



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



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



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB RECORD

SOFTWARE ENGINEERING LAB

B.Tech. III YEAR I SEM (KR21)

ACADEMIC YEAR 2023-24



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.



Daily Laboratory Assessment Sheet

Name of the Lab:

Name of the Student:

Class:

HT.No:

Faculty Incharge

INDEX

S.NO	CONTENTS	PAGE NO
I	Vision/Mission /PEOs/POs/PSOs	6
II	Syllabus	12
III	Course outcomes, CO-PO Mapping	14
Exp No	List of Experiments	
1	<p>Title of Project</p> <p>a) Development of a problem statement. b) Preparation of Software Requirement Specification Document, Design Documents, and related documents. c) Study and use any Design phase CASE tool</p>	
2	<p>Working with Git and GitHub</p> <p>a) Git Installation b) Git configuration c) GitHub account creation</p>	
3	<p>Working with Git commands</p> <p>Git commands – help, pwd, mkdir, notepad, ls, init, ls – la, status, add, commit, log, diff, staged, log –online, revert, branch, checkout, merge, reset</p>	
4	<p>Working with GitHub commands</p> <p>a) Creating repository b) Using GitBash for following commands remote, remote -v, push to push a sample project from local to remote repository c) Creating SSH key in GitHub d) Using GitBash for SSH-Keygen, clone, ls – ltr, push to push a sample project using ssh key from local to remote repository.</p>	
5	<p>Working with Jenkins</p> <p>a) JDK 11 installation and setting environmental path b) Jenkins installation c) Tomcat 9 installation and setting environmental path d) Eclipse installation and Updating e) Configuring Tomcat 9 with Eclipse f) Maven installation and setting environmental path g) Jenkins Global configurations and installing additional 5 pipeline plugins</p>	
6	<p>Working with Maven in Eclipse</p> <p>a) Create a Maven java project and run it in Eclipse environment b) Create a Maven Web Project and run it in Eclipse environment (by adding dependencies) c) Push the above two projects into GitHub.</p>	

S.NO	CONTENTS	PAGE NO
7	<p>Working with Jenkins</p> <ul style="list-style-type: none"> a) Create a CI/CD pipeline using Jenkins for a sample Maven Java project b) Create a CI/CD pipeline using Jenkins for a sample Maven Web project. 	
8	<p>Working with Docker</p> <ul style="list-style-type: none"> a) Docker Installation b) Docker CLI commands c) Pulling an existing image from Docker Hub and running the image as container using run command d) Creating own image of Ubuntu and running as container, pushing the image into Docker Hub 	
9	<p>Working with Docker</p> <ul style="list-style-type: none"> a) Pulling and running the services, Nginx and Tomee b) Use Docker compose with DockerFile c) Use Docker compose with yaml file. 	
10	<p>Working with Kubernetes</p> <ul style="list-style-type: none"> a) Minikube installation b) Start and run minikube c) Using the following minikube commands – version, status d) Create and deploy Nginx service e) Using the following minikube commands – get deployment, get pods, describe pods, scale deployment, expose, service f) Run the Nginx server on the given URL. g) Delete the deployment, Stop minikube 	
11	<p>Working with Nagios</p> <ul style="list-style-type: none"> a) Install Nagios using Docker b) Run the Nagios container, open the browser and monitor localhost. 	
12	<p>Working with AWS</p> <ul style="list-style-type: none"> a) Create an Amazon Web Services (AWS) free account b) Create an Elastic Cloud Computing (EC2) instance of Ubuntu. 	
13	<p>Working with AWS</p> <p>Access Nginx server using EC2 instance with Browser</p>	



Department of Computer Science & Engineering

Vision of the Institution:

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

Mission of the Institution:

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



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering

Vision of the Department:

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the Department:

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.



Department of Computer Science & Engineering

PROGRAM OUTCOMES (POs)

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

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

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

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

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

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

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

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

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

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

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

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



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering

PROGRAM SPECIFIC OUTCOMES (PSOs)

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

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



KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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



Department of Computer Science & Engineering

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

B. Tech. in COMPUTER SCIENCE AND ENGINEERING**III Year I Semester Syllabus (KR21)****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. PP102ES – Programming for Problem Solving Course
2. 21CS403PC – Operating Systems Course
3. 21CC502PC – Software Engineering Course
4. 21CS401PC- Java Programming Course

Course Objectives: The course will help to

1. Develop the process of problem statement.
2. Understand the process of development of Software Requirement Specifications.
3. Have hands on experience in Design, develop and testing various modules in a project.
4. Understand the usage of GitHub and Jenkins.
5. Devise the deployment of project in AWS cloud using Docker and Kubernetes.

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

1. Outline to translate end-user requirements into system and software requirements.
2. Illustrate a high-level design of the system from the software requirements.
3. Apply usecase tools in design phase.
4. Use Jenkins to build project.
5. Devise a project in AWS cloud using Dockers and Kubernetes.

Software to be used: The students must use JDK 11 Version, STAR UML, GIT Bash, Jenkins, Dockers Desktop, Mini KUBE, Eclipse, Tomcat, 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.
6. Building the project in Jenkins
7. Deploying the project in AWS cloud using Docker and Kubernetes and monitoring using Nagios Tool

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 HillInternational Edition, 2015.
2. Software Engineering- Sommerville, 7th edition, Pearson Education, 2017.
3. The unified modeling language user guide Grady Brooch, James Rumbaugh, Ivar Jacobson, PearsonEducation, 2016.
4. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2015.

REFERENCE BOOKS:

1. Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale, 2018.
2. Cloud Native DevOps with Kubernetes by John Arundel, 2016.



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Course Objectives:

- Understand the process of development of a problem statement.
- Understand the process of development of Software Requirement Specifications
- Understand the process of development of Design documents and testing phase-related documents.
- Understand the usage of GitHub and Jenkins
- Understand the deployment of projects in the AWS cloud using Docker and Kubernetes.

Course Outcomes:

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

CO 1: Outline to translate end-user requirements into system and software requirements

CO 2: Illustrate a high-level design of the system from the software requirements

CO 3: Use case tools in the design phase of application development.

CO 4: Use Jenkins to build the project

CO 5: Setup a project in the AWS cloud using Dockers and Kubernetes.

CO-PO MAPPING:

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

CO-PSO MAPPING:

	PSO-1	PSO-2
CO1	2	
CO2	2	1
CO3		2
CO4		2
CO5		2

Software and Hardware Requirements

1. Software Tools

- a. Star UML**
- b. Java 11**
- c. Apache Tomcat 9**
- d. Eclipse IDE**
- e. Visual Studio**
- f. Jenkins**
- g. Git Bash for Windows**
- h. Windows Docker Desktop**
- i. AWS Cloud account (Basic Plan)**

2. Hardware Requirements

- a. Windows 10 Pro**
- b. 8 GB RAM (64-bit Processor)**
- c. 512 GB HDD**
- d. i5 Core Processor**
- e. BIOS-level hardware virtualization support is enabled in the BIOS settings.**

Experiment 2:

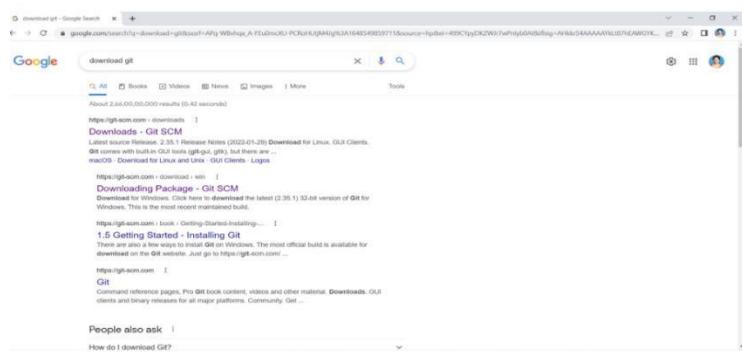
Working with Git and GitHub

- a) Git Installation
- b) Git configuration
- c) GitHub account creation

a) INSTALLATION OF GIT:

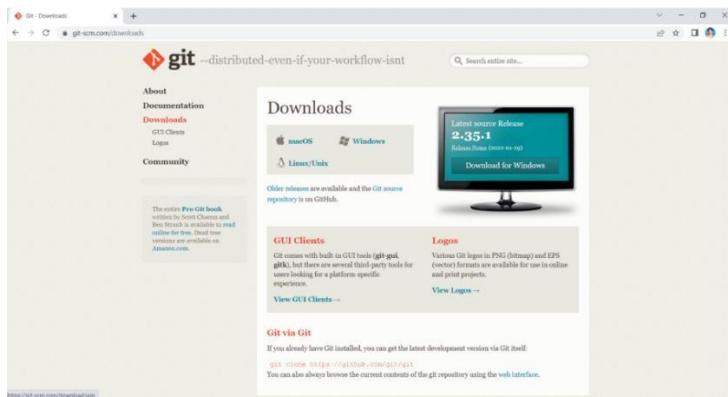
1. We can type download git from google.com and click on the first link Git SCM

INSTALLING GIT IN WINDOWS



2. Select the Git software for the required OS.

INSTALLING GIT IN WINDOWS

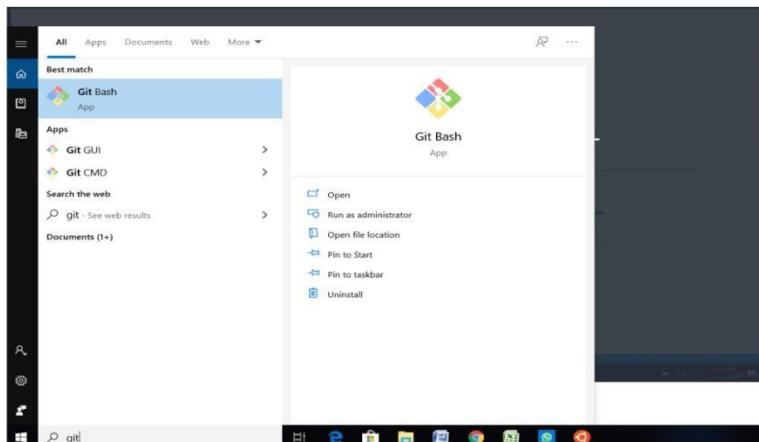


3. Install the Git software downloaded



4.While installing select the option Use Git from Git Bash only option and click Next.

INSTALLING GIT IN WINDOWS



5.Check the version of git using git—version..

INSTALLING GIT IN WINDOWS

```
MINGW64:/c/Users/Madhu
$ git --version
git version 2.35.1.windows.2
$ |
```

A screenshot of a terminal window titled "MINGW64:/c/Users/Madhu". The command \$ git --version is run, and the output "git version 2.35.1.windows.2" is displayed. The window has a dark background and a light-colored text area.

Experiment 3:

Working with Git commands

Git commands – help, pwd, mkdir, notepad, ls, init, ls – la, status, add, commit, log, diff, staged, log – online, revert, branch, checkout, merge, reset

Experiment 4:

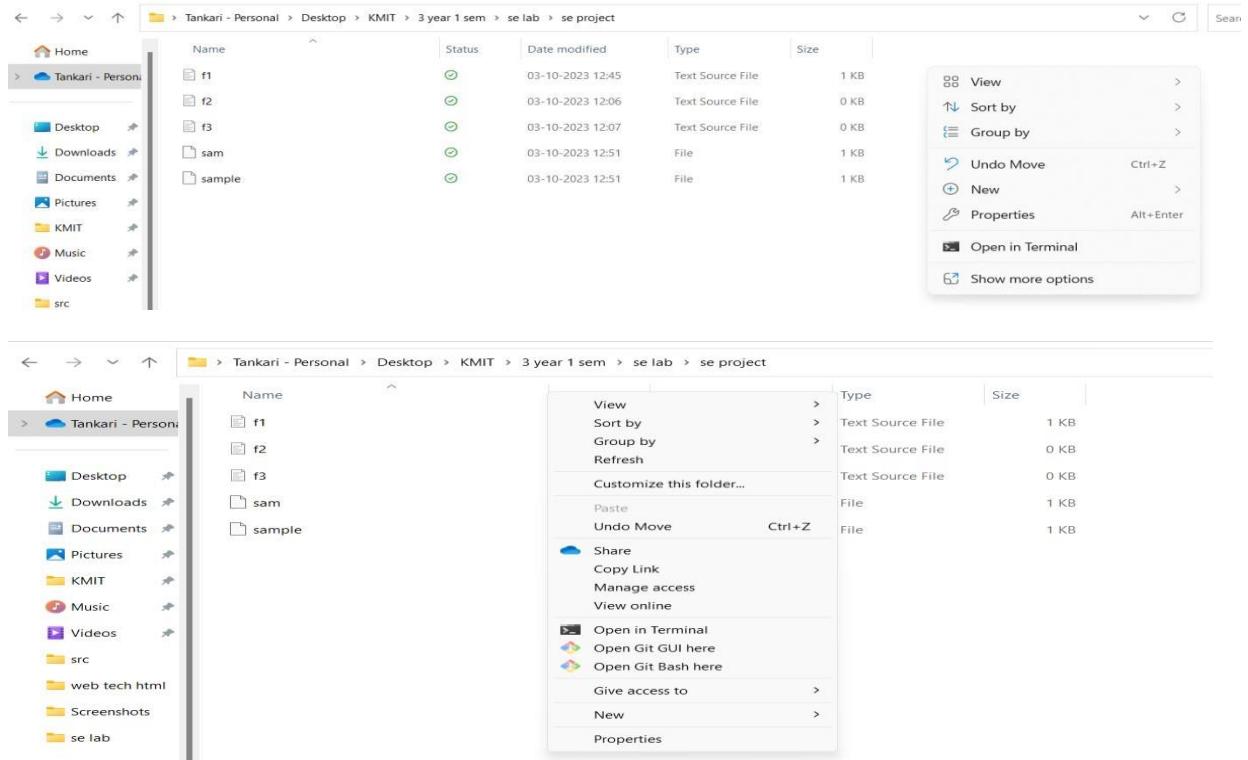
Working with GitHub commands

- a) Creating repository
- b) Using GitBash for following commands remote, remote -v, push to push a sample project from local to remote repository
- c) Creating SSH key in GitHub
- d) Using GitBash for SSH-Keygen, clone, ls – ltr, push to push a sample project using ssh key from local to remote repository.

(NOTE : Both experiment 3 and 4 are shown below)

GIT COMMANDS

You can open Git Bash in a specific folder by right-clicking anywhere in the folder and choosing the Git Bash option



click on git bash here

```
MINGW64:/c/Users/User/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project (master)
$ |
```

Commands

Step-6:

The user id and email are added to the gitbash by using:

```
git config --global user.name"user-id" git
config --global user.email"user-email"
```

The details added to the gitbash can be

known by using: `git config --global --list`

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git config --global user.name "tankariadarsh"
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git config --global user.email "tankariadarsh32@gmail.com"
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git config --global --list
user.name=Tankariadarsh
user.email=tankariadarsh32@gmail.com
```

Step-7:

The information about the present working directory can be known by using:

```
pwd
```

list of files in present working directory can be known by using:

```
ls
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git
$ pwd
/c/Users/User/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git
$ ls
f1.txt  f2.txt
```

Step-8:

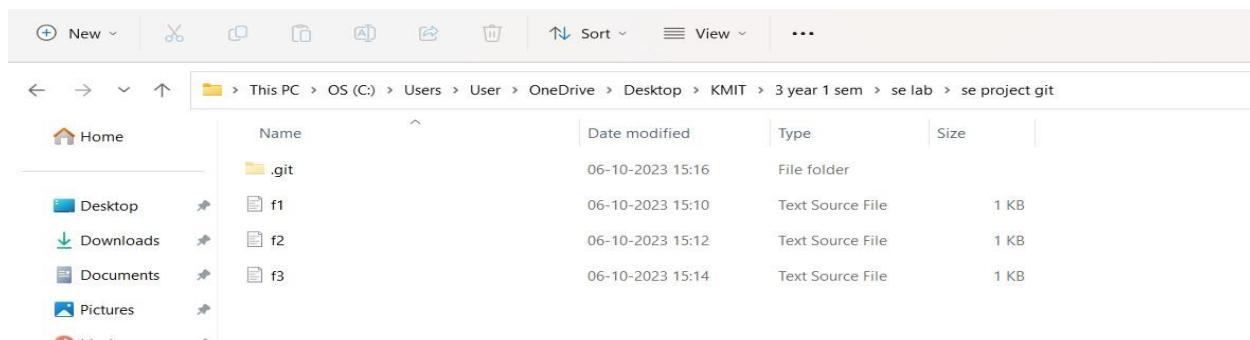
Initialization of empty git repository can be done by using

```
git init
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git
$ git init
Initialized empty Git repository in c:/Users/User/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git/.git/
```

After initialization of empty git repository “.git” folder appears in the folder and (master) appears in the command line

If “.git folder” was not shown on the folder then go to view and click on the hidden files



Step-9:

By using “git status” command ,it displays the state of working directory and the staging area(master)

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1.txt
    f2.txt
    f3.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Step 10:

git add <file name> - helps to add file from untracked files to tracked files

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git add f1.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   f1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f2.txt
    f3.txt
```

Step-11:

git rm –cached <filename> - helps to remove file from tracked file and pushed back to untracked files

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git rm --cached f1.txt
rm 'f1.txt'

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1.txt
    f2.txt
    f3.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Step-12:

git commit -m “message” – helps to commit the file and gives an user friendly message with a unique ID on command line by using “**git log**” and “**git log -oneline**” command

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git add f1.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master

No commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   f1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f2.txt
    f3.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git commit -m"f1 text file commited"
[master (root-commit) 1fa1315] f1 text file commited
  1 file changed, 2 insertions(+)
  create mode 100644 f1.txt
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git log
commit 1fa1315ecd4253d2ced597b574e6ff711127dd65 (HEAD -> master)
Author: Tankariadarsh <tankariadarsh32@gmail.com>
Date:   Fri Oct 6 15:25:34 2023 +0530

  f1 text file commited

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git log --oneline
1fa1315 (HEAD -> master) f1 text file commited
```

Step -13:

By using vi editor the content can be changed in the file and can add new line to the files and a new file can be created

Vi <file name>

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git add f2.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f3.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ vi f2.txt
```

Step-14:

by using “**git diff**” command we can know what changes has been added to the text file the added text appers in the green color

And a new file is created with the modified text ,this file will be in stored in untracked file

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git diff
diff --git a/f2.txt b/f2.txt
index 47c619d..799563d 100644
--- a/f2.txt
+++ b/f2.txt
@@ -1,3 +1,7 @@
 hi
 helloo
 this is f2 text file
+
+
+
+heoooooooooooooooooooooooooooooooooooo
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   f2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f3.txt
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git commit -m"f2 text file committed unmodified"
[master dd28681] f2 text file committed unmodified
 1 file changed, 3 insertions(+)
 create mode 100644 f2.txt
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (master)
$ git log --oneline
dd28681 (HEAD -> master) f2 text file committed unmodified
1fa1315 f1 text file committed
```

Step 15:

By using “**git show <unique id>**” we can get the user friendly message that has provided when the file has committed and the data in the text file in green colour

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git show 1fa1315
commit 1fa1315ecd4253d2ced597b574e6ff711127dd65
Author: Tankariadarsh <tankariadarsh32@gmail.com>
Date:   Fri Oct 6 15:25:34 2023 +0530

f1 text file committed

diff --git a/f1.txt b/f1.txt
new file mode 100644
index 0000000..6e60cd8
--- /dev/null
+++ b/f1.txt
@@ -0,0 +1,2 @@
+hi
+this is f1 text file
```

Step-16:

“git remote add origin <link>” command adds the local repository in the github account

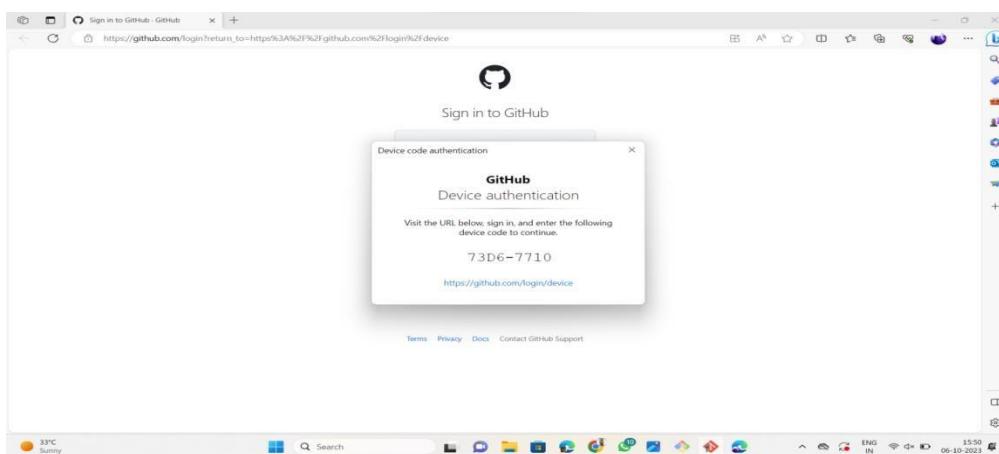
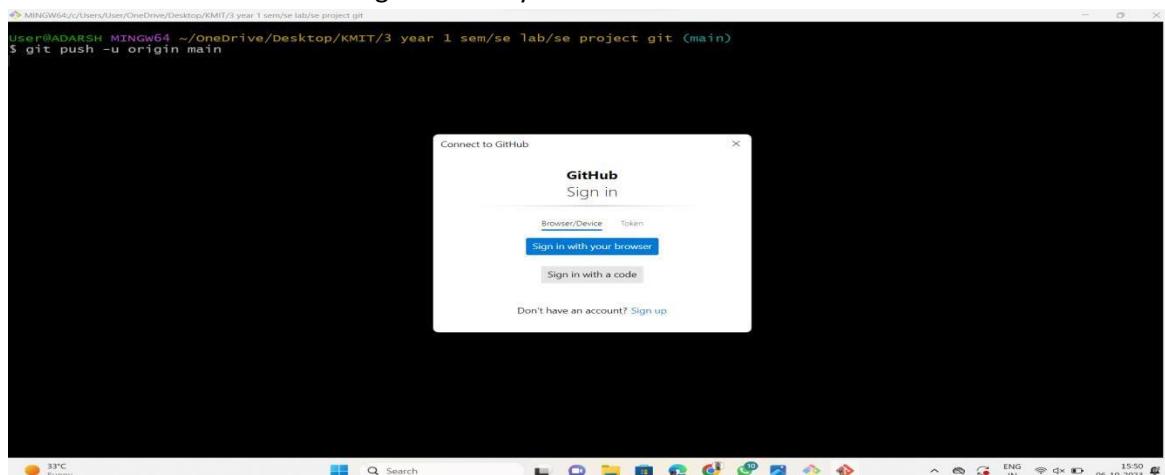
“git ls-remote” command gives the link to the repository where the committed files are stored

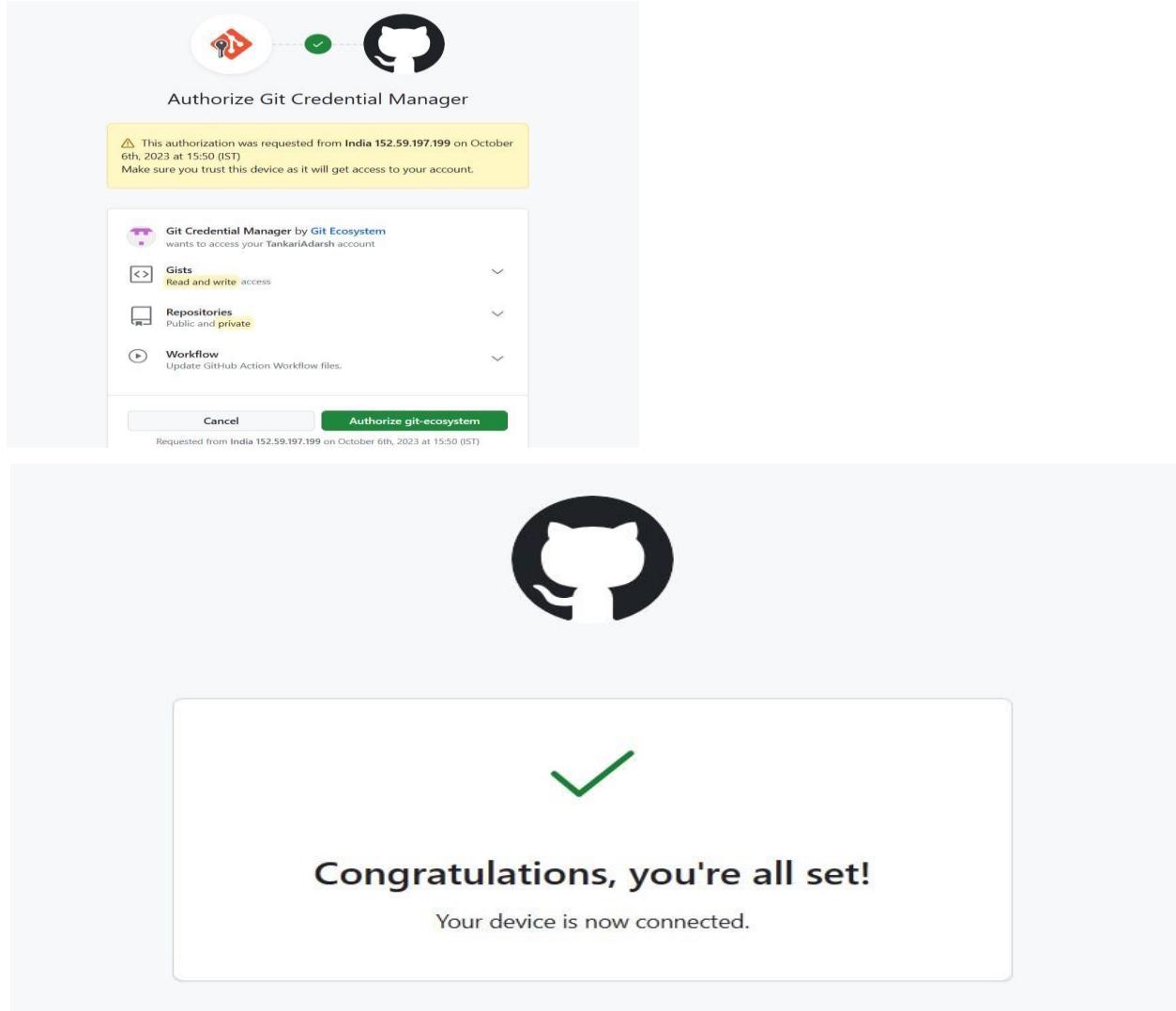
```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git remote add origin https://github.com/TankariAdarsh/seprojectgit.git

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git ls-remote
From https://github.com/TankariAdarsh/seprojectgit.git
```

Step-17:

“git push u –origin main” command helps to push the files to local repository in the git hub For the first time we need to connect to the github with your browser or with the code





```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 519 bytes | 259.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/TankariAdarsh/seprojectgit.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step -18:

“git push” helps to push the committed files to the github account

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git push
Everything up-to-date
```

Step-19:

If any changes made to text file in the git hub then it cannot be pushed into the repository. Here, we need to first pull the text file and then push the file back to the repository.

The screenshot shows a GitHub repository named 'seoprojectgit' owned by 'TankariAdarsh'. The repository has one branch ('main'), one tag ('1 branch'), and zero tags ('0 tags'). The commit history shows two commits from 'TankariAdarsh': 'f2 text file committed unmodified' (dd28681, 23 minutes ago) and 'f1 text file committed' (30 minutes ago). The commit 'f1 text file committed' includes a note 'f1 text file committed modified'. A button at the bottom right says 'Add a README'.

The screenshot shows the file 'f1.txt' in the 'seoprojectgit' repository. The file content is: '1 hi\n2 this is f1 text file\n3 some changes included'. There is a note below the file: '3 lines (3 loc) · 47 Bytes · Code 55% faster with GitHub Copilot'.

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git push
To https://github.com/TankariAdarsh/seoprojectgit.git
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/TankariAdarsh/seoprojectgit.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
User@ADARSH MINGW64 ~/oneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 691 bytes | 46.00 KiB/s, done.
From https://github.com/TankariAdarsh/seoprojectgit
 dd28681..254da59  main      -> origin/main
Updating dd28681..254da59
Fast-forward
  f1.txt | 1 +
  1 file changed, 1 insertion(+)
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git push
Everything up-to-date

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   f2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f3.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Ways to get other repository/project to our repository:

1. “git clone <link>” is used to clone other repository in github
2. The link can be get from any git code

TheAlgorithms / Python

Code Issues 48 Pull requests 166 Discussions Actions Projects Wiki Security Insights

Python Public

master 9 branches 0 tags

Go to file Add file Code

Clone HTTPS SSH GitHub CLI https://github.com/TheAlgorithms/Python.git

About All Algorithms implemented in Python

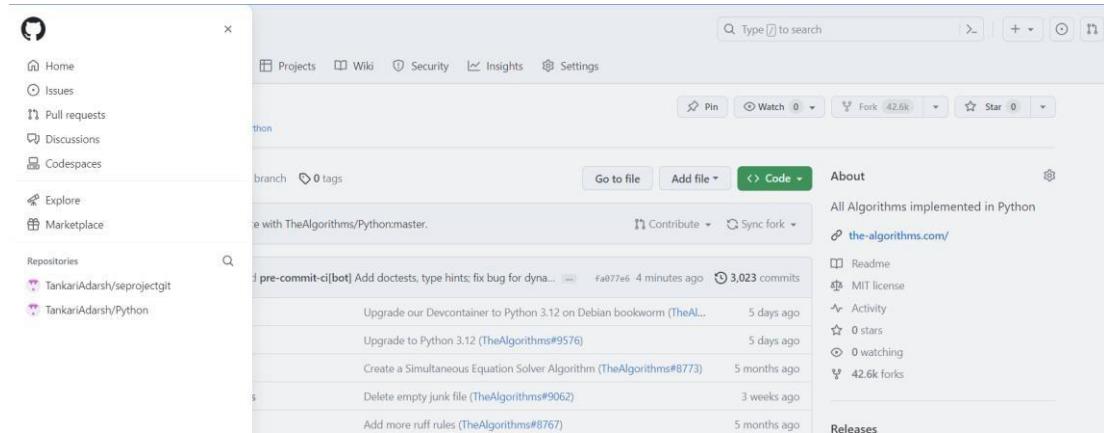
the-algorithms.com/

python education algorithms practice interview sorting-algorithms learn algos algorithm-competitions sorts hacktoberfest algorithms-implemented community-driven searches

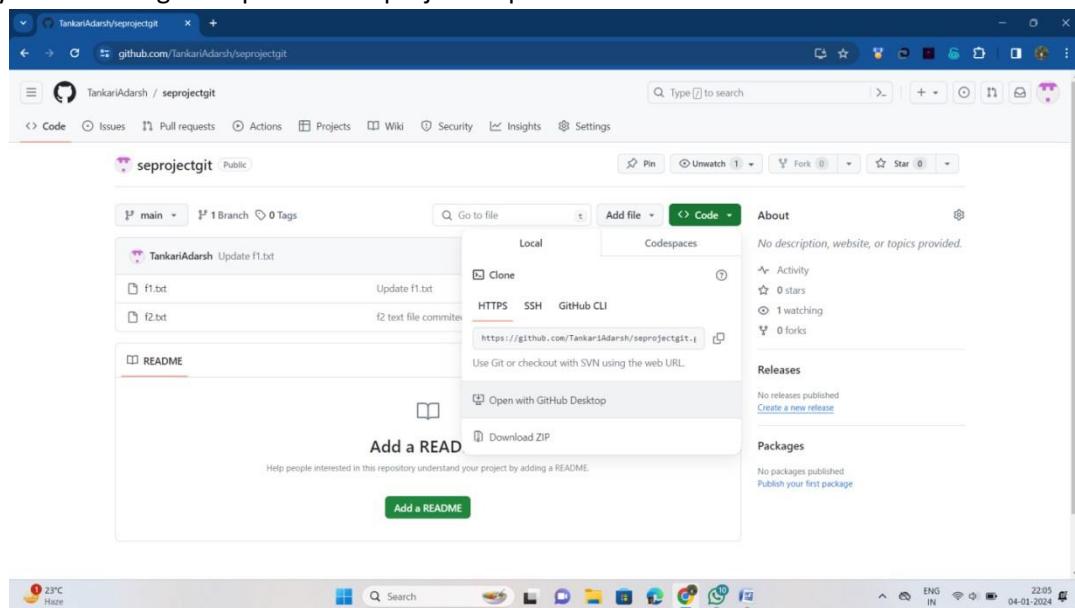
Readme MIT license Code of conduct Activity 171k stars 5.9k watching

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git clone https://github.com/TheAlgorithms/Python.git
Cloning into 'Python'...
remote: Enumerating objects: 17695, done.
remote: Counting objects: 100% (994/994), done.
remote: Compressing objects: 100% (168/168), done.
remote: Total 17695 (delta 886), reused 858 (delta 826), pack-reused 16701
Receiving objects: 100% (17695/17695), 13.70 MiB | 679.00 KiB/s, done.
Resolving deltas: 100% (10848/10848), done.
```

- 2 . by clicking on the “fork” we can get other projects to our repository



3. by downloading the zip file of the project required

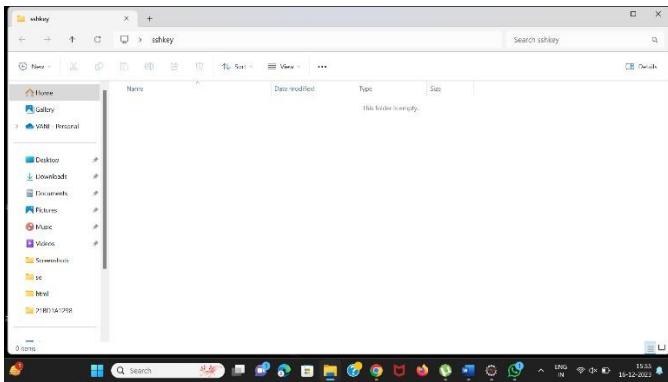


SSH-KEY: Secure Cell Key Generation

An SSH key is an access credential for the SSH (secure shell) network protocol. This authenticated and encrypted secure network protocol is used for remote communication between machines on an unsecured open network. SSH is used for remote file transfer, network management, and remote operating system access.

STEP-1:

On the desktop create a folder for ssh key and add a empty file or a file with some content



STEP-2:

On the git bash create a empty repository using git init.

```
MINGW64:/c/Users/vanik/Desktop/21BD1A1298 se
vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se
$ pwd
/c/Users/vanik/Desktop/21BD1A1298 se
vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se
$ ls
f1.txt f2.txt f3.txt
vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se
$ git init
Initialized empty Git repository in C:/Users/vanik/Desktop/21BD1A1298 se/.git/
vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se (master)
$
```

STEP-3:

Using git add . command add the empty file named to staging area

Using git status check the current working directory or to check whether files are added to staging area or not.

```
vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se (master)
$ git add f1.txt

vanik@konda_vani98 MINGW64 ~/Desktop/21BD1A1298 se (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   f1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f2.txt
    f3.txt
```

STEP-4:

To ensure whether the changes are saved to the repository use the commit command as

Git commit -m “message”

```
vanik@konda_vani98 MINGW64 ~/Desktop/ssh key (master)
$ git commit -m "ssh key"
[master (root-commit) 8ee4f8e] ssh key
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f3
```

STEP-5:

Now execute the following instruction to begin the key creation as:

ssh-keygen

It will be prompted to “Enter a file in which to save the key”; press Enter.

Then it will be prompted for a “Enter passphrase” for 2 times which has be same.

```
anik@konda_vani98 MINGW64 ~/Desktop/ssh key (master)
ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/vanik/.ssh/id_rsa):
/c/Users/vanik/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/vanik/.ssh/id_rsa
Your public key has been saved in /c/Users/vanik/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1+4SSwVjdS24MIDwUcF4jZZHmNh5IqRdTwy1nXcuo vanik@konda_vani98
The key's randomart image is:
---[RSA 3072]----+
 .+=&=.=oo|
 +.%B.=.o.|
 o =.O.o.o o|
 + +.=+ + |
 .S ..*... |
 .. E
 ..
---[SHA256]----+
```

STEP-6:

Now on the created path of ssh key ,go to the specified folder and on the file id_rsa pub :

Using notepad copy the path of ssh key .

Name	Date modified	Type	Size
id_rsa	16-11-2023 11:05	File	3 KB
id_rsa.pub	16-11-2023 11:05	PUB File	1 KB
known_hosts	04-11-2023 21:36	File	1 KB
known_hosts.old	03-11-2023 18:10	OLD File	1 KB

STEP-9:

Using notepad copy the path of ssh key.

```
ssh-rsa
AAAAAB3NzaC1yc2EAAAQABAAQgQCpDAuz
jDWiBM3CELSG+OaSHC27dmCKmxVGYVC88If/
K0MFcy7k+maYv7XHYAAXVg+
2GN18RQCj004wF1pdT1qt3BjDVobB2TIpMvH
iySt7W53aCvzk3A7bV0widuxwlqPrp9nYKtdd
qABsJFvyW2xjkude/0dELYD1VAD1q7DKkqfx
9c1jRzwDmhWkd0pefpqd0c116mKgVDv6diD9
+r061NcBYxTncI/6dgKz1szGujnyEReNLXLm
Gt5wV+G11VxXqdYEFLVFvBuZLq3SPRHxwdh7
Ow+HftNbuh30FXgSHU4tkxEwxUy+wA9k1k
wUjhssFampfGI0bshEek04o6YP8PhkwDdk43
```

STEP-10:

In the remote repository of git hub :

Go to the Settings->SSH and GPG Keys ;click on new ssh key and paste the above path

It will successfully create a key.

SSH keys

New SSH key

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

Authentication Keys

 ssh key	SHA256:1+4SSwVjdBS24MIDwUcF4jZZHmNh5IqRdTwy1nXxcuo	
SSH	Added on Nov 4, 2023	
	Last used within the last week — Read/write	

STEP-11:

Using git remote add origin “path” command the local repository will be connected to the remote repository.

```
vanik@konda_vani98 MINGW64 ~/Desktop/ssh key (master)
$ git remote add origin git@github.com:vani-id98/ssh-keygen-create.git

vanik@konda_vani98 MINGW64 ~/Desktop/ssh key (master)
$ git ls-remote
Enter passphrase for key '/c/Users/vanik/.ssh/id_rsa':
From git@github.com:vani-id98/ssh-keygen-create.git
```

STEP-12:

Change the current working branch from master->main using git branch -M main command .

```
vanik@konda_vani98 MINGW64 ~/Desktop/ssh key (master)
$ git branch -M main
```

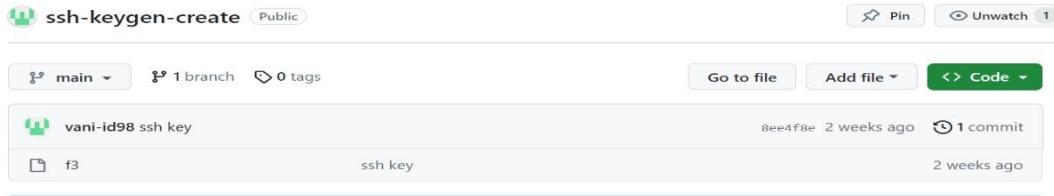
STEP-13:

Push the file to the remote repository using git push -u origin main and enter the same phrase for ssh-key generation.

```
vanik@konda_vani98 MINGW64 ~/Desktop/ssh key (main)
$ git push -u origin main
Enter passphrase for key '/c/Users/vanik/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 202 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:vani-id98/ssh-keygen-create.git
 * [new branch]      main -> main
Branch 'main' set up to track 'origin/main'.
```

STEP-14:

On the respective remote repository the file will be pushed.



BRANCHING CONCEPT

1.CREATION OF BRANCH

Cmd:

git branch branch name

Ex: git branch test

Created a branch with name test

Git checkout branchname

Git chechout branch name will change the main branch to sub branch called test

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git branch test

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year 1 sem/se lab/se project git (main)
$ git checkout test
Switched to branch 'test'
M      f2.txt
```

“ls” command shows the list of files in the folder for the mention folderpath

“Vi <test file>” command will create a new file with the file name and open a vi editor to write content to the file

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ ls
Python/ f1.txt f2.txt f3.txt

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ vi t1

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ ls
Python/ f1.txt f2.txt f3.txt t1
```

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git checkout main
Switched to branch 'main'
M      f2.txt
Your branch is up to date with 'origin/main'.
```

“git merge <branch>” command is used in Git to combine changes from one branch into another.

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (main)
$ ls
Python/ f1.txt f2.txt f3.txt t1

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (main)
$ git merge test
Already up to date.
```

add the file, commit the file by using **git add**, **git commit -m “message”** and **git log**

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git add t1

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git commit -m "t1 committed"
[test c86a8ea] t1 committed
 4 files changed, 8 insertions(+)
  create mode 160000 Python
  create mode 100644 f3.txt
  create mode 100644 t1

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git log --oneline
c86a8ea (HEAD -> test) t1 committed
254da59 (origin/main, main) Update f1.txt
dd28681 f2 text file committed unmodified
1fa1315 f1 text file committed
--oneline. "
```

git ls - remote command gives the link to the repository where the committed files are stored in the test branch

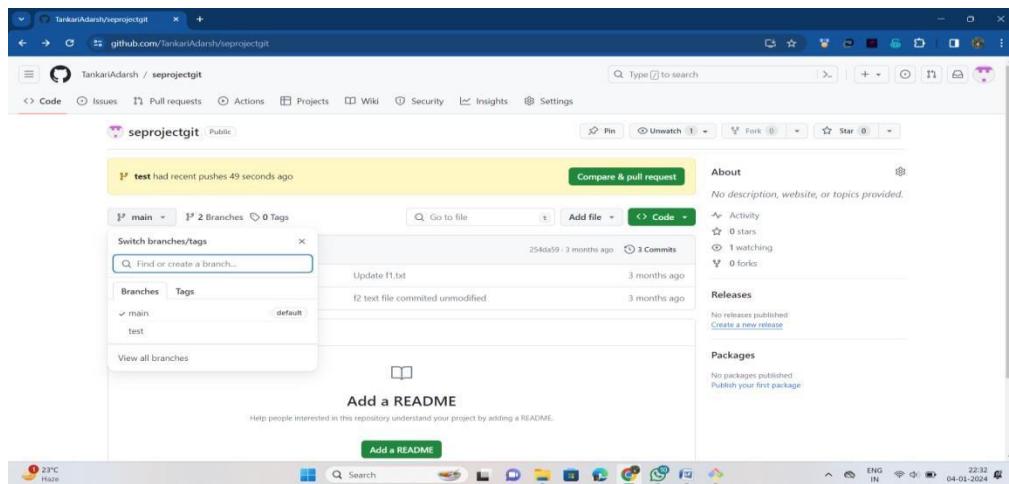
```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git ls-remote
From https://github.com/TankariAdarsh/seprojectgit.
  git
  254da59e8e9612916fb097163591b3f8b968337      HEA
  D
  254da59e8e9612916fb097163591b3f8b968337      ref
  s/heads/main

User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (test)
$ git checkout main
warning: unable to rmdir 'Python': Directory not empty
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

“git push -u origin test” command helps to push the files to local repository in the git hub.

```
User@ADARSH MINGW64 ~/OneDrive/Desktop/KMIT/3 year
1 sem/se lab/se project git (main)
$ git push -u origin test
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 429 bytes | 429.00 KiB
/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused
0
remote:
remote: Create a pull request for 'test' on GitHub
by visiting:
remote:     https://github.com/TankariAdarsh/seprojectgit/pull/new/test
remote:
To https://github.com/TankariAdarsh/seprojectgit.git
 * [new branch]      test -> test
branch 'test' set up to track 'origin/test'.
```

The branches can be seen in the github



Experiment 5:

Working with Jenkins

- a) JDK 11 installation and setting environmental path
- b) Jenkins installation
- c) Tomcat 9 installation and setting environmental path
- d) Eclipse installation and Updating
- e) Configuring Tomcat 9 with Eclipse
- f) Maven installation and setting environmental path
- g) Jenkins Global configurations and installing additional 5 pipeline plugins

ECLIPSE INSTALLATION:

STEP-1: Download eclipse download from browser.

About 58,60,000 results (0.30 seconds)

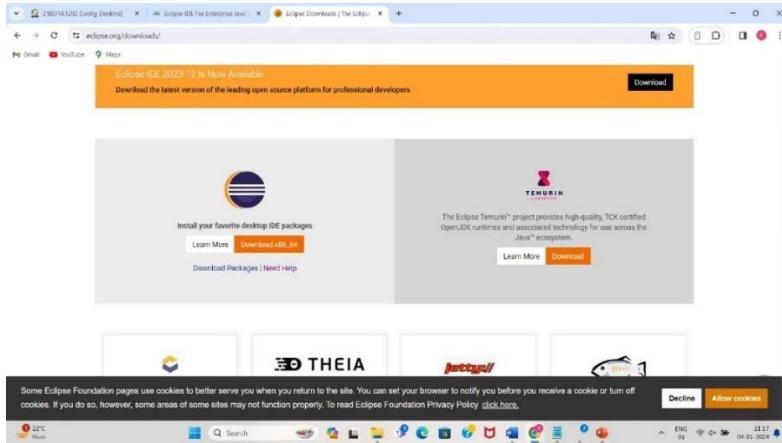
Eclipse
https://www.eclipse.org/packages/releases/eclipse-...
Eclipse IDE for Enterprise Java and Web Developers

17 Mar 2021 – Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, ...

eclipseide.org
Eclipse IDE | The Eclipse Foundation

Better Than Ever. The Eclipse IDE delivers what you need to rapidly innovate. Improved Java Development Tooling, Implementation for Record Patterns, Pattern Eclipse IDE Working Group · New & Noteworthy · Planning Council · Getting Started

STEP-2:Click on the below link.



STEP-3:Select Eclipse IDE for Enterprise java and web Developers

- Eclipse IDE for Java Developers**
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.
- Eclipse IDE for Enterprise Java and Web Developers**
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.
- Eclipse IDE for C/C++ Developers**
An IDE for C/C++ developers, including managed cross-build plug-ins (IAR and RSOC-V) and debugging plug-ins (DEGGER, jLink, OpenOCD), printf, and QEMU, plus a number of compilers to create ready-to-run binary projects.
- Eclipse IDE for Embedded C/C++ Developers**
An IDE for Embedded C/C++ developers, including managed cross-build plug-ins (IAR and RSOC-V) and debugging plug-ins (DEGGER, jLink, OpenOCD), printf, and QEMU, plus a number of compilers to create ready-to-run binary projects.
- Eclipse IDE for PHP Developers**
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, TypeScript, HTML, CSS and JSON, JavaDoc for comments, and Eclipse IDE PHP Performance Tools.
- Eclipse IDE for Eclipse Committers**
Packages required for development of Eclipse itself at Eclipse.org, based on the Eclipse Platform adding PDE, GH, Marketplace Client, source code and CDT, Java, JavaServer Faces, JavaEE, JavaFX, RCP, and RAP Development Tools.
- Eclipse IDE for Java and DSL Developers**
The essential tools for Java and DSL developers, including a Java IDE, a Git client, XML Editor, and Maven integrations.
- Eclipse IDE for RCP and RAP Developers**

STEP-4:Click on install.

Eclipse IDE for Enterprise Java and Web Developers
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.

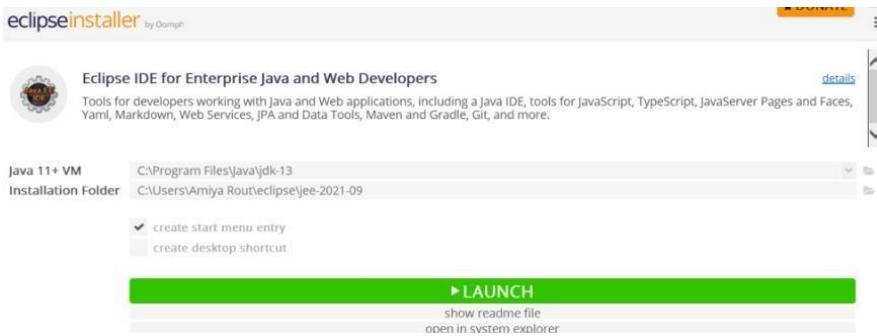
Java 11+ VM: C:\Program Files\Java\jdk-13
Installation Folder: C:\Users\Amiya Rout\eclipse\jee-2021-09

create start menu entry
 create desktop shortcut

INSTALL

STEP-5:

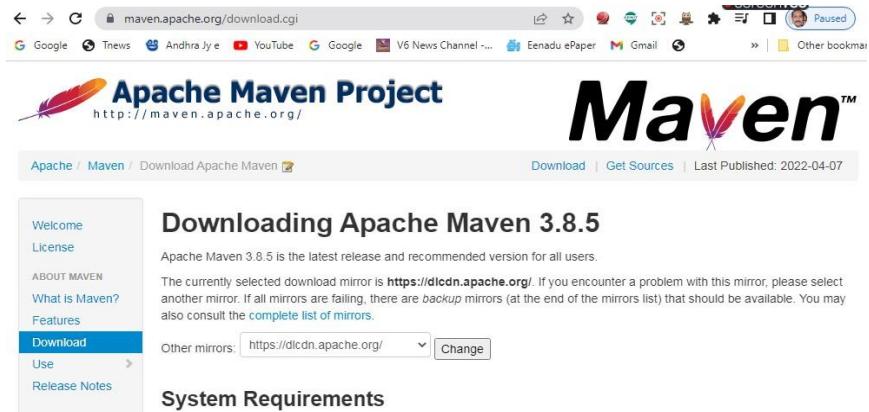
Then with launch the install gets completed.



MAVEN INSTALLATION

Step 1: check whether maven is present in your system or not

Step 2: go to Google type download maven and click on download Apache maven



Welcome
License

ABOUT MAVEN
What is Maven?
Features
Download
Use
Release Notes

Download Apache Maven

Downloading Apache Maven 3.8.5

Apache Maven 3.8.5 is the latest release and recommended version for all users.

The currently selected download mirror is <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are backup mirrors (at the end of the mirrors list) that should be available. You may also consult the [complete list of mirrors](#).

Other mirrors: [Change](#)

System Requirements

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

Link	Checksum	Signature
Binary tar.gz archive	apache-maven-3.3.9-bin.tar.gz	apache-maven-3.3.9-bin.tar.gz.md5
Binary zip archive	apache-maven-3.3.9-bin.zip	apache-maven-3.3.9-bin.zip.md5
Source tar.gz archive	apache-maven-3.3.9-src.tar.gz	apache-maven-3.3.9-src.tar.gz.md5
Source zip archive	apache-maven-3.3.9-src.zip	apache-maven-3.3.9-src.zip.md5

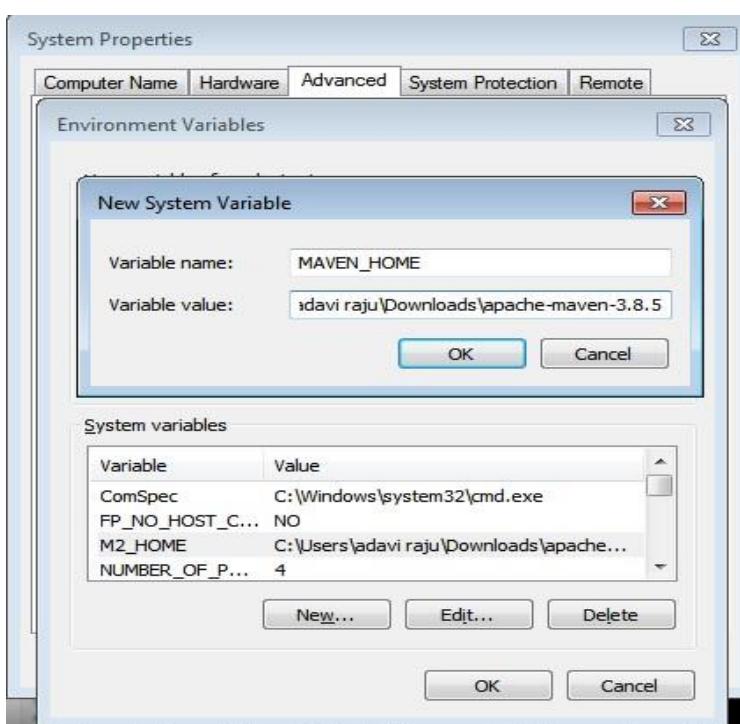
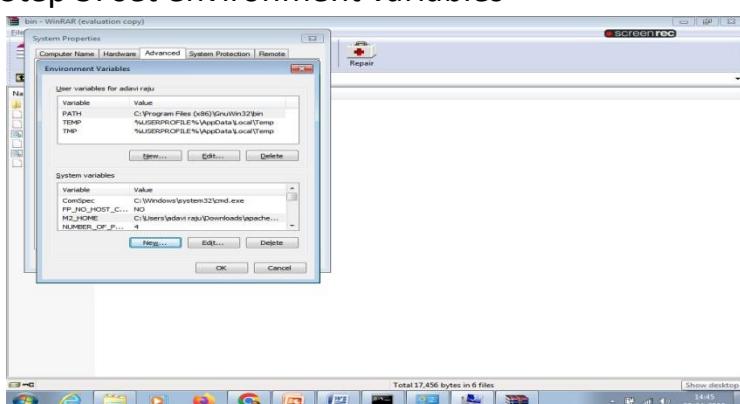
- [Release Notes](#)
- [Reference Documentation](#)
- [Apache Maven Website As Documentation Archive](#)
- All sources (plugins, shared libraries,...) available at <https://www.apache.org/dist/maven/>
- Distributed under the Apache License, version 2.0

Step 3: click on apache-maven-3.3.9-bin zip

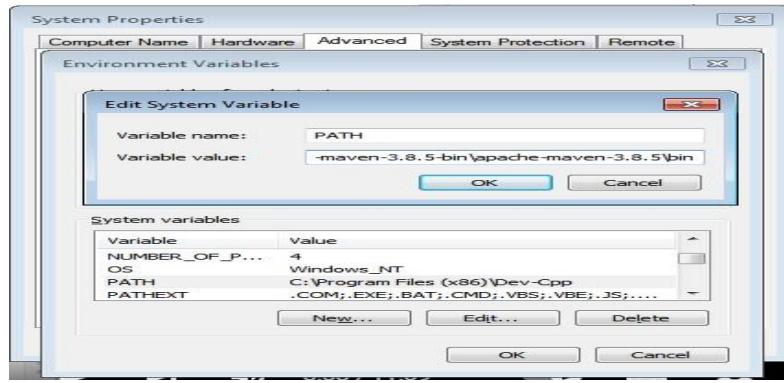
Step 4: extract zip file



Step 5: set environment variables



Step-6: Edit path



STEP-7:

Check the version of maven by using command maven –version in command prompt check whether the installation has done properly .

```
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Madhu>mvn -version
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: E:\apache-maven-3.8.5
Java version: 11.0.13, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11.0.13
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Madhu>
```

Experiment 6:

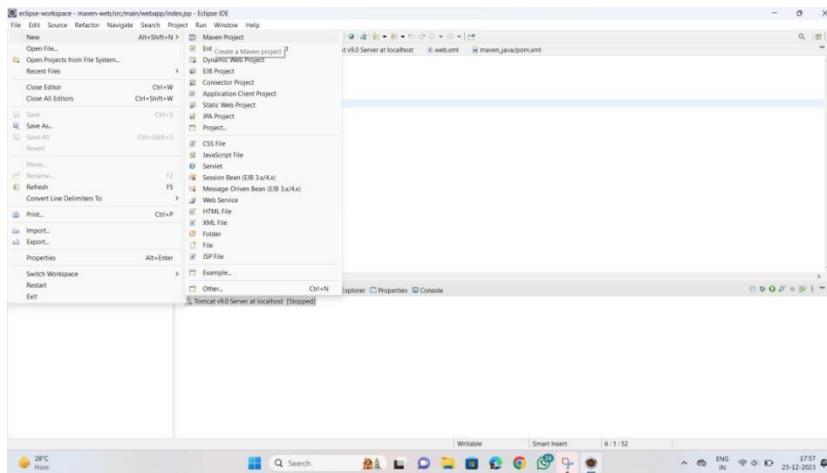
Working with Maven in Eclipse

- a) Create a Maven java project and run it in Eclipse environment
- b) Create a Maven Web Project and run it in Eclipse environment (by adding dependencies)
- c) Push the above two projects into GitHub.

Eclipse

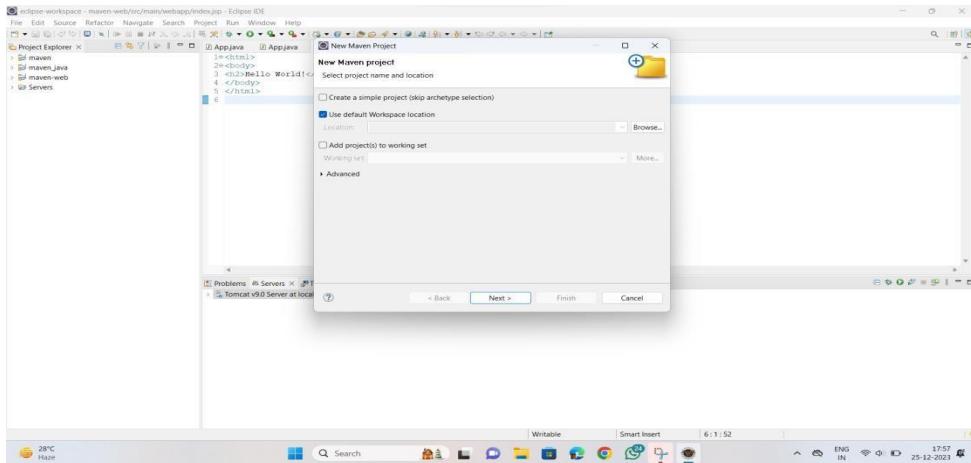
Step 1: Create A New Maven Project

First, we will create a new maven project. For this, we will open our Eclipse IDE then Go to **File > New > Maven Project**.



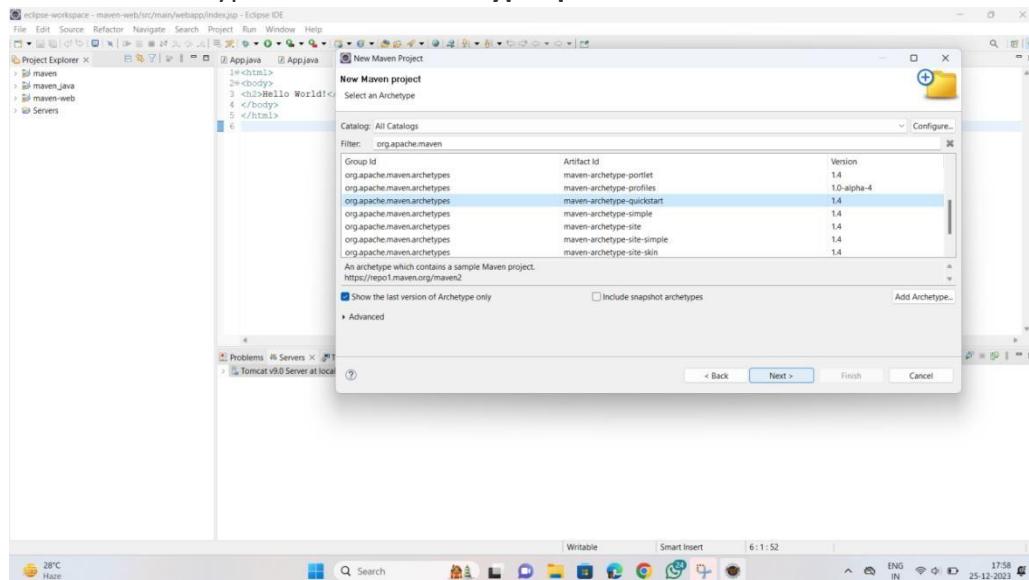
Once we click on Maven Project we will get a prompt for setting up our Workspace location.

After setting Workspace location according to our choice. We will click on Next to go to further project setup steps.



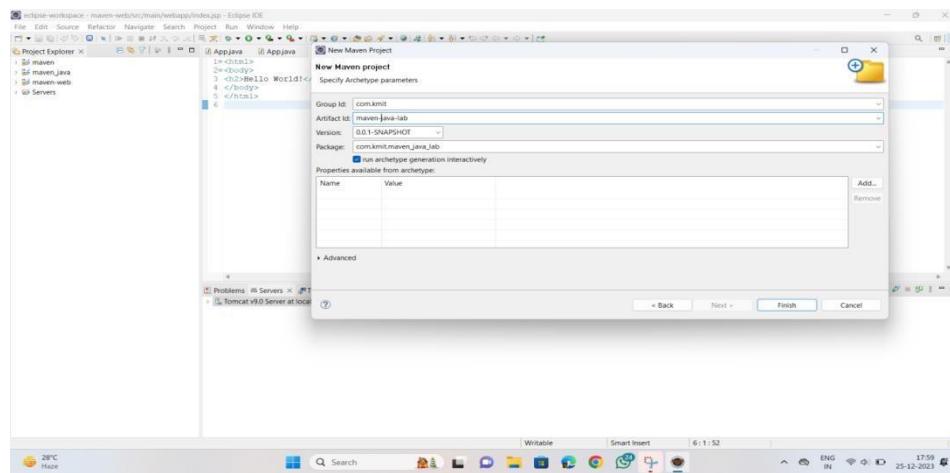
Step 2: Setting Up Archetype

Select the archetype as **maven-archetype-quickstart**

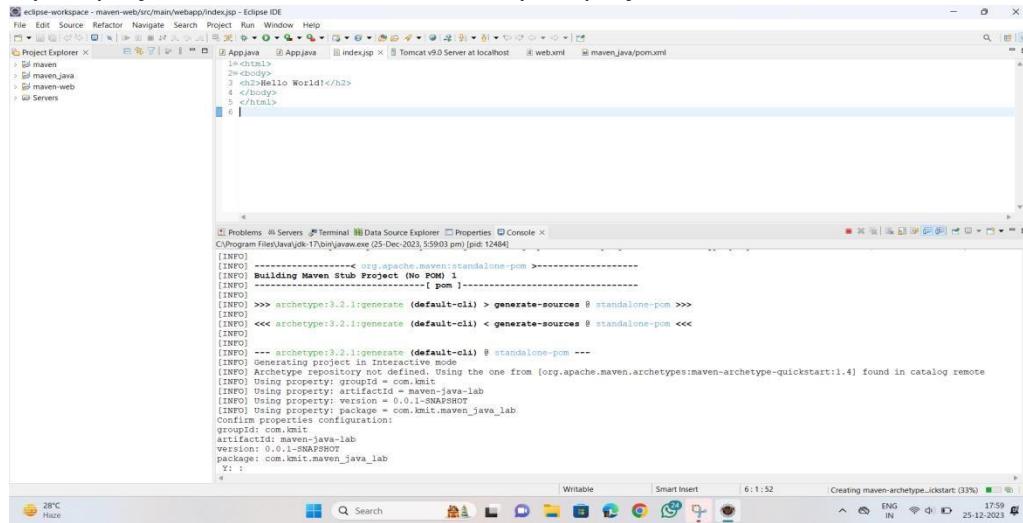


After choosing the archetype we will click on Next and it will ask for the details like Group Id, Artifact Id, and Package information.

- Group Id – It's a unique identifier to uniquely identify the project in a maven repository. ☐
- Artifact Id – It's basically the project name.



- After adding all these details we need to click on the Finish button. Maven will start creating your project and after a few seconds, your project will be created.



- By clicking on console as "y" you can conform the creation of project

Eclipse IDE - Maven Project

```

eclipse-workspace - maven-webapp\main\webapp\index.jsp - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer [App Java] index.jsp Tomcat v9.0 Server at localhost web.xml maven.java/pom.xml
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h2>Hello World!</h2>
</body>
</html>

```

Terminal

```

[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] Parameter: groupId, Value: com.kmit
[INFO] Parameter: artifactId, Value: maven-java-lab
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: com.kmit.maven.java.lab
[INFO] Parameter: packageInPathFormat, Value: com/kmit/maven/java_lab
[INFO] Parameter: package, Value: com.kmit.maven.java.lab
[INFO] Parameter: groupId, Value: com.kmit
[INFO] Parameter: artifactId, Value: maven-java-lab
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\User\eclipse-workspace\maven-java-lab
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 17.720 s
[INFO] Finished at: 2023-12-25T17:59:24+05:30
[INFO] -----

```

Eclipse IDE - Maven Java Project

```

eclipse-workspace - maven-java-lab\src\main\java\com\kmit\maven\java\lab\App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer [App Java] App Java
1 package com.kmit.maven.java.lab;
2
3 /**
4  * Hello world!
5 */
6
7 public class App {
8
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12        System.out.println("Lab documentation");
13    }
14}
15

```

Terminal

```

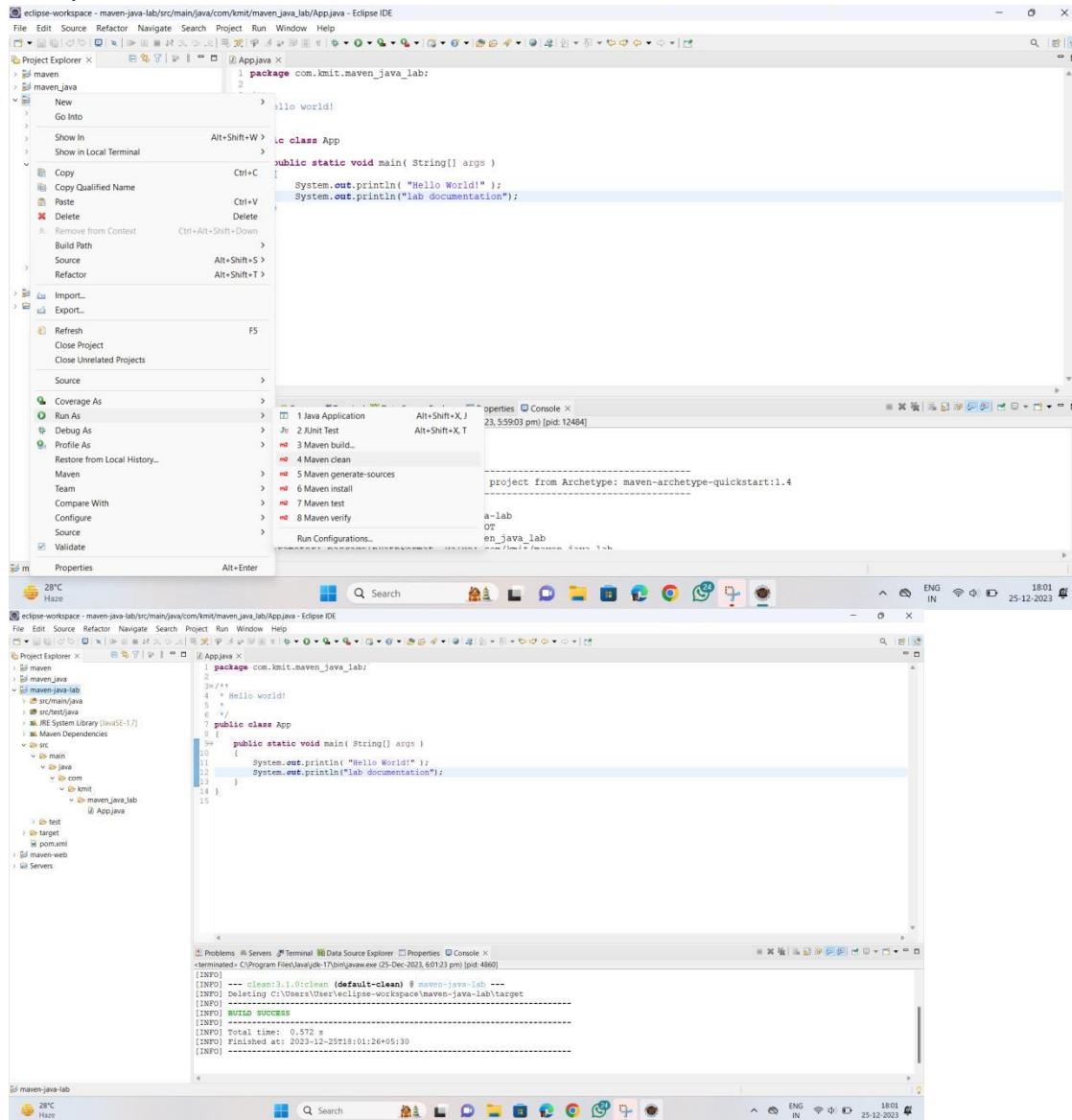
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] Parameter: groupId, Value: com.kmit
[INFO] Parameter: artifactId, Value: maven-java-lab
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: com.kmit.maven.java.lab
[INFO] Parameter: package, Value: com.kmit.maven.java.lab
[INFO] Parameter: groupId, Value: com.kmit
[INFO] Parameter: artifactId, Value: maven-java-lab
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\User\eclipse-workspace\maven-java-lab
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 17.720 s
[INFO] Finished at: 2023-12-25T17:59:24+05:30
[INFO] -----

```

To run the maven project right click on folder and click on run as where go with the process 1)
Maven clean

- 2) Maven install
- 3) Maven test
- 4) Maven build

Finally click on run as->run on server



The screenshot shows the Eclipse IDE interface with three windows open:

- Top Window:** Project Explorer view showing a Java project named "Appjava". The code editor displays a simple "Hello World" application.
- Middle Window:** Run Configurations view showing a list of Maven tasks: 1 Java Application, 2 JUnit Test, 3 Maven build..., 4 Maven clean, 5 Maven generate-sources, 6 Maven install, 7 Maven test, and 8 Maven verify. The "3 Maven build..." task is selected.
- Bottom Window:** Console view showing the output of the Maven build command. The log includes:


```
<terminated> C:\Program Files\Java\jdk-17\bin\javaw.exe (25-Dec-2023, 00123 pm) [pid: 4860]
[INFO] --- cleanci:1.0:clean (default-clean) @ maven-java-lab ---
[INFO] Deleting C:\Users\User\eclipse-workspace\maven-java-lab\target
[INFO] -----
[INFO] --- maven-clean-plugin:3.1.0:clean ---
[INFO] Total time: 0.572 s
[INFO] Finished at: 2023-12-25T18:01:26+05:30
[INFO] -----
```

Screenshot 1: Eclipse IDE - Maven Java Lab Project

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

- Project Explorer:** Shows the project structure with a main Java source folder containing an App.java file.
- Code Editor:** Displays the following code for App.java:


```
package com.kmit.maven_java_lab;
public class App {
    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("lab documentation");
    }
}
```
- Run Configuration:** A context menu is open over the code editor, showing options like "Run As" (Java Application, JUnit Test), "Build Path", "Source", "Refactor", and "Run Configurations...".
- Terminal:** Shows the output of the Maven build command:


```
[INFO] T: S: T: S
[INFO] --- [jar:3.6.2.jar] @ maven-java-lab ---
[INFO] Building jar: C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar
[INFO] 
[INFO] --- install:2.5:install (default-install) @ maven-java-lab ---
[INFO] Installing C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar to C:\Users\User\.m2\repository\com\kmit\maven-java-lab\0.0.1-SNAPSHOT\maven-jar
[INFO] 
[INFO] Total time: 6.125 s
[INFO] Finished at: 2023-12-25T18:01:52+0530
[INFO] 
-----
```

Screenshot 2: Eclipse IDE - Maven Java Lab Project

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

- Project Explorer:** Shows the project structure with a main Java source folder containing an App.java file.
- Code Editor:** Displays the following code for App.java:


```
package com.kmit.maven_java_lab;
public class App {
    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("lab documentation");
    }
}
```
- Terminal:** Shows the output of the Maven build command:


```
[INFO] T: S: T: S
[INFO] --- [jar:3.6.2.jar] @ maven-java-lab ---
[INFO] Building jar: C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar
[INFO] 
[INFO] --- install:2.5:install (default-install) @ maven-java-lab ---
[INFO] Installing C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar to C:\Users\User\.m2\repository\com\kmit\maven-java-lab\0.0.1-SNAPSHOT\maven-jar
[INFO] 
[INFO] Total time: 6.125 s
[INFO] Finished at: 2023-12-25T18:01:52+0530
[INFO] 
-----
```

Screenshot 3: Eclipse IDE - Maven Java Lab Project

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

- Project Explorer:** Shows the project structure with a main Java source folder containing an App.java file.
- Code Editor:** Displays the following code for App.java:


```
package com.kmit.maven_java_lab;
public class App {
    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("lab documentation");
    }
}
```
- Terminal:** Shows the output of the Maven build command:


```
[INFO] T: S: T: S
[INFO] --- [jar:3.6.2.jar] @ maven-java-lab ---
[INFO] Building jar: C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar
[INFO] 
[INFO] --- install:2.5:install (default-install) @ maven-java-lab ---
[INFO] Installing C:\Users\user\workspace\maven-java-lab\target\maven-java-lab-0.0.1-SNAPSHOT.jar to C:\Users\User\.m2\repository\com\kmit\maven-java-lab\0.0.1-SNAPSHOT\maven-jar
[INFO] 
[INFO] Total time: 6.125 s
[INFO] Finished at: 2023-12-25T18:01:52+0530
[INFO] 
-----
```

```

eclipse-workspace - maven-java-lab [src/main/java/com/kmit/maven_java_lab/App.java] - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  -> maven
  -> maven.java
  -> maven.java-lab
    -> src/main/java
      -> com
        -> kmit
          -> maven.java_lab
            -> App.java
  -> test
  -> target
  -> pom.xml
  -> maven-web
  -> Servers

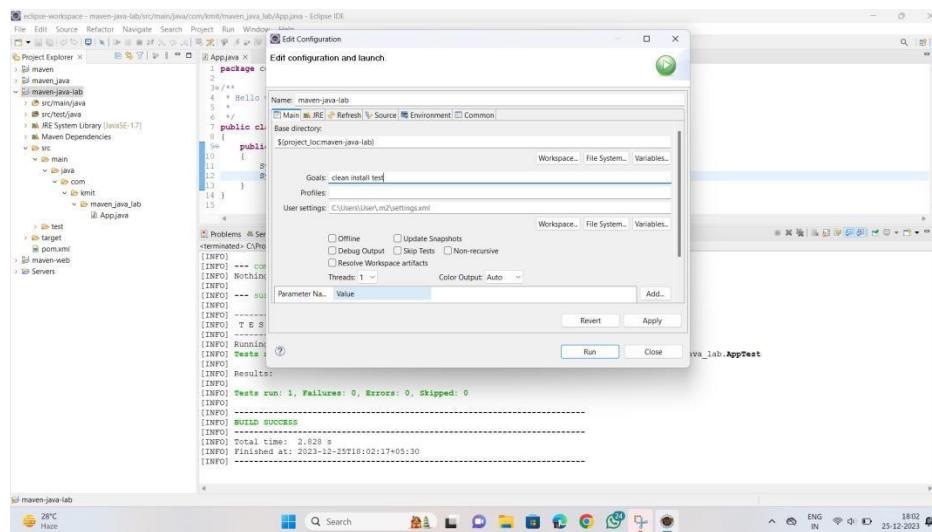
eclipse-workspace - maven.java-lab [src/main/java/com/kmit/maven.java_lab/App.java] - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  -> maven
  -> maven.java
  -> maven.java-lab
    -> src/main/java
      -> com
        -> kmit
          -> maven.java_lab
            -> App.java
  -> test
  -> target
  -> pom.xml
  -> maven-web
  -> Servers

maven.java-lab
28°C Haze
Search
ENG IN 18:02 25-12-2023
eclipse-workspace - maven.java-lab [src/main/java/com/kmit/maven.java_lab/App.java] - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  -> maven
  -> maven.java
  -> maven.java-lab
    -> src/main/java
      -> com
        -> kmit
          -> maven.java_lab
            -> App.java
  -> test
  -> target
  -> pom.xml
  -> maven-web
  -> Servers

maven.java-lab Properties Alt+Enter
28°C Haze
Search
ENG IN 18:02 25-12-2023

```

By clicking on maven build a edit configuration and lunch window appers in goals type clean install test and then click on apply and run



Finally click on java application to run the maven project the output will appear on the console

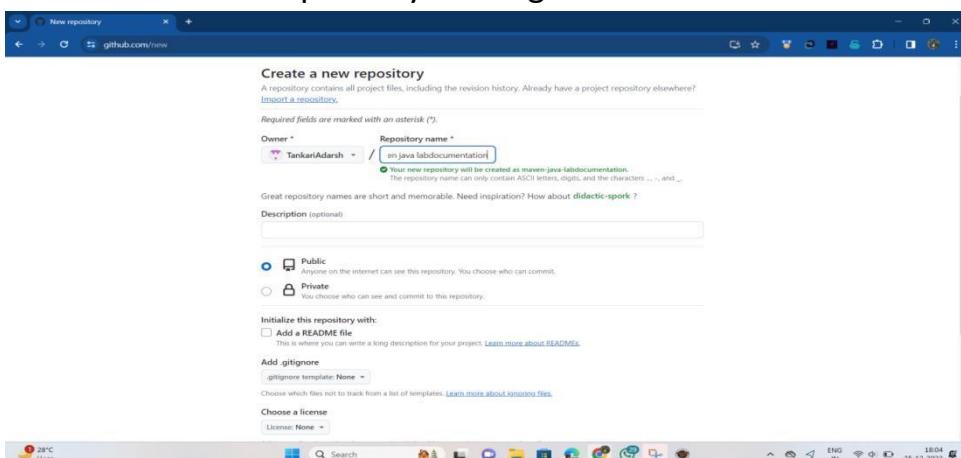
```

eclipse-workspace - Maven Java - [File] [Edit] [Source] [Refactor] [Navigate] [Search] [Project] [Run] [Window] [Help]
Project Explorer X AppJava X
File Edit Source Refactor Navigate Search Project Run Window Help
1 package com.kmit.maven_java_lab;
2
3 /**
4 * Hello world!
5 */
6
7 public class App {
8     public static void main( String[] args ) {
9         System.out.println( "Hello World!" );
10        System.out.println( "lab documentation" );
11    }
12 }
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590

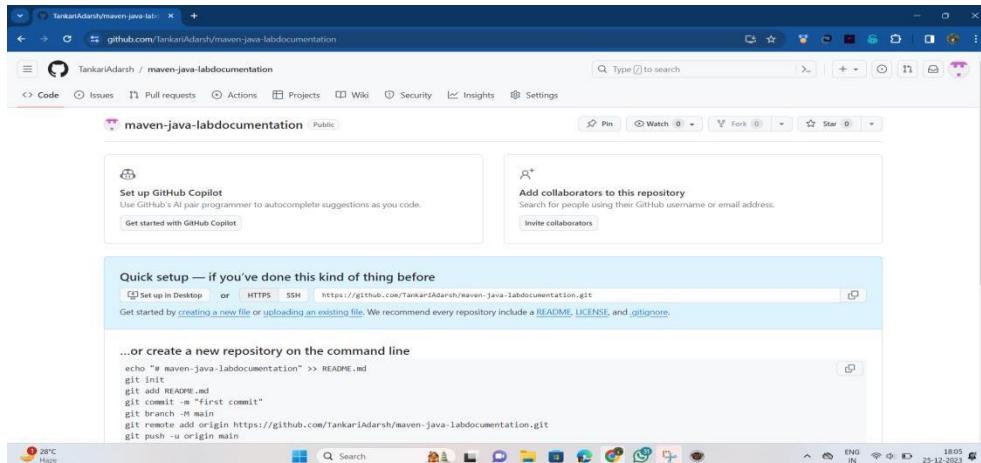
```

PUSHING MAVEN JAVA PROJECT INTO GITHUB REPOSITORY:

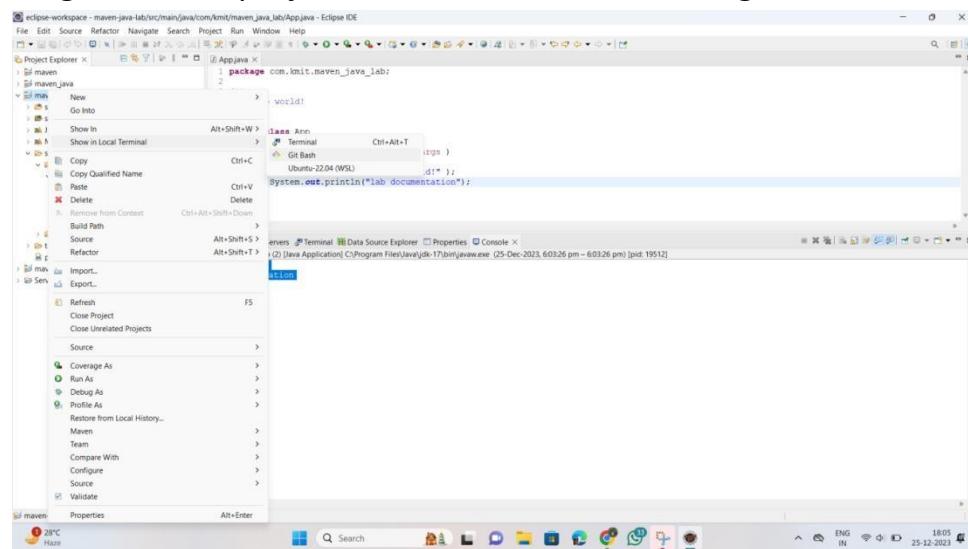
First create a new repository in the github



Copy the http link to push the the project into the github



->Right click on project->show in local terminal->gitbash



- 1)initialize the empty git repository using “git init” command
- 2)add all files to the empty repository
- 3)commit all the files using “git commit” command
- 4)Then after paste the link that has been copied in the github to the command “git remoteadd origin link”

```

Connected - Encoding: UTF-8
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-lab (master)
$ git push
From https://github.com/TankariAdarsh/maven-java-labdocumentation.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-lab (master)
$
```

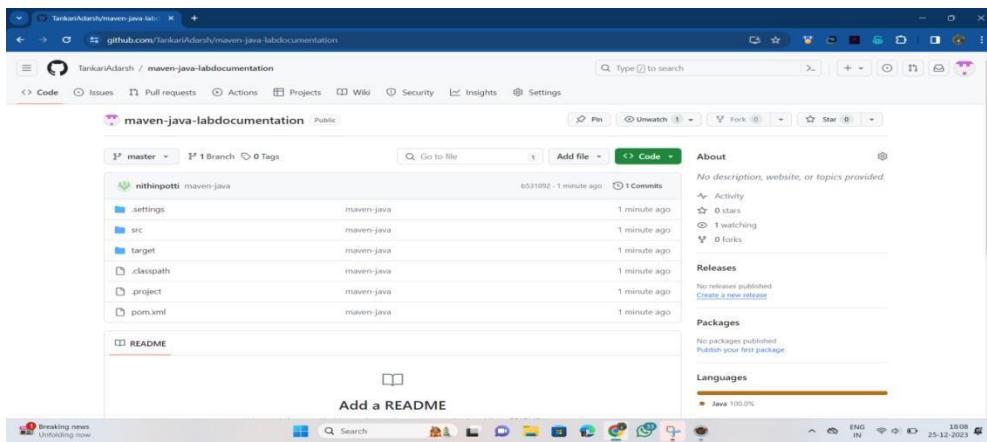
Finally push all the files to the github account using the command “git push” Check the github wheather the files are pushed into it

```

User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-lab (master)
$ git ls-remote
From https://github.com/TankariAdarsh/maven-java-labdocumentation.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

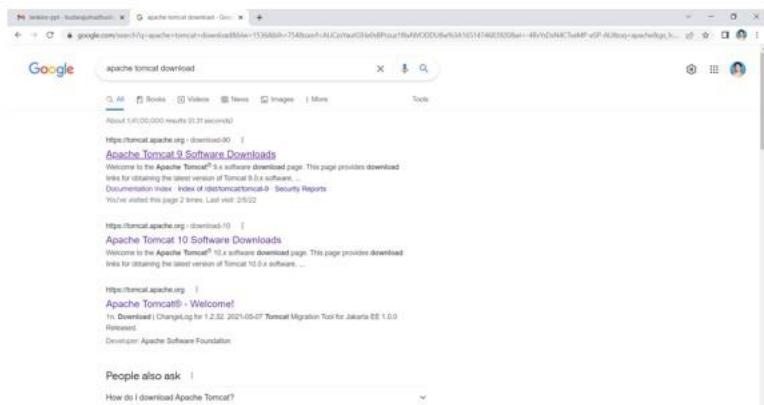
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-lab (master)
$ git push -u origin master
Enumerating objects: 49, done.
Counting objects: 100% (49/49), done.
Delta compression using up to 8 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (49/49), 8.45 KiB | 720.00 KiB/s, done.
Total 49 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/TankariAdarsh/maven-java-labdocumentation.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-lab (master)
$
```

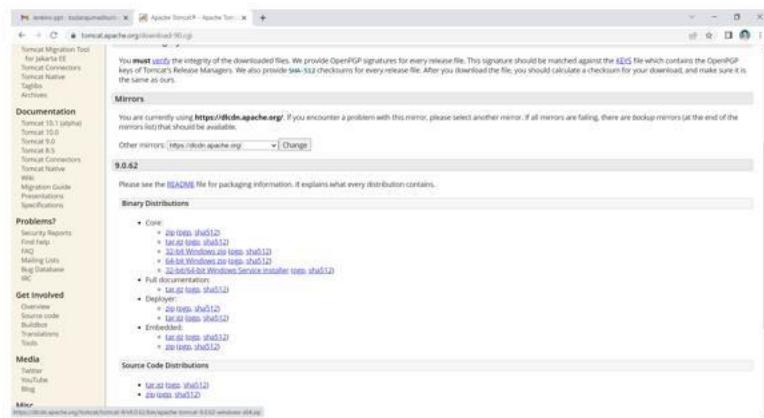


TOMCAT INSTALLATION:

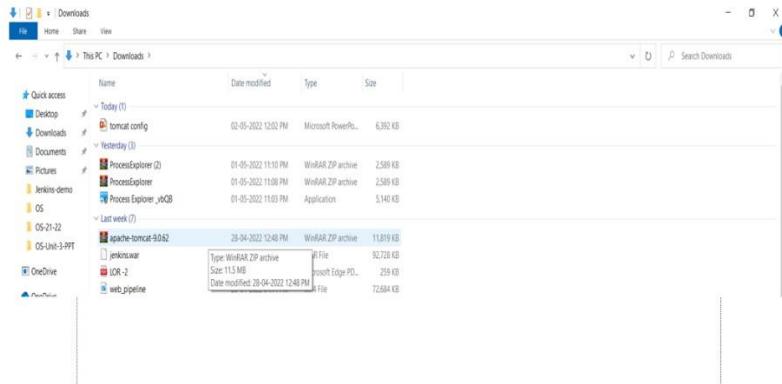
STEP-1:Download the tomcat from the first link as shown



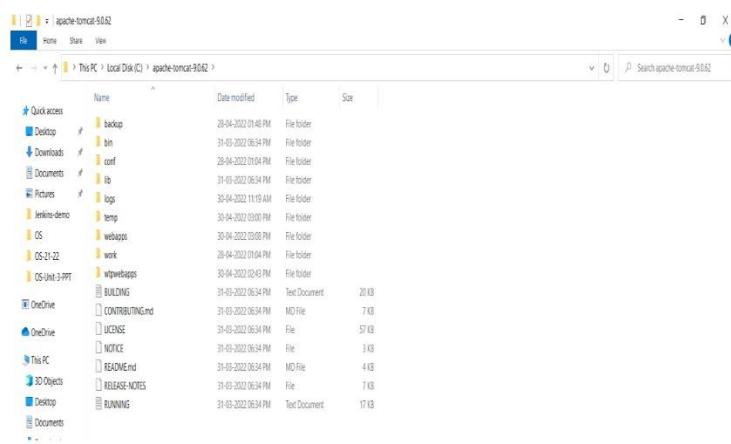
STEP-2:Click on the following link 64-bit windows zip to download the zip file



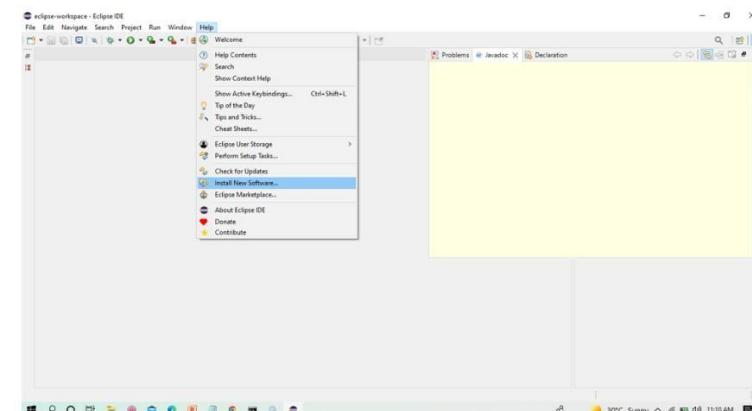
STEP-3:we can view the downloaded file from downloads folder as shown



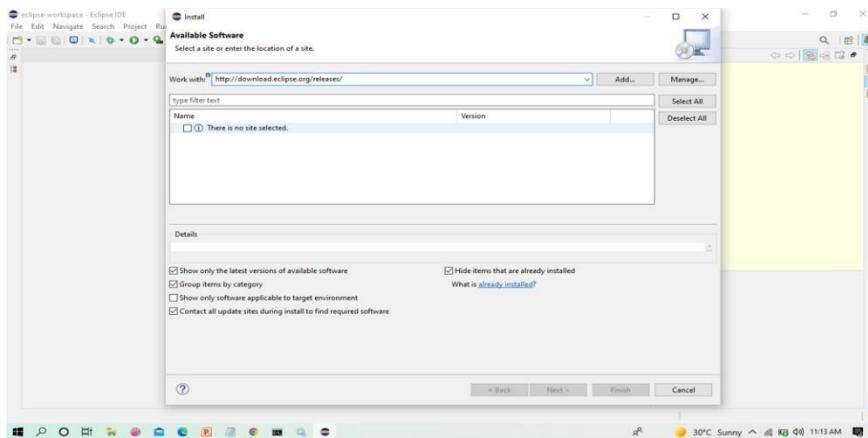
STEP-4:Unzip the downloaded file into path shown



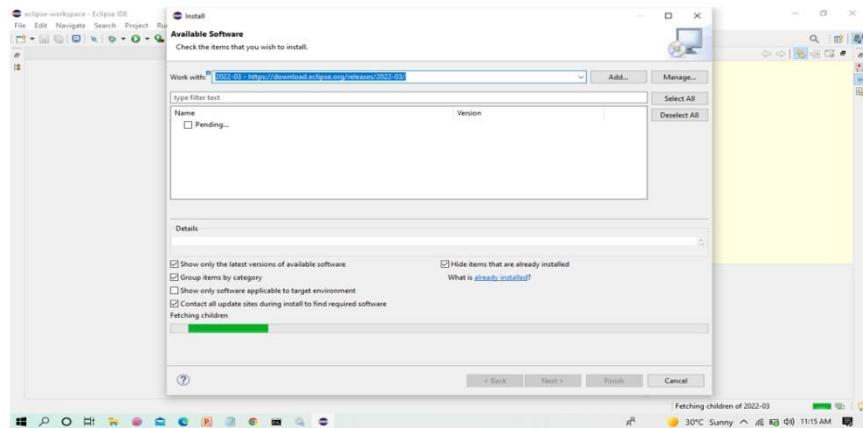
STEP-5:To install tomcat server into eclipse.



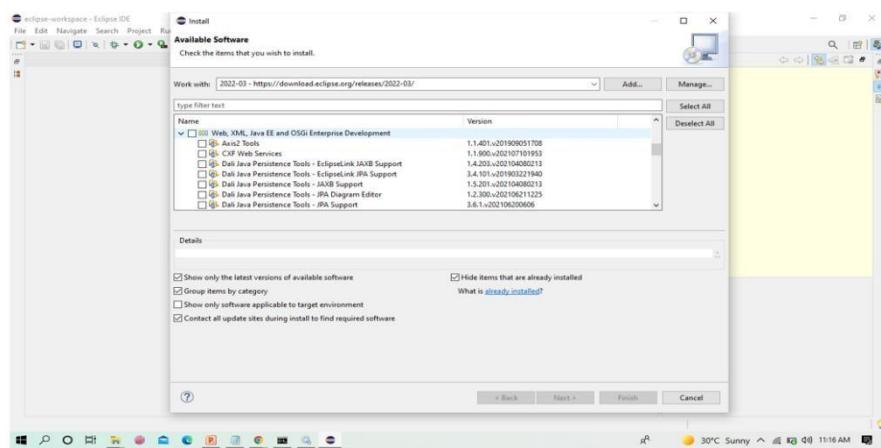
STEP-6:before installing server into eclipse,we need to add java ee to the eclipse ide



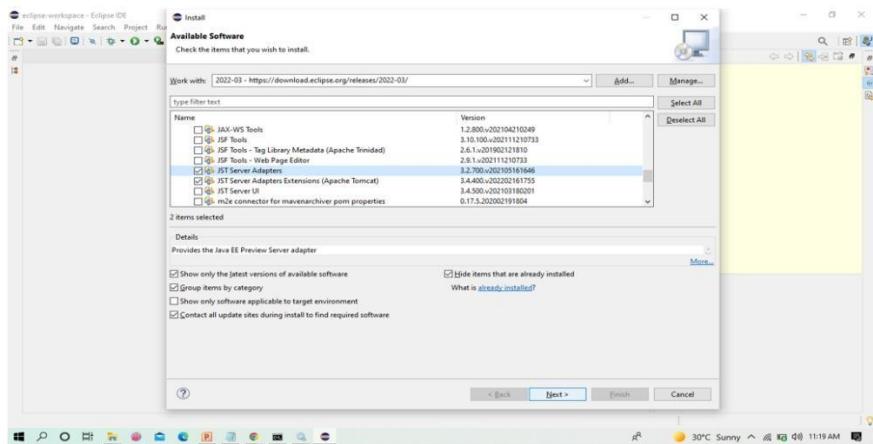
STEP-7: Click on the drop down menu in work with and select the following from the list



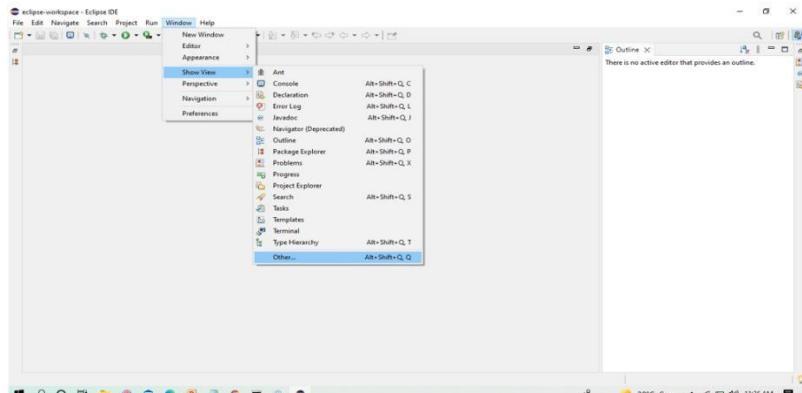
STEP-8: Click the web,xml,java ee box to install the latest software into eclipse in order to execute the dynamic web projects,maven web projects etc.



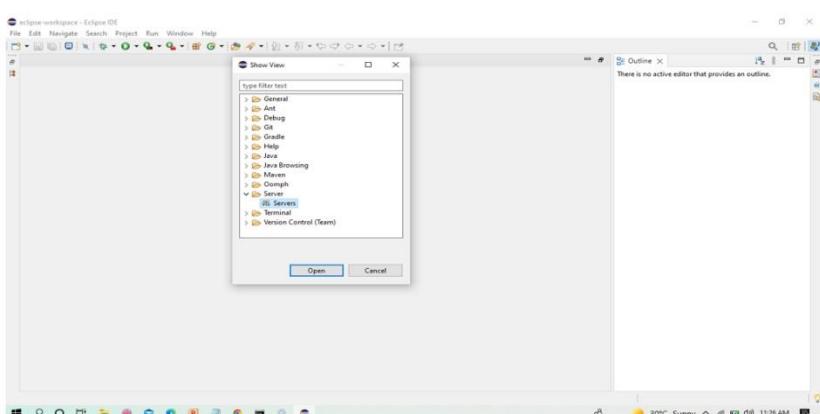
STEP-9: Since we develop only maven web project in devops, we can only check the shown two and click on next.



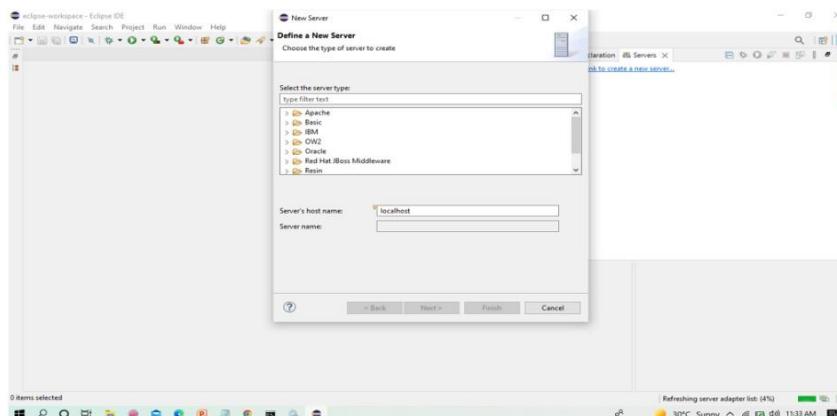
STEP-10: After the shown installation, check whether server is available in eclipse as shown



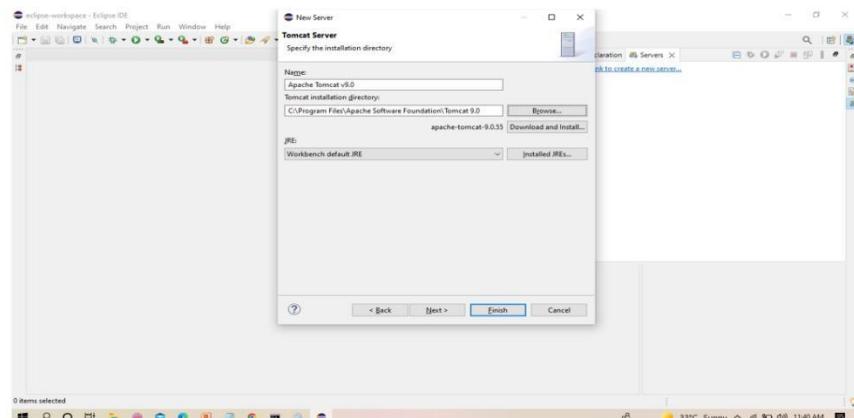
STEP-11: In Show View, we can now see Servers



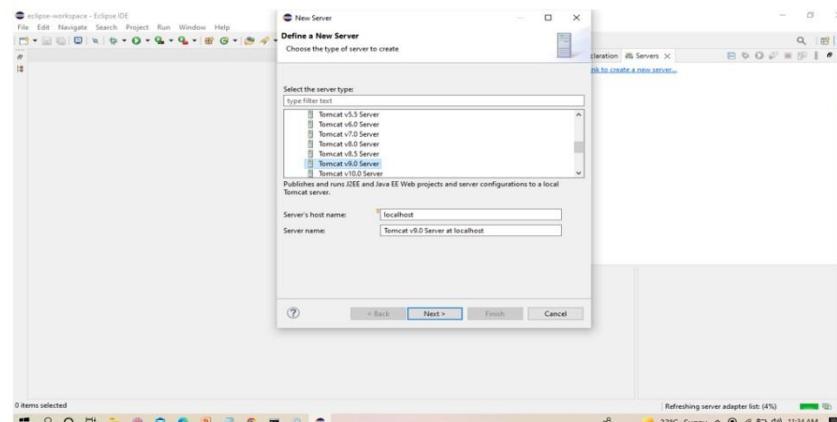
STEP-12: Click on Apache folder



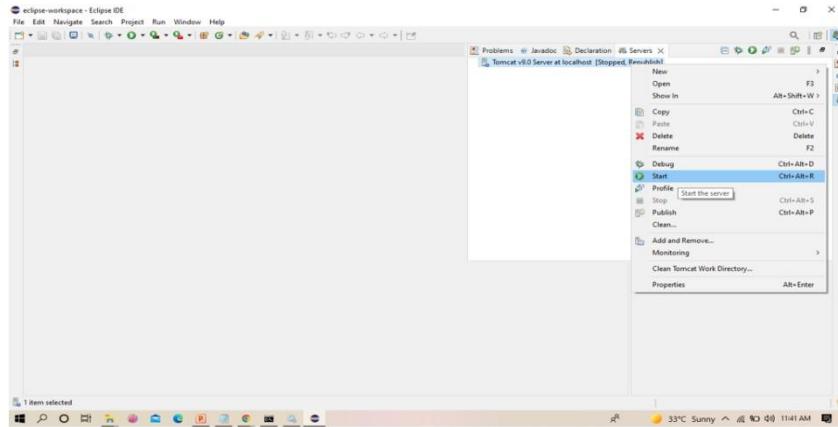
STEP-13:Browse the Tomcat folder available in your local system as shown



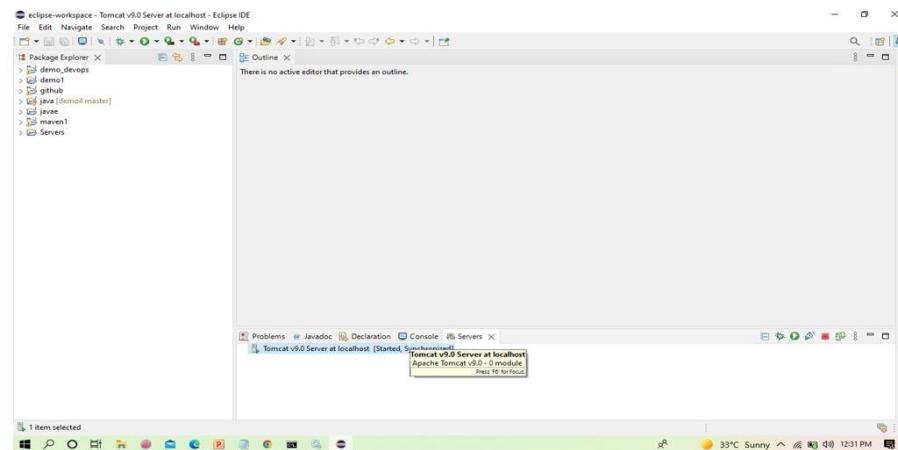
STEP-14:Select Tomcat V 9 as shown



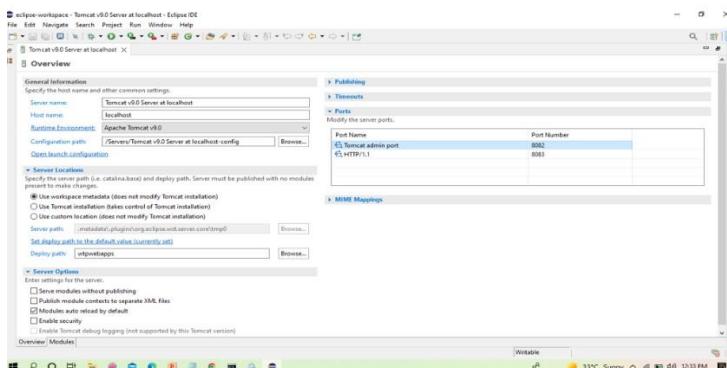
From now on we can start Tomcat server from Eclipse IDE as shown



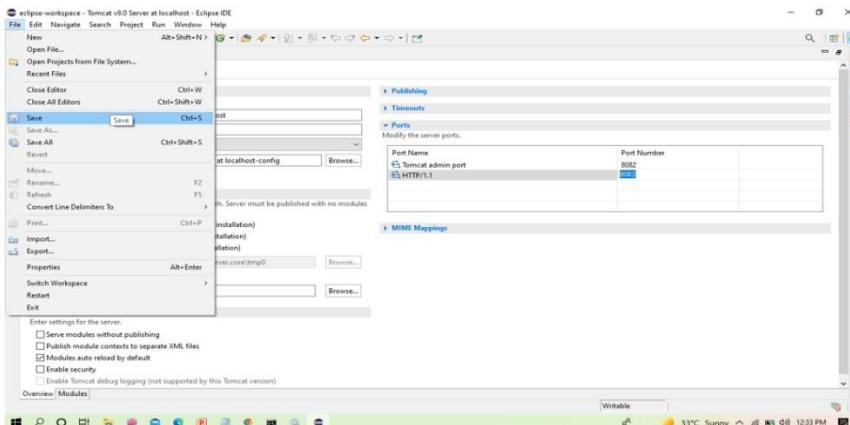
We can now see that the server is started



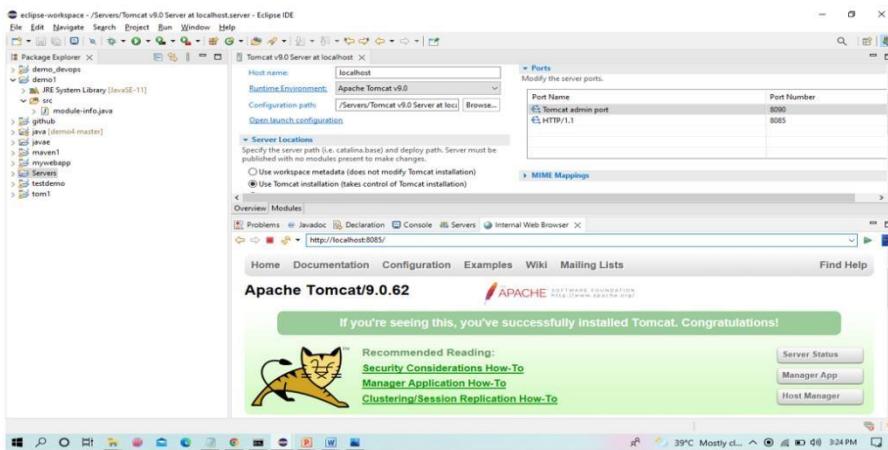
The next step is to change the port number of the Tomcat as Jenkins uses 8080, as shown



In server locations make sure second option is selected



We can see the Tomcat is successfully running on the new port number



Eclipse web

Step 1: Create A New Maven Project

First, we will create a new maven project. For this, we will open our Eclipse IDE then Go to File > New > Maven Project.

```

eclipse-workspace - maven-java-lab/src/main/java/com/kmt/maven_java_lab/App.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
New Open File... Alt+Shift+N > Enterprise Application Project
Open Projects from File System... Ctrl+W Dynamic Web Project
Recent Files Close Editor EJB Project
Close All Editors Connector Project
Ctrl+Shift-W Application Client Project
Ctrl+Shift-S Static Web Project
Save JPA Project
Save As... Project...
Save Save All... F2 Session Bean (EJB 3.x/4.x)
Save All... Revert
Move... Message-Driven Bean (EJB 3.x/4.x)
Rename... F5 Web Service
Convert Line Delimiters To XML File
Print... Folder...
Import... File...
Export... ISP File
Properties Switch Workspace
Restart Ctrl+N Example...
Exit Other... /maven-java-lab (master)
/maven-java-lab (master) /maven-java-labdocumentation.git
git push -u origin master
Counting objects: 49, done.
Delta compression using up to 8 threads
Compressing objects: 100% (49/49), 8.45 KiB / 720.00 KiB, done.
Total 49 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Tankaridash/maven-java-labdocumentation.git
 * [new branch]  master -> master
branch 'master' set up to track 'origin/master'.

```

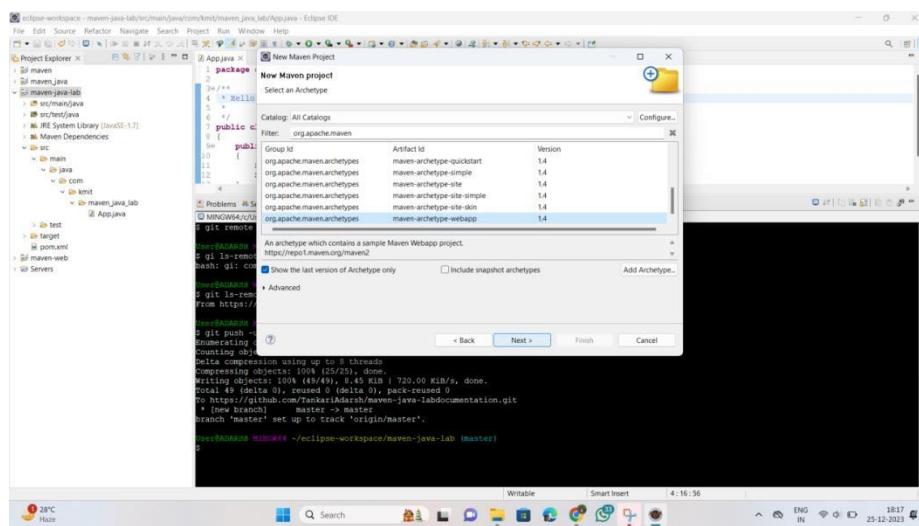
28°C Haze

Once we click on Maven Project we will get a prompt for setting up our Workspace location. After setting Workspace location according to our choice. We will click on Next to go to further project setup steps.

The screenshot shows the Eclipse IDE interface with the 'Project Explorer' view open. In the center, a 'New Maven Project' dialog box is displayed. The 'Create a simple project (skip archetype selection)' checkbox is checked. The 'Location' field contains the path 'C:\User\Umesh\eclipse-workspace\maven-java-lab\src\main\java\com\kmt\maven_java_lab\App.java'. Below the location field, there is a 'Working set' dropdown menu. At the bottom of the dialog box, there are 'Next >' and 'Cancel' buttons.

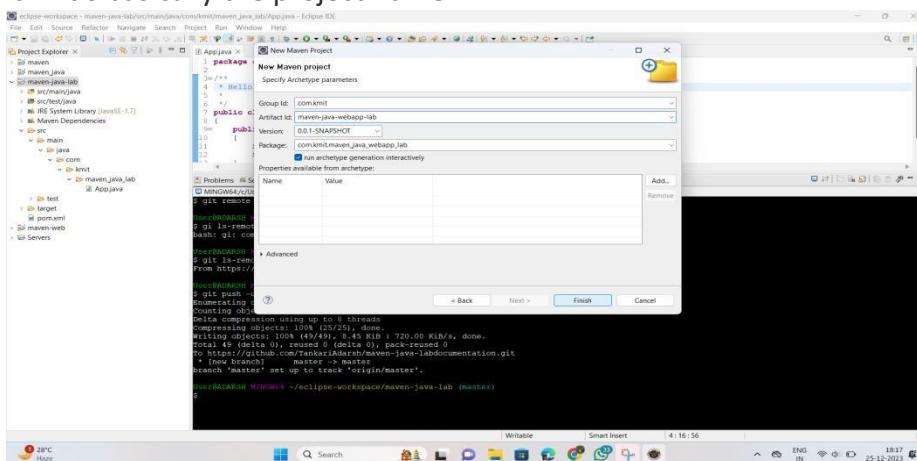
Step 2: Setting Up Archetype

Select the archetype as **maven-archetype-webapp**



After choosing the archetype we will click on Next and it will ask for the details like Group Id, Artifact Id, and Package information.

Group Id – It's a unique identifier to uniquely identify the project in a maven repository
Artifact Id – It's basically the project name.



After adding all these details we need to click on the Finish button. Maven will start creating your project and after a few seconds, your project will be created.

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

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer View:** Shows the project structure:
 - com.kmit.maven_java_lab
 - maven
 - maven.java
 - maven_webs
 - src/main/java
 - src/test/java
 - System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - java
 - com
 - kmt
 - maven.java.lab
 - Aspjava
- Editor View:** Displays the Java code for the `App` class:

```
1 package com.kmit.maven_java_lab;
2
3 /**
4  * Hello world!
5  */
6
7 public class App {
8
9     public static void main( String[] args )
10    {
11        System.out.println( "Hello World!" );
12        System.out.println("Lab documentation");
13    }
14 }
```
- Problems View:** Shows a warning about the URL being copied to the clipboard.

```
Copied URL to clipboard (25-Dec-2023, 6:17:32 pm) [nd: 22240]
downloaded from: https://repo.maven.apache.org/maven/org/apache/maven/maven-archetype-maven-metadata.xml
progress (1): 980 B
```
- Console View:** Shows the output of the Maven command:

```
Downloaded from : https://repo.maven.apache.org/maven2/cry/apache/maven/plugins/maven-archetype-plugin/maven-metadata.xml (980 B at 3.0 kB/s)

[INFO] Scanning for projects...
[INFO] 
[INFO] Building Maven Stub Project (No POM)
[INFO] ------------------------------------------------------------------------
[INFO] 
[INFO] >>> archetype:3.2.1:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< archetype:3.2.1:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO] 
[INFO] --- archetype:3.2.1:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.4] found in catalog remote
[INFO] Using property: artifactId = maven-java-websapp-lab
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: groupId = com.kmit.maven_java_websapp_lab
[INFO] Confirm properties configuration:
groupId: com.kmit
artifactId: maven-java-websapp-lab
version: 0.0.1-SNAPSHOT
package: com.kmit.maven_java_websapp_lab
Y: !
```
- Bottom Status Bar:** Writable, Smart Insert, 4:16:56, Creating maven-archetype-webapp (13%)
- System Icons:** Battery, Network, Date/Time.

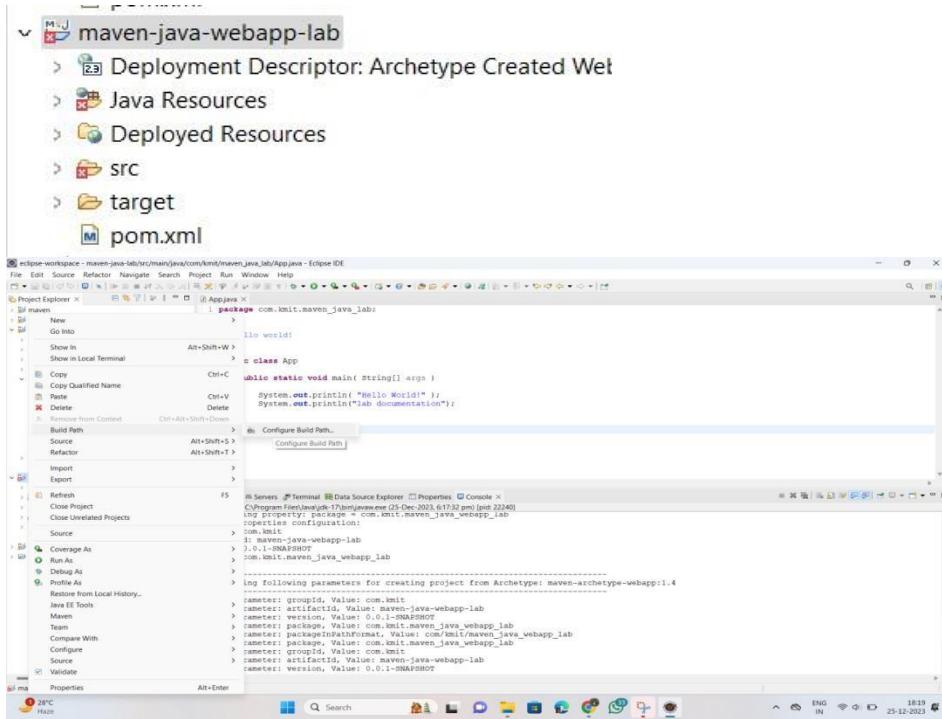
- By clicking on console as "y" you can conform the creation of project

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

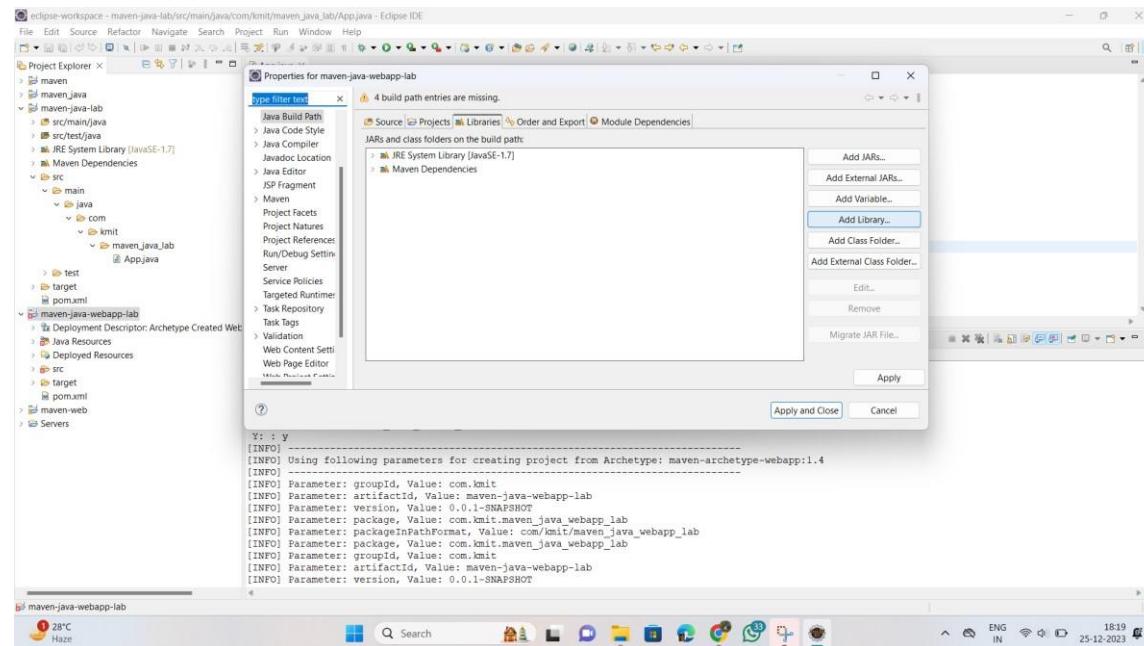
- Project Explorer View:** Shows the project structure:
 - src/main/java: Contains App.java
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src: Contains main, java, com, kmit, and maven_java_lab.
- Terminal View:** Displays the command-line output of the Maven archetype creation process:

```
mvn archetype:generate -DgroupId=com.kmit.maven_java_lab -DartifactId=maven-java-webapp-lab -Dversion=0.0.1-SNAPSHOT -DarchetypeArtifactId=maven-archetype-webapp -Dpackage=com.kmit.maven_java_webapp_lab
[INFO] Scanning for projects...
[INFO] ...
[INFO] Using following parameters for creating project from Archetype: maven-archetype-webapp:1.4
[INFO] Parameter: groupId: com.kmit
[INFO] Parameter: artifactId: maven-java-webapp-lab
[INFO] Parameter: version: 0.0.1-SNAPSHOT
[INFO] Parameter: package: com.kmit.maven_java_webapp_lab
[INFO] Parameter: packageInPathFormat: com/kmit/maven_java_webapp_lab
[INFO] Parameter: package: com.kmit.maven_java_webapp_lab
[INFO] Parameter: groupId: com.kmit
[INFO] Parameter: artifactId: maven-java-webapp-lab
[INFO] Parameter: version: 0.0.1-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\User\eclipse-workspace\maven-java-webapp-lab
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 13.826 s
[INFO] Finished at: 2023-12-25T18:17:49+05:30
[INFO] 
```
- Bottom Status Bar:** Shows system information including battery level (28%), network (WIFI), and date/time (25-12-2023).

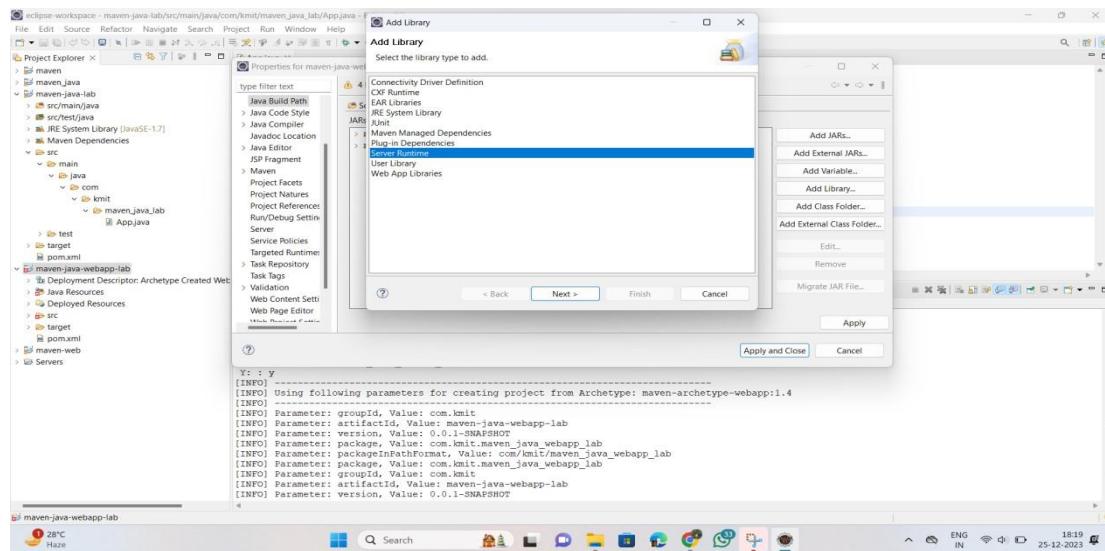
Here “X” mark appears on the project which has created, to remove this marks right click on the folder which has created Build path->config build path



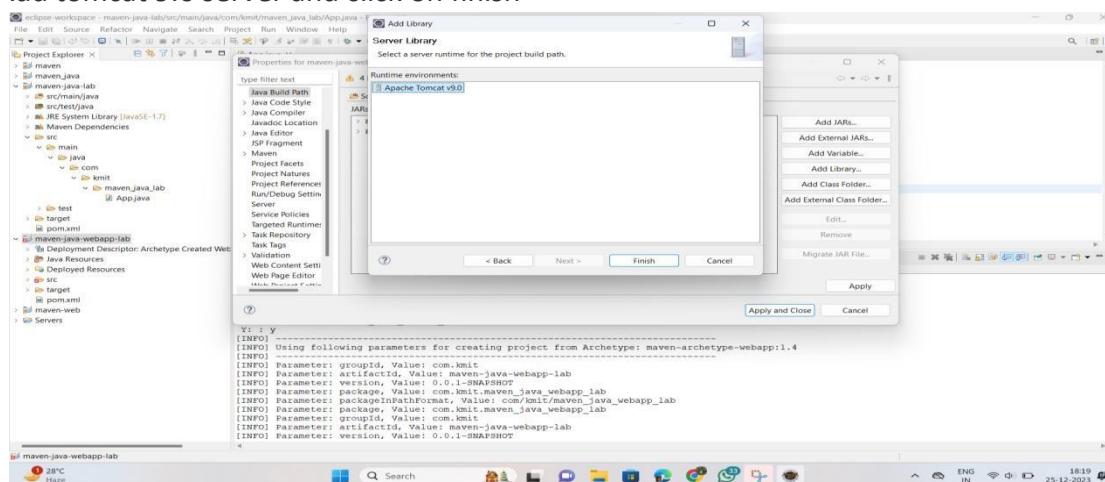
After clicking on config build path we need to add library, so click on add library and add server runtime to the folder



click on next

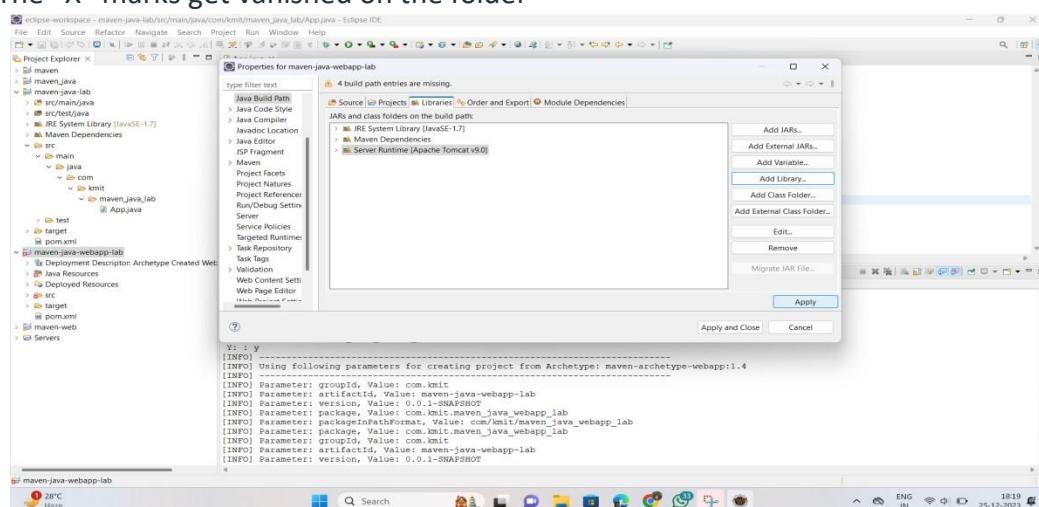


Add tomcat 9.0 server and click on finish



Apply the changes and finish the configuration

The “X” marks get vanished on the folder



To run the maven project right click on folder and click on run as where go with the process

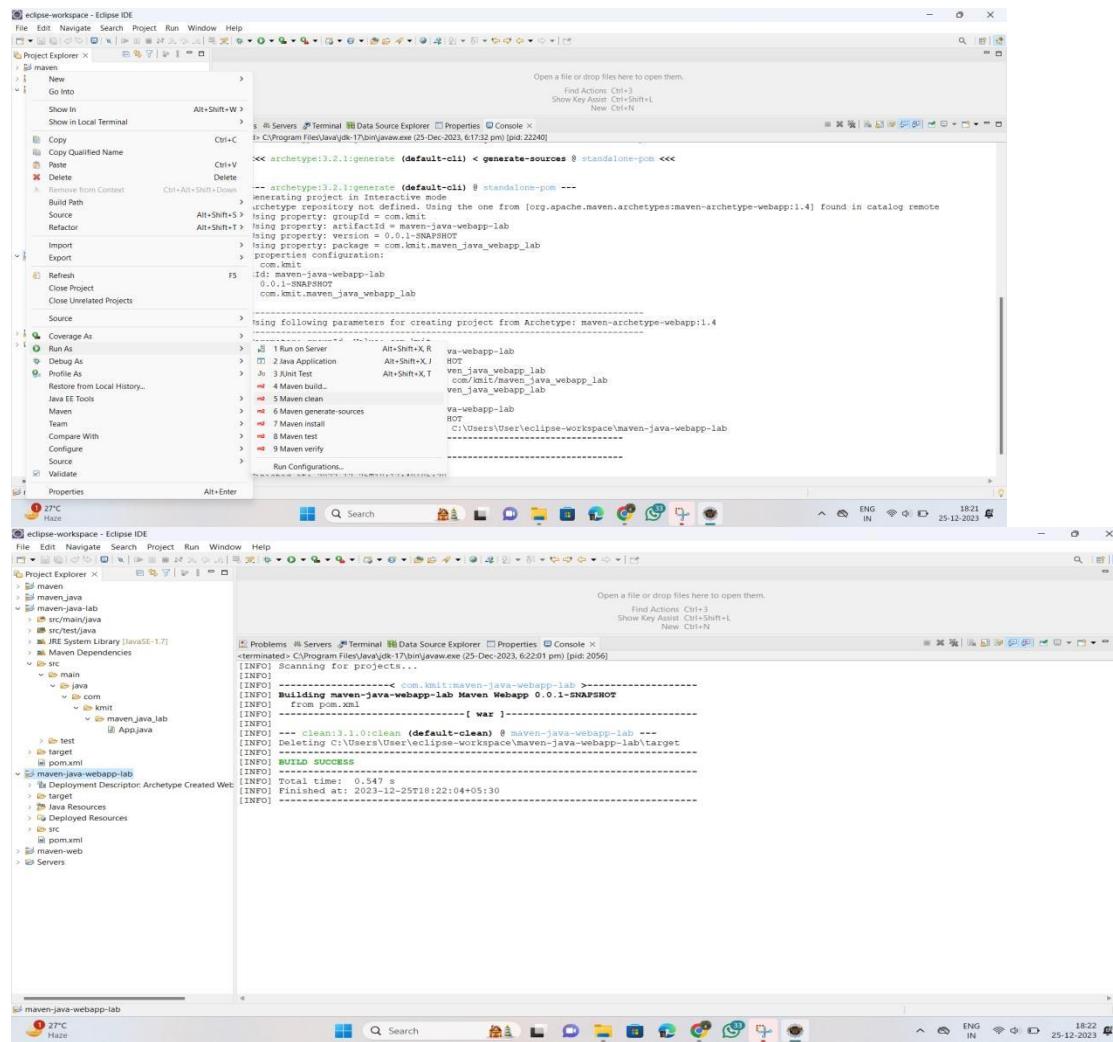
1)Maven clean

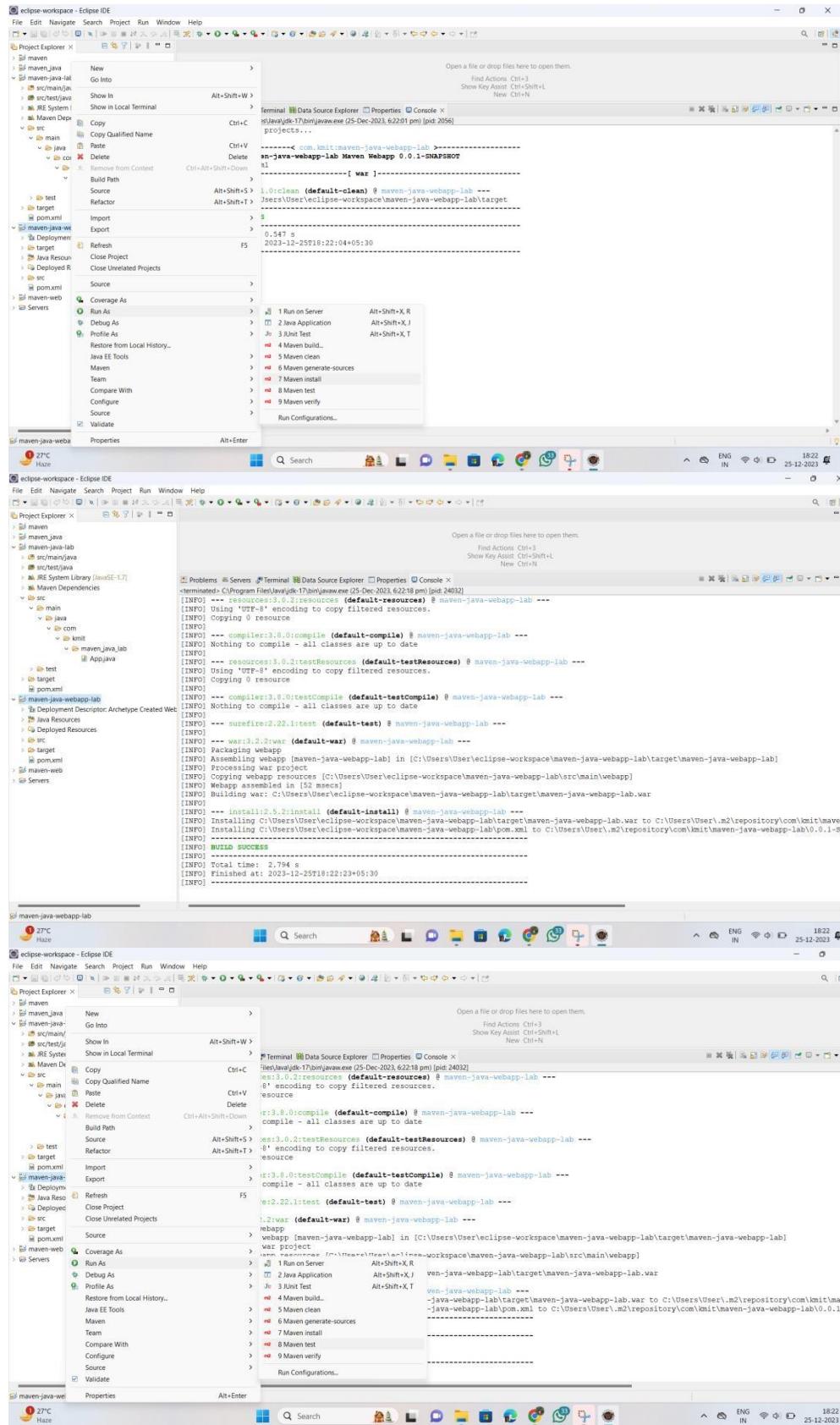
2)Maven install

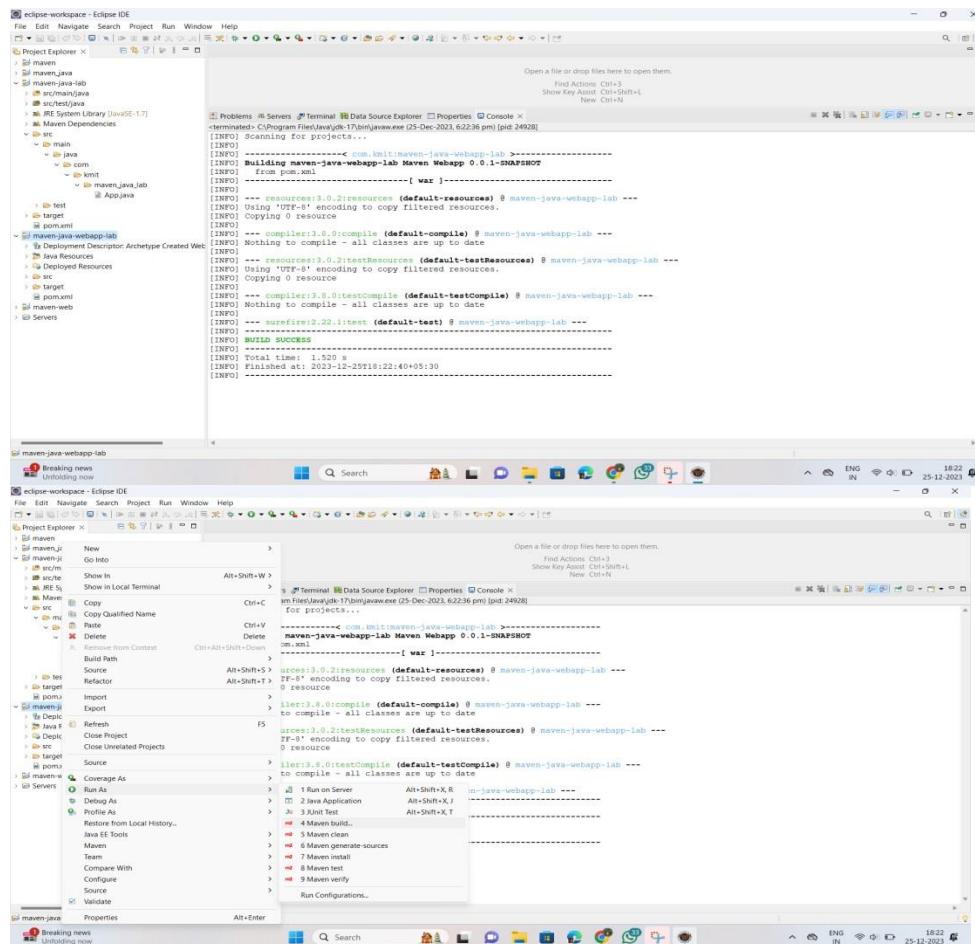
3)Maven test

4)Maven build

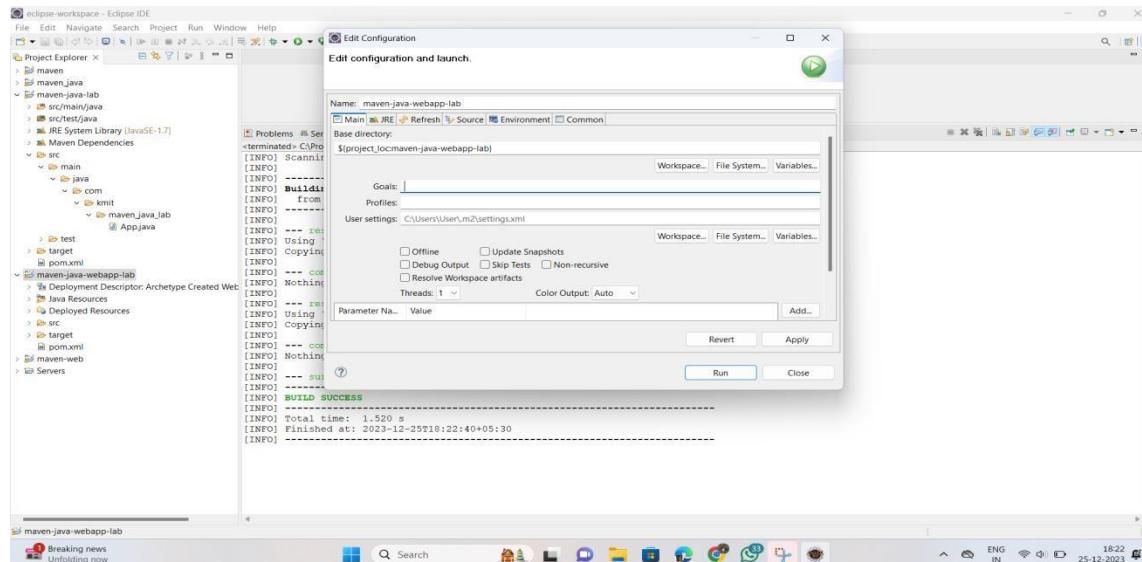
Finally click on run as->run on server

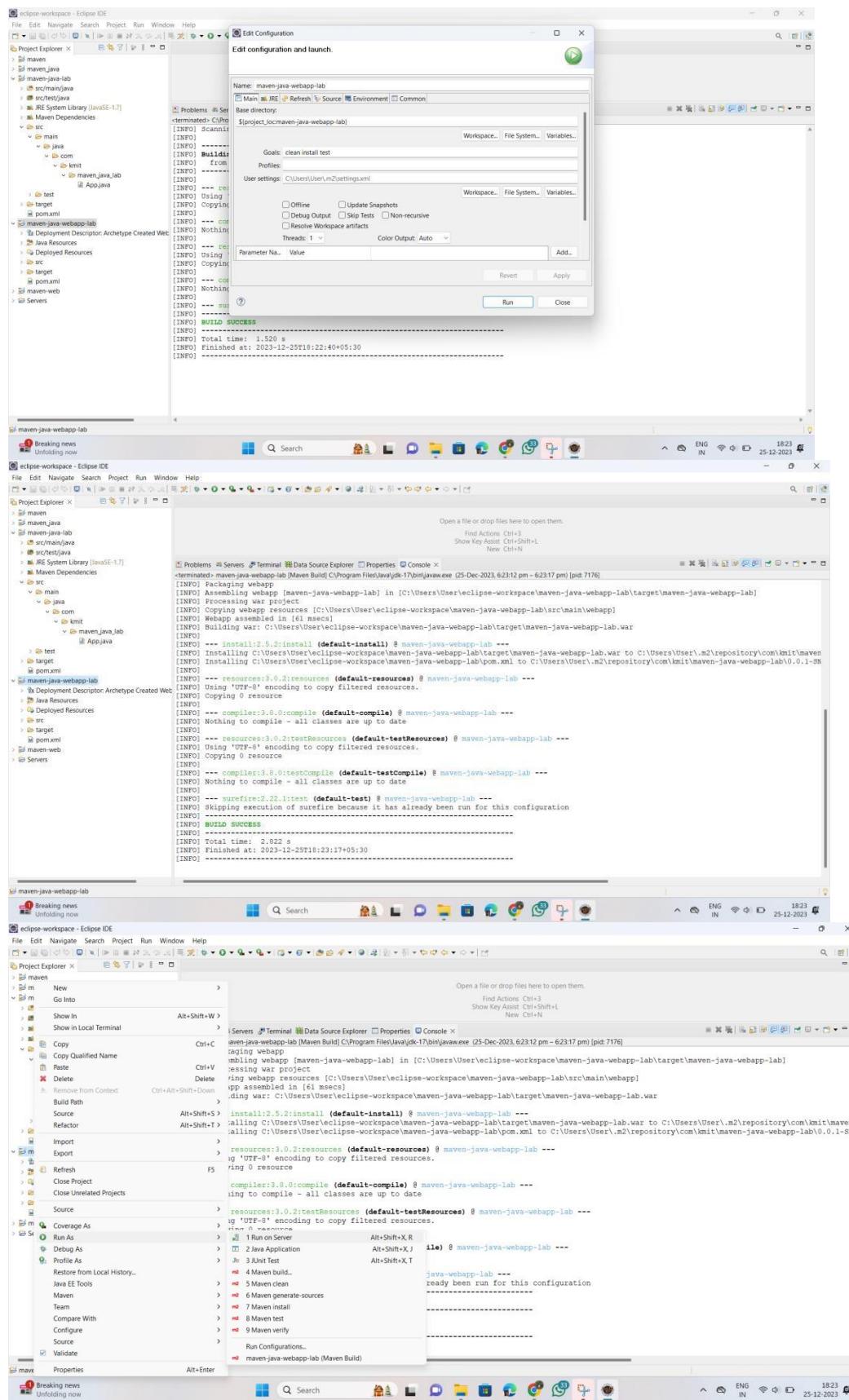




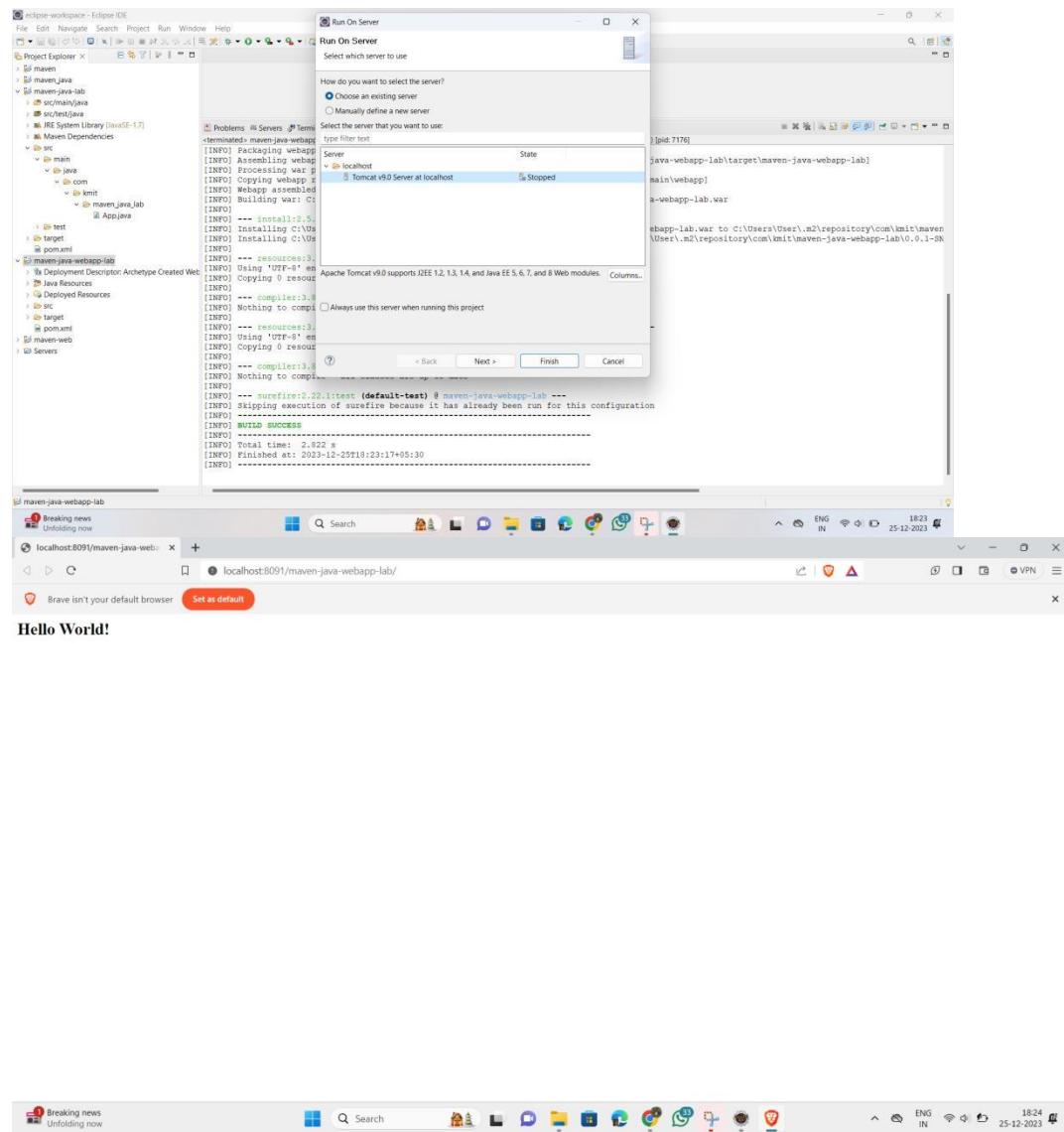


By clicking on maven build a edit configuration and lunch window appers in goals type clean install test and then click on apply and run



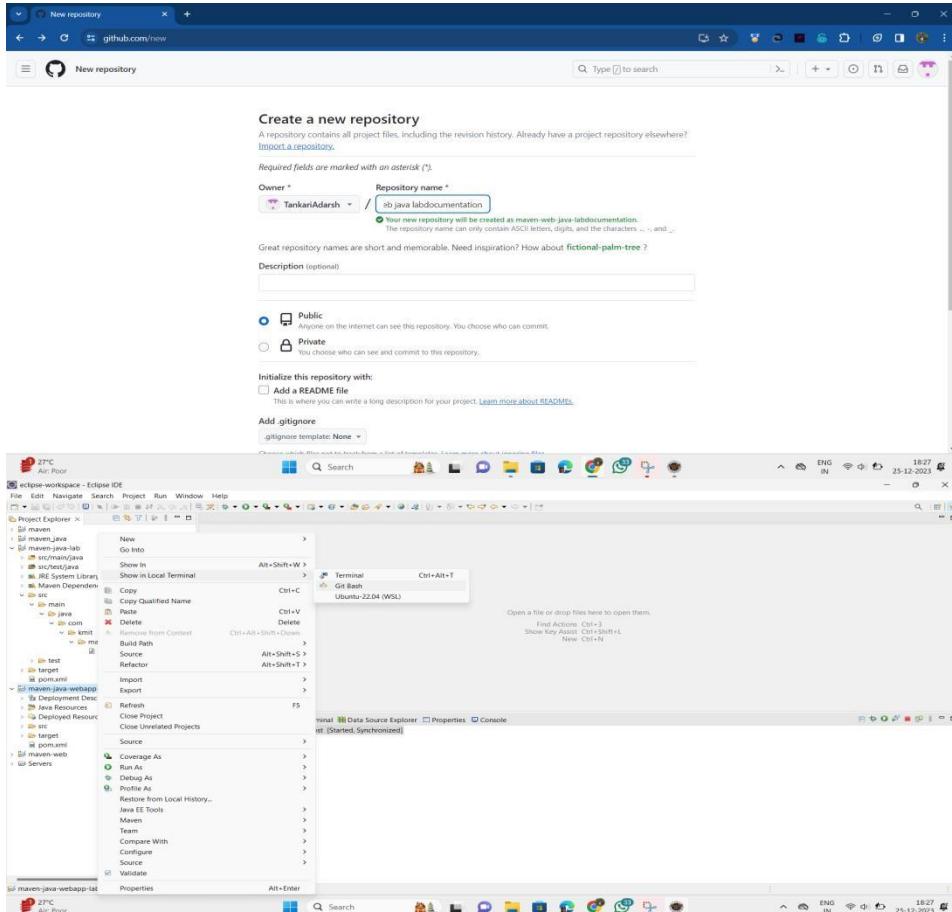


As we click on run as server a window appers where click on tomcat9.0 localhost and Click on next

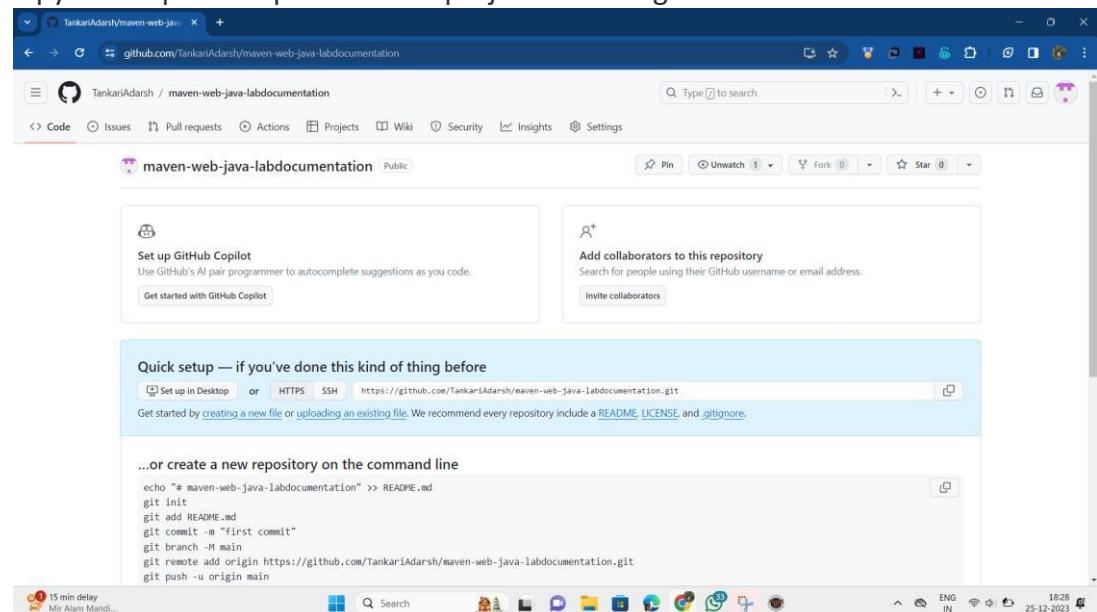


PUSHING MAVEN JAVA PROJECT INTO GITHUB REPIOSTORY:

->Right click on project->show in local terminal->gitbash
First create a new repository in the github



Copy the http link to push the the project into the github



- 1) initialize the empty git repository using “git init” command
- 2) add all files to the empty repository
- 3) commit all the files using “git commit” command
- 4) Then after paste the link that has been copied in the github to the command “git remote add origin link”

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

- Project Explorer:** Shows a Maven project named "maven-java-webapp-lab". The structure includes:
 - maven
 - maven Java
 - maven-java-lab
 - src/main/java
 - com
 - kmit
 - maven_java_lab
 - App.java
 - src/test
 - target
 - pom.xml
 - maven-java-webapp-lab
 - Deployment Descriptor: Archetype Created Web
 - Java Resources
 - Deployed Resources
 - src
 - target
 - pom.xml
 - maven-web
 - pom.xml
 - Servers
- Terminal:** Displays the following terminal session:


```

user@ADARSH:~/eclipse-workspace/maven-Java-webapp-lab
$ ls
pom.xml  src/  target/
user@ADARSH:~/eclipse-workspace/maven-Java-webapp-lab
$ git init
Initialized empty Git repository in C:/Users/User/eclipse-workspace/maven-Java-webapp-lab/.git/
user@ADARSH:~/eclipse-workspace/maven-Java-webapp-lab (master)
$ git add .
warning: in the working copy of 'src/main/webapp/WEB-INF/web.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/webapp/index.jsp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-Java-webapp-lab/WEB-INF/web.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-Java-webapp-lab/index.jsp', LF will be replaced by CRLF the next time Git touches it
user@ADARSH:~/eclipse-workspace/maven-Java-webapp-lab (master)
$ git commit -m"maven web committed"
[master (root-commit) 8378c95] maven web committed
 20 files changed, 228 insertions(+)
create mode 100644 .classpath
create mode 100644 .project
create mode 100644 .settings/.jdtScope
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 src/main/webapp/index.jsp
create mode 100644 target/maven-archiver/pom.properties
create mode 100644 target/maven-Java-webapp-lab.war
create mode 100644 target/maven-Java-webapp-lab/WEB-INF/web.xml
create mode 100644 target/maven-Java-webapp-lab/index.jsp
create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles.lst
create mode 100644 target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst
user@ADARSH:~/eclipse-workspace/maven-Java-webapp-lab (master)
$ git remote add origin "https://github.com/TankariAdarsh/maven-web-Java-labdocumentation.git"
```

Finally push all the files to the github account using the command “git push”
 Check the github wheather the files are pushed into it

Eclipse IDE - eclipse-workspace

File Edit Navigate Search Project Run Window Help

Project Explorer X

- maven
- maven-xml
- maven-java-lab
 - src/main/java
 - src/test/java
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - main
 - java
 - com
 - kmit
 - maven-java-lab
 - App.java
 - target
 - pom.xml
- maven-java-webapp-lab
 - Deployment Descriptor: Archetype Created Web
 - Java Resources
 - src
 - target
 - pom.xml
- maven-web
- Servers

Open a file or drop files here to open them.

Find Actions Ctrl+...

Problems Servers Terminal Data Source Explorer Properties Console

```

MINGW64:/c/Users/User/eclipse-workspace/maven-java-lab MINGW64:/c/Users/User/eclipse-workspace/maven-java-lab X
create mode 100644 src/main/webapp/index.jsp
create mode 100644 target/maven-archiver/pom.properties
create mode 100644 target/maven-java-webapp-lab.war
create mode 100644 target/maven-java-webapp-lab/WEB-INF/web.xml
create mode 100644 target/maven-java-webapp-lab/index.jsp
create mode 100644 target/maven-status/maven-compiler-plugin/default-compile/inputFiles.lst
create mode 100644 target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-webapp-lab (master)
$ git remote add origin "https://github.com/TankariAdarsh/maven-web-java-labdocumentation.git"
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-webapp-lab (master)
$ git ls-remote
From https://github.com/TankariAdarsh/maven-web-java-labdocumentation.git
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-webapp-lab (master)
$ git push -u origin master
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (31/31), 5.7 KiB | 480.00 KiB/s, done.
Total 31 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/TankariAdarsh/maven-web-java-labdocumentation.git
 * [new branch]  master -> master
branch 'master' set up to track 'origin/master'.
User@ADARSH MINGW64 ~/eclipse-workspace/maven-java-webapp-lab (master)
$
```

Connected - Encoding: UTF-8

Haze 27°C

18:29 25-12-2023

TankariAdarsh/maven-web-java-labdocumentation GitHub

github.com/TankariAdarsh/maven-web-java-labdocumentation

TankariAdarsh / maven-web-java-labdocumentation

Type ⌂ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

maven-web-java-labdocumentation Public

master 1 Branch 0 Tags

Go to file Add file Code

nithinpotti maven web committed 8378c95 · 1 minute ago 1 Commits

File	Commit Message	Time
.settings	maven web committed	1 minute ago
src/main/webapp	maven web committed	1 minute ago
target	maven web committed	1 minute ago
.classpath	maven web committed	1 minute ago
.project	maven web committed	1 minute ago
pom.xml	maven web committed	1 minute ago

README

Add a README

Pin Unwatch Fork Star

About

No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

Java 100%

Haze 27°C

18:29 25-12-2023

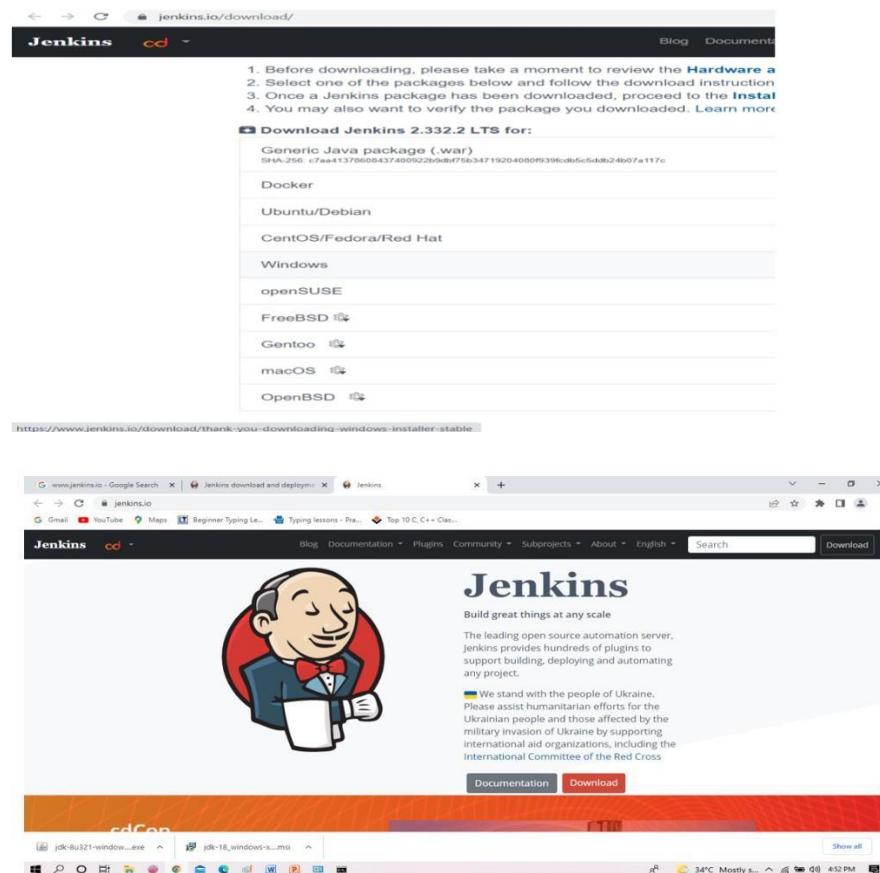
Experiment 7:

Working with Jenkins

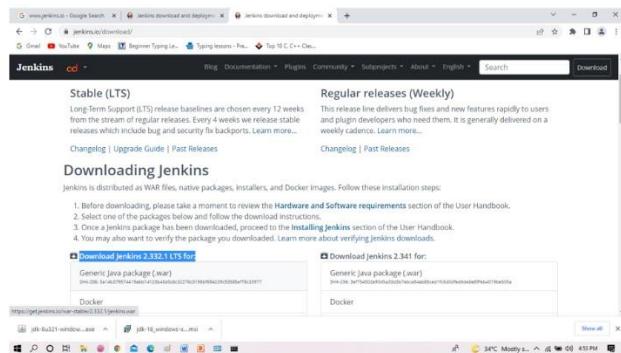
- Create a CI/CD pipeline using Jenkins for a sample Maven Java project
- Create a CI/CD pipeline using Jenkins for a sample Maven Web project.

JENKIN'S INSTLLATION

Step 1) Go to <https://www.jenkins.io/download/> and select the platform. In our case Windows



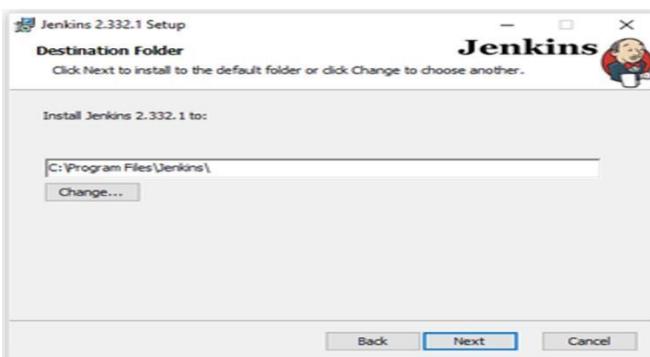
STEP-2: Download jenkins as shown



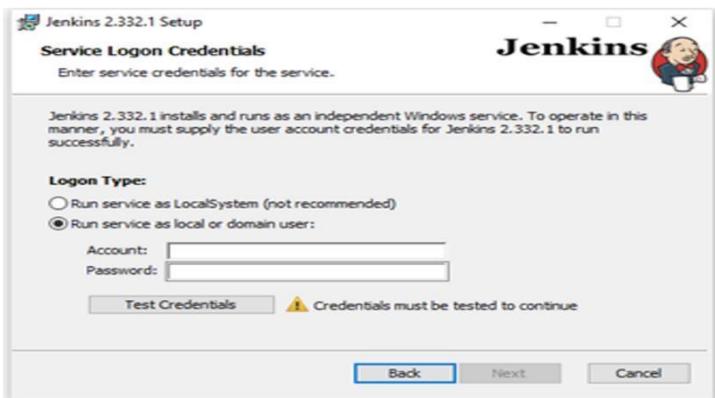
STEP-3: Installation of jenkins



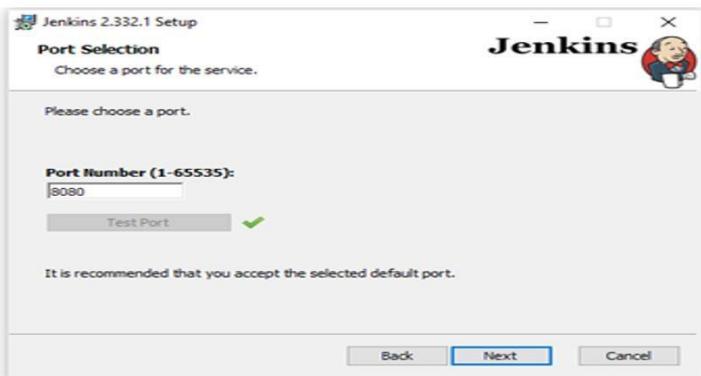
STEP-4: Installation of Jenkins follows..



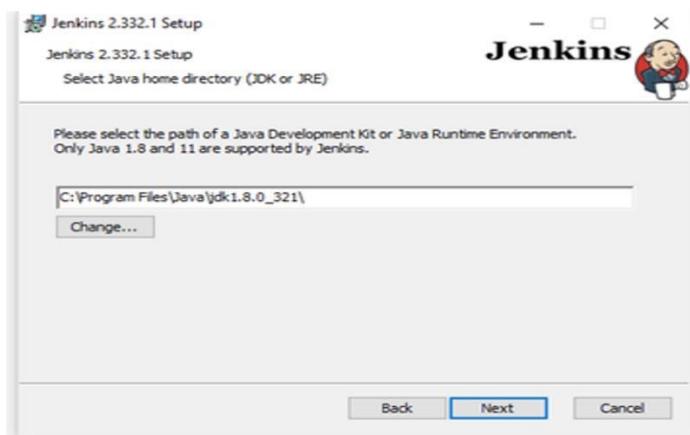
STEP-5: From the server logon credentials select run server as LocalSystem(not recommended)



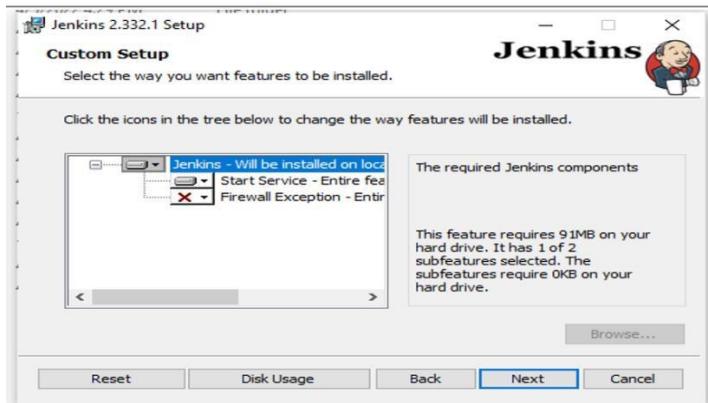
STEP-6:Give any port number and verify it.



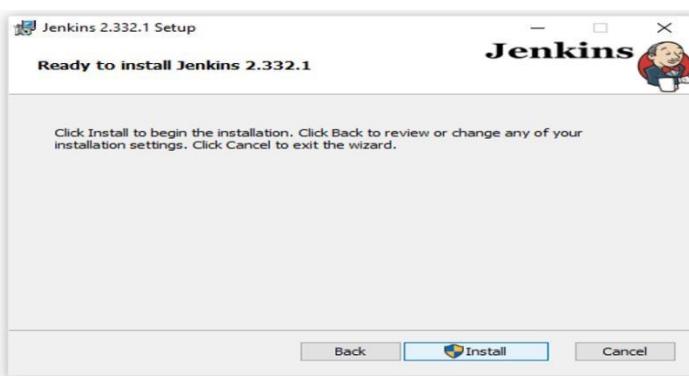
STEP-7:give the java path of either version 11 or 17 or 21



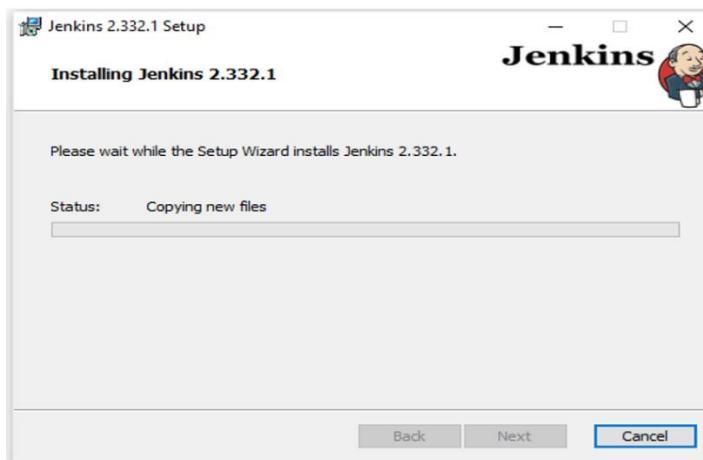
STEP-8:Click on Next



STEP-9:Click on Install



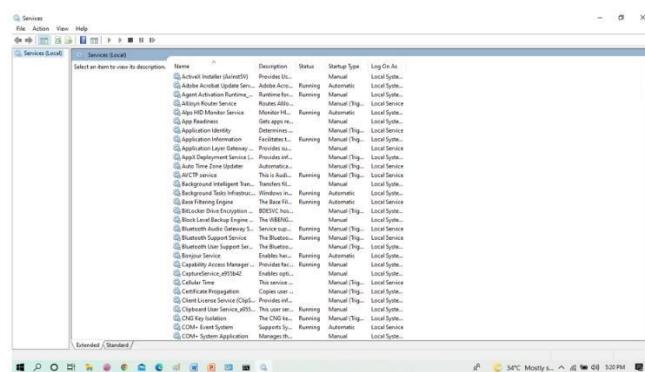
STEP-10:Installation Begins



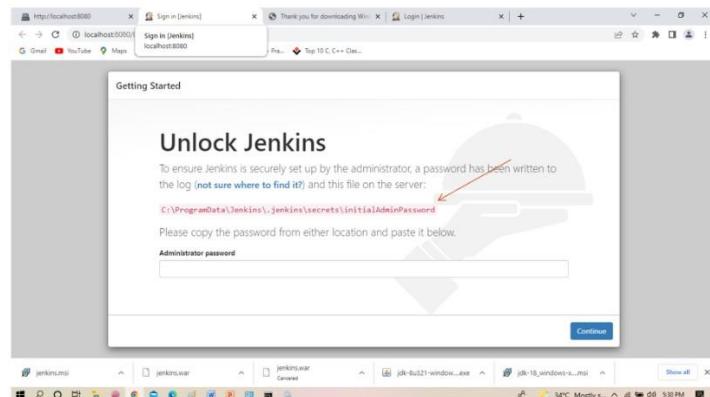
STEP-11:Click on Finish



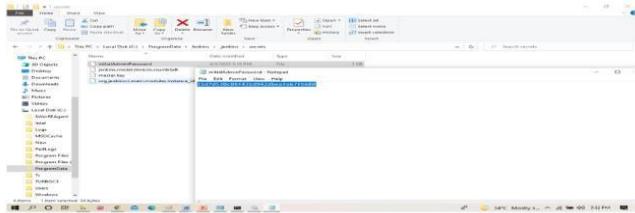
STEP-12:Check the Jenkins installation in services



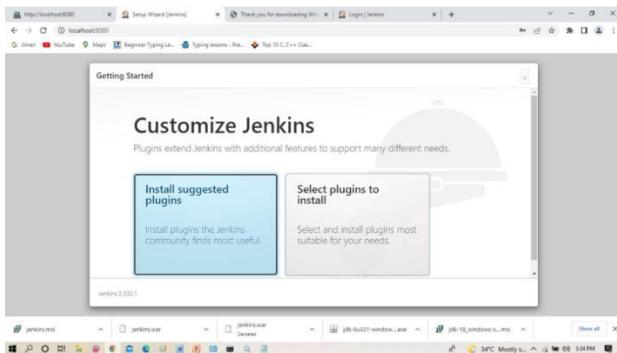
STEP-13:Go to localhost:port number and copy the path of the password



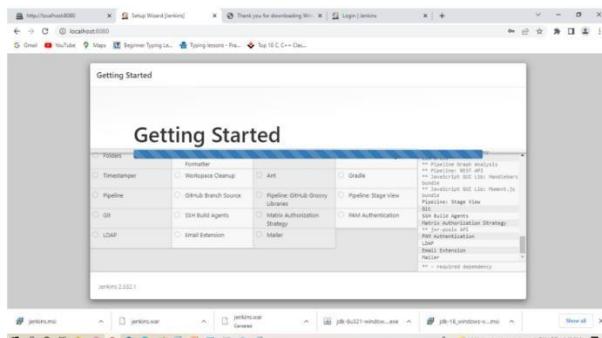
STEP-14:Copy the password which is generated by the path



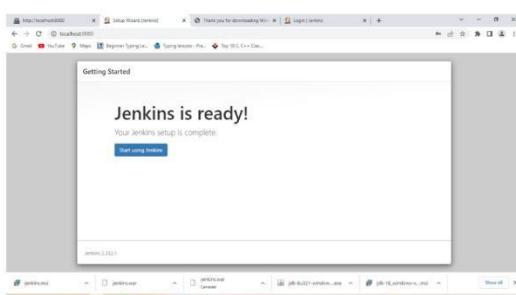
STEP-15:Click on Installed plugins



STEP-16:The plugins gets installed



STEP-17:Jenkins gets started on system.



STEP-18:Sigin into the Jenkins page using username and password



Welcome to Jenkins!

softwareteam

password

Sign in

Keep me signed in



STEP-19:On the dashboard go to ManageJenkins

Name	Last Success	Last Failure	Last Duration
devops.build	9 days 16 hr	#1	39 sec
devops.test	9 days 16 hr	N/A	12 sec
maven.build	1 day 15 hr	#2	17 sec
maven.deploy	N/A	1 day 15 hr	5.7 sec
maven.test	1 day 15 hr	#2	12 sec
mavenbuild_deploy	16 hr	#1	4.8 sec
mavenbuild_build	16 hr	#2	12 sec
mavenbuild_deploy	N/A	#1	4.9 sec
mavenbuild_test	16 hr	#2	7.6 sec

STEP-20:Click on Plugins and install the following plugins without restart

Manage Jenkins

Building on the built-in mode can be a security issue. You should set up distributed builds. See the documentation.

System Configuration

- Configure System
- Global Tool Configuration
- Manage Plugins

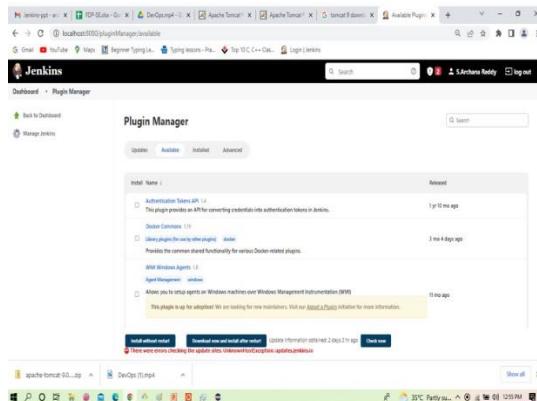
Warnings have been published for the following currently installed components:

- Build Publisher Plugin 1.10
- Stated XSS vulnerability

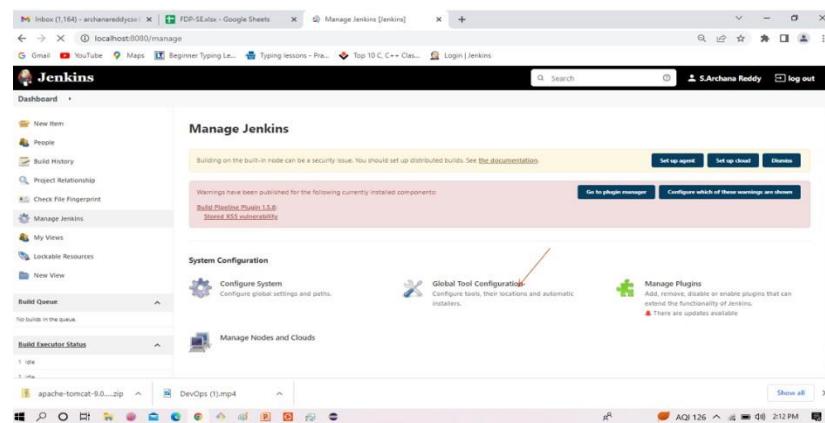
[Go to plugin manager](#) [Configure which of these warnings are shown](#)

Manage Nodes and Clouds

Managed Files



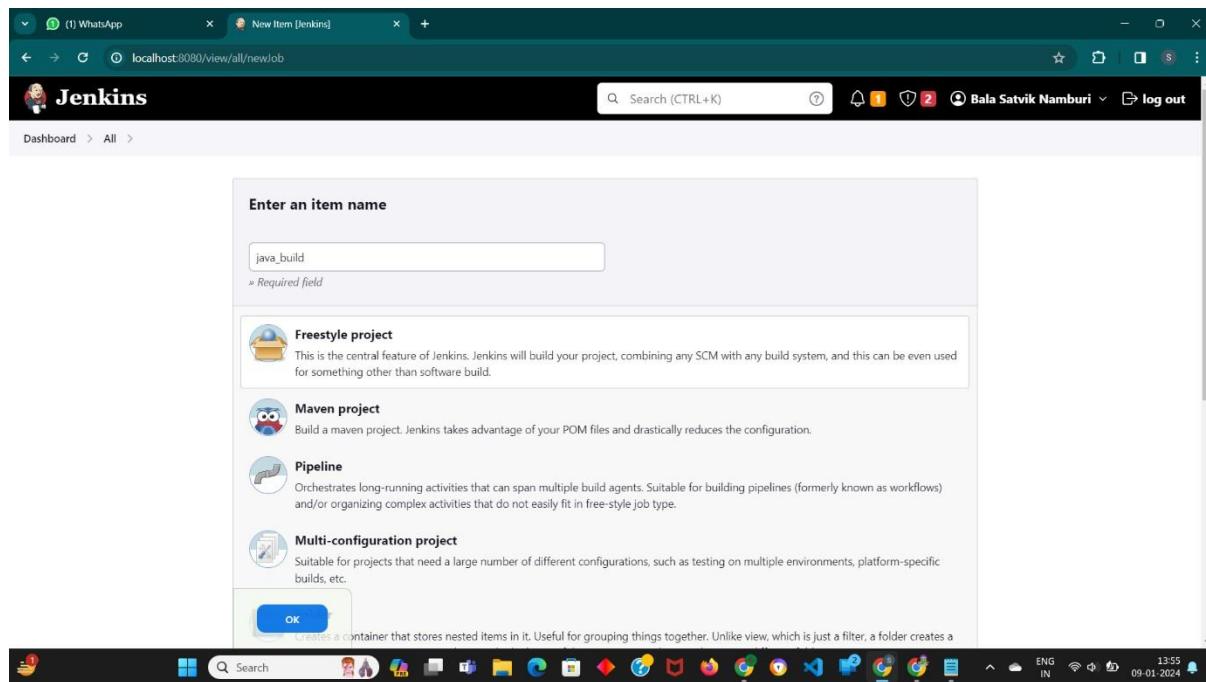
STEP-21:On the Jenkins manager ,click on tools.



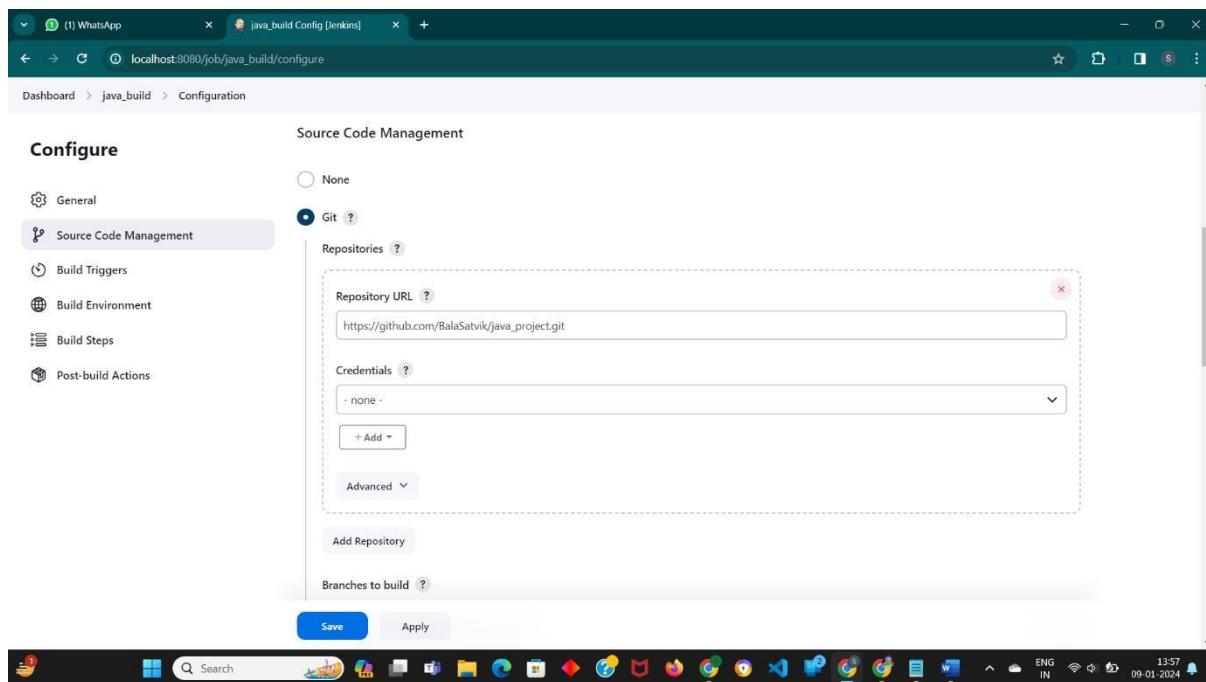
STEP-22:Give the jdk path of java version



STEP-23:Give the Name of installed maven as MAVEN_HOME



Step 2: In source code management select git and paste github repository where java project is present.



Maven

Maven installations

Add Maven

Name: MAVEN_HOME

Install automatically ?

Install from Apache Version: 3.6.5

Delete installer

Add Installer

Name: MAVEN_HOME

Install automatically ?

Install from Apache Version: 3.6.5

Delete installer

Add Installer

Add Maven

Install Maven installations and their artifacts

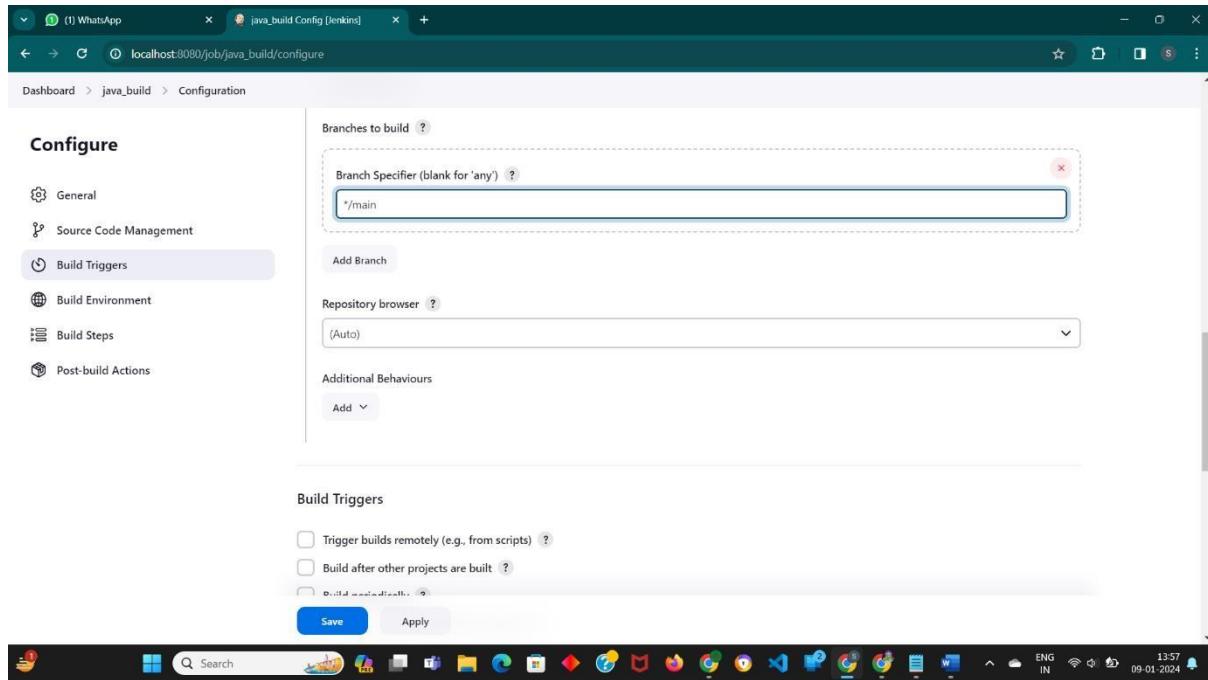
PIPELINE

2-PIPELINE OF MAVEN JAVA PROJECT

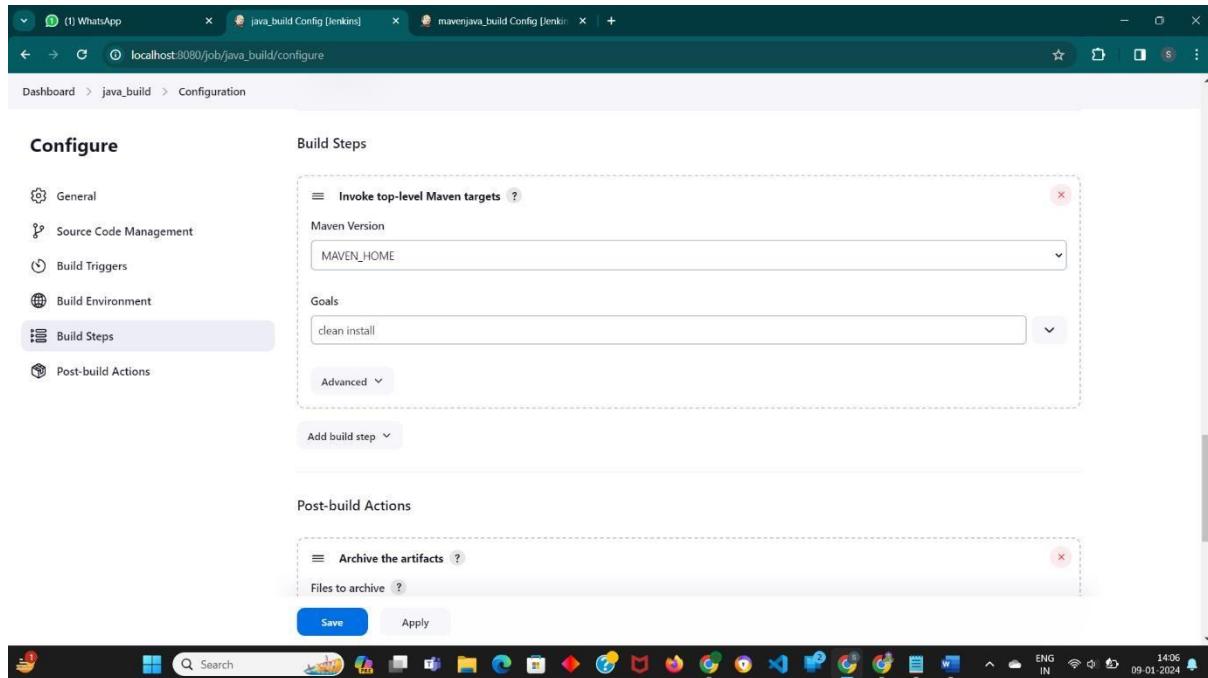
Step 1: Go to Dashboard->New Item. Then enter project name as java_build, select free style project.

Then click on OK.

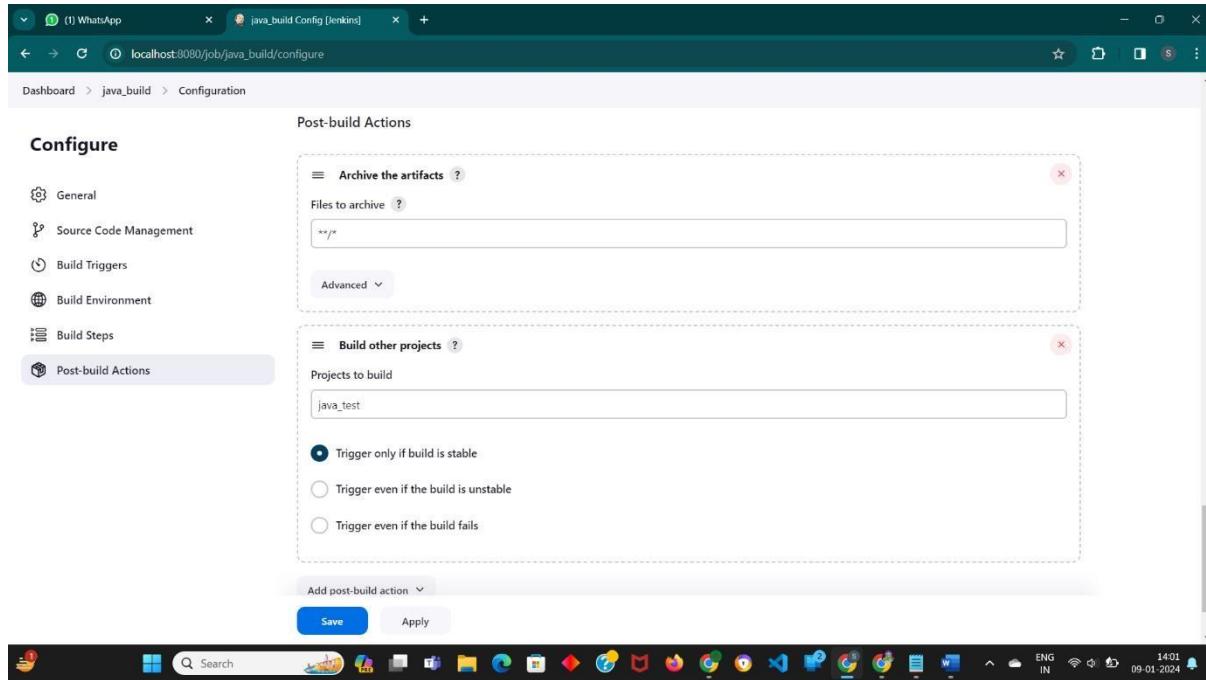
Step 3: Check branch specifier is */main or not. If not update to */main.



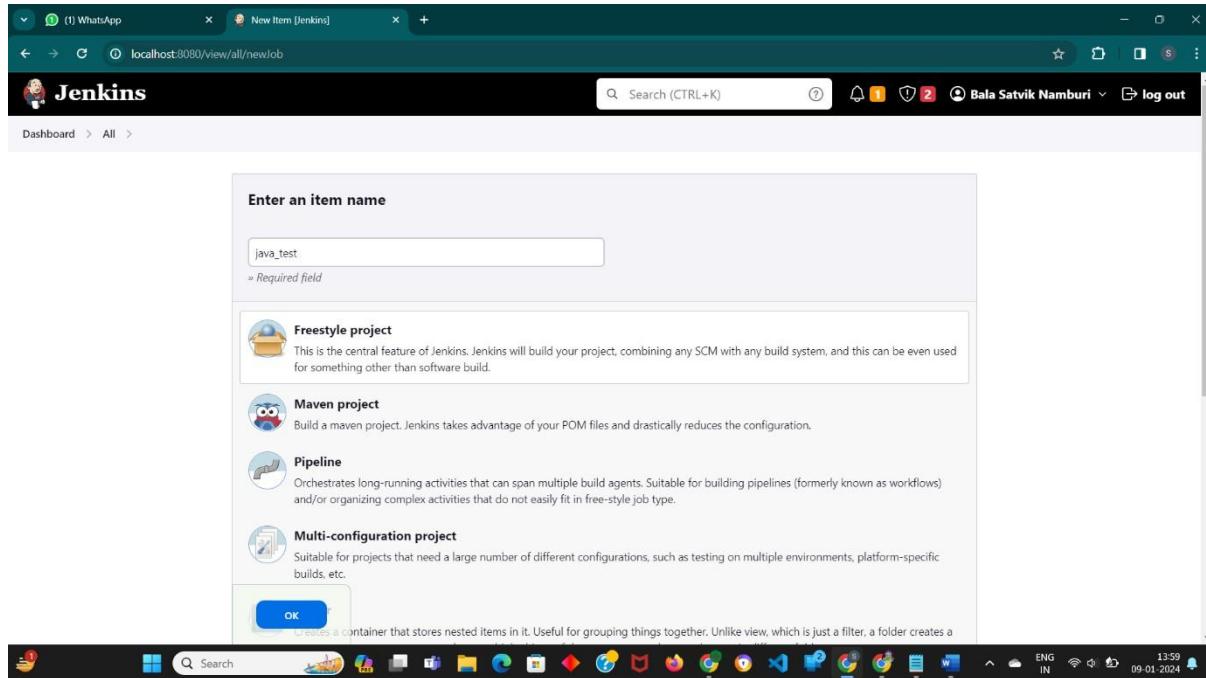
Step 4: In build steps, click on Add build step and select “**Invoke top-level maven targets**”. In that select maven version as MAVEN_HOME and enter goals as “clean install”.



Step 5: In post-build actions click on add post build actions and select “archive the artifacts” and enter files to archive as “**/*”. Again, click on add post build actions and select “Build other projects”. Then enter the project name as “java_test” and then click on apply and save.



Step 6: Again, go to dashboard click on new item and project name as “java_test” and select as freestyle project. Then click on OK.



Step 7: In build environment select “**delete workspace before build starts**”.

A screenshot of a web browser showing the Jenkins job configuration page for 'java_test'. The title bar says 'localhost:8080/job/java_test/configure'. The left sidebar shows 'Configure' sections: General, Source Code Management, Build Triggers (selected), Build Environment, Build Steps, and Post-build Actions. Under 'Build Environment', the checkbox 'Delete workspace before build starts' is checked. Other options like 'Advanced' and 'With Ant' are also listed. At the bottom are 'Save' and 'Apply' buttons.

Step 8: In build steps click on add build step and select “**copy artifacts from other projects**”.

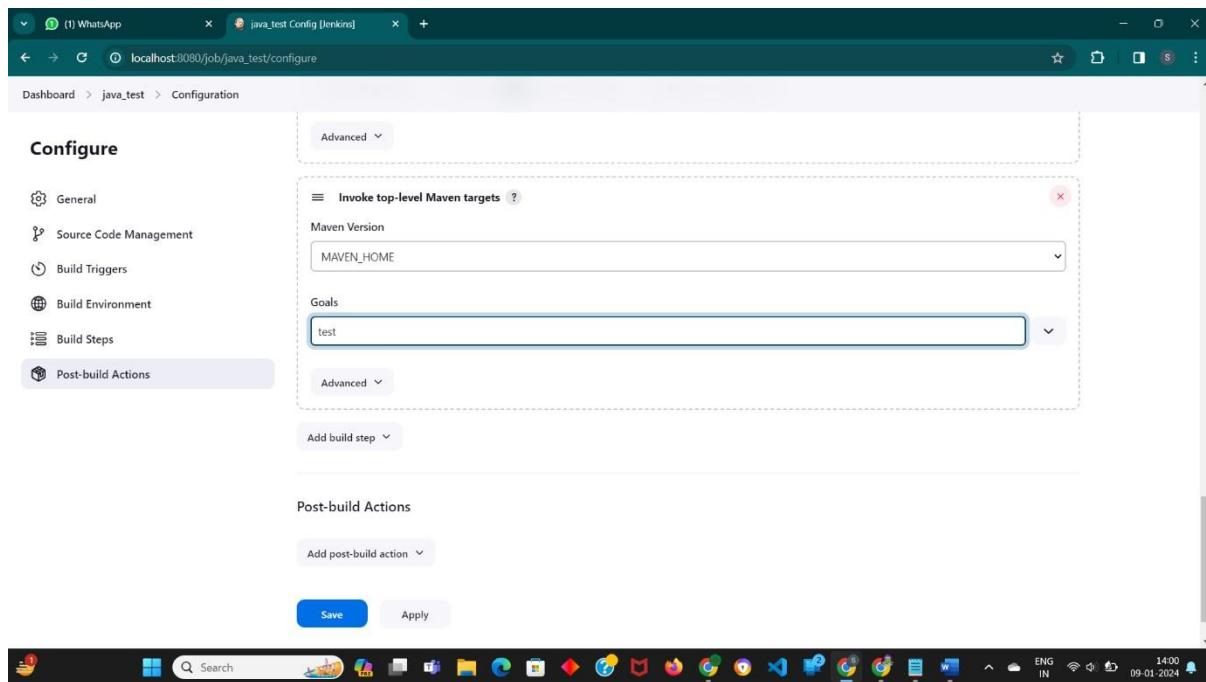
Then in Artifacts to copy enter “**/*”.

The screenshot shows the Jenkins job configuration interface for a job named "java_test". The left sidebar has a "Build Steps" section selected. A sub-section titled "Copy artifacts from another project" is expanded, showing the following configuration:

- Project name:** java_build
- Which build:** Latest successful build
- Stable build only:** Checked
- Artifacts to copy:** **/* (highlighted in blue)
- Artifacts not to copy:** (empty)
- Target directory:** (empty)

At the bottom of the configuration panel are "Save" and "Apply" buttons. The Windows taskbar at the bottom of the screen shows various application icons and system status.

Step 9: Again, click on add build step and select “**Invoke top-level maven targets**”. Then select maven version as MAVEN_HOME and enter goals as “test” and then click on apply and save.



Step 10: Go to dashboard and click on new view.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Build Queue', and 'Build Executor Status'. The 'Build Queue' section indicates 'No builds in the queue.' In the center, there's a search bar and a 'New View' button. Below it, a table displays build history for various jobs. The table has columns: S (Status), W (Last Success), Name, Last Success, Last Failure, and Last Duration. The jobs listed are: java_build (#3), java_test (#1), mavenbuild (#1), mavendeploy (#1), mavenjava_build (#1), mavenjava_test (#1), maventest (#1), mavenweb_build (#2), and mavenweb_deploy (#1). The dashboard also includes a weather widget at the bottom left and a system status bar at the bottom right.

S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	java_build	21 hr #3	21 hr #2	13 sec
✓	Sun	java_test	21 hr #1	N/A	7.4 sec
✓	Sun	mavenbuild	21 hr #1	N/A	12 sec
✓	Sun	mavendeploy	21 hr #1	N/A	1.9 sec
✓	Sun	mavenjava_build	23 days #1	N/A	2 min 0 sec
✓	Sun	mavenjava_test	23 days #1	N/A	5.6 sec
✓	Sun	maventest	21 hr #1	N/A	5.1 sec
✓	Sun	mavenweb_build	27 days #2	N/A	8.3 sec
✓	Sun	mavenweb_deploy	27 days #1	N/A	0.95 sec

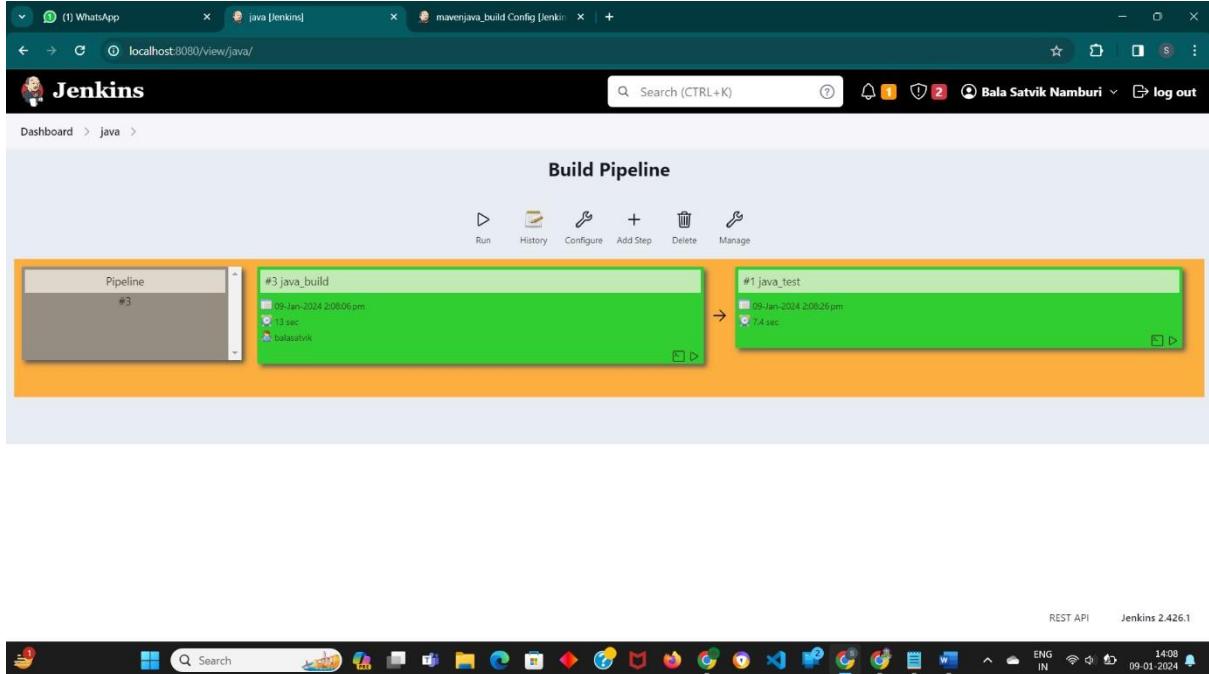
Step 11: Give name and select “build pipeline view”. Click on create.

The screenshot shows the Jenkins 'New view' configuration page. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area has a 'Name' field containing 'JAVA' and a 'Type' section where 'Build Pipeline View' is selected. Below it, there are two other options: 'List View' and 'My View'. At the bottom right of the main area is a blue 'Create' button. The status bar at the bottom indicates 'REST API Jenkins 2.426.1'. The taskbar at the very bottom shows various application icons and the date/time '09-01-2024 14:02'.

Step 12: In select initial job make sure that pipeline starts from “java_build”.

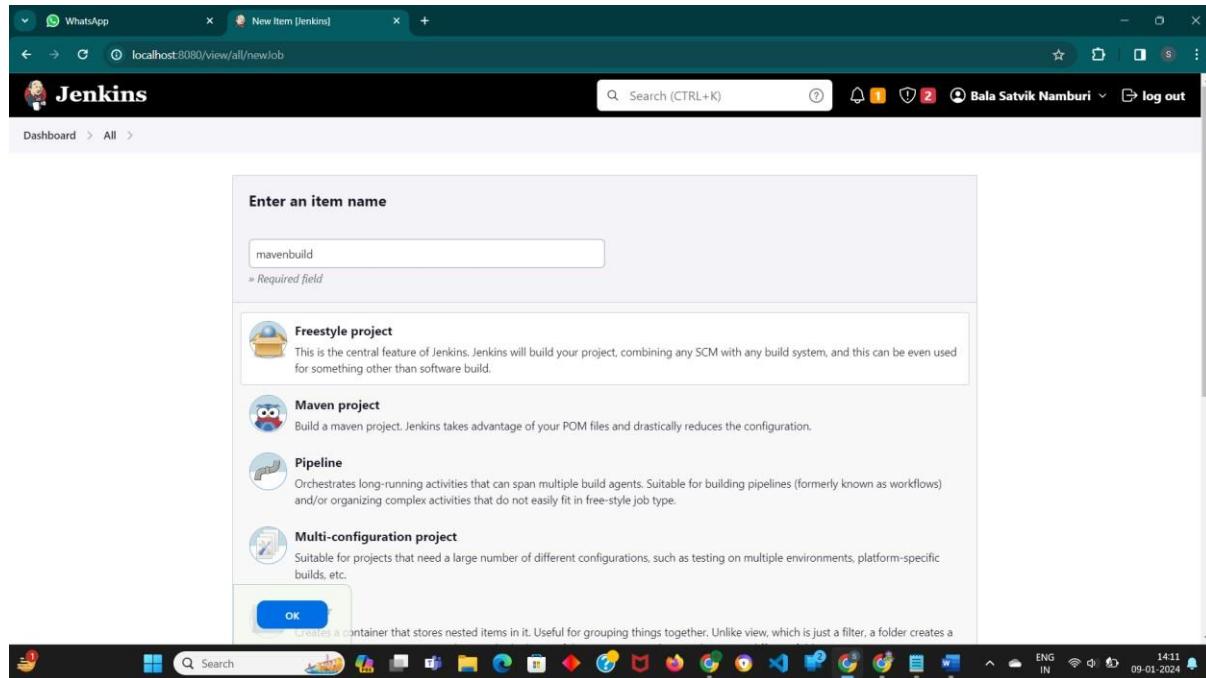
The screenshot shows the Jenkins 'Edit View [JAVAV] configure' page. It's a continuation of the previous step. The 'Build Pipeline View Title' field is empty. Under 'Pipeline Flow', the 'Layout' dropdown is set to 'Based on upstream/downstream relationship'. In the 'Upstream / downstream config' section, the 'Select Initial Job' dropdown is set to 'java_build'. At the bottom, there's a 'Trigger Options' section with a 'Build Cards' dropdown set to 'Standard build card' and two buttons: 'OK' and 'Apply'. The status bar at the bottom indicates 'REST API Jenkins 2.426.1' and the date/time '09-01-2024 14:03'.

Step 13: Click on run button.

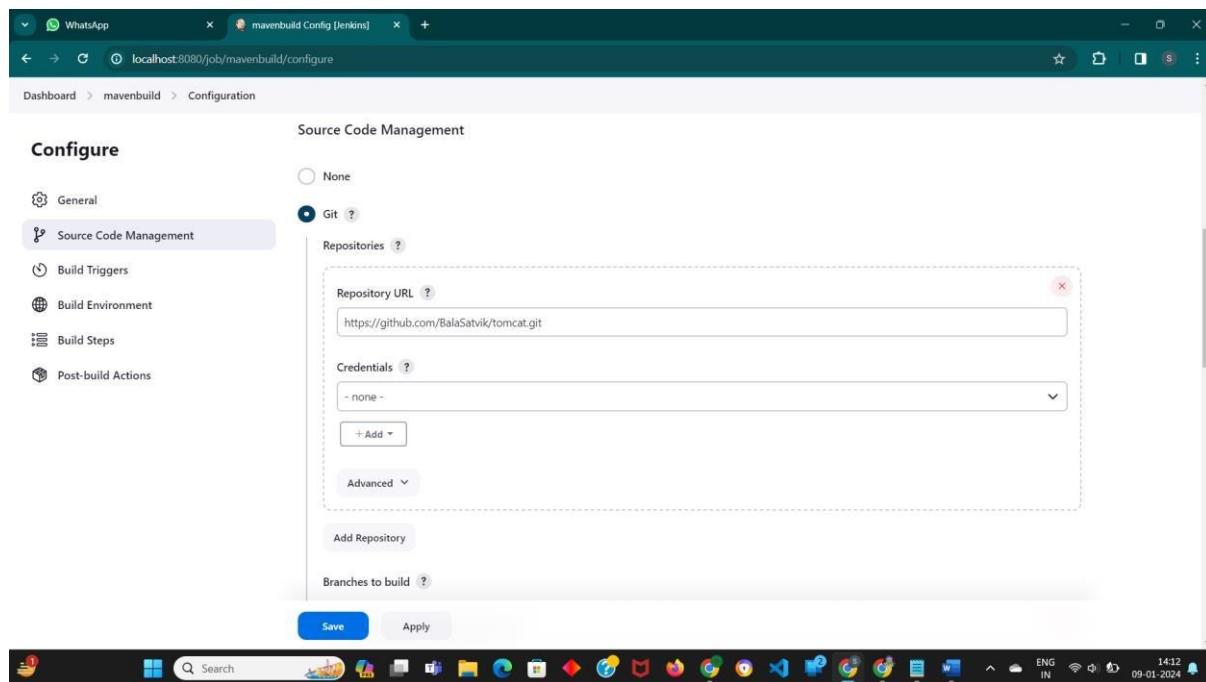


3-Pipeline of Maven-Web Project

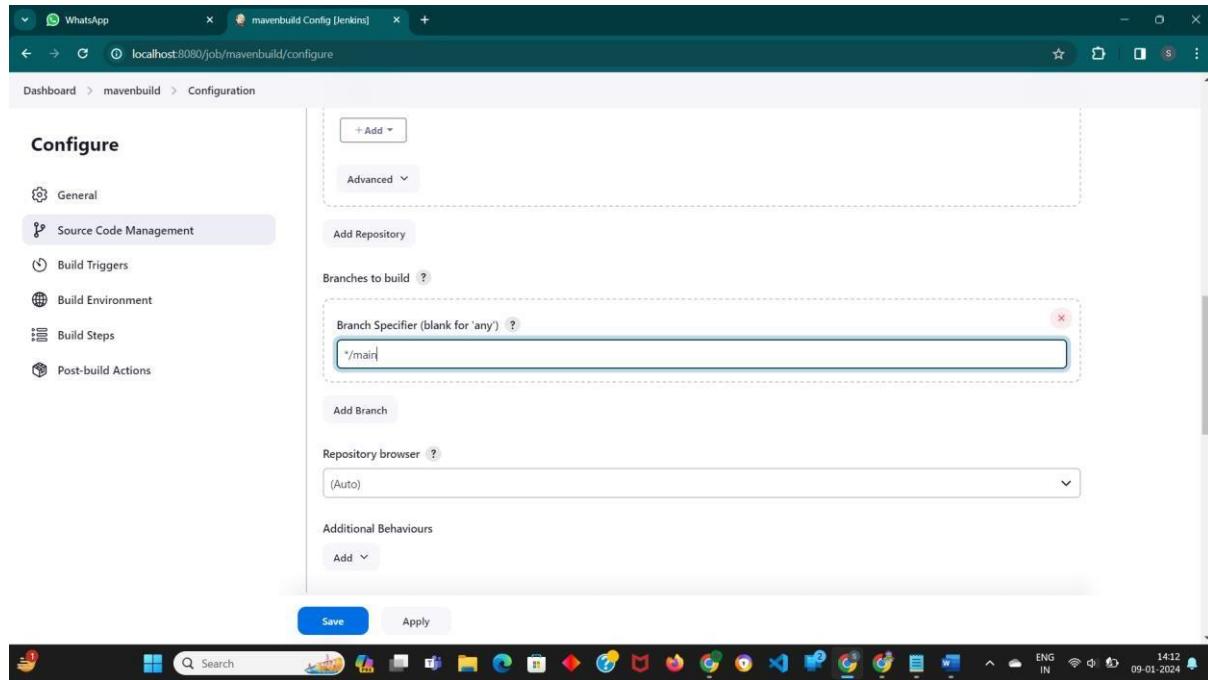
Step 1: Go to Dashboard->New Item. Then enter project name as mavenbuild, select freestyle project. Then click on OK.



Step 2: In source code management select git and paste github repository where web project is present.

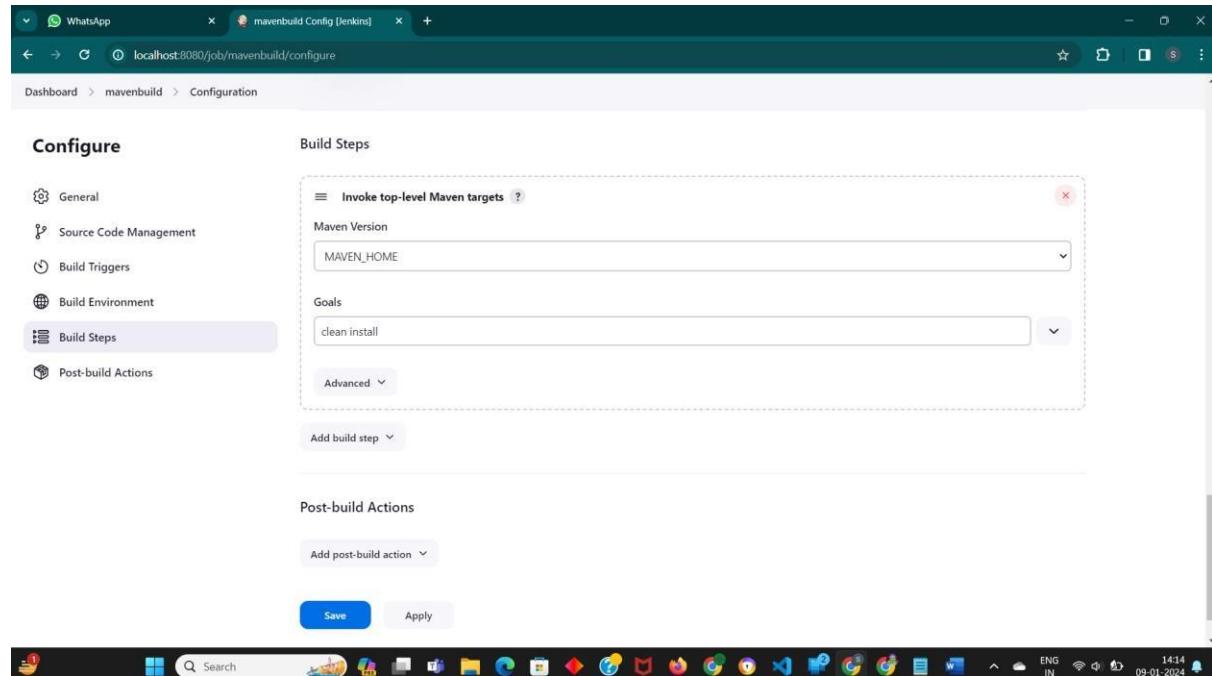


Step 3: Check branch specifier is */main or not. If not update to */main.



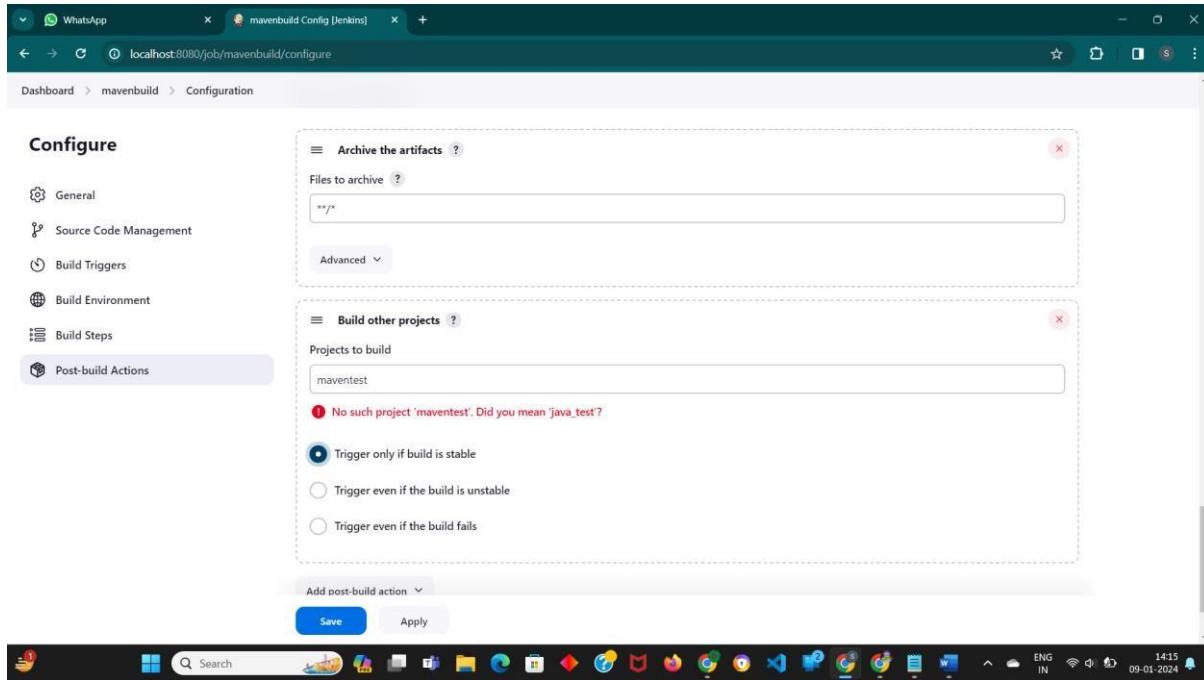
The screenshot shows the Jenkins configuration interface for a job named 'mavenbuild'. In the 'Configure' section, under 'Build Environment', the 'Branch Specifier' field contains the value '/main'. This indicates that the build will be triggered for any branch starting with 'main'.

Step 4: In build steps, click on Add build step and select “**Invoke top-level maven targets**”. In that select maven version as MAVEN_HOME and enter goals as “clean install”.

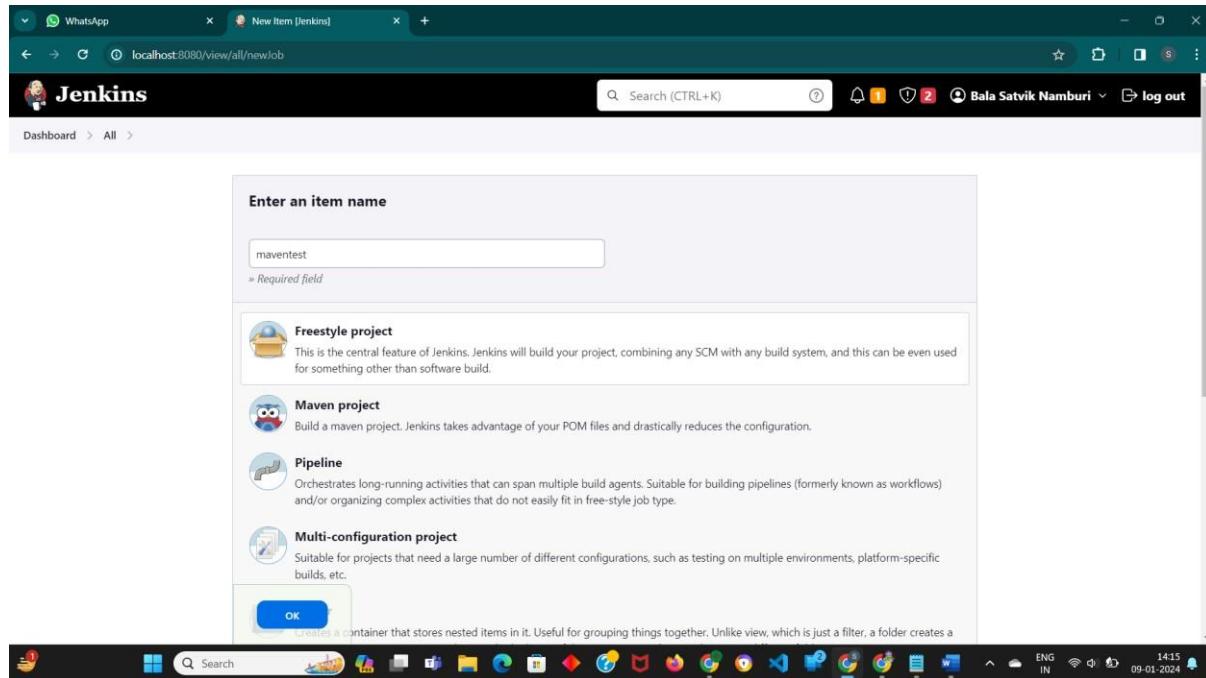


The screenshot shows the Jenkins configuration interface for the same job. In the 'Build Steps' section, there is one step selected: 'Invoke top-level Maven targets'. The 'Maven Version' dropdown is set to 'MAVEN_HOME', and the 'Goals' field contains 'clean install'. This configuration tells Jenkins to run the 'clean install' command using the Maven version specified in the environment variable 'MAVEN_HOME'.

Step 5: In post build actions click on add post build actions and select “archive the artifacts” and enter files to archive as “**/*”. Again, click on add post build actions and select “Build other projects”. Then enter the project name as “maventest” and then click on apply and save.



Step 6: Again, go to dashboard click on new item and project name as “maventest” and select as freestyle project. Then click on OK.

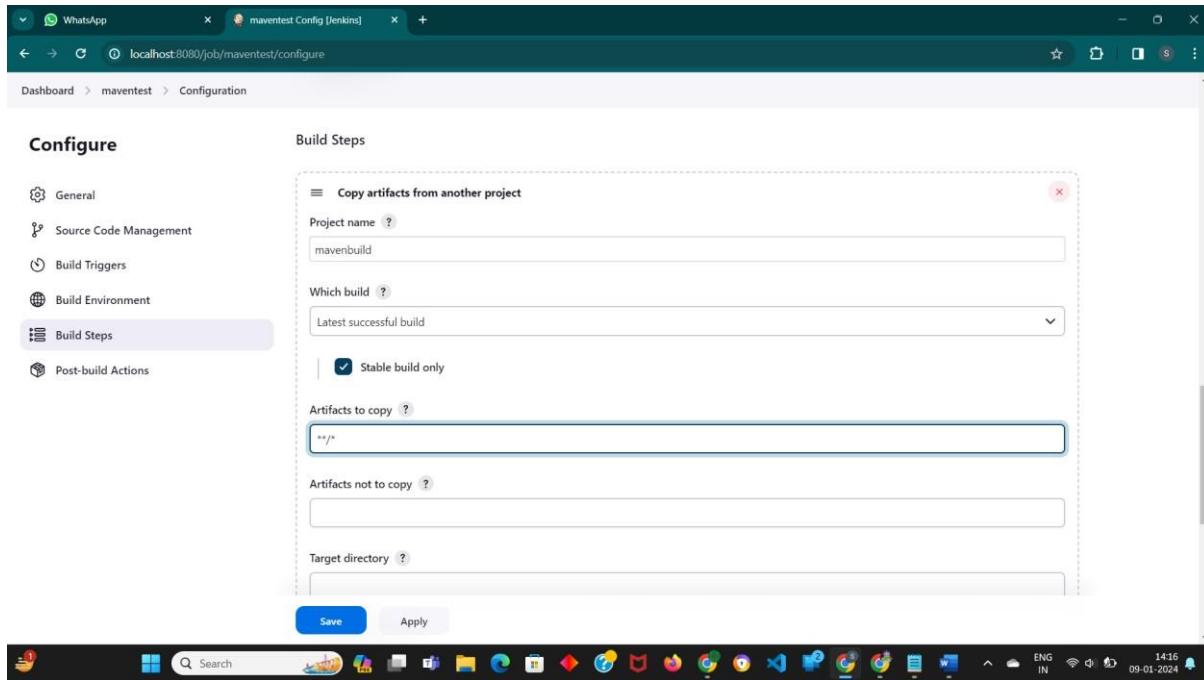


Step 7: In build environment select “delete workspace before build starts”.

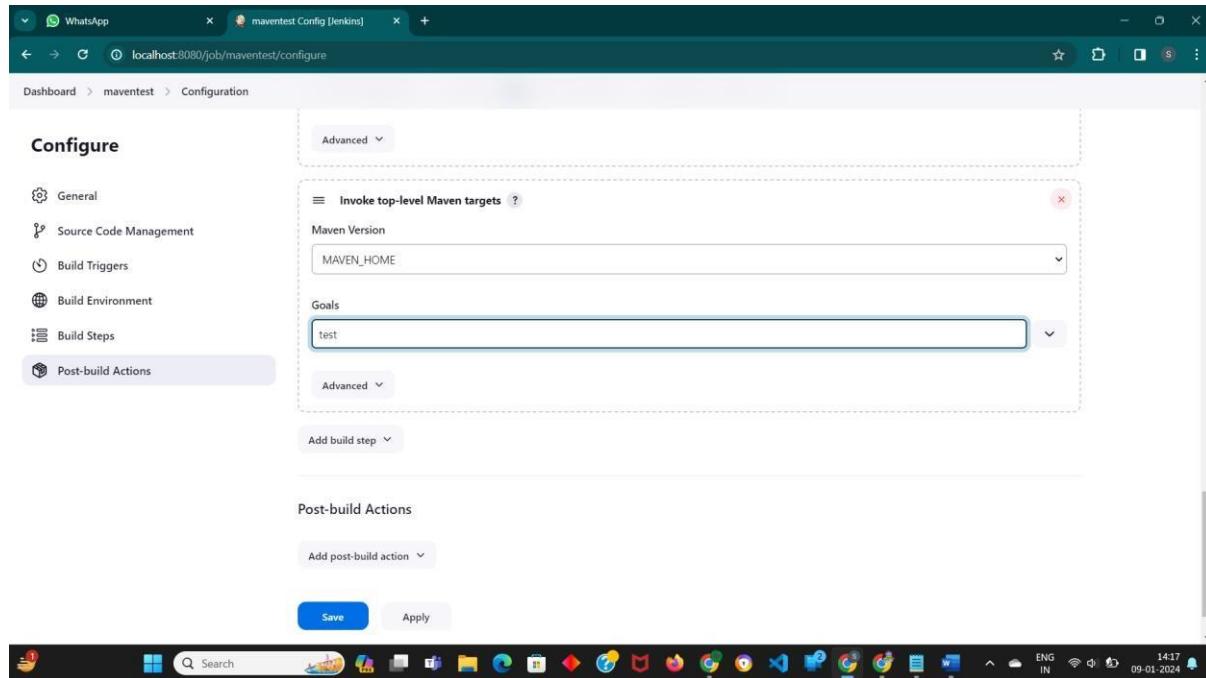
A screenshot of a web browser showing the Jenkins job configuration page for 'maventest'. The title bar says 'localhost:8080/job/maventest/configure'. The left sidebar shows 'Configure' sections: General, Source Code Management, Build Triggers (selected), Build Environment, Build Steps, and Post-build Actions. Under 'Build Triggers', the 'Delete workspace before build starts' checkbox is checked. The 'Build Environment' section contains several other unchecked checkboxes. The 'Build Steps' section has a 'Save' button at the bottom.

Step 8: In build steps click on add build step and select “copy artifacts from other projects”.

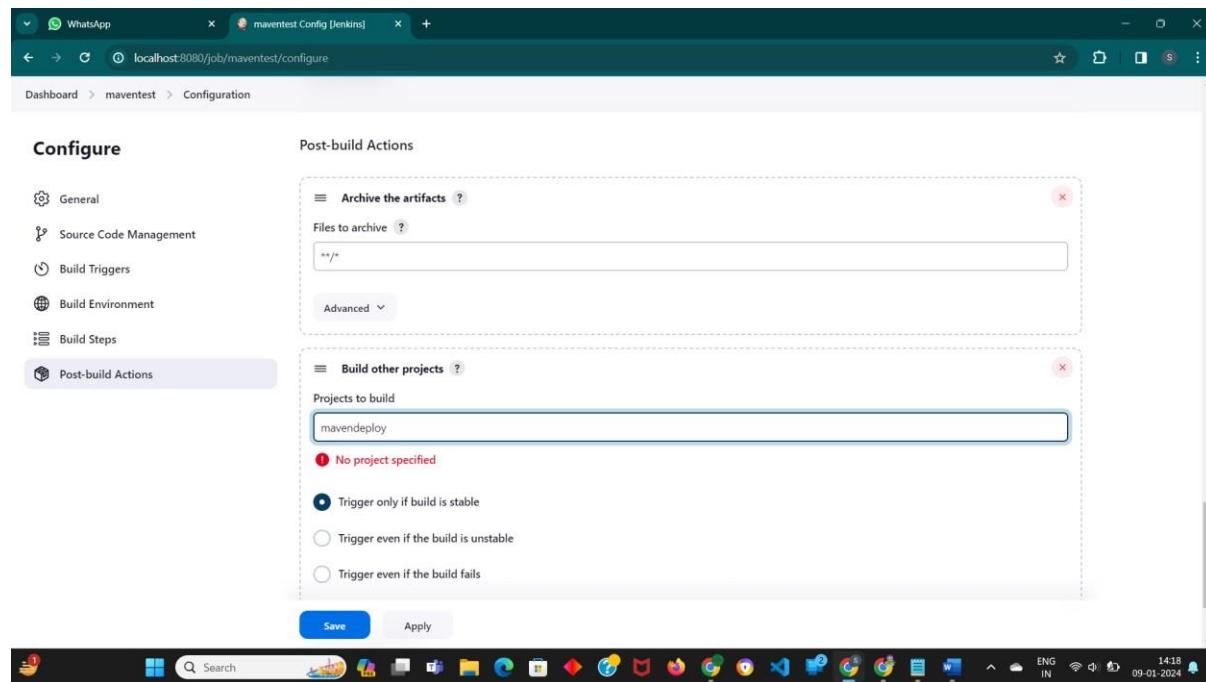
Then in Artifacts to copy enter “**/*”.



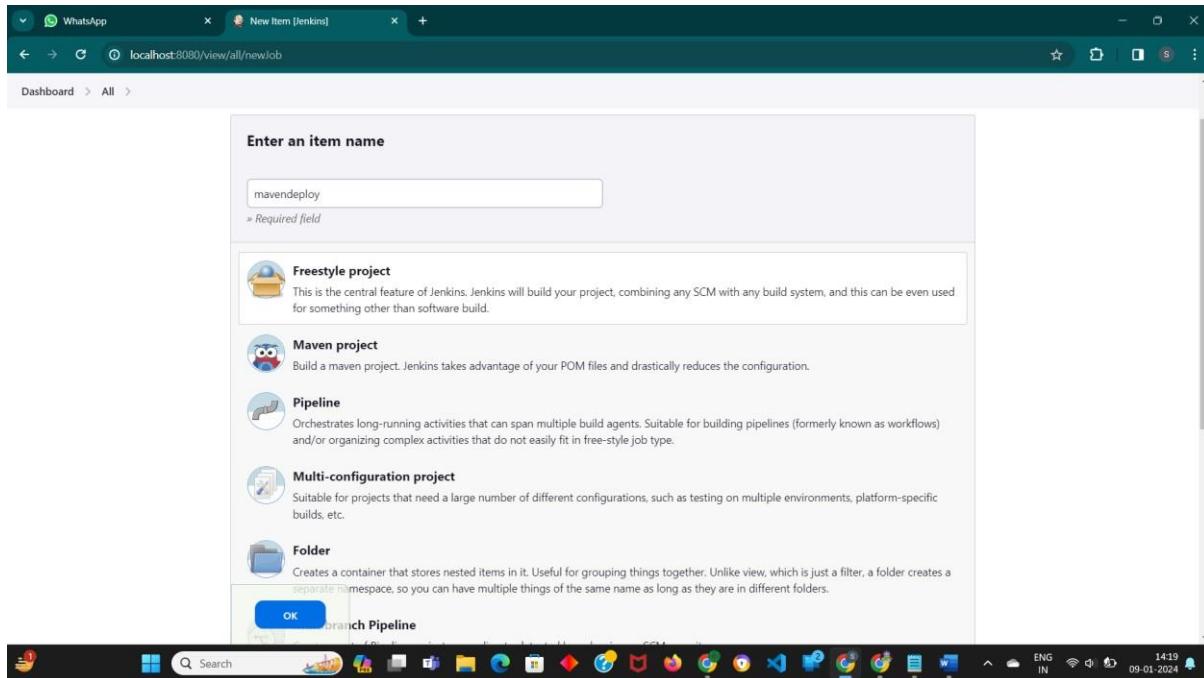
Step 9: Again, click on add build step and select “**Invoke top-level maven targets**”. Then select maven version as MAVEN_HOME and enter goals as “test” and then click on apply and save.



Step 10: In post-build actions click on add post build actions and select “archive the artifacts” and enter files to archive as “**/*”. Again, click on add post build actions and select “Build other projects”. Then enter the project name as “mavendeploy” and then click on apply and save.

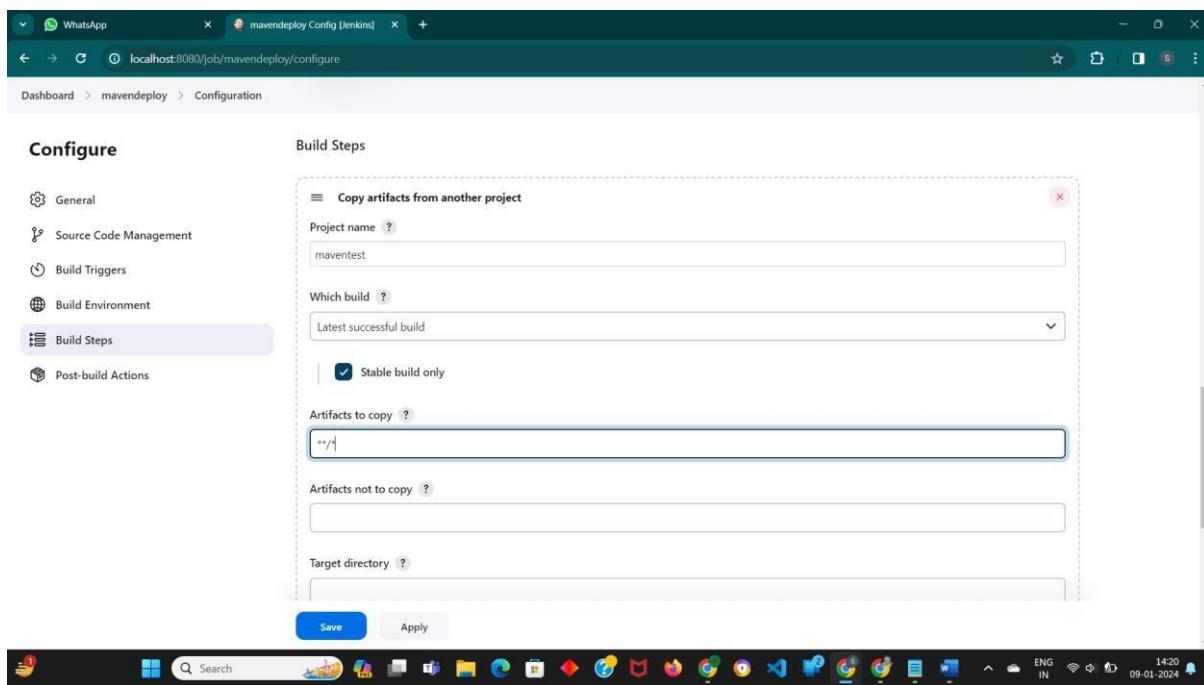


Step 11: Again, go to dashboard click on new item and project name as “mavendeploy” and select as freestyle project. Then click on OK.



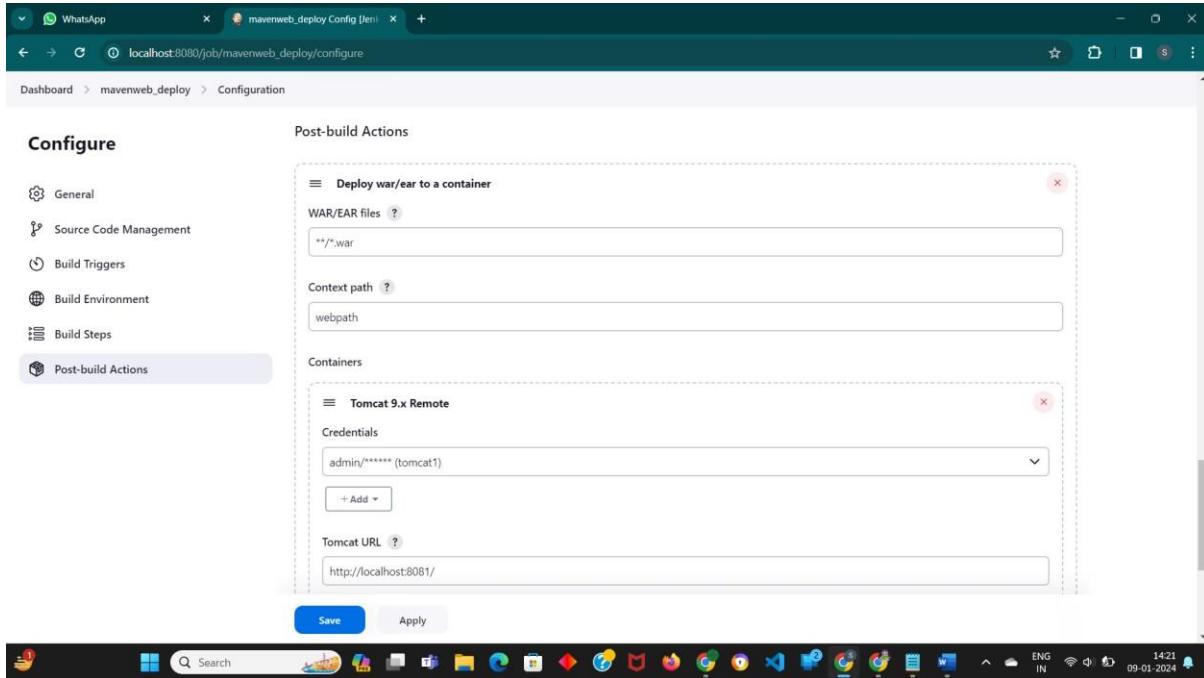
Step 12: In build steps click on add build step and select “copy artifacts from other projects”.

Then in Artifacts to copy enter “**/*”.



Step 13: In post-build actions click on add post build actions and select “**deploy war/ear to a container**” and enter war/ear files as “****/*.war**”, enter context path as **webpath**.

In containers select tomcat 9 and give tomcat url. Then click on apply and save.



Step 14: Go to dashboard and click on new view.

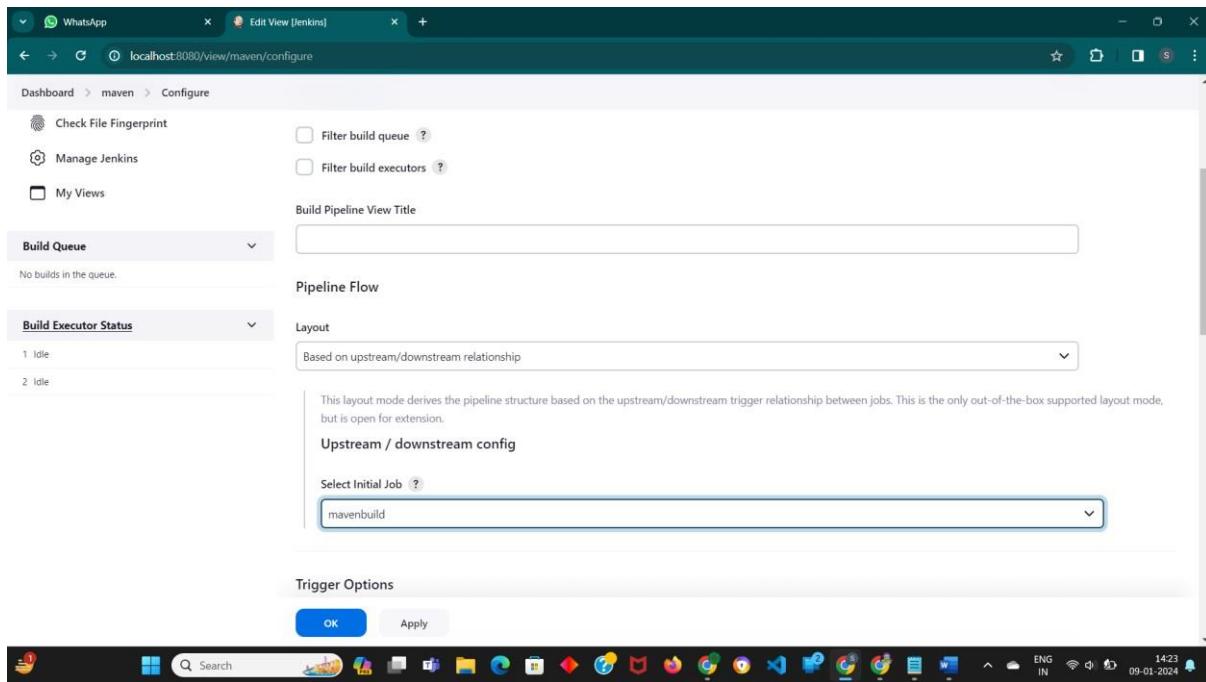
The screenshot shows the Jenkins dashboard at localhost:8080. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. Under 'Build Queue', it says 'No builds in the queue.' Under 'Build Executor Status', there are two entries: '1 Idle' and '2 Idle'. The main area displays a table of build jobs:

S	W	Name	Last Success	Last Failure	Last Duration
✓	Cloud	java_build	21 hr #3	21 hr #2	13 sec
✓	Sun	java_test	21 hr #1	N/A	7.4 sec
✓	Sun	mavenbuild	21 hr #1	N/A	12 sec
✓	Sun	mavendeploy	21 hr #1	N/A	1.9 sec
✓	Sun	mavenjava_build	23 days #1	N/A	2 min 0 sec
✓	Sun	mavenjava_test	23 days #1	N/A	5.6 sec
✓	Sun	maventest	21 hr #1	N/A	5.1 sec
✓	Sun	mavenweb_build	27 days #2	N/A	8.3 sec
✓	Sun	mavenweb_deploy	27 days #1	N/A	0.95 sec

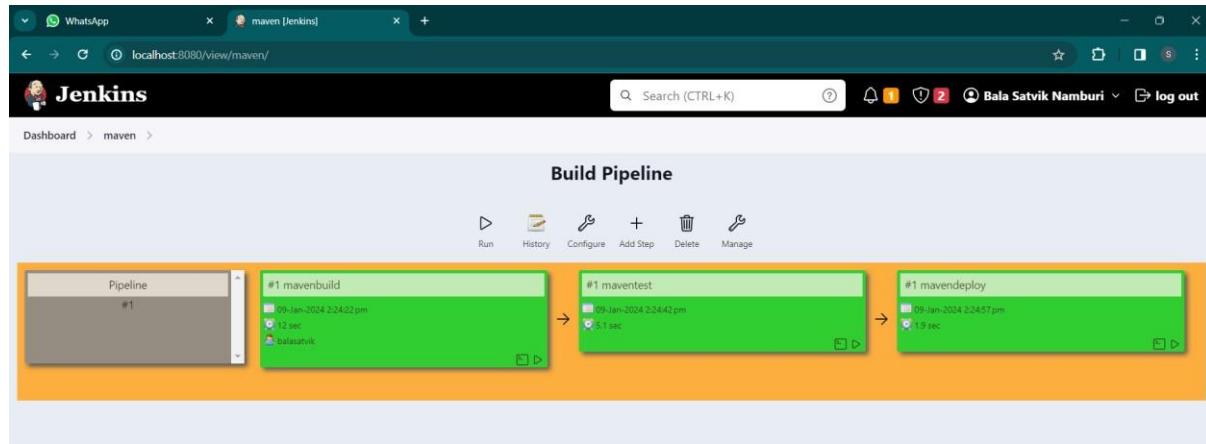
Step 15: Give name and select “build pipeline view”. Click on create.

The screenshot shows the 'New view' creation dialog in Jenkins at localhost:8080/newView. The 'Name' field is filled with 'maven'. The 'Type' section has 'Build Pipeline View' selected, with a description: 'Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.' Below it are 'List View' and 'My View' options. At the bottom right is a 'Create' button. The bottom of the screen shows the Windows taskbar with various icons and system status.

Step 16: In select initial job make sure that pipeline starts from “mavenbuild”.



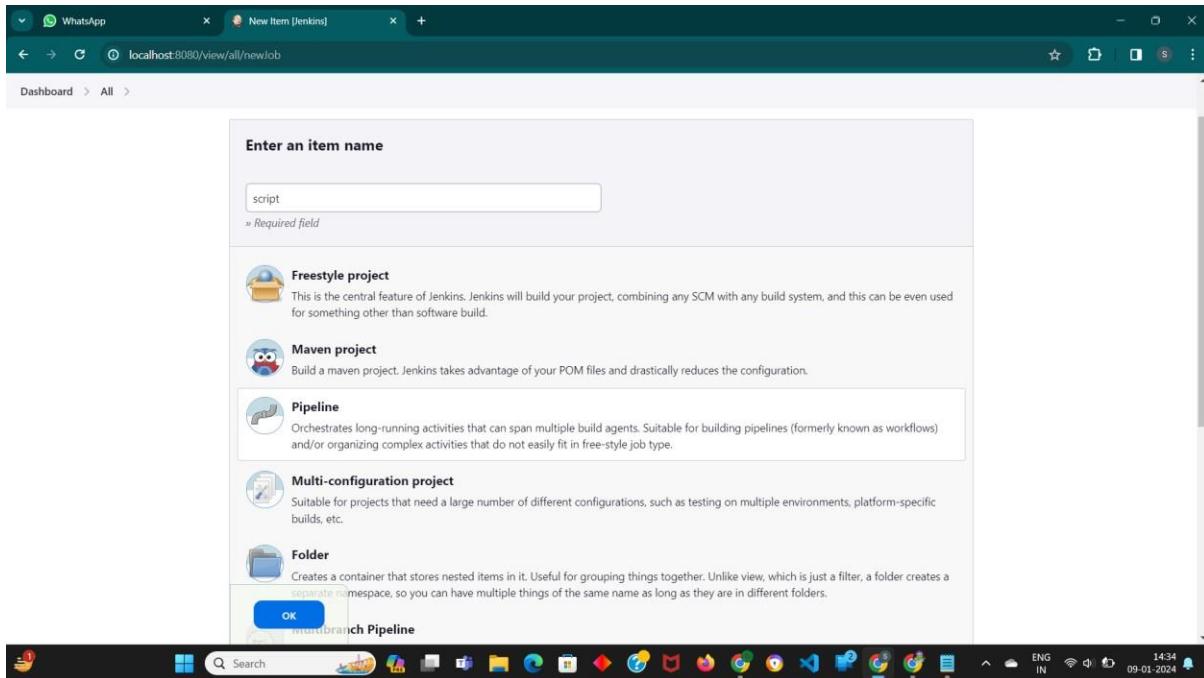
Step 17: Click on run button.



Script

Step 1: Go to Dashboard->New Item. Then enter project name as script, select pipeline.

Then click on OK.



Step 2: Write the Jenkins script. Then click apply and save.

The screenshot shows the Jenkins configuration page for the 'script' job. The title bar says 'script Config [Jenkins]'. The left sidebar shows 'General', 'Advanced Project Options', and 'Pipeline' (selected). The main area has a 'Definition' dropdown set to 'Pipeline script'. Below it is a large text area labeled 'Script' containing Groovy pipeline code. A checkbox 'Use Groovy Sandbox' is checked. At the bottom are 'Save' and 'Apply' buttons. The status bar at the bottom shows system information like battery level, signal strength, and date/time.

```
1 - pipeline {
2   agent any
3   tools{
4     maven 'MAVEN_HOME'
5   }
6   stages {
7     stage('Preparation') {
8       steps {
9         deleteDir()
10      }
11    }
12    stage('git repo & clean') {
13      steps {
14        bat "rm -rf .m2"
15        bat "git clone https://github.com/BalaSatvik/java_project.git"
16        bat "mvn clean -f java_project"
17      }
18  }
```

Step 3: Click on build now.

The screenshot shows the Jenkins interface for the 'script' job. The main content area is titled 'Stage View' and contains a table with the following data:

	Declarative: Tool Install	Preparation	git repo & clean	install	test	package
Average stage times: (Average full run time: ~39s)	368ms	1s	7s	10s	7s	7s
#2 Jan 09 14:39 No Changes	293ms	554ms	7s	10s	7s	7s
#1 Jan 09, 2024, 2:39 PM No Changes	444ms	1s	7s	10s	7s	7s

Below the table, there is a 'Build History' section with two entries:

- #2 Jan 09, 2024, 2:39 PM No Changes
- #1 Jan 09, 2024, 2:39 PM No Changes

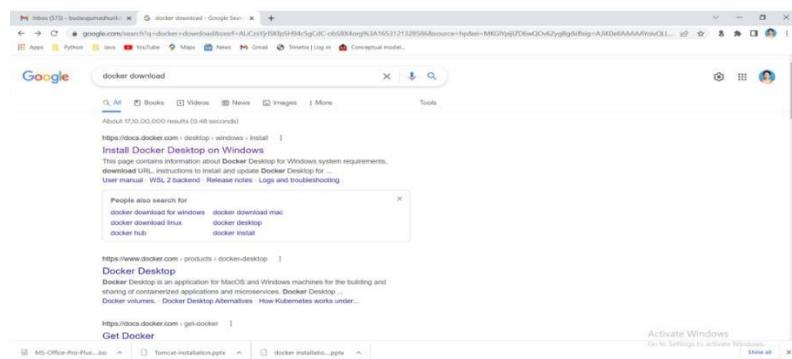
At the bottom of the page, there are 'Permalinks' and a Windows taskbar.

Experiment 8: **Working with Docker**

- a) Docker Installation
- b) Docker CLI commands
- c) Pulling an existing image from Docker Hub and running the image as container using run command
- d) Creating own image of Ubuntu and running as container, pushing the image into Docker Hub

DOCKERS INSTALLATION:

STEP-1: Download the docker from the link as shown



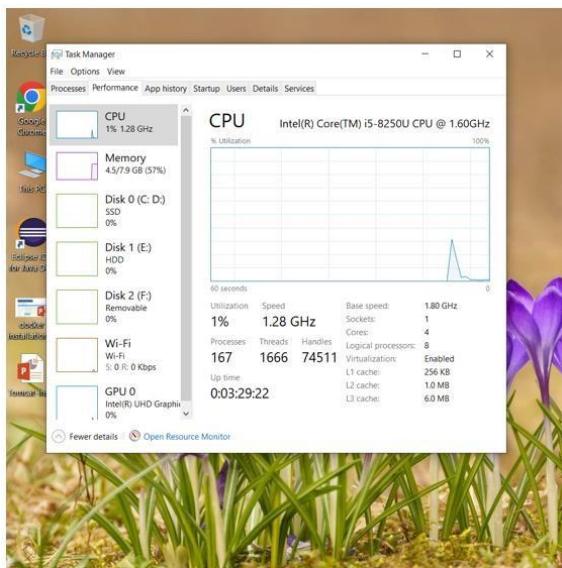
STEP-2: Click on the following link to download the executable file



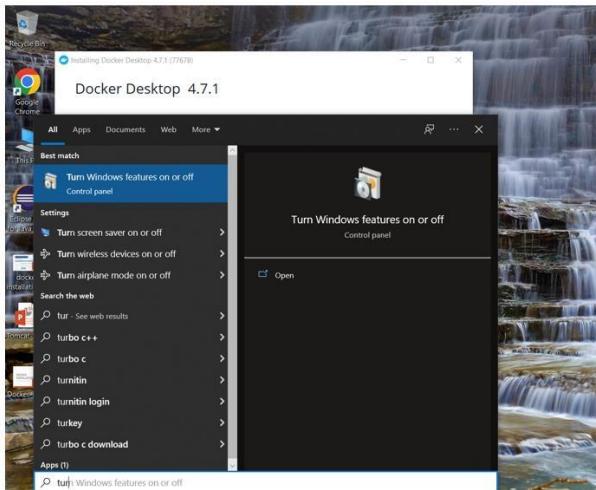
STEP-3: Type winver in Run to check the prerequisites - OS version-Windows 10 pro preferably



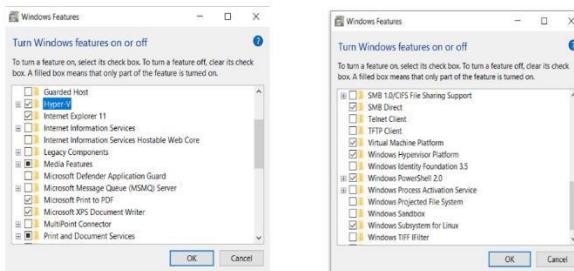
STEP-4: Make sure Virtualization is enabled in the system, we can check it in task manager as shown



STEP-5: Click on turn windows features on or off in start



STEP-6: Make sure to check the Hyper-V, Virtual Machine Platform, Windows Hypervisor Platform and Windows subsystem for Linux checkboxes and click ok



STEP-7: Download Docker for Windows

download docker desktop for windows

Videos Images Shopping News Books Maps Flights Finance

About 4,83,00,000 results (0.34 seconds)

Docker Docs https://docs.docker.com/desktop/windows-install

Install Docker Desktop on Windows

Get started with Docker for Windows. This guide covers system requirements, where to download, and instructions on how to install and update.

Run Docker Desktop for... · FAQs for Windows · Understand permission...

<https://www.docker.com/products/docker-desktop>

Docker Desktop

Docker Desktop is collaborative containerization software for developers. Get started and download Docker Desktop today on Mac, Windows, or Linux.

<https://docs.docker.com/engine/install>

Install Docker Engine

Install Docker Engine. Docker Desktop for Linux. Docker Desktop helps you build, share, and run containers on Mac and Windows as you do on Linux. Docker ...

<https://docs.docker.com/get-docker>

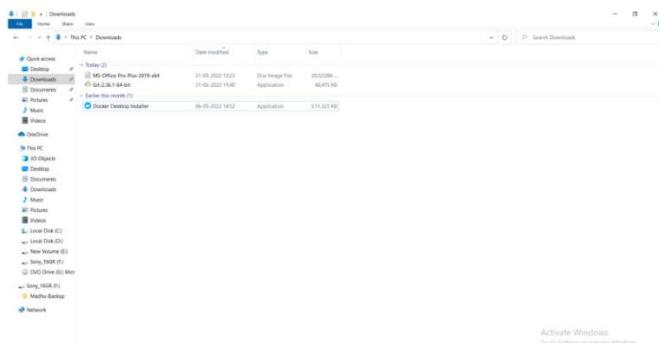
Get Docker

Download and install Docker on the platform of your choice, including Mac, Linux, or Windows.

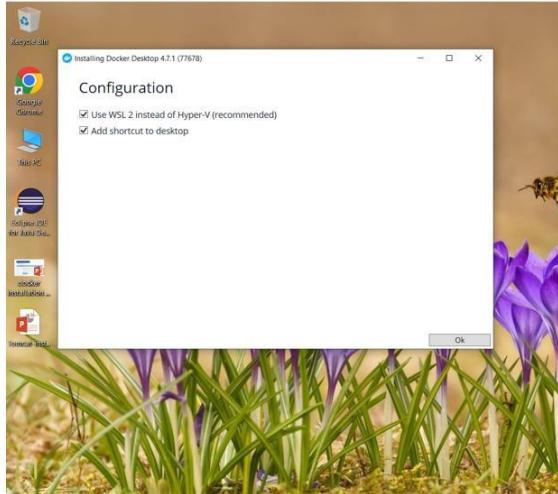
STEP-8: Installation follows as:

The screenshot shows a web browser window with the URL <https://docs.docker.com/docker-desktop/install/>. The page title is "Install Docker Desktop on Windows". It contains instructions and links for Docker Desktop for Windows, including download URLs and system requirements. A "Give Feedback" button is visible on the right.

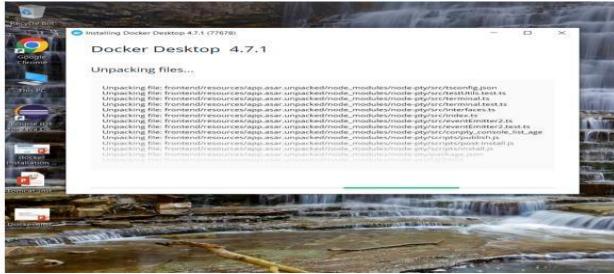
Download Docker for Windows



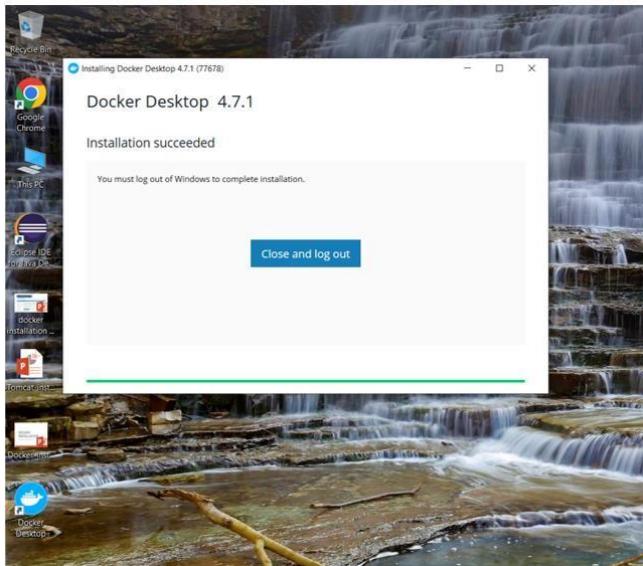
Double click the executable file and click ok as shown, make sure the following check boxes are selected



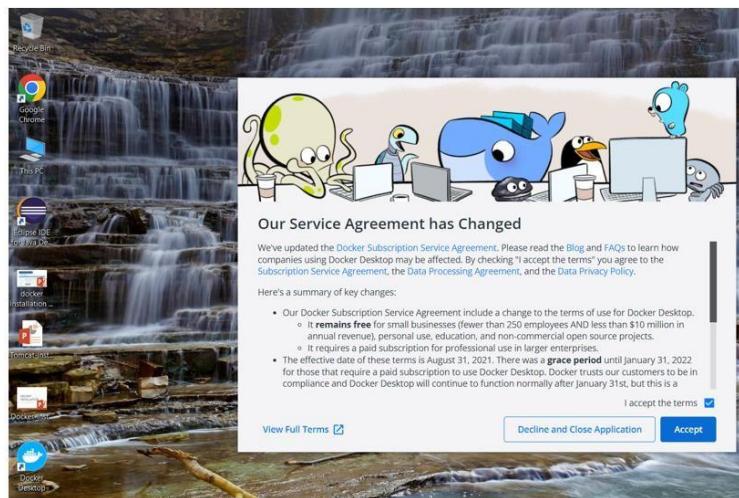
The download has started



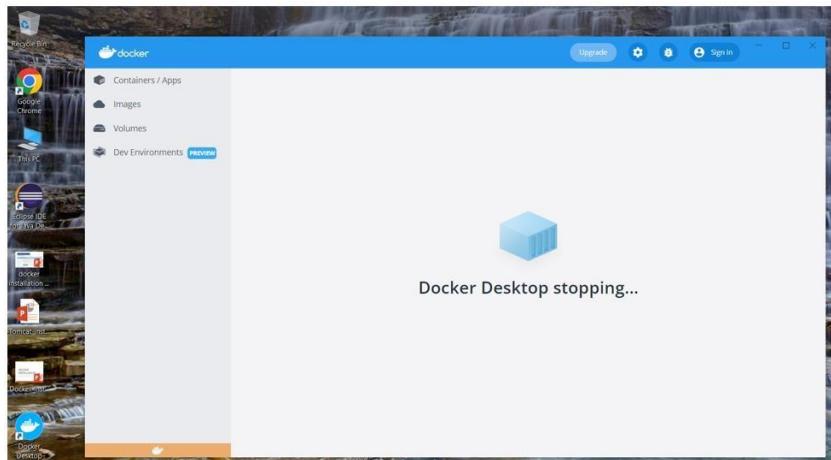
Click on close and log out



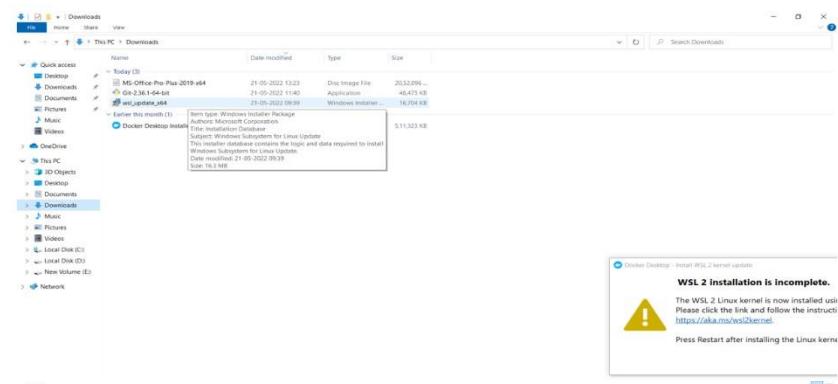
Accept the license -> click AcceptThe local Docker Hub is created



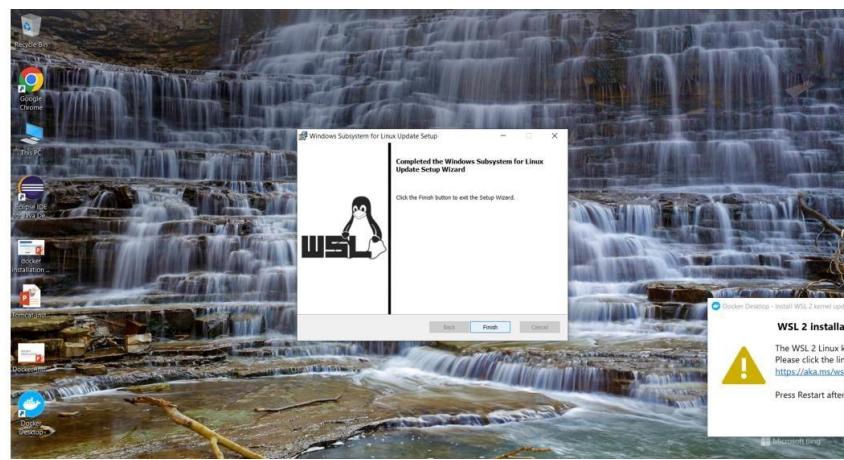
The local Docker Hub is created



Download WSL2 as shown

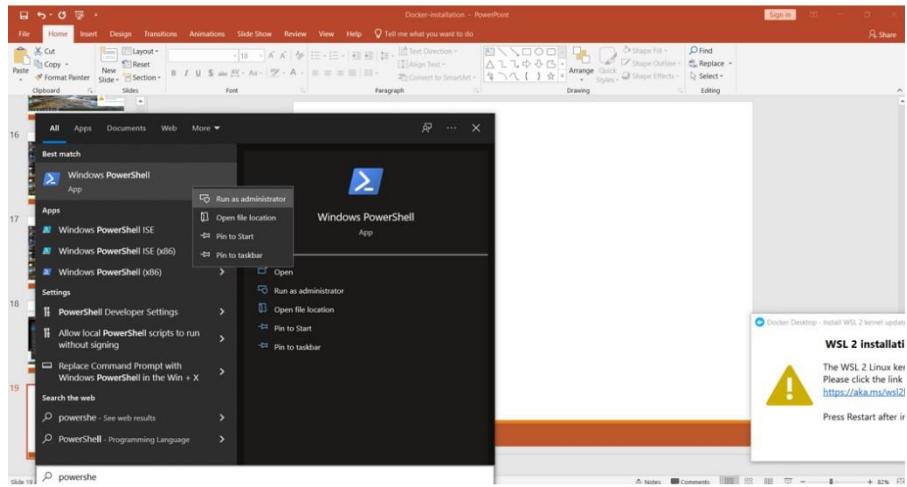


Double click the executable file and click next as shown

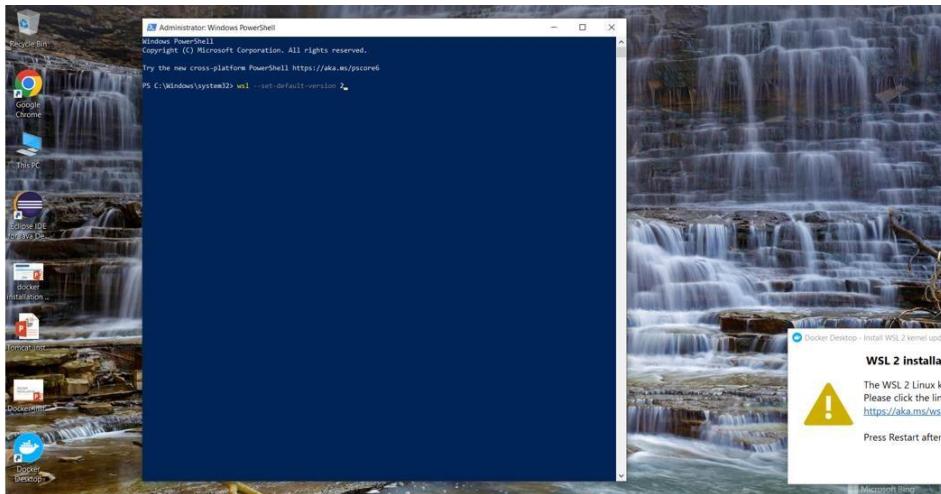


Open Windows PowerShell as shown from Start

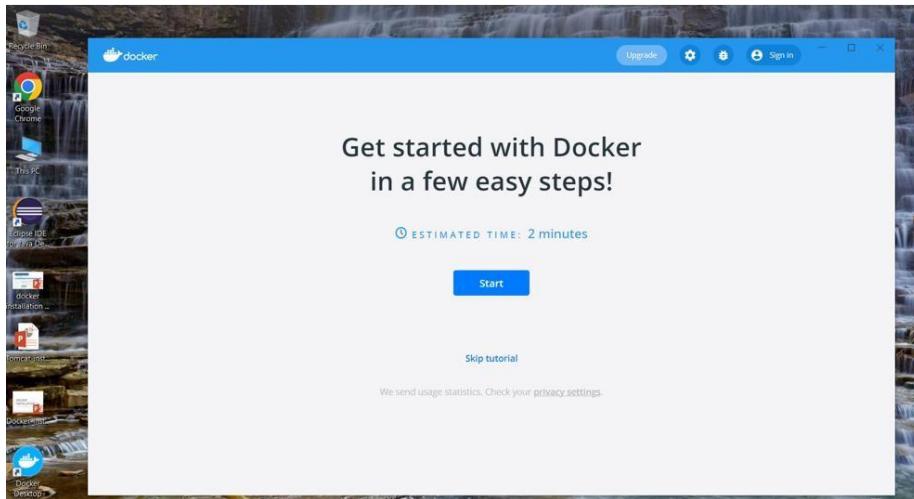
Run as administrator



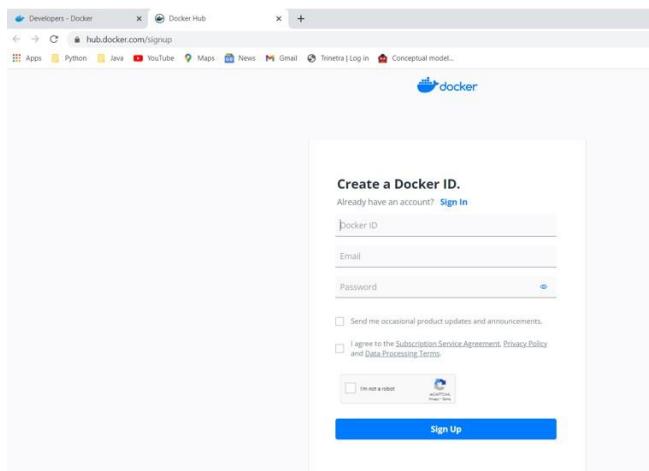
Type the following command, to set WSL as default version



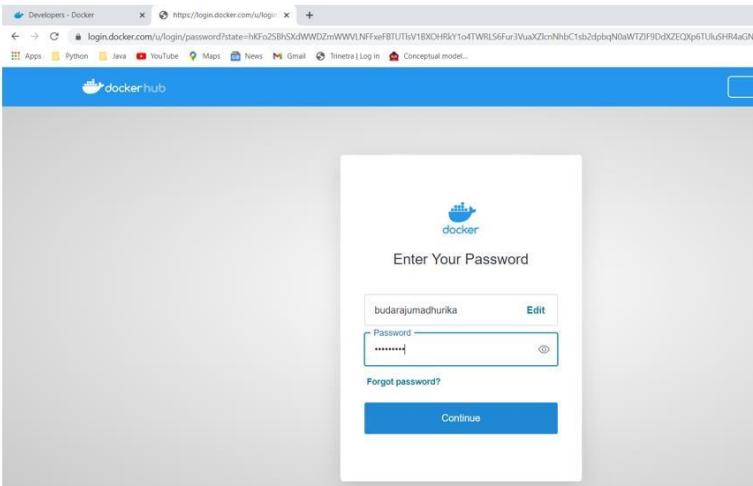
Now we can start the Docker



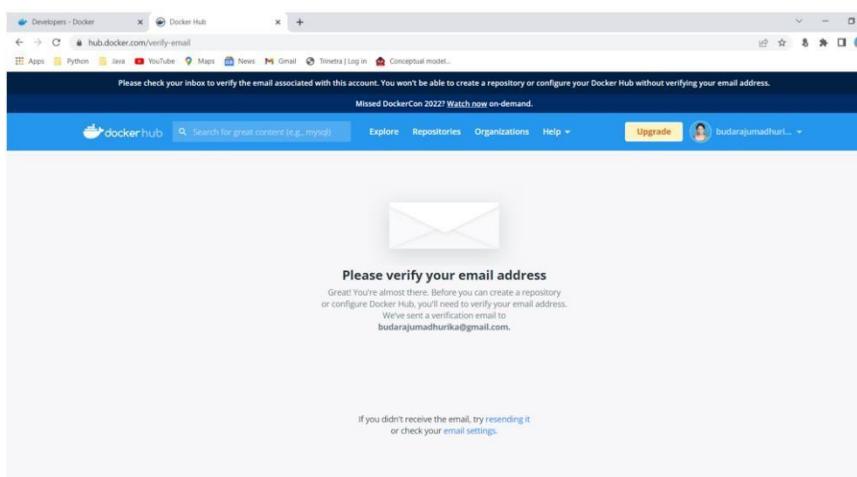
Click on hub.docker.com and signup as shown



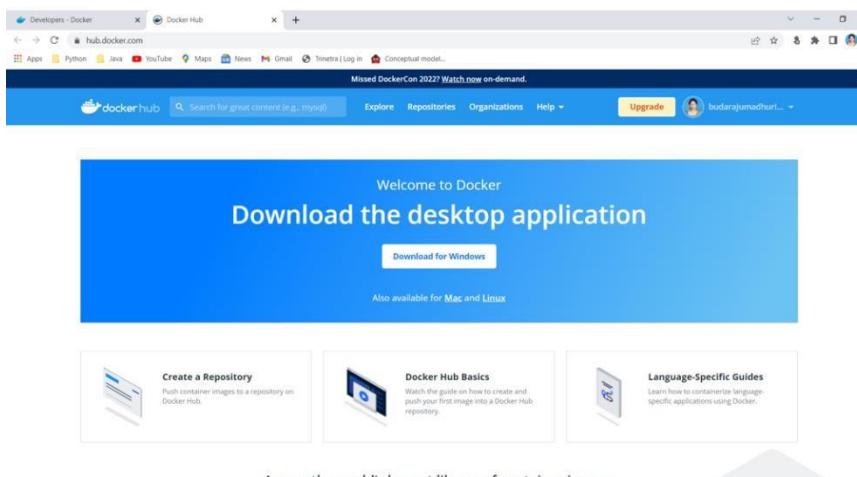
Sign In into Docker Hub



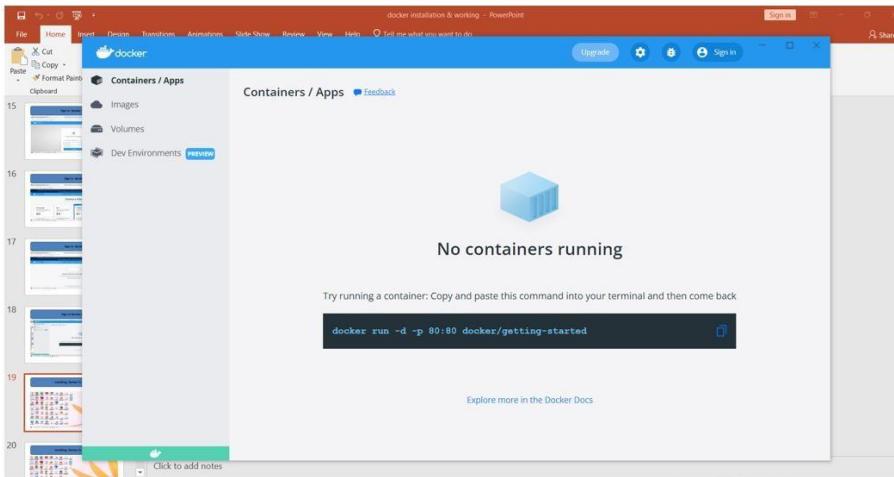
Click on personal-> continue with free



Go to Gmail and verify your account



We have successfully logged in into the Docker Hub

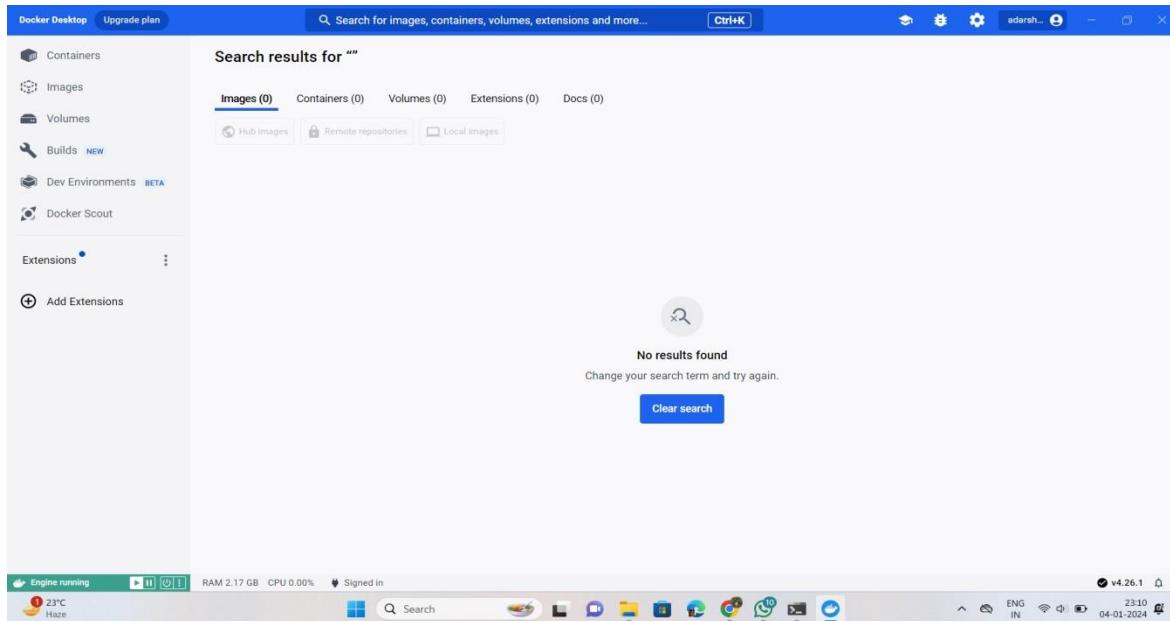


Experiment 9: **Working with Docker**

- a) Pulling and running the services, Nginx and Tomee
- b) Use Docker compose with DockerFile
- c) Use Docker compose with yaml file.

DOCKERS

->Dockers desktop will as below shown after installed



->The “**docker --version**” command shows the version number of the Docker that you are using

->The “**docker images**” command lists the images that are available on your system. It shows the repository, tag, image ID, creation date, and size of each image

->The “**docker ps -a**” command shows all the containers on your system, both running and stopped. It displays the container ID, image name, command, creation date, status, ports, and names

```
PS C:\Users\User> docker --version
Docker version 24.0.7, build afdd53b
PS C:\Users\User> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
PS C:\Users\User> docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
```

To know the docker pull command for the images we need to visit the **hub.docker.com** official site

The screenshot shows the Docker Hub search interface for the query "ubuntu". The results page displays three official Docker images:

- ubuntu**: Docker Official Image, 1B+, 10K+. Updated 3 days ago. Description: Ubuntu is a Debian-based Linux operating system based on free software. Supported architectures: Linux, ARM, ARM 64, PowerPC 64 LE, IBM Z, 386, riscv64, x86-64. Pulls: 26,457,189 (Last week). Learn more.
- websphere-liberty**: Docker Official Image, 10M+, 296. Updated 19 days ago. Description: WebSphere Liberty multi-architecture images based on Ubuntu 18.04. Supported architectures: Linux, x86-64, PowerPC 64 LE, IBM Z, ARM 64, 386. Pulls: 5,516 (Last week). Learn more.
- open-liberty**: Docker Official Image, 10M+, 62. Updated 19 days ago. Description: Open Liberty multi-architecture images based on Ubuntu 18.04. Supported architectures: Linux, x86-64, PowerPC 64 LE, IBM Z, ARM 64, 386. Pulls: 3,746 (Last week). Learn more.

On the left, there are filters for Products (Images, Extensions, Plugins), Trusted Content (Docker Official Image, Verified Publisher, Sponsored OSS), Operating Systems (Linux, Windows), Architectures (ARM), and Operating Systems (Linux, Windows).

At the bottom of the search results, a command is copied to the clipboard: `docker pull ubuntu`.

The “`docker pull`” command downloads an image from a registry, such as Docker

```
PS C:\Users\User> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
a48641193673: Pull complete
Digest: sha256:6042500cf4b44023ea1894effe7890666b0c5c7871ed83a97c36c76ae560bb9b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
Hub View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
```

Docker images show the pulled image of ubuntu

PS C:\Users\User> docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	174c8c134b2a	3 weeks ago	77.9MB

The “**docker run**” command creates and starts a new container from an image.

The **--name myubuntu** option assigns a name to the container, which can be used to identify and manage it later.

The **-it** option runs the container in an interactive terminal or console for the container.

The **ubuntu** argument specifies the image to use, which is the official Ubuntu image from Docker Hub

“ **Docker run –name myubuntu –it ubuntu**” command runs a new Ubuntu container with a terminal and a name of myubuntu.

```
PS C:\Users\User> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\User> docker run --name myubuntu -it ubuntu
root@260a58434b8c:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@260a58434b8c:/# exit
exit
PS C:\Users\User> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
260a58434b8c ubuntu "/bin/bash" 13 seconds ago Exited (0) 7 seconds ago
myubuntu

PS C:\Users\User> docker run -it ubuntu
root@a974dd2d6ef7:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@a974dd2d6ef7:/# exit
exit
PS C:\Users\User> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a974dd2d6ef7 ubuntu "/bin/bash" 11 seconds ago Exited (0) 6 seconds ago
260a58434b8c ubuntu "/bin/bash" 51 seconds ago Exited (0) 44 seconds ago
myubuntu
PS C:\Users\User> |
```

The “**docker run –it ubuntu**” command create and starts new container from ubuntu image

And runs in interactive terminal

To remove the container first we need to stop the container through command “**docker stop container id**” and then we need to remove the container by “**docker rm containerid**”

To remove the image the following command is used “**docker rmi imageid or image name**”

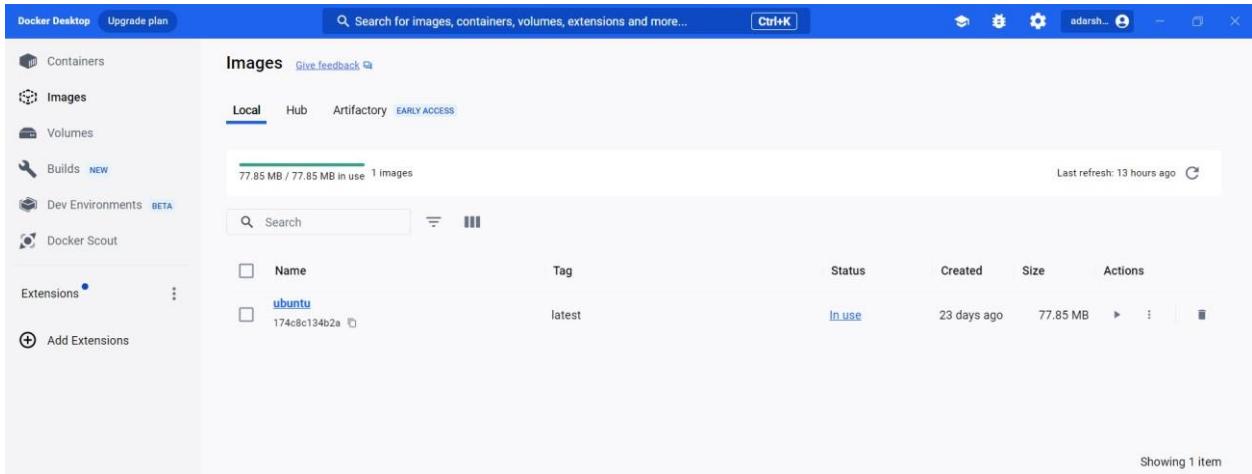
```

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

PS C:\Users\User> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest d453dd892d93 2 months ago 187MB
PS C:\Users\User> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
911e715e0810 nginx "/docker-entrypoint..." 13 hours ago Exited (255) 2 minutes ago 0.0.0.0:9090->80/tcp intelligent_albattani
PS C:\Users\User> docker stop 911e715e0810
911e715e0810
PS C:\Users\User> docker rm 911e715e0810
911e715e0810
PS C:\Users\User> docker rmi d453dd892d93
Untagged: nginx:latest
Untagged: nginx@sha256:2bd49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
Deleted: sha256:d453dd892d9357f73559b967478ae9cbc417b52de66b53142f6c16c8a275486b9
Deleted: sha256:efaf0324a701d6accf82a523dcaba9aadf21943e214a8879d10c13284bffffcd5f
Deleted: sha256:4178f7184379d32ddc77d881dca384ac307dedb0c42f521fe633c5a5f308cd6
Deleted: sha256:454122e697d0fd11338a4e00fd1d20acd2eaaf133344321160606c908ca395911
Deleted: sha256:b4ad020cdacfccfc4faf3cd7984f600391d3972063112dd2b37fdb30105993
Deleted: sha256:6f226612ab7aafdd91fcc90917ad3d7a667237a78785fe2309fa1f160559a69a7
Deleted: sha256:9671ab29815f09e9c2552b872e0097732d4b5efb5fdc91630853d7bf7221f1a
Deleted: sha256:7292cf786aa89399bca4e3edd105d3b2ee0683a46ef1f5ff436c0f9d1d49e765
PS C:\Users\User> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\User> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\User>

```

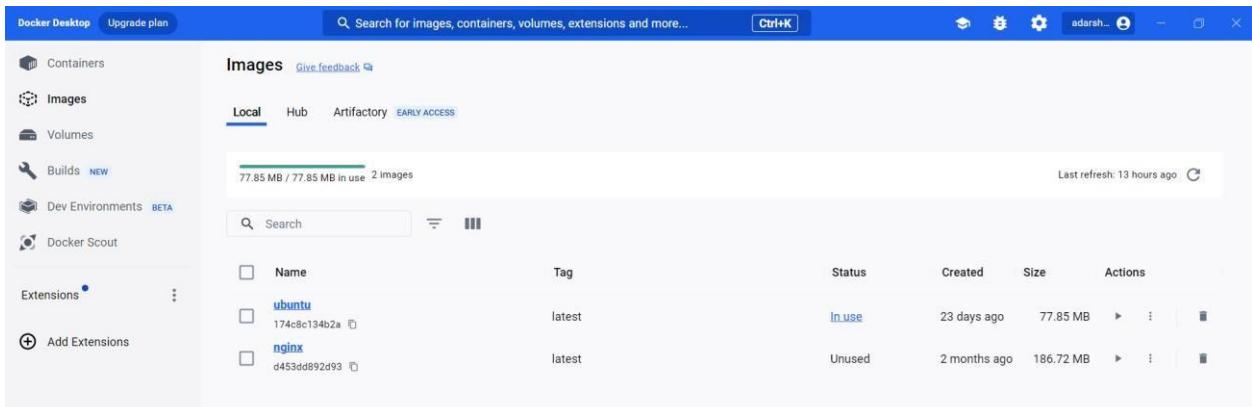
The screenshot shows a Windows desktop environment. At the top, there's a taskbar with various icons including a weather widget (23°C Haze), a search bar, and system status indicators (ENG IN, battery level, date/time). Below the taskbar is a browser window displaying the DockerHub website. The URL is 'https://hub.docker.com/r/nginx'. The page content includes a 'Quick reference' section with maintainer information and help links, a 'Supported tags and respective Dockerfile links' section, and a 'Recent Tags' sidebar. A prominent button labeled 'Pull command copied' is visible on the right side of the page.



```
PS C:\Users\User> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
af107e978371: Pull complete
336ba1f05c3e: Pull complete
8c37d2ff6efa: Pull complete
51d6357098de: Pull complete
782f1ecce57d: Pull complete
5e99d351b073: Pull complete
7b73345df136: Pull complete
Digest: sha256:2bdc49f2f28ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview nginx](#)



"docker run -d -p userportnumber:default portnumber" is used to run a container with port mapping

```
PS C:\Users\User> docker run -d -p 9090:80 nginx
563ca0a08873f1b20ca73f8fd7302f9ed57ff6225cf9b472d52ccd4aaa192bc2
PS C:\Users\User> |
```

After the succesfull mapping then we can check whether the port is mapped or not by going to browser and type
localhost:userportnumber



Welcome to nginx!

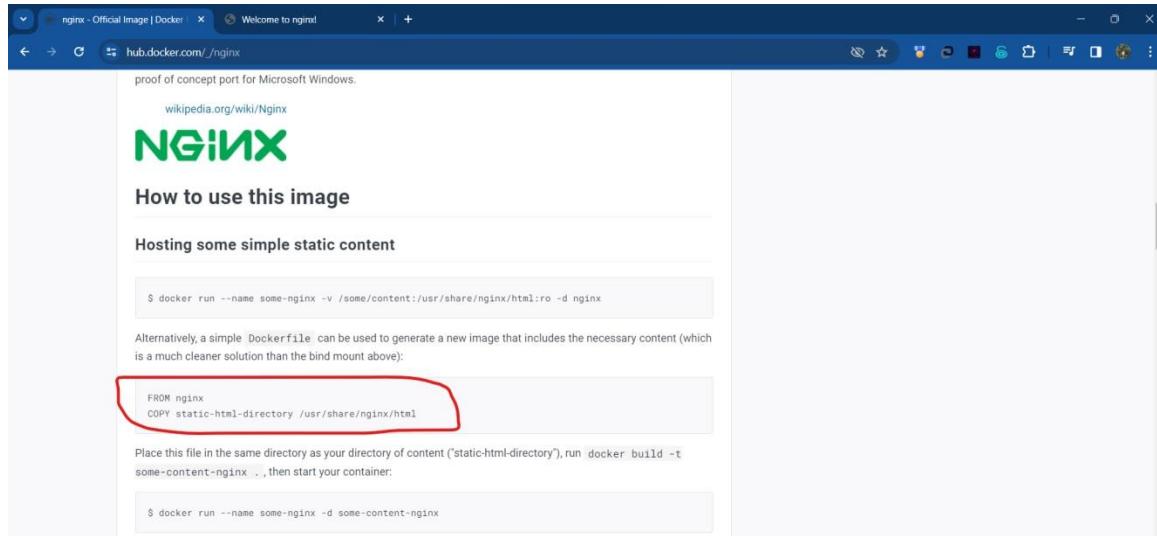
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

To get into the iterative terminal of any image:

We need to check the official image site in hub.docker.com and check for the path where the files are there ,mostly appear in **how to use this image** for this example we are going with nginx and changing the html file



The “**docker exec -it container id /bin/bash**” command runs an interactive bash shell inside a running container, The -it option attaches a terminal to the container’s standard input and output

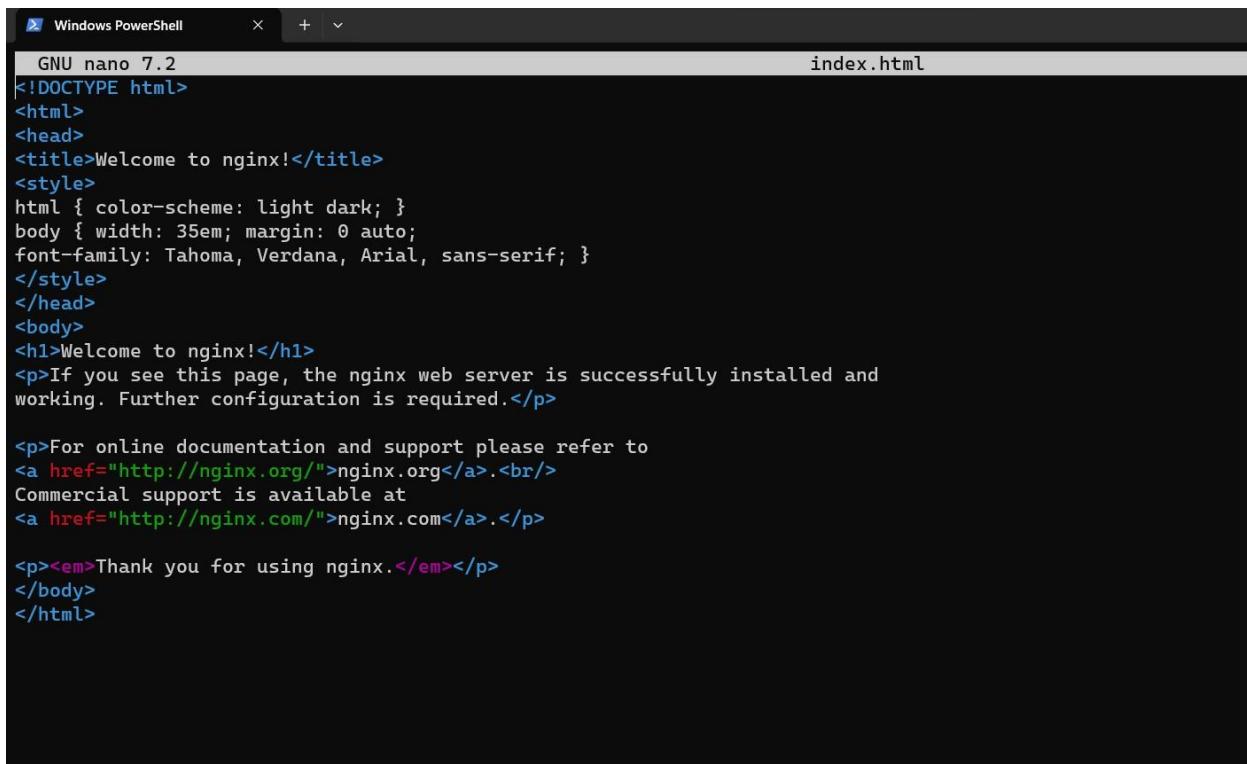
IN terminal

Change the working directory to the **usr/share.nginx/html** ,which can be copied from nginx image official site ,under how to use this image

```
PS C:\Users\User> docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
563ca0a08873        nginx              "/docker-entrypoint..."   3 minutes ago      Up 3 minutes          0.0.0.0:9090->80/tcp   interesting_rosalind
a974dd2d6ef7        ubuntu              "/bin/bash"            24 minutes ago     Exited (0) 24 minutes ago   blissful_beaver
260a58434b8c        ubuntu              "/bin/bash"            24 minutes ago     Exited (0) 24 minutes ago   myubuntu

PS C:\Users\User> docker exec -it 563ca0a08873 /bin/bash
root@563ca0a08873:~# ls
bin  dev  docker-entrypoint.sh  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot docker-entrypoint.d  etc    lib    lib64  media  opt  root  sbin  sys  usr
root@563ca0a08873:~# cd /usr/share/nginx/html
root@563ca0a08873:/usr/share/nginx/html# sudo nano index.html
bash: sudo: command not found
root@563ca0a08873:/usr/share/nginx/html# nano index.html
bash: nano: command not found
root@563ca0a08873:/usr/share/nginx/html# apt-get update -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [12.7 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [134 kB]
Fetched 9185 kB in 9s (1078 kB/s)
Reading package lists... Done
root@563ca0a08873:/usr/share/nginx/html# apt-get install nano -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libcursesw6
Suggested packages:
  gpm hunspell
The following NEW packages will be installed:
  libgpm2 libcursesw6 nano
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 837 kB of archives.
After this operation, 3339 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libcursesw6 amd64 6.4-4 [134 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 nano amd64 7.2-1 [689 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libgpm2 amd64 1.20.7-10+b1 [14.2 kB]
Fetched 837 kB in 1s (1257 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libcursesw6:amd64.
(Reading database ... 7590 files and directories currently installed.)
Preparing to unpack .../libcursesw6_6.4-4_amd64.deb ...
Unpacking libcursesw6:amd64 (6.4-4) ...
Selecting previously unselected package nano.
Preparing to unpack .../archives/nano_7.2-1_amd64.deb ...
Unpacking nano (7.2-1) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../libgpm2_1.20.7-10+b1_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-10+b1) ...
Setting up libgpm2:amd64 (1.20.7-10+b1) ...
Setting up libcursesw6:amd64 (6.4-4) ...
Setting up nano (7.2-1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link gr
```

```
root@563ca0a08873:/usr/share/nginx/html# nano index.html
```



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "nano index.html" being run. The content of the file "index.html" is displayed in the terminal. The file contains an HTML document with a title, a style block, and several paragraphs of text, including links to nginx.org and nginx.com.

```
GNU nano 7.2                               index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p><a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Change the content of html file as you wish

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

root@563ca0a08873:/usr/share/nginx/html# nano index.html
root@563ca0a08873:/usr/share/nginx/html# exit
exit
PS C:\Users\User> |
```



Welcome to kmit(keshav memorial institute of tech)

If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.

For online documentation and support please refer to nginx.org.

Commercial support is available at nginx.com.

Thank you for using nginx.

```
PS C:\Users\User> mkdir itb

Directory: C:\Users\User

Mode                LastWriteTime         Length Name
----                -----          ---- - 
d----        04-01-2024      23:45            itb

PS C:\Users\User> cd itb
PS C:\Users\User\itb> vim index.html
vim : The term 'vim' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or
if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ vim index.html
+ ~~~
+ CategoryInfo          : ObjectNotFound: (vim:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

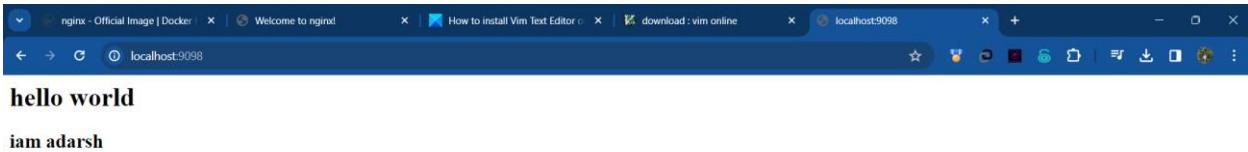
```
PS C:\Users\User\itb\vim\src> vim index.html
```

A screenshot of a terminal window. The top half shows a Vim session editing an HTML file named 'index.html'. The code contains an H1 tag with 'hello world' and an H2 tag with 'iam adarsh'. The bottom half shows a command line interface with the following text:

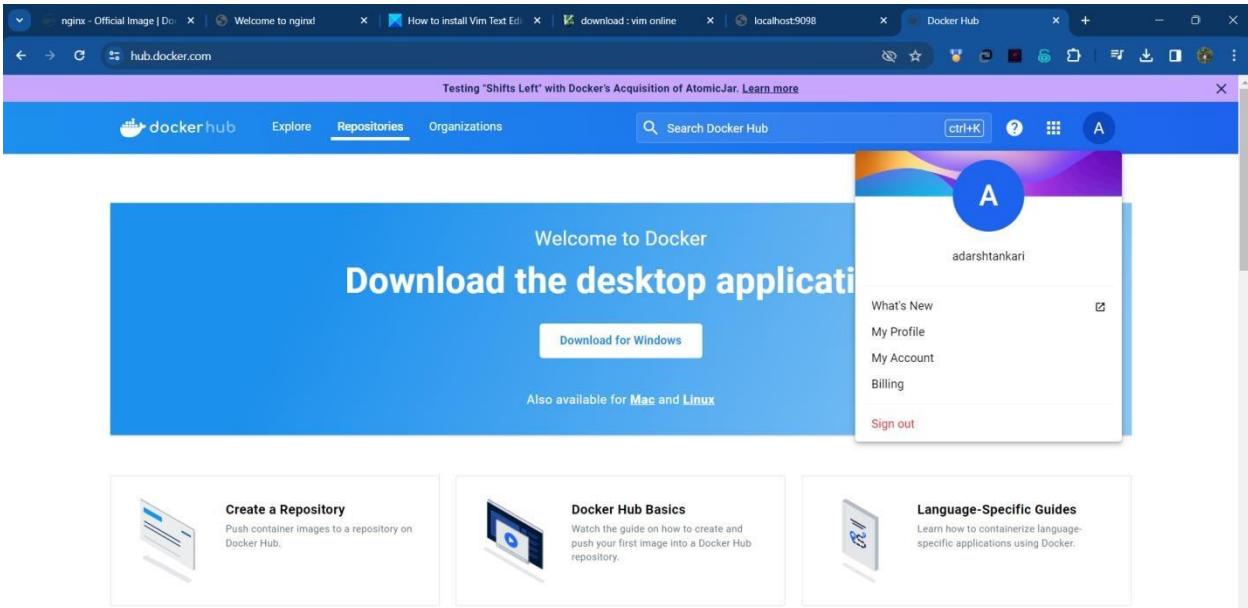
```
PS C:\Users\User\itb\vim\src> vim Dockerfile
```



```
PS C:\Users\User\itb\vim\src> docker run -d -p 9098:80 04nginx  
d282862aae77eb3595e220cd1d4f46b9a5f2de36449b9d215781d70b9ae8e81a  
PS C:\Users\User\itb\vim\src> |
```



Pushing the container to dockerhub repositories:



“**Docker tag**” is a way to label a Docker image with a name and a version, so that it can be easily identified and managed.

The “**docker tag 04nginx adarshtankari/04nginx**” command creates a tag for the local image 04nginx with the name adarshtankari/04nginx.

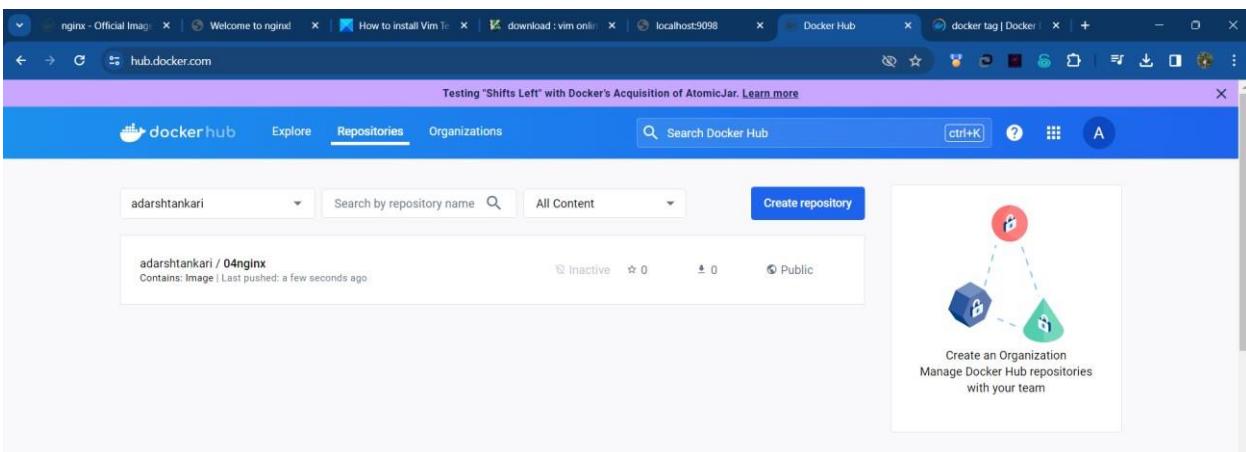
The tag also allows the image to be pushed to a registry, such as Docker Hub, under the namespace **adarshtankari**

“**Docker push**” is a command that uploads a local Docker image to a remote registry, such as Docker hub

Before pushing an image, **you need to tag** it with the registry name and login to the registry if required

```
PS C:\Users\User\itb\vim\src> docker tag 04nginx adarshtankari/04nginx
PS C:\Users\User\itb\vim\src> docker push adarshtankari/04nginx
Using default tag: latest
The push refers to repository [docker.io/adarshtankari/04nginx]
59c42526c6cd: Pushed
b074db3b55e1: Mounted from library/nginx
e50c68532c4a: Mounted from library/nginx
f6ba584ca3ec: Mounted from library/nginx
01aaa195cdad: Mounted from library/nginx
2a13e6a7cca6: Mounted from library/nginx
370869eba6e9: Mounted from library/nginx
7292cf786aa8: Mounted from library/nginx
latest: digest: sha256:fe0e0b7ef8e6098007b893533129e9d9437d75a5fac1ab3c589a5e2eb29c0793 size: 1985
PS C:\Users\User\itb\vim\src> |
```

After push we can see the local docker image in remote registry



DOCKER COMPOSE(multi container application)

```
PS C:\Users\User> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
04nginx             latest   f28c8936f605  19 hours ago  187MB
adarshankari/04nginx latest   f28c8936f605  19 hours ago  187MB
ubuntu              latest   174c8c134b2a   3 weeks ago   77.9MB
nginx               latest   d453dd892d93  2 months ago  187MB
PS C:\Users\User> docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED          STATUS           PORTS          NAMES
d282862aae77      04nginx            "/docker-entrypoint..."   19 hours ago    Exited (255)  About a minute ago   0.0.0.0:9098->80/tcp   youthful_napier
563ca0a08873      nginx              "/docker-entrypoint..."   19 hours ago    Exited (255)  About a minute ago   0.0.0.0:9090->80/tcp   interesting_rosalind
a974dd2d6ef7      ubuntu              "/bin/bash"            19 hours ago    Exited (0)   19 hours ago
260a58434b8c      ubuntu              "/bin/bash"            19 hours ago    Exited (0)   19 hours ago
myubuntu
```

The command “**vim docker-compose.yml**” opens the file docker-compose.yml in the vim text editor. This file is used to define and run multiple Docker containers as a single service.

```
PS C:\Users\User> vim docker-compose.yml
```

```
version: '3.1'
services:
  backend:
    image: mysql:5
    environment:
      MYSQL_ROOT_PASSWORD: mmrraa2003
  frontend:
    image: wordpress
    ports:
      - '5050:80'
  links:
    - backend:mysql
```

The “**docker-compose up -d**” command is used to create and start containers for a service in the background, as defined in the ***docker-compose.yml file***.

```
PS C:\Users\User> docker-compose up -d
[*] Running 34/34
  ✓ frontend 21 layers [████████████████████████████████████████████████]
    ✓ af107e978371 Already exists
    ✓ 6488d4ad61d2 Pull complete
    ✓ 95f5176ce8b8 Pull complete
    ✓ 0ebe7ec824c4 Pull complete
    ✓ 673e01769ec9 Pull complete
    ✓ 74f0c50b3097 Pull complete
    ✓ 1a19a72eb529 Pull complete
    ✓ 214daec82f86 Pull complete
    ✓ 9b52d2d65a2f44 Pull complete
    ✓ b89d85dfd9f0 Pull complete
    ✓ 991e05662acd Pull complete
    ✓ 1106a3b879f1 Pull complete
    ✓ 94fcfc71d4a9 Pull complete
    ✓ 0910769091bd Pull complete
    ✓ 2e1351a2f941 Pull complete
    ✓ 006bd105f188 Pull complete
    ✓ 0ad2b9a7aaef9 Pull complete
    ✓ e95473a7f69c Pull complete
    ✓ b7d03c854f68 Pull complete
    ✓ ad70c25cb0f6 Pull complete
    ✓ dcace32866be Pull complete
  ✓ backend 11 layers [████████████████████]
    ✓ 20e4dcdae4c69 Pull complete
    ✓ 1c56c3d4fce7f Pull complete
    ✓ e9f03a1c24ce Pull complete
    ✓ 68c3898c2015 Pull complete
    ✓ 6b95a940e7b6 Pull complete
    ✓ 90986bb8de6e Pull complete
    ✓ ae71319cb779 Pull complete
    ✓ ffc89e9df088 Pull complete
    ✓ 43d05e932198 Pull complete
    ✓ 064b2d298fb1 Pull complete
    ✓ df9a4d85569b Pull complete
[*] Running 3/3
  ✓ Network user_default      Created
  ✓ Container user-backend-1  Started
  ✓ Container user-frontend-1 Started
```

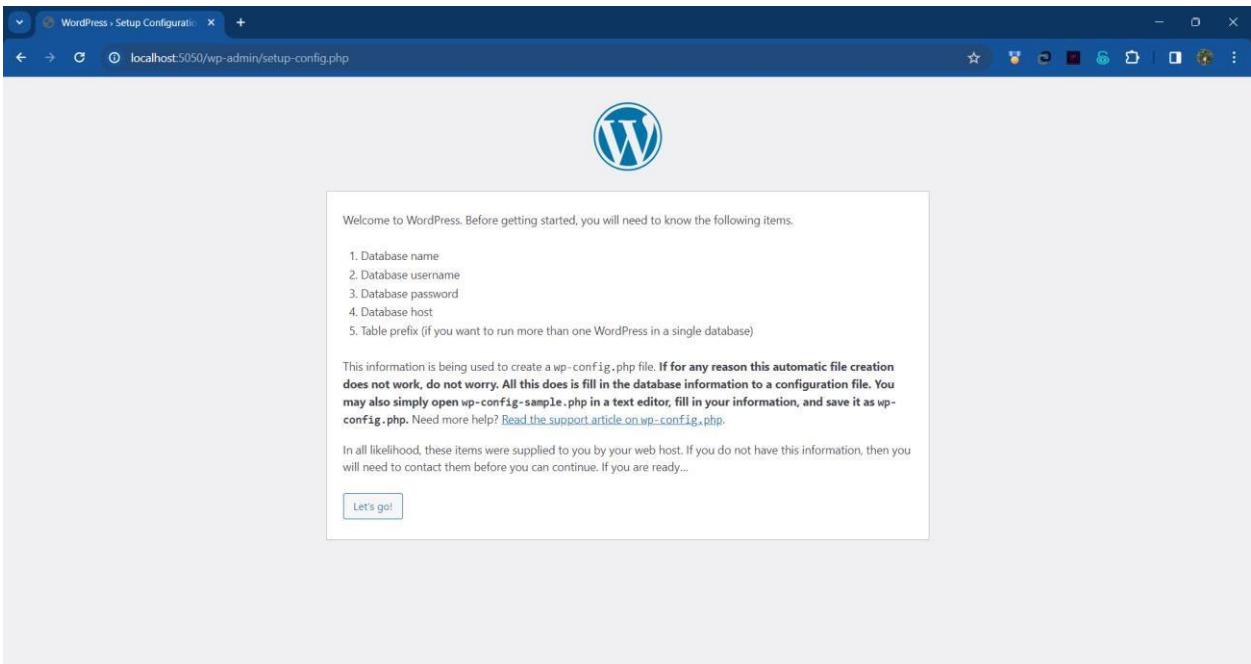
Here we can see the newly created images(highlighted with yellow colour) and container(highlighted with blue colour)

```

PS C:\Users\User> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
adarshankari/04nginx    latest   f28c8936f605  19 hours ago  187MB
04nginx             latest   f28c8936f605  19 hours ago  187MB
mysql               5        5107333e08a8  3 weeks ago   501MB
ubuntu              latest   174c8c134b2a  3 weeks ago   77.9MB
wordpress            latest   9871407ed1c8  4 weeks ago   740MB
nginx               latest   d453dd892d93  2 months ago  187MB
PS C:\Users\User> docker ps -a
CONTAINER ID        IMAGE           COMMAND          CREATED        STATUS          PORTS          NAMES
868127881b3f      wordpress      "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes   0.0.0.0:5050->80/tcp   user-frontend-1
231c657a21e3      mysql:5       "docker-entrypoint.s..." 5 minutes ago  Up 5 minutes   3306/tcp, 33060/tcp  user-backend-1
d282862aae77      04nginx       "/docker-entrypoint..." 19 hours ago  Exited (255) 18 minutes ago  0.0.0.0:9098->80/tcp  youthful_napier
563ca0a08873      nginx         "/docker-entrypoint..." 19 hours ago  Exited (255) 18 minutes ago  0.0.0.0:9090->80/tcp  interesting_rosalind
a974dd2d6ef7      ubuntu         "/bin/bash"      20 hours ago  Exited (0) 20 hours ago   blissful_beaver
260a58434b8c      ubuntu         "/bin/bash"      20 hours ago  Exited (0) 20 hours ago   myubuntu

```

Wordpress sever can be opened in localhost:5050 as mentioned in dockercompose file



The “**docker-compose down**” command is used to stop and remove containers, created by the **docker-compose up** command.

Here we can see that the containers are removed

```

PS C:\Users\User> docker-compose down
[+] Running 3/3
  ✓ Container user-frontend-1  Removed
  ✓ Container user-backend-1  Removed
  ✓ Network user_default  Removed
PS C:\Users\User> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
adarshankari/04nginx    latest   f28c8936f605  19 hours ago  187MB
04nginx             latest   f28c8936f605  19 hours ago  187MB
mysql               5        5107333e08a8  3 weeks ago   501MB
ubuntu              latest   174c8c134b2a  3 weeks ago   77.9MB
wordpress            latest   9871407ed1c8  4 weeks ago   740MB
nginx               latest   d453dd892d93  2 months ago  187MB
PS C:\Users\User> docker ps -a
CONTAINER ID        IMAGE           COMMAND          CREATED        STATUS          PORTS          NAMES
d282862aae77      04nginx       "/docker-entrypoint.s..." 19 hours ago  Exited (255) 24 minutes ago  0.0.0.0:9098->80/tcp  youthful_napier
563ca0a08873      nginx         "/docker-entrypoint..." 19 hours ago  Exited (255) 24 minutes ago  0.0.0.0:9090->80/tcp  interesting_rosalind
a974dd2d6ef7      ubuntu         "/bin/bash"      20 hours ago  Exited (0) 20 hours ago   blissful_beaver
260a58434b8c      ubuntu         "/bin/bash"      20 hours ago  Exited (0) 20 hours ago   myubuntu
PS C:\Users\User>

```

The “**docker-compose up -d --scale**” command is used to create and start containers for a service in the background, and to scale the service to a specified number of instances.

The “**docker-compose up -d --scale frontend=3**” command is used to create and start three containers for the frontend service in the background.

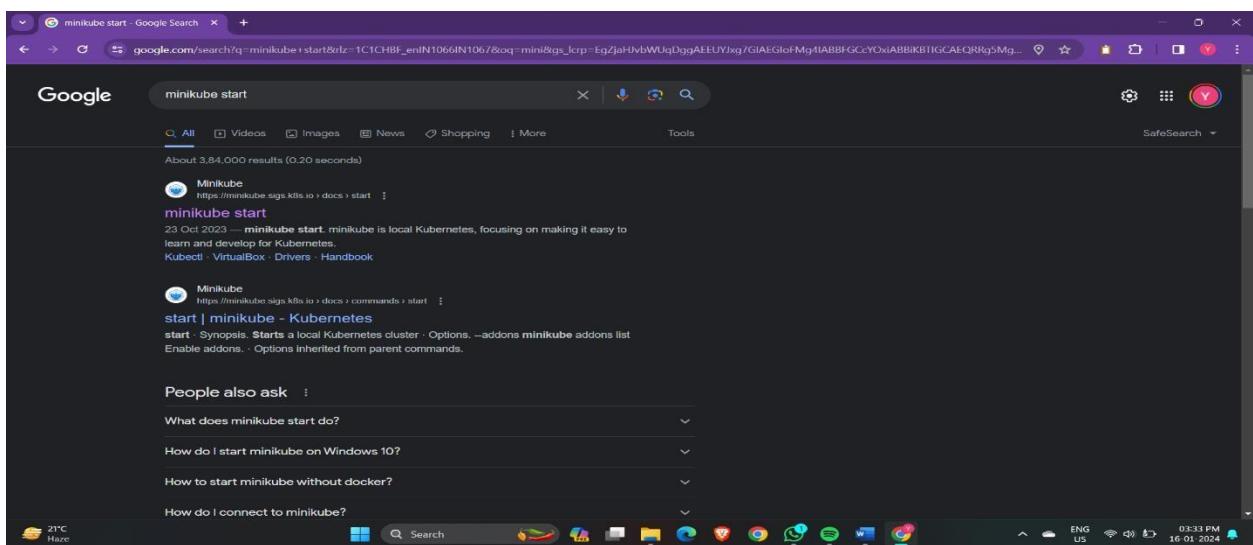
```
PS C:\Users\>User> docker-compose up -d --scale frontend=3
[+] Running 5/5
  ✓ Network user_default      Created
  ✓ Container user-backend-1  Started
  ✓ Container user-frontend-1 Created
  ✓ Container user-frontend-2 Created
  ✓ Container user-frontend-3 Started
Error response from daemon: driver failed programming external connectivity on endpoint user-frontend-1 (18041b79ec6blea91d739dfec23545b483d423596539307f637
bf21cf32dbb59): Bind for 0.0.0.0:5050 failed: port is already allocated
PS C:\Users\User> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS                         NAMES
9a806f1d7783        wordpress          "docker-entrypoint.s..."   17 seconds ago     Up 12 seconds      0.0.0.0:5050->80/tcp    user-frontend-3
64fafafef5a65       wordpress          "docker-entrypoint.s..."   17 seconds ago     Created           0.0.0.0:8080->80/tcp    user-frontend-1
d3f7ee8c5b4f        wordpress          "docker-entrypoint.s..."   17 seconds ago     Created           0.0.0.0:8081->80/tcp    user-frontend-2
aa8b04bb850c        mysql:5           "docker-entrypoint.s..."   17 seconds ago     Up 13 seconds      3306/tcp, 33060/tcp     user-backend-1
d282862ae77        04nginx            "/docker-entrypoint.s..."  19 hours ago      Exited (255) 27 minutes ago  0.0.0.0:9098->80/tcp    youthful_napier
563ca0a08873        nginx              "/docker-entrypoint.s..."  19 hours ago      Exited (255) 27 minutes ago  0.0.0.0:9090->80/tcp    interesting_rosalind
a974dd2d6ef7        ubuntu              "/bin/bash"          20 hours ago     Exited (0) 20 hours ago   0.0.0.0:22              blissful_beaver
260a58434b8c        ubuntu              "/bin/bash"          20 hours ago     Exited (0) 20 hours ago   0.0.0.0:4243            myubuntu
```

Experiment 10: **Working with Kubernetes**

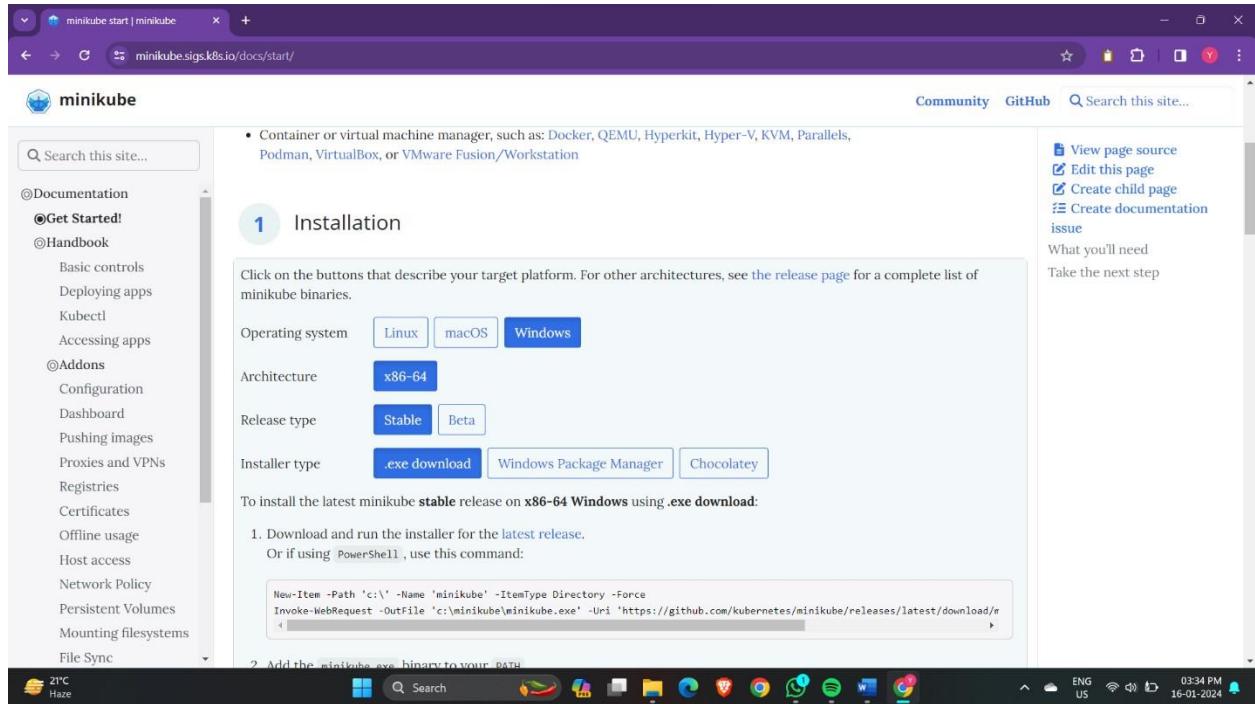
- a) Minikube installation
- b) Start and run minikube
- c) Using the following minikube commands – version, status
- d) Create and deploy Nginx service
- e) Using the following minikube commands – get deployment, get pods, describe pods, scale deployment, expose, service
- f) Run the Nginx server on the given URL.
- g) Delete the deployment, Stop minikube

Kubernetes:

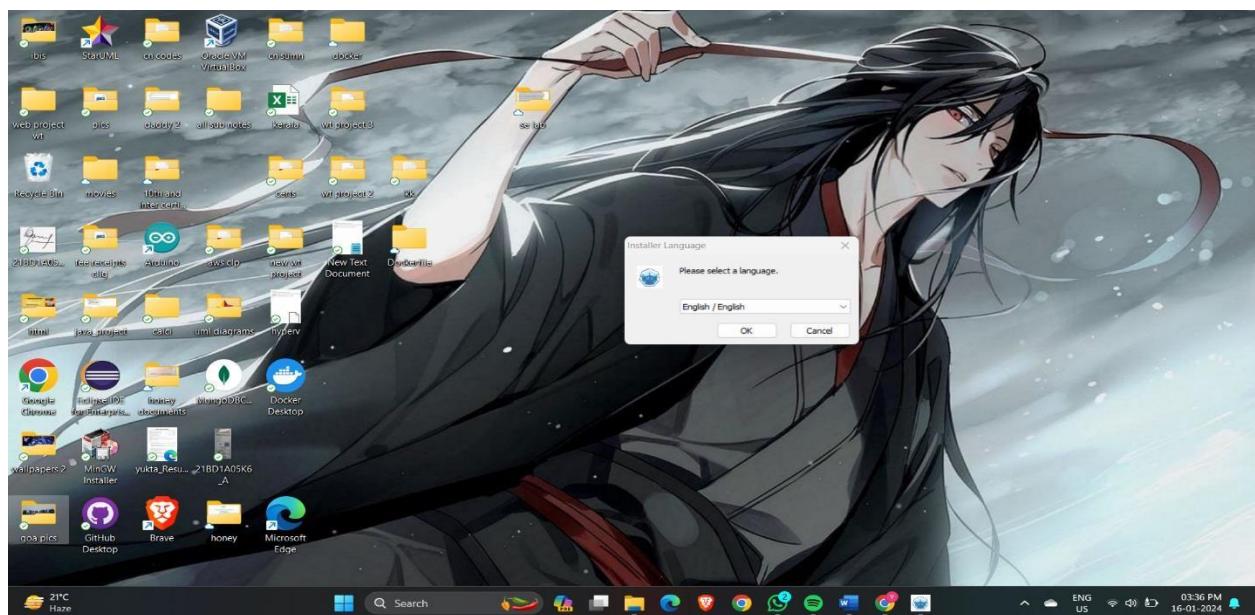
Step 1: search for minikube install in google. Click on the highlighted link as shown.



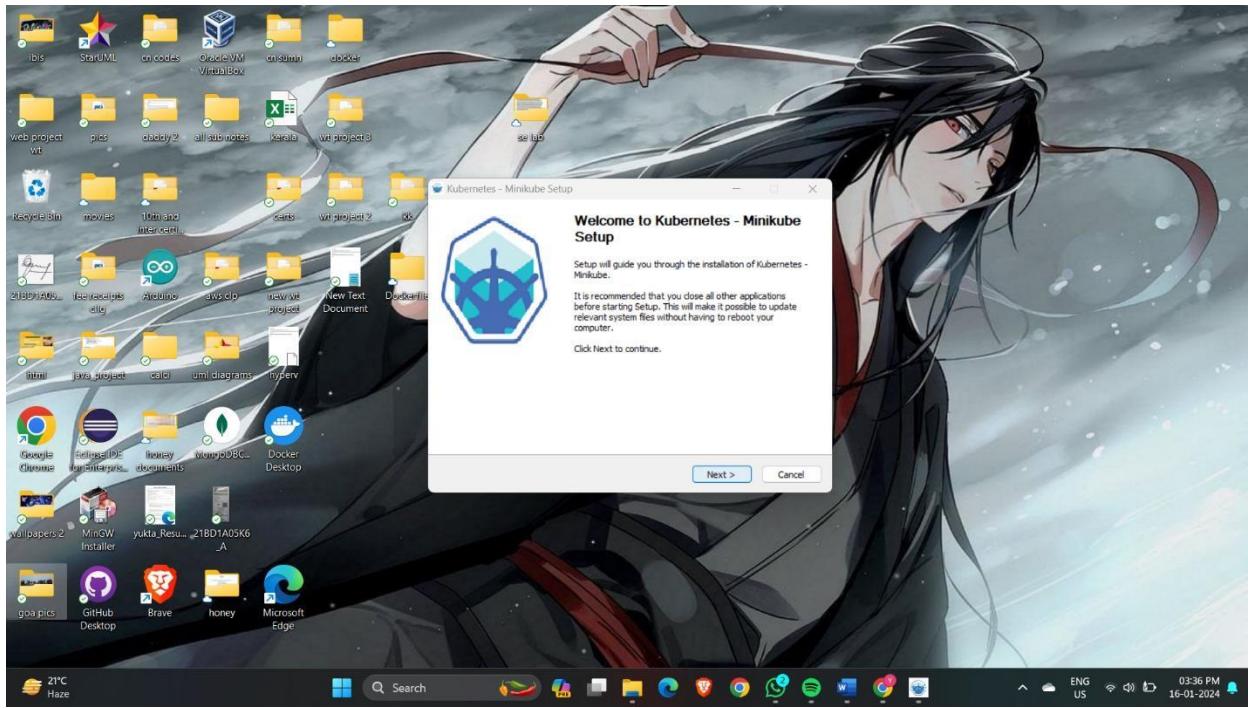
Step 2: Make sure the following options are highlighted and click on latest release to download the executable file.



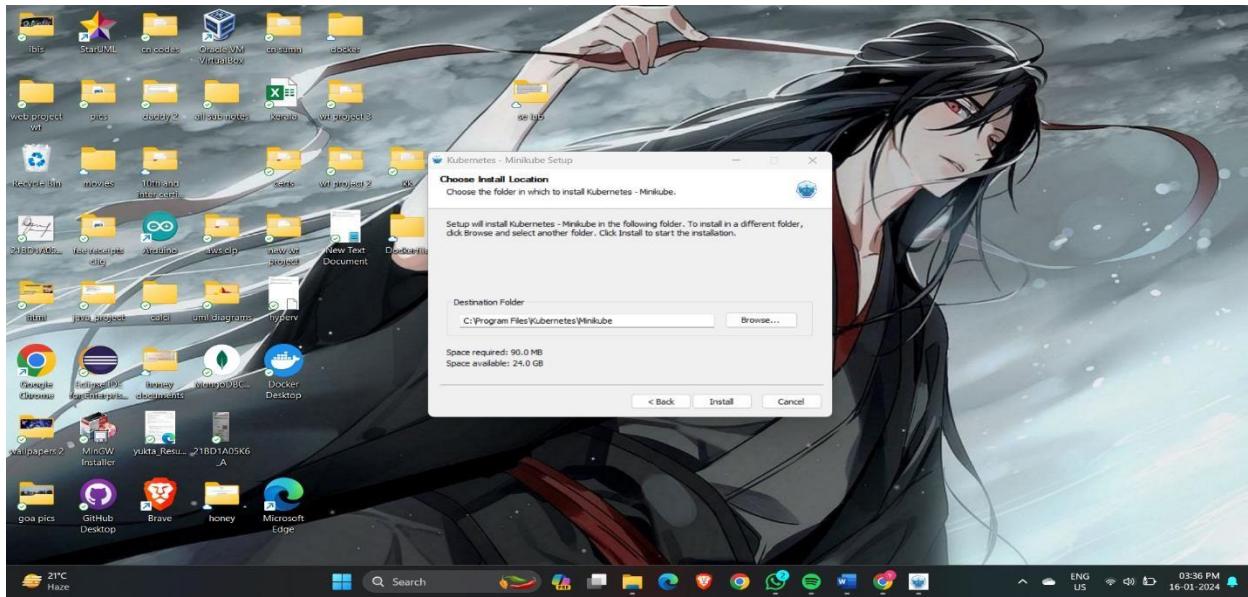
**Step 3: The minikube-installer is downloaded successfully.
Select the language preferable English.**



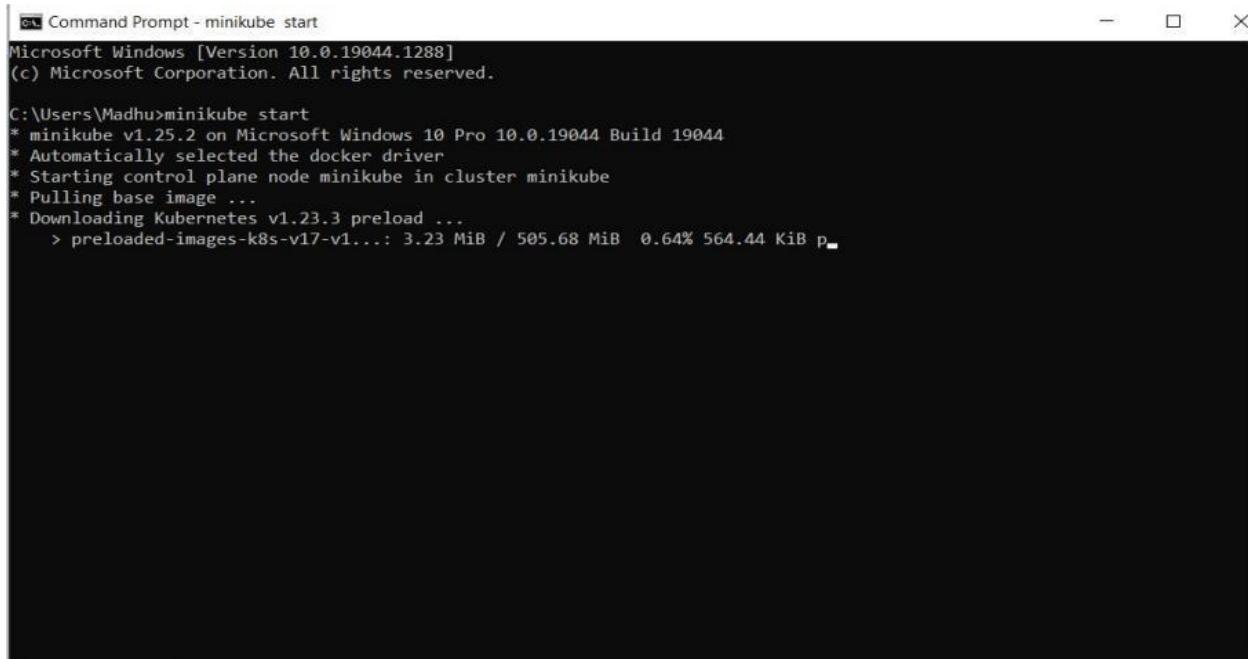
Step 4: Click on Next and agree to the license.



Step 5: select the path where you want to store it and Installation is successfully completed, Click on Next and finish.



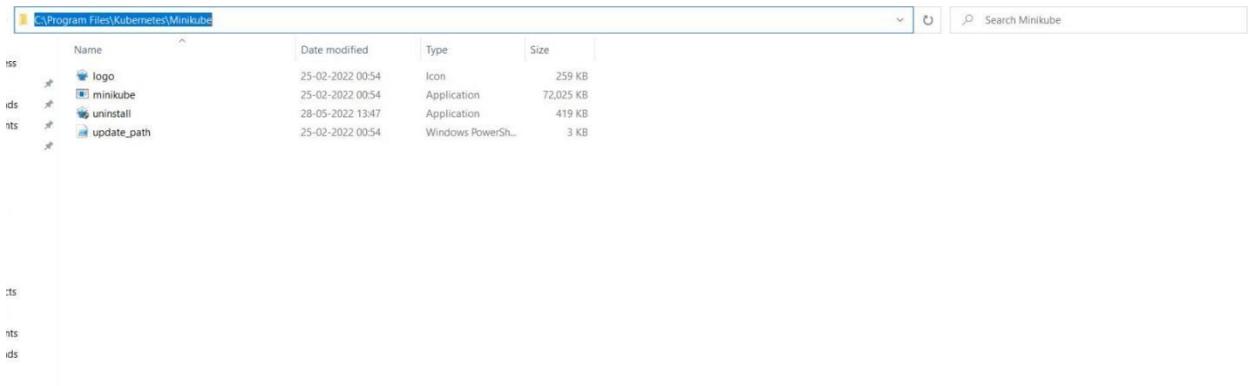
Step 6: Open the command prompt/ Windows PowerShell always in administrative mode and paste the command minikube start.



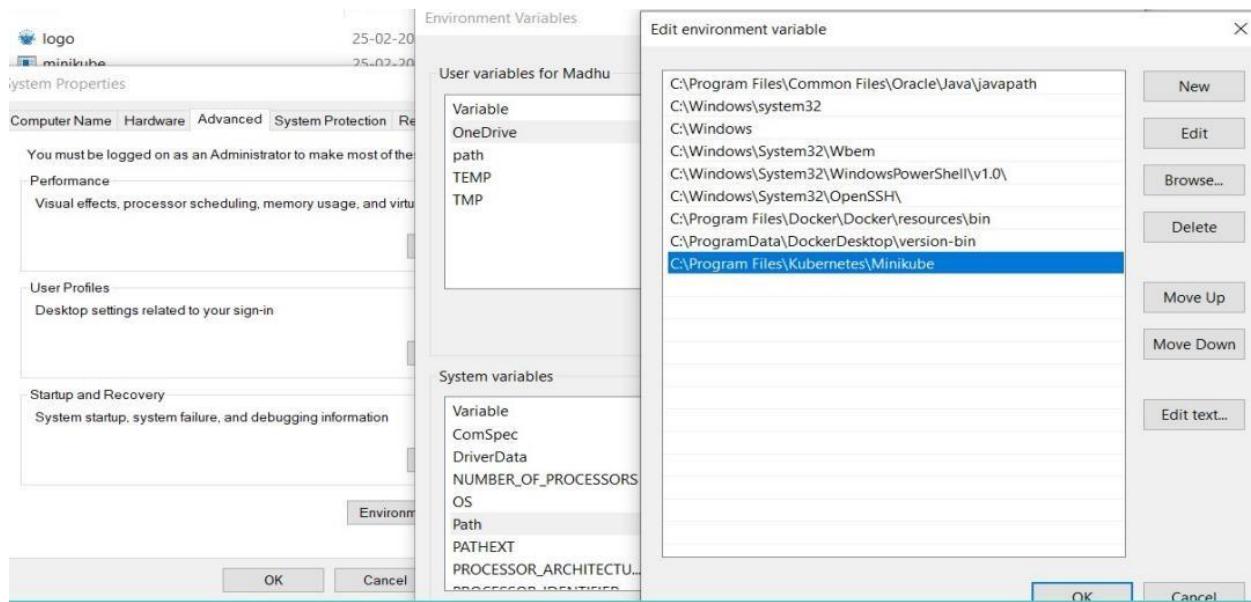
```
Command Prompt - minikube start
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Madhu>minikube start
* minikube v1.25.2 on Microsoft Windows 10 Pro 10.0.19044 Build 19044
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.23.3 preload ...
  > preloaded-images-k8s-v17-v1...: 3.23 MiB / 505.68 MiB  0.64% 564.44 KiB p...
```

Step 7: Sometimes it gives an error saying, internal command, in that case we need to set the path in environmental variables, firstly copy the minikube path as shown.



Step 8: Open Environmental variables->DoubleClick path in System variables and paste the copied path as shown->click Ok.



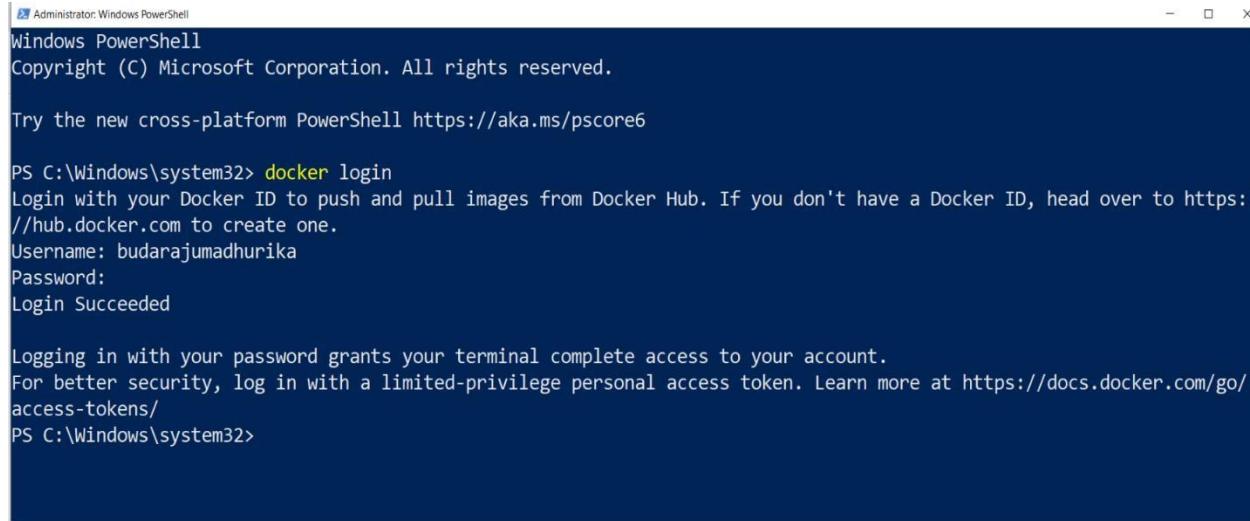
Step 9: Restart the command prompt/Windows PowerShell and type the minikube start command again.

```
C:\ Command Prompt
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Madhu>minikube start
* minikube v1.25.2 on Microsoft Windows 10 Pro 10.0.19044 Build 19044
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.23.3 preload ...
  > gcr.io/k8s-minikube/kicbase: 0 B [=====] 100% ? p/s 9m6s
  > preloaded-images-k8s-v17-v1...: 505.68 MiB / 505.68 MiB 100.00% 935.94 K
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  - kubelet.housekeeping-interval=5m
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* 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

C:\Users\Madhu>
```

Step 10: Open Windows PowerShell in Administrator Mode and login into Docker.



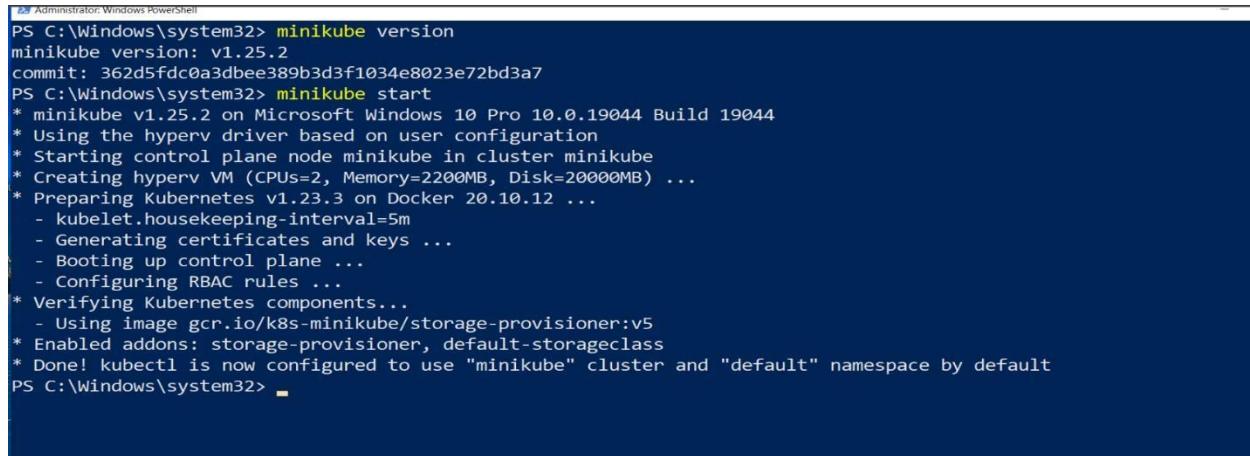
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: budarajumadhurika
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
PS C:\Windows\system32>
```

Step 11: We can start minikube using Hyper-V, Docker or VM
minikube start –vm-driver=virtualbox if I use only minikube
start by default Linux: docker (if installed) or kvm2, Windows:
hyperv, macOS: hyperkit will be used.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube version
minikube version: v1.25.2
commit: 362d5fdc0a3dbe389b3d3f1034e8023e72bd3a7
PS C:\Windows\system32> minikube start
* minikube v1.25.2 on Microsoft Windows 10 Pro 10.0.19044 Build 19044
* Using the hyperv driver based on user configuration
* Starting control plane node minikube in cluster minikube
* Creating hyperv VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
* Preparing Kubernetes v1.23.3 on Docker 20.10.12 ...
  - kubelet.housekeeping-interval=5m
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* 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
PS C:\Windows\system32>
```

Step 12: The minikube status command is used to check and display the current status of the Minikube cluster running on your local machine. In simple terms, it provides information about whether the Minikube cluster is up and running or if it's stopped.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

PS C:\Windows\system32>
```

Step 13: lets deploy an application in Kubernetes. kubectl create deployment mynginx --image=nginx.

```
Select Administrator: Windows PowerShell
PS C:\Windows\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

PS C:\Windows\system32> kubectl create deployment mynginx --image=nginx
```

step 14: kubectl get deployment

when you execute this command, Kubernetes responds by showing you a list that includes the names of your deployment groups, how many instances of your applications are running, and other useful details.

kubectl get pods running **kubectl get pods** is a quick way to check which of your applications are currently active and doing their jobs inside the Kubernetes system.

*Here the status shows that the containercreating.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

PS C:\Windows\system32> kubectl create deployment mynginx --image=nginx
deployment.apps/mynginx created
PS C:\Windows\system32> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   0/1     1           0           35s
PS C:\Windows\system32> kubectl get pods
NAME                  READY   STATUS            RESTARTS   AGE
mynginx-6b78685d4d-8cqxr   0/1     ContainerCreating   0          92s
PS C:\Windows\system32>
```

Step 15: This command opens up each pod's file and tells you everything about it. It's like reading a detailed report that includes the current state, recent events, and all the configurations of each of your applications. So, running `kubectl describe pods` is a way to get a thorough understanding of what's happening inside each pod in your Kubernetes cluster.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl describe pods
Name:           mynginx-6b78685d4d-8cqxr
Namespace:      default
Priority:      0
Node:          minikube/172.23.45.150
Start Time:    Thu, 15 Dec 2022 11:53:58 +0530
Labels:        app=mynginx
               pod-template-hash=6b78685d4d
Annotations:  <none>
Status:       Running
IP:          172.17.0.3
IPs:          IP: 172.17.0.3
Controlled By: ReplicaSet/mynginx-6b78685d4d
Containers:
  nginx:
    Container ID: docker://7cc34f8ebbcd648c9688126cf3c8511eaca66f1b62199b26199f02b76b735ca
    Image:        nginx
    Image ID:    docker-pullable://nginx@sha256:75263be7e5846fc69cb6c42553ff9c93d653d769b94917dbda71d42d3f3c00d3
    Port:         <none>
    Host Port:   <none>
    State:       Running
      Started:   Thu, 15 Dec 2022 11:56:22 +0530
    Ready:       True
    Restart Count: 0
    Environment: <none>
    Mounts:
```

Step 16: Type the following commands once again `kubectl get deployment` `kubectl get pods`.

```
Administrator: Windows PowerShell
PodScheduled      True
Volumes:
kube-api-access-qkjsz:
  Type:          Projected (a volume that contains injected data from multiple sources)
  TokenExpirationSeconds: 3607
  ConfigMapName:    kube-root-ca.crt
  ConfigMapOptional: <nil>
  DownwardAPI:     true
QoS Class:        BestEffort
Node-Selectors:   <none>
Tolerations:      node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
Type    Reason     Age   From            Message
----    ----       --   --              --
Normal  Scheduled  3m4s  default-scheduler  Successfully assigned default/mynginx-6b78685d4d-8cqxr to minikube
Normal  Pulling    3m3s  kubelet         Pulling image "nginx"
Normal  Pulled     40s   kubelet         Successfully pulled image "nginx" in 2m23.084699052s
Normal  Created    40s   kubelet         Created container nginx
Normal  Started    40s   kubelet         Started container nginx
PS C:\Windows\system32> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  1/1     1           1           3m15s
PS C:\Windows\system32> kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
mynginx-6b78685d4d-8cqxr  1/1     Running   0          3m20s
PS C:\Windows\system32>
```

Step 17: kubectl scale deployment mynginx --replicas=4

this command is used to increase the number of replicas (copies) of the "mynginx" deployment in your Kubernetes cluster to 4.

scale deployment mynginx: Specifies that you want to scale the deployment named "mynginx."

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32>
```

Step 18: Now if we check the following command,kubectl get deployment.We can see that 4 instances of mynginx are available.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx  4/4     4           4           8m23s
PS C:\Windows\system32>
```

Step 19: if we check the following command,kubectl get pods.We can see that 4 instances of mynginx are running.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   4/4     4           4           8m23s
PS C:\Windows\system32> kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
mynginx-6b78685d4d-8cqxr  1/1     Running   0          9m3s
mynginx-6b78685d4d-hrc9h  1/1     Running   0          77s
mynginx-6b78685d4d-nk76q  1/1     Running   0          77s
mynginx-6b78685d4d-qlr5k  1/1     Running   0          77s
PS C:\Windows\system32>
```

Step 20: type kubectl describe pod mynginx-6b78685d4dhrc9h. This command provides detailed information about a specific pod .

```
Select Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl scale deployment mynginx --replicas=4
deployment.apps/mynginx scaled
PS C:\Windows\system32> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   4/4     4           4           8m23s
PS C:\Windows\system32> kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
mynginx-6b78685d4d-8cqxr  1/1     Running   0          9m3s
mynginx-6b78685d4d-hrc9h  1/1     Running   0          77s
mynginx-6b78685d4d-nk76q  1/1     Running   0          77s
mynginx-6b78685d4d-qlr5k  1/1     Running   0          77s
PS C:\Windows\system32> kubectl describe pod mynginx-6b78685d4d-hrc9h
```

Step 21:

```

Select Administrator: Windows PowerShell
IP: 172.17.0.6
Controlled By: ReplicaSet/mynginx-6b78685d4d
Containers:
  nginx:
    Container ID: docker://54d65c50c3abab016f139a68e356578ecaa94bc987651af515f18bd605df5137
    Image: nginx
    Image ID: docker-pullable://nginx@sha256:75263be7e5846fc69cb6c42553ff9c93d653d769b94917dbda71d42d3f3c00d3
    Port: <none>
    Host Port: <none>
    State: Running
    Started: Thu, 15 Dec 2022 12:01:54 +0530
    Ready: True
    Restart Count: 0
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-4bc8t (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-4bc8t:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
Select Administrator: Windows PowerShell
/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-4bc8t (ro)
Select Administrator: Windows PowerShell
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-4bc8t (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-4bc8t:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:   <nil>
    DownwardAPI:        true
  QoS Class:  BestEffort
  Node-Selectors:  <none>
  Tolerations:  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age  From          Message
  ----  -----  --  --  -----
  Normal Scheduled  3m30s default-scheduler  Successfully assigned default/mynginx-6b78685d4d-hrc9h to minikube
  Normal Pulling   3m29s kubelet        Pulling image "nginx"
  Normal Pulled    3m21s kubelet        Successfully pulled image "nginx" in 8.150664403s
  Normal Created   3m21s kubelet        Created container nginx
  Normal Started   3m20s kubelet        Started container nginx
PS C:\Windows\system32>

```

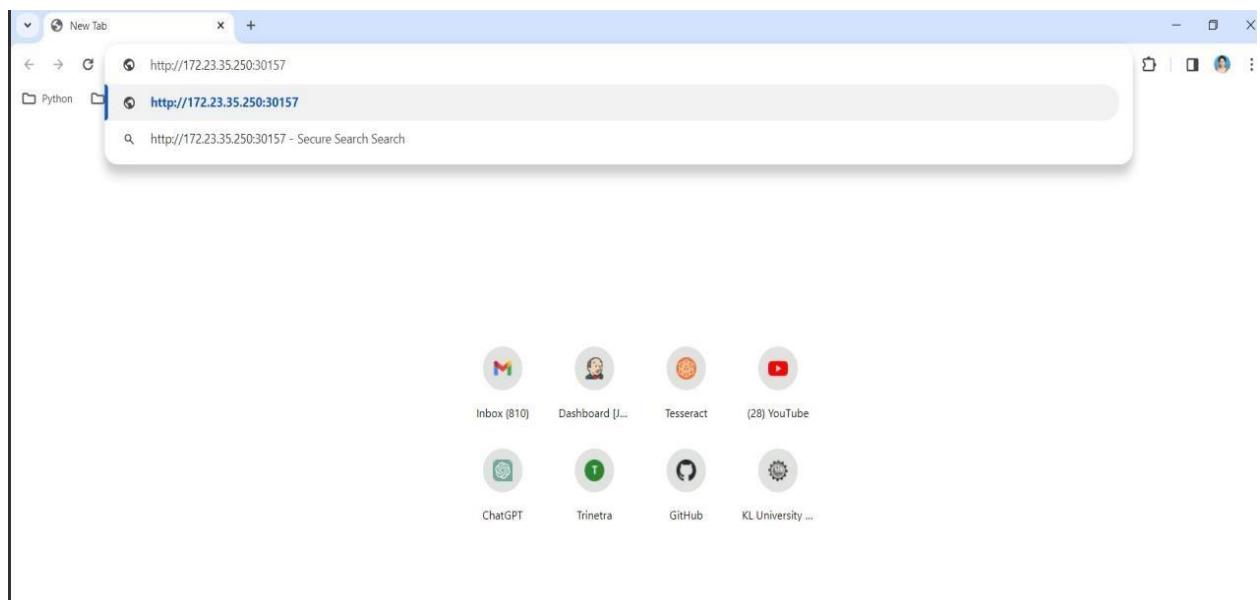
Step 22: type kubectl expose deployment mynginx - type=NodePort --port=88. This command is used to make a service publicly accessible from outside the Kubernetes cluster.

```
PS Select Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=88
```

Step 23: type minikube service mynginx –url.This command is used with Minikube, a tool for running Kubernetes locally, to get the URL that you can use to access a service deployed in your Minikube cluster.

```
PS Select Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl expose deployment mynginx --type=NodePort --port=88
service/mynginx exposed
PS C:\Windows\system32> minikube service mynginx --url
http://172.23.45.150:31242
PS C:\Windows\system32>
```

Step 24: Copy and Paste the URL generated.



Step 25: The nginx server is now accessible



Step 26: type minikube dashboard command it is used to open the Kubernetes Dashboard, a web-based user interface, when you are working with Minikube.

```
PS C:\Windows\system32> minikube dashboard
* Enabling dashboard ...
- Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
- Using image docker.io/kubernetesui/dashboard:v2.7.0
* Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server

* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:52304/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...

```

Activate Windows
Go to Settings to activate Windows.

Step 27: Minikube dashboard.

Name	Images	Labels	Pods	Created
mynginx5	nginx	app: mynginx5	1 / 1	17 minutes ago

Step 28: type kubectl delete deployment mynginx1 .this command is telling Kubernetes to delete the deployment

named "myngnix1" along with its associated resources (like pods).

```
PS C:\Windows\system32> kubectl delete deployment mynginx1
deployment.apps "mynginx1" deleted
PS C:\Windows\system32> minikube stop
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\Windows\system32>
```

Experiment 11:

Working with Nagios

a)Install Nagios using Docker

B)Run the Nagios container, open the browser and monitor localhost.

NAGIOS :

```
>Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

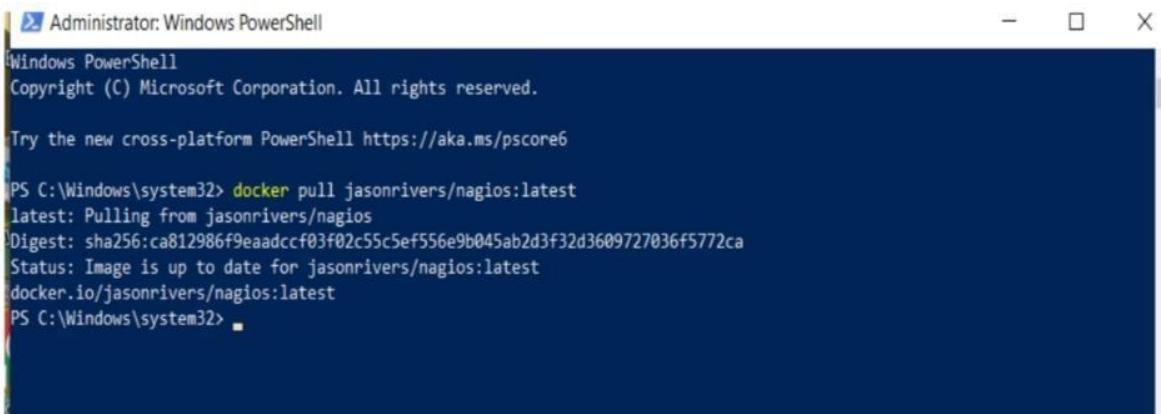
S C:\Windows\system32> docker pull jasonrivers/nagios:latest
```

```
>Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

S C:\Windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
88c01a0ec47e: Pull complete
8f2c286f74de: Downloading [=====] 53.23MB/214.9MB
:b72f88dd616: Download complete
4db858437011: Download complete
fd1fa5ad641: Download complete
5caaed8d4e2: Download complete
875b4a604e9: Download complete
58b2550c9dc: Download complete
5c8bc410dd5: Download complete
:e726826e8da: Download complete
1cdf4ce802a: Download complete
76a47ef1e6d: Download complete
45bcac744c2: Download complete
52fa82ffbef: Download complete
f6696a5c913: Download complete
9e122612ea9: Download complete
8173f9ece45: Download complete
01b473c218a: Download complete
021066c03e6: Download complete
d9f1450044c: Download complete
7d3e545730c: Download complete
b11164f4445: Download complete
e0387ef7e93: Download complete
d99a3464e3a: Download complete
381d60abab7: Waiting
ec8746654ee: Waiting
97d3238b8e5: Waiting
```

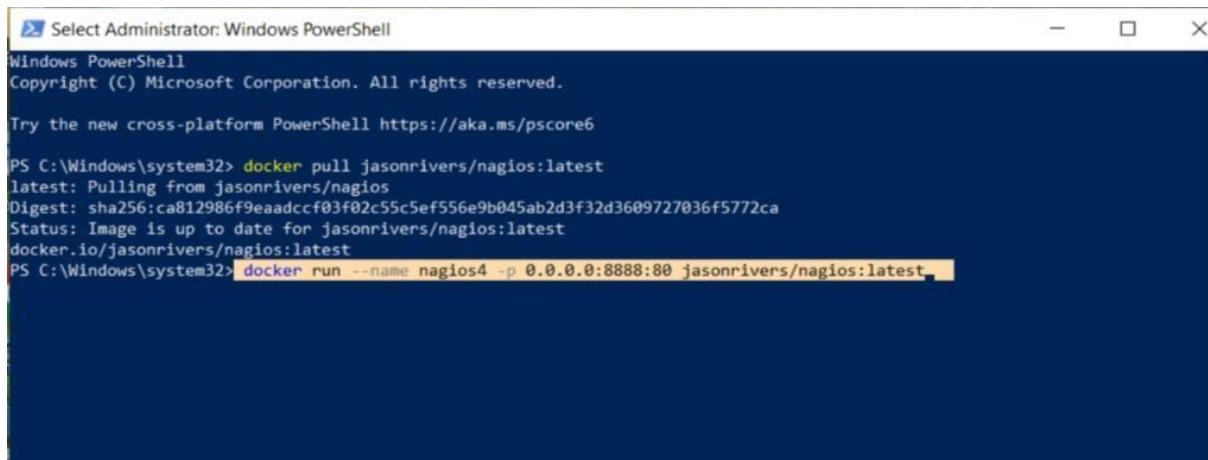
Nagios is successfully pulled



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
Digest: sha256:ca812986f9eaadccf03f02c55c5ef556e9b045ab2d3f32d3609727036f5772ca
Status: Image is up to date for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
PS C:\Windows\system32>
```



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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> docker pull jasonrivers/nagios:latest
latest: Pulling from jasonrivers/nagios
Digest: sha256:ca812986f9eaadccf03f02c55c5ef556e9b045ab2d3f32d3609727036f5772ca
Status: Image is up to date for jasonrivers/nagios:latest
docker.io/jasonrivers/nagios:latest
PS C:\Windows\system32> docker run --name nagios4 -p 0.0.0.0:8888:80 jasonrivers/nagios:latest
```

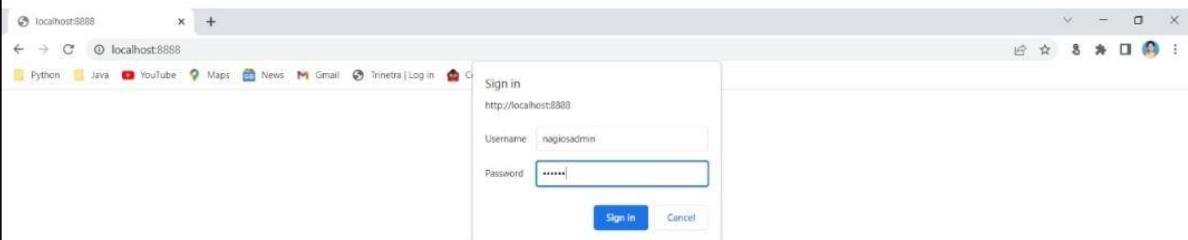
```
| Select Administrator: Windows PowerShell
Last Modified: 2020-04-28
License: GPL

Website: https://www.nagios.org
Nagios 4.4.6 starting... (PID=19)
Local time is Fri Jun 03 04:10:16 UTC 2022
nagios: Nagios 4.4.6 starting... (PID=19)
nagios: Local time is Fri Jun 03 04:10:16 UTC 2022
nagios: LOX VERSION: 2.0
wproc: Successfully registered manager as @wproc with query handler 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
wproc: Registry request: name=Core Worker 0;pid=0
nagios: wproc: Registry request: name=Core Worker 39;pid=39
nagios: wproc: Registry request: name=Core Worker 40;pid=40
nagios: wproc: Registry request: name=Core Worker 40;pid=40
wproc: Registry request: name=Core Worker 41;pid=41
nagios: wproc: Registry request: name=Core Worker 41;pid=41
wproc: Registry request: name=Core Worker 43;pid=43
wproc: Registry request: name=Core Worker 44;pid=44
wproc: Registry request: name=Core Worker 45;pid=45
wproc: Registry request: name=Core Worker 42;pid=42
nagios: wproc: Registry request: name=Core Worker 43;pid=43
nagios: wproc: Registry request: name=Core Worker 44;pid=44
nagios: wproc: Registry request: name=Core Worker 45;pid=45
nagios: wproc: Registry request: name=Core Worker 42;pid=42
wproc: Registry request: name=Core Worker 46;pid=46
nagios: wproc: Registry request: name=Core Worker 46;pid=46
nagios: wproc: Registry request: name=Core Worker 47;pid=47
nagios: wproc: Registry request: name=Core Worker 47;pid=47
wproc: Registry request: name=Core Worker 49;pid=49
nagios: wproc: Registry request: name=Core Worker 49;pid=49
wproc: Registry request: name=Core Worker 50;pid=50
nagios: wproc: Registry request: name=Core Worker 50;pid=50
Successfully launched command file worker with pid 51
nagios: Successfully launched command file worker with pid 51
postfix/master[18]: daemon started -- version 3.4.13, configuration /etc/postfix
-
```

We can access Nagios by opening the browser and typing localhost:8888, as shown



1. Username : nagiosadmin
2. Password: nagios



The Nagios dashboard will be opened, we can now monitor the services

The screenshot shows the Nagios Core 4.4.6 dashboard. On the left, a sidebar menu includes sections for General, Current Status, Reports, and System. The Current Status section is expanded, showing sub-options like Host Overview, Map (Legacy), Hosts, Services, Host Groups, and Problems. The Reports section includes Availability, Trends (Legacy), Alerts, History, Summary, Histogram (Legacy), Notifications, and Event Log. The System section includes Comments, Downtime, Process Info, Performance Info, Scheduling Queue, and Configuration.

The main content area features the Nagios Core logo at the top right, with a message indicating a process is running with PID 19. Below the logo, the text "Nagios® Core™ Version 4.4.6" is displayed, along with the date "April 28, 2020" and a link to "Check for updates". A blue banner at the bottom of the main content area states "A new version of Nagios Core is available! Visit nagios.org to download Nagios 4.4.7."

On the right side, there are three promotional banners:

- Nagios XI**: Easy Configuration Advanced Reporting. Includes a "Download" button.
- Nagios Log Server**: Monitor and analyze logs from anywhere. Includes a "Download" button.
- Nagios Network Analyzer**: Real-time netflow and bandwidth analysis. Includes a "Download" button.

At the bottom right, there are links for "Get started", "NAGIOS CORE PROJECT TOUR", and "Go to Site".

If we click on Hosts on the left side list, we can observe localhost as shown

The screenshot shows the Nagios dashboard with the "Hosts" option selected in the sidebar. The main content area displays the "Host Status Details For All Host Groups" table. The table has columns for Host, Status, Last Check, Duration, and Status Information. There is one entry for "localhost" with the status "UP".

Host	Status	Last Check	Duration	Status Information
localhost	UP	06-03-2022 04:12:24	0d 0h 5m 10s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Below the table, a message says "Results 1 - 1 of 1 Matching Hosts".

The sidebar also includes sections for General, Current Status, Reports, and System, mirroring the structure of the first screenshot.

You can now click on the localhost(highlighted)

Nagios: localhost

localhost:8088

Python Java YouTube Maps News Gmail Trineta | Log in Conceptual model...

Nagios®

General Home Documentation

Current Status

- Tactical Overview Map (Legacy)
- Hosts Services Host Groups Summary Grid
- Service Groups Summary Grid
- Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search

Reports Availability Trends (Legacy) Alerts History Summary Histogram (Legacy) Notifications

Current Network Status

Last Updated: Fri Jun 3 04:15:27 UTC 2022
Updated every 90 seconds
Nagios Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0
All Problems	All Types		
0	1		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	0	0
All Problems	All Types			
1	7			

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
localhost	UP	06-03-2022 04:12:24	0d 0h 5m 10s	PING OK - Packet loss = 0%, RTA = 0.03 ms

Results 1 - 1 of 1 Matching Hosts.

We can even view various service groups by clicking on the highlighted option

Nagios: localhost

localhost:8088

Python Java YouTube Maps News Gmail Trineta | Log in Conceptual model...

Nagios®

General Home Documentation

Current Status

- Tactical Overview Map (Legacy)
- Hosts Services Host Groups Summary Grid
- Service Groups Summary Grid
- Problems Services (Unhandled) Hosts (Unhandled) Network Outages Quick Search

Reports Availability Trends (Legacy) Alerts History Summary Histogram (Legacy) Notifications

Current Network Status

Last Updated: Fri Jun 3 04:16:18 UTC 2022
Updated every 90 seconds
Nagios Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin

View Service Status Detail For All Service Groups
View Status Summary For All Service Groups
View Status Grid For All Service Groups

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0
All Problems	All Types		
0	1		

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
6	1	0	0	0
All Problems	All Types			
1	7			

Service Overview For All Service Groups

There are no service groups defined.

We can observe various services like Currentload, HTTP, PING,etc by clicking on the highlighted link

The screenshot shows the Nagios web interface at localhost:8888. The main dashboard displays "Current Network Status" and "Service Status Totals". Below these are two tables: "Host Status Totals" and "Service Status Details For All Service Groups". The "Service Status Details" table lists various services with their status, last check time, duration, attempts, and status information. A legend indicates that green means OK, yellow means WARNING, and red means CRITICAL.

Service Status Details For All Service Groups					
Host	Service	Status	Last Check	Duration	Attempts
localhost	Current Load	OK	06-03-2022 04:15:58	0d 0h 5m 41s	1/1
	Current Users	OK	06-03-2022 04:11:41	0d 0h 6m 23s+	1/1
	HTTP	WARNING	06-03-2022 04:15:24	0d 0h 1m 15s	4/4
	PING	OK	06-03-2022 04:13:07	0d 0h 6m 23s+	1/1
	Root Partition	OK	06-03-2022 04:13:50	0d 0h 6m 23s+	1/1
	Swap Usage	OK	06-03-2022 04:14:33	0d 0h 6m 23s+	1/1
	Total Processes	OK	06-03-2022 04:15:16	0d 0h 6m 23s+	1/1

If we observe the docker containers in DockerHub, we can see the latest Nagios Installed running on port:8888

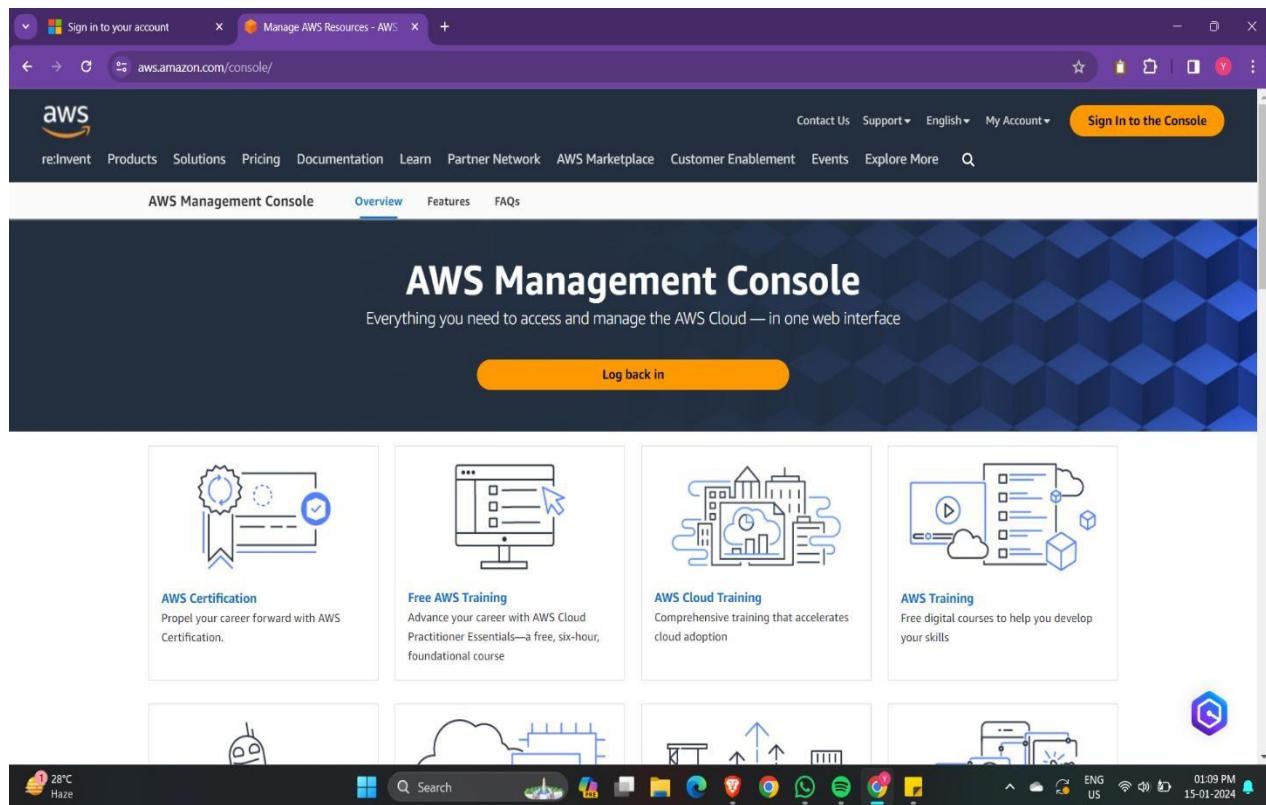
The screenshot shows the Docker Hub interface with the "Containers / Apps" tab selected. It displays a list of running containers, each with its name, image, and status. At the bottom right, there are icons for creating a new container, cloning an existing one, and deleting it.

Container	Image	Status
mystifying_rosalind	myimage1:1.0	EXITED (0)
unruffled_mclaren	myimage1:1.0	EXITED (0)
jolly_moore	ubuntu	EXITED (137)
hungry_hopper	hello-world	EXITED (0)
dreamy_colden	docker/getting-started	EXITED (255) PORT: 80
amazing_lovelace	hello-world	EXITED (0)
nagios4	jenkins/nagios4	EXITED (4) PORT: 8888

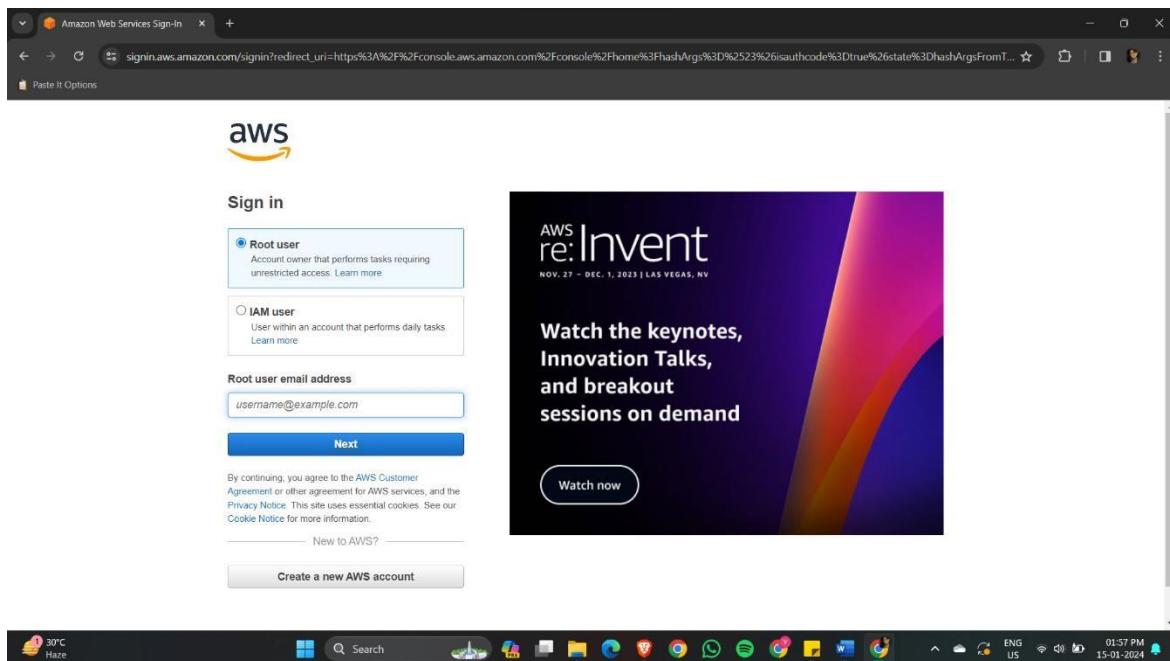
Experiment 12: Working with AWS

- a) Create an Amazon Web Services (AWS) free account
- b) Create an Elastic Cloud Computing (EC2) instance of Ubuntu.

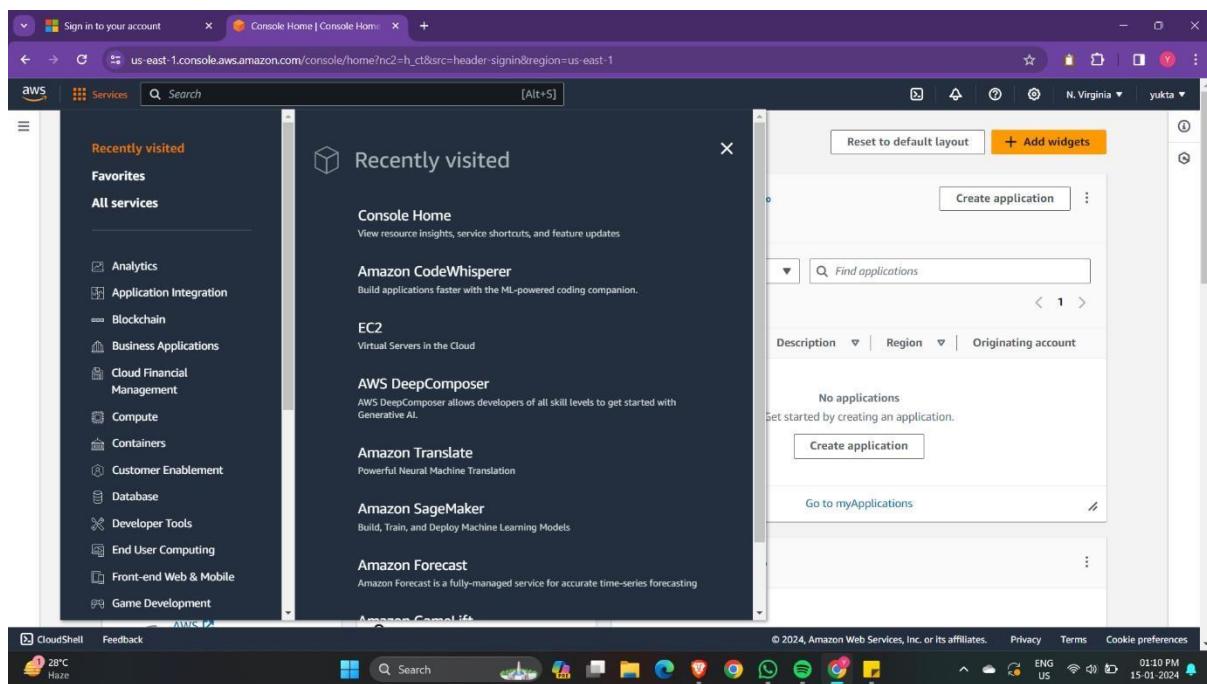
Step 1: go to the AWS management console on google. Click on the sign in to the console option.



Step 2: Sign in to the console as root user if you have an account . If you do not have an AWS account create one.



Step 3: once you are signed in to the console click on all services tab to view all the AWS services.

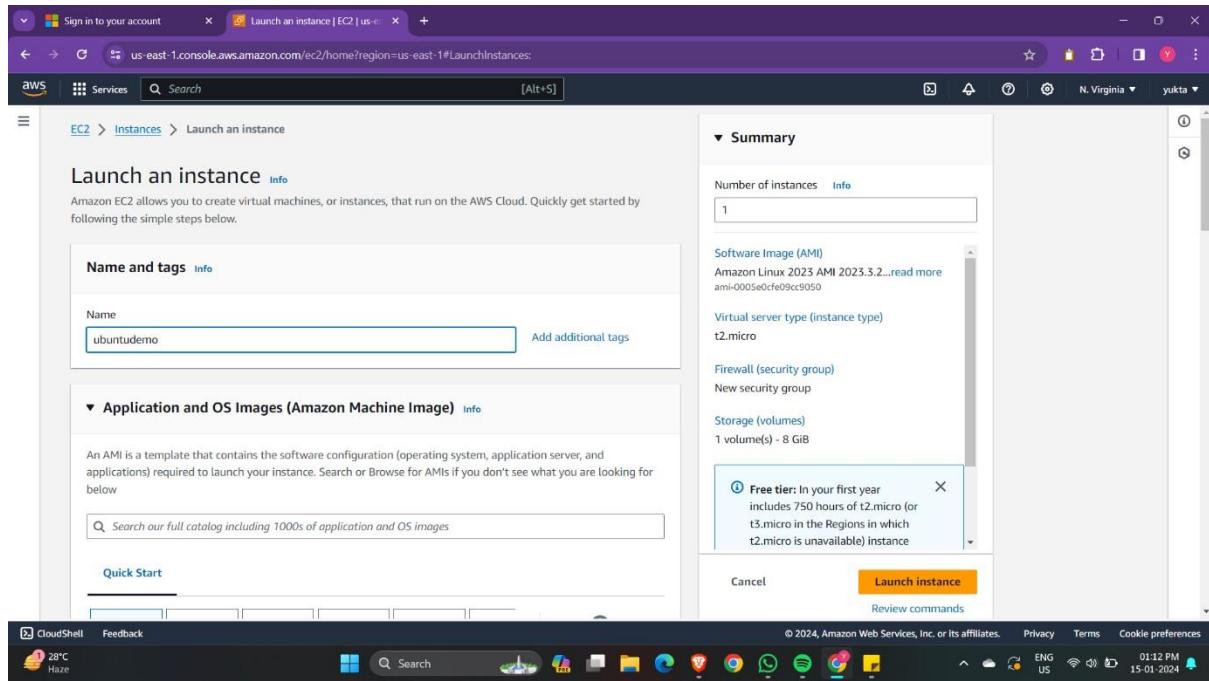


Step 4: select EC2 (elastic cloud compute) service in the services list and you are redirected to the EC2 dashboard .

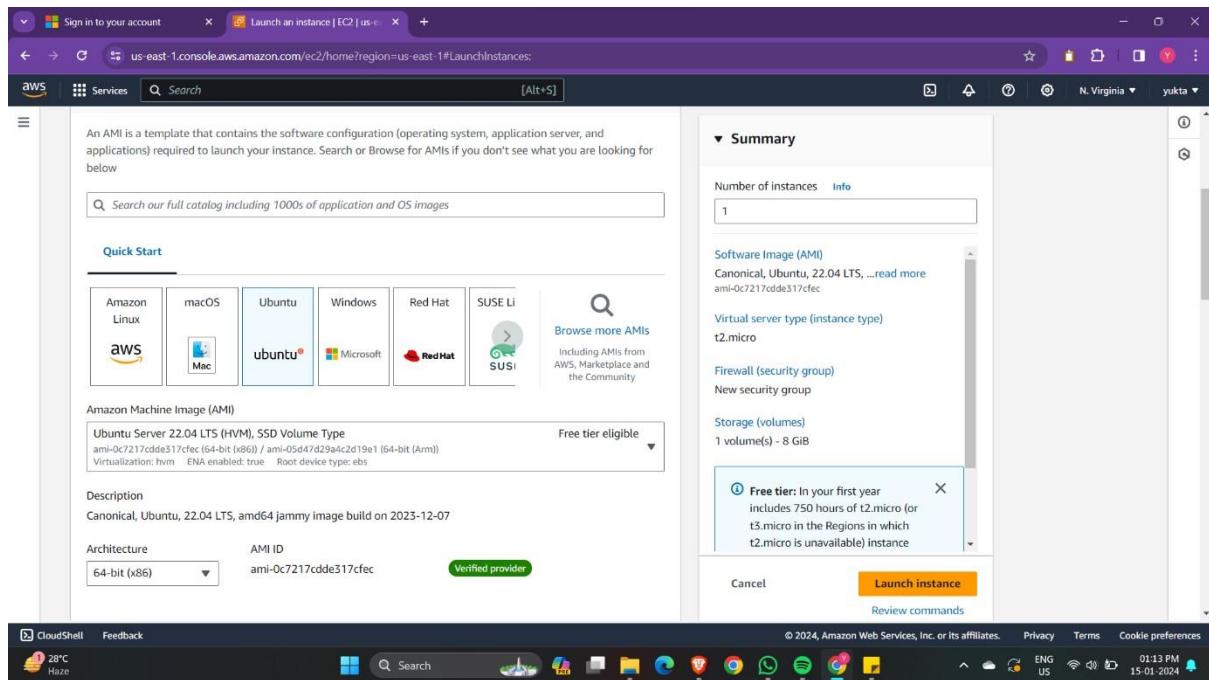
The screenshot shows the AWS EC2 Dashboard for the us-east-1 region. The left sidebar lists various services like EC2 Global View, Events, and Instances. The main area displays EC2 resources: 0 instances (running), 0 auto scaling groups, 0 dedicated hosts, 0 elastic IPs, 0 instances, 0 key pairs, 0 load balancers, 0 placement groups, 1 security group, 0 snapshots, and 0 volumes. Below this, there's a 'Launch instance' section with a large orange 'Launch Instance' button and a 'Migrate a server' link. A note says instances will launch in the US East (N. Virginia) Region. To the right, there's a 'Service health' section with a link to the AWS Health Dashboard, and a 'Zones' section showing zone names and IDs. On the far right, there are sections for 'EC2 Free Tier' offers, 'Offer usage (monthly)', and 'Account attributes'. The bottom of the screen shows the Windows taskbar with various pinned icons.

Step 5: select the location in which you want to launch the instance.

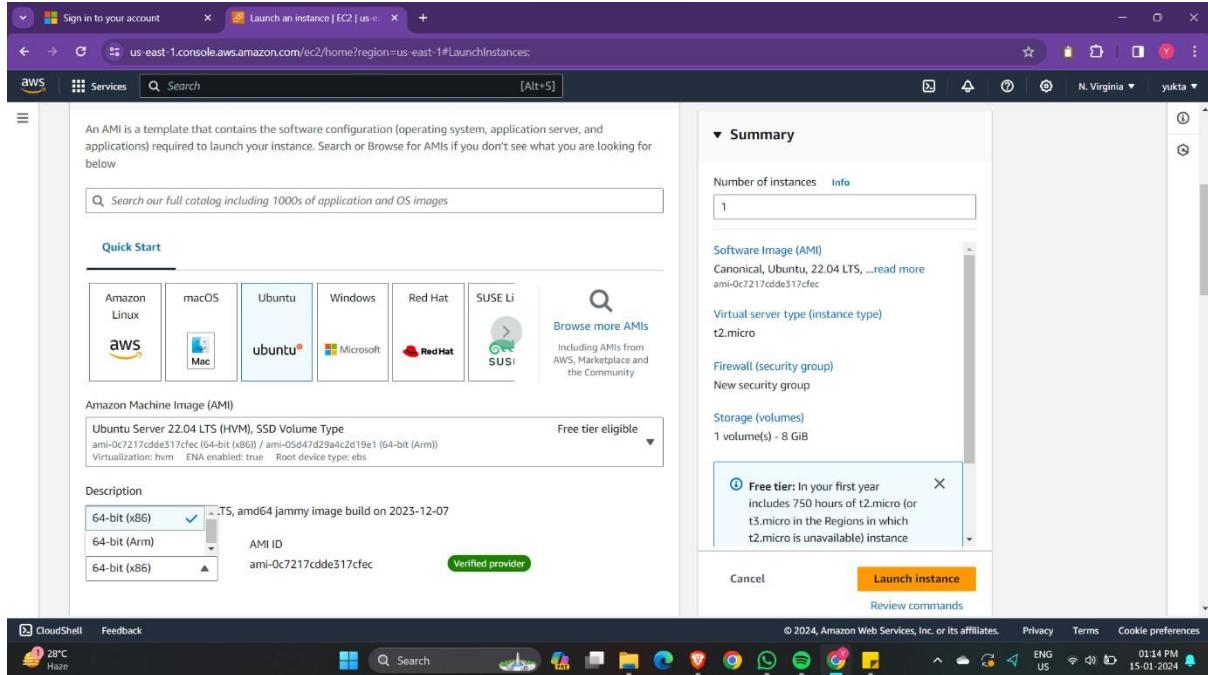
Click on launch instance.



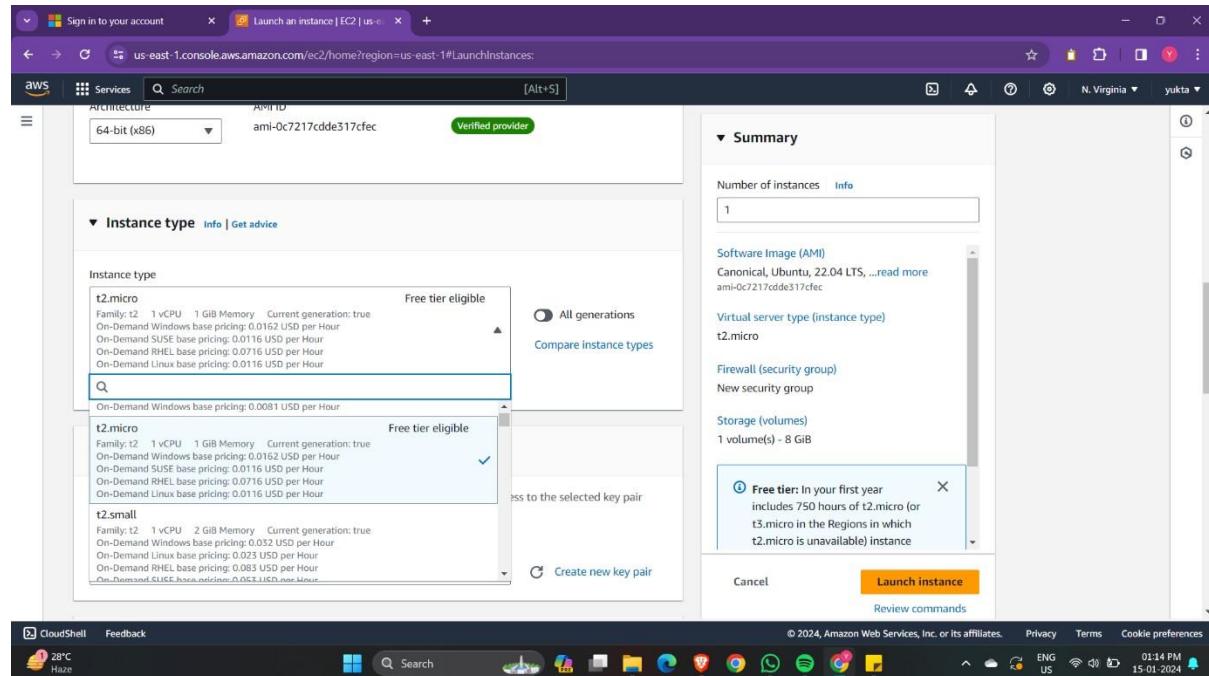
Step 6: select on which server you want to run the instance on .Here we select ubuntu.



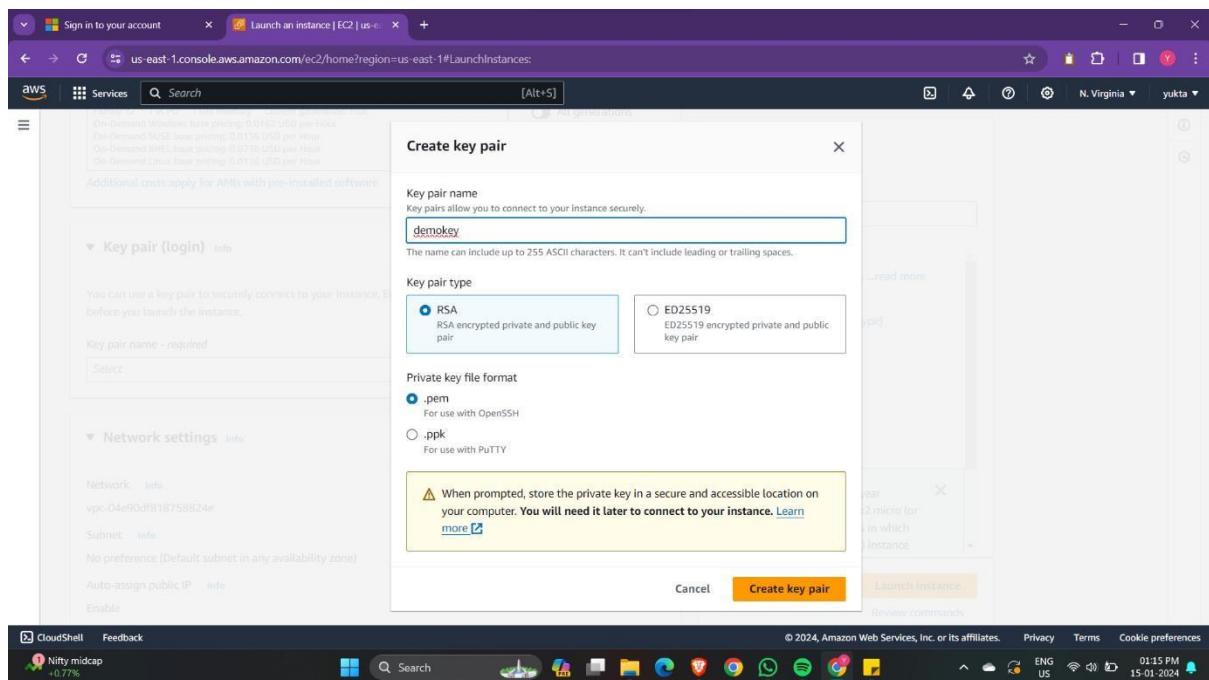
Step 7: select the ubuntu server 22.04 LTS in the free tier eligible option and select the type of architecture your system runs on (here 64 bit).



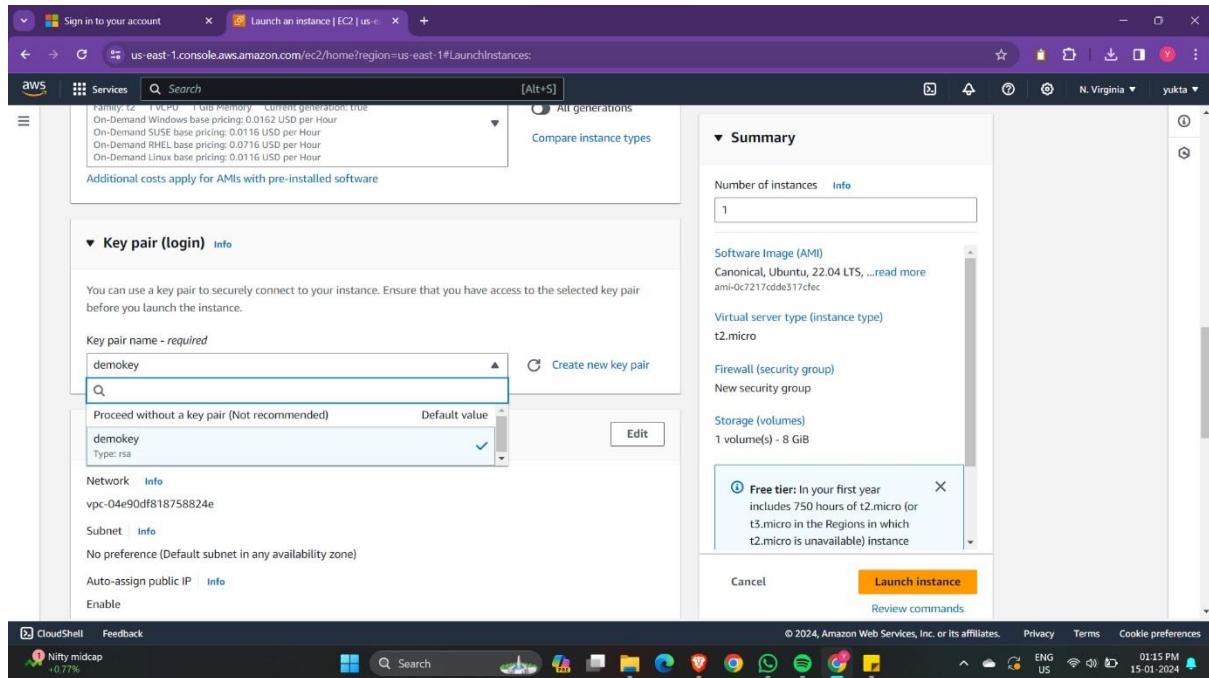
Step 8: select the instance type , here we select t2.micro (free tier).



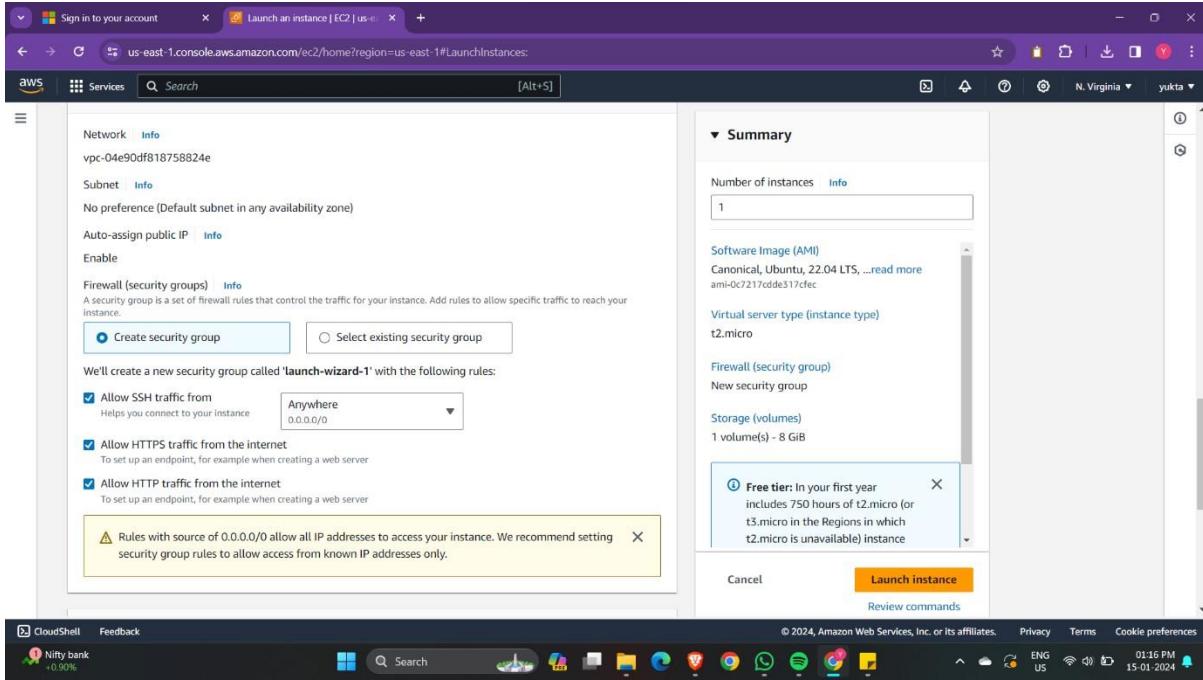
Step 9: create a new key pair and enter a key pair name and click on create key pair.



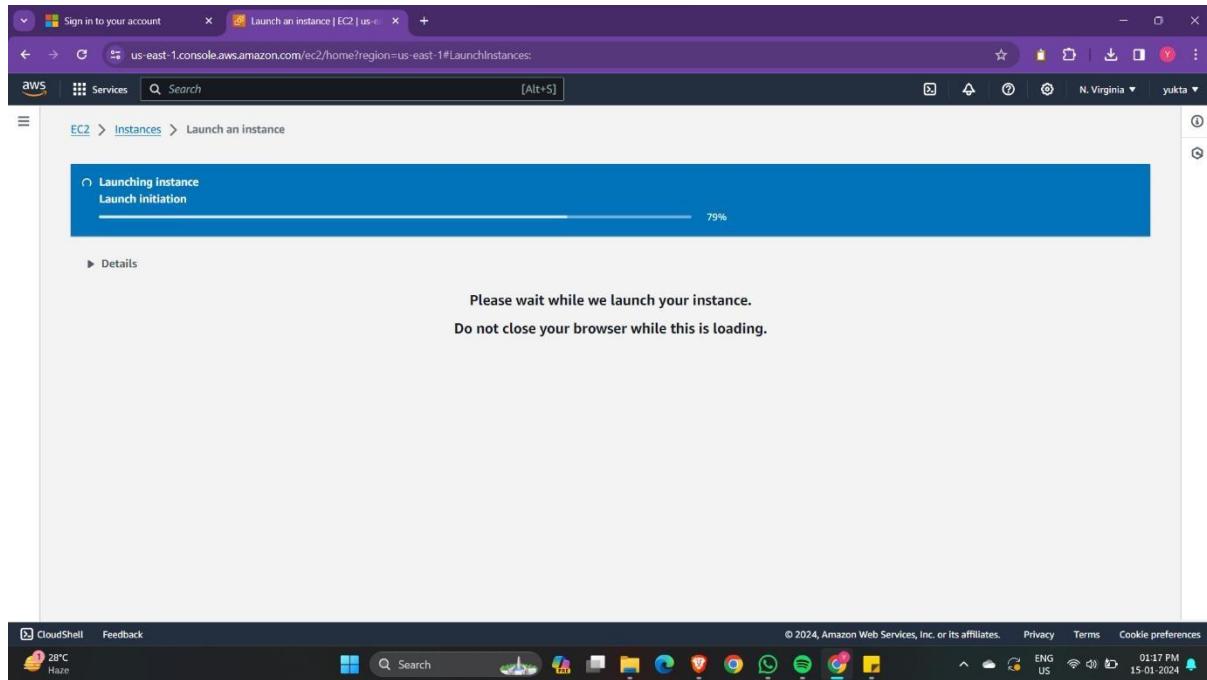
Step 10 : The demokey pair is saved in your system and select the key you created is visible , now select it.



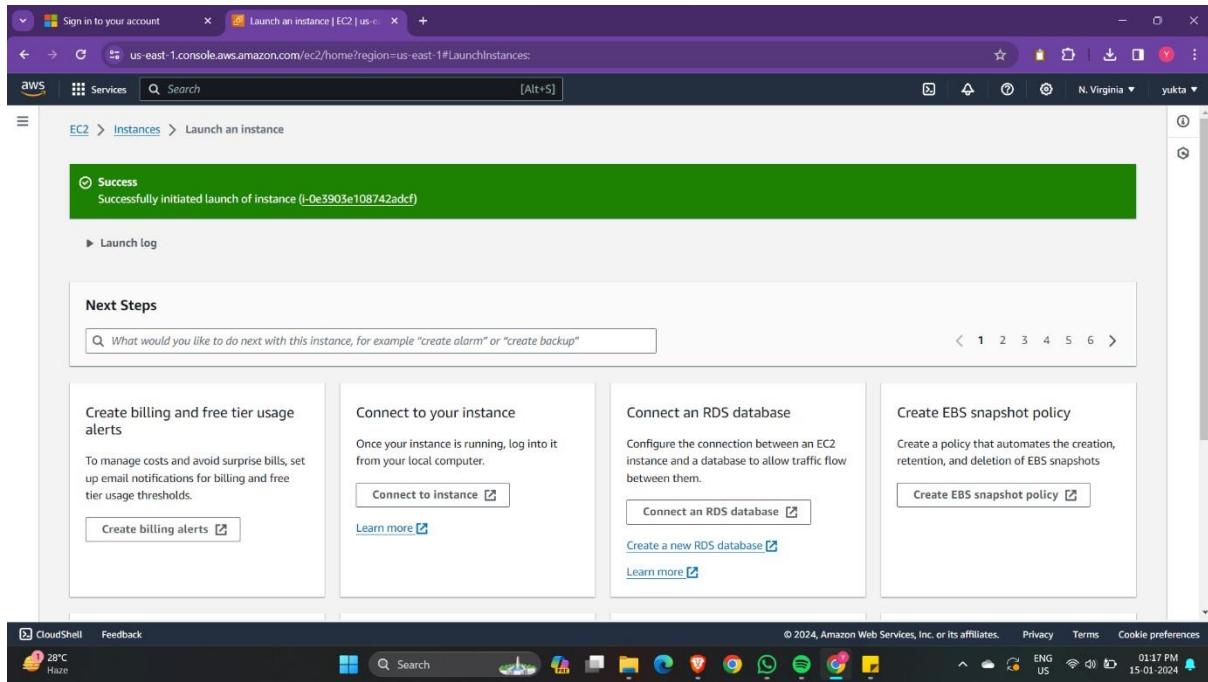
Step 11: now select the both the check boxes of allow https traffic from internet.



Step 12: In configure storage options change it from 8 gib to 20 gib or to the amount of storage you require and launch the instance.



Step 13: The instance is now launched successfully and select view all instances to view it.

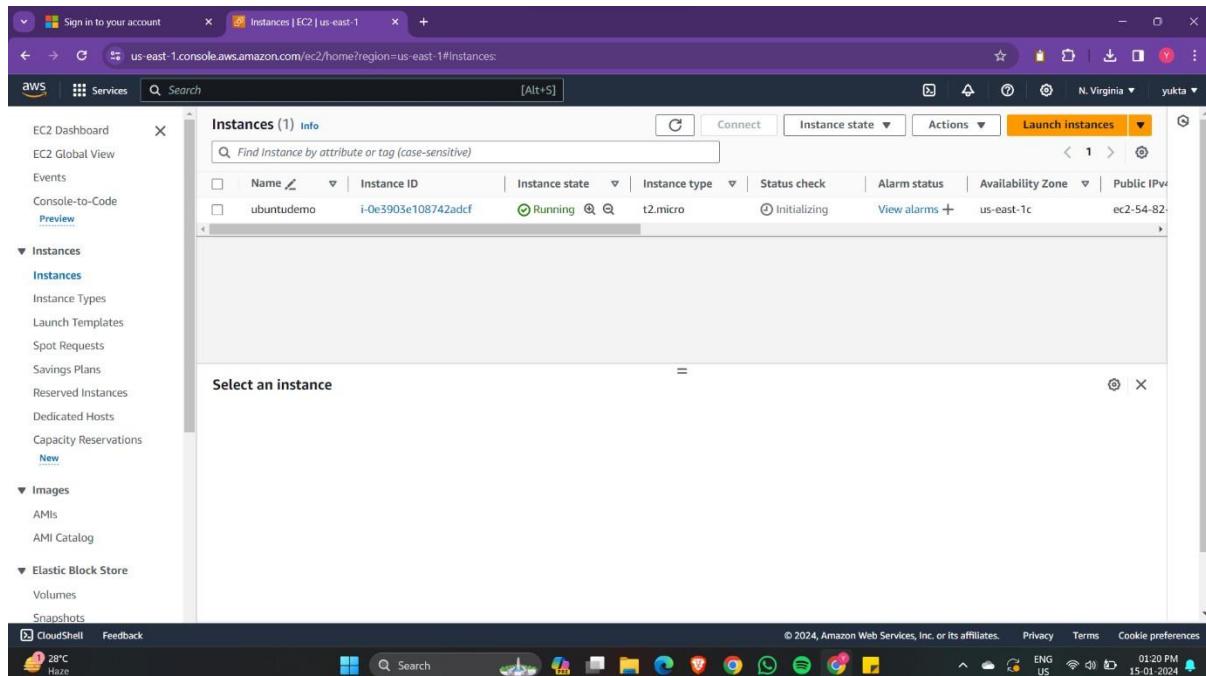


Experiment 13:

Working with AWS

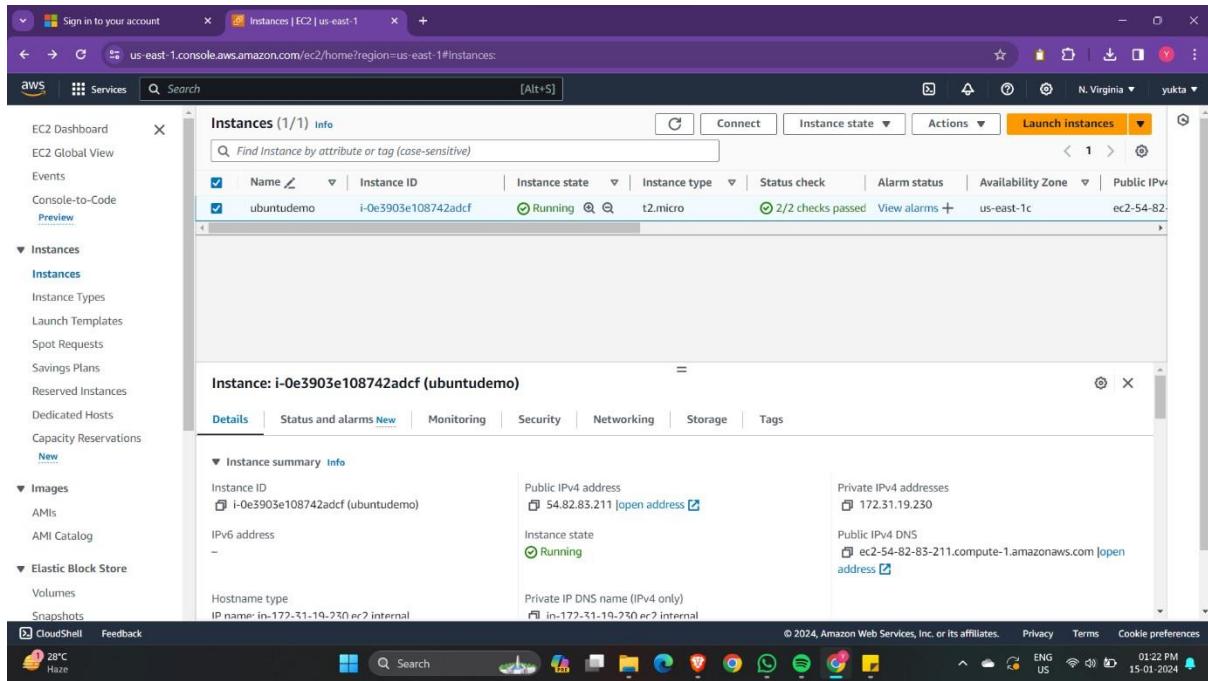
Access Nginx server using EC2 instance with Browser

Step 14: The instance `ubuntudemo` which we created is now displayed and the instance state is shown as running.



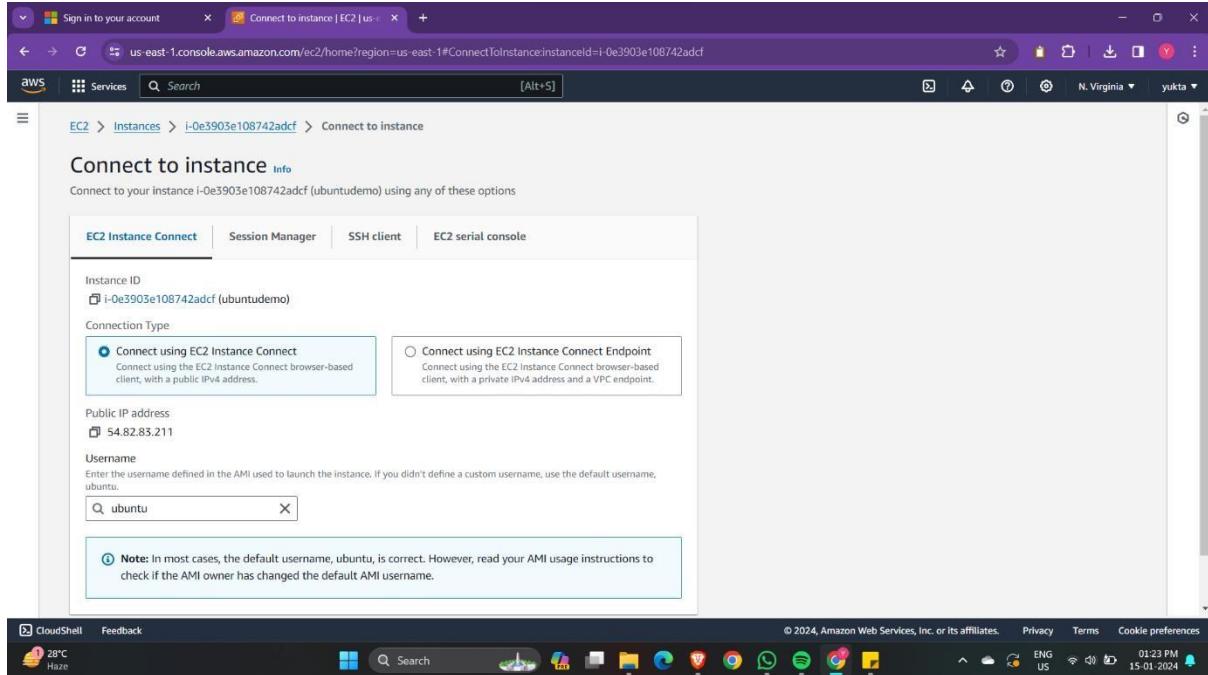
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Images, and Elastic Block Store. The main content area displays a table titled "Instances (1) Info". The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. One row is listed: "ubuntudemo" (Instance ID: i-0e3903e108742adcf, Instance state: Running, Instance type: t2.micro, Status check: Initializing, Availability Zone: us-east-1c, Public IPv4: ec2-54-82). Below the table, a modal window titled "Select an instance" is open, showing the same instance information. At the bottom of the screen, there's a Windows taskbar with various icons and system status indicators.

Step 15: wait till the status check gets 2 checks.



Step 16: right click on the instance name to view the options and select connect instance and the connect to instance windows opens.

Enter the username(ubuntu) and connect to the instance.



Step 17: the instance connect terminal is launched. Type sudo apt update.

A screenshot of a terminal window titled 'EC2 Instance Connect | us-east-1'. The terminal shows the output of the 'sudo apt update' command. The output includes system usage statistics, a note about ESM Apps, and a warning about the age of available updates. It also includes the standard Ubuntu license and warranty information. The terminal prompt is 'ubuntu@ip-172-31-19-230:~\$'. At the bottom of the terminal, it says 'i-0e3903e108742adcf (ubuntudemo)' and 'Public IPs: 54.82.83.211 Private IPs: 172.31.19.230'. The terminal is part of a larger AWS CloudShell session.

Step 18: now type sudo apt install ngnix.

```

Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [388 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1062 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [208 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1244 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [203 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [826 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [155 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 28.9 MB in 5s (5421 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-19-230:~$ sudo apt install nginx

```

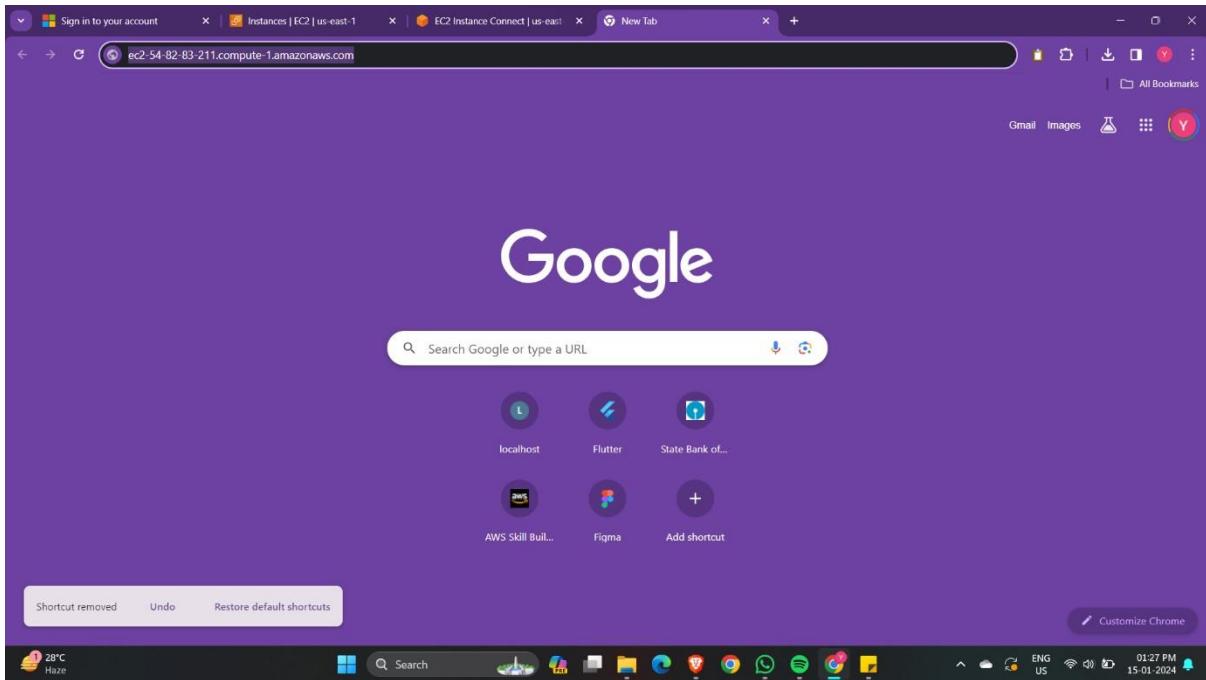
i-0e3903e108742adcf (ubuntudemo)
PublicIPs: 54.82.83.211 PrivateIPs: 172.31.19.230

Step 19: go back to the instances window and copy the public IPV4 address DNS.

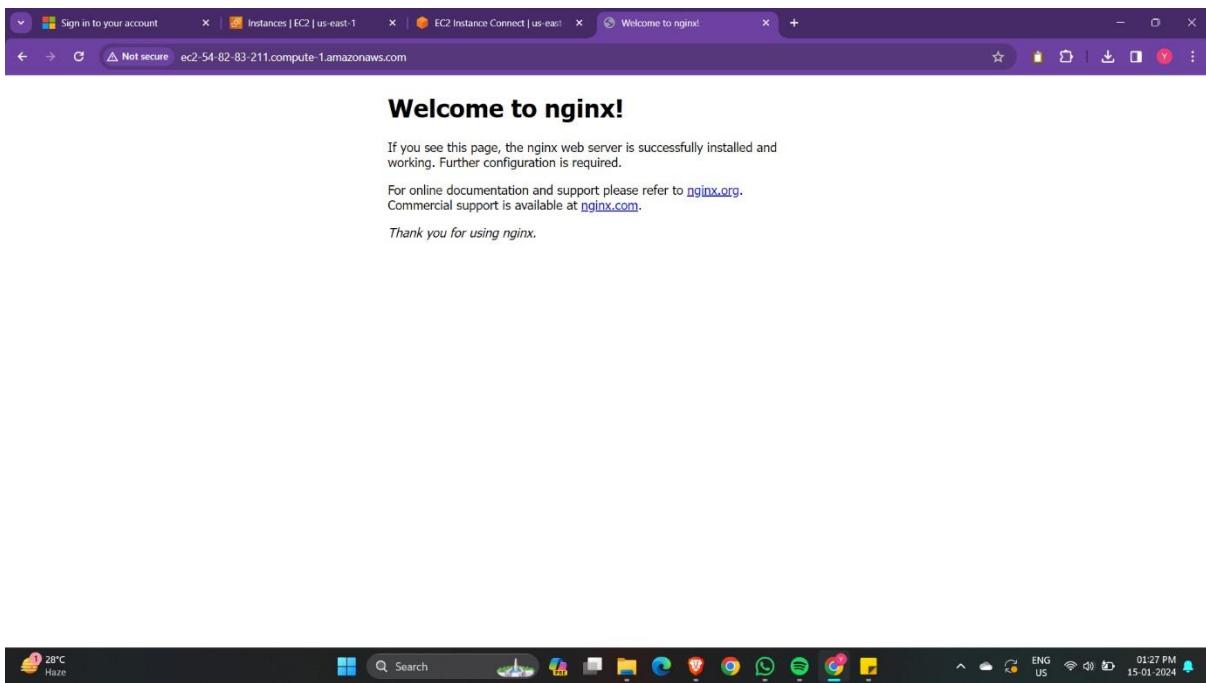
Instances (1/1) Info	
<input checked="" type="checkbox"/> Name	Instance ID
<input checked="" type="checkbox"/> ubuntudemo	i-0e3903e108742adcf
	Running
	t2.micro
	2/2 checks passed
	View alarms +
	us-east-1c
	ec2-54-82-

Instance: i-0e3903e108742adcf (ubuntudemo)	
Details	Status and alarms New
Instance summary	Info
Instance ID	Public IPv4 address 54.82.83.211 [open address]
IPv6 address	Private IPv4 addresses 172.31.19.230
Hostname type	Public IP DNS name (IPv4 only) ec2-54-82-83-211.compute-1.amazonaws.com [open]
	Copy public IPv4 DNS to clipboard

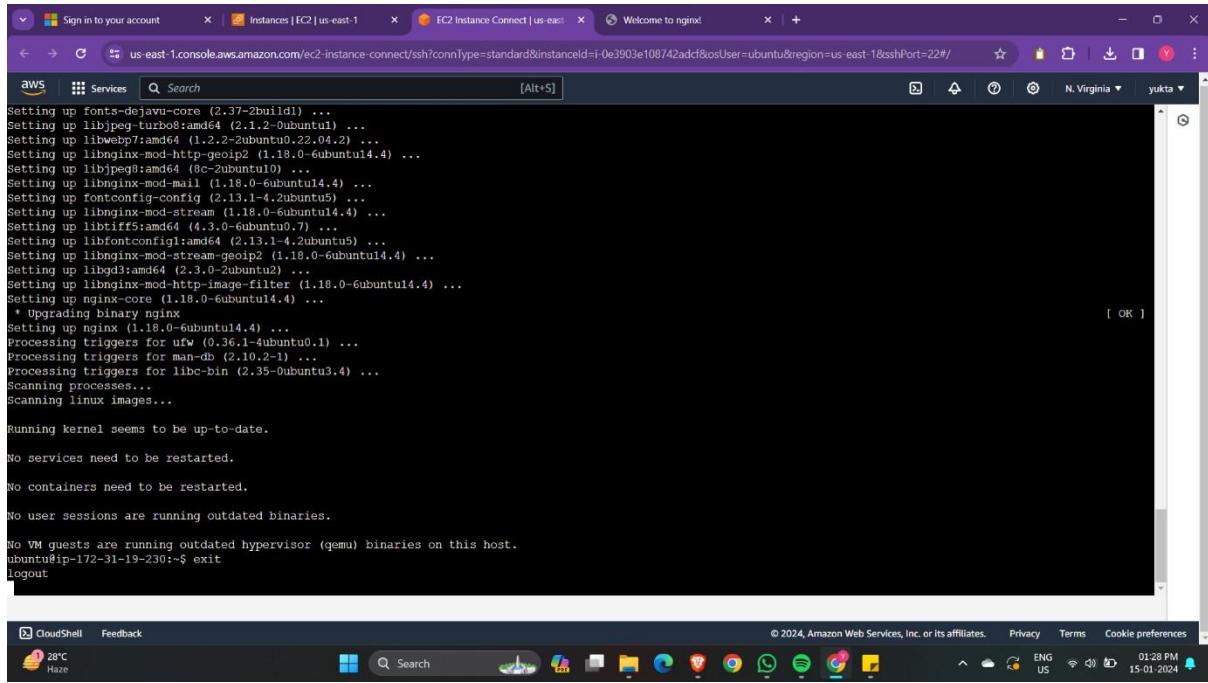
Step 20: open a new tab in the browser and paste the address you copied.



Step 21: welcome to nginx window is displayed.



Step 22: now go back the to EC2 instance connect window and type exit



```
Setting up fonts-dejavu-core (2.37-2build1) ...
Setting up libjpeg-turbo8:amd64 (2.1.2-0ubuntu1) ...
Setting up libwebp7:amd64 (1.2.2-0ubuntu0.22.04.2) ...
Setting up libnginx-mod-http-geoip2 (1.18.0-6ubuntu14.4) ...
Setting up libjpeo8:amd64 (9c-2ubuntu10) ...
Setting up libnginx-mod-mail (1.18.0-6ubuntu14.4) ...
Setting up fontconfig-config (2.13.1-4.2ubuntu5) ...
Setting up libnginx-mod-stream (1.18.0-6ubuntu14.4) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu0.7) ...
Setting up libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Setting up libnginx-mod-stream-geoip2 (1.18.0-6ubuntu14.4) ...
Setting up libgd3:amd64 (2.3.0-2ubuntu2) ...
Setting up libnginx-mod-http-image-filter (1.18.0-6ubuntu14.4) ...
Setting up nginx-core (1.18.0-6ubuntu14.4) ...
 * Upgrading binary nginx
Setting up nginx (1.18.0-6ubuntu14.4) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (gemu) binaries on this host.
ubuntu@ip-172-31-19-230:~$ exit
logout
```

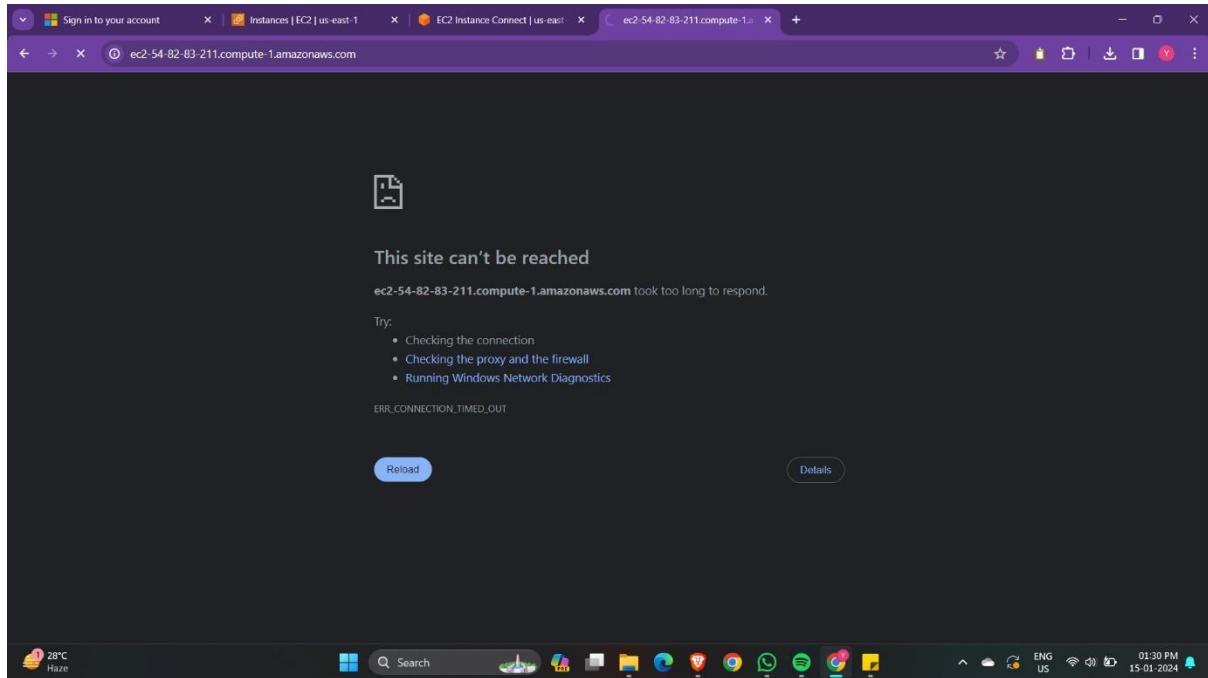
Step 23: go back to the instances window right click on the instance name (ubuntu) to display the options and select stop instance to stop it

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists various services like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, and Elastic Block Store. The Instances section is expanded, showing sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and AMIs. The main content area displays a table of instances. One instance, named 'ubuntudemo', is selected. Its details are shown in a modal window. The instance is currently 'Running' (t2.micro), has 2/2 checks passed, and is located in the 'us-east-1c' availability zone. It has a public IPv4 address of 54.82.83.211 and a private IP4 address of 172.31.19.230. The instance was launched on 2024-01-15 at 01:28 PM.

Step 24: the instance is stopped.

This screenshot sequence illustrates the process of stopping an EC2 instance. The first part shows the 'Stop instance?' confirmation dialog for the instance 'ubuntudemo'. The second part shows the 'Successfully stopped' message. The third part shows the instance status as 'Stopped' in the EC2 Instances table. The instance's public IP address is now listed as 'Stopped' under the 'Instance state' column.

Step 25: go back to the nginx page and reload it.



**SOFTWARE REQUIREMENT SPECIFICATION
FOR
ONLINE BANKING SYSTEM**

Version 1.0 approved

Prepared by:

Sandeep Bhukya - 21BD1A0547

Krishna Teja C - 21BD1A054Z

Katukam Harshith - 21BD1A054R

Aryan Chandra - 21BD1A055B

Keshav Memorial Institute Of Technology

Problem Proposition:

We are faced with the challenge of creating a reliable and user-friendly feature for generating digital bank statements inside the framework of our online banking system.

Key Points:

1. Inconsistent User Journey:

- Users may encounter inconsistencies when navigating through various sections of the online banking system, leading to a disjointed experience.
- User accessibility issues cause customers to become frustrated and spend less time using our online banking platform by making it difficult for them to access and browse their transaction history.

2. Customization of bank statements:

- Due to a lack of options, clients are unable to alter the style and content of their bank statements to meet their unique requirements.
- Some platforms lack essential features that customers expect, such as bill pay options, financial goal tracking, or personalized financial advice.

Objectives:

- a. Improve the user experience** by simplifying the interface, enhancing navigation, and adding relevant features. A more user-friendly design that enables customers to quickly access and download their bank statements would improve the user experience.
- b. Ensure a consistent and user-friendly journey** across all aspects of the online banking system.

Expected Outcomes:

- a. Improved user experience** and user-friendly interface.
- b. New features** such as financial goal tracking and personalized financial advice.

Conclusion:

Emphasizing the need to enhance the user experience with a satisfying online banking platform.

Online Banking System

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the development of the Online Banking System. This document serves as a comprehensive guide for stakeholders, including developers, designers, and project managers, to understand the functionality and constraints of the system.

1.2 Scope

The scope of this project encompasses the design, development, testing, and deployment of the Online Banking System. It covers both the front-end and back-end components of the system, including user interfaces, databases, and external integrations.

1.3 Definitions, Acronyms, and Abbreviations

- **Online Banking System (OBS):** Refers to the software solution described in this document.
- **User:** Any individual who accesses the online banking system for banking services.
- **Admin:** A user with administrative privileges responsible for system management and user management.

1.4 References

1. [ISO/IEC 27001:2013 - Information technology — Security techniques — Information security management systems — Requirements.](#)
2. [PCI Data Security Standard \(PCI DSS\) Version 3.2.](#)
3. [Reserve Bank of India \(RBI\) Guidelines for Internet Banking in India.](#)
4. ["Responsive Web Design" by Ethan Marcotte - A List Apart, May 25, 2010.](#)
5. [User Interface Design Principles - Nielsen Norman Group.](#)

2. Problem Statement

2.1 The Problem

The current online banking system lacks essential features and experiences several usability issues, making it difficult for users to perform their banking tasks efficiently and securely. Customers have reported inconsistent user journeys, limited customization options for digital bank statements, and missing features such as bill payment options, financial goal tracking, and personalized financial advice.

2.2 Context

The Online Banking System is a critical component of our banking services, serving a large customer base. In the competitive banking industry, it is imperative to provide a seamless and feature-rich online banking experience to meet customer expectations.

2.3 Impact

The existing issues within the system have resulted in reduced user satisfaction, increased customer support requests, and a potential loss of customers to competing banks with more advanced online banking solutions.

2.4 Objectives

The objectives of this project are as follows:

- a. Improve the user experience** by simplifying the interface, enhancing navigation, and adding relevant features.
- b. Ensure a consistent and user-friendly journey** across all aspects of the online banking system.
- c. Implement customization options** for digital bank statements.
- d. Integrate bill payment functionality, financial goal tracking, and personalized financial advice features.**

2.5 Scope and Boundaries

The scope includes the design and development of the Online Banking System with the specific features mentioned in the objectives. It does not include hardware procurement or physical security measures at bank branches.

2.6 Stakeholders

The primary stakeholders for this project are:

- Bank customers (end-users)
- Bank employees (admins)
- Project managers and development team

2.7 Constraints

The project must adhere to regulatory compliance requirements, including data security and privacy regulations. It is also constrained by budget and time limitations.

2.8 Assumptions

It is assumed that users have basic knowledge of online banking and will have access to the internet and a compatible device. Additionally, network connectivity is assumed to be stable during typical usage.

2.9 Dependencies

The project depends on third-party services for certain features, such as payment processing and financial data aggregation.

3. Functional Requirements

3.1 User Registration and Authentication

Requirement: Users must register with a valid email and password. Two-factor authentication (2FA) via SMS or email should be implemented.

3.2 Account Management

Requirement: Users should be able to view account balances, transaction history, and download statements. They can request account closures and temporary freezes online.

3.3 Funds Transfer

Requirement: Users can transfer funds between their own accounts and to other accounts within the same bank. International transfers require additional verification steps and have daily limits.

3.4 Bill Payments

Requirement: Users can pay bills to registered merchants. Payment scheduling and recurring payments should be supported. Confirmation receipts are sent via email.

3.5 Transaction History

Requirement: Users can search and filter transaction history based on dates, amounts, and transaction types. The system retains at least 12 months of transaction data.

3.6 Alerts and Notifications

Requirement: Users can set up transaction alerts for specific amounts or account activities. Notifications are sent via SMS, email, or in-app notifications.

3.7 Customer Support

Requirement: Users can initiate support requests through the system. A chatbot provides initial assistance, and complex issues are escalated to human agents. Response time should be within 24 hours.

3.8 Security Features

Requirement: The system employs encryption for data transmission and storage. Regular security audits and vulnerability assessments are conducted. Failed login attempts are limited, and suspicious activities trigger account lockouts.

4. Non-Functional Requirements

4.1 Performance

Requirement: The system should handle at least 10,000 concurrent users without performance degradation. Response time for common actions should be within 2 seconds.

4.2 Security

Requirement: The system must comply with industry standards like PCI DSS. Data backups should be encrypted and stored off-site. Regular security training for employees is mandatory.

4.3 Usability

Requirement: The user interface must be intuitive and accessible. The system should be compatible with major web browsers and responsive for various devices (desktops, tablets, and smartphones).

4.4 Reliability

Requirement: The system should have a 99.9% uptime. Failover mechanisms and backup servers should ensure service continuity. Scheduled maintenance should be communicated to users in advance.

5. External Interface Requirements

5.1 User Interfaces

Requirement: The user interface should follow a clean and consistent design. It should be compatible with popular web browsers such as Chrome, Firefox, Safari, and Edge.

5.2 Integration with Existing Systems

Requirement: The online banking system should integrate with the bank's core banking system for real-time transaction processing. APIs should be available for third-party integrations.

5.3 Third-party Services

Requirement: Integration with payment gateways, credit bureaus, and fraud detection services is required. These integrations should be secure and comply with regulatory standards.

6. System Features

6.1 User-Friendly Digital Bank Statement Generation

Requirement: Users can customize their digital bank statements by selecting date ranges, transaction types, and preferred file formats (PDF, CSV). Customized statements can be downloaded instantly.

6.2 Consistent User Experience

Requirement: The user interface should maintain consistency across all modules. Navigational elements, color schemes, and typography should be uniform.

6.3 Additional Features

6.3.1 Bill Payment Integration

Requirement: Users can add payees, schedule one-time or recurring payments, and receive reminders for upcoming bills. Payment status and history are accessible.

6.3.2 Financial Goal Tracking Module

Requirement: Users can set financial goals (e.g., saving for a vacation) with target amounts and deadlines. The system provides progress reports and suggestions for achieving goals.

6.3.3 Personalized Financial Advice Module

Requirement: Users can input financial data, and the system provides personalized advice on budgeting, investments, and loan management. Advice is generated based on user profiles and financial goals.

7. Appendices

7.1 Glossary

- Administrator (Admin): A user role with elevated privileges responsible for system management, user management, and administrative tasks.
- Two-Factor Authentication (2FA): A security process in which a user provides two different authentication factors to verify their identity.
- API: Abbreviation for Application Programming Interface, a set of rules and protocols that allows different software applications to communicate with each other.
- PCI DSS: Abbreviation for Payment Card Industry Data Security Standard, a set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.

7.2 References

1. [ISO/IEC 27001:2013 - Information technology — Security techniques — Information security management systems — Requirements.](#)

2. [PCI Data Security Standard \(PCI DSS\) Version 3.2.](#)
3. [Reserve Bank of India \(RBI\) Guidelines for Internet Banking in India.](#)
4. ["Responsive Web Design" by Ethan Marcotte - A List Apart, May 25, 2010.](#)
5. [User Interface Design Principles - Nielsen Norman Group.](#)