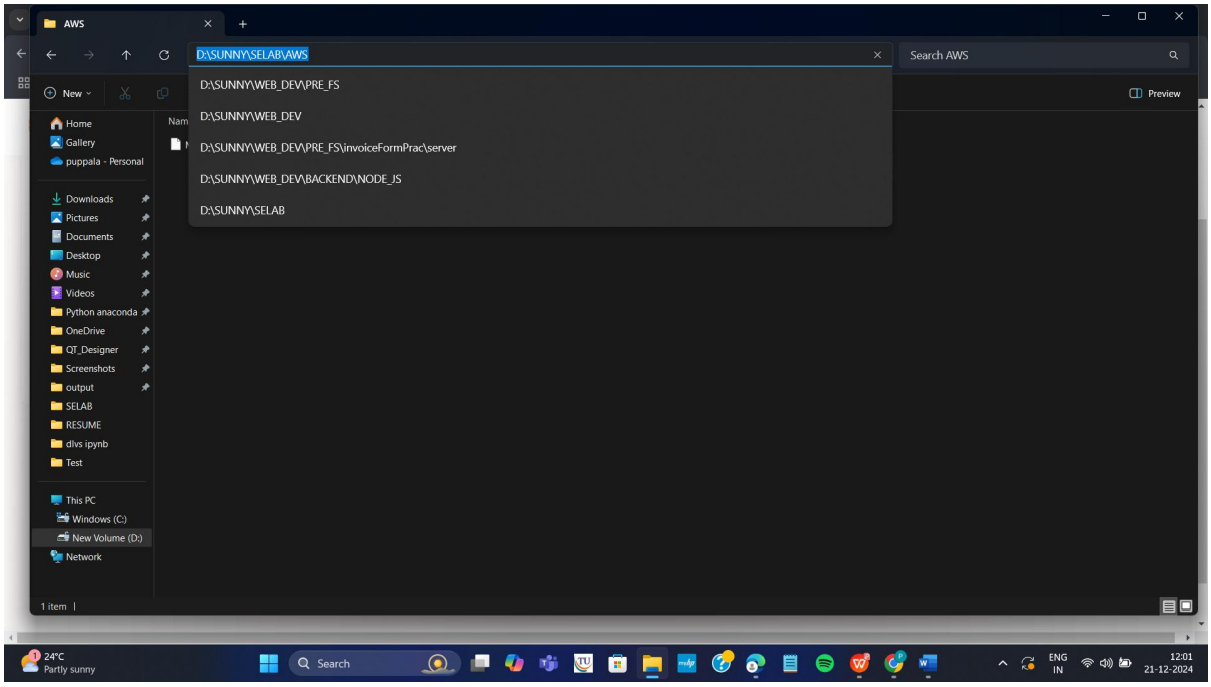
You can see MyExampleWebServer is Running and wait for it to initialize



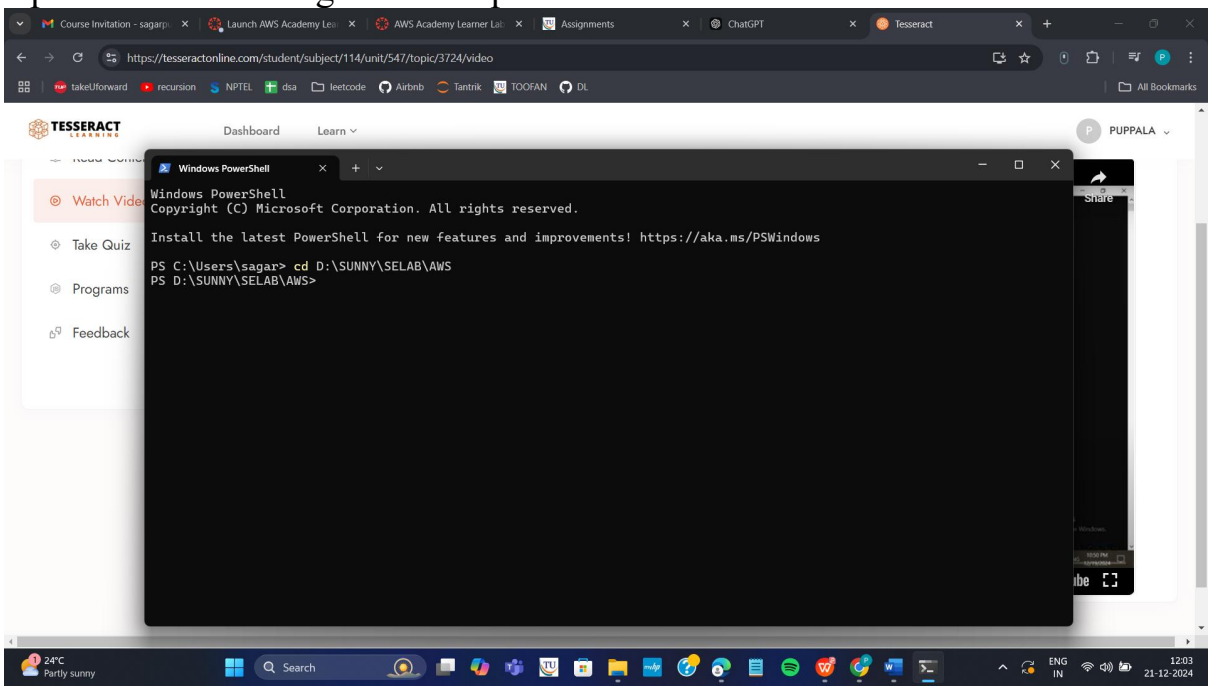
You have get 2 tests passes. Now check the box and click on connect.



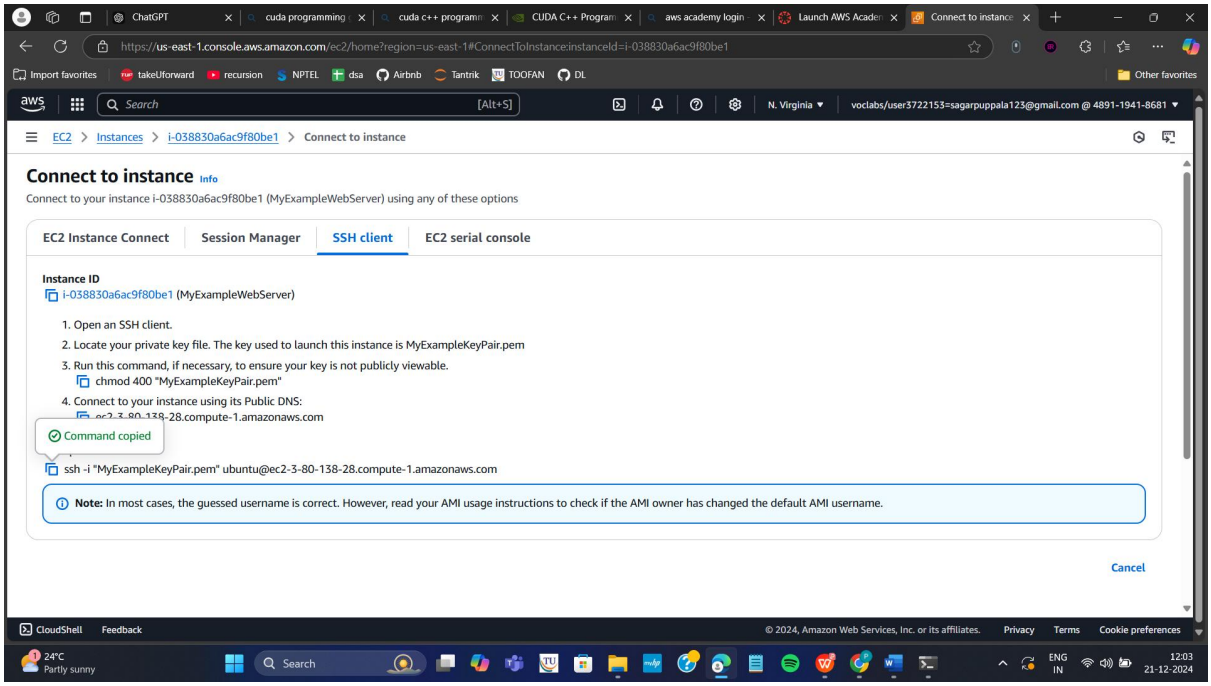
Copy the path of .pem file that you saved earlier



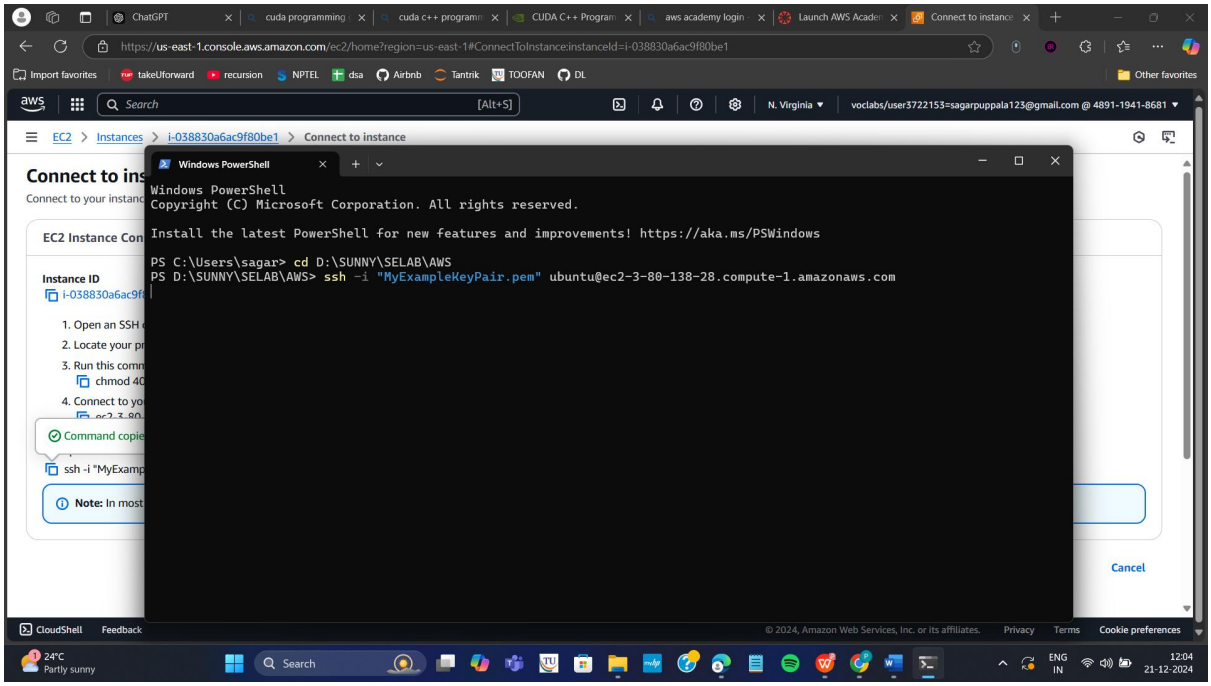
Open CMD and navigate to that path



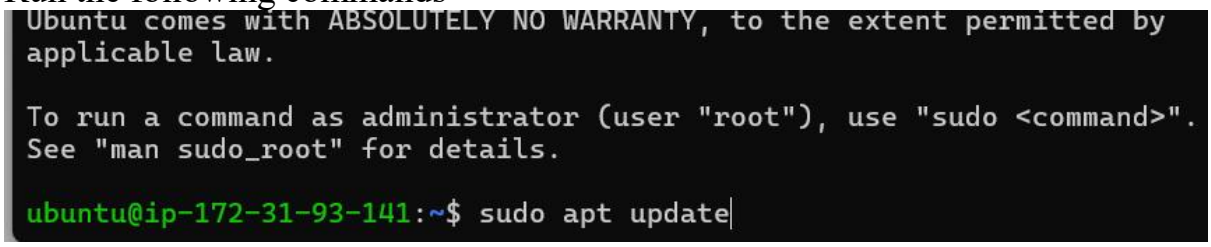
Go to SSh and copy the command which is present at the below



Run the command in the terminal



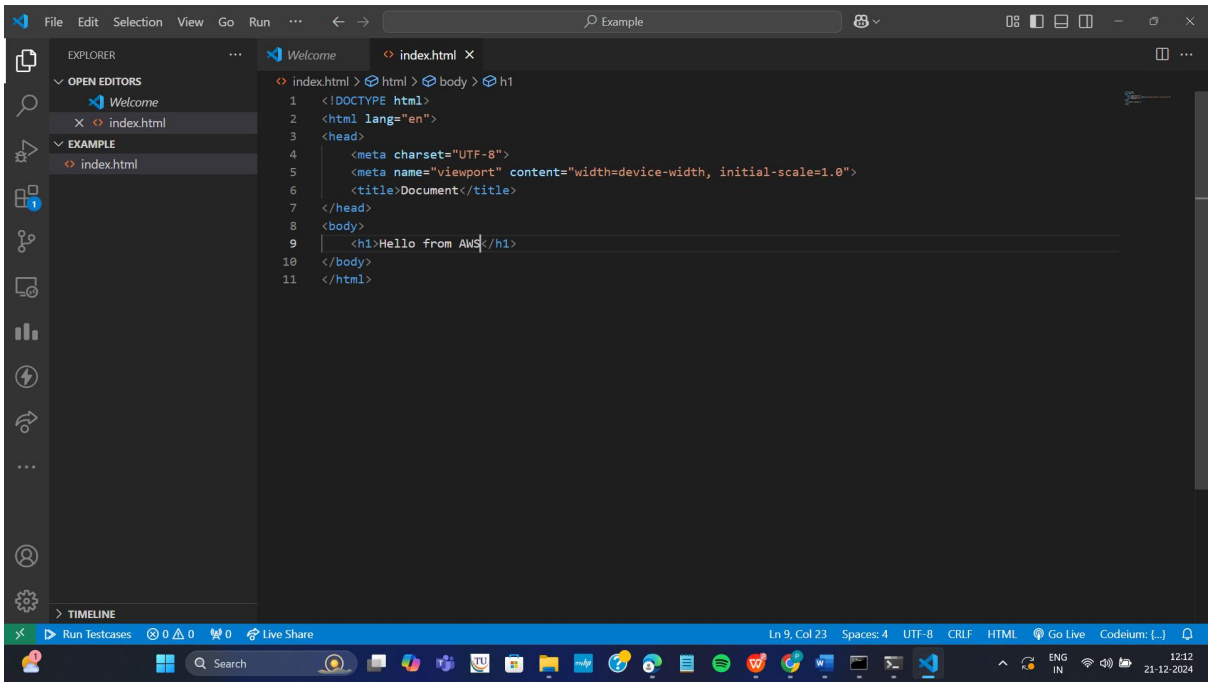
Run the following commands



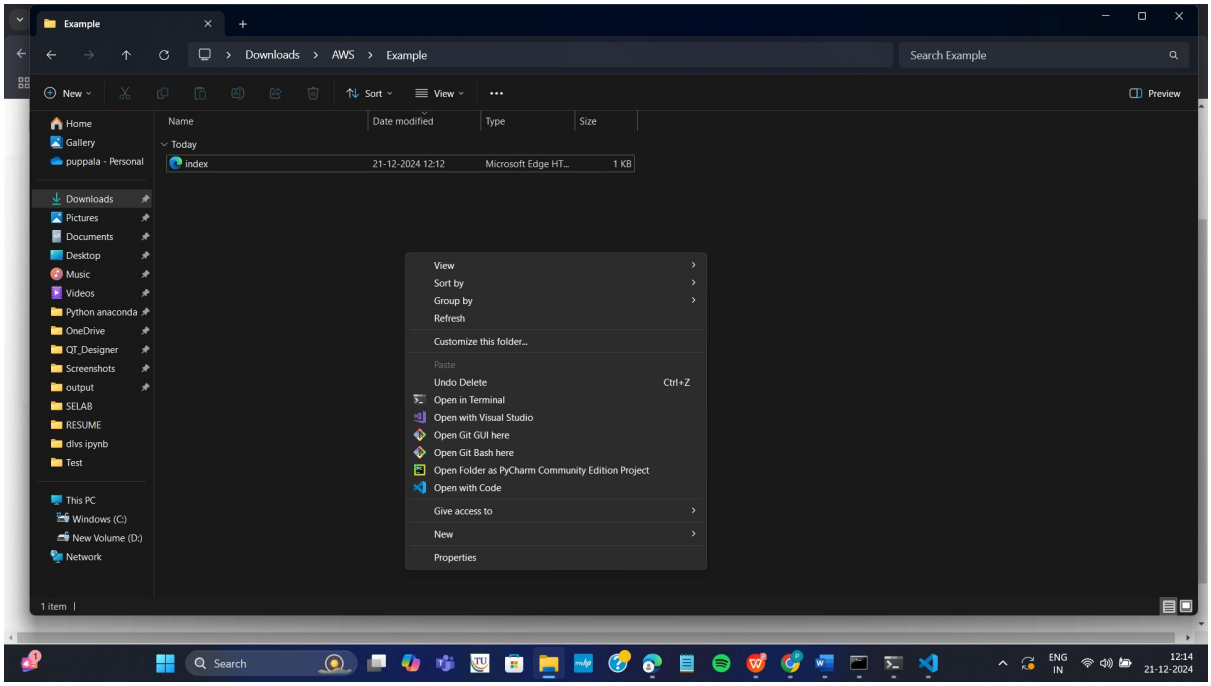
```
ubuntu@ip-172-31-93-141:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docke
  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-172-31-93-141:~$ 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.
```

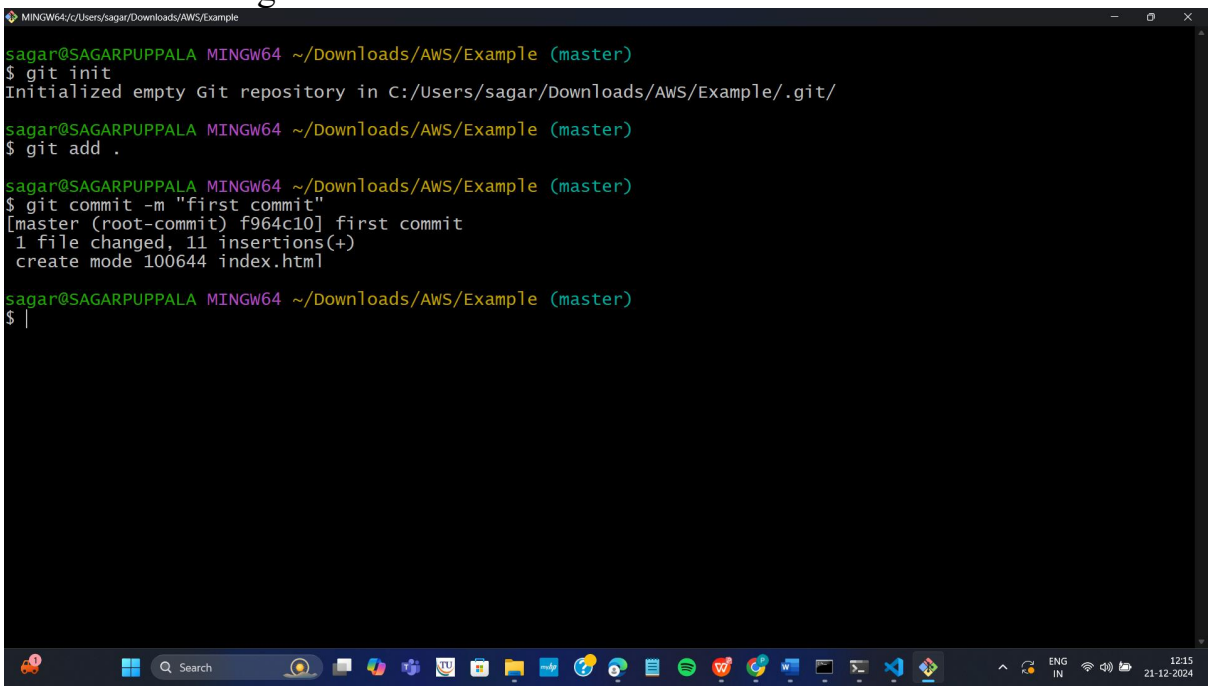
Create basic index.html file



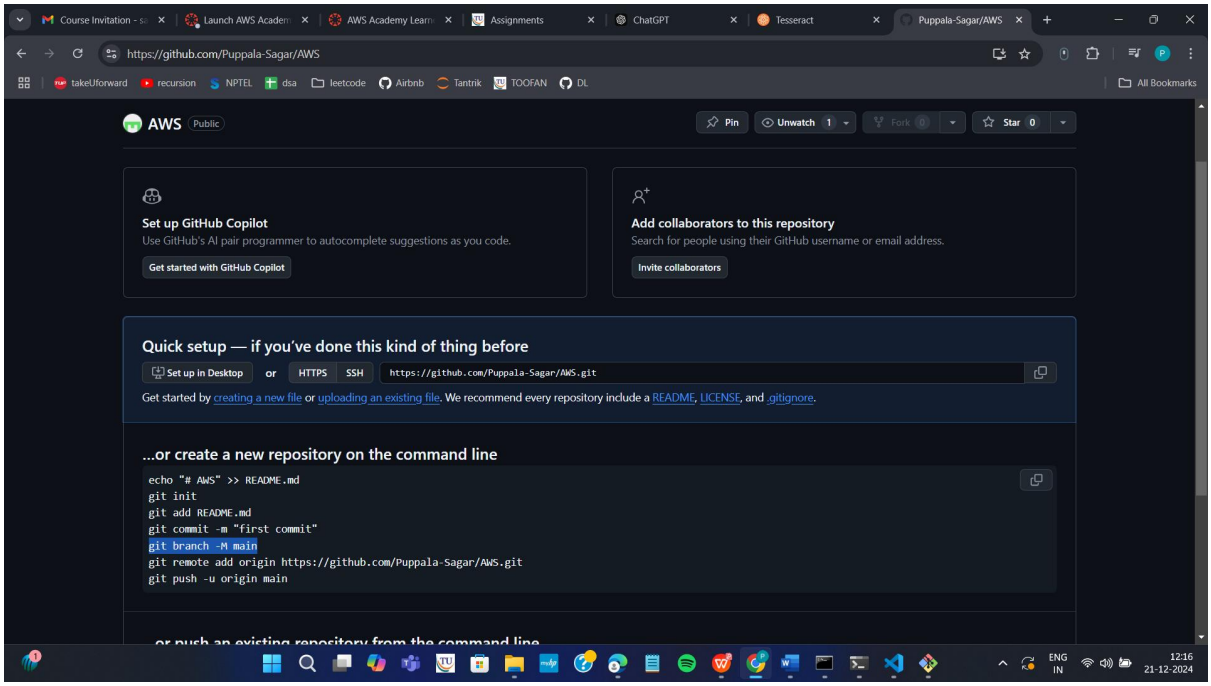
Open git Bash



Run the following commands



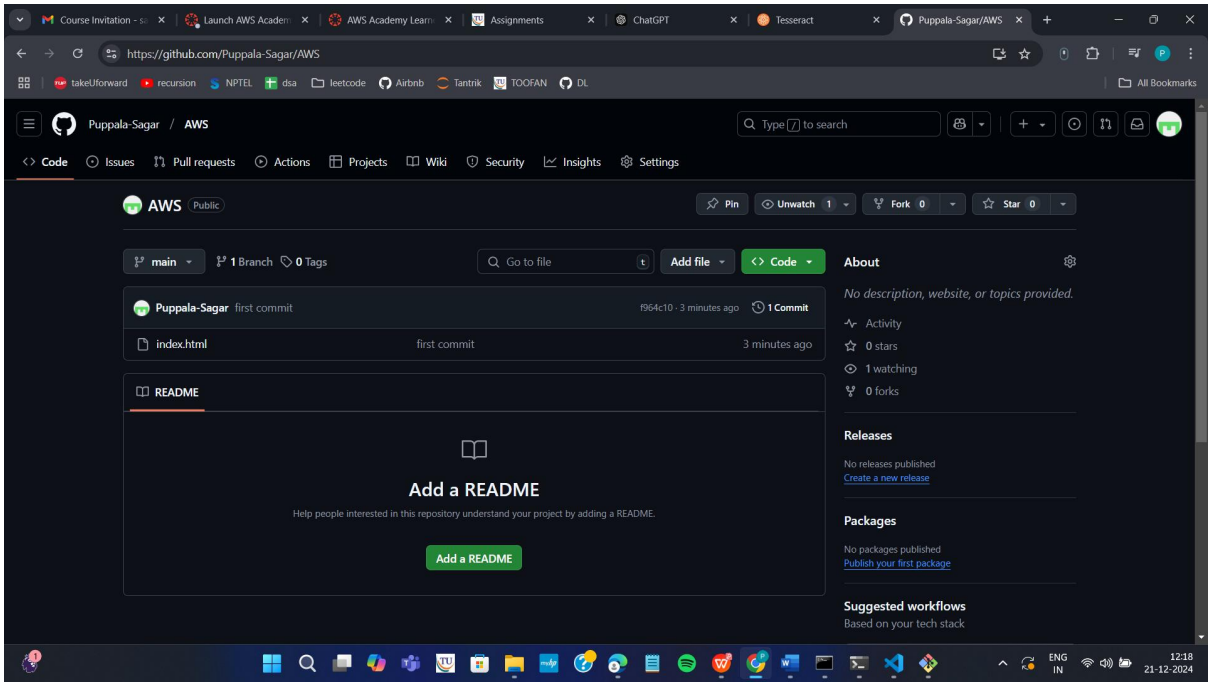
Create git repo



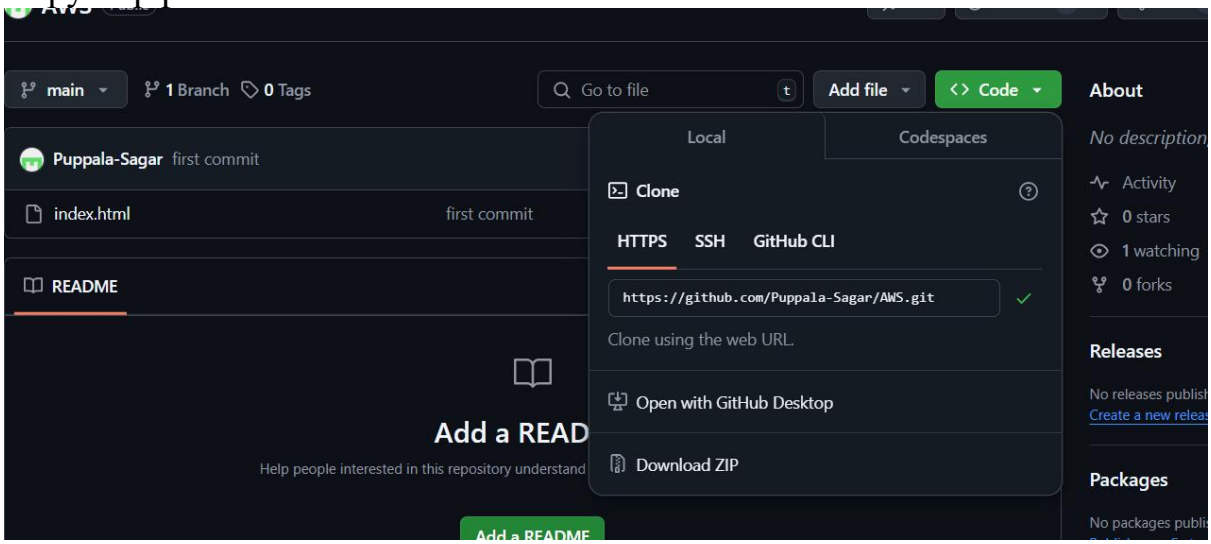
Now run the following commands

```
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



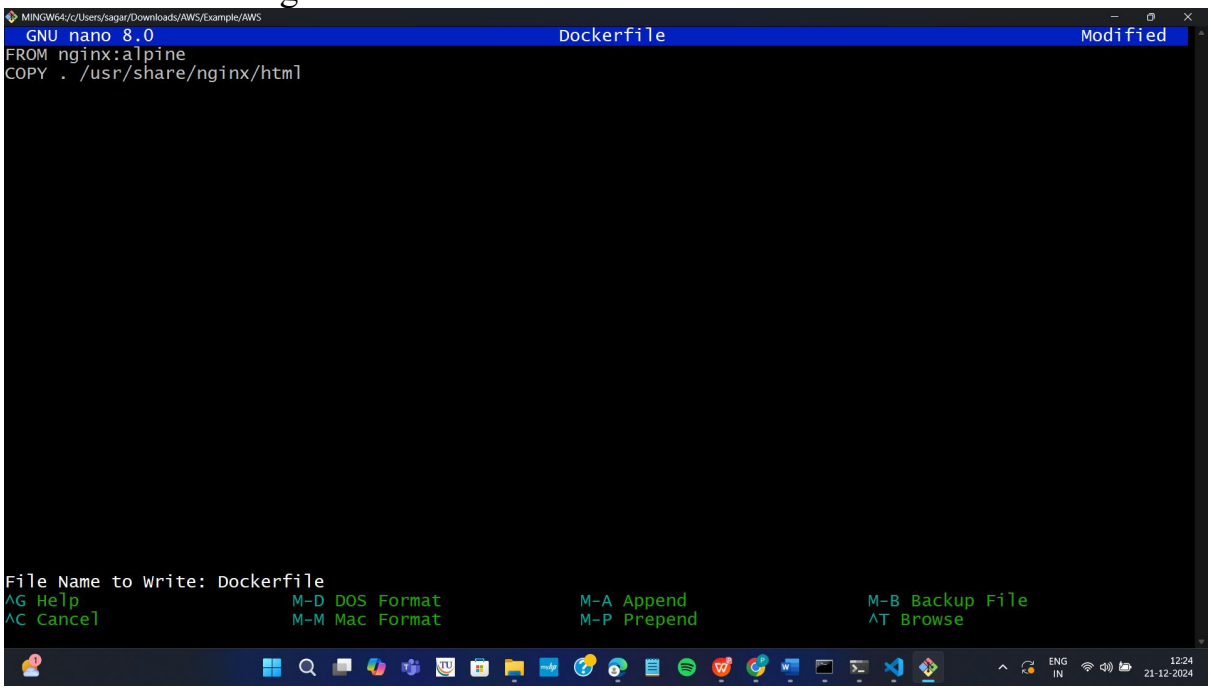
Clone the repository

```
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



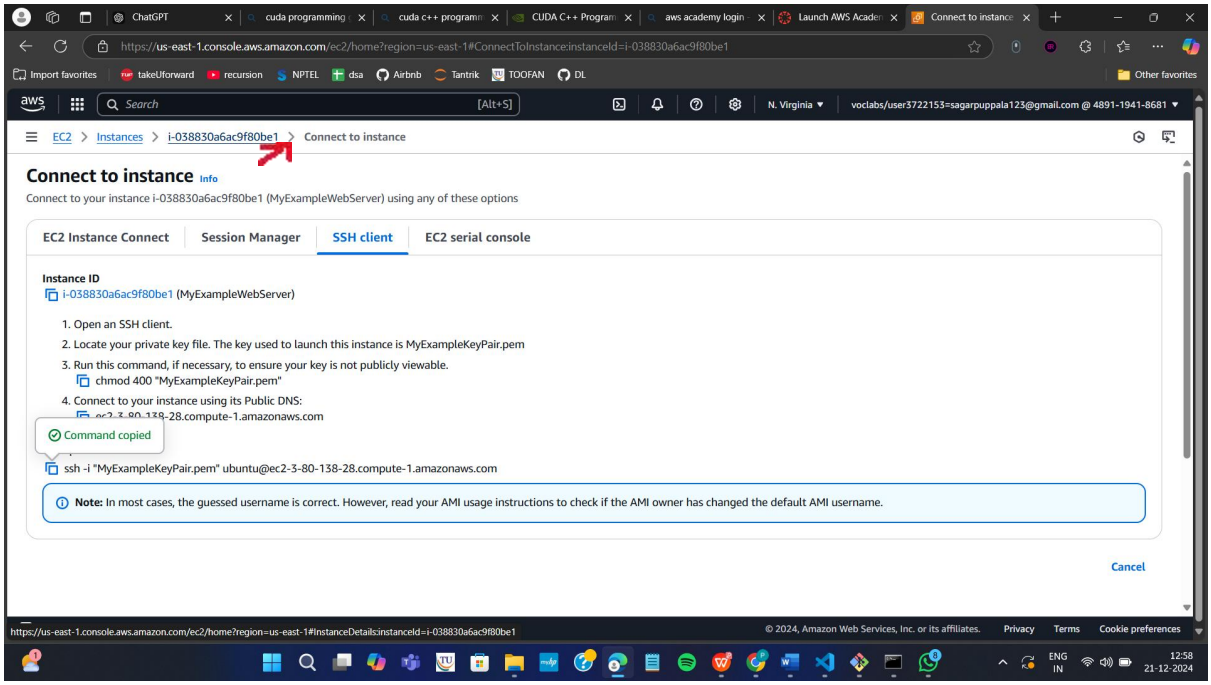
Build docker image by executing the following command

```
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
fbbf7d28be71: 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
```

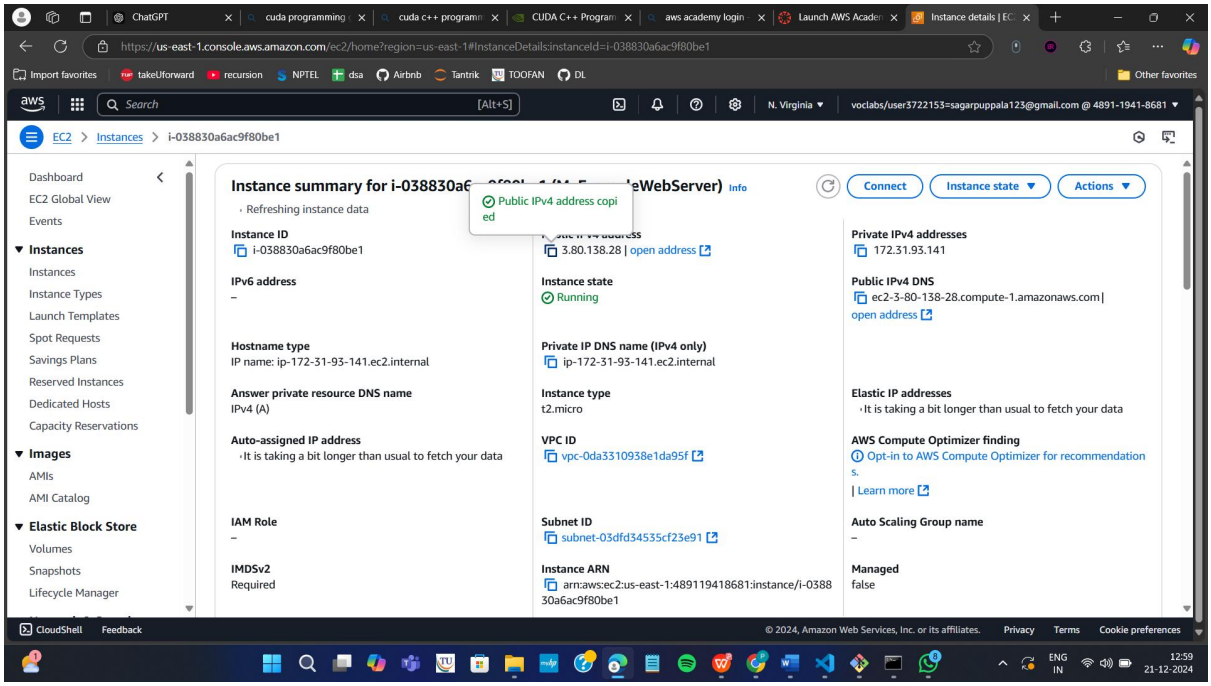
Run the container

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

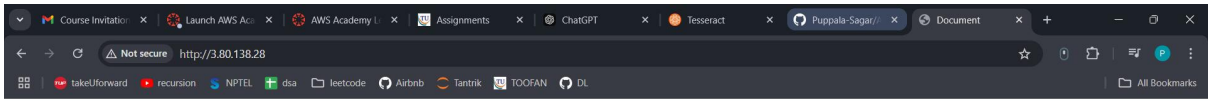
Click on



Copu public ip



Paste it in the browser

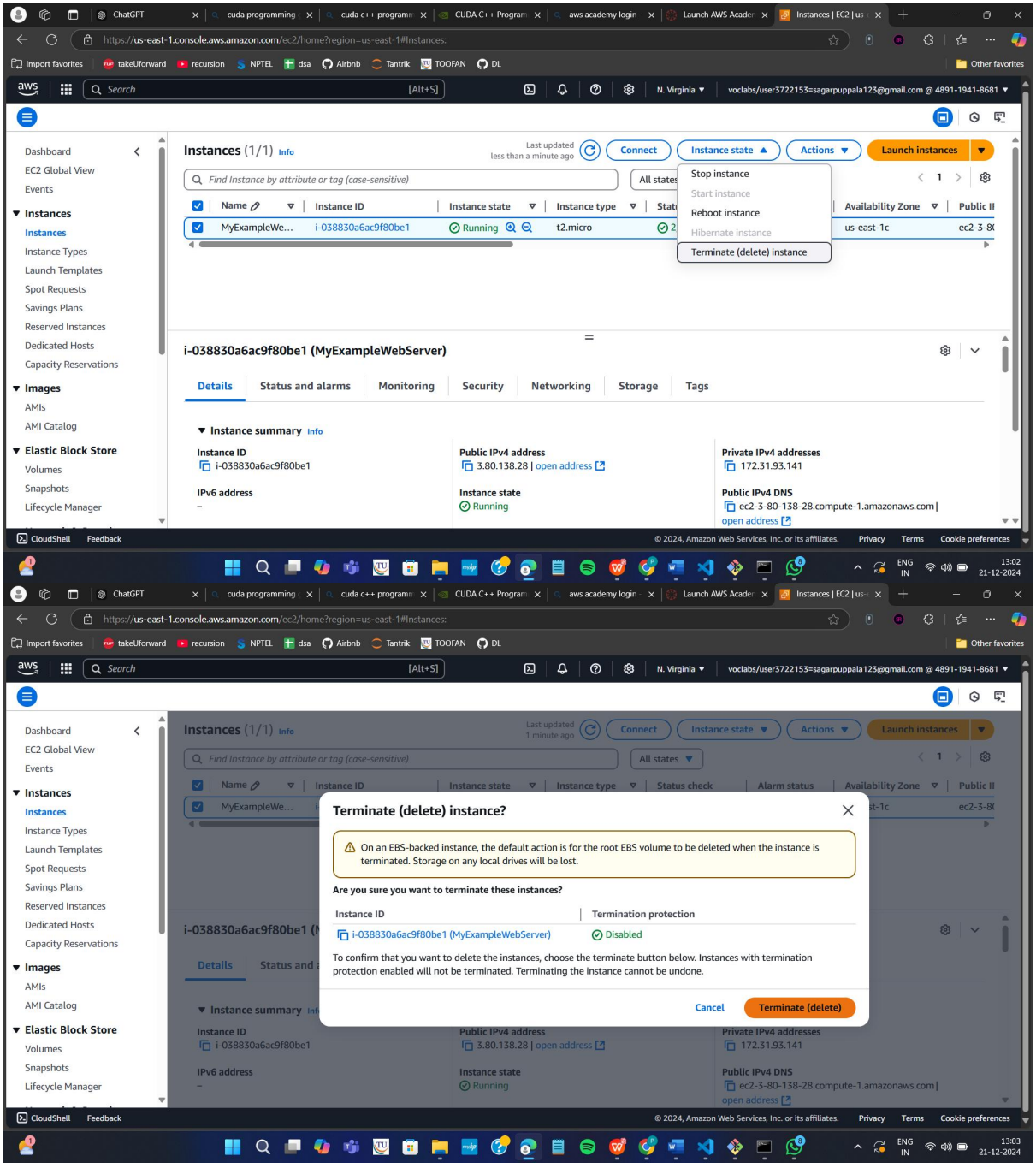


Hello from AWS

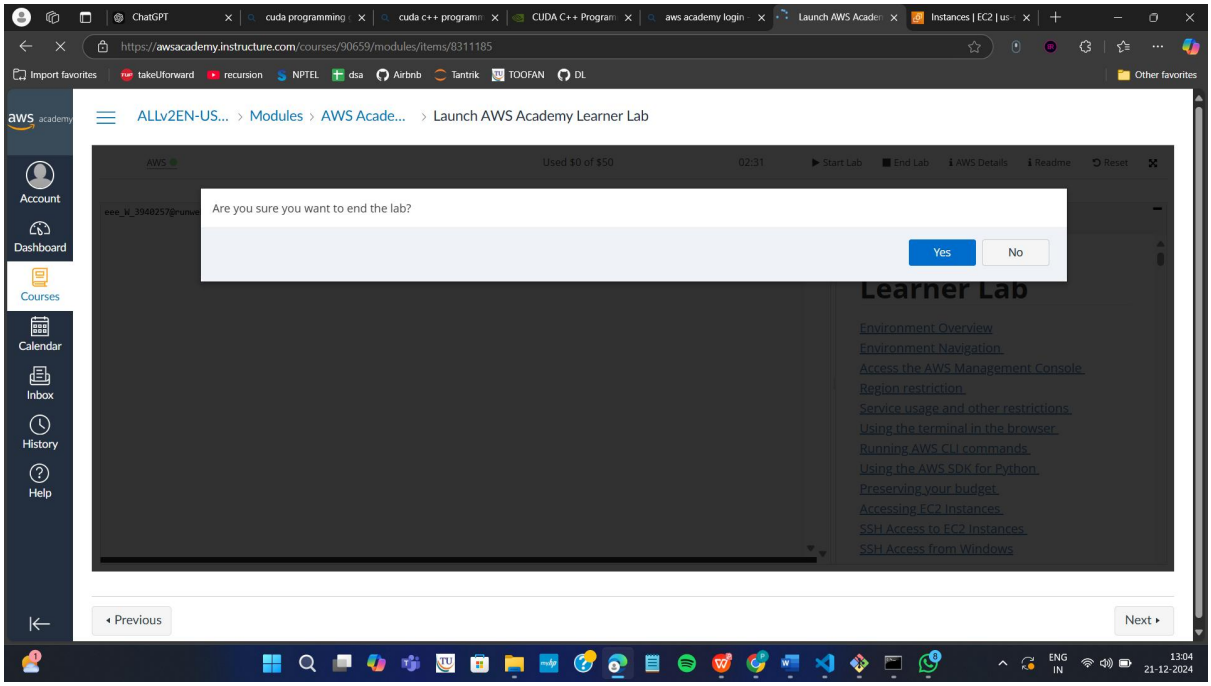
Run the following command to stop the container

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

Click on terminate instances

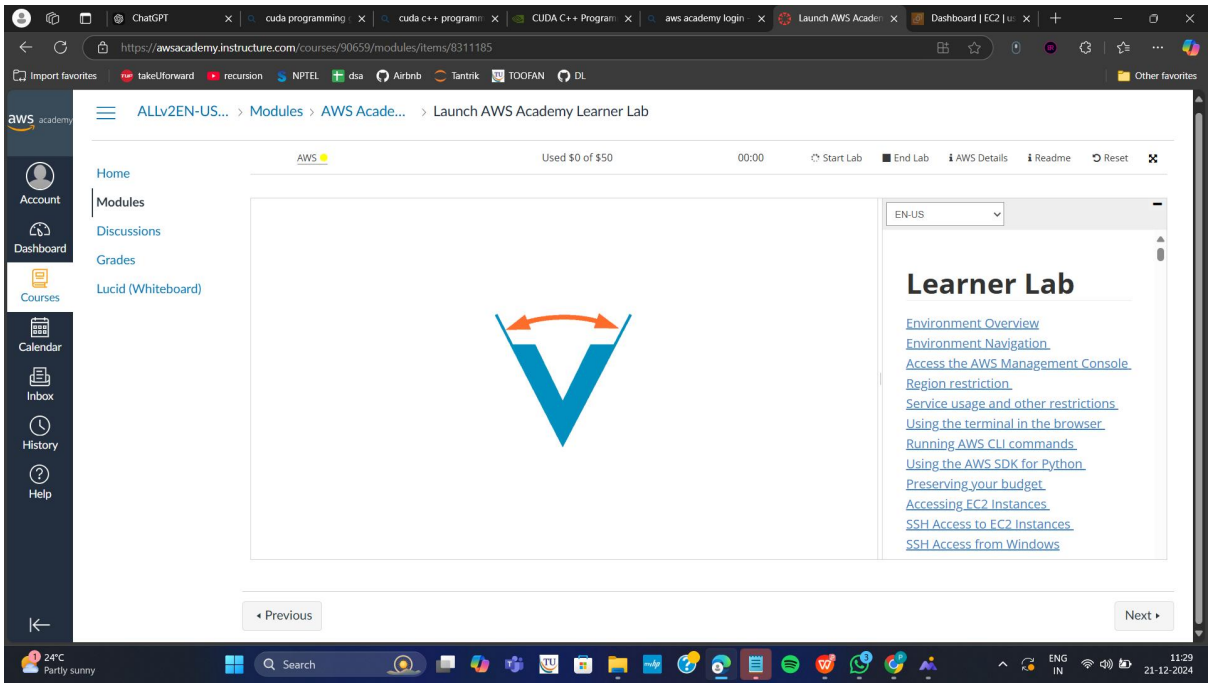


Now click on end lab

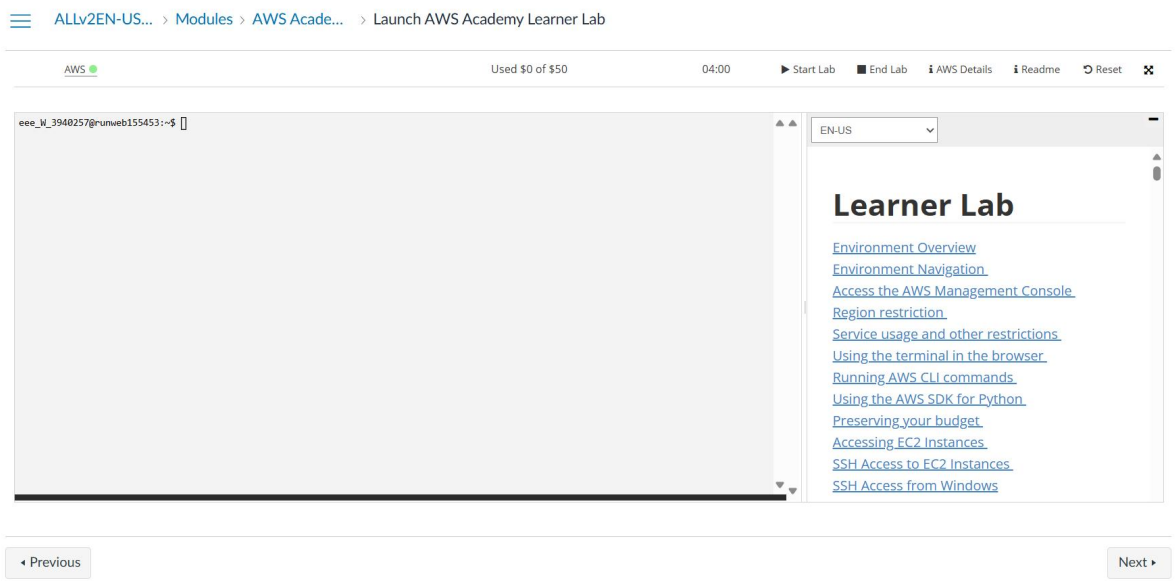


2 MAVEN WEB PROJECT DEPLOYMENT IN AWS

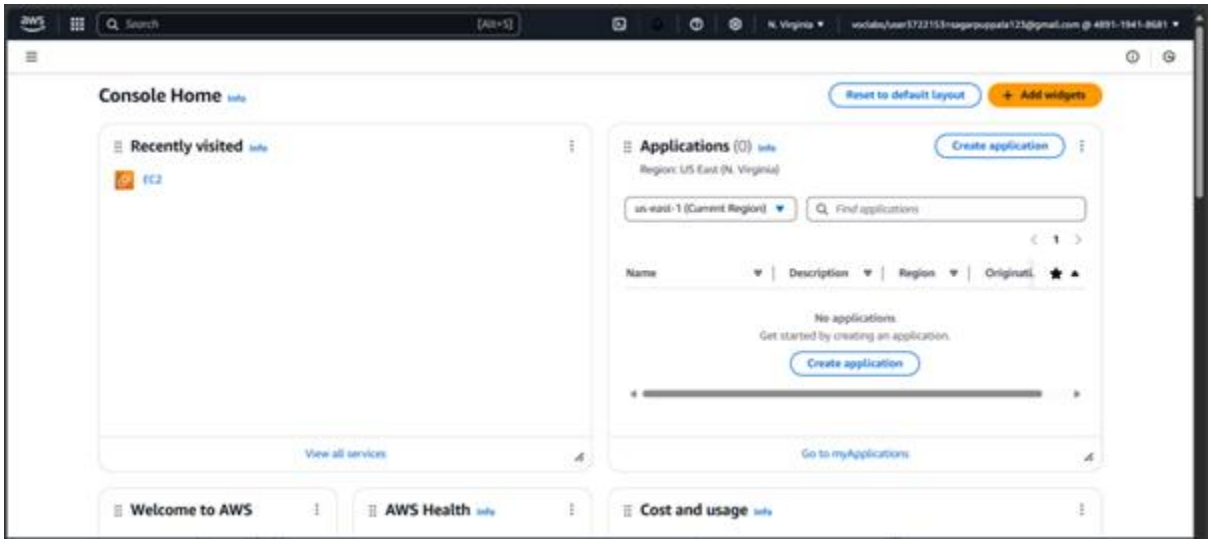
Click on start lab



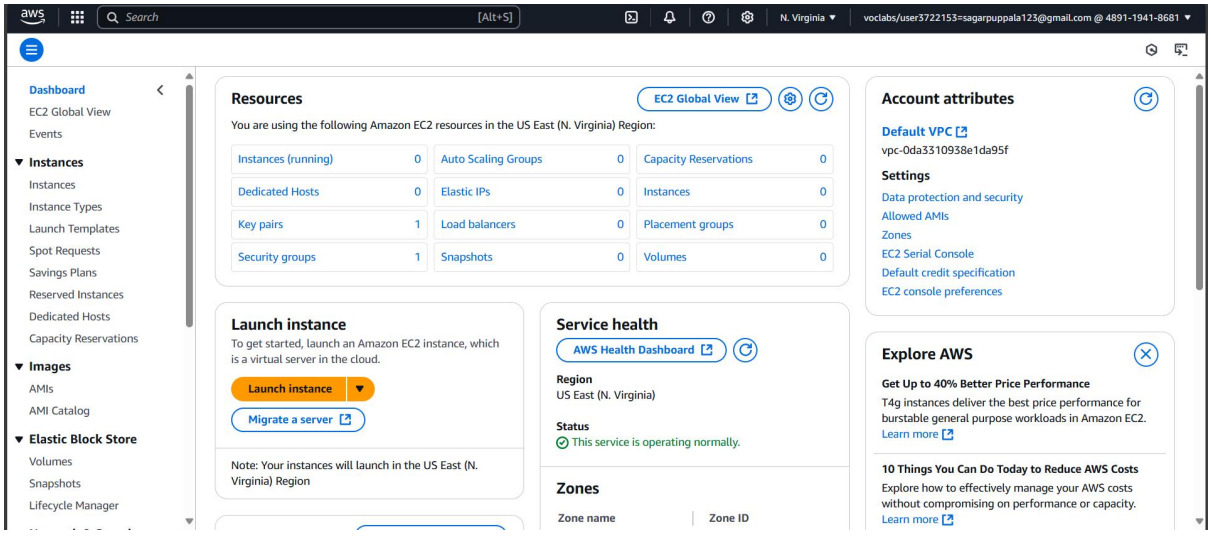
Click on AWS



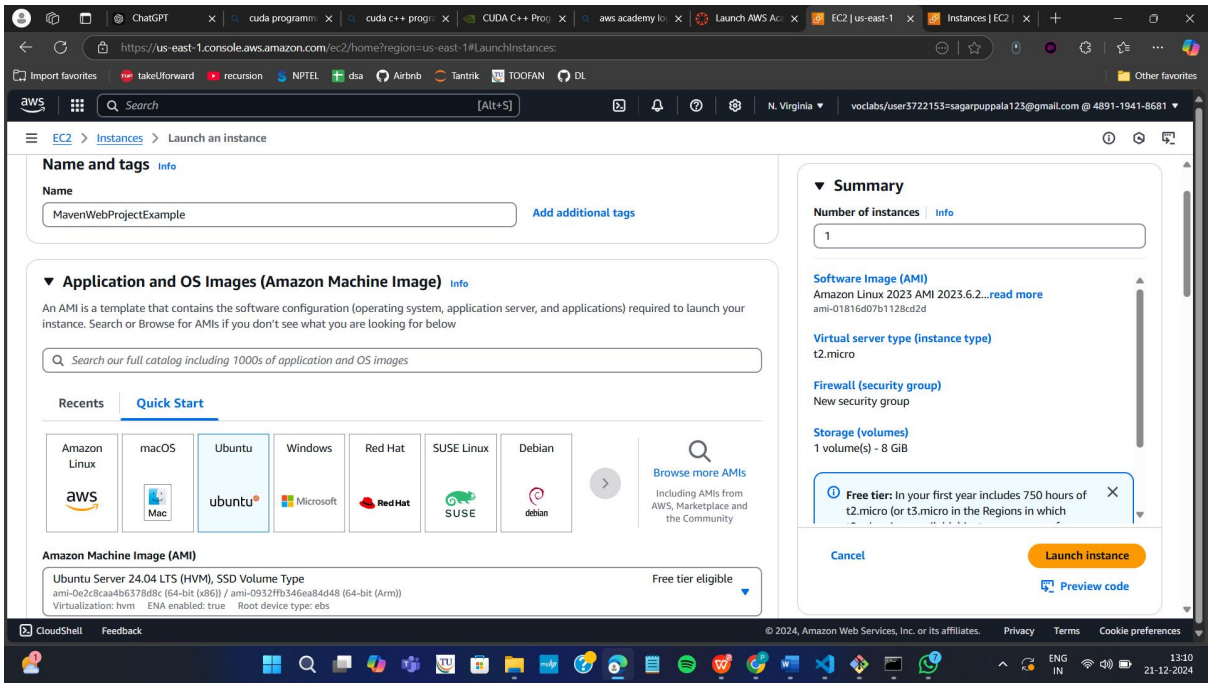
Click on EC2



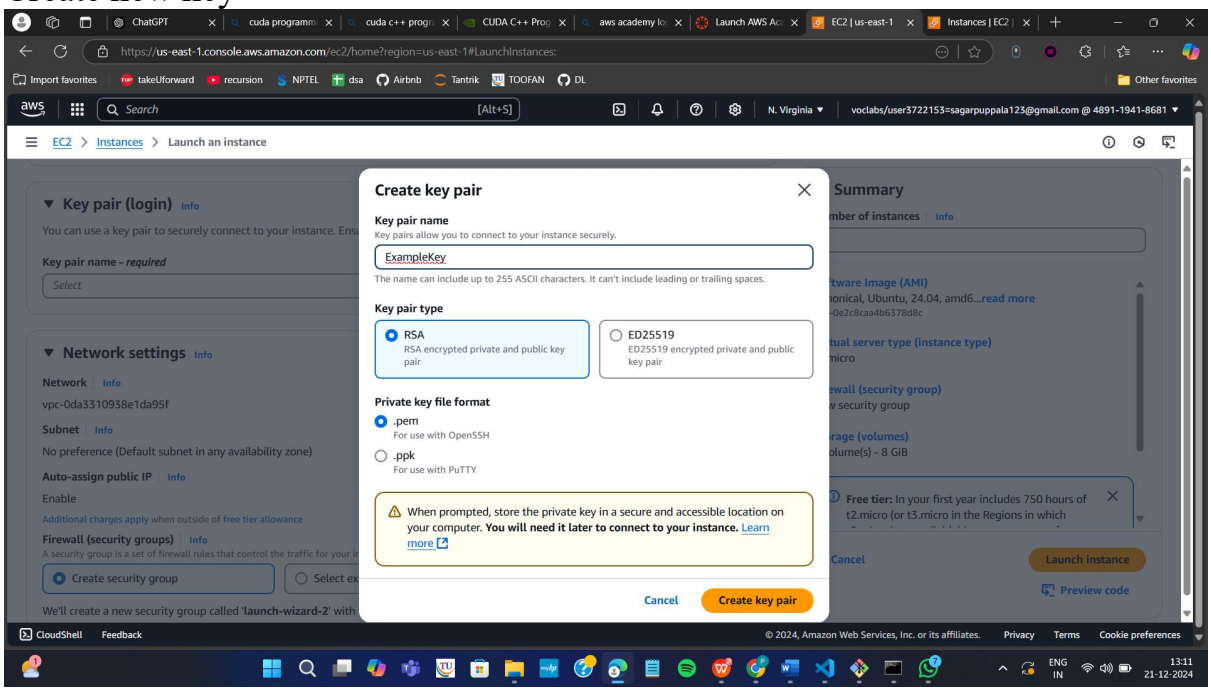
Click on launch instance



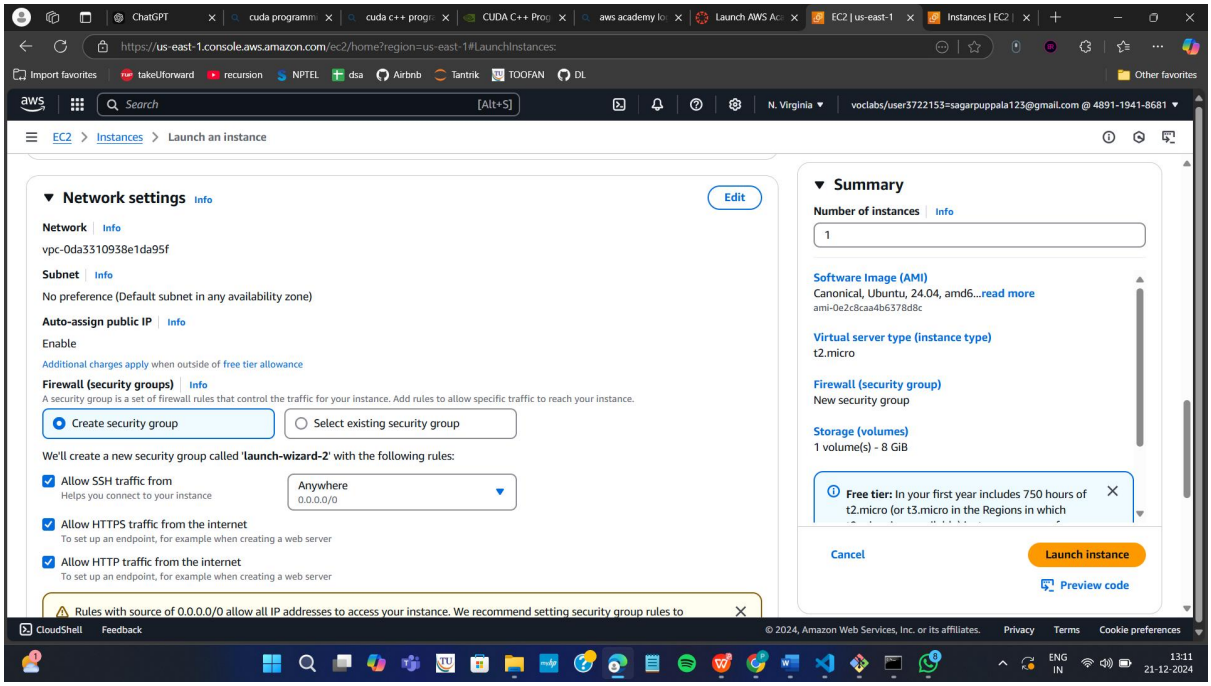
Give name



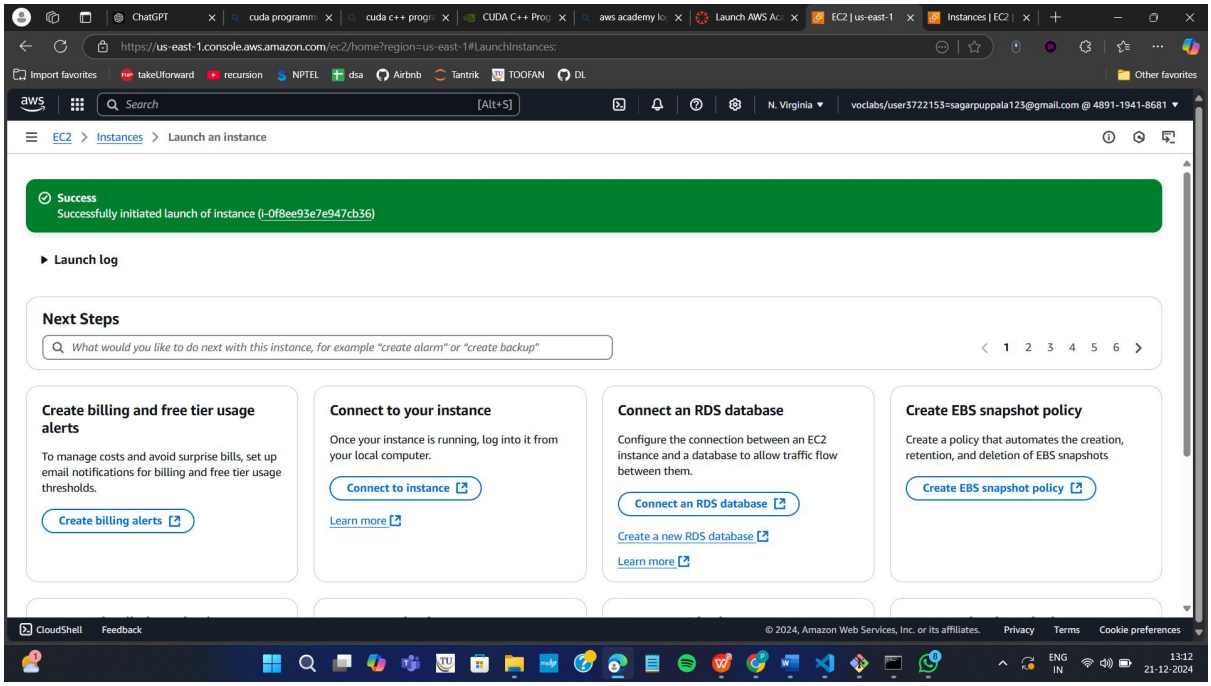
Create new key



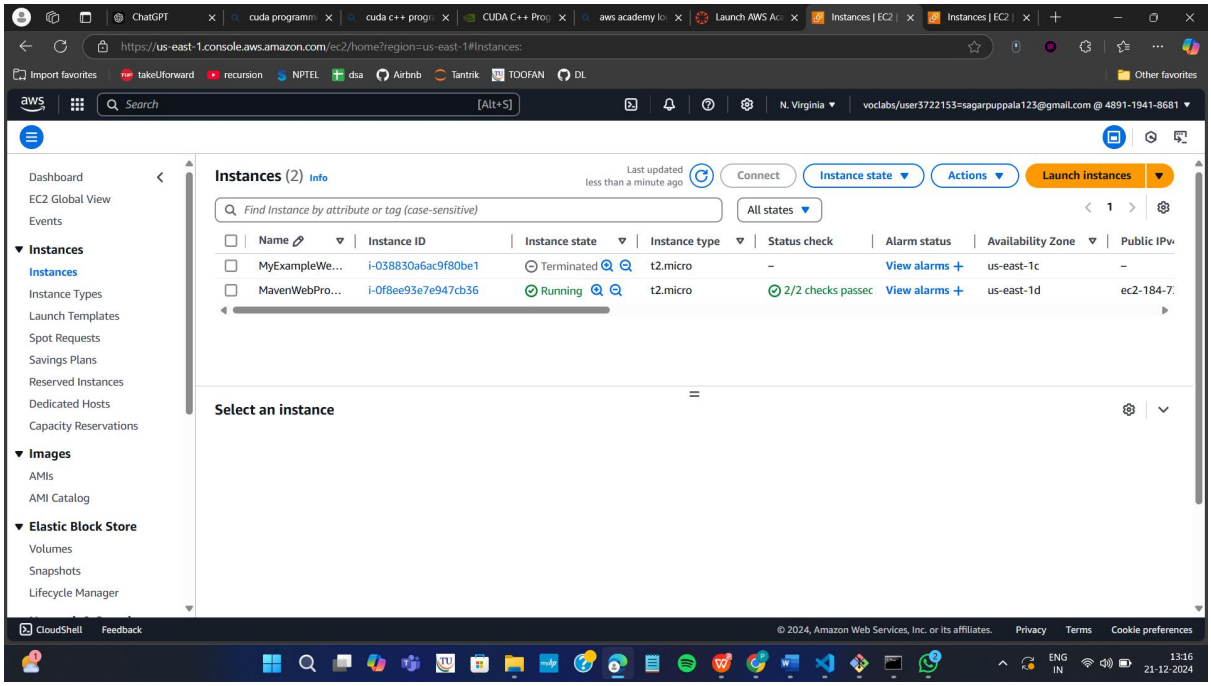
Check all the boxes under network and click on launch instance



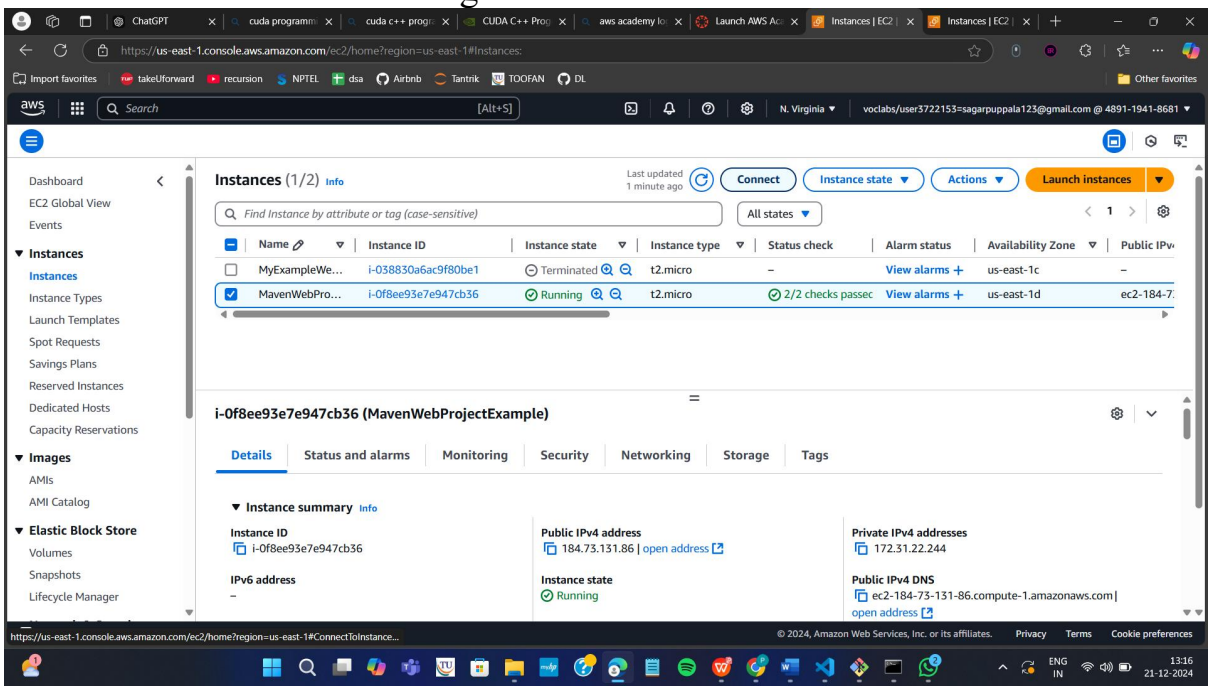
Click on instances



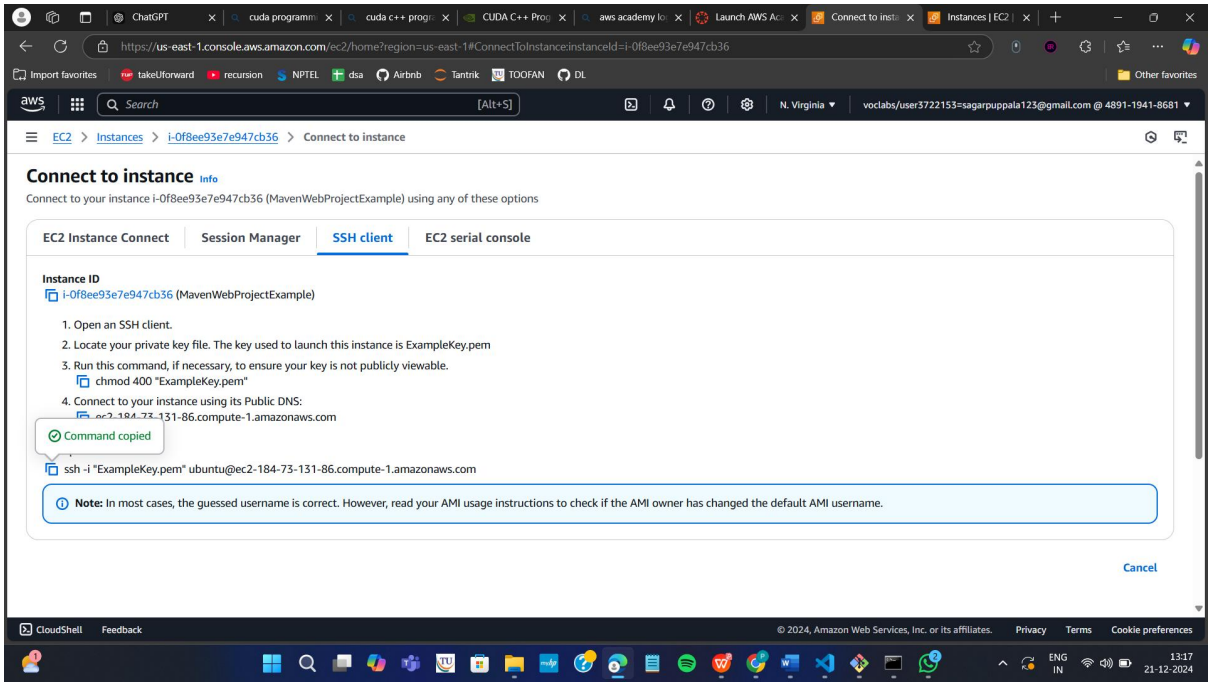
Wait until you get 2 tests passes



Click on connect after checking the box



Copy ssh command



Open the terminal, Navigate to the path

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

Run the ssh command

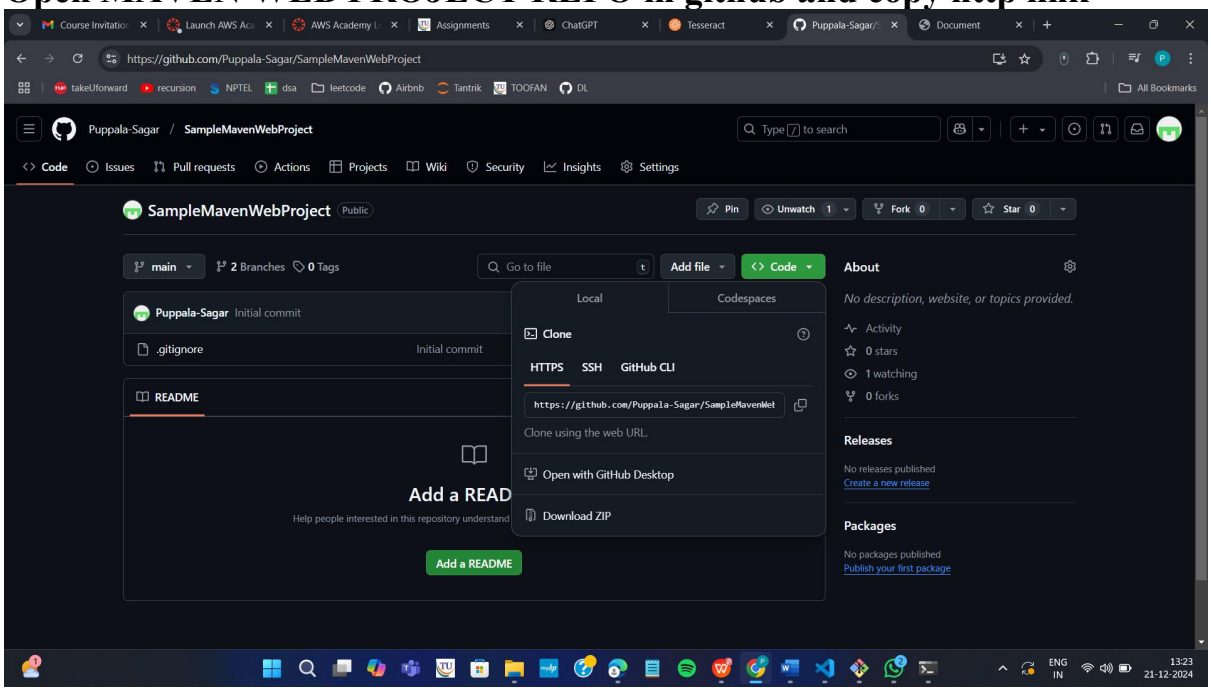
```
C:\Users\sagar\Downloads\AWS>ssh -i "ExampleKey.pem" ubuntu@ec2-184-73-131-86.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * 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 [15
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
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components
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 Metad
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages
```

```
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-compos
  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:~$ sudp apt install git
Command 'sudp' not found, did you mean:
  command 'ssdp' from snap ssdp (0.0.1)
  command 'sfdp' 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



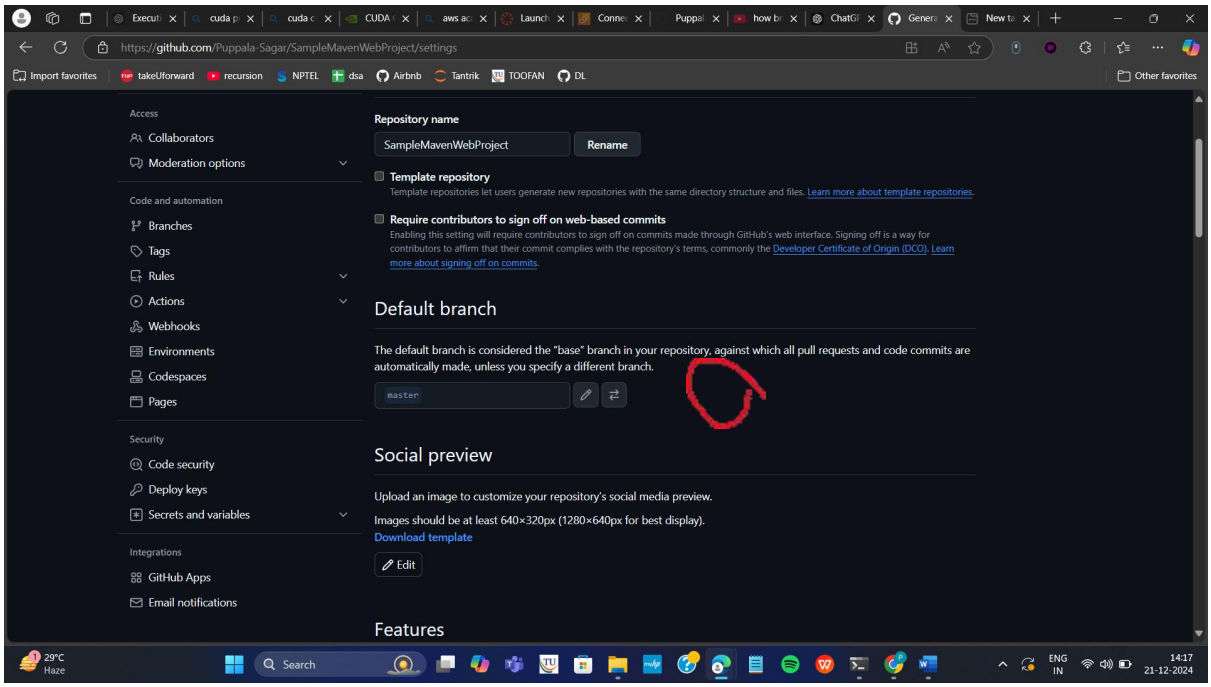
Clone the repository

```
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:~$ 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
```

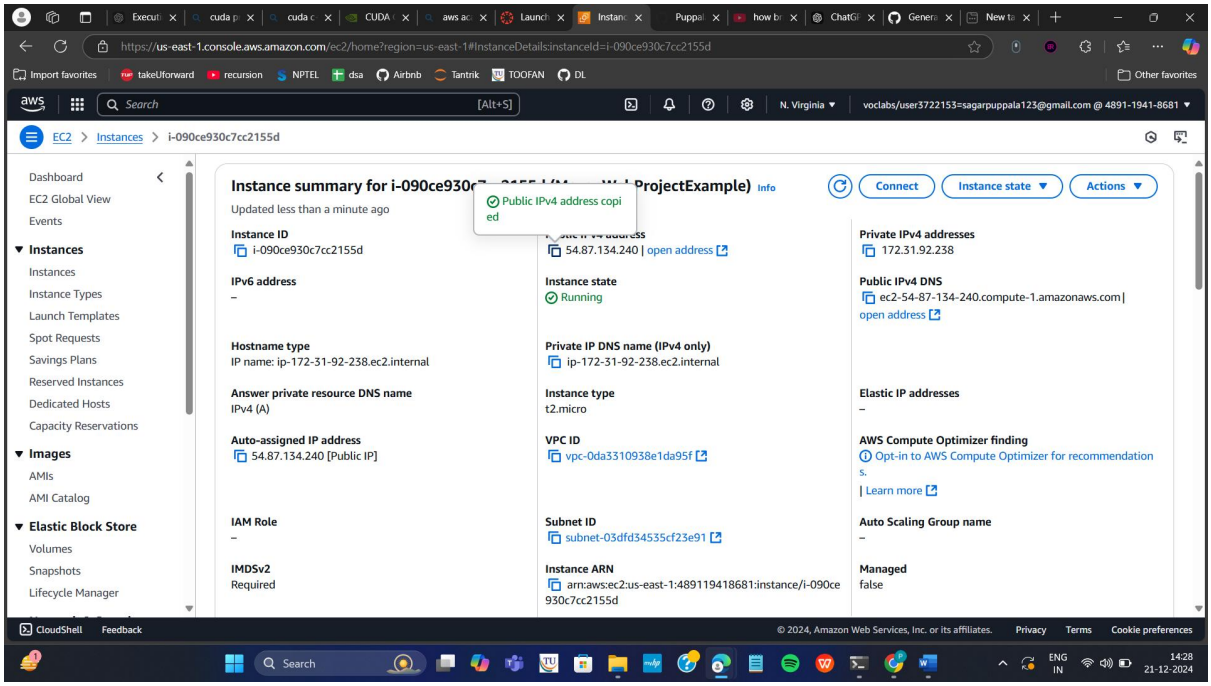
Build docker image

```
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
6d3bf2e80222: Download complete
9ab2f23fa0e1: Download complete
4f4fb700ef54: Download complete
cdbea10bf012: Download complete
```

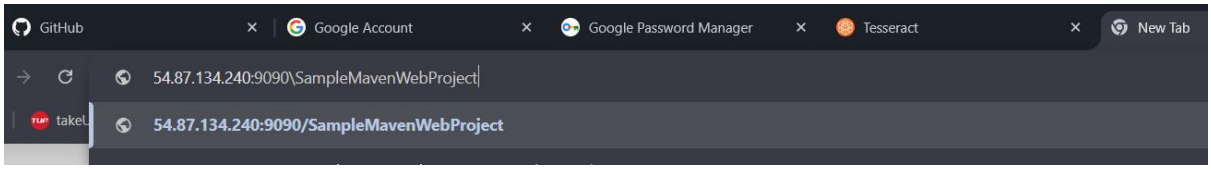
Run the container

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

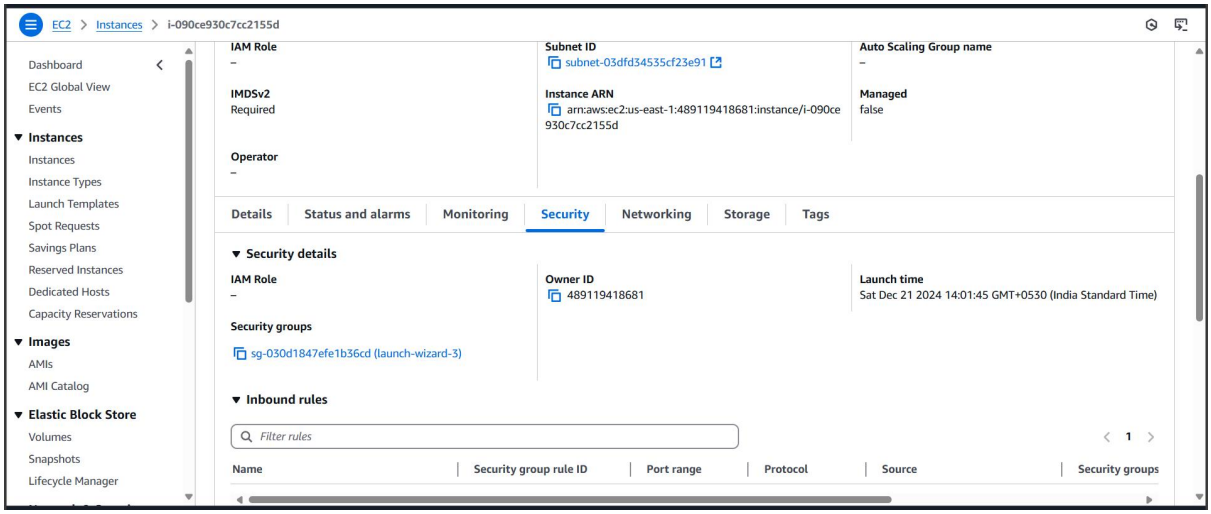
Copy public ipv4



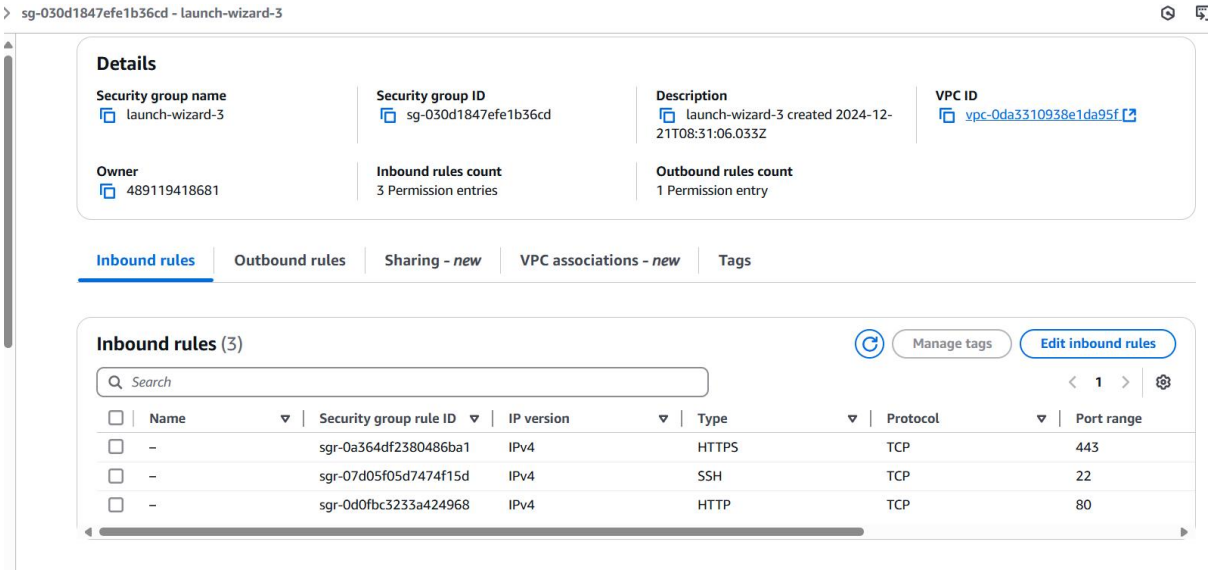
Enter this url in browser and click enter



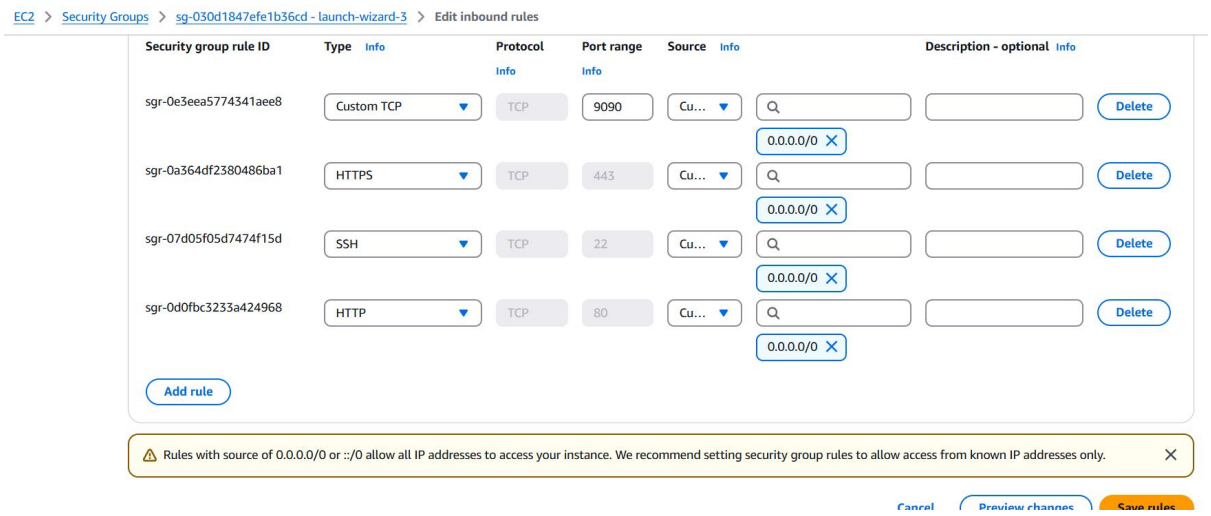
If your app is not running goto security and Click on security groups



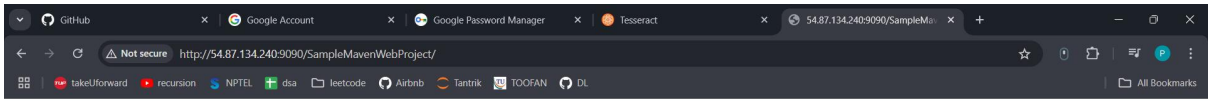
Click on edit inbound rules



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



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



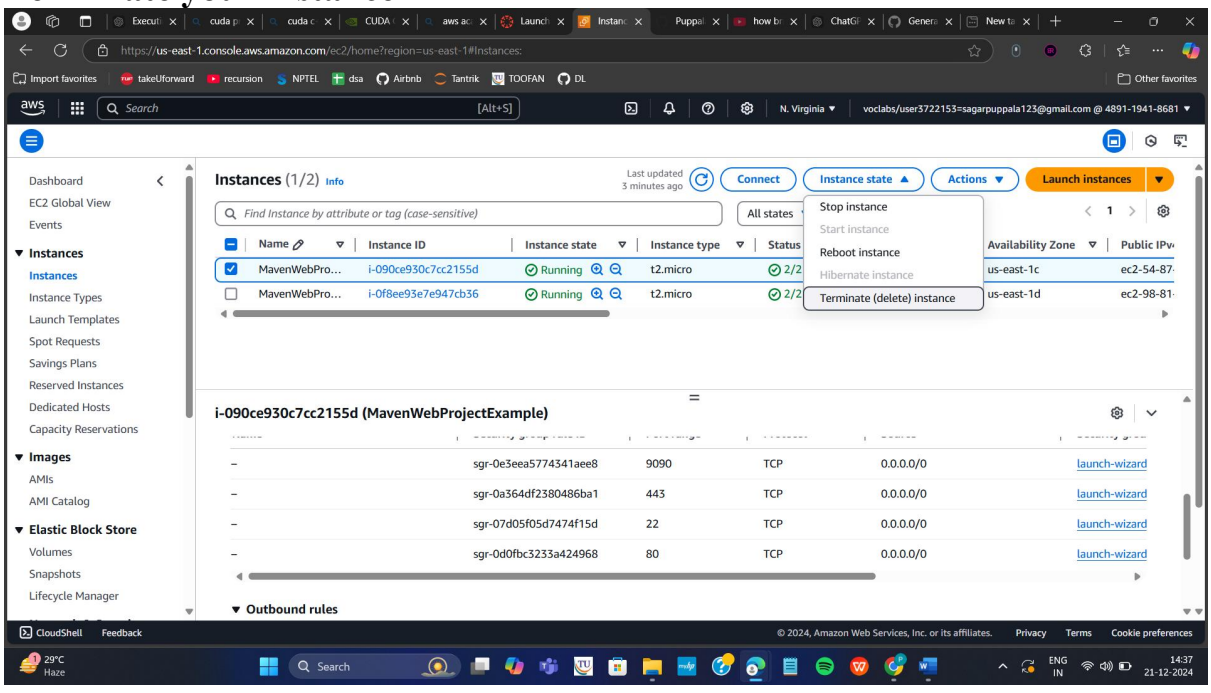
Hello World!

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

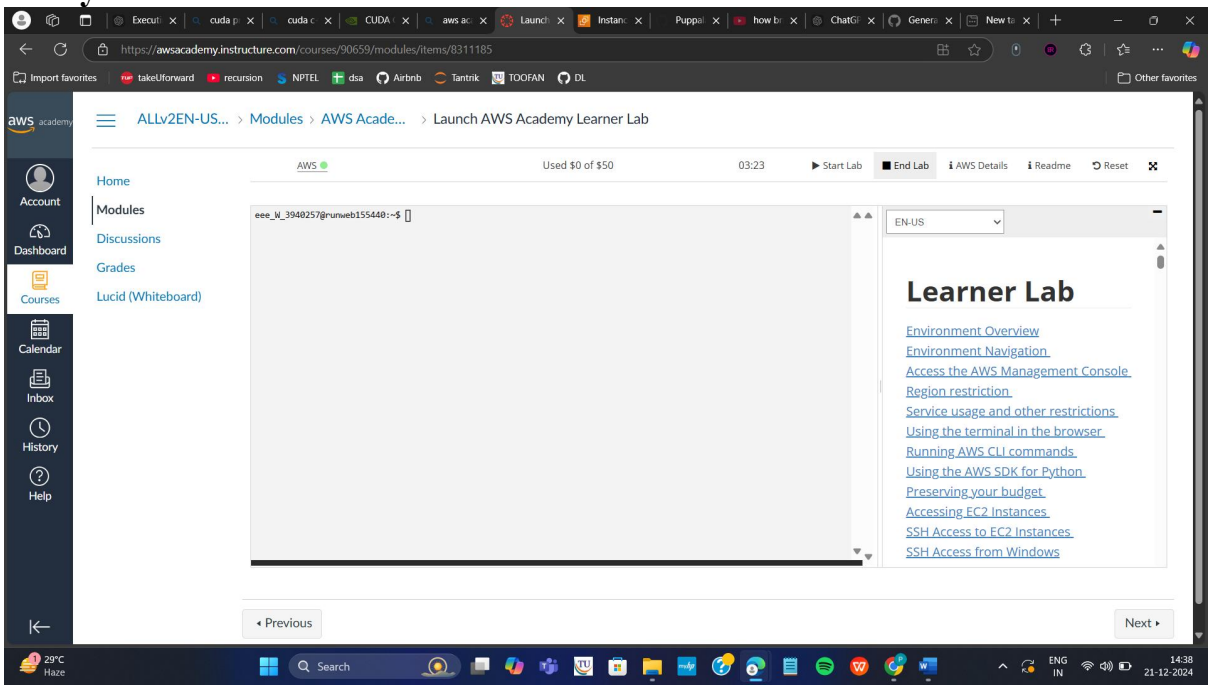
Run the following commands to stop the container

```
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
e83304610c89
ubuntu@ip-172-31-92-238:~/SampleMavenWebProject$
```

Terminate your instance



End your lab


https://awsacademy.instructure.com/courses/90659/modules/items/8311185
ALLv2EN-US... › Modules › AWS Acade... › Launch AWS Academy Learner Lab
Used $0 of $50
03:23
Start Lab
End Lab
AWS Details
Readme
Reset
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
Account
Dashboard
Courses
Calendar
Inbox
History
Help
eee_W_3940257@runweb155440:~$
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
Search