

A Lab Manual
on
DEVOPS LAB
R22
(III- B. Tech. – II– Semester)
Submitted to

DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING

By
Dr Ganga Sanuvala
(Asst Professor, Dept. of CSE)

Mr. A. Sarveswar Reddy
(Asst Professor, Dept. of CSE)

Mr. Maheedhar Varaprasad
(Asst Professor, Dept. of CSE)

Mr. S. Malli Babu
(Asst Professor, Dept. of CSE)



CMR INSTITUTE OF TECHNOLOGY
Kandlakoya(V), Medchal Road, Hyderabad – 501 401
Ph. No. 08418-222042, 22106 Fax No. 08418-222106

(2024-25)

CONTENTS

Sl. No.	Particulars	Page No.
1.	Vision and Mission	2
2.	Syllabus	4
3.	Student Entry Behavior or Pre-requisites	5
4.	Course Outcomes	6
5.	Mapping of Course with PEOs-POs	7
6.	Mapping Of Course Outcomes with POs	9
7.	Direct Course Assessment	10
8.	Indirect Course Assessment	11
9.	Overall Course Assessment and Attainment level	13
10.	Pi diagrams, Bar charts, Histograms for representing results	14
11.	Lesson/Course Plan	15
12.	Programs	16

1. Vision and Mission of CMR Institute of Technology

I. Vision, Mission and Quality Policy

Vision: To create world class technocrats for societal needs.

Mission: Achieve global quality technical education by assessing learning environment through

- Innovative Research & Development
- Eco-system for better Industry institute interaction
- Capacity building among stakeholders

Quality Policy: Strive for global professional excellence in pursuit of key-stakeholders.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

Vision: Develop competent software professionals, researchers and entrepreneurs to serve global society.

Mission: The department of Computer Science and Engineering is committed to

- create technocrats with proficiency in design and code for software development
- adapt contemporary technologies by lifelong learning and face challenges in IT and ITES sectors
- quench the thirst of knowledge in higher education, employment, R&D and entrepreneurship

B.Tech. - Computer Science and Engineering (CSE)

I. Programme Educational Objectives (PEOs): Engineering Graduates will

1. Pursue successful professional career in IT and IT-enabled sectors.
2. Pursue lifelong learning skills to solve complex problems through multidisciplinary-research.
3. Exhibit professionalism, ethics and inter-personal skills to develop leadership qualities.

II. Programme Specific Outcomes (PSOs): Engineering Graduates will be able to

1. Design and Develop Software Systems using appropriate SDLC Models.
2. Apply cutting-edge technologies to solve real world problems.

2. Syllabus

Devops LAB

Course	B.Tech.-VI-Sem.	L	T	P	C
Course Code	22CDPC66	-	-	2	1

Course Outcomes (COs) & CO-PO Mapping (3-Strong; 2-Medium; 1-Weak Correlation)

COs	Upon completion of course the students will be able to	PO4	PO5	PO9	PSO2
CO1	identify DevOps workflow	3	3	3	3
CO2	use eclipse and Jenkins for DevOps	3	3	3	3
CO3	develop docker image	3	3	3	3
CO4	take part in grid deployment	3	3	3	3
CO5	make use of monitoring, operations tools in DevOps	3	3	3	3

List of Experiments

Week	Title/Experiment
1	Start DevOps with a workflow that includes four phases: to do, in progress, code review, and done.
2	Setup Eclipse for DevOps.
3	Jenkins Setup on AWS.
4	Build WAR file in DevOps.
5	Ansible Setup and SSH keys.
6	Deploy the artifact on the Test Server.
7	Perform automation using Jenkins.
8	Build and deploy a grid for Chrome and Firefox based testing.
9	Create deployment resource using Kubernetes.
10	Create a docker image for any application using Docker file and push it to Docker hub.
11	Setup Grafana for Devops.
12	Setup Prometheus for Devops.

References	
1.	DevOps Lab Manual, Department of CSE, CMRIT, Hyd.
2.	https://www.udemy.com/course/practical-devops-for-beginners/

Micro-Projects: Student should submit a report on one of the following/any other micro-project(s) approved by the lab faculty before commencement of lab internal examination.
<ol style="list-style-type: none"> 1. Deploy a Containerized Web Application. 2. Develop a Version Control System/Tool: GIT. 3. Create a Monitoring Dashboard for any Application. 4. Implement a Continuous Integration/Continuous Delivery (CI/CD) Pipeline for an application. 5. Implement DevOps Lifecycle with Amazon Web Services (AWS). 6. Build a Scalable Application with Docker. 7. Create a Jenkins project that connects to a remote Jenkins server and controls it. 8. Deploy an application (with high availability) with a database 9. Create a Continuous Delivery of a Java Web Application. 10. Build and execute a selenium project.

3. Student Entry Behavior or Pre-requisites

- Students should have basic knowledge on Linux commands
- Students should have basic knowledge on basic programming.
- Student should have knowledge on software engineering concepts
- These prerequisites are taken by the students during the first two years. However during the initial sessions the topics are reviewed.

4. Course Outcomes

Course Outcome	Course Outcome Statements
CO - 1	identify DevOps workflow
CO - 2	use eclipse for DevOps
CO - 3	develop docker image
CO - 4	take part in grid deployment
CO - 5	make use of Jenkins framework in DevOps

5. Mapping of Course with PEOs-POs

(Only Ticking)

Program Educational Objectives (PEOs)

Sl. No.	PEOs Name	Program Education Objective Statements
1	PEO - 1	Pursue successful professional career in IT and IT-enabled sectors. [PO's: 1,2,3,4,5,7,8,9,10,11 and 12] [PSO's: 1 and 2]
2	PEO – 2	Pursue lifelong learning skills to solve complex problems through multidisciplinary-research. [PO's: 1,2,3,4,5,6,7,8,9,10 and 12] [PSO's: 1, 2]
3	PEO – 3	Exhibits professionalism, ethics and inter-personal skills to develop leadership qualities. [PO's: 1,2,3,4,5,6,7,8,9,10,11 and 12] [PSO's: 2]

Program Outcomes (POs)

PO Name	Graduate Attributes	PO Statements
PO1	Engineering knowledge	Apply mathematics, science, engineering fundamentals to solve complex engineering problems. [PEO's: 1,2 and 3] [PSO's: 1,2]
PO 2	Problem analysis	Identify, formulate and analyze complex engineering problems to reach substantiated conclusions. [PEO's: 1,2 and 3] [PSO's: 1,2]
PO 3	Design/ development of solutions	Design and develop a component/system/process to solve complex societal engineering problems. [PEO's: 1,2 and 3] [PSO's: 1,2]
PO 4	Conduct investigations of complex problems	Design and conduct experiments to analyze, interpret and synthesize data for valid conclusions. [PEO's: 1,2 and 3] [PSO's: 1,2]
PO 5	Modern tool usage	Create, select and apply modern tools, skills, resources to solve complex engineering problems. [PEO's: 1,2 and 3] [PSO's: 1,2]
PO 6	The engineer and society	Apply contextual engineering knowledge to solve societal issues. [PEO's: 2 and 3]
PO 7	Environment and sustainability	Adapt modern engineering practices with environmental safety and sustainable development. [PEO's: 1,2 and 3]
PO 8	Ethics	Apply professional code of ethics, responsibilities and norms in engineering practices. [PEO's: 1,2 and 3] [PSO's: 2]
PO 9	Individual and team work	Compete as an individual and/or as a leader in collaborative cross cultural teams. [PEO's: 1,2 and 3]
PO 10	Communication	Communicate effectively through technical reports, designs, documentations and presentations. [PEO's: 1,2 and 3] [PSO's: 2]
PO 11	Project management and finance	Endorse cognitive management skills to prepare project report using modern tools and finance. [PEO's: 1 and 3] [PSO's: 2]
PO 12	Life-long learning	Engage in independent and life-long learning in the broad context of technological changes. [PEO's: 1,2 and 3] [PSO's: 1,2]

6. Mapping Of Course Outcomes With POs

No	Course Outcomes	Po1	Po2	Po3	Po4	Po5	Po6	Po7	Po8	Po9	Po10	Po11	Po12	Avg
1	CO - 1				3	3				3				
2	CO – 2				3	3				3				
3	CO – 3				3	3				3				
4	CO – 4				3	3				3				
5	CO – 5				3	3				3				
	Avg				3	3				3				

7. Direct Course Assessment

(As mentioned in following table of 10 parameters, of which consider only the parameters required for this courses)

No	Description	Targeted Performance	Actual Performance	Remarks	Course Attainment
1	Internal Marks(25)	80% of Students(182 Students) should Secure 60% of Internal Marks i.e., 15 Marks			
2	External Marks(50)	80% of Students(182 Students) should Secure 70% of External Marks i.e., 35 Marks			
3	Clearing of Subject	A minimum of 95% of Students(216 Students) should clear this course in first attempt			
4	Getting First Class	90% of Students(205 Students) should Secure I Class Marks i.e., 45 Marks in my course			
5	Distinction	80% of Students (182 Students) should secure First Class With Distinction i.e., 53 Marks in my course			
6	Outstanding Performance	60% of Students (137 Students) should secure 80% and above Marks. i.e., 60 Marks in my course			

8. Indirect Course Assessment

(As mentioned-strong (3), moderate (2), weak (1) & no comment (0))

Mission Statement of CSE

- **Impart fundamentals through state of art technologies for research and career in Computer Science & Engineering.**
- **Create value-based, socially committed professionals for anticipating and satisfying fast changing societal requirements.**
- **Foster continuous self learning abilities through regular interaction with various stake holders for holistic development.**

Correlation of Mission Elements with Mission Statement of CSE Department related to the Course (only Ticking given by faculty)

No	Mission Elements	Strong	Moderate	Weak	No Comment
M-1	Impart Fundamentals	√			
M-2	State Of Art Technologies	√			
M-3	Research & Career Development	√			
M-4	Value based Socially Committed Professional	√			
M-5	Anticipating & Satisfying Industry Trends		√		
M-6	Changing Societal Requirements			√	
M-7	Foster Continuous Learning	√			
M-8	Self-Learning Abilities	√			
M-9	Interaction with stakeholders	√			
M-10	Holistic Development		√		

Indirect Course Assessment through Student Satisfaction Survey

(Note for *: Parameters used for course teaching like

a: Classroom teaching	b: Simulations	c:labs	d: Mini_Projects
e: Major Projects	f: Conferences	g: professional activities	
h: Technical Clubs	i: Guest Lectures	j: Workshops	k: Technical Fests l:Tutorials
m:NPTLs	n:Digital Library	o: Industrial Visits	p: software Tools
Internship/training	r:Technical Seminars		
s: NSS	t: NSS	u: sports etc.	q:

No	Question Based on PEO/ PO/PSO/CO	Parameters (a /b /c.../)*	Strong (3)	Moderate (2)	Weak (1)	No comment (0)
1	Did the course impart fundamentals through interactive learning and contribute to core competence?					
2	Did the course provide the required knowledge to foster continuous learning?					
3	Whether the syllabus content anticipates & satisfies the industry and societal needs?					
4	Whether the course focuses on value based education to be a socially committed professional?					
5	Rate the role of the facilitator in mentoring and promoting the self learning abilities to excel academically and professionally?					
6	Rate the methodology adopted and techniques used in teaching learning processes?					
7	Rate the course in applying sciences & engineering fundamentals in providing research based conclusions with the help of modern tools?					
8	Did the course have any scope to design, develop and test a system or component?					
9	Rate the scope of this course in addressing cultural, legal, health, environment and safety issues?					
10	Scope of applying management fundamentals to demonstrate effective technical project presentations & report writing?					
	Total					
	Average					
	Total Average					

9. Overall Course Assessment

(80% Direct + 20% Indirect, if any)

No	Assessment Type	Weightage	Attainment Level
1	Direct-Assignment, Quiz, Subjective, University Exams, Results, Bench Marks	0.8	
2	Indirect-Surveys-Questionnaire	0.2	
	Overall		

DEVOPS LAB Course Attainment level:

10. Pi diagrams, Bar charts, Histograms

(For representing previous results, if any)

ICS Pass % for Last 4 Academic Years	Appeared	Passed	Pass%

11. Lesson/Course Plan

Week No.	Name of the Program	Week	Text Books	Mode of Assessment
1	Start DevOps with a workflow that includes four phases: to do, in progress, code review, and done	1	R1	By observations, lab records, viva-voice
2	Setup Eclipse for DevOps.	2	R1	By observations, lab records, viva
3	Jenkins Setup on AWS.	3	R1	By observations, lab records, viva
4	Build WAR file in DevOps.	4	R1	By observations, lab records, viva
5	Ansible Setup and SSH keys.	5	R1	By observations, lab records, viva
6	Deploy the artifact on the Test Server.	6	R1	By observations, lab records, viva
7	Perform automation using Jenkins.	7	R1	By observations, lab records, viva
8	Build and deploy a grid for Chrome and Firefox based testing.	8	R1	By observations, lab records, viva
9	Create deployment resource using Kubernetes.	9	R1	By observations, lab records, viva
10	Create a docker image for any application using Docker file and push it to Docker hub.	10	R1	By observations, lab records, viva
11	Setup Grafana for Devops.	11	R1	By observations, lab records, viva
12	Setup Prometheus for Devops.	12	R1	By observations, lab records, viva

12. Programs

Experiment-1

Aim: Start DevOps with a workflow that includes four phases: to do, in progress, code review, and done.

Require Software& Tools: JIRA, KANBAN.

Procedure:

Phase 1: To Do

- **Objective:** Identify and prioritize tasks or features to be developed.
- **Key Actions:**
 - Define tasks clearly in a backlog.
 - Prioritize tasks based on impact, urgency, and dependencies.
 - Assign owners or teams to each task.

Tools:

Jira, Trello, GitHub Issues, or Asana.

Phase 2: In Progress

- **Objective:** Actively work on tasks selected from the "To Do" phase.
- **Key Actions:**
 - Begin coding or configuring based on task requirements.
 - Update the task status to reflect ongoing work.
 - Ensure team members collaborate effectively (e.g., stand-ups, pair programming).

Best Practices:

- Use branches in version control systems for individual tasks (e.g., Git feature branches).
- Write unit tests alongside development.

Phase 3: Code Review

- **Objective:** Validate the quality, functionality, and security of the code.
- **Key Actions:**
 - Submit pull requests for peer review.
 - Review code for adherence to standards, logic, and potential issues.
 - Approve or request changes.

Tools:

GitHub Pull Requests, GitLab Merge Requests, Bitbucket.

Automation:

Integrate CI/CD pipelines to run tests automatically during reviews.

Phase 4: Done

- **Objective:** Mark tasks as completed and deploy changes if necessary.
- **Key Actions:**
 - Merge the approved code into the main branch.
 - Deploy to staging or production environments.
 - Monitor deployment and validate functionality.

Post-Completion:

- Add documentation for the changes.
 - Gather feedback from stakeholders or users.
-

Workflow Visualization

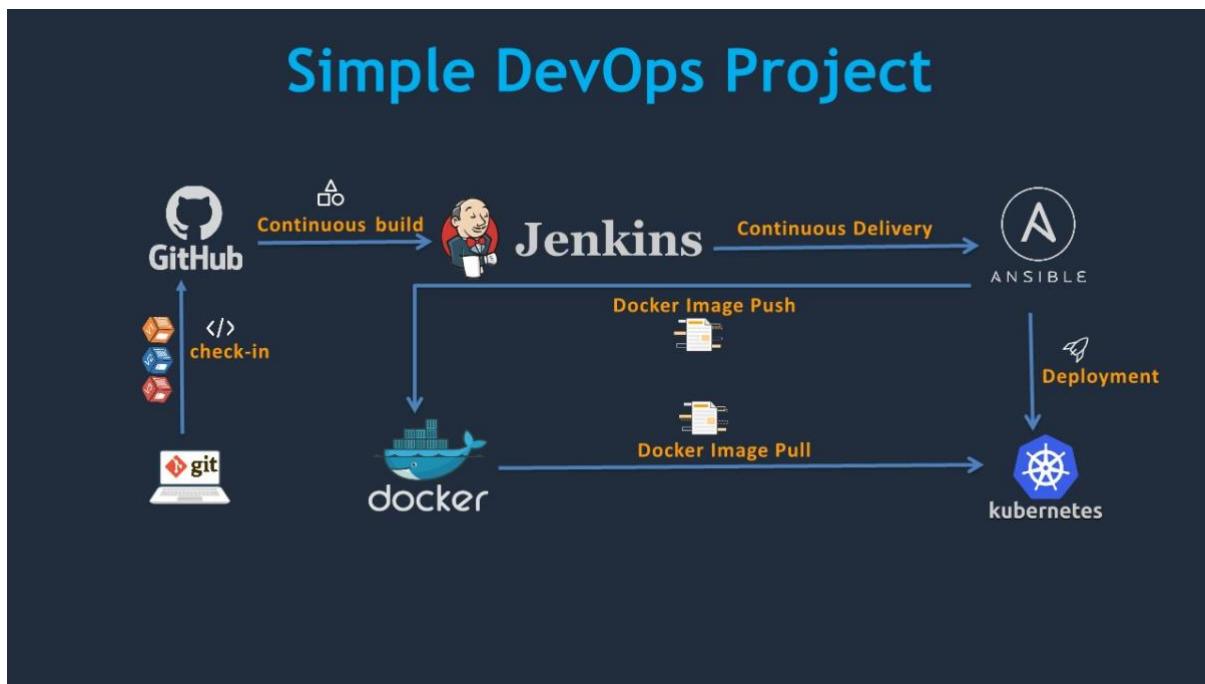
A Kanban board or similar visual representation can help track the status of tasks across these phases. For example:

1. **To Do:** Contains all pending tasks.
2. **In Progress:** Tasks currently being worked on.
3. **Code Review:** Tasks awaiting review or approval.
4. **Done:** Completed and deployed tasks.

Tools:

Trello, Jira, Azure.

Output:-



Experiment-2

Aim: Setups Eclipse for Devops

Note: Write the given steps in your lab manual. The provided image is just for your better understanding.

Require Software& Tools: Eclipse, Java jdk-17, Tomcat v.9, TestNG and Dependencies.

Procedure:

Step-1: Install Jdk-17 and set the java path in System environment

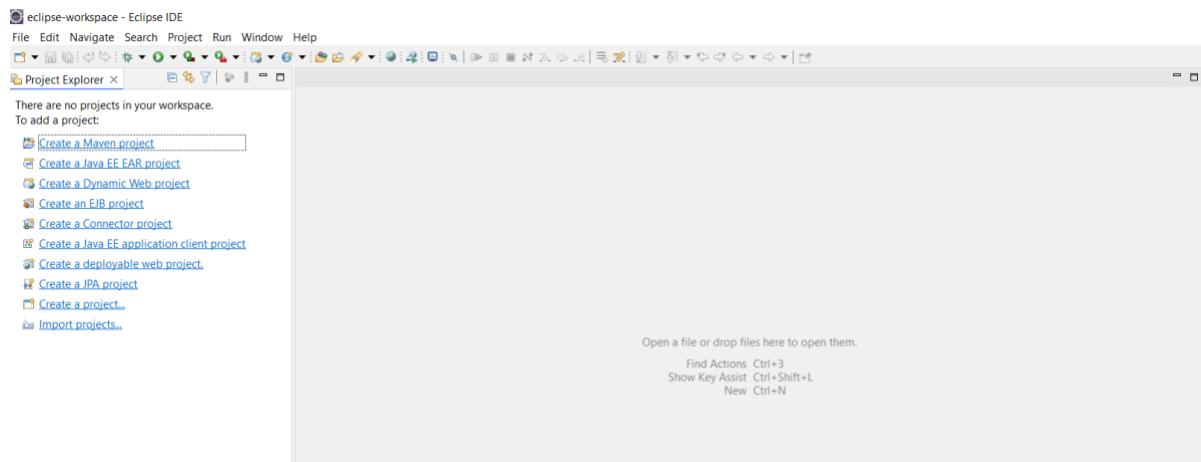
Step-2: Download eclipse zipfile and extract the contents the all eclipse file

Step-3: Create a Maven Project from eclipse as:

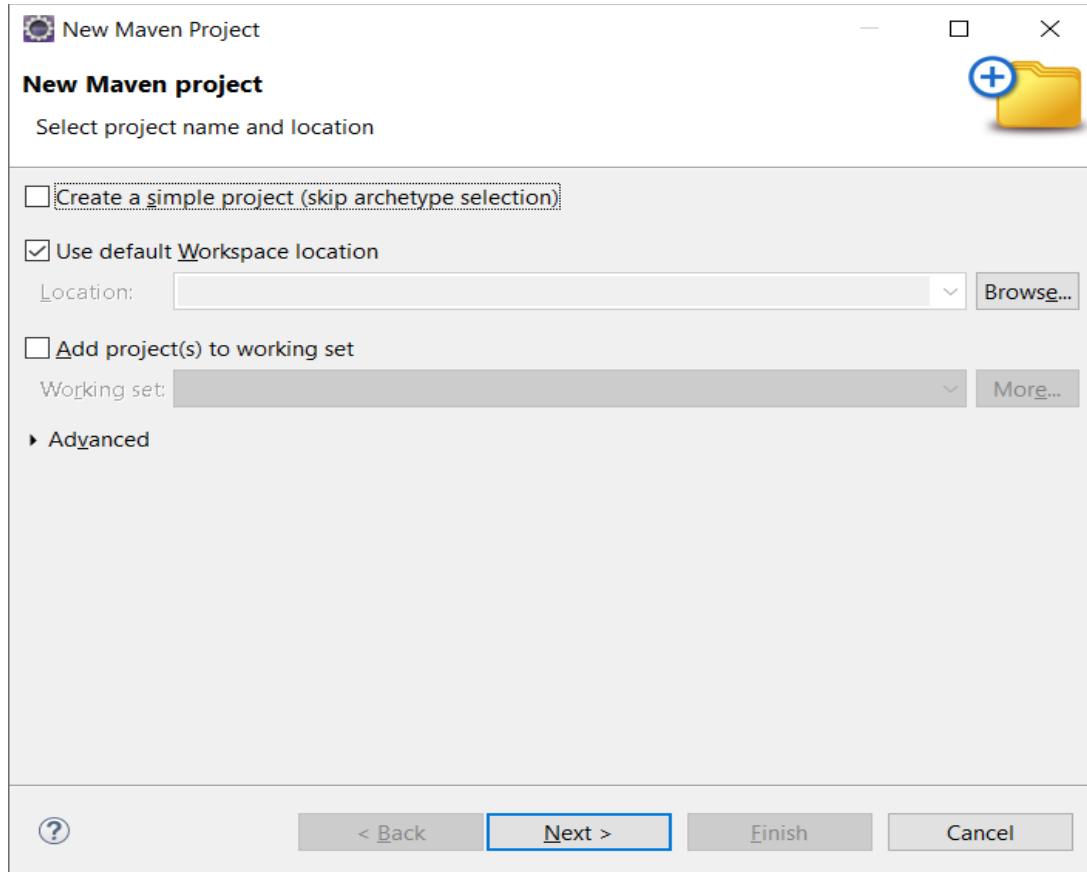
OR

Click on File in left corner -> Click on new -> click on Maven Project and follow the given image steps.

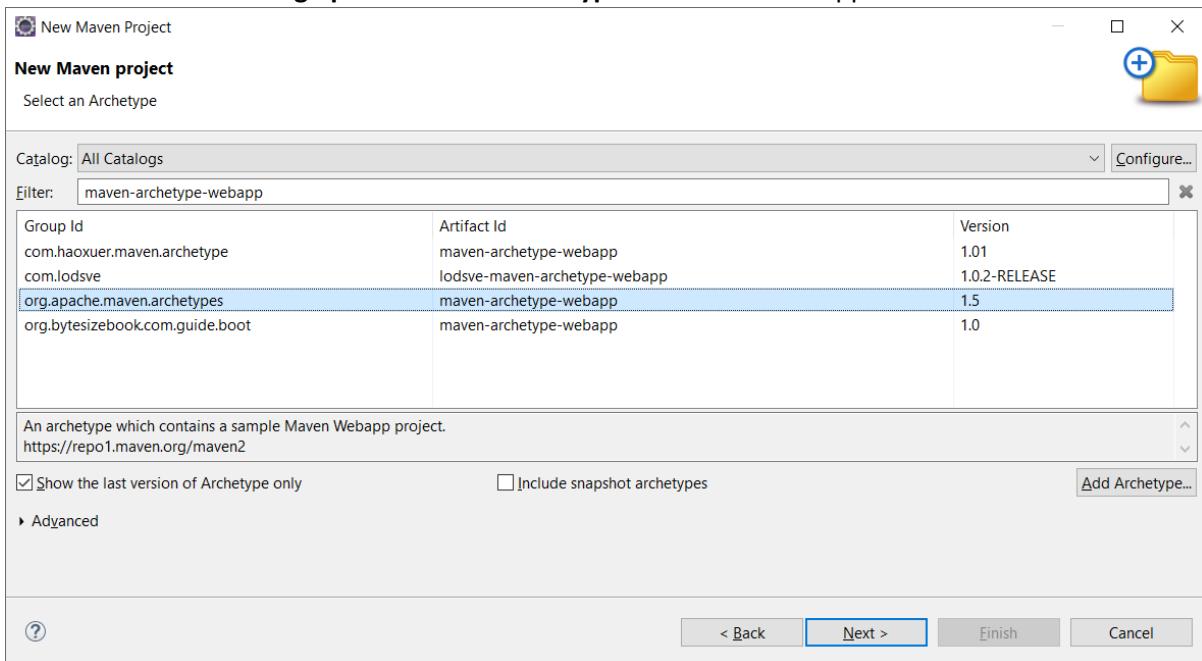
a.



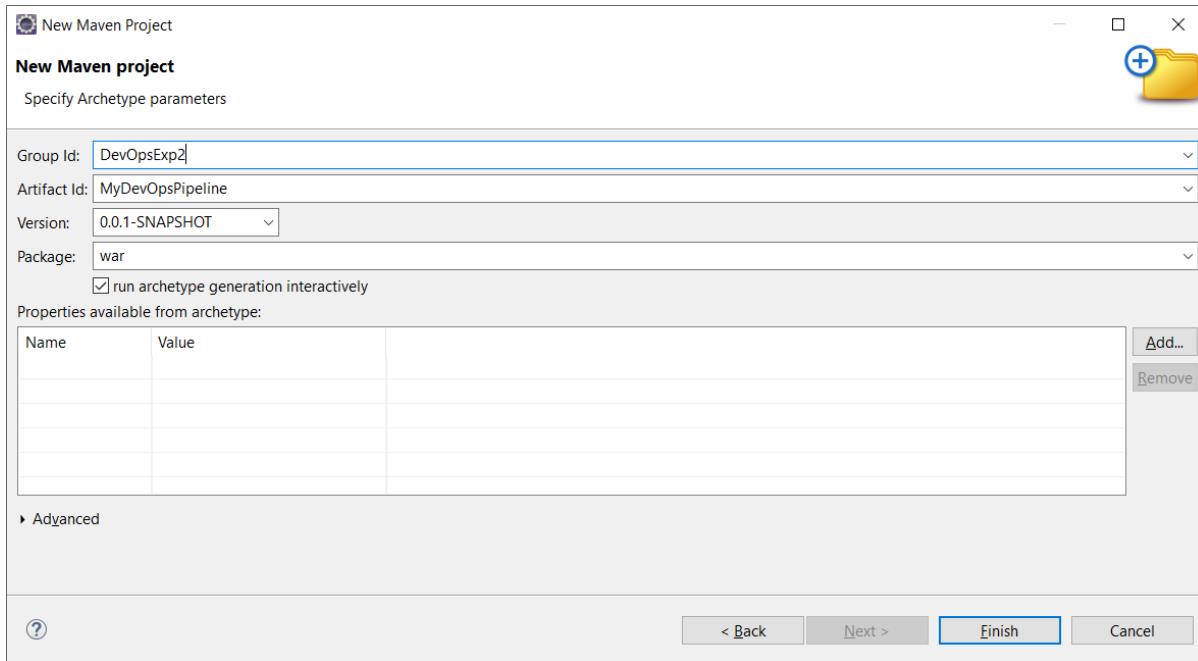
b.



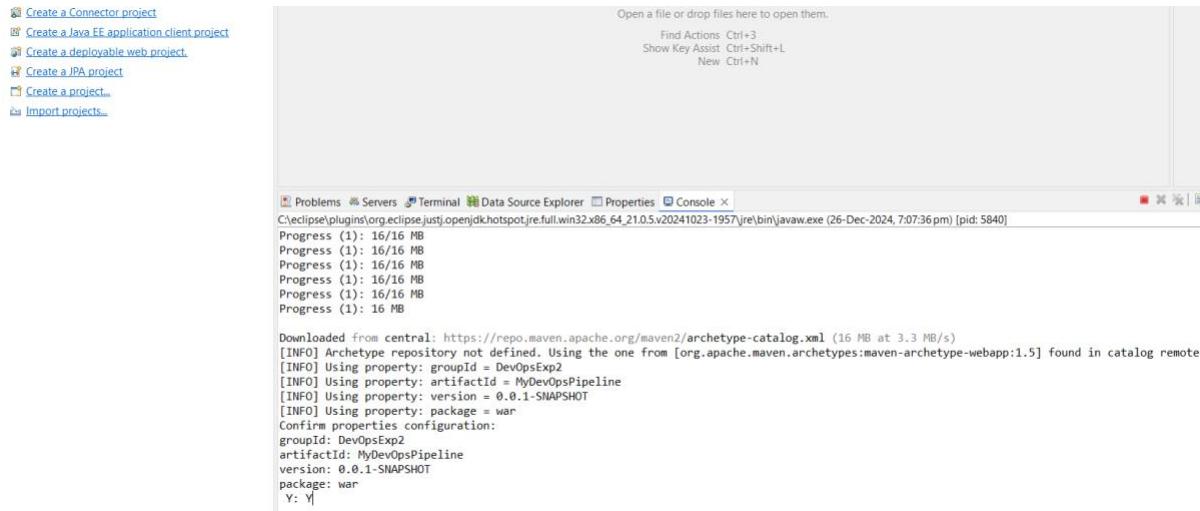
- b. Click Next and Search **org.apache.maven.archetypes** and select webapp file



- c. In a **group id** you can type anything like name and in **artifact id:** you can type anything like your roll number



- d. Click Finish



- e. Type Y and Press enter, You should see a Build Success message as below:

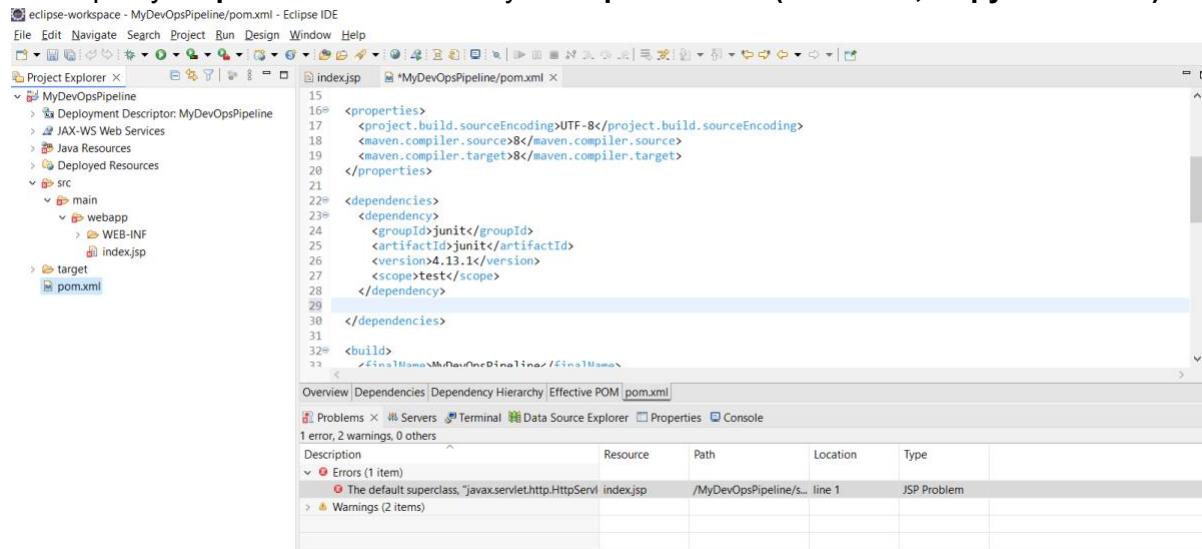
```

Show Key Assist Ctrl+Shift+L
New Ctrl+N

Problems Servers Terminal Data Source Explorer Properties Console X
<terminated> C:\eclipse\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (26-Dec-2024, 7:07:36 pm) [pid: 5840]
[INFO] Parameter: groupId, Value: DevOpsExp2
[INFO] Parameter: artifactId, Value: MyDevOpsPipeline
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: war
[INFO] Parameter: packageInPathFormat, Value: war
[INFO] Parameter: groupId, Value: DevOpsExp2
[INFO] Parameter: artifactId, Value: MyDevOpsPipeline
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] CP Don't override file C:\Users\DeLL\workspace\MyDevOpsPipeline\src\main\webapp
[WARNING] CP Don't override file C:\Users\DeLL\workspace\MyDevOpsPipeline\.mvn
[INFO] Project created from Archetype in dir: C:\Users\DeLL\workspace\MyDevOpsPipeline
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:22 min
[INFO] Finished at: 2024-12-26T19:09:00+05:30
[INFO]

```

Step-4: now open your pom.xml file and add your dependencies (Given file, Copy and Paste)



Step-5: Update your project once (Right click on Project -> click on Maven -> click on Update Project)

Step-6: Download Apache tomcat v9 from Official website.

The Apache Tomcat® software is an open source implementation of the Jakarta Servlet, Jakarta Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications. These specifications are part of the Jakarta EE platform.

The Jakarta EE platform is the evolution of the Java EE platform. Tomcat 10 and later implement specifications developed as part of Jakarta EE. Tomcat 9 and earlier implement specifications developed as part of Java EE.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here.

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat Migration Tool for Jakarta EE 1.0.9 Released 2025-01-21

The Apache Tomcat Project is proud to announce the release of 1.0.9 of the Apache Tomcat Migration Tool for Jakarta EE. This release contains a number of bug fixes and improvements compared to version 1.0.8.

The notable changes in this release are:

- Fix an issue that matchExcludesAgainstPathName didn't work for files. Based on a pull request by Semiao Marco.
- Added a new profile, SERVLET that only migrates the javax.servlet package and sub-packages. Provided by Ralf Wiebicke.
- Update dependencies

Full details of these changes, and all the other changes, are available in the [changelog](#).

Mirrors

You are currently using <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 mirrors list) that should be available.

Other mirrors: <https://dlcdn.apache.org/>

9.0.98

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

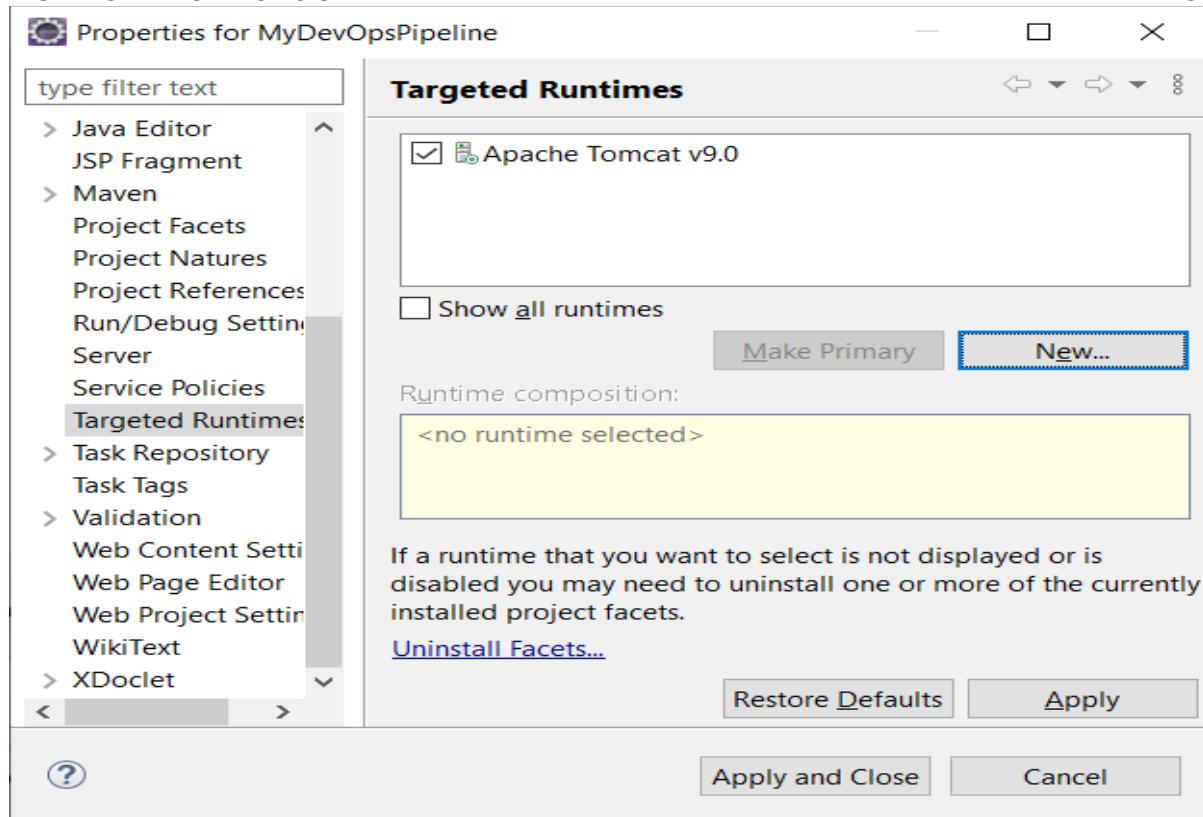
Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

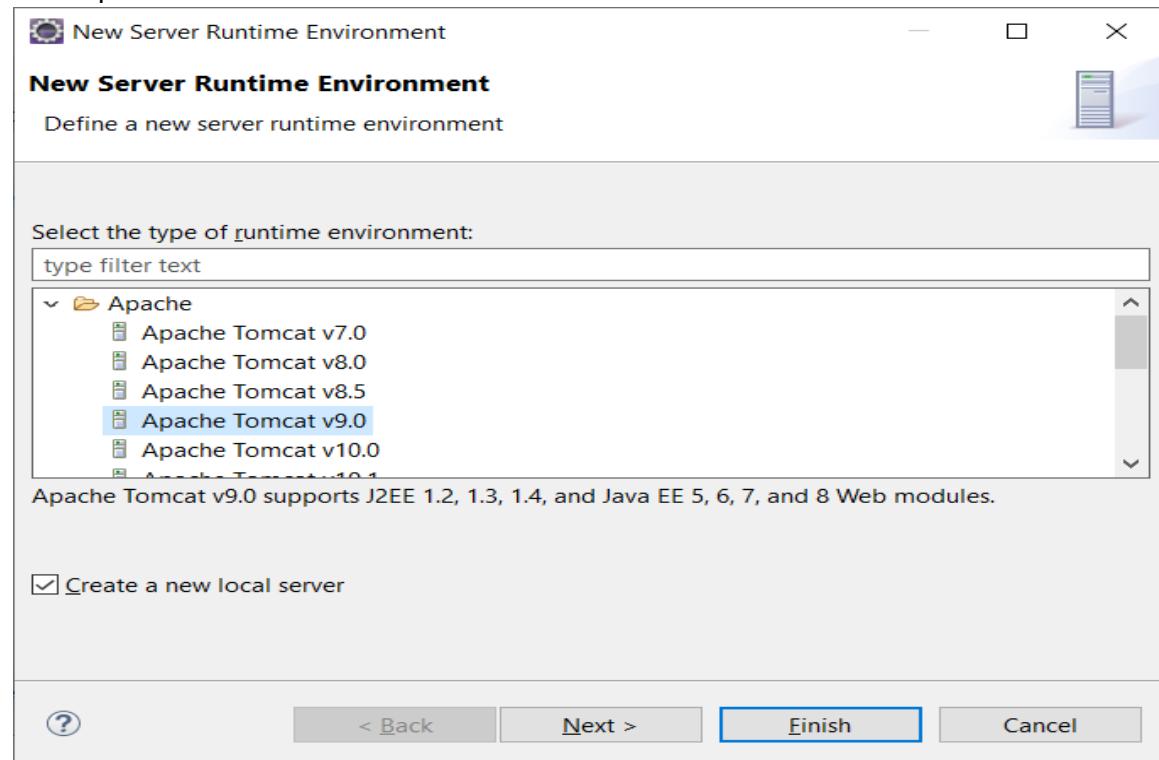
Step-7: After Download the Apache tomcat, Extract the .zip file and paste your apache-tomcat-9.0.98 folder in your folder

Step-8: Now click on your project option in Menu -> Click on Properties -> Click on Targeted Runtime

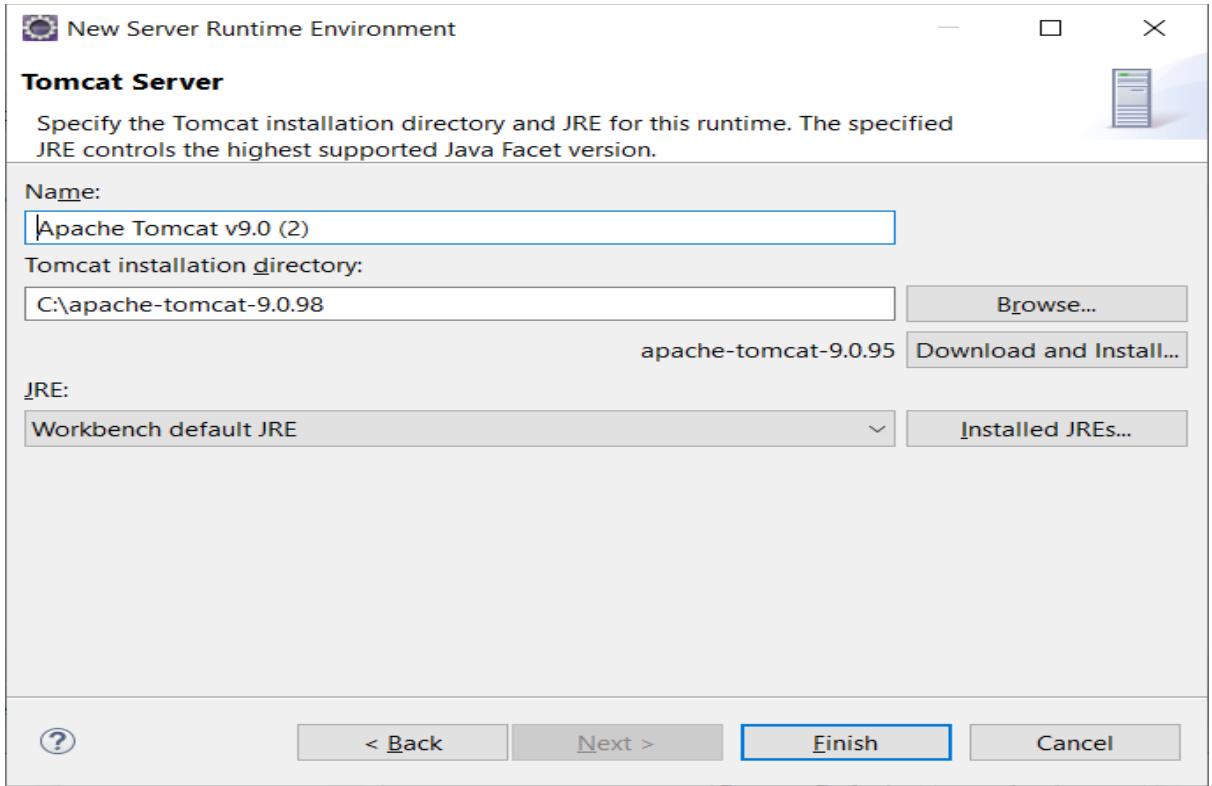
Step-9: Click on new or follow the given image



Step-10: Select Apache Tomcat v9.0



Step-11: Click on Browse and Select your Extracted file and then click on finish as given image

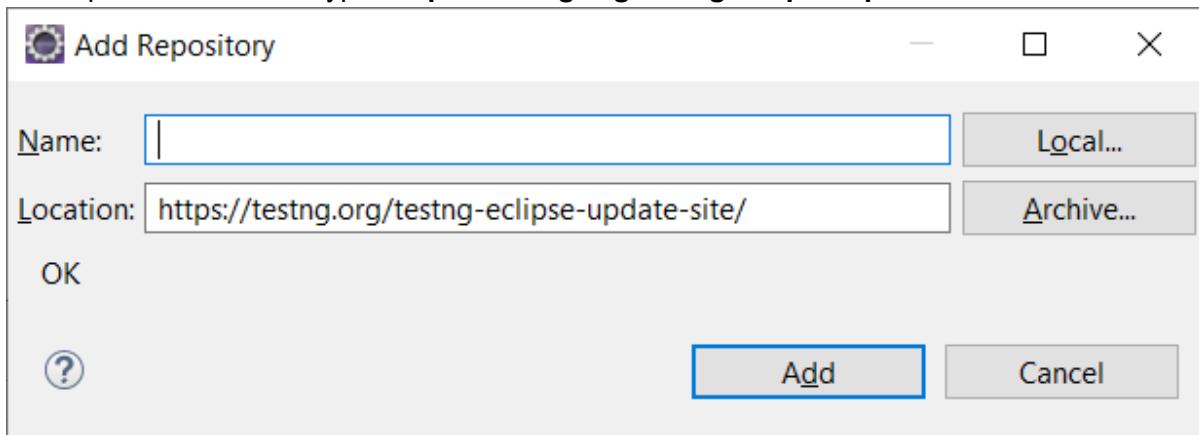


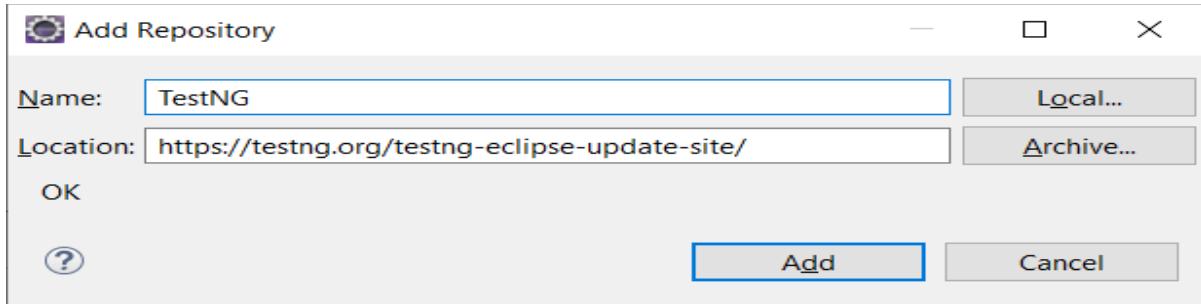
Step-12: Now Click on **help Menu ->click on Install new Software.**

Step-13: Click on **Add** and it will show a **popup dialog box** like given image

In the place of Name type: **TestNG**

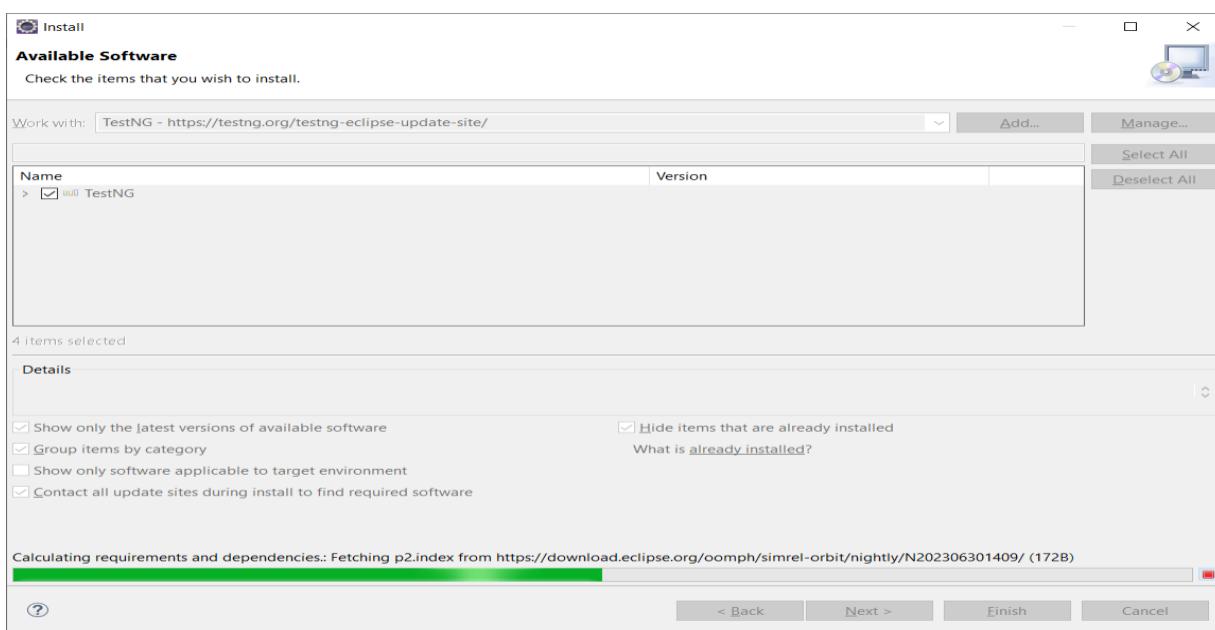
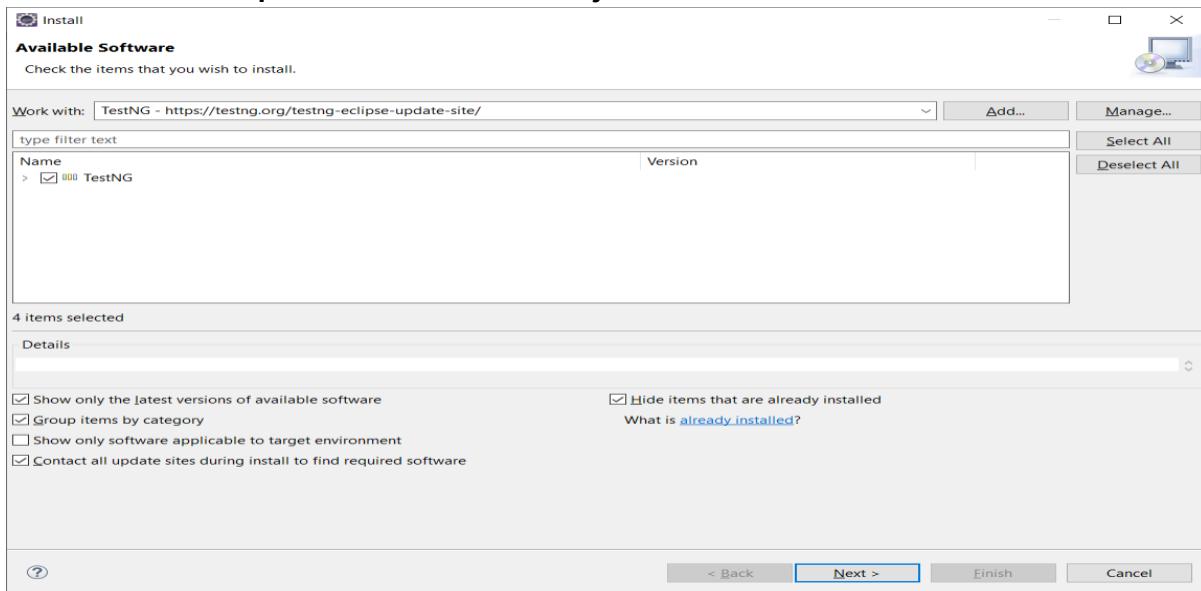
In the place of Location type: <https://testng.org/testng-eclipse-update-site/>



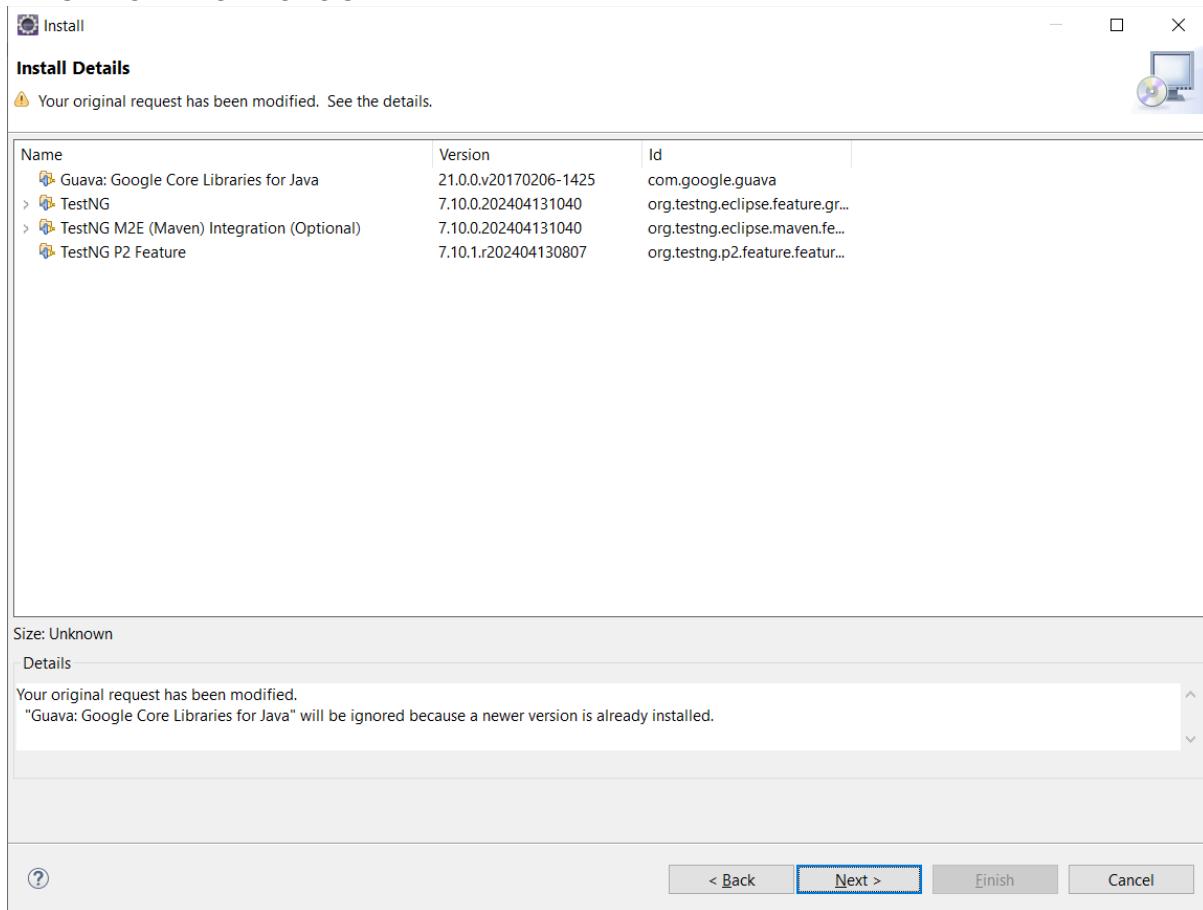


Step-14: Click on **Add** -> It will load a **TestNG Dependencies** -> Select **TestNg** like given Image and then click **Next**.

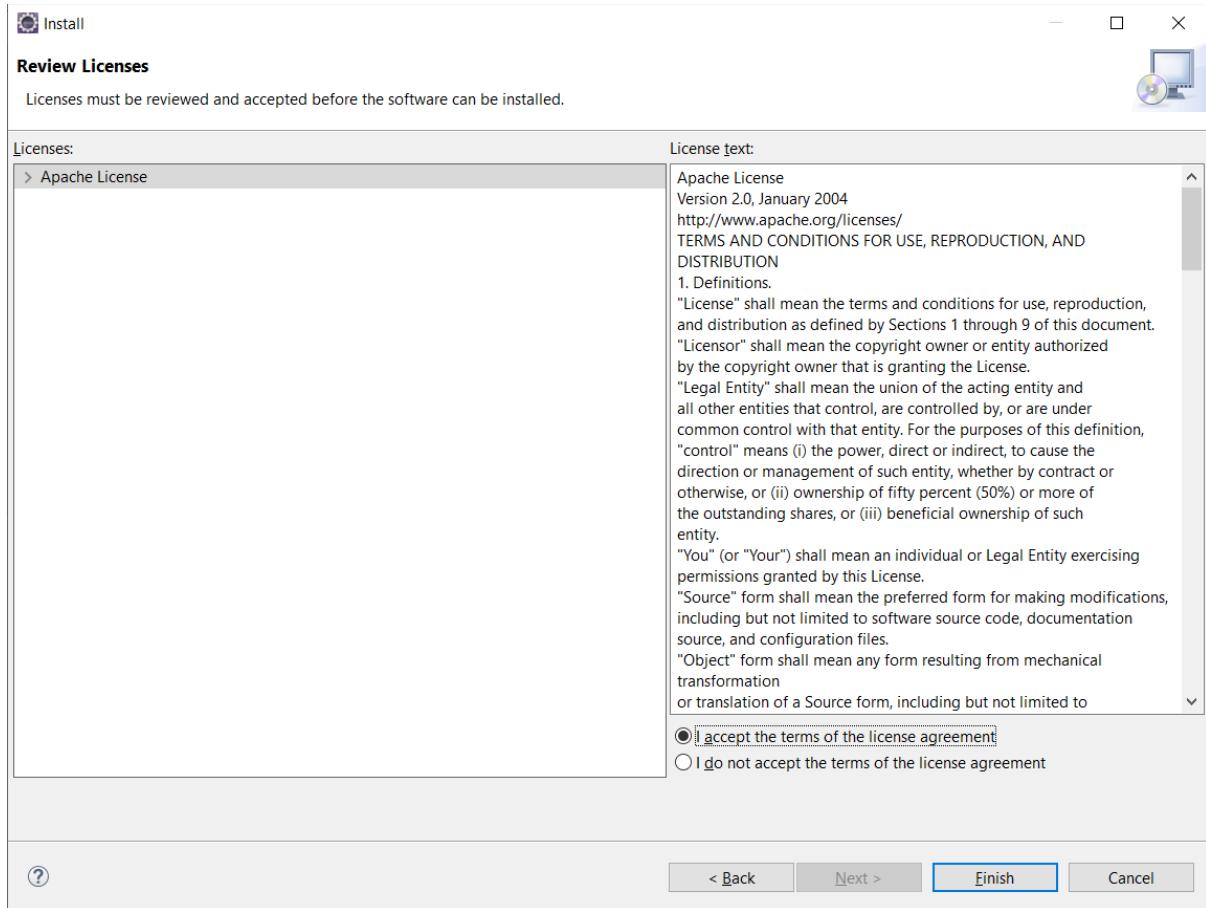
It will take 10 minute to update TestNG in our Project



Step-15: After downloading the all dependencies it will show some file select all and click on next.



Step-16: Accept Terms and condition and click on finish



Step-17: After finish it will show **restart option (Restart the Project)** otherwise just **update** once of your project.

Step-18: Now Login your GitHub Account.

Step-19: Create a New Repository and Copy your Repository and paste in notepad

Step-20: After that Click on your Profile in Right corner -> Click on Setting.

Step-21: It will show a new page, scroll down and select the **developer setting** ->click on **personal access token** ->select **Token(Classic)** ->click on **Generate new token** and select **Generate new token(Classic)** -

>write your **token name** and select **repo option** and scroll down and click on **Generate Token**. (Follow the given Image)

The screenshot shows the GitHub profile settings page. A red arrow points from the top right corner of the image to the user icon in the top right of the browser window. Another red arrow points from the bottom left towards the 'Developer settings' section.

Developer settings:

- Packages
- Copilot
- Pages
- Saved replies
- Security
- Code security
- Integrations
- Applications
- Scheduled reminders
- Archives
- Security log
- Sponsorship log
- Developer settings

Tokens (classic):

A red arrow points to this section. It contains a 'Preview' button and a 'Tokens (classic)' section.

GitHub Apps:

A red arrow points to this section. It shows a 'No GitHub Apps' message with a 'New GitHub App' button and a 'View documentation' link.

No personal access token created

Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

[Generate new token ▾](#)

Generate new token (Beta)
Fine-grained, repo-scoped

Generate new token (classic)
For general use

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration *

30 days ▾ The token will expire on Wed, Feb 26 2025

Select scopes

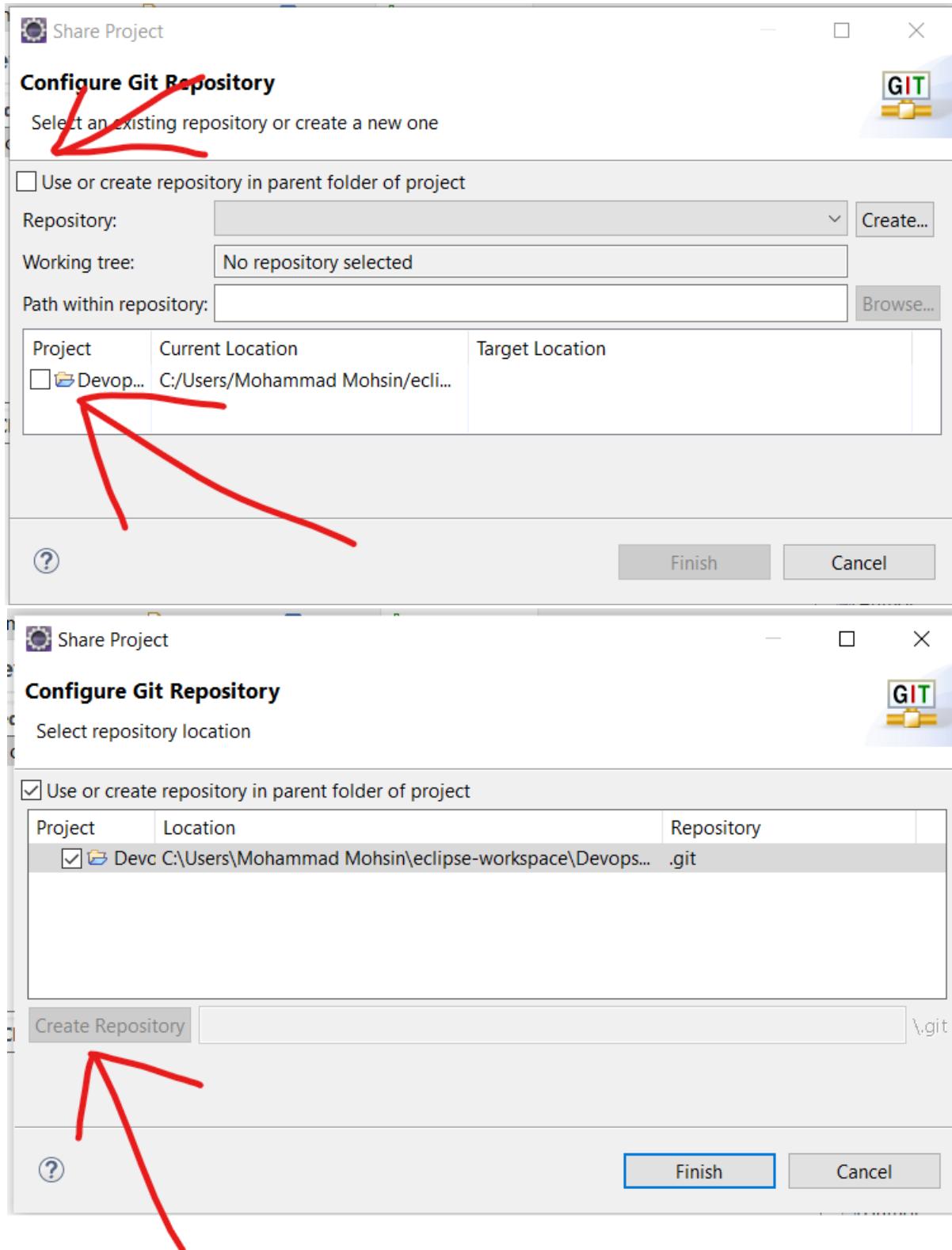
Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

Scope	Description
<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

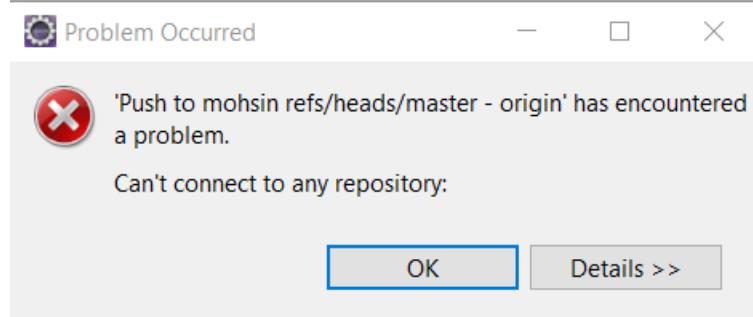
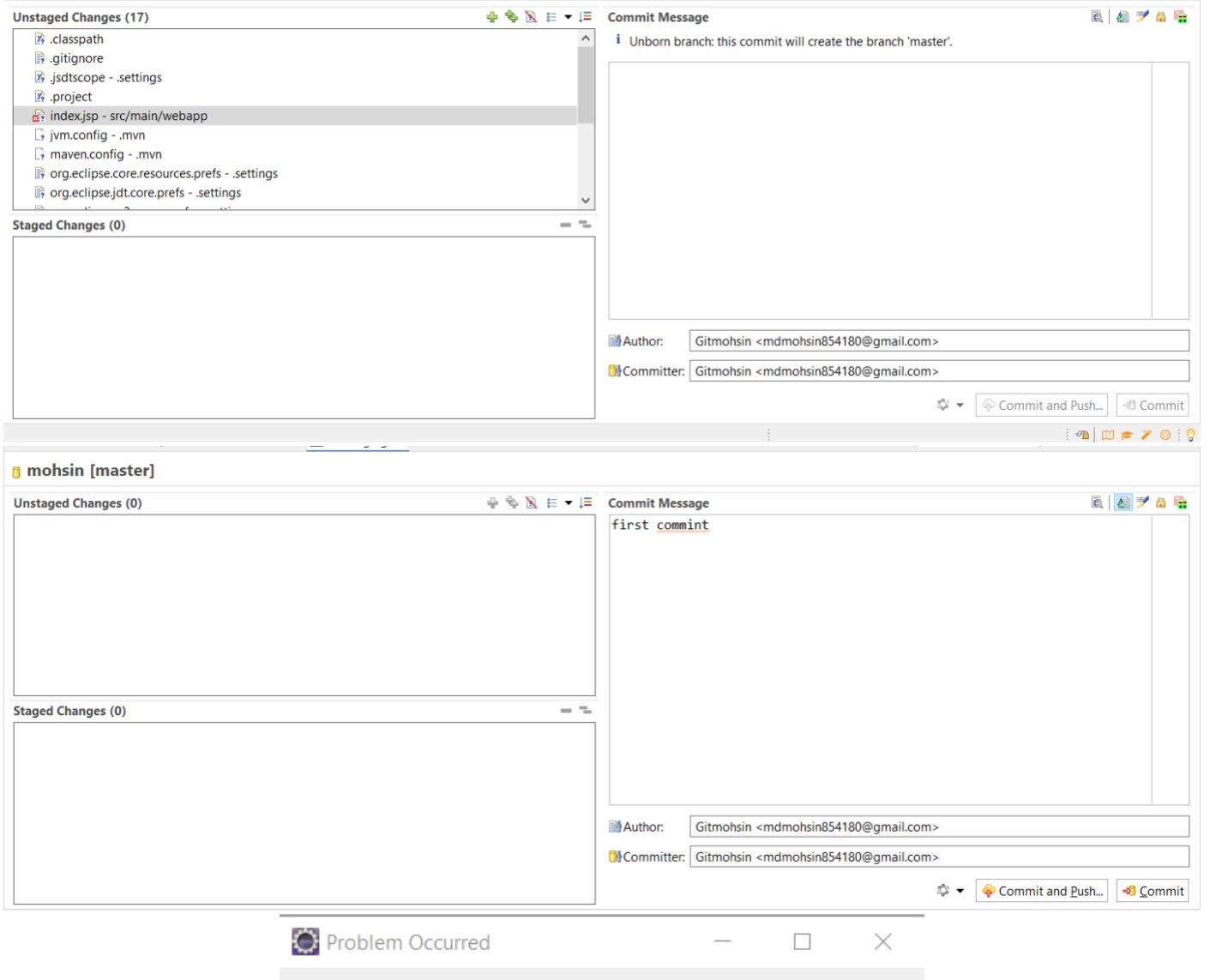
Step-22: After Generating the **token** copy the token id and paste in a**NotePad**.

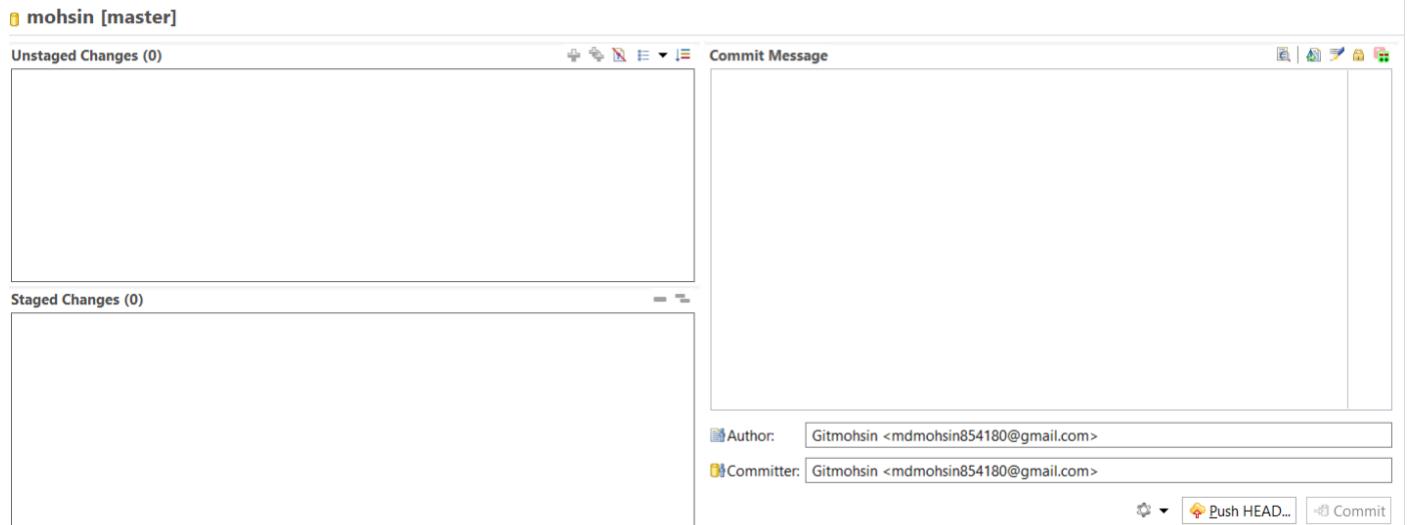
Step-23: Now come on your project and right click on your**project** ->Click on **Team** ->Click on **Share Project**.

Step-24: It will open a Dialog Box for**GitHubSetup**, select the option **Use or create repository in parent folder of project** ->Select your Project and Click on **Create Repository** and click on **Finish**.

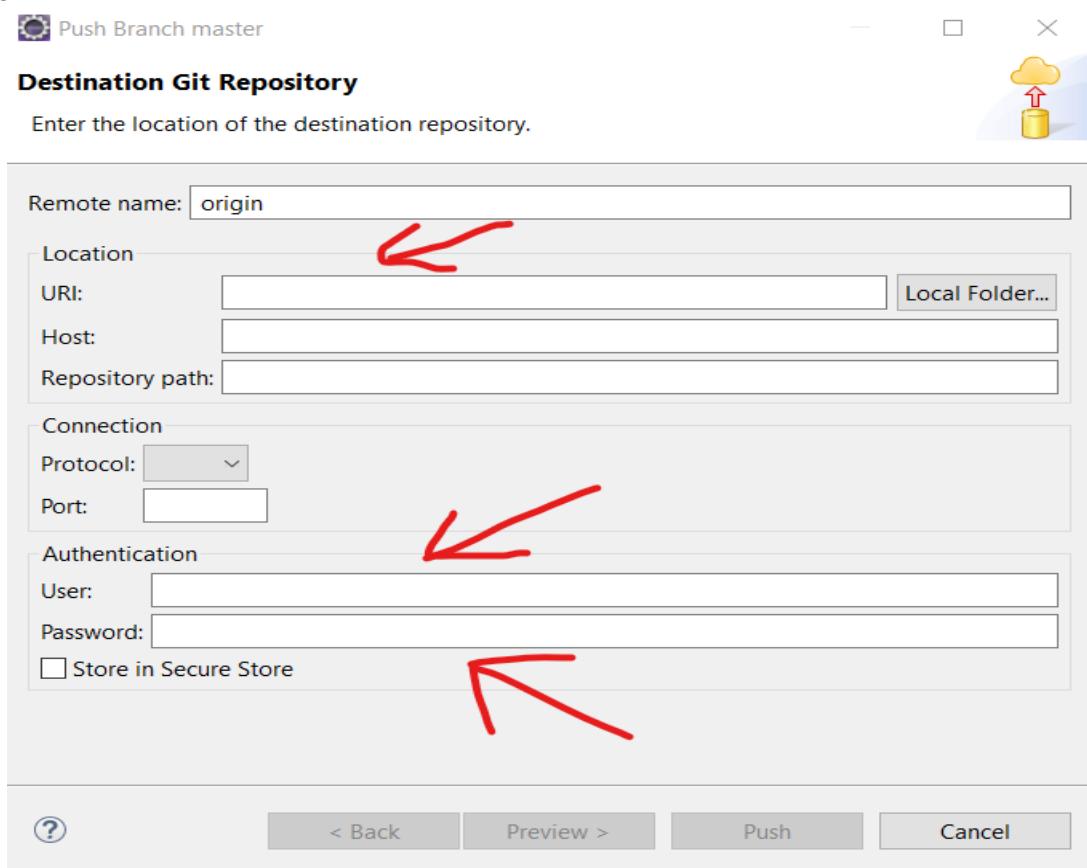


Step-25: After that again Right click on yourProject and select the Team -> click on Commit-> and stage your all file -> and Write a comment (i.e. First Commit) and click on Commit and push-> after that it show an error dialog ->click OK ->now again click on Push Head Button



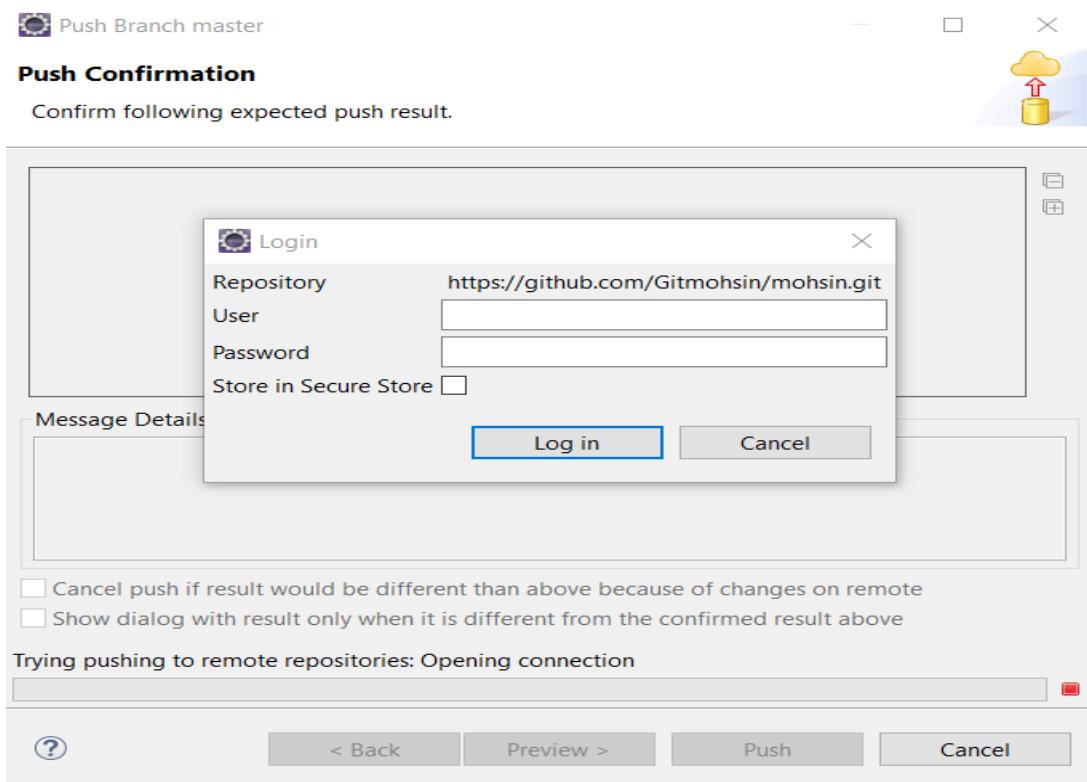


Step-26: After that again click on **Push Head**, it will show a **dialog** like given image, paste your **Repository URL** in **URL** section and type your**GitHub User Id and Password** in User, password section -> Click on **preview** ->Again clickon **Preview**.



Step-27: After that it will again show a **user Id and Password** option ->just type your**GitHub id** in user section and paste your **Token id** in Password section -> click on **push** ->one more time it will ask **user id and**

password just repeat your last step with **user id and token id** ->now check your **repository on github**, your file is **uploaded or not**



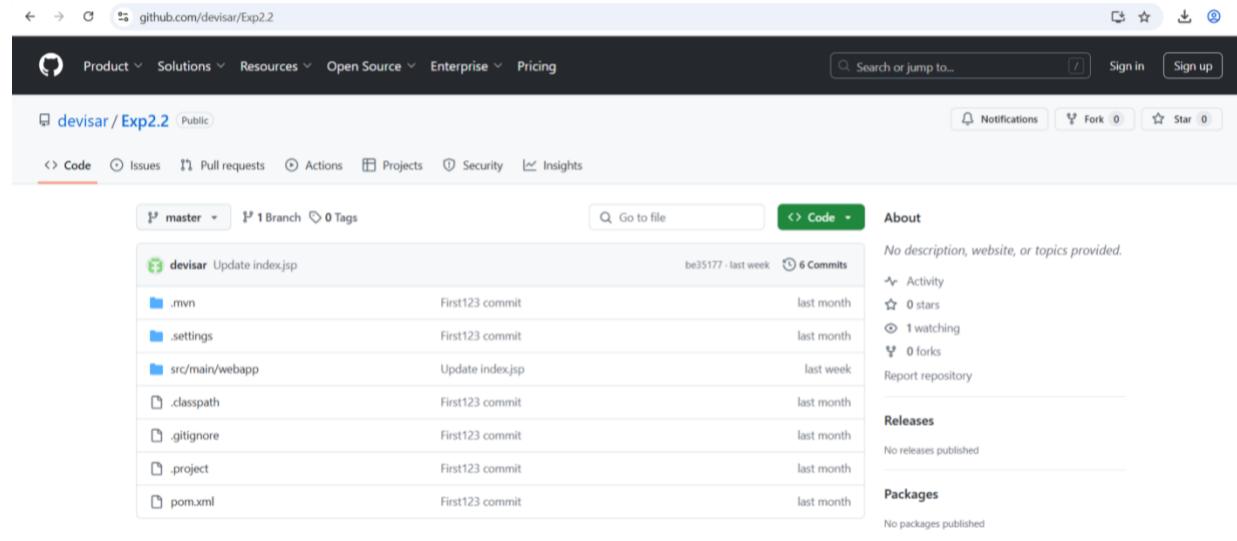
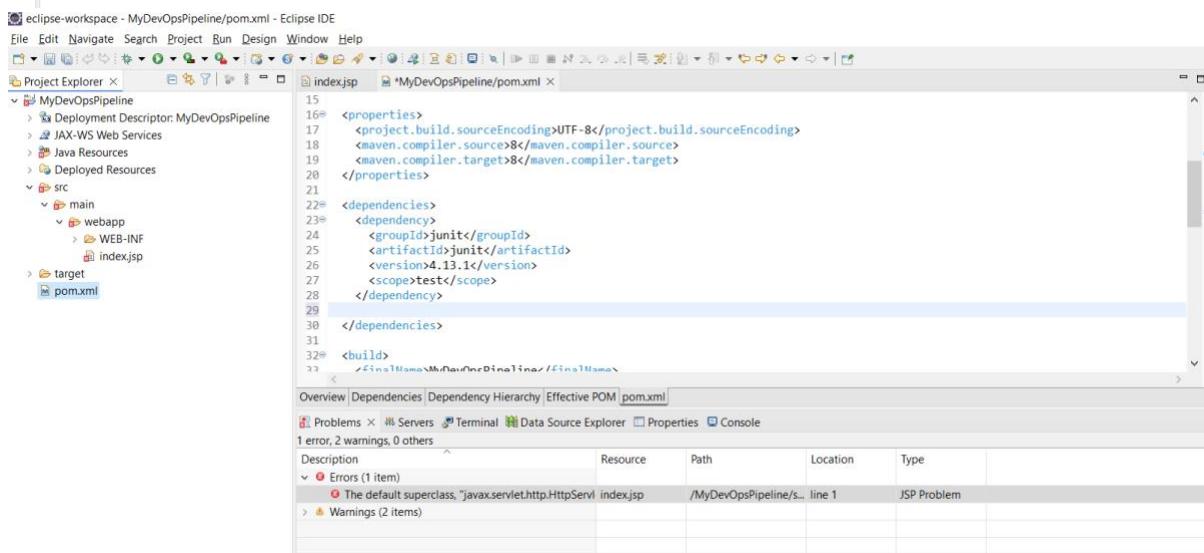
Step-28:Now you have to create a simple java code in SRC File, so first open your project from **file manager**->open **SRC** ->Create two folder in **SRC** -> **first name: java, second name: test** ->now open **test** folder and create two more folder in **test folder** ->now come on your **eclipse IDE** and **Update** your project once -> After that create a java class file with a Statement “Hello World”in your **SRC/TEST/java** folder.

Step-29:Now Push again your **all unstaged files** in your **GitHub Repository**with **different version or Comment** (Its just for Version Control).

Step-30: Now Check again your Repository your recent file is uploaded or not with different version.

```
Show Key Assist Ctrl+Shift+L  
New Ctrl+N
```

Problems Servers Terminal Data Source Explorer Properties Console ×
<terminated> C:\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (26-Dec-2024, 7:07:36 pm) [pid: 5840]
[INFO] Parameter: groupId, Value: DevOpsExp2
[INFO] Parameter: artifactId, Value: MyDevOpsPipeline
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] Parameter: package, Value: war
[INFO] Parameter: packageInPathFormat, Value: war
[INFO] Parameter: package, Value: war
[INFO] Parameter: groupId, Value: DevOpsExp2
[INFO] Parameter: artifactId, Value: MyDevOpsPipeline
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[WARNING] CP: Don't override file C:\Users\DELL\eclipse-workspace\MyDevOpsPipeline\src\main\webapp
[WARNING] CP: Don't override file C:\Users\DELL\eclipse-workspace\MyDevOpsPipeline\.mvn
[INFO] Project created from Archetype in dir: C:\Users\DELL\eclipse-workspace\MyDevOpsPipeline
[INFO] -----
[INFO] **BUILD SUCCESS**
[INFO] -----
[INFO] Total time: 01:22 min
[INFO] Finished at: 2024-12-26T19:09:00+05:30
[INFO] -----



Experiment-3

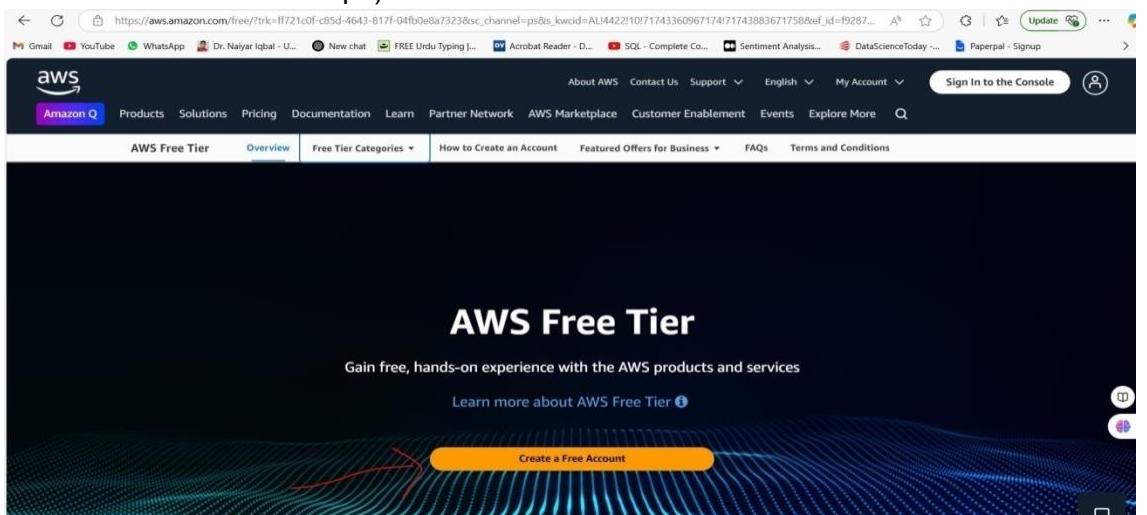
Aim: Jenkins Setup on AWS.

Note: Write the given steps in your lab manual. The provided image is just for your better understanding.

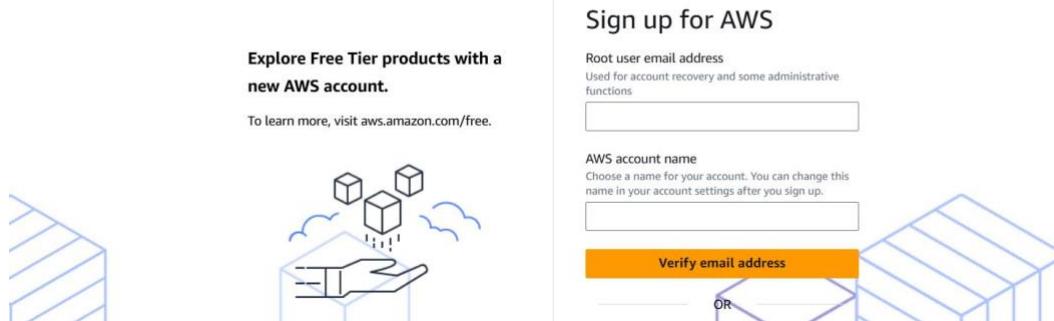
Require Software & Tools: AWS, MobaXterm Software.

Procedure:

Step-1: Search AWS Free Tier Account in any Browser and Create an Account. (if you have already an account then leave the steps)



The screenshot shows the AWS Free Tier landing page. At the top, there's a navigation bar with links like 'Products', 'Solutions', 'Pricing', 'Documentation', 'Learn', 'Partner Network', 'AWS Marketplace', 'Customer Enablement', 'Events', and 'Explore More'. Below the navigation, there's a main heading 'AWS Free Tier' and a sub-headline 'Gain free, hands-on experience with the AWS products and services'. A prominent yellow button labeled 'Create a Free Account' is centered. To the right of the button, there's a small 'Learn more about AWS Free Tier' link. The background features a dark blue digital wave pattern. On the left side of the main content area, there's a 'Sign in' form with options for 'Root user' or 'IAM user', and a field for 'Root user email address' containing 'username@example.com'. At the bottom of the sign-in form, there's a 'Next' button and a note about agreeing to the AWS Customer Agreement. To the right of the sign-in form, there's a promotional banner for 'AWS re:Invent' with the text 'Automate code reviews with Amazon Q Developer - get immediate feedback on your code so you can iterate faster'. A red arrow points from the 'Create a new AWS account' button in the sign-in form towards the 'Learn more' button on the re:Invent banner.



Step-2: After Creating the Account, Just Login as a **ROOT User**

Step-3: After Login It will show your account -> Click on **EC2** option -> Click on **Launch Instance**

This screenshot shows the AWS Console Home page. On the left, under 'Recently visited', the 'EC2' icon is highlighted with a red arrow pointing to it. On the right, the 'Applications' section shows '0' applications. At the bottom, there are links for 'View all services' and 'Go to myApplications'.

This screenshot shows the AWS EC2 Dashboard. The left sidebar includes sections for Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main area has a 'Resources' summary table and a 'Launch instance' button highlighted with a red arrow. Other sections include 'Service health' and 'Explore AWS'.

Step-4: After Clicking on Launch Instance It will ask Instance Name and Other things -> Write Instance Name as Jenkins -> Select Application and OS Image as Ubuntu -> Scroll Down and Come on Instance Type Option and Select t2.Medium -> Now Come in Key Pair(Login) Section and Click on Create New Key Pair -> Write your Key Pair Name as Exp3 -> Select RSA -> Select .pem-> Click on Create Key Pair.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

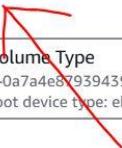
Amazon Linux 	macOS 	Ubuntu 	Windows 	Red Hat 	SUSE Linux 	Debian 
---	--	---	--	--	---	--

[!\[\]\(b779b47da2f559edb8694da06823a6c9_img.jpg\) Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-04b4f1a9cf54c11d0 (64-bit (x86)) / ami-0a7a4e87939439934 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

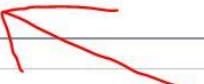


Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
 Add additional tags



▼ Instance type [Info](#) | [Get advice](#)

Instance type
t2.medium 
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour
On-Demand Linux base pricing: 0.0464 USD per Hour On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations  Compare instance types

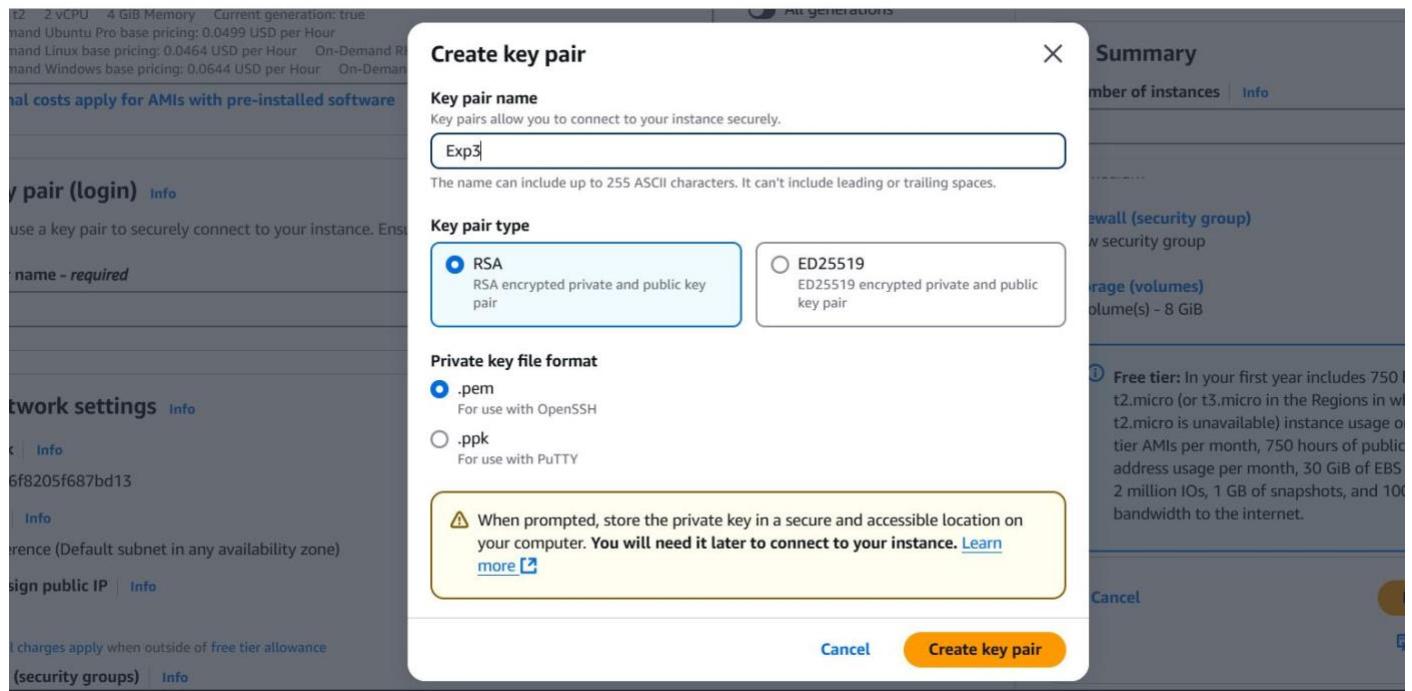
Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

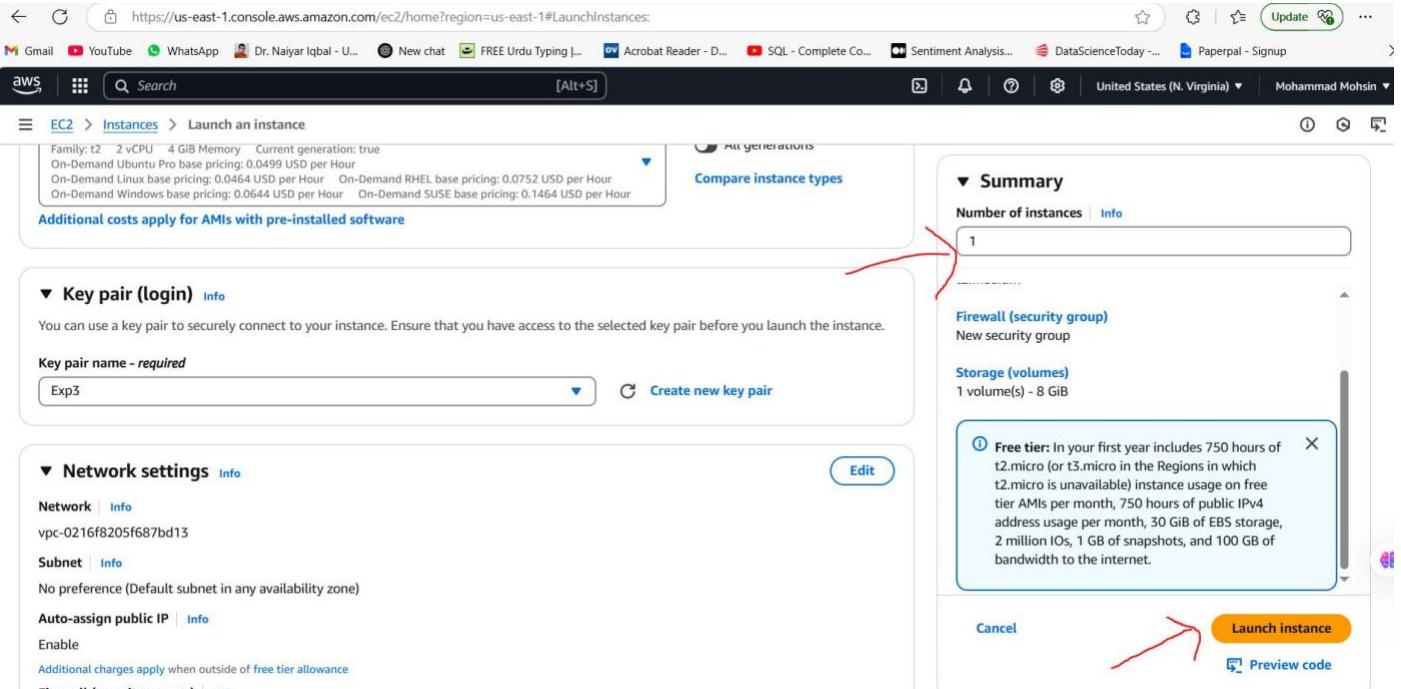
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required
  





Step-5: After Click on **Create Key Pair** it will download a **Exp3.pem** file ->After that in Right Side we have a Option **Number of Instances** Select 1 -> Click on **Launch Instances** (It will take some time and will create an instance).



Step-6: After that Refresh the page -> Select the **Created Instance** -> Click on **Security**.

The screenshot shows the AWS EC2 Global View dashboard. On the left, a sidebar menu includes options like Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main area displays a table of resources:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	1
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	2	Snapshots	0	Volumes	1

Below the resources, there's a section for launching instances and another for service health. The account attributes panel on the right shows the Default VPC (vpc-0216f8205f687bd13) and various settings.

The screenshot shows the AWS EC2 Instances page. The sidebar menu is identical to the previous dashboard. The main table shows one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Jenkins	i-09fa49fc85615f9d2	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-34-

A red arrow points to the checkbox next to the Jenkins instance name.

The screenshot shows the AWS EC2 Instances page with the Jenkins instance selected. A red arrow points to the checkbox next to the Jenkins instance name. The instance details page for i-09fa49fc85615f9d2 (Jenkins) is displayed at the bottom, with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Security tab is highlighted with a red arrow.

Step-7: After Clicking on Security Group -> It Will Show a Blue Link name as **Security Group** -> Click on that **blue Link** it will Open a Page -> Click On **Edit Inbounds rule** -> Click on **Add Rule** -> Add one Rule **HTTP** and Source type **Anywhere IPV4** -> Add one More Rule **Custom TCP**, port range **8080** and Source Anywhere **IPV4** -> Click on **Save Rule**

Instances (1/1) Info

Last updated 28 minutes ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Jenkins	i-09fa49fc85615f9d2	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-34-

i-09fa49fc85615f9d2 (Jenkins)

security details

IAM Role -

Owner ID 102250941657

Security groups sg-02a4089f784882382 (launch-wizard-1)

Launch time Wed Feb 05 2025 13:45:56 GMT+0530 (India Standard Time)

Inbound rules

Details

Security group name launch-wizard-1	Security group ID sg-02a4089f784882382	Description launch-wizard-1 created 2025-02-05T07:51:46.636Z	VPC ID vpc-0216f8205f687bd13
Owner 102250941657	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

Inbound rules (1)

Search

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-075fd1ed42717eadd	IPv4	SSH	TCP	22

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-075fd1ed42717eadd	SSH	TCP	22	Custom	0.0.0.0/0

Add rule

Step-7: After that click on Dashboard -> click on instance Running -> select Jenkins -> click on connect.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, there is a red arrow pointing to the 'Instances' link. In the main content area, there is a table of resources. To the right of the table, there is a 'Connect' button, which is also highlighted with a red arrow. The table data is as follows:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	1
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	2	Snapshots	0	Volumes	1

Step-8: After that click on connect it will open a Linux terminal -> just type `clear` command and follow the **Step-9** documents.

```
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Sat Feb 15 04:20:11 UTC 2025

System load: 0.0      Processes:          101
Usage of /: 21.7% of 7.57GB  Users logged in: 0
Memory usage: 21%      IPv4 address for eth0: 172.31.1.31
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

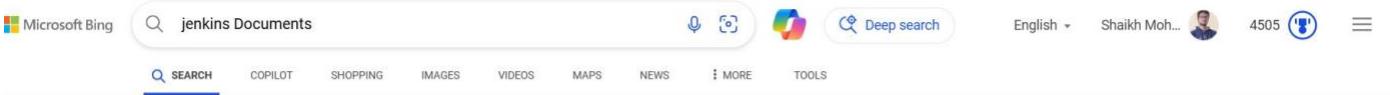
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

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

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

ubuntu@ip-172-31-1-31:~$
```

Step-9: Now You have to Setup Jenkins -> Search [Jenkins Document](#) in Any Browser -> Click on first Jenkins link -> Click on [Installing Jenkins](#) -> click on [Linux](#) -> Click on [Debian/Ubuntu](#) -> After that it will show the [Linux Command](#)



The screenshot shows a Microsoft Bing search results page. The search query is "jenkins Documents". The top result is a link to "Jenkins User Documentation" from Jenkins.io. A red arrow points to this link. Below it, there's a snippet of text about Jenkins installation and a link to "Installing Jenkins". To the right, there's a summary box for "Jenkins - Wikipedia" with a Jenkins logo, a brief description, and a link to "See more on Wikipedia". Another red arrow points to the Jenkins logo in this box.

Jenkins User Documentation
 Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed. This documentation begins wi...

Installing Jenkins

Jenkins - Wikipedia
 Continuous integration tool
 Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration, and continuous delivery. It is a server-base...

Jenkins User Documentation

Welcome to the Jenkins user documentation - for people wanting to *use* Jenkins's existing functionality and plugin features.

If you want to extend the functionality of Jenkins by developing your own Jenkins plugins, please refer to the [Extend Jenkins](#) (development) documentation).

What is Jenkins?

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing and delivering or deploying software.

Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime

[User Documentation Home](#) [User Handbook Overview](#) [Index](#) [Docker](#)

Installing Jenkins

The procedures in this chapter are for new installations of Jenkins.

Jenkins is typically run as a standalone application in its own process. The Jenkins WAR file bundles [Winstone](#), a [Jetty](#) servlet container wrapper, and can be started on any operating system or platform with a version of Java supported by Jenkins.

Chapter Sub-Sections
[Docker](#)
[Kubernetes](#)
[Linux](#)
[macOS](#)
[Windows](#)

User Handbook

- User Handbook Overview
- **Installing Jenkins**
 - Docker
 - Kubernetes
 - **Linux**
 - macOS
 - Windows
 - Other Systems
 - WAR file

User Handbook

- User Handbook Overview
- **Installing Jenkins**
 - Docker
 - Kubernetes
 - **Linux**
 - macOS
 - Windows
 - Other Systems
 - WAR file
 - Other Servlet Containers
 - Offline Installations
 - Initial Settings
- Platform Information
- Using Jenkins
- Pipeline
- Blue Ocean
- Managing Jenkins

Linux

Jenkins installers are available for several Linux distributions.

- Debian/Ubuntu
- Fedora
- Red Hat Enterprise Linux and derivatives

On Debian and Debian-based distributions like Ubuntu you can install Jenkins through [apt](#).

Long Term Support release

A [LTS \(Long-Term Support\) release](#) is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [debian-stable apt repository](#).

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

[Explain](#) [Edit](#)

Weekly release

A new release is produced weekly to deliver bug fixes and features to users and plugin developers. It can be installed from the [debian apt repository](#).

Step-10: Now Come on your [MobaXtrem Application](#) -> Type [Clear](#) and Hit [Enter key](#) ->it will clear your Command Screen -> Now copy the the[Long Term Support release](#) code till [/dev/null](#) -> Paste in your [Linux Terminal](#) and hit the [Enter Key](#) -> it will Download the Jenkins Support File

Code:

```
sudowget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

Step-11: Clear your Screen -> Now Update your Project Using Given Code.

Code:

```
sudoapt-get update
```

Step-12: clear your screen ->Now Check Java version Using Command: [java --version](#)

Step-13: if It Show [Java Not Found](#) then it show the option [install Java](#) -> so install java using given Command -> it will ask [Y/N](#) just type Y and hit Enter -> It will take some time to download java.

Code:

```
sudoaptinstall fontconfig openjdk-17-jre
```



```

Reading package lists... Done
ubuntu@ip-172-31-18-41:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.12+7-1ubuntu2~24.04, or
sudo apt install openjdk-21-jre-headless # version 21.0.4+7-1ubuntu2~24.04
sudo apt install default-jre          # version 2:1.17-75
sudo apt install openjdk-11-jre-headless # version 11.0.24+8-1ubuntu3~24.04.1
sudo apt install openjdk-8-jre-headless # version 8u422-b05-1~24.04
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless # version 22~22ea-1
ubuntu@ip-172-31-18-41:~$ █

libwayland-client0 libwayland-server0 libwebp7 libx11-xcb1 libxcb-dri2-0 libxcb-dr
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1 libxdamage
libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6t64 libxtst6 libxv1 libxf86dga1 libxxf
openjdk-17-jre-headless session-migration ubuntu-mono x11-common x11-utils

The following packages will be upgraded:
  libdrm-common libdrm2
2 upgraded, 120 newly installed, 0 to remove and 83 not upgraded.
Need to get 123 MB of archives.
After this operation, 543 MB of additional disk space will be used.
Do you want to continue? [Y/n] █

```

Step-14: After Downloading the Java, Clear your Screen using [clear](#) command -> Once again Update your Project Using [Update Command](#) (follow Step-14 command) -> now Install Jenkins Using Jenkins Command, i.e given code (It will ask again [Y/N](#) just type Y and Hit Enter) -> After Installing the Jenkins Once again Update your Project using [Update Command](#) Just Check Java and Jenkins Version Using ([java –version](#) and [Jenkins –version](#) Command)

Code: [Sudo apt-get install jenkins](#)

Step-15: Now [Enable the Jenkins Server](#) Using Given Command

Command: [Sudo systemctl enable jenkins](#)

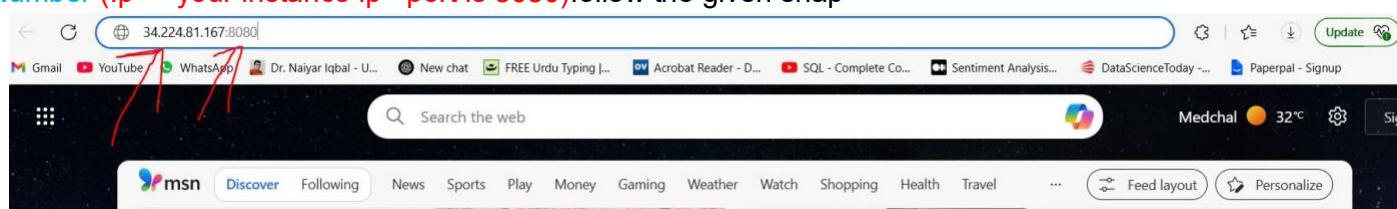
Step-16: After [Enable the Jenkins Server](#) Start [Jenkins Server](#) Using Given Command

Command: [Sudo systemctl start jenkins](#)

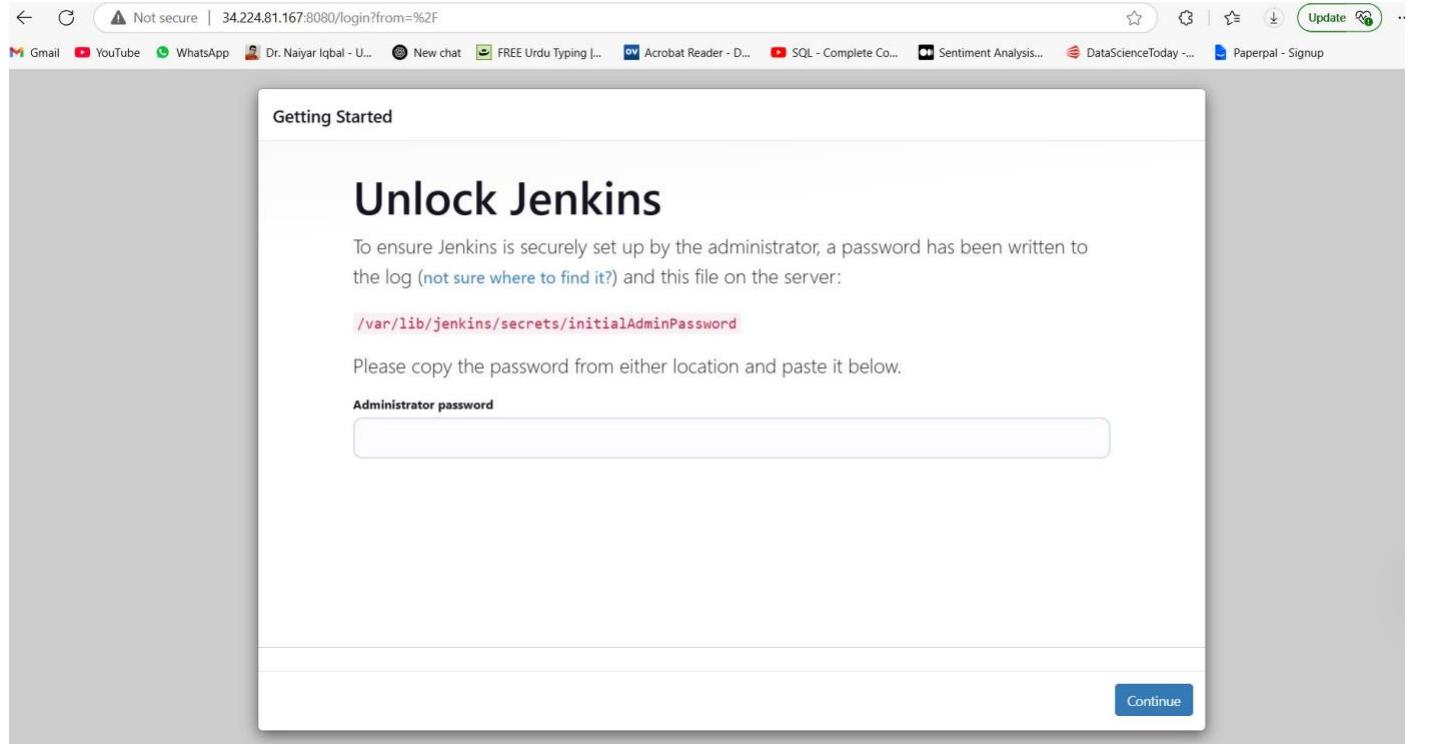
Step-17: After [Starting the Jenkins Server](#) Check [Jenkins Server](#) Status using Given Command. It will show the Status Active running ans show the status of memory, CPU etc.

Command: [Sudo systemctl status jenkins](#)

Step-18: Now Open your Any Browser and Search the Jenkins Server using [Instance IP](#) and [Custom Port Number](#) ([Ip= <your instance ip> port is 8080](#)) follow the given snap



Step-19: It will start the [Jenkins Server](#), show some path with red color and Ask the Administrator Password like Given Image.



Step-20: Copy the **RED color path** and Paste in your **Linux Terminal** followed by **sudo cat** command and hit the Enter it will show **64-bit Alpha-Nemuric code** -> Copy that code and paste in the place of **password** in your browser -> click on continue option.

```
Feb 05 09:57:39 ip-172-31-18-41 jenkins[4342]: 2025-02-05 09:57:39.224+0000 [id=48]      INFO
ubuntu@ip-172-31-18-41:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
55bd1877468949c783c222d4823c8068
ubuntu@ip-172-31-18-41:~$ ^C
ubuntu@ip-172-31-18-41:~$
```

Step-21: After clicking on **continue** option it will show **two option** for **installing the plugging**-> Just click on **installed suggested plugin** -> it will take some time for install all plugins -> After installing all plugins, it will ask Creating an account using**user ID** and **password** so create an account using user id and password (fill all related field). -> Click on **Save and Continue**

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

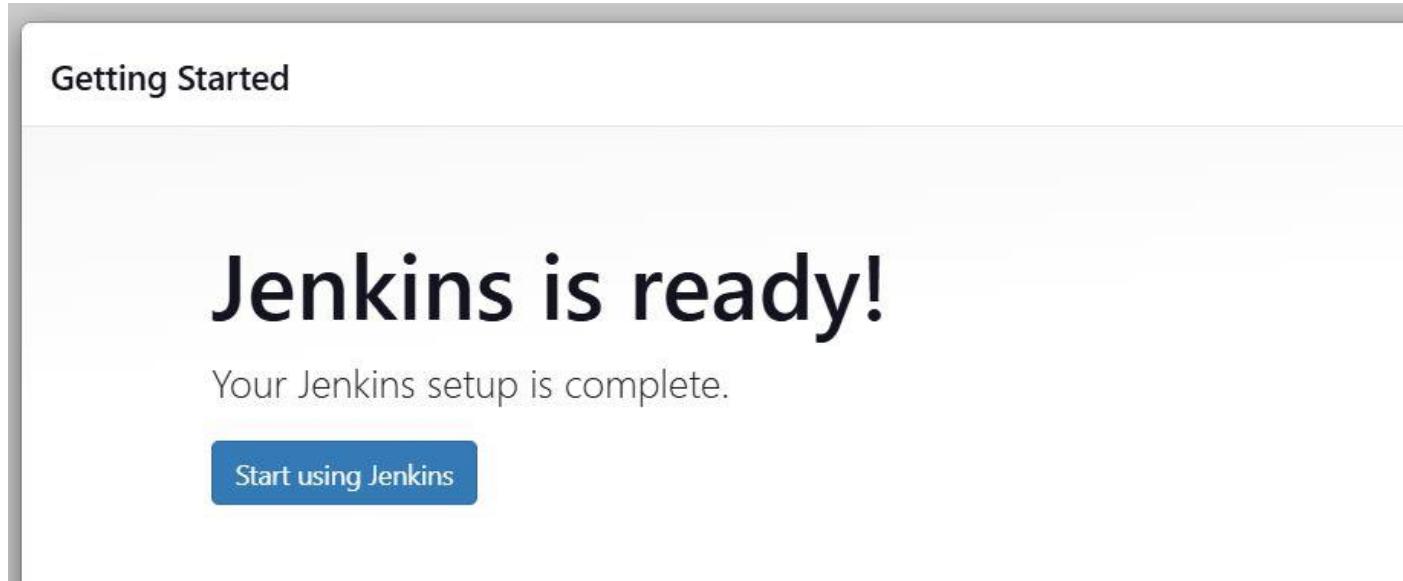
Jenkins 2.479.3

Skip and continue as admin

Save and Continue

Step-22: After click on [Save and continue](#) -> it show some links and show the option [Save & Finish](#) -> Click on [Save & Finish](#) -> After that it show Jenkins in Ready and show a option [Start using Jenkins](#) -> just click on it -> It will open a Jenkins Dashboard.

Output:-



OUTPUT

Your Output is the Jenkins Dashboard

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: '+ New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below that are two dropdown menus: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The main content area has a 'Welcome to Jenkins!' message, a 'Start building your software project' button, and several other buttons for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. At the top, there's a navigation bar with various icons and a user profile for 'Mohammad Mohsin'.

Experiment-4

Aim:Ansible Setup and SSH key

(Note: All Blue Words is a command, just copy and paste)

Require Software & Tools: AWS Account, Puttygen software, WinSCP Software.

Procedure:

Step-1: Login AWS and Create three Instances on AWS account using ubuntu OS and save key pair also.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'EC2' selected and a list of options: Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, and Spot Requests. The main area is titled 'Instances (3) Info' with a sub-header 'Instances state = running'. It includes filters for 'Find Instance by attribute or tag (case-sensitive)' and 'All states'. A table lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
ansible	i-06e5a4b61b232e855	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-15-207
server1	i-03a3d92b48681be23	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-52-66-
server2	i-017b60943062166ca	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-110-

Step-2: change the 2nd and 3rd instance name as a server1 and server2 (you can named anything) -> Edit inbound rules of 1st instance (ansible). Edit inbound rules like given details.

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

The screenshot shows the 'Inbound rules' section for the 'ansible' security group. It lists three rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0cfb8bf827fabbfa2	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0e5fdd175949f7cc8	Custom TCP	TCP	8080	Custom	0.0.0.0/0
sgr-0cacba689eaaf68e8	SSH	TCP	22	Custom	0.0.0.0/0

At the bottom, there's a blue 'Add rule' button.

Step-3: Now come on instance and select 1st instance (ansible) and click on connect. It will open a linux terminal -> type **Clear** command.

```

Usage of /: 24.9% of 6.71GB Users logged in: 0
Memory usage: 20% IPv4 address for enX0: 172.31.13.107
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-172-31-13-107:~$
```

i-06e5a4b61b232e855 (ansible)

PublicIPs: 15.207.222.240 PrivateIPs: 172.31.13.107

Step-4: type command `sudo apt update` and hit enter.

Step-5: type command `sudo apt install ansible` and hit enter -> it will ask yes/no type yes and hit enter.

Step-6: type command `sudo apt install openssh-client` and hit enter.

Step-7: type command `sudo apt install openssh-server` and hit enter.

Step-8: type command `cd ~/.ssh/` and hit enter.

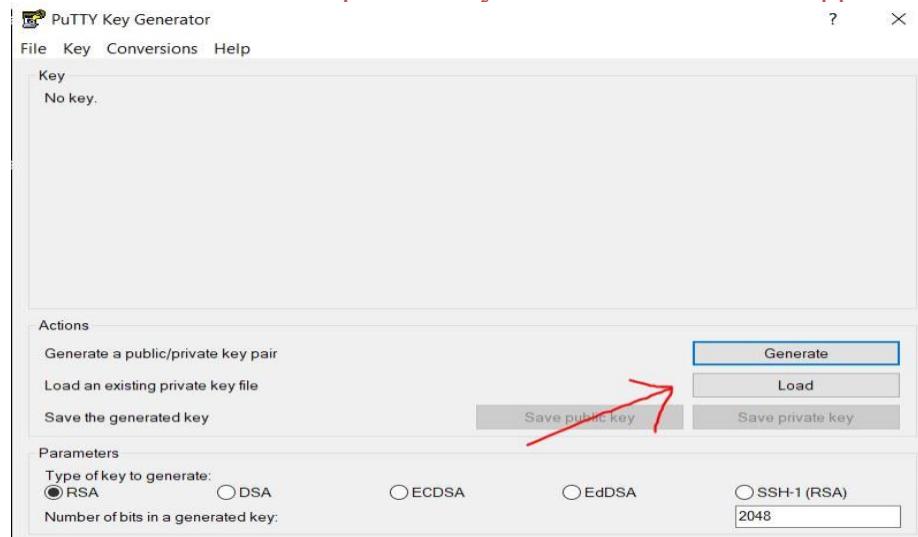
Step-9: type command `ls` and hit enter.

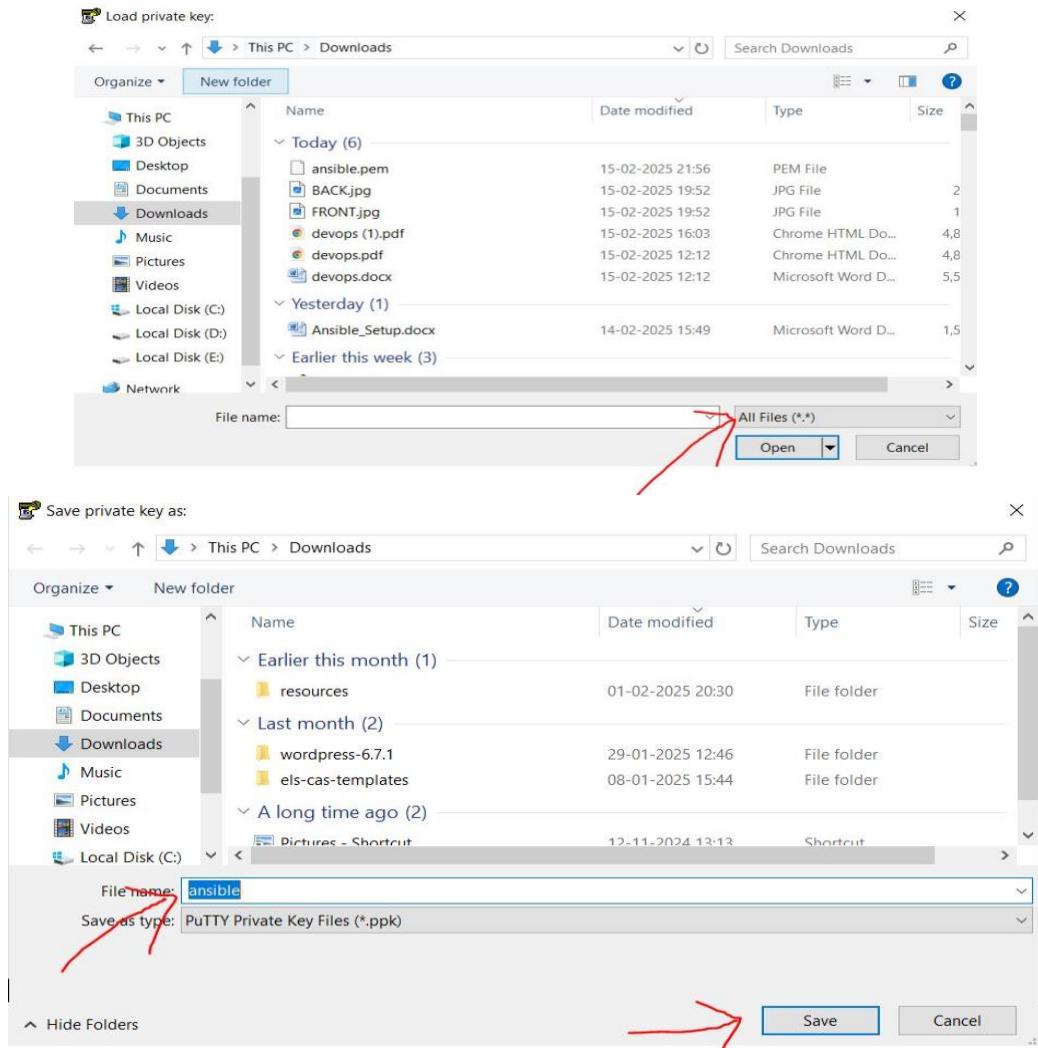
Step-10: type command `cd` and hit enter.

Step-11: type command `cd -` and hit enter. (it will show ssh path)

Step-12: now again type command `cd` and hit enter.

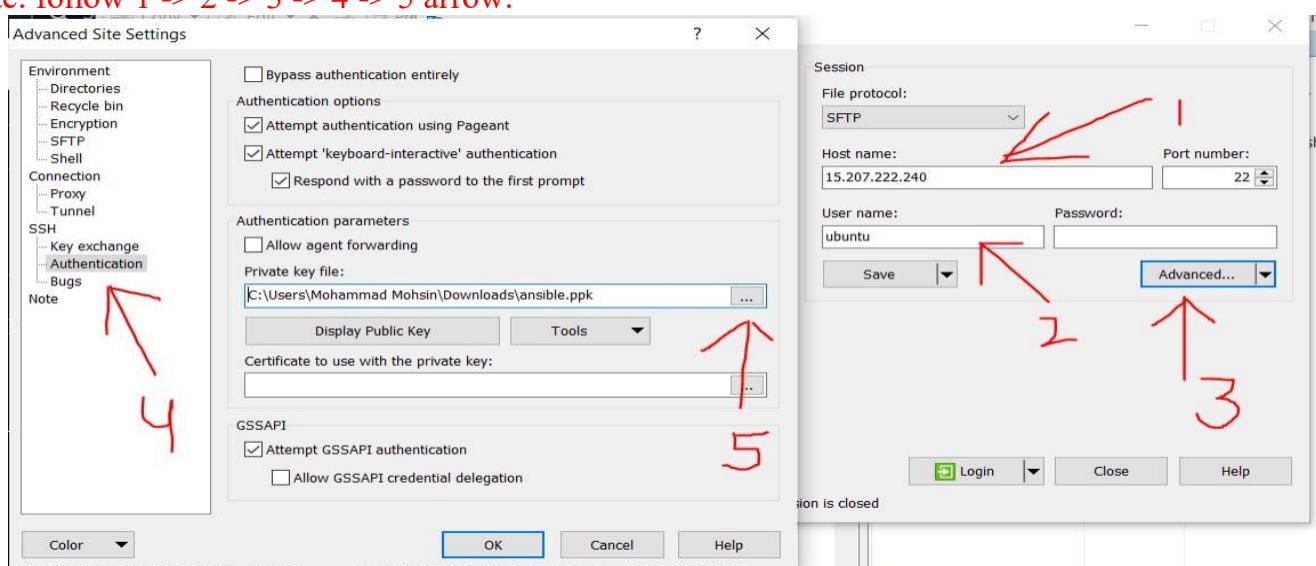
Step-13: open `puttygen` software -> click on `load` -> select as `all files` -> select `ansible.pem`(your key pair) -> click on `save` and -> click on `save private key` and save the file with `.ppk` extension.





Step-14: Now open winCP software -> it will ask for **host name** just copy and paste your **ansibleinstance Ip**-> In the place of **user name** type **Ubuntu** -> click on **advance** -> select **authentication** -> in the place of **private key** select your downloaded **.ppk file** -> click on **Ok** -> click on **Login**. (it will connect to your server and show a **Ubuntu wincp window**)

Note: follow 1 -> 2 -> 3 -> 4 -> 5 arrow.



Step-15: after connecting to your server it will show **two directory** (one is your local disk, another is ssh directory) -> right click in **ssh directory** -> click on **new** -> select **directory** -> create a **folder name as key** -> now select your **.pemfile** from left side local disk directory and **drag in your key folder**.

Step-16: Now come on in your terminal -> type **cd** command and hit enter -> type **ls** command hit enter, you will see key.

```
ubuntu@ip-172-31-13-107:~/.ssh$ cd
ubuntu@ip-172-31-13-107:~$ ls
ubuntu@ip-172-31-13-107:~$ cd
ubuntu@ip-172-31-13-107:~$ ls
key
ubuntu@ip-172-31-13-107:~$
```

Step-17: Now type **cd key/** and hit enter, you will see **ansible.pem**

Step-18: Now type **cp ansible.pem ~/ssh/** command and hit enter -> after that type **cd** hit enter -> after that **ls ~/ssh/** and hit enter, you will see **ansible.pem** and **authorized_key**

Step-19: Now create one more directory using **mkdir ansible-script** command and hit enter-> select directory using **cd ansible-script/** command and hit enter.

Step-20: Now write **vi inventory.ini** command and hit enter -> it will open your Vim Editor -> copy the given command and paste in your Vim editor -> after that in the place of [server1-ip] copy and paste server1 instance public ip and in the place of [server2-ip] copy and paste server2 instance public ip -> after that save this vim editor using **:wq!** Command and hit enter

Command:

```
[webservers]
server1 ansible_host=[server1-ip] ansible_user=ubuntu ansible_ssh_private_key_file=
/home/ubuntu/.ssh/ansible.pem
server2 ansible_host=[server2-ip] ansible_user=ubuntu ansible_ssh_private_key_file=
/home/ubuntu/.ssh/ansible.pem
```

Step-21: Now write the command **vi playbook.yml** and create a yml file -> it will again open your Vim editor -> copy and paste the given command in your vim editor after save and exit using command **:wq!** and hit enter.

Command:

```
-- name: Install Nginx on AWS EC2
hosts: webservers
become: yes
tasks:
  - name: Install Nginx
    apt:
      name: nginx
      state: present
      when: ansible_os_family == "Debian"

  - name: start and enable nginx server
    systemd:
      name: nginx
      state: started
```

Step-22: Now you can check your .ini file and .yml file using `cat inventory.ini` command and `cat playbook.yml` command.

Step-23: Now type `ansible-playbook -i playbook.yml` command and hit enter.

Step-24: Now type `ansible-playbook -i inventory.ini playbook.yml` command and hit enter -> it will ask for type `yes` and again `yes` and again `yes` or may be it will show server Unreachable.

Step-25: Now type `cd` and hit enter -> after that `cd ~/.ssh/` hit enter, you will see `ansible.pemauthorized_key` and `known_hosts`.

Step-26: Now again run the command `ansible-playbook -i inventory.ini playbook.yml`. it will run your server or may be it will show again an error server unreachable.

Step-27: If server is reachable is then check your server using server ip otherwise -> write command `chmod 400 ~/.ssh/ansible.pem` hit enter -> after that `ls -l ~/.ssh/` and hit enter, it will show Ubuntu file.

Step-28: Now again run the command `ansible-playbook -i inventory.ini playbook.yml`. now it will run your server properly. After that you can check your server in any browser using server ip.

OUTPUT:-

```
ubuntu@ip-172-31-4-180:~$ sudo apt-get install -y ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
ubuntu@ip-172-31-4-180:~$ ansible-playbook playbook1.yml --syntax-check
playbook: playbook1.yml
ubuntu@ip-172-31-4-180:~$ vim playbook1.yml
ubuntu@ip-172-31-4-180:~$ ansible-playbook playbook1.yml --syntax-check
playbook: playbook1.yml
ubuntu@ip-172-31-4-180:~$ ansible-playbook playbook1.yml -b
PLAY [Configure tomcat9] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.7.139 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.7.139]
[WARNING]: Platform linux on host 172.31.5.200 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.5.200]
TASK [Install tomcat] ****
ok: [172.31.7.139]
ok: [172.31.5.200]
TASK [Change port no of tomcat 8080 to 9090] ****
ok: [172.31.7.139]
ok: [172.31.5.200]
PLAY RECAP ****
172.31.5.200 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.7.139 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Not secure 43.204.218.193

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.

Not secure 13.233.103.243

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.

Experiment-5

(Note: All Blue Words is a command, just copy and paste)

Aim: Build War file in Devops

Require Software & Tools: Jenkins, Jdk-17 or jdk-21, Git bash.

Procedure:

Step-1: First open any browser and Search **download Jenkins** -> click on **first link** -> scroll down and click on **windows logo** -> **download will be started**.

Microsoft Bing

Search: download jenkins

ALL COPILOT IMAGES VIDEOS MAPS NEWS SHOPPING MORE TOOLS

Jenkins Continuous integration tool

Overview Download Documentation

Copilot Answer

Jenkins https://www.jenkins.io › download

Download and deploy - Jenkins

Jenkins is an open source automation server for software development. Choose from different

Jenkins Jenkins Blog Success Stories Contributor Spotlight Documentation Plugins Community Subprojects Security About Download

Docker
Kubernetes
Ubuntu/Debian
Red Hat Enterprise Linux and derivatives
Fedora
Windows
openSUSE

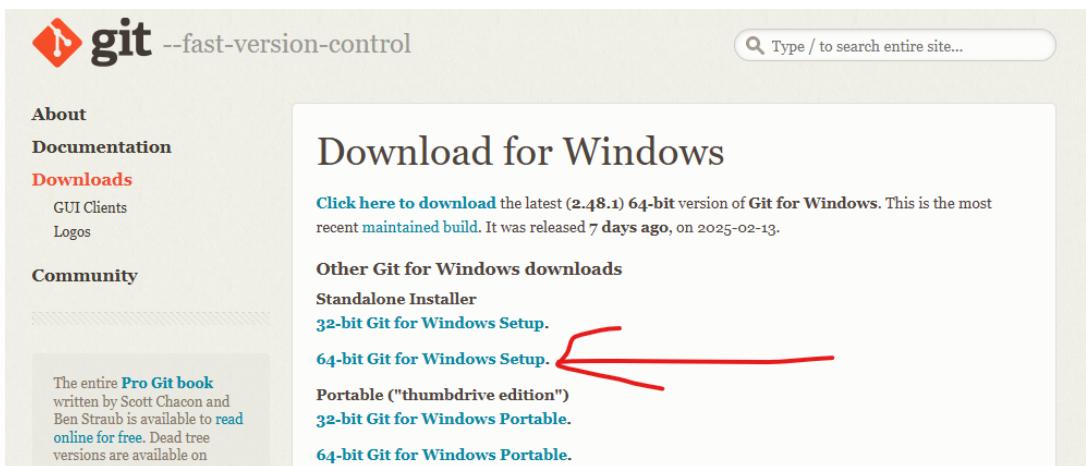
Docker
Ubuntu/Debian
Red Hat Enterprise Linux and derivatives
Fedora
Windows
openSUSE
Arch Linux

Step2: First check java jdk in your system, if java is available check the java version using cmd. otherwise download and install jdk-17.

```
C:\Users\student>java --version
java 17.0.12 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 17.0.12+8-LTS-286)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.12+8-LTS-286, mixed mode, sharing)

C:\Users\student>
```

Step-3: Download Git Bash 64-bit Standalone Installer, and install git bash in your system. After installation set the **git.exe path in your runtime environment.**



Step-4: Now install Jenkins in your system -> click on **next** and again **next** -> select **run service as a local system** -> click on **next** and click on **Test port** -> click on **next** and click on **change** and set the **java jdk path** -> click on **next** and again **next** and click on **install**.

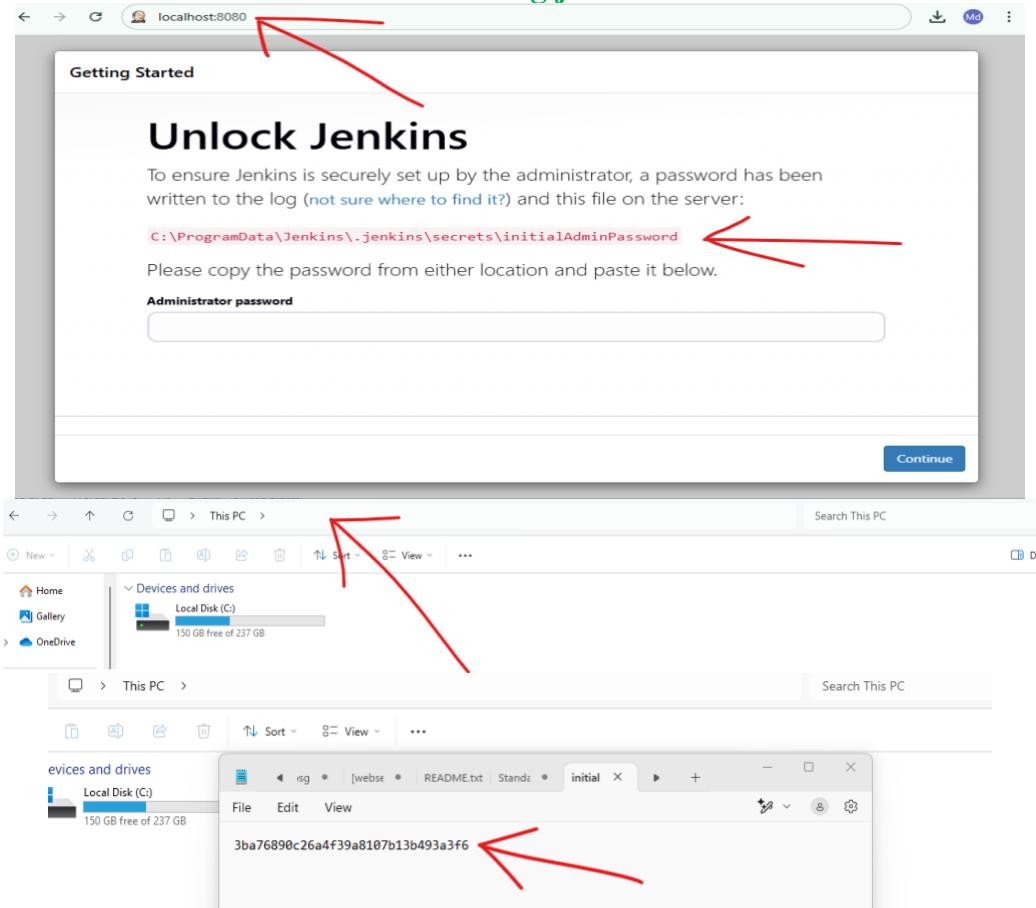


Step-5: After that install Git Bash in your system (no need to change anything just click on next next and next till last).

Step-6: Now open a browser -> search localhost:8080

Step-7: It will open your Jenkins server -> Copy red line location and search it to in your File manager.

Step-7: It will show [open with option](#) just select [notepad](#) -> after that [copy the password](#) and [paste in Jenkins server](#) -> click on [continue](#) -> click on [suggested plugin](#) -> it will take some time for installation -> create [admin user](#) using given details and click on [save and continue](#) -> after that click on [save and finish](#) -> Click on [start using jenkins](#) It will show Jenkins Dashboard.



Getting Started

The screenshot shows the Jenkins 'Getting Started' page. On the left, there's a sidebar with links like 'Dashboard', 'Manage Jenkins', 'Build History', 'My Views', 'Build Queue' (with 'No builds in the queue.'), and 'Build Executor Status' (with '0/2'). The main content area has a large title 'Getting Started'. Below it is a grid of plugin cards. Some cards have green checkmarks in the top-left corner, indicating they are installed. The cards include:

- Folders
- OWASP Markup Formatter
- Build Timeout
- Credentials Binding
- Timestamper
- Workspace Cleanup
- Ant
- Gradle
- Pipeline
- Github Branch Source
- Pipeline: GitHub Groovy Libraries
- Pipeline Graph View
- Git
- SSH Build Agents
- Matrix Authorization Strategy
- PAM Authentication
- LDAP
- Email Extension
- Mailer
- Dark Theme

On the right side of the grid, there's a vertical sidebar with sections like 'Ionicons API', 'Folders', 'OWASP Markup Formatter', and 'Pipeline: Step API'. At the bottom of the grid, it says '** - required dependency'.

Jenkins 2.498

Step-8: After that click on manage Jenkins -> click on plugin -> click on available plugin -> search maven -> select maven integration -> click on install -> scroll down and check maven integration and and loading plugin is green and success or not -> After success click on go back to the top page

The screenshot shows the 'Manage Jenkins' page. On the left, there's a sidebar with 'New Item', 'Build History', 'Manage Jenkins' (which is highlighted with a red arrow), and 'My Views'. The main content area has a heading 'Manage Jenkins' with a sub-note: 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.' Below this are several configuration sections:

- System Configuration** (with 'System' and 'Tools' options)
- Nodes** (with 'Nodes' and 'Clouds' options)
- Tools** (with 'Configure tools, their locations and automatic installers.')
- Plugins** (with 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.' and a red arrow pointing to it)
- Appearance** (with 'Configure the look and feel of Jenkins')

Plugins

Dashboard > Manage Jenkins > Plugins

Search: maven

Install Name 1 Released

Maven Integration 3.25 13 days ago

This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTS as well as the automated configuration of various Jenkins publishers such as Junit.

Config File Provider 982.vb_a_e458a_37021 26 days ago

Groovy-related External Site/Tool Integrations Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files...) loaded through the UI which will be copied to the job workspace.

Jira 3.13 11 mo ago

External Site/Tool Integrations Maven jira This plugin integrates Jenkins to Atlassian Jira.

Plugin	Status
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
Javadoc	Success
Dev Tools Symbols API	Success
JSch dependency	Success
Maven Integration	Success
Loading plugin extensions	Success

→ Go back to the top page
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Step-9: Now again click on **Manage Jenkins** -> **click on tools** -> scroll down and check **Git path** -> after that scroll down and in a **Maven installation section** -> **click on add maven** -> write maven name and **click on apply** -> **click on save**.

Manage Jenkins

System Configuration

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers. (This is the section pointed to by the first red arrow.)
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Appearance**: Configure the look and feel of Jenkins.

Git installations

Git

Name: Default

Path to Git executable: git.exe (This is the field pointed to by the second red arrow.)

Install automatically

Maven installations

Add Maven

Maven

Name: maven

Install automatically

Install from Apache

Version: 3.9.9

Save **Apply**

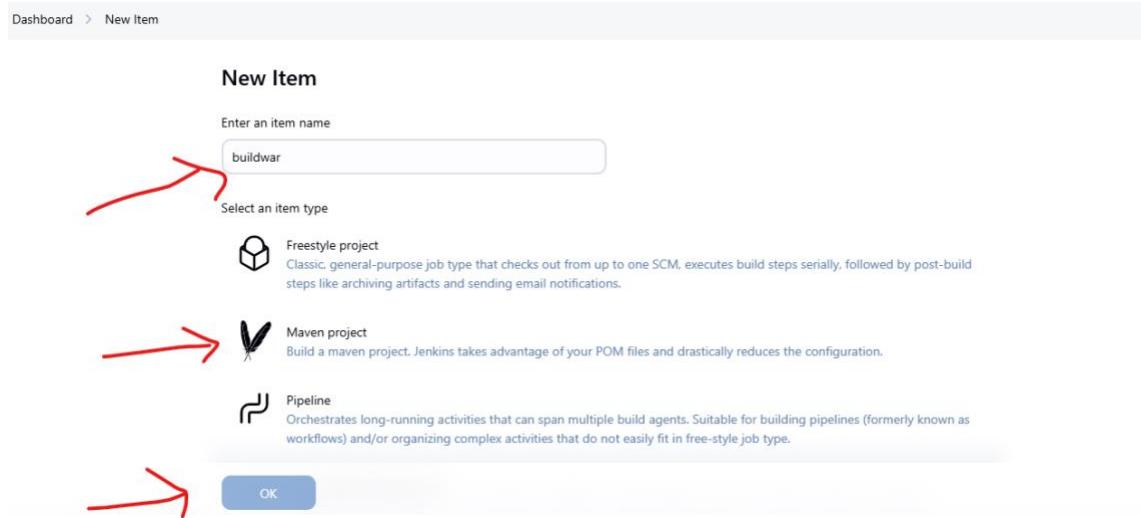
Step-10: now click on **dashboard** -> click on **create a job** -> type your **buildwar name** (i.e. myWar) -> select **Maven Project** -> click on **OK**

Welcome to Jenkins!

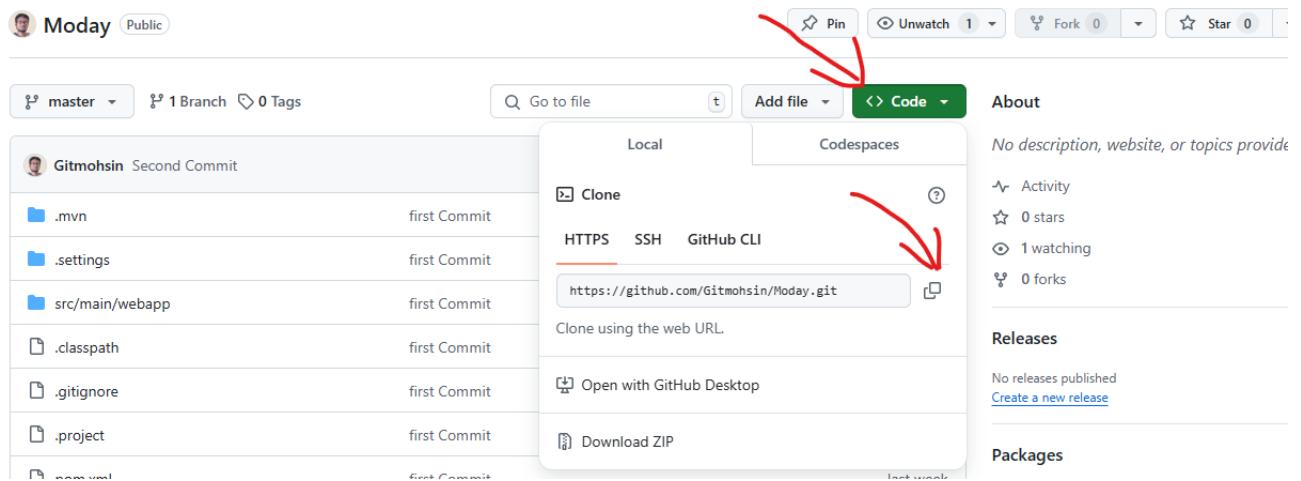
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job **+**



Step-10: Now take a new page -> login your **GitHub** -> check second experiment repository like **Devops_pipeline** -> copy the git link of **second experiment**.



Step-11: scroll down and go to **Source code management** section -> select **git** -> paste git link -> scroll down go to **build option** and type **clean install package** in **goals** option -> click on **apply** -> click on **save**.

Configure

General

Source Code Management

Triggers

Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Source Code Management

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

None

Git

Repositories

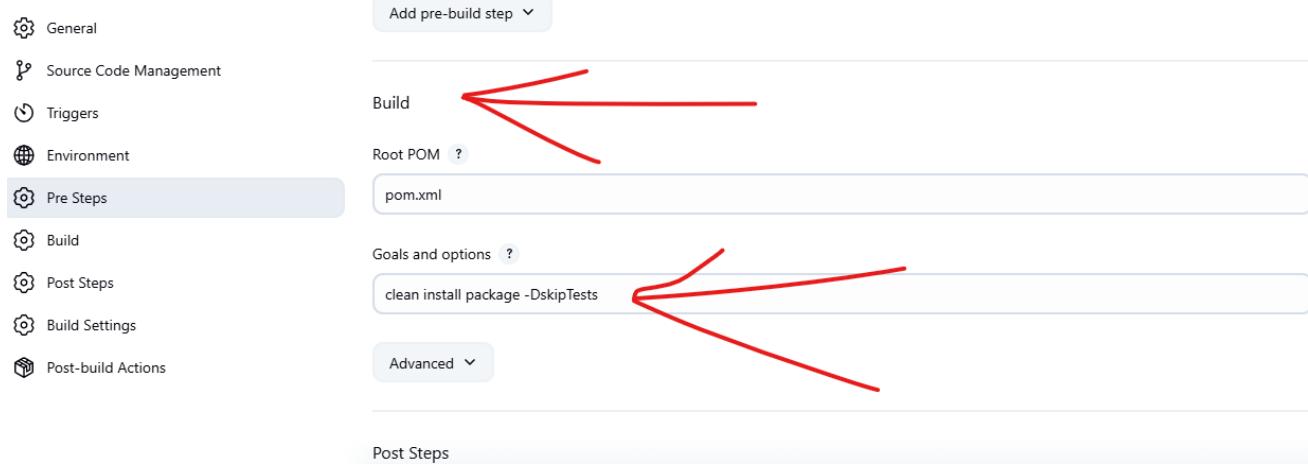
Repository URL

https://github.com/Gitmohsin/Moday.git

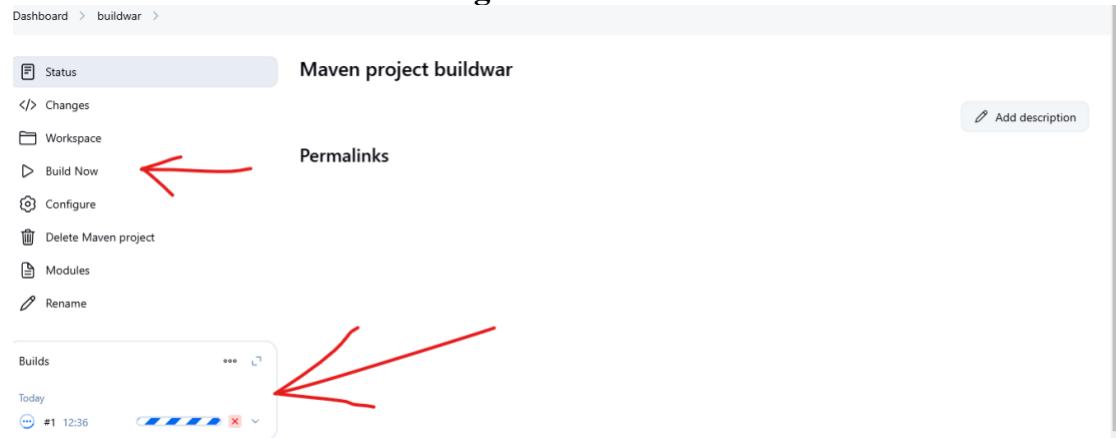
Credentials

- none -

+ Add



Step-12: After save the file -> it will show an option in left side build now -> click on build now -> It will take two to five minute for creating build file.



Step-12: After building it will show success with green mark -> click on that section -> it will open new section -> click on console output -> scroll down and check status, build success or not.

Status Maven project buildwar Ac

</> Changes Workspace Build Now Configure Delete Maven project Modules Rename

Builds Filter Today #1 12:36

Permalinks

- Last build (#1), 6 min 42 sec ago
- Last stable build (#1), 6 min 42 sec ago
- Last successful build (#1), 6 min 42 sec ago
- Last completed build (#1), 6 min 42 sec ago

#1 (20 Feb 2025, 12:36:30)

Status Changes Console Output Edit Build Information Delete build '#1' Timings Git Build Data Redeploy Artifacts See Fingerprints

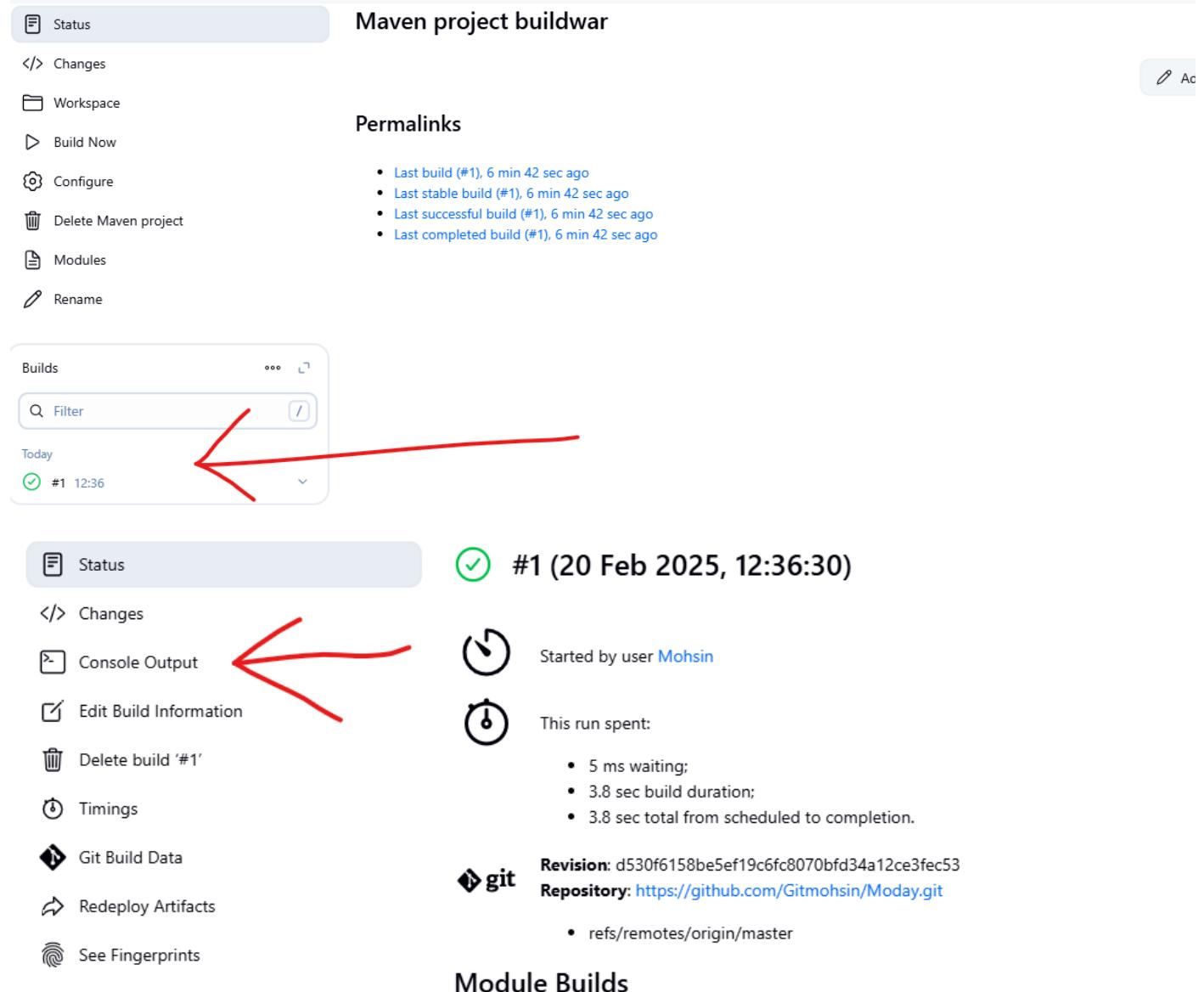
Started by user Mohsin

This run spent:

- 5 ms waiting;
- 3.8 sec build duration;
- 3.8 sec total from scheduled to completion.

git Revision: d530f6158be5ef19c6fc8070bfd34a12ce3fec53
Repository: <https://github.com/Gitmohsin/Moday.git>

Module Builds



Output:-

Dashboard > warjar >

Maven project warjar

Status Changes Add description

Workspace Build Now Permalinks

Configure Delete Maven project Modules Rename

Builds

Today #1 1:06 AM

```
utils/4.0.1/plexus-utils-4.0.1.jar (193 kB at 10 MB/s)
[INFO] Installing /home/ubuntu/.jenkins/workspace/warjar/pom.xml to
/home/ubuntu/.m2/repository/DevOpsCourse/MyDevPipeline/0.0.1-SNAPSHOT/MyDevPipeline-0.0.1-SNAPSHOT.pom
[INFO] Installing /home/ubuntu/.jenkins/workspace/warjar/target/MyDevPipeline.war to
/home/ubuntu/.m2/repository/DevOpsCourse/MyDevPipeline/0.0.1-SNAPSHOT/MyDevPipeline-0.0.1-SNAPSHOT.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  13.732 s
[INFO] Finished at: 2025-02-15T01:07:19Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /home/ubuntu/.jenkins/workspace/warjar/pom.xml to DevOpsCourse/MyDevPipeline/0.0.1-
SNAPSHOT/MyDevPipeline-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /home/ubuntu/.jenkins/workspace/warjar/target/MyDevPipeline.war to
DevOpsCourse/MyDevPipeline/0.0.1-SNAPSHOT/MyDevPipeline-0.0.1-SNAPSHOT.war
channel stopped
Finished: SUCCESS
```

Jenkins

Dashboard >

New Item Build History Project Relationship Check File Fingerprint Manage Jenkins My Views

All + Add description

S	W	Name	Last Success	Last Failure	Last Duration
Green	Yellow	warjar	8 min 40 sec #1	N/A	28 sec

Experiment-6

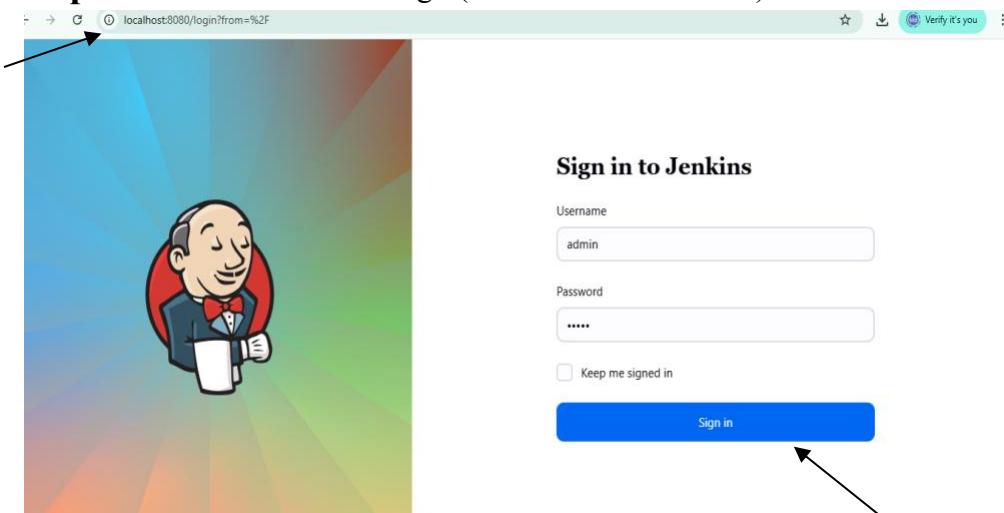
AIM:-Deploy the artifact on the Test Server

Softwares: Jenkins(port:8080), JDk 17 or 21, Tomcat 9 (port:9090), Gitbash

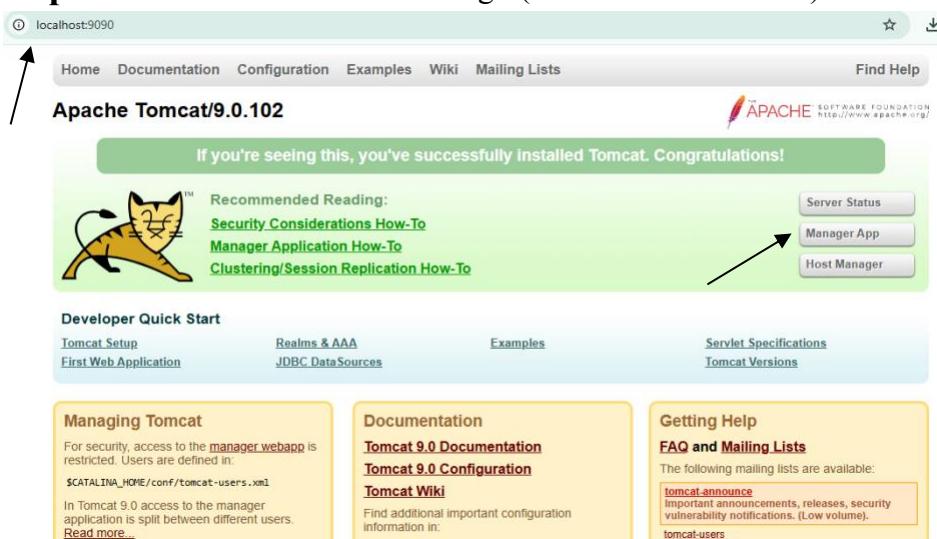
(Here we are creating one artifact war file. That war file we are deploy to test server(Tomcat server) using Jenkins Server)

(Note:-Images are purpose of understanding only don't paste it in records but output images paste it in output area)

Step1:-Lunch Jenkins and Login(Use AWS or Software)



Step2:-Lunch Tomcat Server and login(use AWS or Software)



Open Jenkins do following steps

Step 3:-Install plugins

Go to Manage Jenkins → Plugins → Install Plugin select verify Git → Go to Available plugins install Maven integration and Deploy to container install.



Dashboard >

+ New Item

Build History

All

+

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

S	W	Name ↓	Last Success	La
		deploy-war-tomcat	2 days 22 hr #6	2 c
		deploypipeline	2 days 22 hr #11	2 c
		week6	3 hr 47 min #4	N/

No builds in the queue.

Build Executor Status

Dashboard > Manage Jenkins

Icon: S M L

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

(0 of 2 executors busy)

Warnings have been published for the following currently installed components:

[Go to plugin manager](#)[Configure which of these warnings are shown](#)[EDDSA API Plugin 0.3.0-13.v7cb_69ed68f00:](#)[EDDSA implementation exhibits signature malleability](#)[A fix for this issue is available. Go to the plugin manager to update the plugin.](#)

System Configuration

System

Configure global settings and paths.

Tools

Configure tools, their locations and automatic installers.

Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes

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

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Q git

Updates

9

Available plugins

Installed plugins

Advanced settings

Download progress

Name ↓

[Git client plugin 6.1.2](#)

Utility plugin for Git support in Jenkins

[Report an issue with this plugin](#)[Git plugin 5.7.0](#)

This plugin integrates Git with Jenkins.

[Report an issue with this plugin](#)

Above picture three plugins are installed , If not showing please go to available plugins install it.

Screenshot of the Jenkins Plugins page. The search bar at the top contains the text "maven". A single result is shown: "Maven Integration plugin 3.25". The status is "Enabled" with a green checkmark icon. Below the plugin details, there is a link to "Report an issue with this plugin". On the left sidebar, the "Installed plugins" option is highlighted with a red arrow pointing to it from below.

Screenshot of the Jenkins Plugins page. The search bar at the top contains the text "deploy". A single result is shown: "Deploy to container Plugin 1.16". The status is "Enabled" with a green checkmark icon. Below the plugin details, there is a link to "Report an issue with this plugin". On the left sidebar, the "Installed plugins" option is highlighted with a red arrow pointing to it from below.

Step 4:-set up git path and maven installation in Tools

Go to manage Jenkins → Tools → In git installation select path(Git bash) if your using Jenkins software or AWS leave it → In maven Enter name Maven and default installation → Apply and Save.

The screenshot shows the Jenkins Manage Jenkins page. On the left, there's a sidebar with links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (which is selected), and My Views. A message at the top right says "Warnings have been published for the following currently installed components: EdDSA API Plugin 0.3.0-13x7cb_69ed68f00. EdDSA implementation exhibits signature malleability. A fix for this issue is available. Go to the [plugin manager](#) to update the plugin." Below this, the "System Configuration" section is visible, featuring icons for System, Tools, Nodes, Clouds, Plugins, and Appearance. An arrow points from the text "A fix for this issue is available. Go to the [plugin manager](#) to update the plugin." to the "Tools" icon.

The screenshot shows the Jenkins Tools configuration page under the "Manage Jenkins" section. It's specifically for "Git installations". A form is displayed with fields for "Name" (set to "Default"), "Path to Git executable" (set to "C:\Program Files\Git\bin\git.exe"), and a checkbox for "Install automatically". An arrow points from the text "A fix for this issue is available. Go to the [plugin manager](#) to update the plugin." to the "Tools" icon in the main navigation bar above.

Maven installations

Maven installations Edited

Add Maven

Maven

Name: **Maven**

Install automatically ?

Install from Apache

Version: 3.9.9

Add Installer

Save Apply

Step 5:-Creating New Job for deploy artifact in tomcat test server

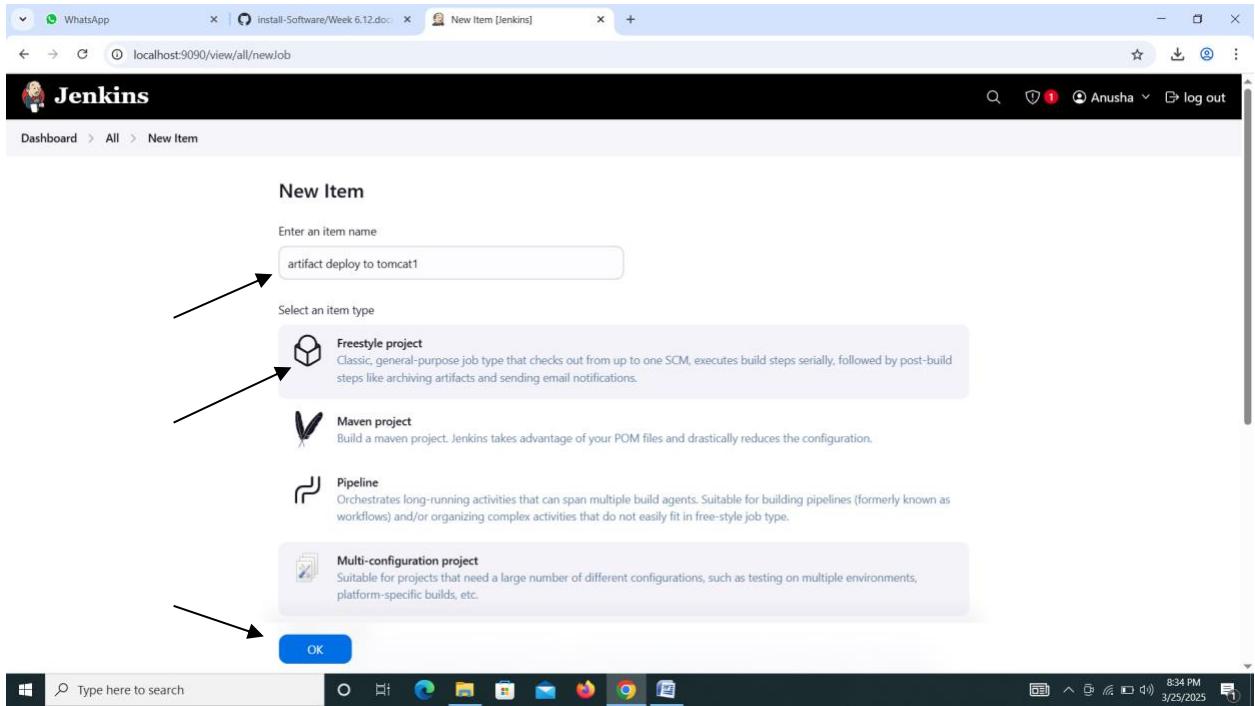
1. Create New Item → Enter name DeploytoTestServer → select freestyle project → click ok

+ New Item

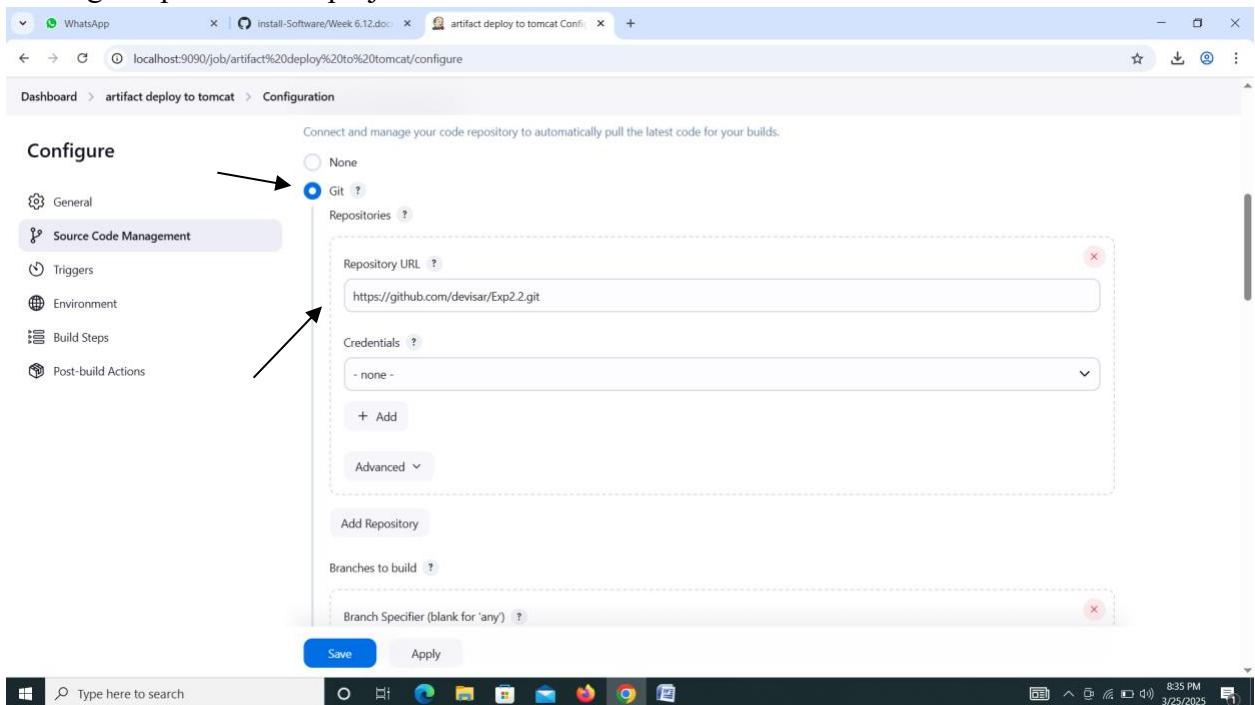
All +

S	W	Name	Last Success	Last Failure	Last Duration
Green	Cloud	artifact deploy to tomcat	7 days 4 hr #5	7 days 7 hr #4	14 sec
Green	Cloud	buildwar	26 days #6	26 days #5	1 min 9 sec
Green	Sun	deploy2	6 days 5 hr #4	N/A	21 sec
Green	Cloud	deploy3pipeline	6 days 4 hr #11	6 days 4 hr #8	16 sec

Build History Project Relationship Check File Fingerprint Manage Jenkins My Views Build Queue Build Executor Status REST API Jenkins 2.492.2



2. Select git → paste maven project url form Github → Must and should match branches



3. Go to Environment → Select Add timestamps to the Console Output

The screenshot shows the Jenkins configuration interface for a job named 'artifact deploy to tomcat'. The 'Triggers' section is currently active. A large black arrow points to the 'Add timestamps to the Console Output' checkbox, which is checked. Other visible options in this section include 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'.

4. Go to Build Setups → Select Invoke top-level Maven targets → Maven Version enter maven → Goals clean install

The screenshot shows the Jenkins configuration interface for the same job. The 'Build Steps' section is now active. A large black arrow points to the 'Invoke top-level Maven targets' step. Within this step, the 'Goals' field contains 'clean install'. Other fields in this step include 'Maven Version' set to 'Maven' and an 'Advanced' dropdown.

5. Go to Post-build Actions → Select Deploy war/ear to a container → WAR/EAR files enter **/*.war → select Containers Tomcat 9.x Remote → Add credentials like username, password, Id and description.-->Tomcat URL http://localhost:9090 → Apply and save.

Configure

Post-build Actions

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

Deploy war/ear to a container

WAR/EAR files ? `**/*.war`

Context path ? `/`

Containers

Tomcat 9.x Remote

Credentials `admin/******** (Deploy)`

+ Add

Tomcat URL ? `http://localhost:8080`

Containers

Tomcat 9.x Remote

Credentials `admin/******** (Deploy)`

+ Add

Tomcat URL ? `http://localhost:8080`

Advanced

Add Container ▾

Deploy on failure

Add post-build action ▾

Save **Apply**

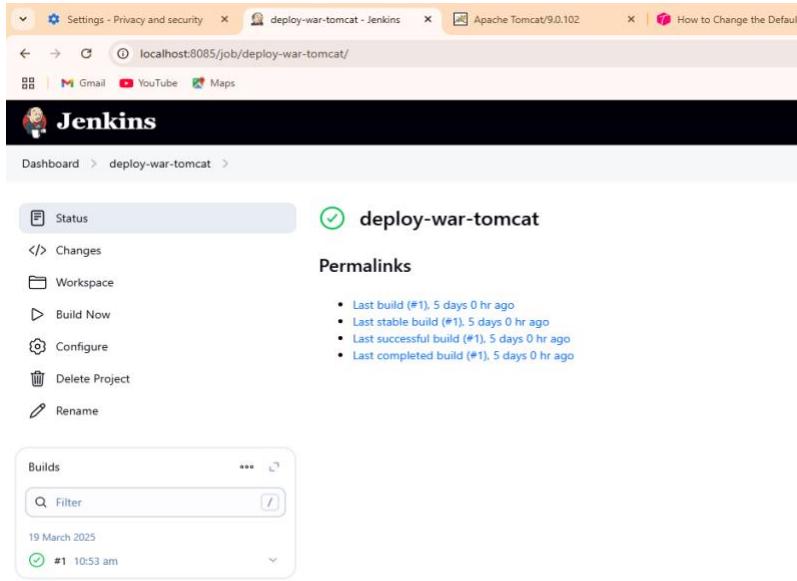
Step 6:- Build war File

1. Click Build now → Go to Build select console output now you got Successfully Finish.

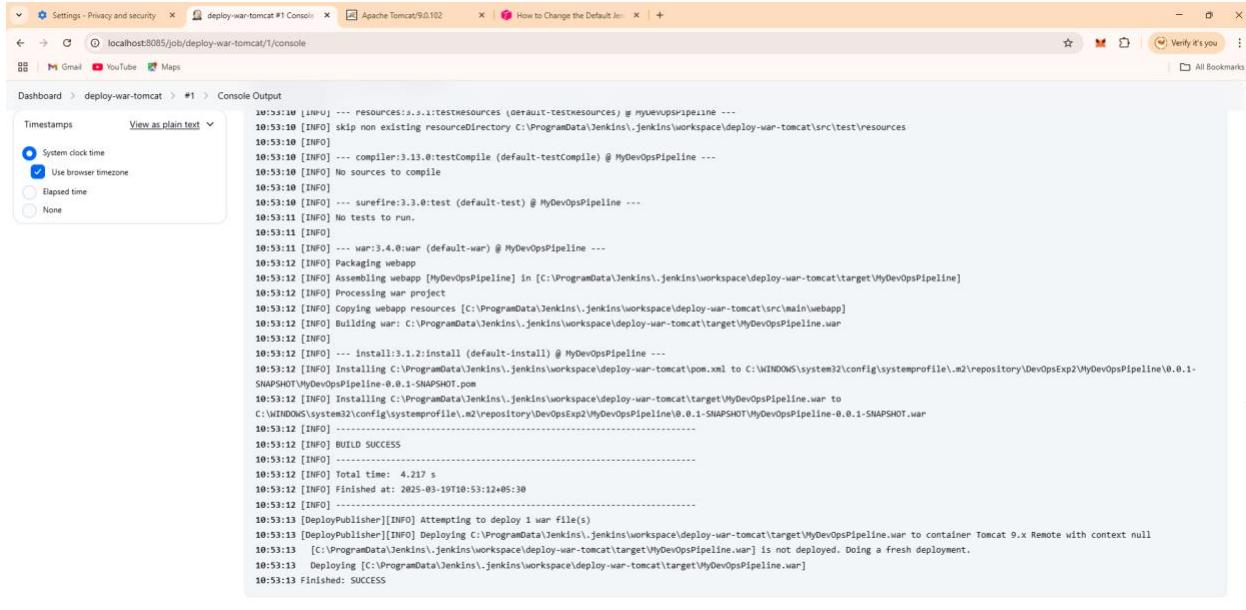
The screenshot shows the Jenkins dashboard for the 'deploy-war-tomcat' job. The build status is green with a checkmark, indicating a successful build (#1) from March 19, 2025, at 10:53 am. The sidebar on the left includes options like 'Build Now', 'Configure', 'Delete Project', and 'Rename'. A black arrow points to the 'Build Now' button.

The screenshot shows the Jenkins console output for build #1. The log output details the deployment process, starting with resource skipping, compiler compilation, and surefire tests. It then moves on to war packaging, assembly, processing, copying resources, building the war file, and finally installing it to the Tomcat target. The log concludes with a BUILD SUCCESS message and a total execution time of 4.217 seconds. A black arrow points to the end of the log output.

Output:-



The screenshot shows the Jenkins dashboard for the 'deploy-war-tomcat' job. The job is marked as 'Status: Success'. It has four builds listed, all from 5 days ago. The most recent build (#1) was run at 10:53 am on March 19, 2025. A 'Build Now' button is visible.



The screenshot shows the Jenkins console output for the 'deploy-war-tomcat' job. The log output is as follows:

```

10:53:10 [INFO] --- resources:3.1.1:testResources (default-testResources) @ MyDevOpsPipeline ---
10:53:10 [INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\src\test\resources
10:53:10 [INFO]
10:53:10 [INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ MyDevOpsPipeline ---
10:53:10 [INFO] No sources to compile
10:53:10 [INFO]
10:53:10 [INFO] --- surefire:3.3.0:test (default-test) @ MyDevOpsPipeline ---
10:53:11 [INFO] No tests to run.
10:53:11 [INFO]
10:53:11 [INFO] --- war:3.4.8:war (default-war) @ MyDevOpsPipeline ---
10:53:12 [INFO] Packaging webapp
10:53:12 [INFO] Assembling webapp [MyDevOpsPipeline] in [C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline]
10:53:12 [INFO] Processing war project
10:53:12 [INFO] Copying webapp Resources [C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\src\main\webapp]
10:53:12 [INFO] Building war: C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline.war
10:53:12 [INFO]
10:53:12 [INFO] --- install:3.1.2:install (default-install) @ MyDevOpsPipeline ---
10:53:12 [INFO] Installing C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\pom.xml to C:\Windows\system32\config\systemprofile\.m2\repository\DevOpsExp2\MyDevOpsPipeline\0.0.1-SNAPSHOT\MyDevOpsPipeline-0.0.1-SNAPSHOT.pom
10:53:12 [INFO] Installing C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline.war to C:\Windows\system32\config\systemprofile\.m2\repository\DevOpsExp2\MyDevOpsPipeline\0.0.1-SNAPSHOT\MyDevOpsPipeline-0.0.1-SNAPSHOT.war
10:53:12 [INFO] -----
10:53:12 [INFO] BUILD SUCCESS
10:53:12 [INFO] -----
10:53:12 [INFO] Total time: 4.217 s
10:53:12 [INFO] Finished at: 2025-03-19T10:53:12+05:30
10:53:12 [INFO] -----
10:53:13 [DeployPublisher][INFO] Attempting to deploy 1 war file(s)
10:53:13 [DeployPublisher][INFO] Deploying C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline.war to container Tomcat 9.x Remote with context null
10:53:13 [INFO] [C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline.war] is not deployed. Doing a fresh deployment.
10:53:13 [INFO] Deploying [C:\ProgramData\Jenkins\jenkins\workspace\deploy-war-tomcat\target\MyDevOpsPipeline.war]
10:53:13 [INFO] finished: SUCCESS

```

The screenshot shows the Apache Tomcat Web Application Manager interface. At the top, there's a message bar with 'Message: OK'. Below it is a navigation bar with tabs: 'Manager', 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. The main area is titled 'Tomcat Web Application Manager' and contains a table of applications.

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

Below the table is a 'Deploy' section with fields for 'Context Path', 'Version (for parallel deployment)', 'XML Configuration file path', and 'WAR or Directory path'. A 'Deploy' button is at the bottom of this section.

At the bottom of the page, the browser address bar shows 'localhost:9090/MyDevOpsPipeline/' and the page content 'Hello Students12345'.

Experiment-7

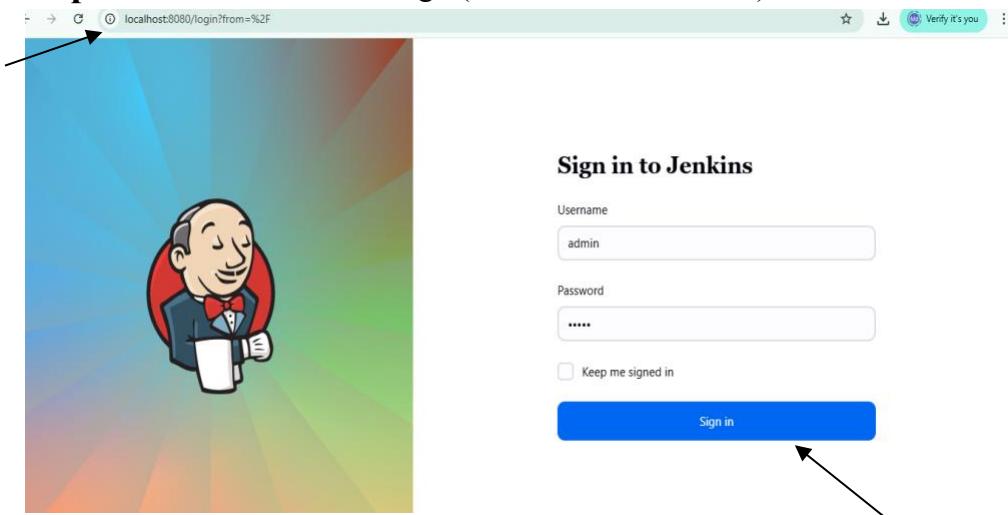
AIM:-Perform Automation using Jenkins.

Softwares:Jenkins(port:8080),JDk 17 or 21,Tomcat 9 (port:9090),Gitbash

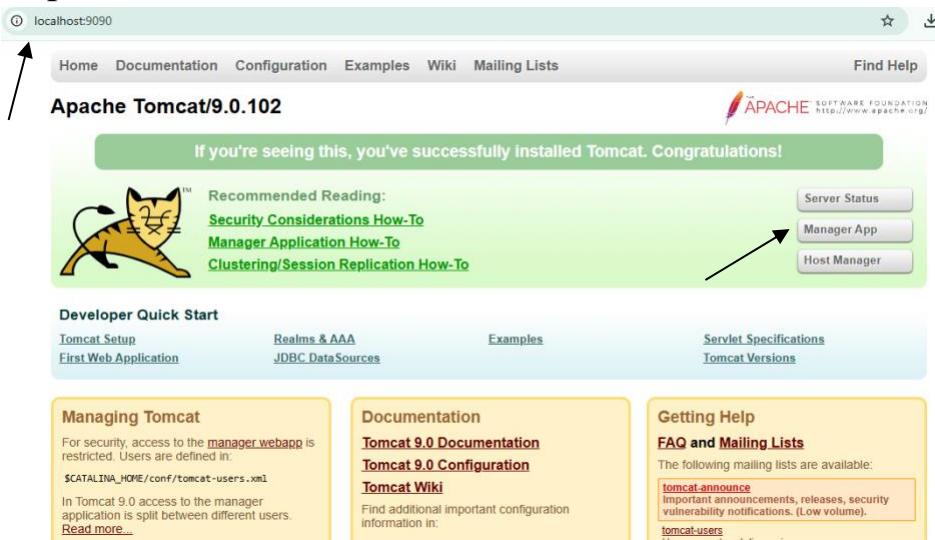
(Here we are creating one artifact war file. That war file we are deploy to test server(Tomcat server) using Jenkins Server)

(Note:-Images are purpose of understanding only don't paste it in records but output images paste it in output area)

Step1:-Lunch Jenkins and Login(Use AWS or Software)



Step2:-Lunch Tomcat Server and login(use AWS or Software)



Open Jenkins do following steps

Step 3:-Install plugins

Go to Manage Jenkins → Plugins → Install Plugin select verify Git → Go to Available plugins install Maven integration and Deploy to container install.



Dashboard >

+ New Item

Build History

All

+

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

S	W	Name ↓	Last Success	La
		deploy-war-tomcat	2 days 22 hr #6	2 c
		deploypipeline	2 days 22 hr #11	2 c
		week6	3 hr 47 min #4	N/

No builds in the queue.

Build Executor Status

Dashboard > Manage Jenkins

Icon: S M L

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

(0 of 2 executors busy)

Warnings have been published for the following currently installed components:

[Go to plugin manager](#)[Configure which of these warnings are shown](#)[EDDSA API Plugin 0.3.0-13.v7cb_69ed68f00:](#)[EDDSA implementation exhibits signature malleability](#)[A fix for this issue is available. Go to the plugin manager to update the plugin.](#)

System Configuration

System

Configure global settings and paths.

Tools

Configure tools, their locations and automatic installers.

Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes

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

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Q git

Updates

9

Available plugins

Installed plugins

Advanced settings

Download progress

Name ↓

[Git client plugin 6.1.2](#)

Utility plugin for Git support in Jenkins

[Report an issue with this plugin](#)[Git plugin 5.7.0](#)

This plugin integrates Git with Jenkins.

[Report an issue with this plugin](#)

Above picture three plugins are installed , If not showing please go to available plugins install

The screenshot shows the Jenkins 'Plugins' page. A search bar at the top contains the text 'maven'. Below it, a sidebar on the left lists 'Updates', 'Available plugins', 'Installed plugins' (which is selected and highlighted in grey), and 'Advanced settings'. The main area displays the 'Maven Integration plugin' version 3.25, which is enabled. A tooltip for the plugin states: 'This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.' There is also a link to 'Report an issue with this plugin'. The bottom right corner of the page indicates 'Jenkins 2.492.2'.

This screenshot is identical to the one above, showing the Jenkins 'Plugins' page with the 'Deploy to container Plugin' version 1.16 installed. The sidebar shows 'Installed plugins' is selected. The tooltip for the plugin states: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'. The bottom right corner indicates 'Jenkins 2.492.2'.

Step 4:-set up git path and maven installation in Tools

Go to manage Jenkins → Tools → In git installation select path(Git bash) if your using Jenkins software or AWS leave it → In maven Enter name Maven and default installation → Apply and Save.

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (which is selected), and My Views. A message at the top right indicates a warning about the EdDSA API Plugin. The main area is titled 'System Configuration' and contains sections for System, Tools, Nodes, Clouds, Plugins, and Appearance.

This screenshot shows the 'Tools' section of the Jenkins Manage Jenkins page. It's titled 'Git installations'. A form is displayed for adding a new Git configuration, with fields for 'Name' (set to 'Default'), 'Path to Git executable' (set to 'C:\Program Files\Git\bin\git.exe'), and an unchecked checkbox for 'Install automatically'. An arrow points from the top of the 'Tools' section towards this form.

This screenshot shows the 'Gradle installations' section of the Jenkins Manage Jenkins page. It includes a 'Save' button and an 'Apply' button. An arrow points from the bottom of the 'Tools' section towards this 'Gradle installations' section.

Maven installations

Add Maven

Maven

Name: Maven

Install automatically

Install from Apache

Version: 3.9.9

Add Installer

Save Apply

Step 5:-Creating New Job for deploy artifact in tomcat test server

6. Create New Item → Enter name DeployPipeline → select pipeline project → click ok

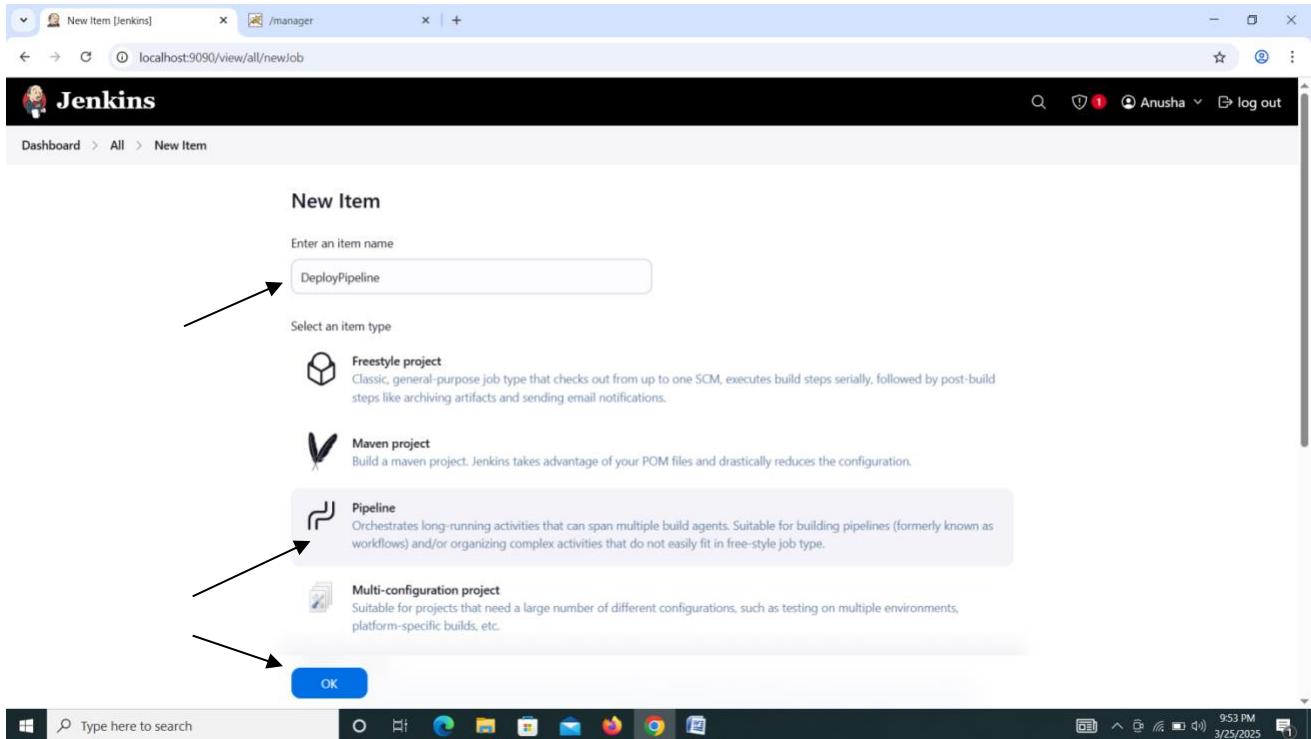
+ New Item

All +

S	W	Name	Last Success	Last Failure	Last Duration
Green	Cloud	artifact deploy to tomcat	7 days 4 hr #5	7 days 7 hr #4	14 sec
Green	Cloud	buildwar	26 days #6	26 days #5	1 min 9 sec
Green	Sun	deploy2	6 days 5 hr #4	N/A	21 sec
Green	Cloud	deploy3pipeline	6 days 4 hr #11	6 days 4 hr #8	16 sec

Build History Project Relationship Check File Fingerprint Manage Jenkins My Views Build Queue Build Executor Status REST API Jenkins 2.492.2

localhost:9090/view/all/newjob



Step 2:-Copy paste below commands pipeline script

```

pipeline{
    agent any
    tools{
        maven 'Maven'
    }
    stages{
        stage('clone the repository'){
            steps{
                git url: 'https://github.com/devisar/Exp2.5.git'
            }
        }
        stage('Build the code'){
            steps{
                bat 'mvn clean install'
            }
        }
        stage('deploy to tomcat'){
            steps{
                deploy adapters: [tomcat9(credentialsId: 'Deploy', path: '', url: 'http://localhost:8080')], contextPath: null, war: '**/*.war'
            }
        }
    }
}

```

The screenshot shows the Jenkins Pipeline configuration page for a job named 'deploy3pipeline'. The pipeline script is defined in the 'Pipeline script' section:

```

9+     steps{
10+       git url: 'https://github.com/devisar/Exp2.5.git'
11+
12+
13+   stage('Build the code'){
14+     steps{
15+       bat 'mvn clean install'
16+
17+     }
18+   stage('deploy to tomcat'){
19+     steps{
20+       deploy adapters: [tomcat9(credentialsId: 'Deploy', path: '', url: 'http://localhost:8080')], contextPath: null, war: '**/*'
21+
22+     }
23+
24+   }

```

A large arrow points from the left towards the script editor area. Another arrow points from the right towards the 'Pipeline Syntax' label.

Step 6:-deploy to tomcat you want to create pipeline script go to → pipeline syntax → Select in sample step deploy: Deploy war/ear to a container → enter war/ear files: **/*.war → Select Containers tomcat 9 → select credentials tomcat user name and password. → Enter Tomcat Url like http://localhost:9090. → Click Generate Pipeline Script copy and paste it original pipeline deployment stage.

The screenshot shows a Jenkins Pipeline Syntax configuration page. At the top, there are several tabs: 'Selenium Grid', 'localhost:8080/job/DeployPipelineweek7/pipeline-syntax', 'DeployPipelineweek7 Config', 'Pipeline Syntax Snippet Generator', and 'Apache Tomcat/9.0.102'. The active tab is 'localhost:8080/job/DeployPipelineweek7/pipeline-syntax'.

The main content area is titled 'Pipeline Syntax' and shows a 'Steps' section. A 'Sample Step' is displayed, which is a 'deploy' step. The configuration for this step includes:

- WAR/EAR files**: `**/*.war`
- Context path**: `''`
- Containers** section:
 - A container named 'Tomcat 9.x Remote' is selected.
 - Credentials**: `deployer/******** (Deployer12345)`
 - Tomcat URL**: `http://localhost:9090`

Below the container configuration, there is an 'Advanced' dropdown and a checkbox for 'Deploy on failure'. A large blue button labeled 'Generate Pipeline Script' is prominently displayed. To the right of this button, the generated pipeline script is shown:

```
deploy adapters: [tomcat9(credentialsId: 'Deployer12345', path: '', url: 'http://localhost:9090')], contextPath: null, war: '**/*.war'
```

The screenshot shows the Jenkins Pipeline configuration page. On the left, there's a sidebar with tabs: General, Triggers, Pipeline (which is selected and highlighted in grey), and Advanced. The main area is titled 'Pipeline' with the sub-section 'Definition'. A dropdown menu shows 'Pipeline script'. Below it is a code editor containing Groovy pipeline code:

```

9+   steps{
10+     git url: 'https://github.com/devisar/Exp2.5.git'
11+
12+
13+   }
14+   stage('Build the code'){
15+     steps{
16+       bat 'mvn clean install'
17+     }
18+   }
19+   stage('deploy to tomcat'){
20+     steps{
21+       deploy adapters: [tomcat9(credentialsId: 'Deploy', path: '', url: 'http://localhost:8080')], contextPath: null, war: '**/*.war'
22+     }
23+   }
24+

```

Use Groovy Sandbox ?

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

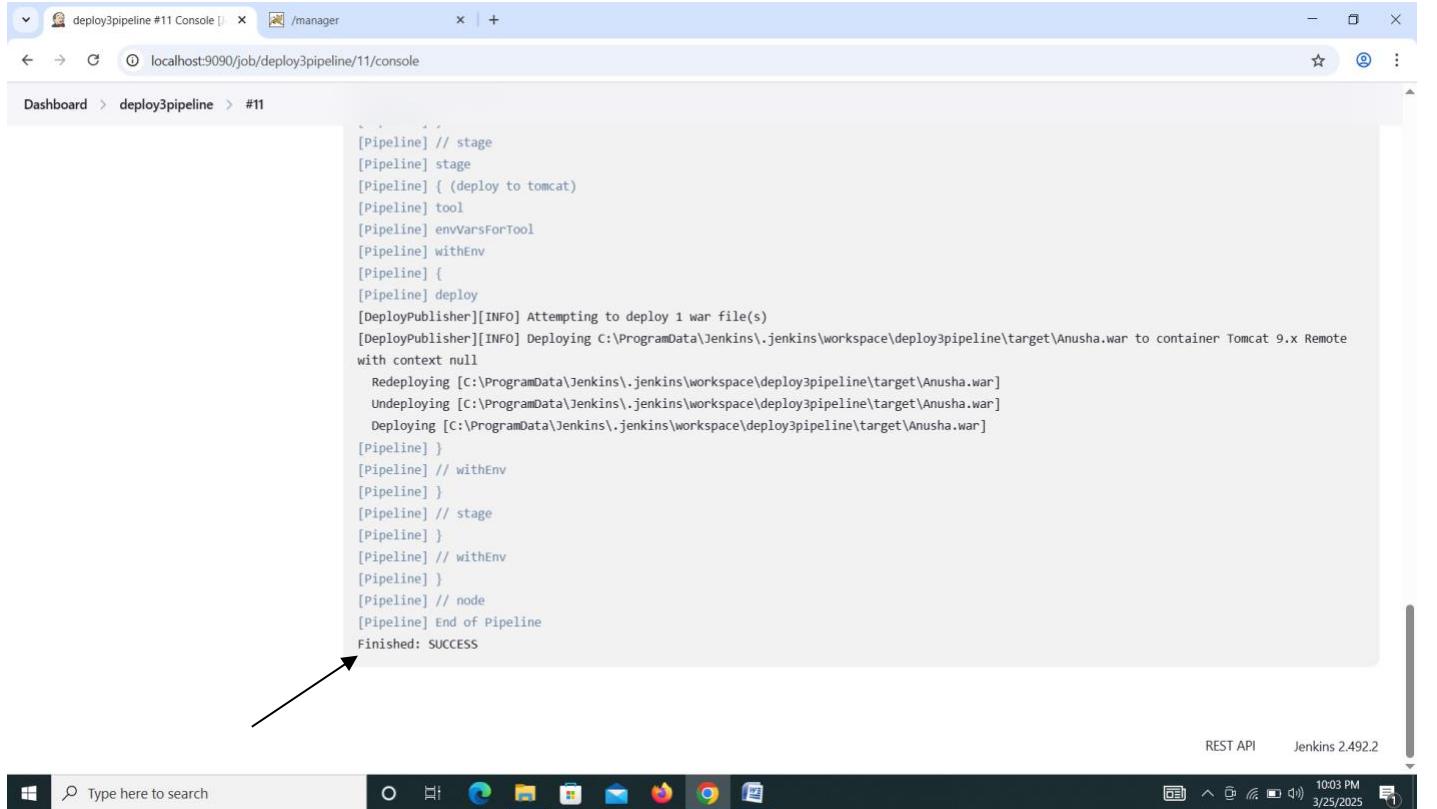
Step 7:- Build war File

2. Click Build now → Go to

The screenshot shows the Jenkins dashboard for the 'deploy3pipeline' job. On the left, there's a sidebar with options: Status (selected), Changes, Build Now (highlighted with an arrow), Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. The main area shows the job name 'deploy3pipeline' with a green checkmark icon. Below it is a 'Permalinks' section with a list of recent builds. At the bottom is a 'Builds' table showing the last four builds:

#	Build Time
11	4:14 PM
10	4:13 PM
9	4:10 PM
8	4:09 PM

select console output now you got Successfully Finish.



```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deploy to tomcat)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] deploy
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying C:\ProgramData\Jenkins\jenkins\workspace\deploy3pipeline\target\Anusha.war to container Tomcat 9.x Remote with context null
    Redeploying [C:\ProgramData\Jenkins\jenkins\workspace\deploy3pipeline\target\Anusha.war]
    Undeploying [C:\ProgramData\Jenkins\jenkins\workspace\deploy3pipeline\target\Anusha.war]
    Deploying [C:\ProgramData\Jenkins\jenkins\workspace\deploy3pipeline\target\Anusha.war]
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.492.2



Output:-

The screenshot shows the Jenkins interface for the 'deploy3pipeline' job. The left sidebar contains options like Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. The main content area displays the build history for March 19, 2025, showing builds #11, #10, #9, and #8. Build #11 is the most recent and successful. A 'Permalinks' section lists all builds from #8 to #11. The bottom of the page includes a search bar and navigation links.

The screenshot shows the Jenkins console for build #11. The log output is displayed, showing the deployment process. It starts with pipeline stages, then logs from the DeployPublisher plugin attempting to deploy 'Anusha.war' to a Tomcat 9.x container. The log ends with a 'Finished: SUCCESS' message. The Jenkins status bar at the bottom indicates the build was run on March 25, 2025, at 10:03 PM.

This screenshot is identical to the one above, showing the Jenkins console for build #11. The log output is identical, showing the deployment process and a successful completion. The Jenkins status bar at the bottom indicates the build was run on March 25, 2025, at 10:03 PM.

The screenshot shows the Apache Tomcat Web Application Manager interface. At the top, there's a navigation bar with tabs for 'Manager', 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. Below this is a section titled 'Applications' with a table. The table has columns for 'Path', 'Version', 'Display Name', 'Running', 'Sessions', and 'Commands'. The 'Display Name' column lists various Tomcat components like 'Welcome to Tomcat', 'Archetype Created Web Application', etc. The 'Commands' column for each row contains buttons for 'Start', 'Stop', 'Reload', and 'Undeploy', along with a link to 'Expire sessions with idle ≥ [value] minutes'. The application '/Anusha' is highlighted with a green background and has an arrow pointing to it from the left. The bottom of the screen shows a taskbar with icons for File, Home, Task View, Start, Taskbar View, Task Switcher, Taskbar Help, and Taskbar Settings.

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

Hello World!

Experiment 8

Week 8

AIM:-Build and deploy a grid for Chrome and Firefox based testing.

Step1:- Download Selenium Grid

A screenshot of a search results page from a search engine. The search query 'selenium download' is highlighted with a red arrow. Below the search bar, there are category links: All, Videos, Images, Shopping, Short videos, News, Forums, and More. The 'All' link is underlined.



Selenium

<https://www.selenium.dev/downloads> :

Downloads

Downloads. Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and ...

Selenium Server (Grid)

The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).

Latest stable version [4.30.0](#)

To use the Selenium Server in a Grid configuration see the [documentation](#).

To run the Grid with popular browsers using Docker see the [repository](#).

To deploy the Grid to Kubernetes cluster see the Helm chart [configuration](#).

Step2:-Download Chrome Driver

chrome driver download

All Videos Images Short videos Shopping News Forums More



Chrome for Developers

[https://developer.chrome.com › Docs › Chromedriver](https://developer.chrome.com/docs/chromedriver/) ::

Downloads | ChromeDriver - Chrome for Developers

17 Dec 2024 — Earlier Chrome versions · [ChromeDriver 114.0.5735.90](#) · [ChromeDriver 114.0.5735.90](#)
· [ChromeDriver 113.0.5672.63](#) · [ChromeDriver 113.0.5672.24](#).

Downloads



⚠ Warning: If you're using Chrome version 115 or newer, consult the [Chrome for Testing availability dashboard](#). On this dashboard, you'll find [JSON endpoints](#) to download specific ChromeDriver versions.

Earlier Chrome versions ↪

Chrome for Testing availability



This page lists the latest available cross-platform Chrome for Testing versions and assets per Chrome

Consult [our JSON API endpoints](#) if you're looking to build automated scripts based on Chrome for Test

Last updated @ 2025-03-28T07:09:34.087Z

Channel	Version	Revision	Status
Stable	134.0.6998.165	r1415337	<input checked="" type="checkbox"/>
<u>Beta</u>	135.0.7049.41	r1427262	<input checked="" type="checkbox"/>
<u>Dev</u>	136.0.7091.2	r1437865	<input checked="" type="checkbox"/>
<u>Canary</u>	136.0.7093.0	r1438506	<input checked="" type="checkbox"/>

Below picture address copy and paste it new tab. Automatically chrome driver downloaded

Version: 134.0.6998.165 (r1415337)

Binary	Platform	URL	HTTP status
chrome	linux64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/linux64/chrome-linux64.zip	200
chrome	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/mac-arm64/chrome-mac-arm64.zip	200
chrome	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/mac-x64/chrome-mac-x64.zip	200
chrome	win32	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win32/chrome-win32.zip	200
chrome	win64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win64/chrome-win64.zip	200
chromedriver	linux64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/linux64/chromedriver-linux64.zip	200
chromedriver	mac-arm64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/mac-arm64/chromedriver-mac-arm64.zip	200
chromedriver	mac-x64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/mac-x64/chromedriver-mac-x64.zip	200
chromedriver	win32	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win32/chromedriver-win32.zip	200
chromedriver	win64	https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win64/chromedriver-win64.zip	200

https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win64/chromedriver-win64.zip

<https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win64/chromedriver-win64.zip>

https://storage.googleapis.com/chrome-for-testing-public/134.0.6998.165/win64/chromedriver-win64.zip - Google Search

Step3:-Download Gecko Driver

firefox driver download

All Images Videos Shopping Short videos News Forums ::



[GitHub](#)

<https://github.com/mozilla/geckodriver/releases>

⋮

You visit often

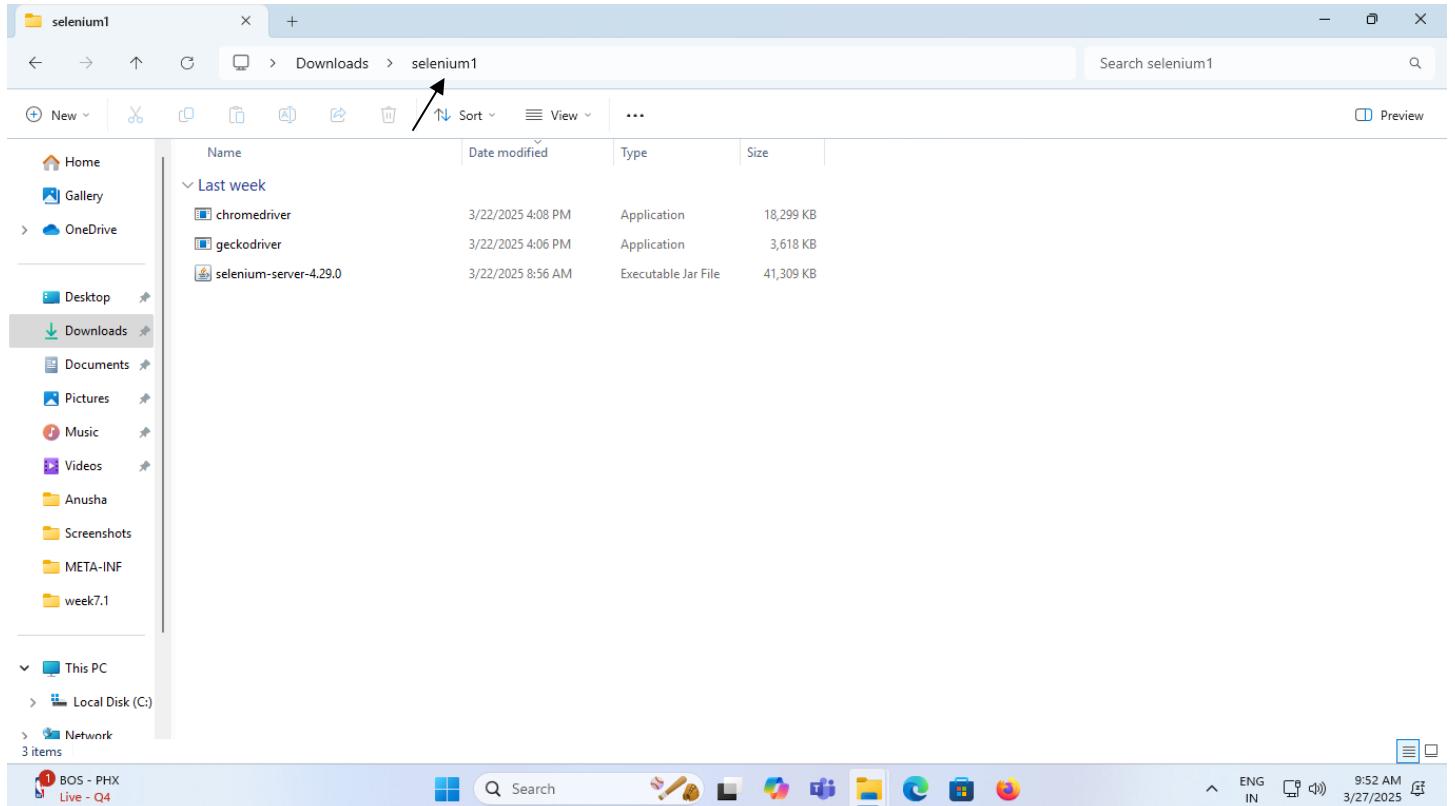
[Releases · mozilla/geckodriver](#)

25 Feb 2025 — WebDriver for Firefox. Contribute to mozilla/geckodriver development on GitHub.

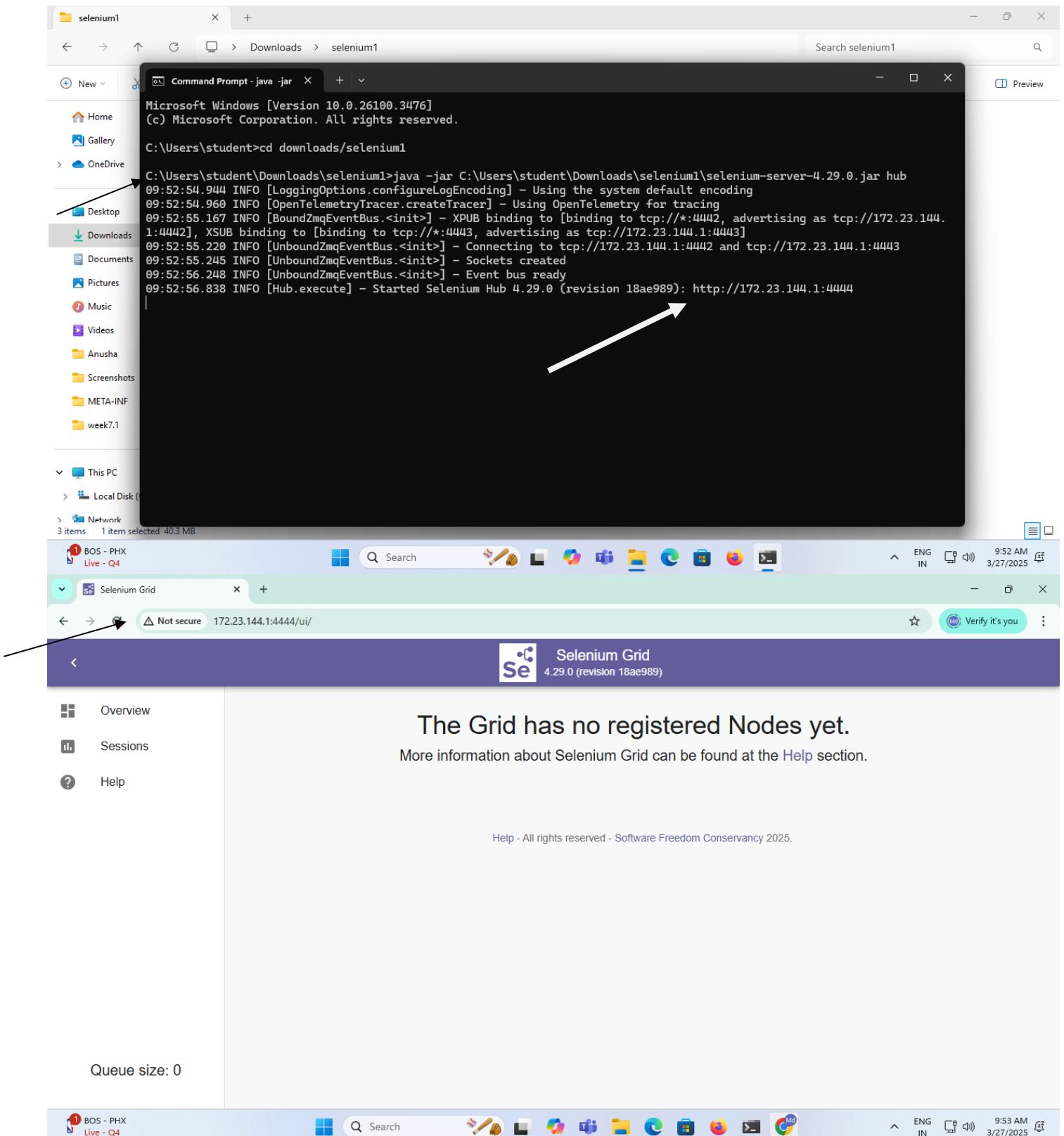
▼ Assets

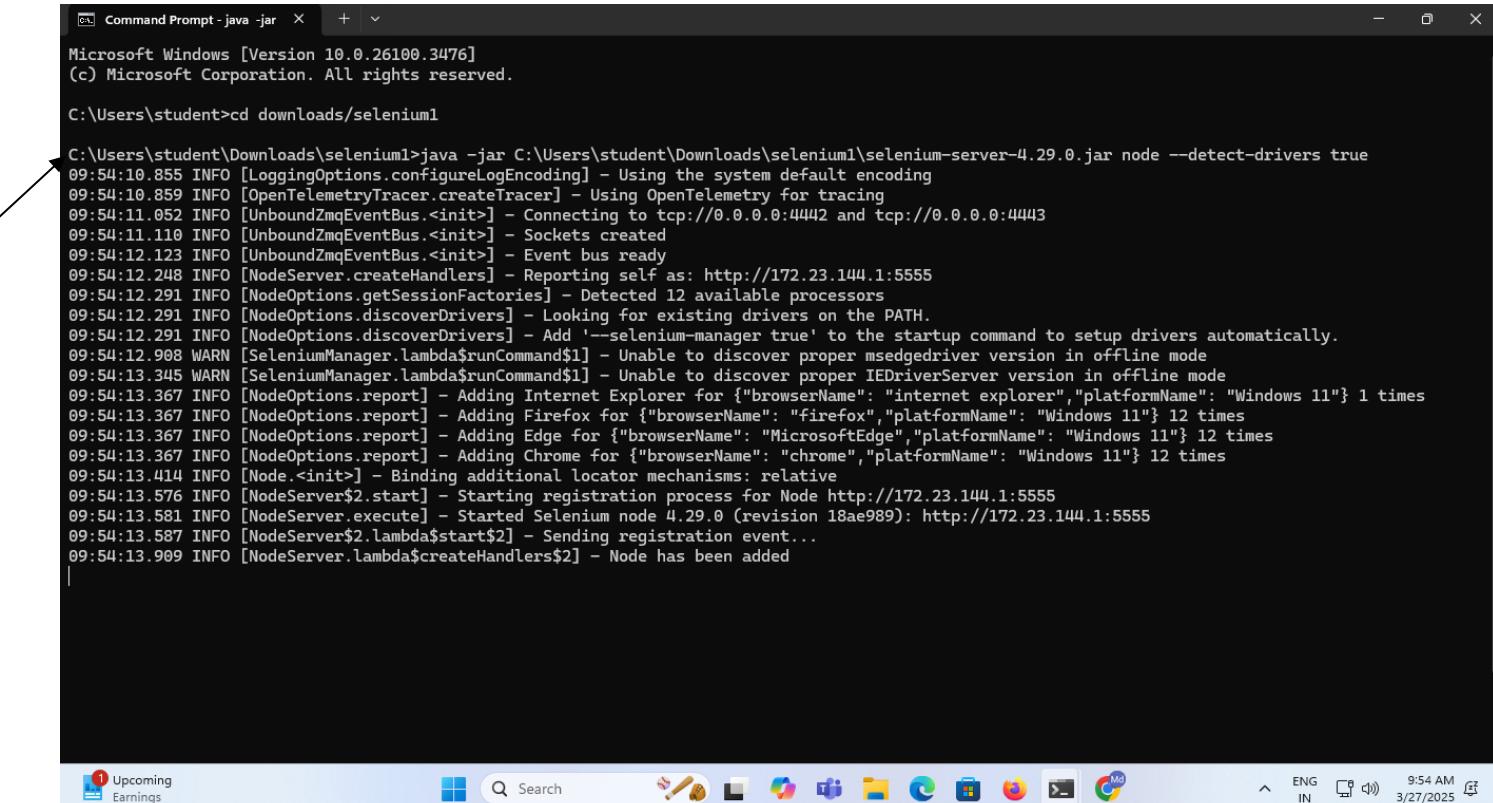
13

geckodriver-v0.36.0-linux-aarch64.tar.gz	2.07 MB	last month
geckodriver-v0.36.0-linux-aarch64.tar.gz.asc	833 Bytes	last month
geckodriver-v0.36.0-linux32.tar.gz	2.16 MB	last month
geckodriver-v0.36.0-linux32.tar.gz.asc	833 Bytes	last month
geckodriver-v0.36.0-linux64.tar.gz	2.2 MB	last month
geckodriver-v0.36.0-linux64.tar.gz.asc	833 Bytes	last month
geckodriver-v0.36.0-macos-aarch64.tar.gz	1.97 MB	last month
geckodriver-v0.36.0-macos.tar.gz	2.19 MB	last month
geckodriver-v0.36.0-win-aarch64.zip	1.62 MB	last month
geckodriver-v0.36.0-win32.zip	1.7 MB	last month
Source code (zip)		last month
Source code (tar.gz)		last month
Show all 13 assets		

Step 4:-Put all drivers one folder**Step 5:-Run hub use following command and node also separate cmd. In cmd change directory where available all drivers**

1. java -jar selenium-server-4.29.0.jar hub
 2. java -jar selenium-server-4.29.0.jar node --detect-drivers true
- Also run chrome using ip address path it will generate after running hub and verify

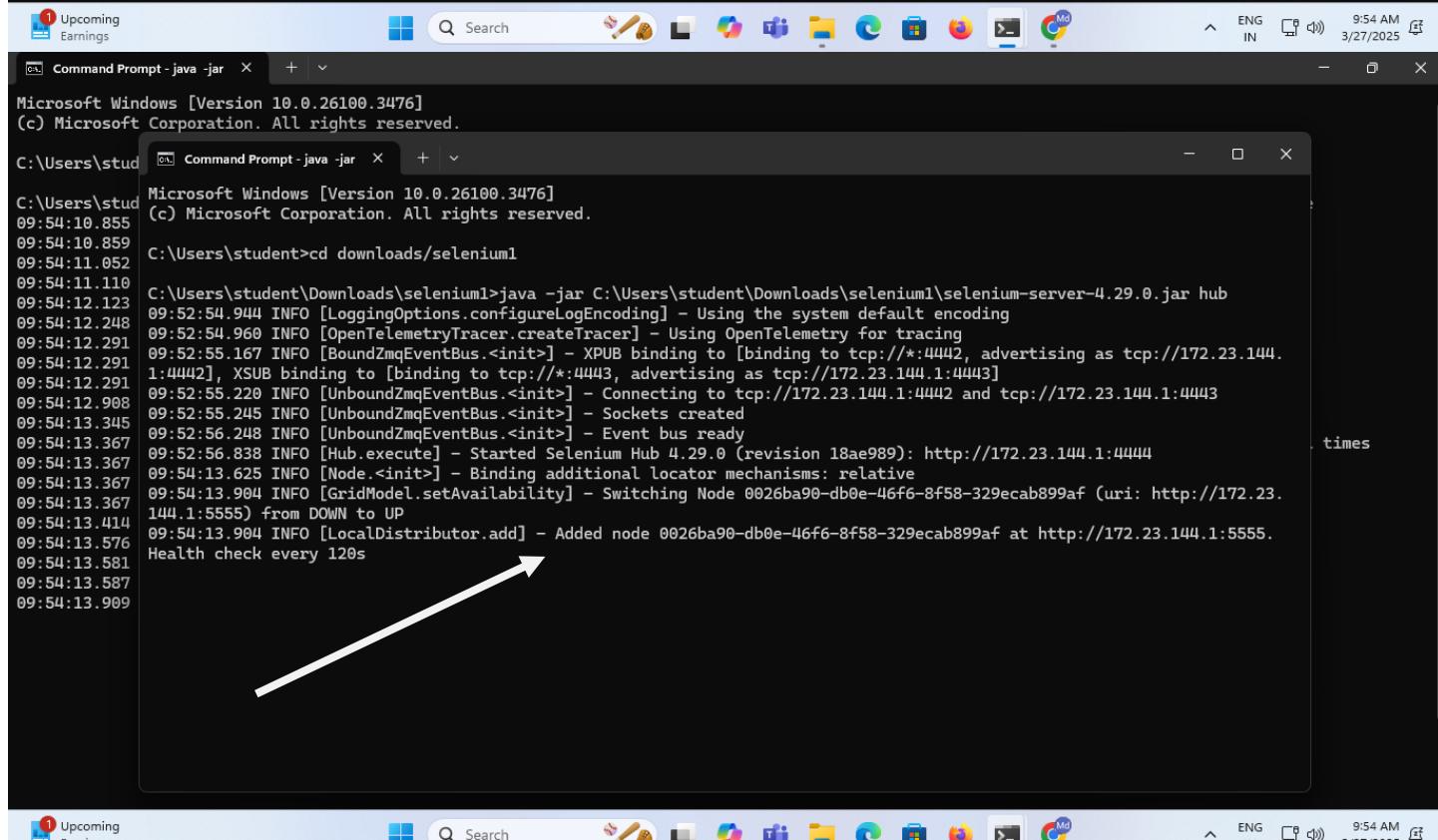




```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd downloads\selenium1

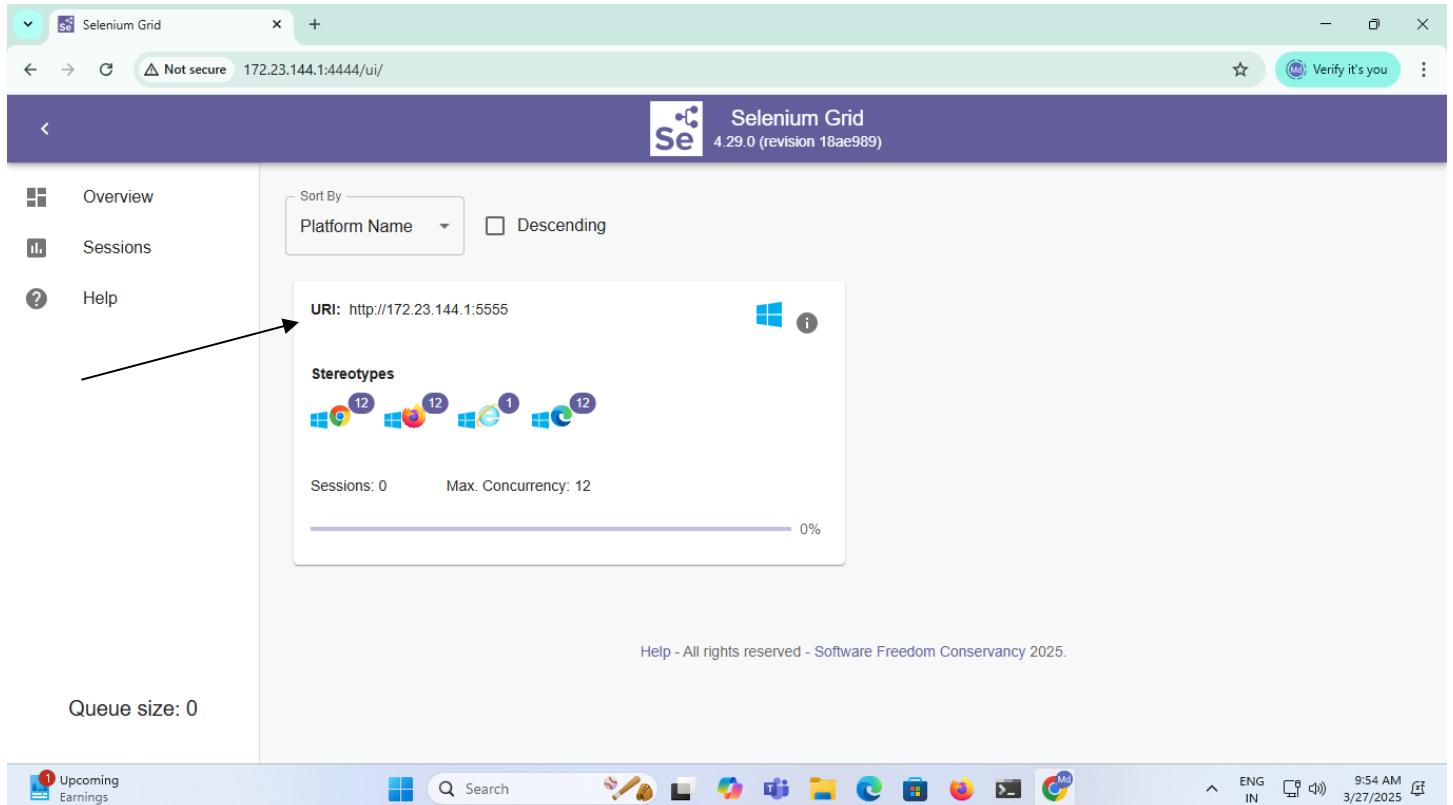
C:\Users\student\Downloads\selenium1>java -jar C:\Users\student\Downloads\selenium1\selenium-server-4.29.0.jar node --detect-drivers true
09:54:10.855 INFO [LoggingOptions.configureLogEncoding] - Using the system default encoding
09:54:10.859 INFO [OpenTelemetryTracer.createTracer] - Using OpenTelemetry for tracing
09:54:11.052 INFO [UnboundZmqEventBus.<init>] - Connecting to tcp://0.0.0.0:4442 and tcp://0.0.0.0:4443
09:54:11.110 INFO [UnboundZmqEventBus.<init>] - Sockets created
09:54:12.123 INFO [UnboundZmqEventBus.<init>] - Event bus ready
09:54:12.248 INFO [NodeServer.createHandlers] - Reporting self as: http://172.23.144.1:5555
09:54:12.291 INFO [NodeOptions.getSessionFactories] - Detected 12 available processors
09:54:12.291 INFO [NodeOptions.discoverDrivers] - Looking for existing drivers on the PATH.
09:54:12.291 INFO [NodeOptions.discoverDrivers] - Add '--selenium-manager true' to the startup command to setup drivers automatically.
09:54:12.908 WARN [SeleniumManager.lambda$runCommand$1] - Unable to discover proper msedgedriver version in offline mode
09:54:13.345 WARN [SeleniumManager.lambda$runCommand$1] - Unable to discover proper IEDriverServer version in offline mode
09:54:13.367 INFO [NodeOptions.report] - Adding Internet Explorer for {"browserName": "internet explorer", "platformName": "Windows 11"} 1 times
09:54:13.367 INFO [NodeOptions.report] - Adding Firefox for {"browserName": "firefox", "platformName": "Windows 11"} 12 times
09:54:13.367 INFO [NodeOptions.report] - Adding Edge for {"browserName": "MicrosoftEdge", "platformName": "Windows 11"} 12 times
09:54:13.367 INFO [NodeOptions.report] - Adding Chrome for {"browserName": "chrome", "platformName": "Windows 11"} 12 times
09:54:13.414 INFO [Node.<init>] - Binding additional locator mechanisms: relative
09:54:13.576 INFO [NodeServer$2.start] - Starting registration process for Node http://172.23.144.1:5555
09:54:13.581 INFO [NodeServer.execute] - Started Selenium node 4.29.0 (revision 18ae989): http://172.23.144.1:5555
09:54:13.587 INFO [NodeServer$2.lambda$start$2] - Sending registration event...
09:54:13.909 INFO [NodeServer.lambda$createHandlers$2] - Node has been added
```



```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

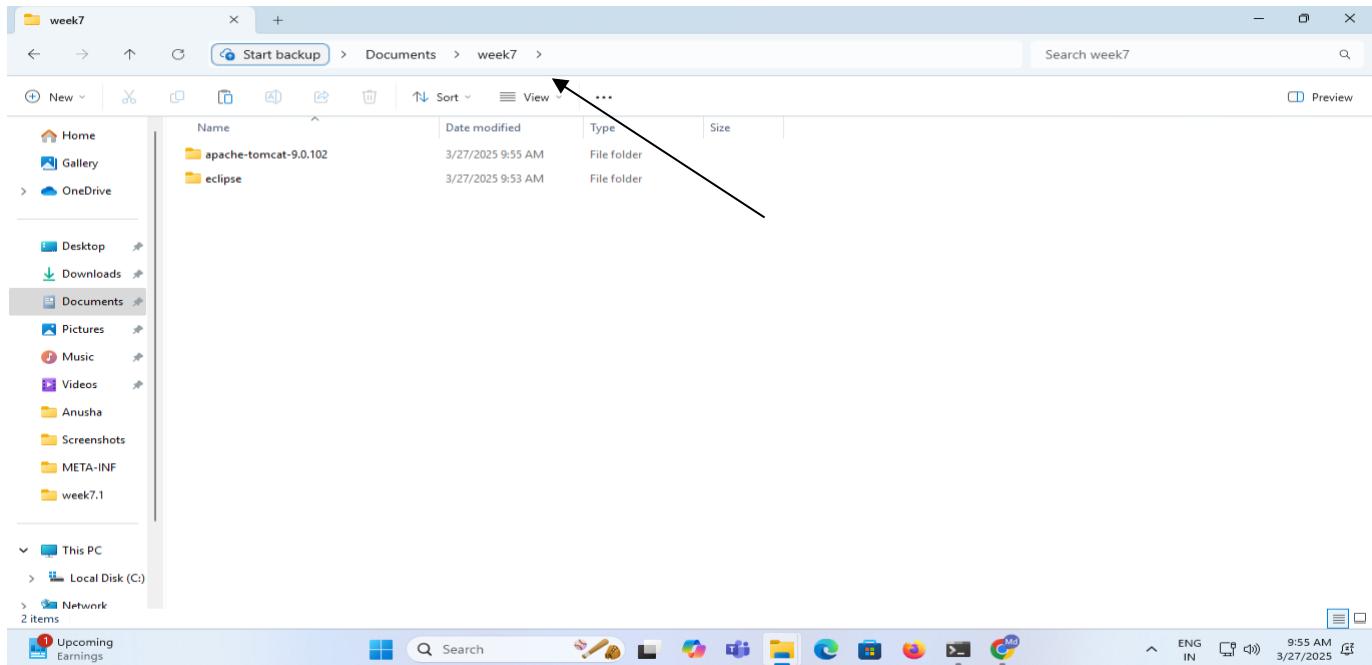
C:\Users\stud>cd downloads\selenium1

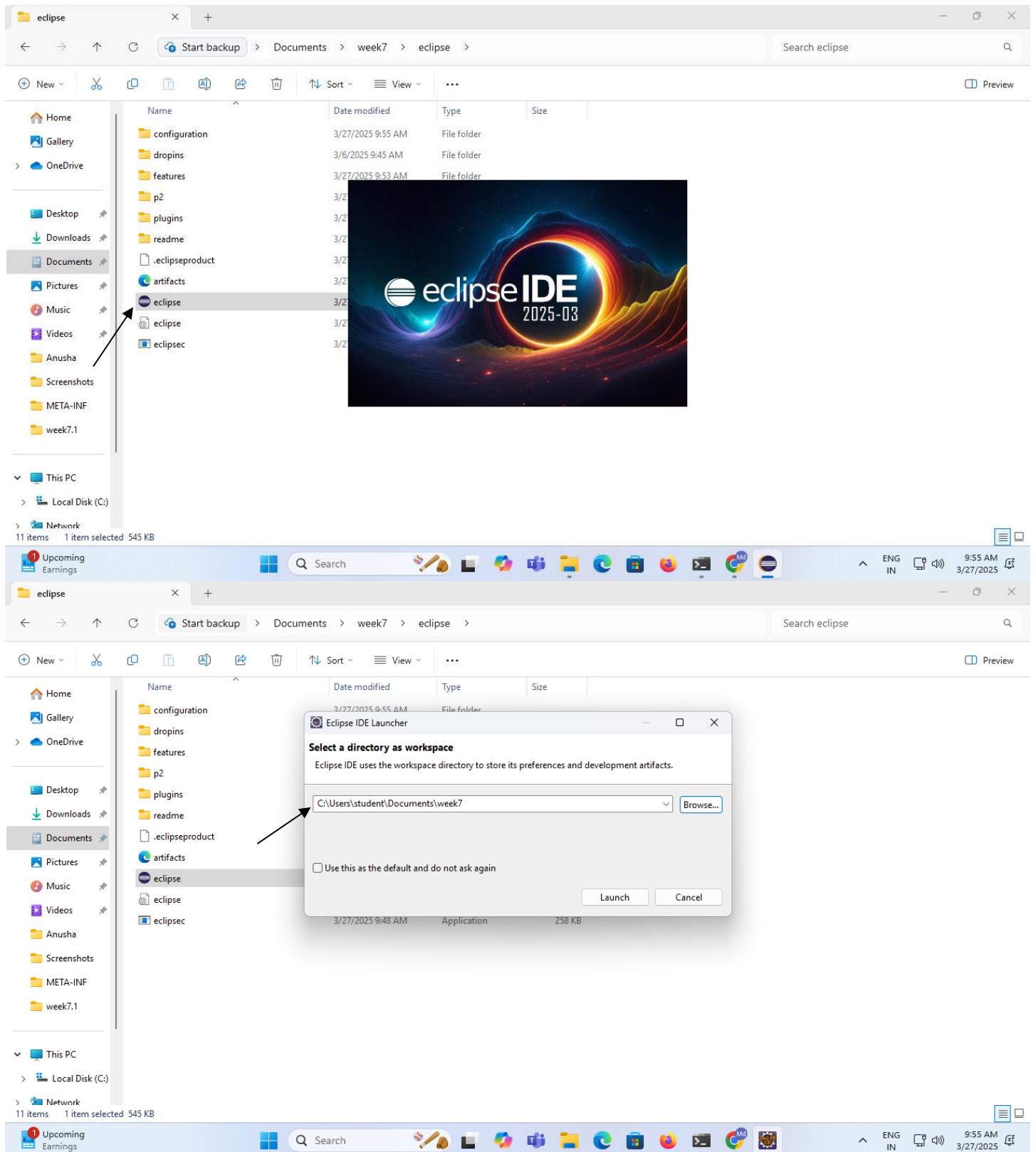
C:\Users\stud>java -jar C:\Users\student\Downloads\selenium1\selenium-server-4.29.0.jar hub
09:52:54.944 INFO [LoggingOptions.configureLogEncoding] - Using the system default encoding
09:52:54.960 INFO [OpenTelemetryTracer.createTracer] - Using OpenTelemetry for tracing
09:52:55.167 INFO [BoundZmqEventBus.<init>] - X PUB binding to [binding to tcp://*:4442, advertising as tcp://172.23.144.1:4442], XSUB binding to [binding to tcp://*:4443, advertising as tcp://172.23.144.1:4443]
09:52:55.220 INFO [UnboundZmqEventBus.<init>] - Connecting to tcp://172.23.144.1:4442 and tcp://172.23.144.1:4443
09:52:55.245 INFO [UnboundZmqEventBus.<init>] - Sockets created
09:52:56.248 INFO [UnboundZmqEventBus.<init>] - Event bus ready
09:52:56.838 INFO [Hub.execute] - Started Selenium Hub 4.29.0 (revision 18ae989): http://172.23.144.1:4444
09:54:13.625 INFO [Node.<init>] - Binding additional locator mechanisms: relative
09:54:13.904 INFO [GridModel.setAvailability] - Switching Node 0026ba90-db0e-46f6-8f58-329ecab899af (uri: http://172.23.144.1:5555) from DOWN to UP
09:54:13.904 INFO [LocalDistributor.add] - Added node 0026ba90-db0e-46f6-8f58-329ecab899af at http://172.23.144.1:5555.
Health check every 120s
```



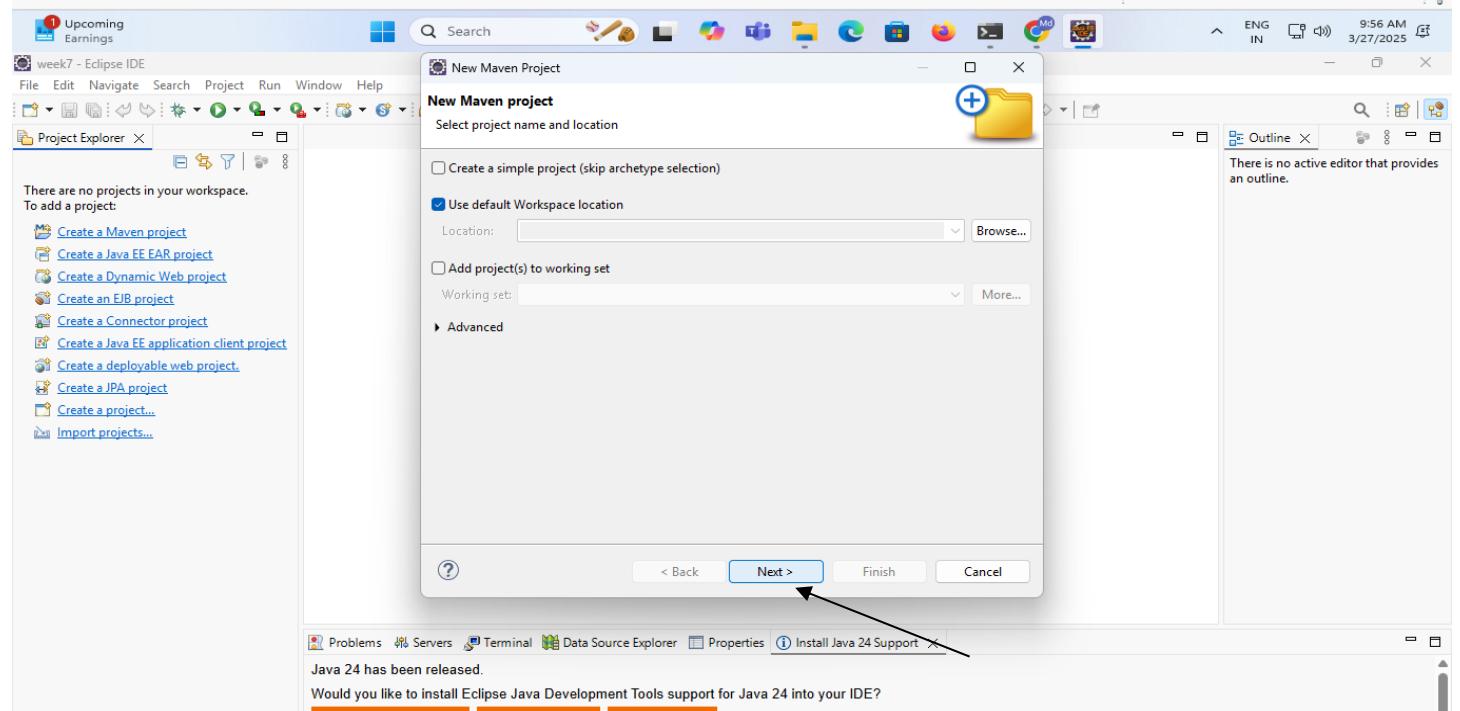
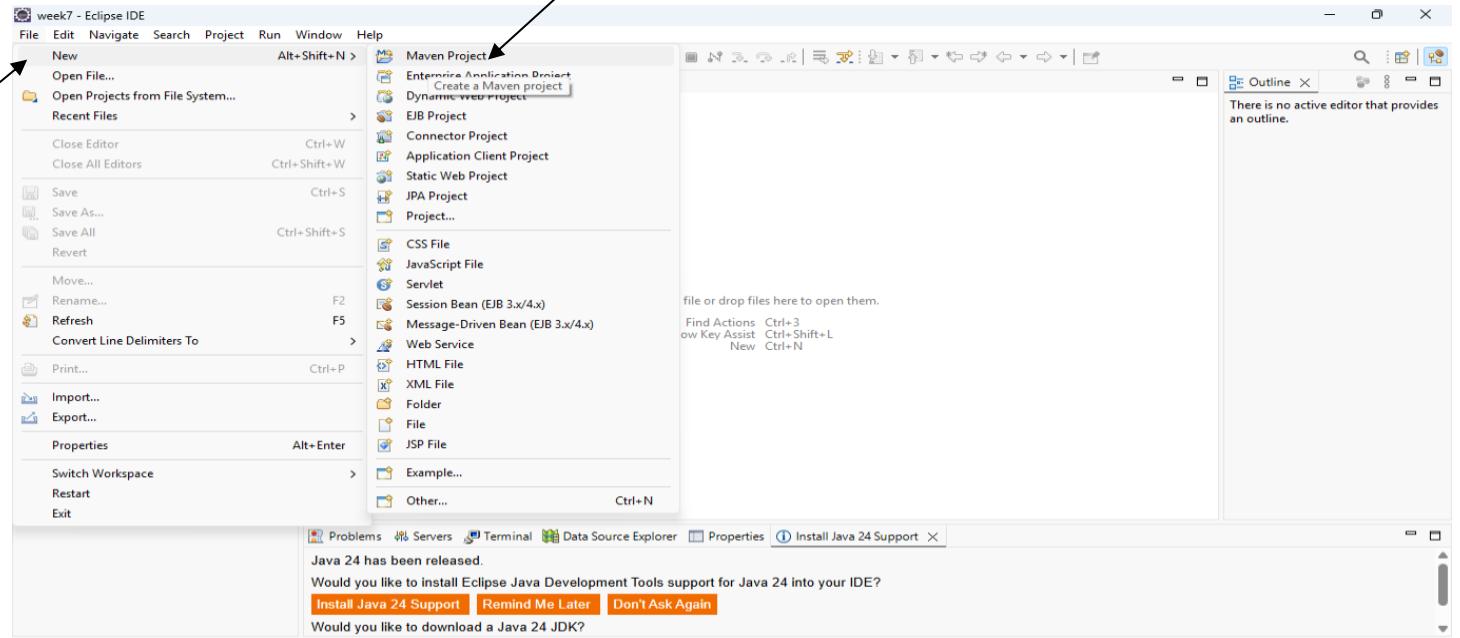
Step 6:-Open Eclipse Attach Apache Tomcat and TestNG

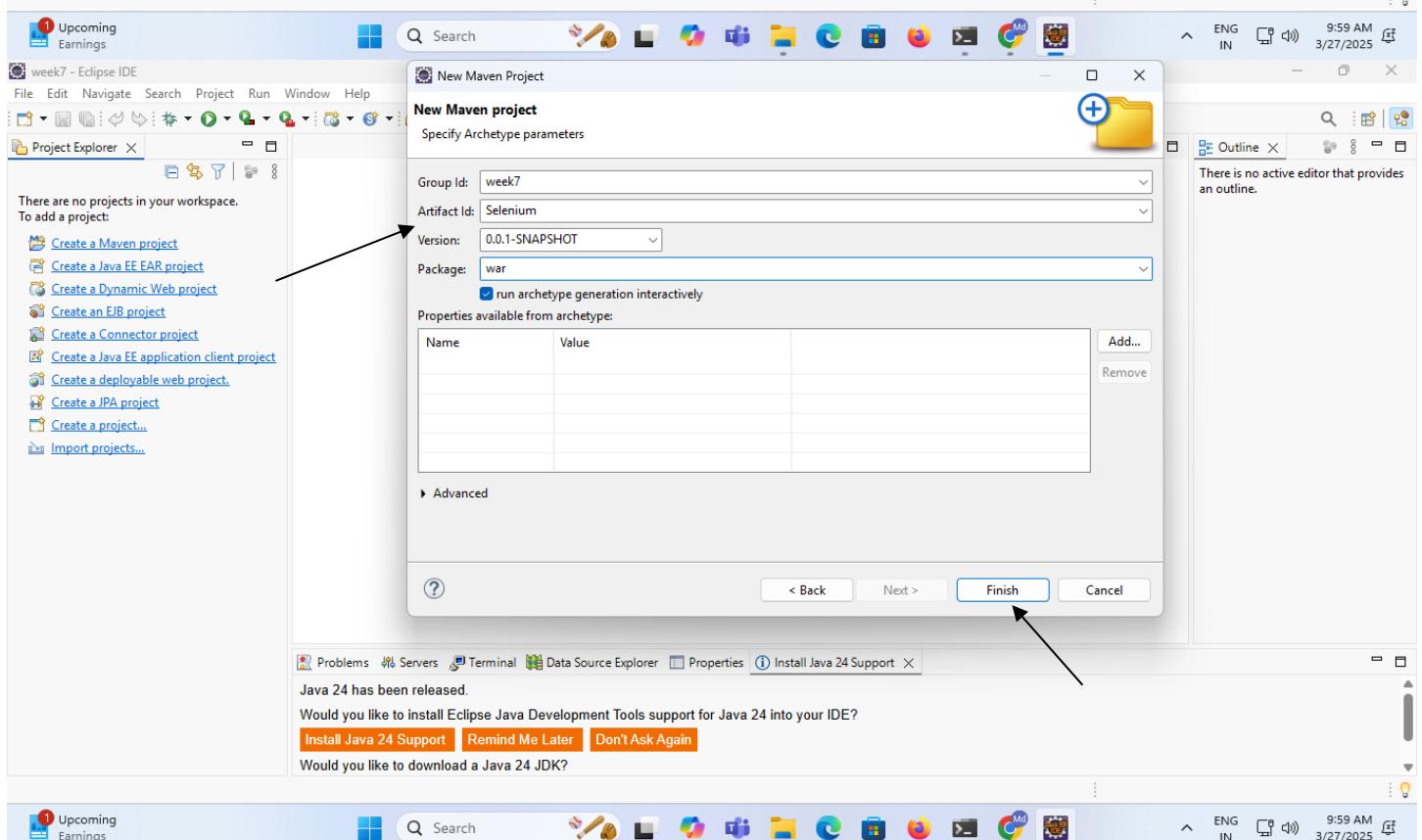
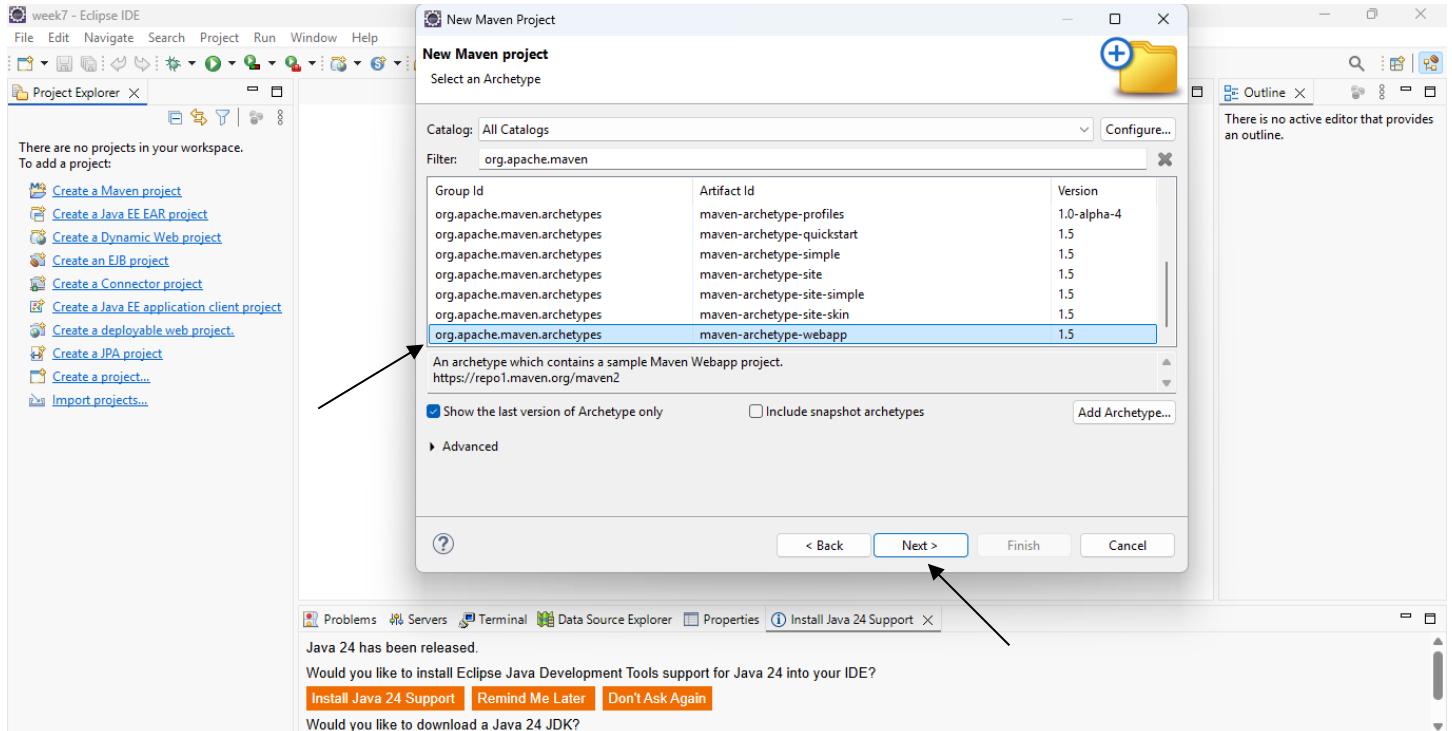
1. Store Apache and eclipse folders only single folder and lunch Eclipse same folder

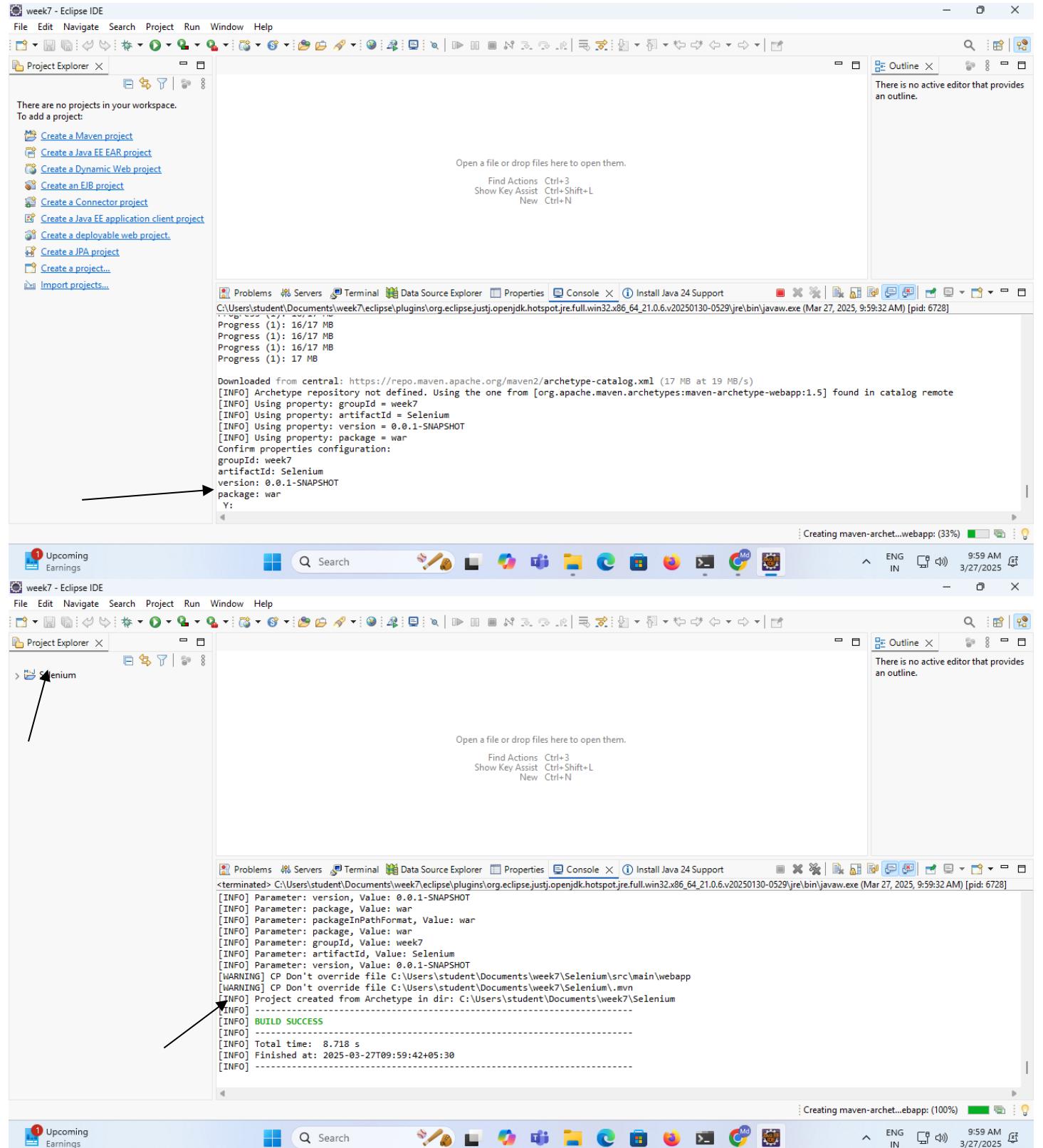




3.Create maven project goto File→New→Maven Project Select→Next→select org.apache.maven.archetypes select webapp→Enter Group Id:week7 ArtifactId:selenium Package:war→Finish. Enter Y you get Build Success in console.







3. Open Pom.xml copy paste below code this all I added dependies and plugins. Next Right click Project → Maven → Update project → ok.

(Note: If you not update every changes it will not save.)

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>DevOps</groupId>
  <artifactId>MyDevPipeline</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>MyDevPipeline Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <!--
  
```

<https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api> -->

```

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
      <scope>provided</scope>
    </dependency>
    <!--
  
```

<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-api> -->

```

    <dependency>
  
```

```
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>selenium-server</artifactId>
<version>3.141.59</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
<groupId>org.testng</groupId>
<artifactId>testng</artifactId>
<version>6.14.3</version>    <!-- Use the latest version
available(7.11.0) -->
<scope>test</scope>
</dependency>

<dependency>
<groupId>io.github.bonigarcia</groupId>
<artifactId>webdrivermanager</artifactId>
<version>5.3.0</version>
</dependency>

</dependencies>
<build>
    <finalName>MyDevPipeline</finalName>
    <pluginManagement>
        <plugins>
            <plugin>

<groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-clean-plugin</artifactId>
            <version>3.4.0</version>
        </plugin>
        <plugin>
```

```
<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-resources-
plugin</artifactId>
        <version>3.3.1</version>
    </plugin>
    <plugin>

<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-
plugin</artifactId>
        <version>3.13.0</version>
        <configuration>
            <source>17.0.12</source>
            <target>17.0.12</target>
        </configuration>
    </plugin>
    <plugin>

<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-
plugin</artifactId>
        <version>3.3.0</version>
    </plugin>
    <plugin>

<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
        <version>3.4.0</version>
    </plugin>
    <plugin>
```

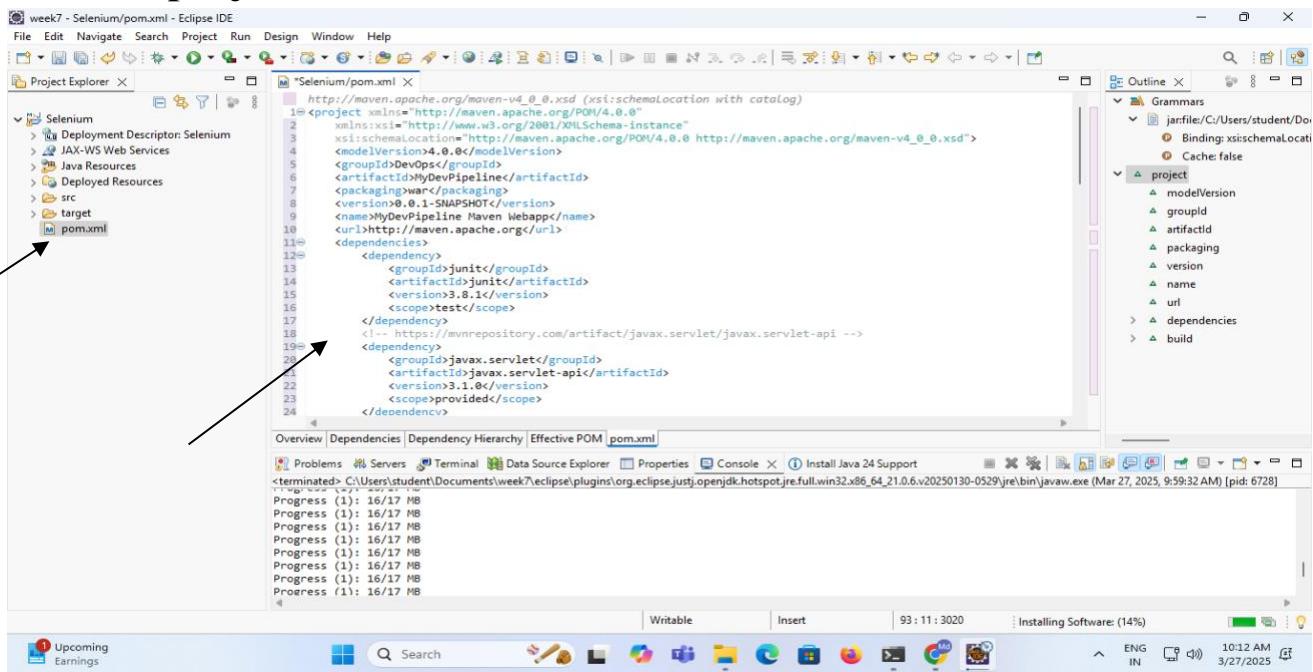
```

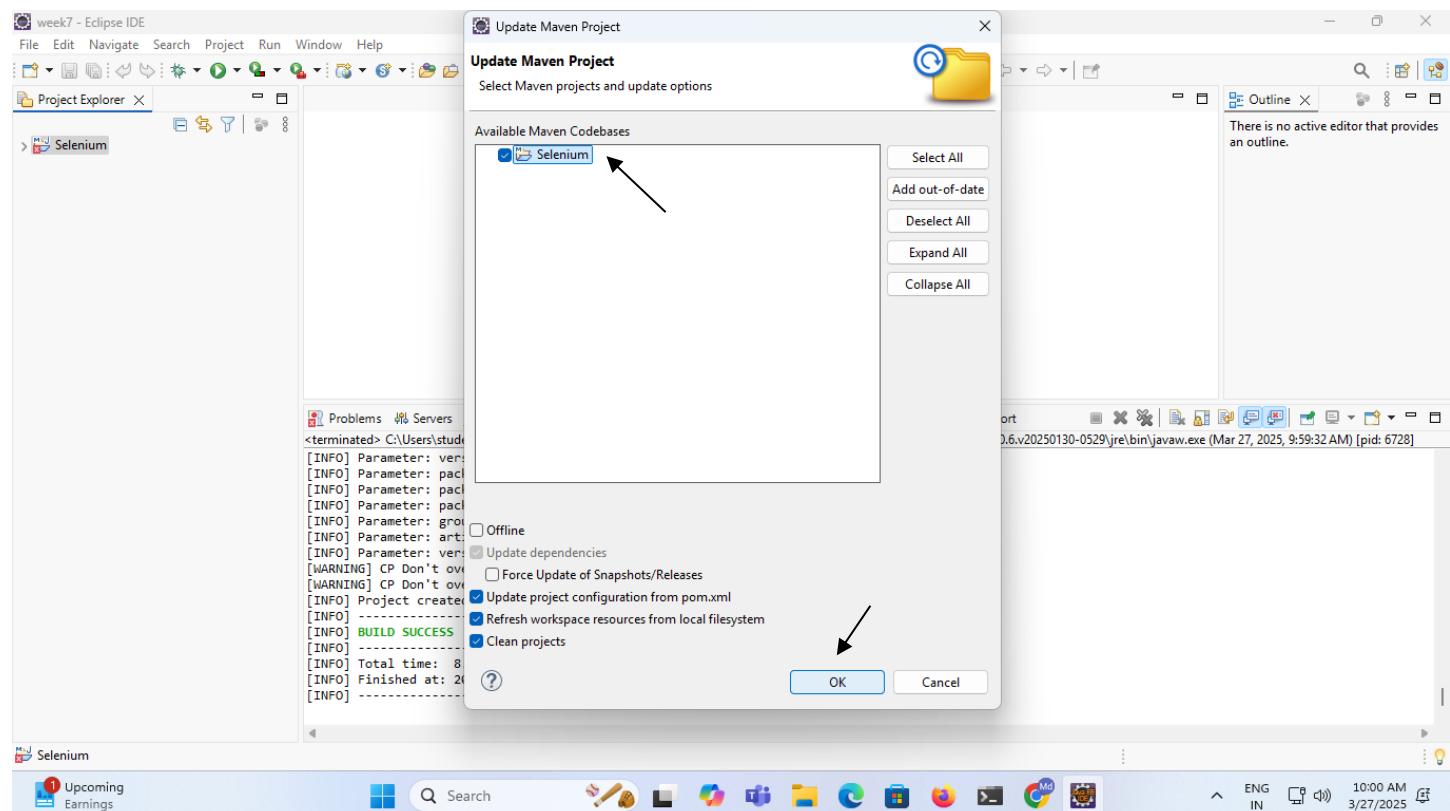
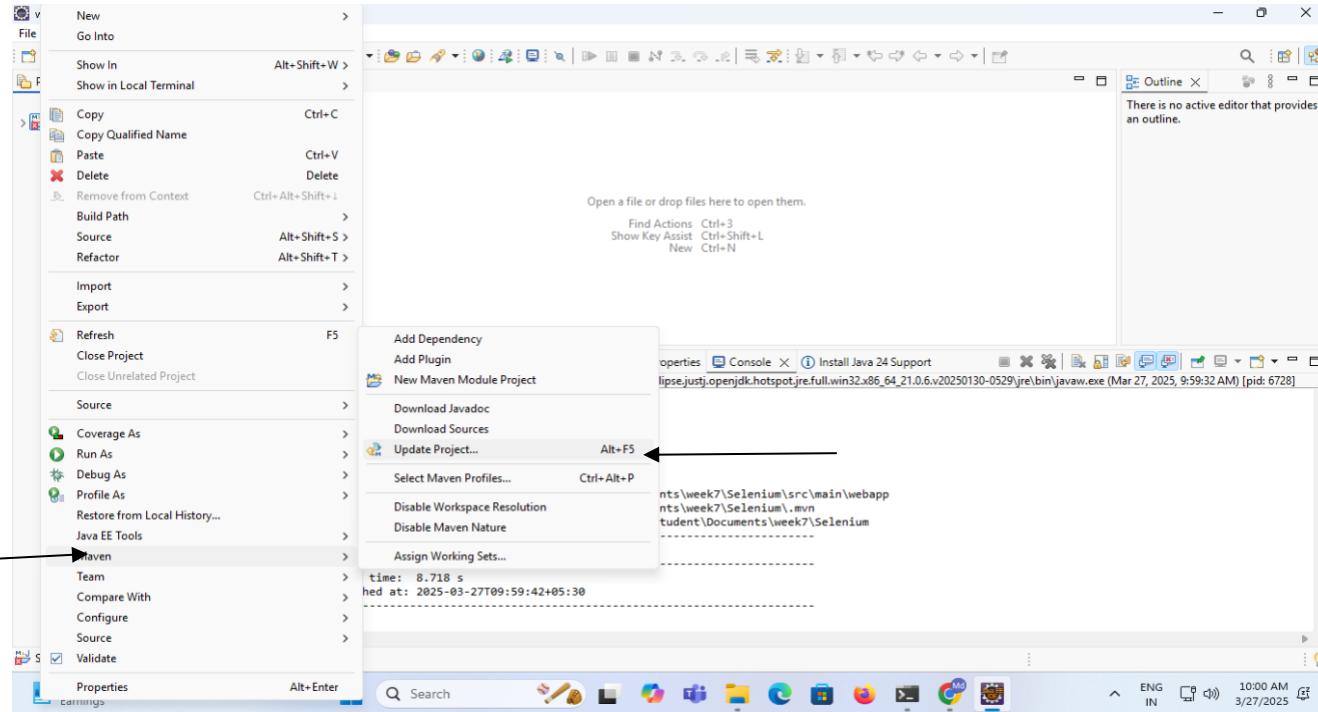
<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-install-plugin</artifactId>
    <version>3.1.2</version>
</plugin>
<plugin>

<groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-deploy-plugin</artifactId>
    <version>3.1.2</version>
</plugin>
</plugins>
</pluginManagement>

</build>
</project>

```

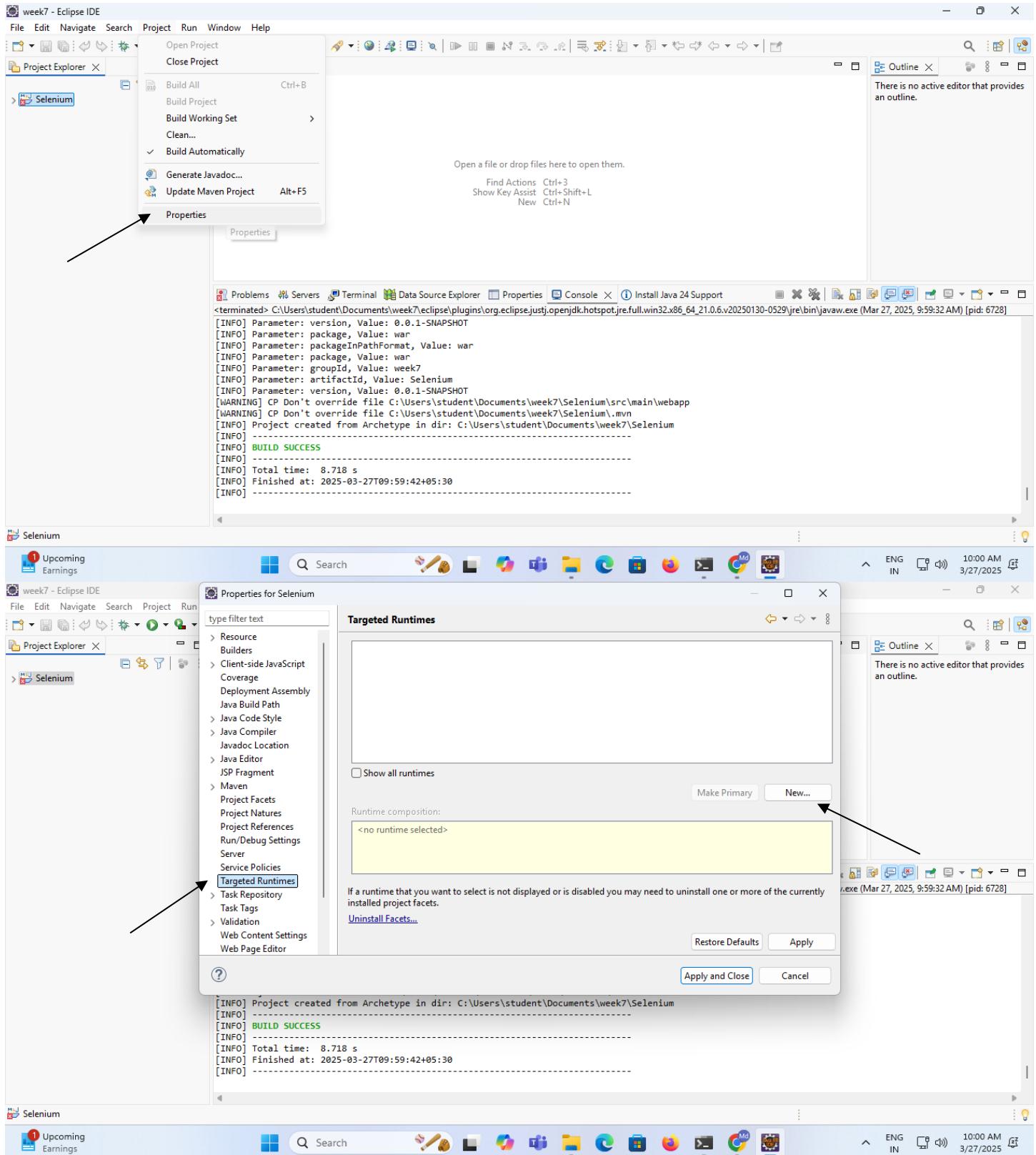


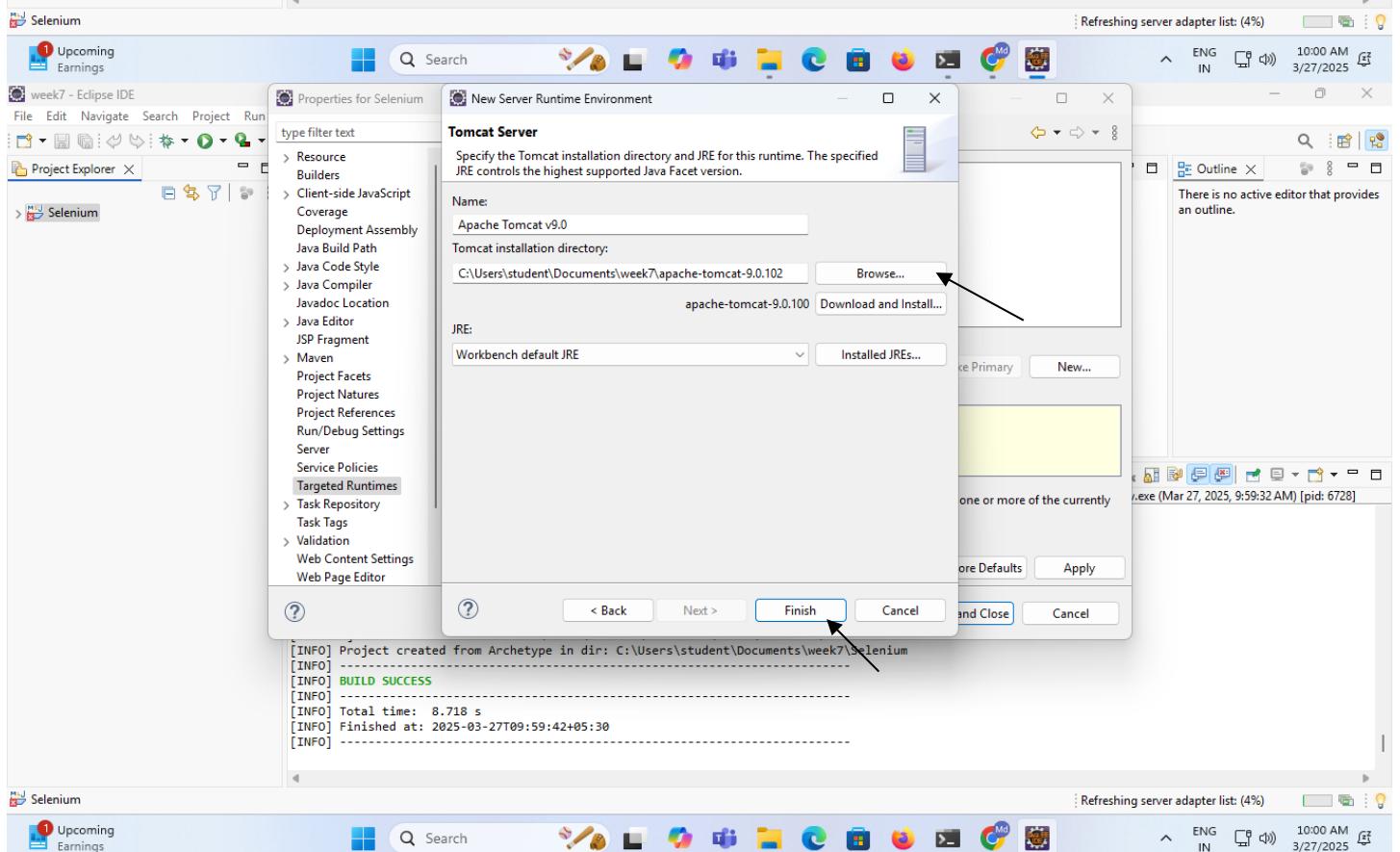
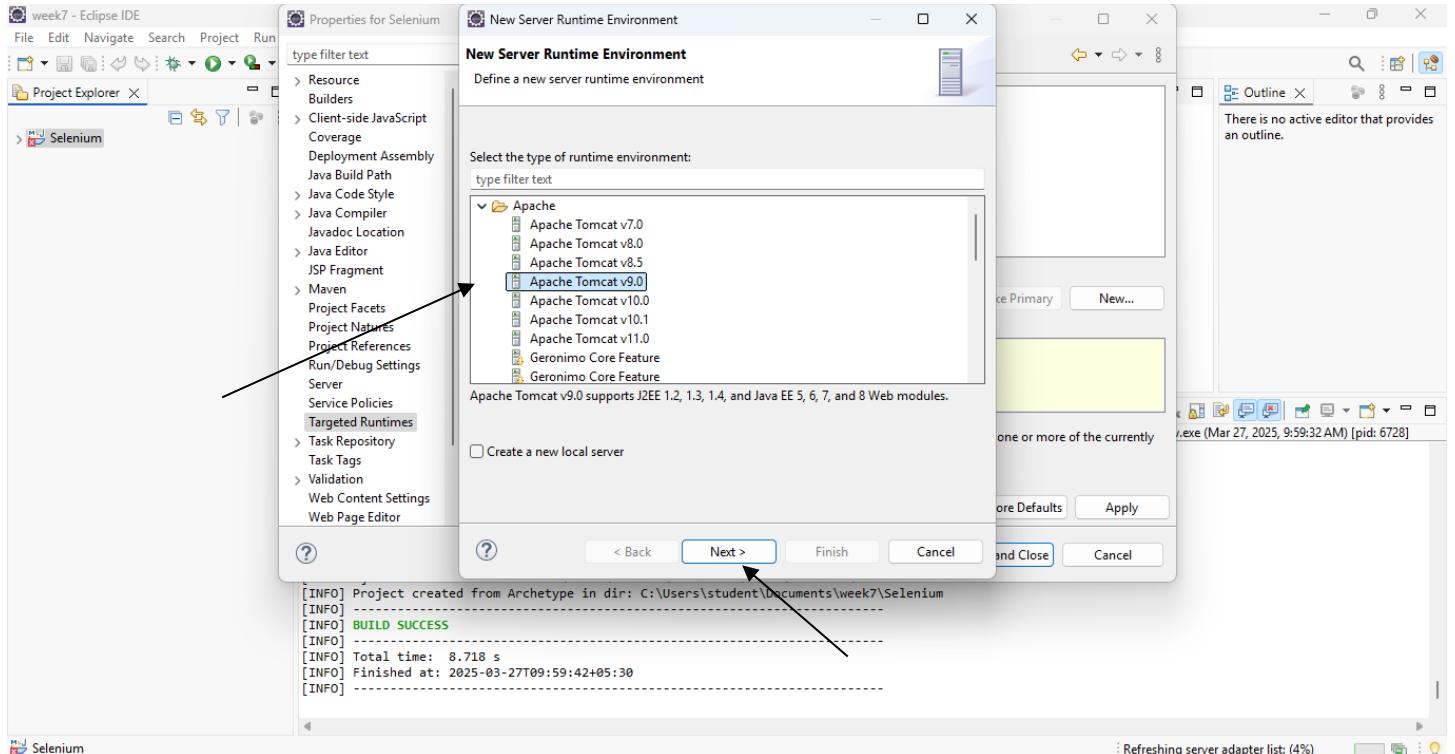


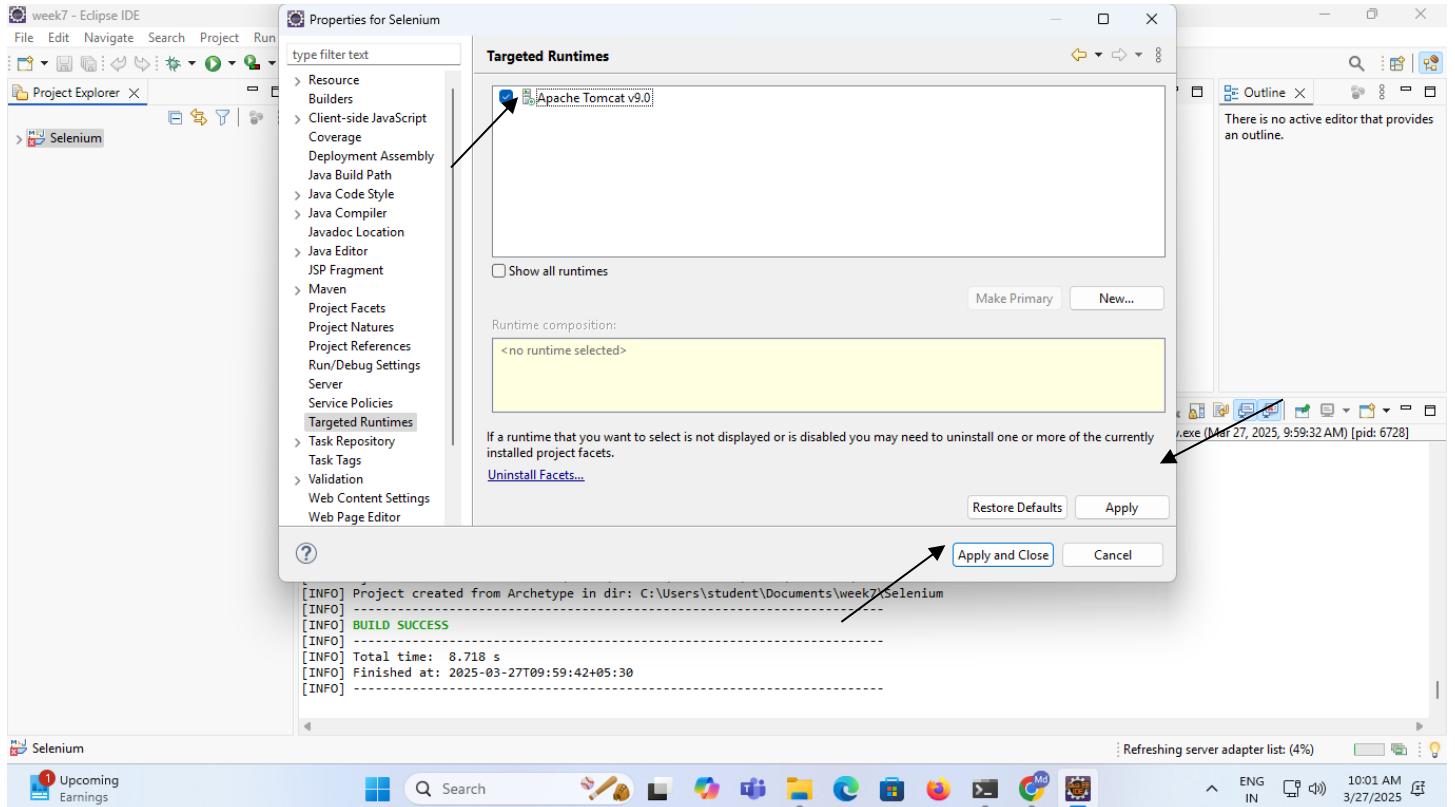
4.Adding Apache tomcat server

Go to Project → Properties → Maven → Targeted Runtimes → new → select apache tomcat 9 → next → browse apache folder address → finish → click check box apache 9 and apply → apply and close.

(Note:-because running src→index.js file running purpose if you not adding it will show errors.)

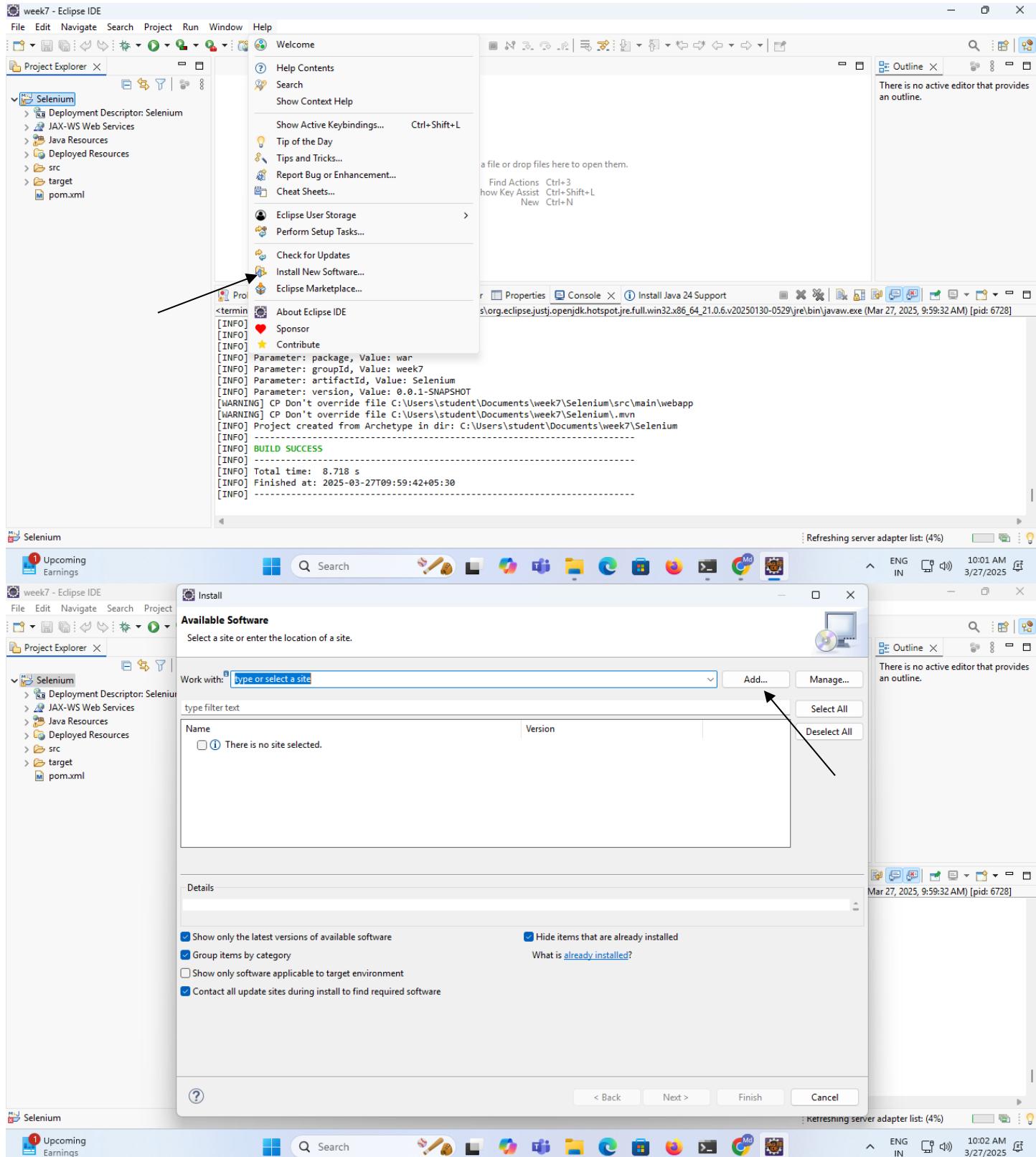


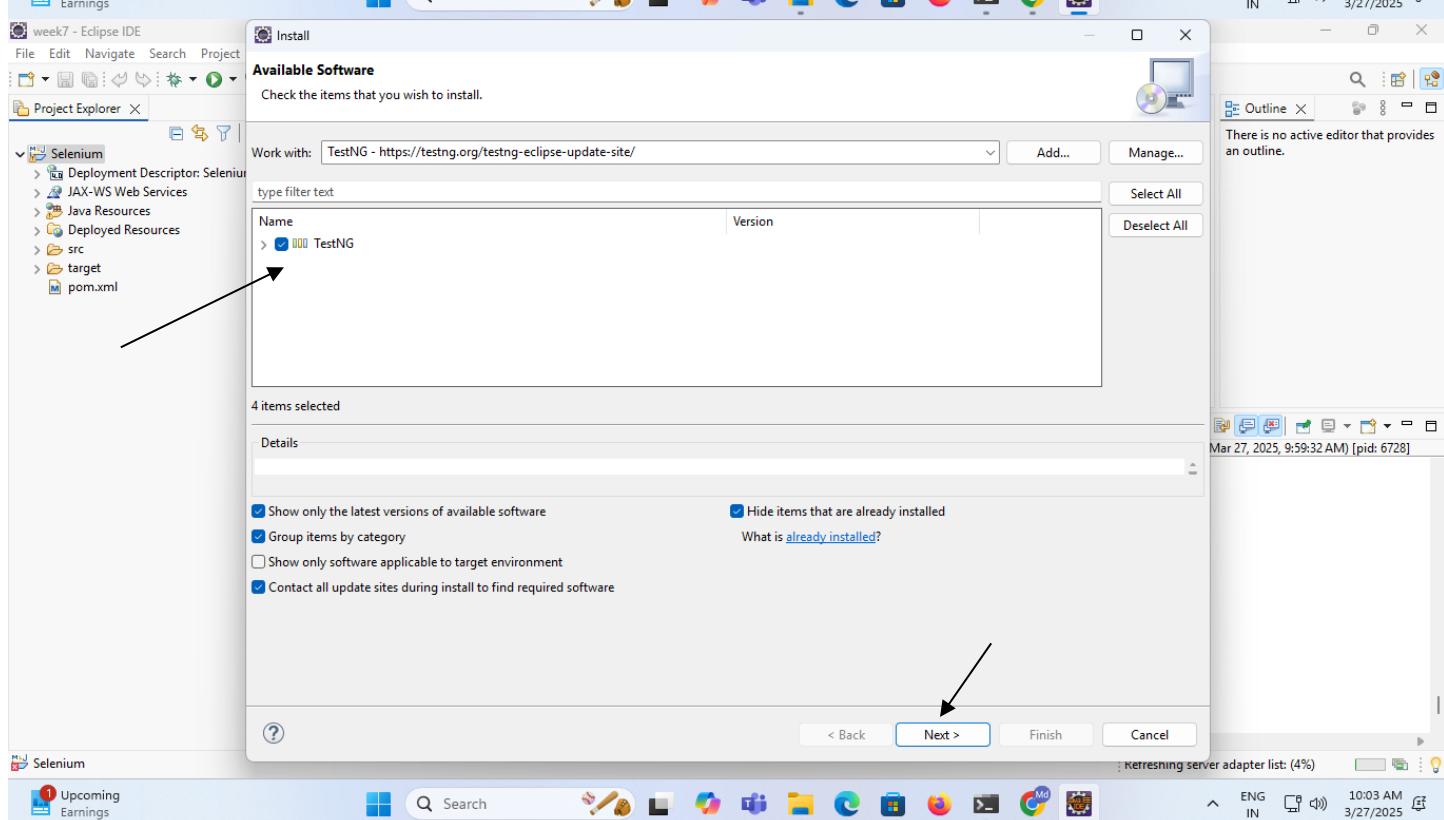
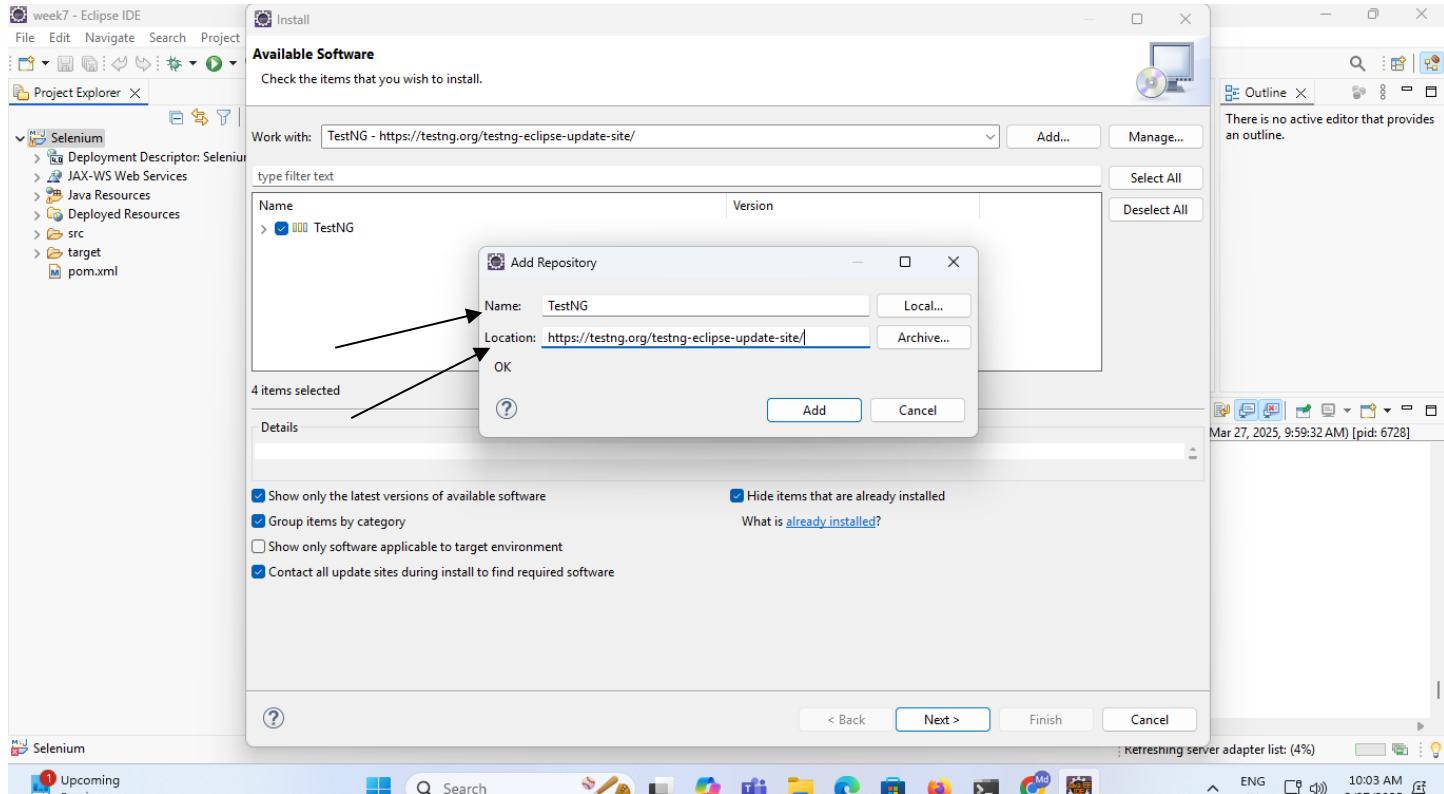


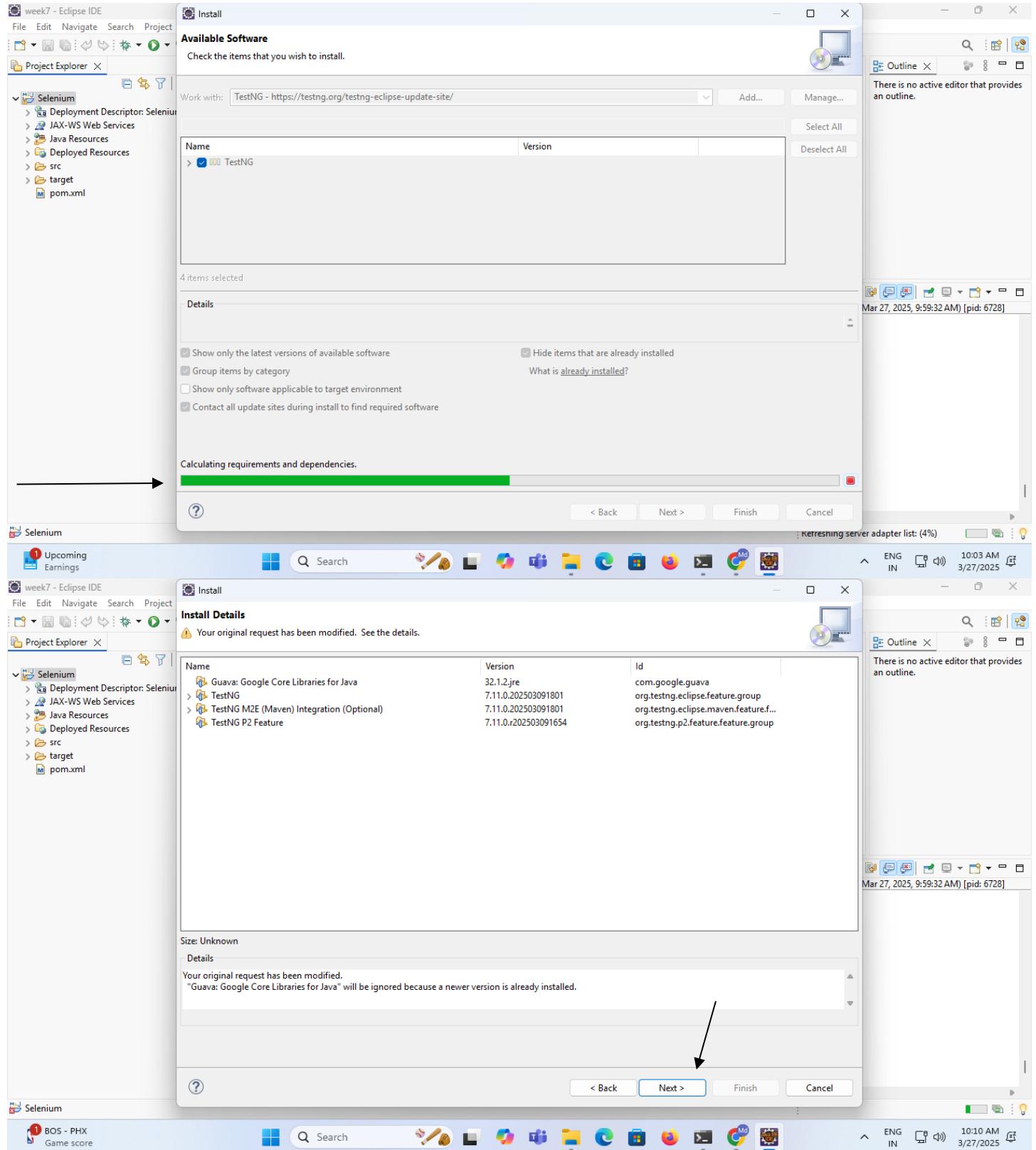


5. Adding TestNG in Eclipse.

Go to help → Install New Software → Add → Enter Name:TestNG, Location:<https://testng.org/testing-eclipse-update-site/> → Add. Select TestNG check box → Next → Next → I accept → Finish Trust Selected → Trust Selected.







Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses: Apache License

License text:

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

- Definitions.
- "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
- "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
- "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise; or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
- "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
- "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
- "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

I accept the terms of the license agreement
 I do not accept the terms of the license agreement

Trust Authorities

Do you trust content originating from the following authorities? Installing content involves performing actions that alter the installation's configuration and may potentially be used for malicious purposes.

Authority / Update Site	Units	Secured
https://download.oracle.com	1	✓
https://testng.org	16	✓

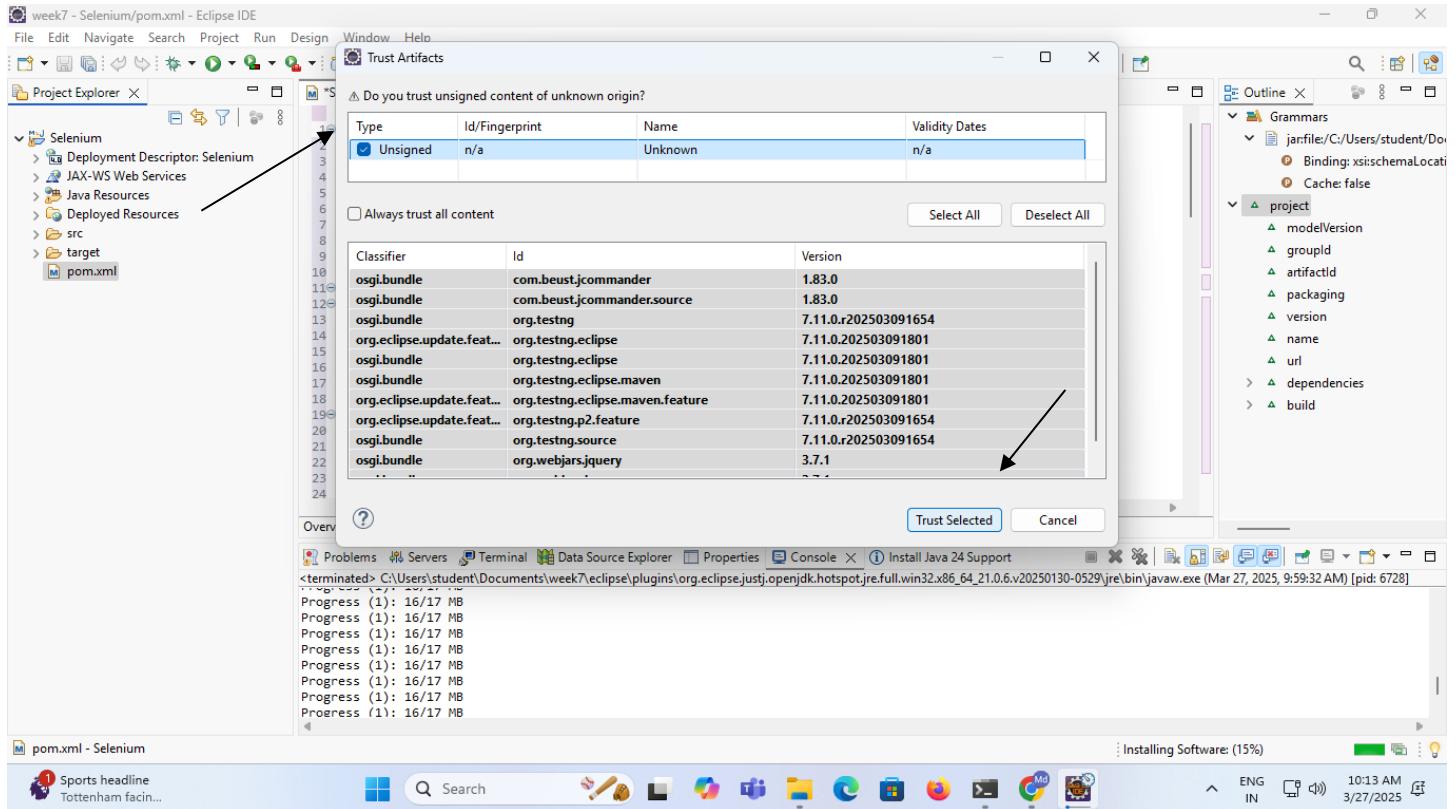
Remember selected authorities Always trust all authorities Select All Deselect All Expand All Collapse All

download.oracle.com; Oracle Corporation
DigiCert TLS RSA SHA256 2020 CA; DigiCert Inc

Details... **Export...**

Id	Version
javaeighth	1.1.6.v20160919-1400
com.beust.jcommander	1.83.0
com.beust.jcommander.source	1.83.0
org.testng	7.11.0.r202503091654
org.testng.eclipse	7.11.0.r202503091801
org.testng.eclipse.feature.group	7.11.0.r202503091801
org.testng.eclipse.feature.jar	7.11.0.r202503091801
org.testng.eclipse.maven	7.11.0.r202503091801
org.testng.eclipse.maven.feature.feature.g...	7.11.0.r202503091801
org.testng.eclipse.maven.feature.feature.jar	7.11.0.r202503091801

Trust Selected Cancel



6.Create Folder inside src →test inside java.java folder select build path →current folder.Right click that folder→new→others→class→next→give package name and class name→ok

Java file created copy paste below code.change hub address must and should.

Same page right click→run→TestNG →Finally its run test passes.

SeleniumGridDemo.java

```
package com.testautomation.grid;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.testng.annotations.Test;
import java.net.URL;
```

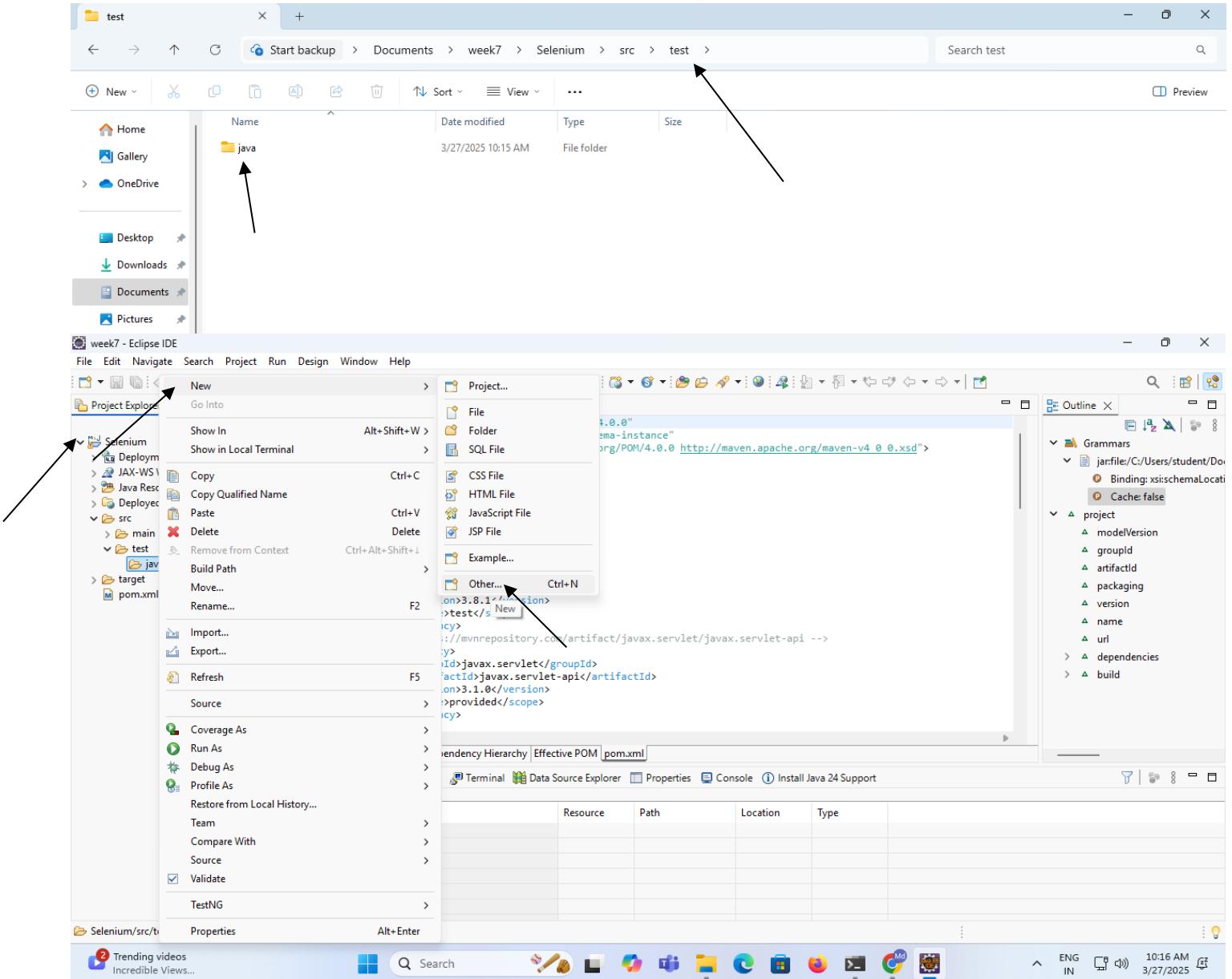
```
public class SeleniumGridDemo {
```

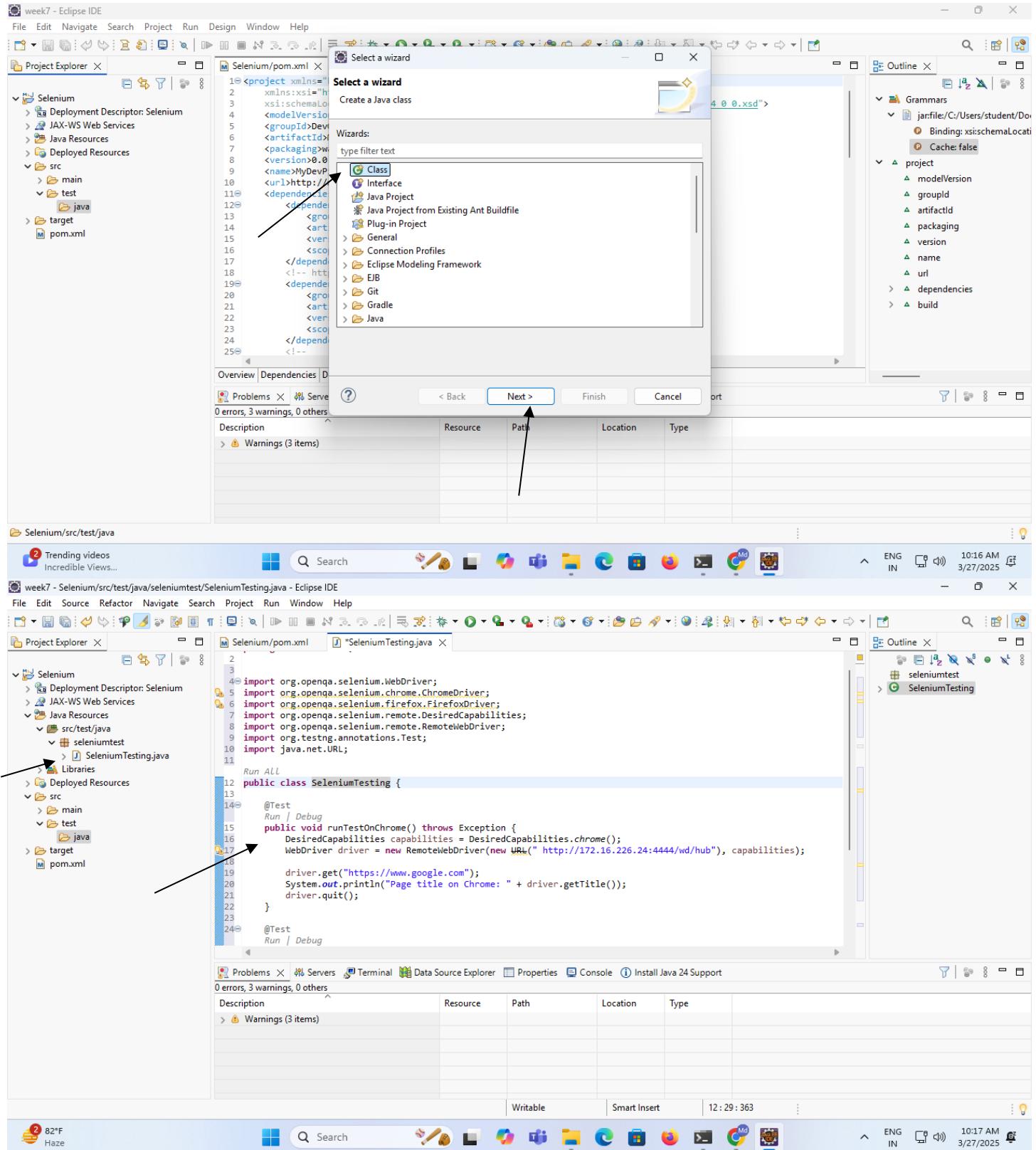
@Test

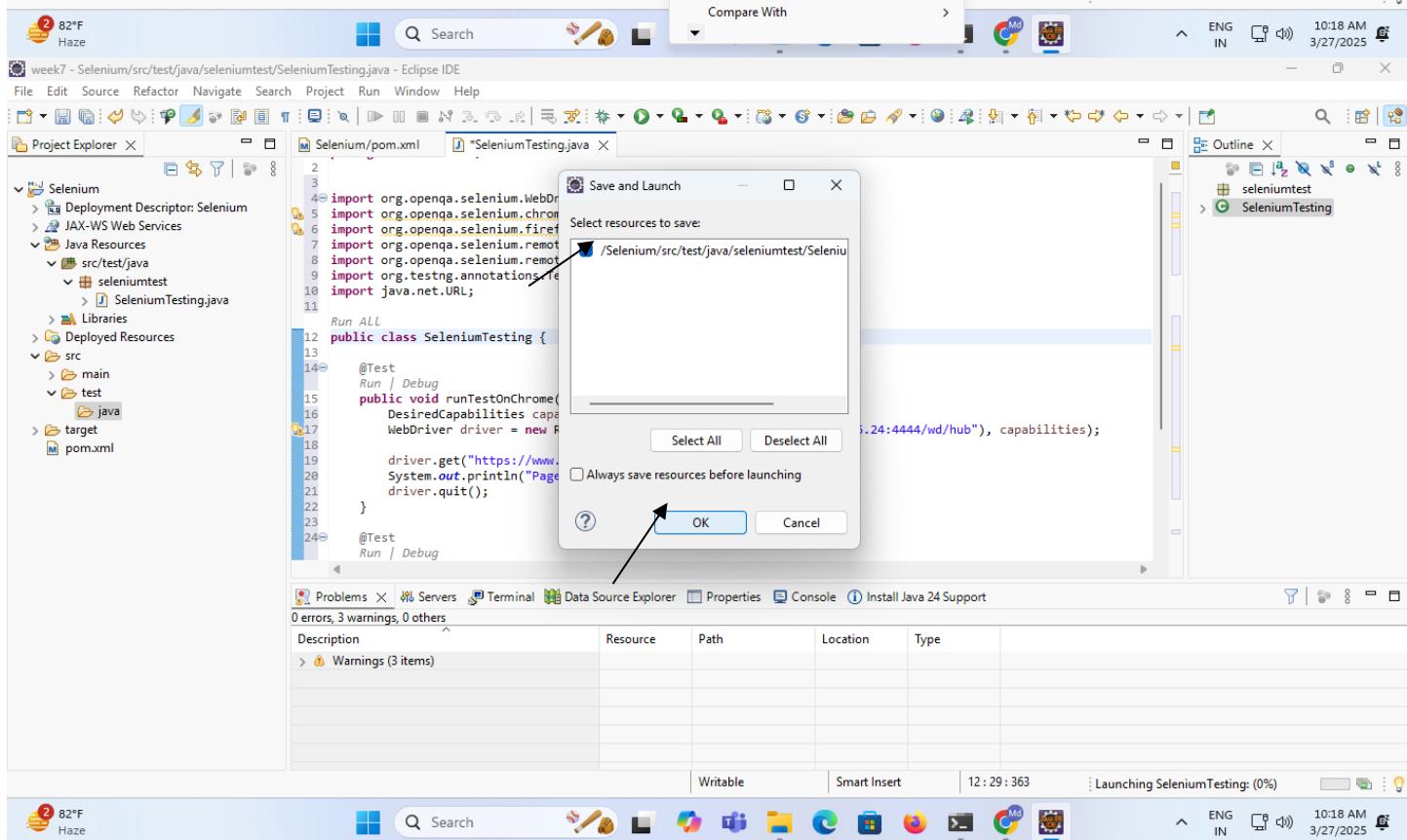
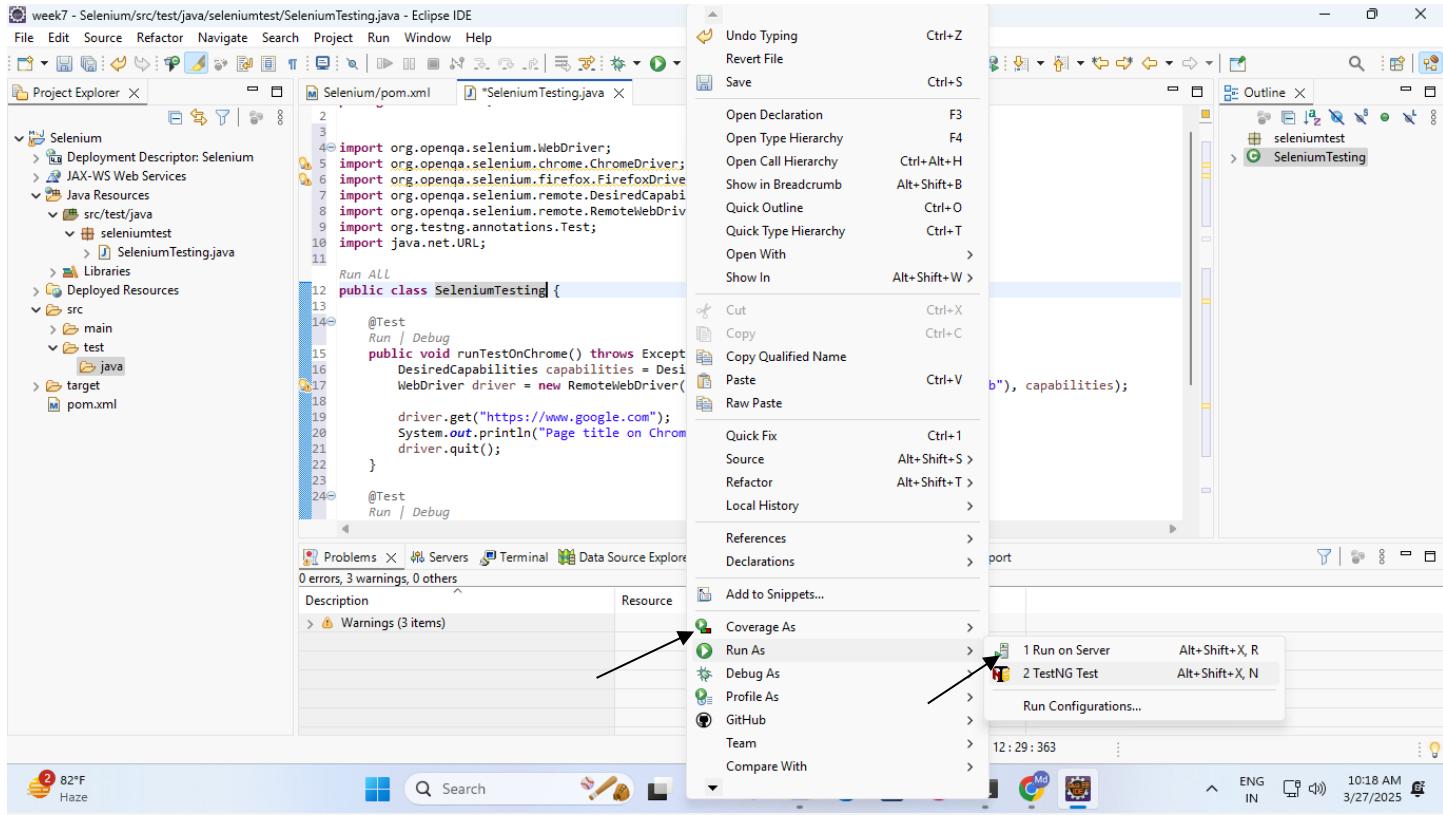
```
public void runTestOnChrome() throws Exception {  
    DesiredCapabilities capabilities = DesiredCapabilities.chrome();  
    WebDriver driver = new RemoteWebDriver(new URL("http://172.16.226.24:4444/wd/hub"), capabilities);  
  
    driver.get("https://www.google.com");  
    System.out.println("Page title on Chrome: " + driver.getTitle());  
    driver.quit();  
}
```

@Test

```
public void runTestOnFirefox() throws Exception {  
    DesiredCapabilities capabilities = DesiredCapabilities.firefox();  
    WebDriver driver = new RemoteWebDriver(new URL("http://172.16.226.24:4444/wd/hub"), capabilities);  
  
    driver.get("https://www.google.com");  
    System.out.println("Page title on Firefox: " + driver.getTitle());  
    driver.quit();  
}  
}
```







Output:-

SeleniumTesting.java

```

1 package seleniumtest;
2
3
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 import org.openqa.selenium.firefox.FirefoxDriver;
7 import org.openqa.selenium.remote.DesiredCapabilities;
8 import org.openqa.selenium.remote.RemoteWebDriver;
9 import org.testng.annotations.Test;
10 import java.net.URL;
11
12 public class SeleniumTesting {
13
14     @Test
15     public void runTestOnChrome() throws Exception {
16         DesiredCapabilities capabilities = DesiredCapabilities.chrome();
17
18         WebDriver driver = new ChromeDriver(capabilities);
19
20         URL url = new URL("http://www.google.com");
21         driver.get(url);
22
23         String title = driver.getTitle();
24         System.out.println(title);
25
26         driver.quit();
27     }
28 }

```

Console Output:

```

<terminated> SeleniumTesting [TestNG] C:\Users\student\Documents\week7\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (Mar 27, 2025, 10:31:01 AM)
PASSED: seleniumtest.SeleniumTesting.runTestOnFirefox

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====


```

SeleniumTesting.java

```

1 package seleniumtest;
2
3
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 import org.openqa.selenium.firefox.FirefoxDriver;
7 import org.openqa.selenium.remote.DesiredCapabilities;
8 import org.testng.annotations.Test;
9 import java.net.URL;
10
11 public class SeleniumTesting {
12
13     @Test
14     public void runTestOnChrome() throws Exception {
15         DesiredCapabilities capabilities = DesiredCapabilities.chrome();
16
17         WebDriver driver = new ChromeDriver(capabilities);
18
19         URL url = new URL("http://www.google.com");
20         driver.get(url);
21
22         String title = driver.getTitle();
23         System.out.println(title);
24
25         driver.quit();
26     }
27 }

```

Console Output:

```

<terminated> SeleniumTesting [TestNG] C:\Users\student\Documents\week7\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (Mar 27, 2025, 10:31:01 AM)
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Mar 27, 2025 10:30:52 AM org.openqa.selenium.remote.DesiredCapabilities chrome
INFO: Using `new ChromeOptions()` is preferred to `DesiredCapabilities.chrome()`
Mar 27, 2025 10:30:56 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Page title on Chrome: Google
Mar 27, 2025 10:30:56 AM org.openqa.selenium.remote.DesiredCapabilities firefox
INFO: Using `new FirefoxOptions()` is preferred to `DesiredCapabilities.firefox()`
Mar 27, 2025 10:31:00 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Page title on Firefox: Google
PASSED: seleniumtest.SeleniumTesting.runTestOnChrome
PASSED: seleniumtest.SeleniumTesting.runTestOnFirefox

=====
Default test
Tests run: 2, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
=====


```

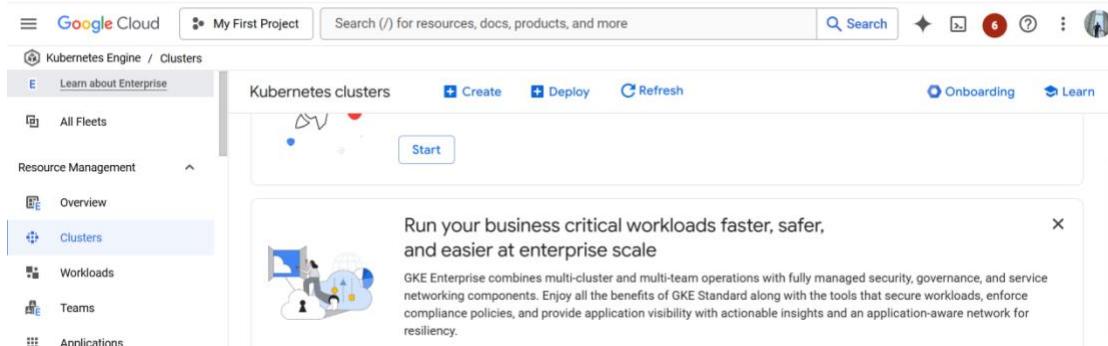
Experiment-9

Aim:-Create deployment resource using Kubernetes.

- Create the google cloud console free account
- It is a two step process
- It is deducting the 2 rupees from your account and it will give the 330\$ free credit points.

NOTE: Don't active the full account

- Once the account is created u can login to google cloud console



- NOW CREATE THE KUBERNETES CLUSTER

Open the cloud shell



To see the cluster list run the below command

Gcloud container clusters list (no clusters are there)

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ gcloud container clusters list
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

You create the cluster with below command

```
gcloud container clusters create my-cluster --zone us-central1-a
```

Cluster creation is taking 5 to 10 mints time

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ gcloud container clusters create my-cluster --zone us-central1-a
Note: The Kubelet readonly port (10255) is now deprecated. Please update your workloads to use the recommended alternatives. See https://cloud.google.com/kubernetes-engine/docs/how-to/disable-kubelet-readonly-port for ways to check usage and for migration instructions.
Note: Your Pod address range ('--cluster-ipv4-cidr') can accommodate at most 1008 node(s),
Creating cluster my-cluster in us-central1-a...working.[]
```

Once the cluster is created u can see the below message automatically

```
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload/_gcloud/us-central1-a/my-cluster?project=hidden-solstice-454006-n0
kubeconfig entry generated for my-cluster.
NAME: my-cluster
LOCATION: us-central1-a
MASTER_VERSION: 1.31.6-gke.1020000
MASTER_IP: 34.66.171.95
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.31.6-gke.1020000
NUM_NODES: 3
STATUS: RUNNING
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

Now u go and check kubernetes engine--->cluster , you can see the my-cluster is running

Status	Name	Location	Tier	Number of nodes	Total vCPUs	Total memory
<input checked="" type="checkbox"/>	my-cluster	us-central1-a	Standard	3	6	12 GB

Run the below command

gcloud container clusters get-credentials my-cluster --zone us-central1-a

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ gcloud container clusters get-credentials my-cluster --zone us-central1-a
Fetching cluster endpoint and auth data.
kubeconfig entry generated for my-cluster.
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

To see the list of nodes

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get nodes
NAME           STATUS   ROLES      AGE    VERSION
gke-my-cluster-default-pool-020e8447-03bv   Ready    <none>    8m43s   v1.31.6-gke.1020000
gke-my-cluster-default-pool-020e8447-1510   Ready    <none>    8m43s   v1.31.6-gke.1020000
gke-my-cluster-default-pool-020e8447-rv2r   Ready    <none>    8m43s   v1.31.6-gke.1020000
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

Create the pods

Kubectl run --image tomcat webserver

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl run --image tomcat webserver
pod/webserver created
```

To see the pods list

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
webserver   1/1    Running   0          20m
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

To get the list of pods along with ip address and which node the pod is running

Kubectl get pods -o wide

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get pods -o wide
NAME     READY   STATUS    RESTARTS   AGE   IP           NODE   NOMINATED NODE   READINESS GATES
webserver   1/1    Running   0          22m   10.40.0.4   gke-my-cluster-default-pool-020e8447-1510   <none>        <none>
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

Actually u can create the pod using definition file

Create pd-df1.yaml

Vim pd-df1.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: jenkins-pod
spec:
  containers:
    - name: myjenkins
      image: jenkins/jenkins
      ports:
        - containerPort: 8080
          hostPort: 8080
```

for accessing the application u need to open the port

How to open the port

gcloud compute firewall-rules create rule2 --allow tcp:8080

```
sarvesh_ambala@cloudshell:~/sample1 [hidden-solstice-454006-n0]$ gcloud compute firewall-rules create rule2 --allow tcp:8080
Creating firewall...working..Created [https://www.googleapis.com/compute/v1/projects/hidden-solstice-454006-n0/global/firewalls/rule2].
Creating firewall...done.
NAME: rule2
NETWORK: default
DIRECTION: INGRESS
PRIORITY: 1000
ALLOW: tcp:8080
DENY:
DISABLED: False
```

Kubectl create -f pd-df1.yaml

Kubectl get pods -o wide

```
sarvesh_ambala@cloudshell:~/sample1 [hidden-solstice-454006-n0]$ kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP           NODE   NOMINATED NODE   READINESS GATES
jenkins-pod   0/1     ImagePullBackOff 0  3h25m  10.40.0.5  gke-my-cluster-default-pool-020e8447-1510  <none>  <none>
jenkins-pod1  0/1     ImagePullBackOff 0  3h8m   10.40.0.6  gke-my-cluster-default-pool-020e8447-1510  <none>  <none>
jk-pod        1/1     Running   0  4m33s  10.40.1.5  gke-my-cluster-default-pool-020e8447-rv2r  <none>  <none>
webserver     1/1     Running   0  4h5m   10.40.0.4  gke-my-cluster-default-pool-020e8447-1510  <none>  <none>
```

Kubectl get nodes -o wide

```
sarvesh ambala@cloudshell:~/samples [hidden-solstice-454006-n0]$ kubectl get nodes -o wide
NAME          STATUS   ROLES      AGE       VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION CONTAINER-RUNTIME
gke-my-cluster-default-pool-020e8447-03bv Ready    <none>    4h19m    v1.31.6-gke.1020000 10.128.0.15  35.225.52.28 Container-Optimized OS from Google
6.6.72+        containerd://1.7.24
gke-my-cluster-default-pool-020e8447-1510 Ready    <none>    4h19m    v1.31.6-gke.1020000 10.128.0.13  34.60.92.233 Container-Optimized OS from Google
6.6.72+        containerd://1.7.24
gke-my-cluster-default-pool-020e8447-rv2r Ready    <none>    4h19m    v1.31.6-gke.1020000 10.128.0.14  34.56.2.233 Container-Optimized OS from Google
6.6.72+        containerd://1.7.24
sarvesh ambala@cloudshell:~/samples [hidden-solstice-454006-n0]$
```

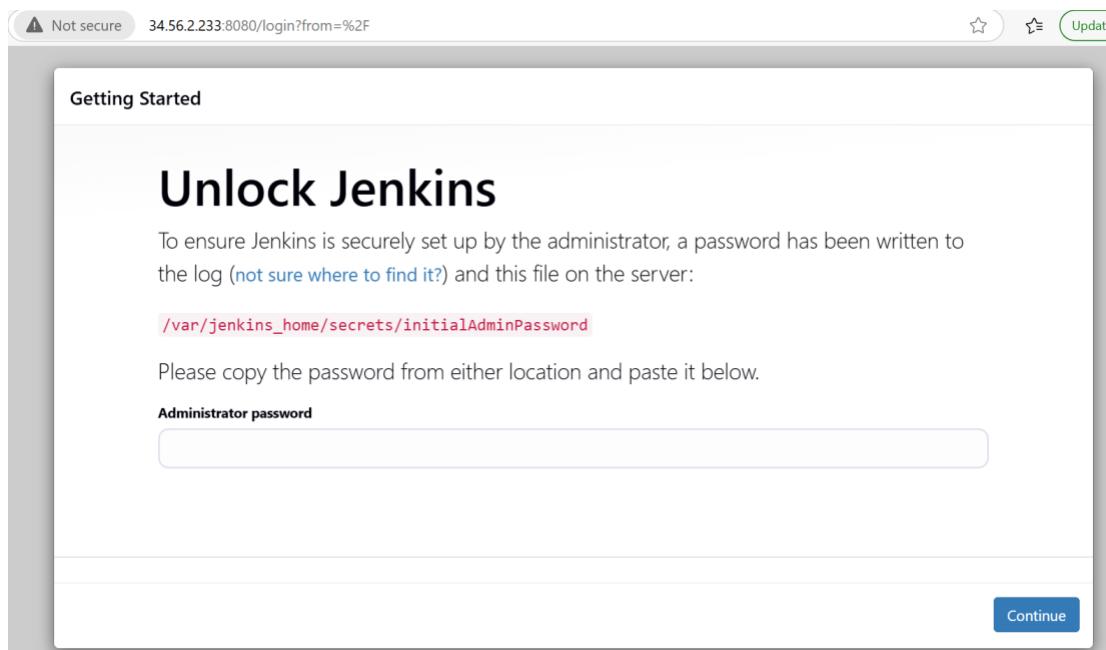
How can we access the pod

Take the external ip add the port no 8080

Open the browser paste ipaddress:8080

Now u can able to see the Jenkins

Output:-



Experiment 10

AIM:-Create a docker image for any application using docker file and push it to Docker Hub.

Step 1:-Connecting AWS Instance Ubuntu using Mobaxterm

1. Login AWS(Amazon Web Services) Account
2. Launch Instance name Docker
3. Connect to Ubuntu or mobaxterm

(note:-Follow this url for docker file and application—<https://github.com/devisar/devopslab>)

Step 2:-Create Docker Hub Account and create repository in Docker Hub

Step 3:-Install Docker and Check Status and Start Docker in mobaxterm

1. sudo apt update -y
2. sudo apt install docker.io -y
3. sudo systemctl status docker(come outside use command **ctrl+z**)

```
ubuntu@ip-172-31-90-47:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-02-28 03:50:40 UTC; 2min 5s ago
    TriggeredBy: • docker.socket
      Docs: https://docs.docker.com
       Main PID: 2110 (dockerd)
          Tasks: 8
         Memory: 35.8M (peak: 38.1M)
            CPU: 253ms
           CGroup: /system.slice/docker.service
                     └─2110 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 28 03:50:38 ip-172-31-90-47 systemd[1]: Starting docker.service - Docker Application Container Engine...
Feb 28 03:50:38 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:38.974271527Z" level=info msg="Starting up"
Feb 28 03:50:38 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:38.975783980Z" level=info msg="detected 127.0.0.53 name"
Feb 28 03:50:39 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:39.094061737Z" level=info msg="Loading containers: star
Feb 28 03:50:39 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:39.560107166Z" level=info msg="Loading containers: done
Feb 28 03:50:40 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:40.287913914Z" level=info msg="Docker daemon" commit="2
Feb 28 03:50:40 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:40.288020817Z" level=info msg="Daemon has completed ini
Feb 28 03:50:40 ip-172-31-90-47 dockerd[2110]: time="2025-02-28T03:50:40.338547546Z" level=info msg="API listen on /run/docke
Feb 28 03:50:40 ip-172-31-90-47 systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-21/21 (END)
```

Above status command is docker running means no problem if not run use command below to run

4. sudo systemctl start docker

Step 4:- Grant Access

Why we give grant access means

A easy way to verify your Docker installation is by running the below command

docker run hello-world

If the output says:

```
ubuntu@ip-172-31-90-47:~$ docker run hello-world
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
ubuntu@ip-172-31-90-47:~$
```

This can mean two things,

1. Docker deamon is not running.(start docker using “sudo systemctl start docker”)
2. Your user does not have access to run docker commands.

Grant Access to your user to run docker commands

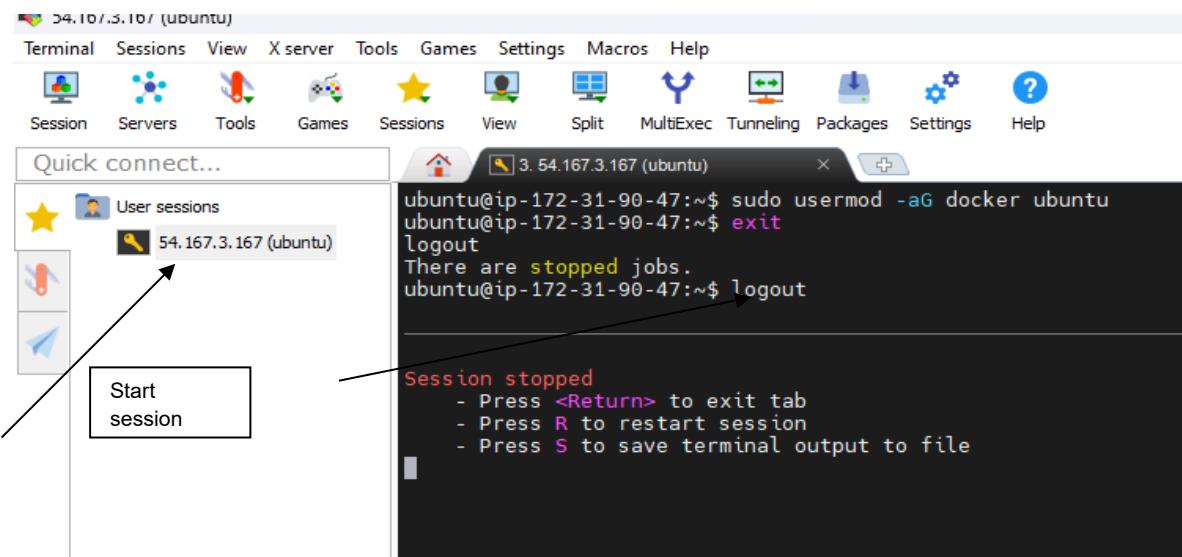
1. sudo usermod -aG docker ubuntu

In the above command ubuntu is the name of the user, you can change the username appropriately.

NOTE: : You need to logout and login back for the changes to be reflected.

2. Logout purpose use commands exit or logout

Again run command “docker run hello-world”



```
ubuntu@ip-172-31-90-47:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:bfb0cc14f13f9ed1ae86abc2b9f11181dc50d779807ed3a3c5e55a6936dbdd5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
```

Step 5:-Creating application and Docker file

1. Mkdir docker1
2. cd docker1
3. vim app.py

```
print("hello world")
```

4. cat app.py

5. vim Dockerfile (below pic commands write lab record) typing command vim before click “i” for insert data after completion Docker file commands save before click esc use :wq!

```

FROM ubuntu:latest
# Set the working directory in the image
WORKDIR /app
# Copy the files from the host file system to the image file system
COPY . /app
# Install the necessary packages
RUN apt-get update && apt-get install -y python3 python3-pip
# Set environment variables
ENV NAME World
# Run a command to start the application
CMD ["python3", "app.py"]
~"
~"
~"

```

Below image for understanding purpose

```

ubuntu@ip-172-31-90-47:~$ mkdir docker1
ubuntu@ip-172-31-90-47:~$ cd docker1
ubuntu@ip-172-31-90-47:~/docker1$ vim app.py
ubuntu@ip-172-31-90-47:~/docker1$ cat app.py
print("hello world")
ubuntu@ip-172-31-90-47:~/docker1$ vim Dockerfile
ubuntu@ip-172-31-90-47:~/docker1$ cat Dockerfile
FROM ubuntu:latest

# Set the working directory in the image
WORKDIR /app

# Copy the files from the host file system to the image file system
COPY . /app

# Install the necessary packages
RUN apt-get update && apt-get install -y python3 python3-pip

# Set environment variables
ENV NAME World

# Run a command to start the application
CMD ["python3", "app.py"]

ubuntu@ip-172-31-90-47:~/docker1$ 

```

Step 6:- Build and Check Docker image

Syntax:- docker build –t dockerhub_username/repositoryname:tag .

1. docker build –t anu1308/dockerimage:latest .
2. docker images

Below images for understanding purpose

```

ubuntu@ip-172-31-90-47:~/docker1$ docker build -t anu1308/dockerimage:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

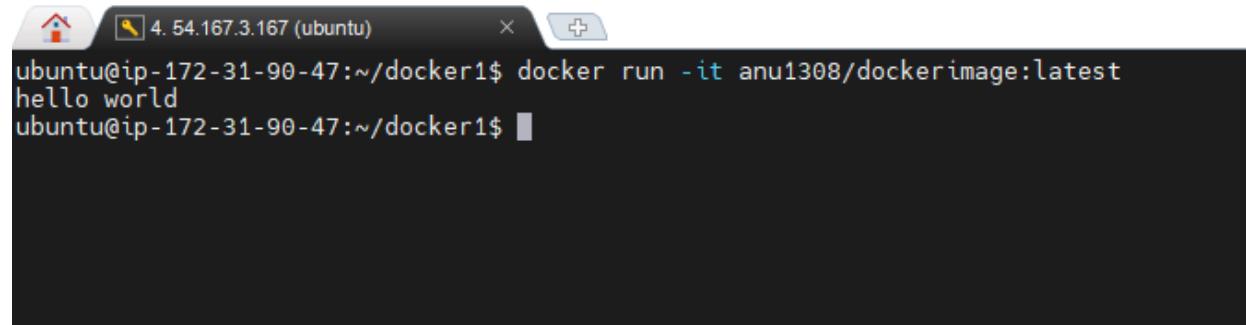
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
--> a04dc4851cbc
Step 2/6 : WORKDIR /app
--> Running in 490762c2b766
--> Removed intermediate container 490762c2b766
--> f19d0296889a
Step 3/6 : COPY . /app
--> 05e2e2564f3a
Step 4/6 : RUN apt-get update && apt-get install -y python3 python3-pip
--> Running in 0cad8878ec6b
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [842 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [24.2 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [807 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1053 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1131 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [28.8 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [881 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1336 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [16.0 kB]
Fetched 28.3 MB in 3s (8996 kB/s)
Successfully tagged anu1308/dockerimage:latest
ubuntu@ip-172-31-90-47:~/docker1$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
anu1308/dockerimage    latest   cdbf156a1eda  56 seconds ago  574MB
ubuntu              latest   a04dc4851cbc  4 weeks ago   78.1MB
hello-world          latest   74cc54e27dc4  5 weeks ago   10.1kB
ubuntu@ip-172-31-90-47:~/docker1$ 
```

Step 7:- Run your First Docker Container

1. docker run -it anu1308/dockerimage:latest

Output

Hello World



```

ubuntu@ip-172-31-90-47:~/docker1$ docker run -it anu1308/dockerimage:latest
hello world
ubuntu@ip-172-31-90-47:~/docker1$ 
```

Step 8:-Docker Login

1. docker login

```
ubuntu@ip-172-31-90-47:~/docker1$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, you can
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better
ess-tokens/
Username: anu1308
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-90-47:~/docker1$ █
```

Step 9:- Push the Image to DockerHub and share it with the world

1. docker push anu1308/dockerimage:latest

```
Login Succeeded
ubuntu@ip-172-31-90-47:~/docker1$ docker push anu1308/dockerimage:latest
The push refers to repository [docker.io/anu1308/dockerimage]
2d65eeb8aca1: Pushed
9863a1399ed4: Pushed
8870c5a70606: Pushed
4b7c01ed0534: Mounted from library/ubuntu
latest: digest: sha256:ec420e83236afe2170d2feac0ba4ff0af40d1eb580bb3c83a5e50a374c5d1746 size: 1155
ubuntu@ip-172-31-90-47:~/docker1$ █
```

Output:-

```
Successfully tagged anu1308/dockerimage:latest
ubuntu@ip-172-31-90-47:~/docker1$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
anu1308/dockerimage    latest   cdbf156a1eda  56 seconds ago  574MB
ubuntu              latest   a04dc4851cbc  4 weeks ago   78.1MB
hello-world         latest   74cc54e27dc4  5 weeks ago   10.1kB
ubuntu@ip-172-31-90-47:~/docker1$ █
```

The screenshot shows a Docker Hub repository page for 'anu1308/dockerimage'. The repository contains 1 tag, 'latest', which is an Image type. It was pushed less than a day ago and pulled less than a day ago. There are sections for 'Tags', 'Automated builds', and 'Repository overview'.

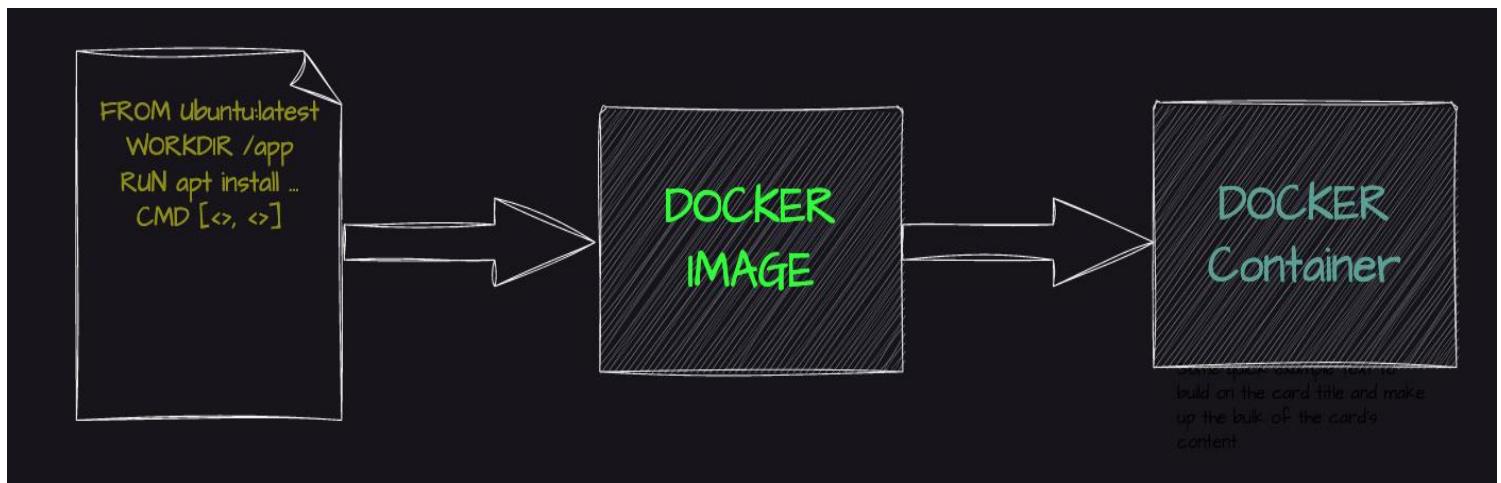
Definitions:-

Docker LifeCycle

We can use the above Image as reference to understand the lifecycle of Docker.

There are three important things,

1. docker build -> builds docker images from Dockerfile
2. docker run -> runs container from docker images
3. docker push -> push the container image to public/private registries(docker hub) to share the docker images.



What is a container ?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Why are containers light weight ?

Containers are lightweight because they use a technology called containerization, which allows them to share the host operating system's kernel and libraries, while still providing isolation for the application and its dependencies. This results in a smaller footprint compared to traditional virtual machines, as the containers do not need to include a full operating system. Additionally, Docker containers are designed to be minimal, only including what is necessary for the application to run, further reducing their size.

What is Docker ?

Docker is a containerization platform that provides easy way to containerize your applications, which means, using Docker you can build container images, run the images to create containers and also push these containers to container registries such as DockerHub

Docker daemon

The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

Docker client

The Docker client (docker) is the primary way that many Docker users interact with Docker. When you use commands such as docker run, the client sends these commands to dockerd, which carries them out. The docker command uses the Docker API. The Docker client can communicate with more than one daemon.

Docker Desktop

Docker Desktop is an easy-to-install application for your Mac, Windows or Linux environment that enables you to build and share containerized applications and microservices. Docker Desktop includes the Docker daemon (dockerd), the Docker client (docker), Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper. For more information, see Docker Desktop.

Docker registries

A Docker registry stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry. Docker objects

When you use Docker, you are creating and using images, containers, networks, volumes, plugins, and other objects. This section is a brief overview of some of those objects.

Dockerfile

Dockerfile is a file where you provide the steps to build your Docker Image.

Images

An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization. For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.

Below Images for Understanding purpose

Connecting AWS Instance to MobaXterm

A screenshot of a Google search results page. The search query "mobaxterm download" is entered in the search bar. Below the search bar, there are navigation links: All, Videos, Images, Shopping, Short videos, News, Forums, More, and Tools. The first search result is for "MobaXterm" with the URL <https://mobaxterm.mobatek.net/download>. An arrow points from the left towards this result. The result title is "MobaXterm Xserver with SSH, telnet, RDP, VNC and X11 - Download". Below the title, it says "Free X server for Windows with tabbed SSH terminal, telnet, RDP, VNC and X11-forwarding - Download.". There are three main sections under the result: "Home Edition", "Subscription", and "Plugins". Each section has a brief description and a link to "More results from mobatek.net »".

The screenshot shows the MobaXterm website with two main sections: 'Home Edition' and 'Professional Edition'.
Home Edition: Labeled 'Free', it lists features including Full X server and SSH support, Remote desktop (RDP, VNC, Xdmcp), Remote terminal (SSH, telnet, rlogin, Mosh), X11-Forwarding, Automatic SFTP browser, Master password protection, Plugins support, Portable and installer versions, Full documentation, Max. 12 sessions, Max. 2 SSH tunnels, Max. 4 macros, and Max. 360 seconds for Tftp, Nfs and Cron. A 'Download now' button is at the bottom.
Professional Edition: Labeled '\$69 / 49€ per user*', it lists additional features: Every feature from Home Edition +, Customize your startup message and logo, Modify your profile script, Remove unwanted games, screensaver or tools, Unlimited number of sessions, Unlimited number of tunnels and macros, Unlimited run time for network daemons, Enhanced security settings, 12-months updates included, Deployment inside company, and Lifetime right to use. It also includes payment icons for Pay, Visa, and MasterCard, and a 'Subscribe online / Get a quote' button.
A black arrow points from the text 'MobaXterm' in the page header down to the 'Download now' button in the Home Edition section.

MobaXterm Home Edition

Download MobaXterm Home Edition (current version):

[MobaXterm Home Edition v25.0 \(Portable edition\)](#)

[MobaXterm Home Edition v25.0 \(Installer edition\)](#)

Download previous stable version: [MobaXterm Portable v24.4](#) [MobaXterm Installer v24.4](#)

You can also get early access to the latest features and improvements by downloading MobaXterm Preview version:

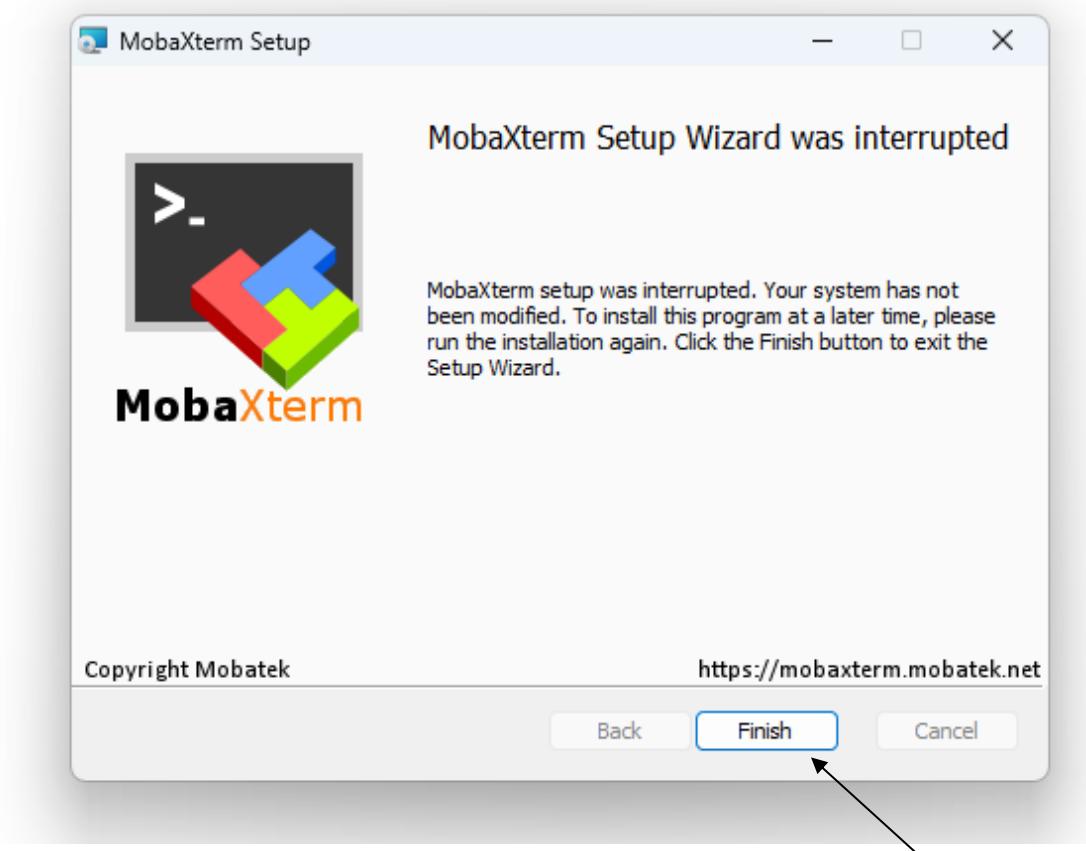
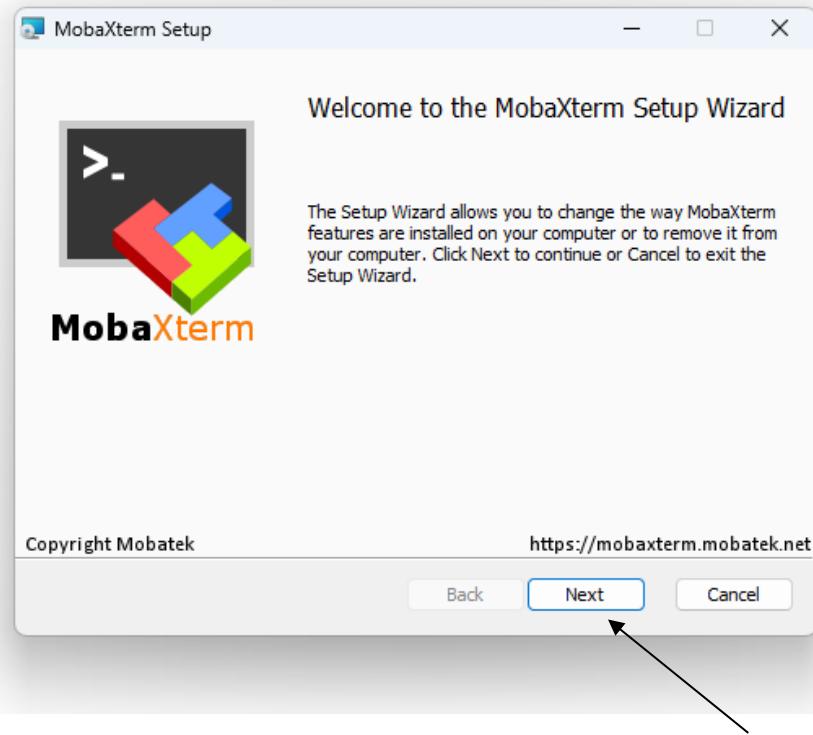
[MobaXterm Preview Version](#)

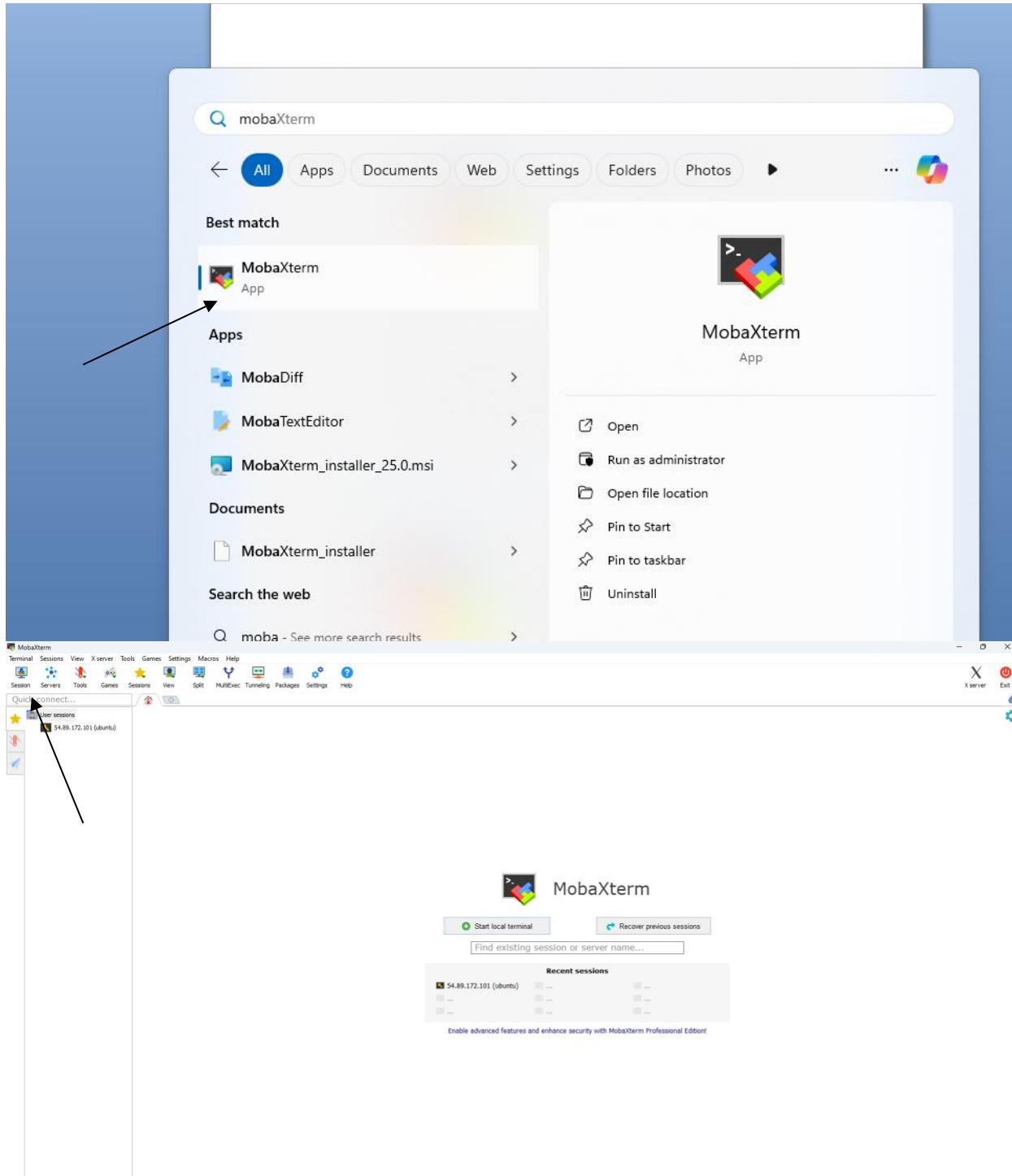
By downloading MobaXterm software, you accept [MobaXterm terms and conditions](#)

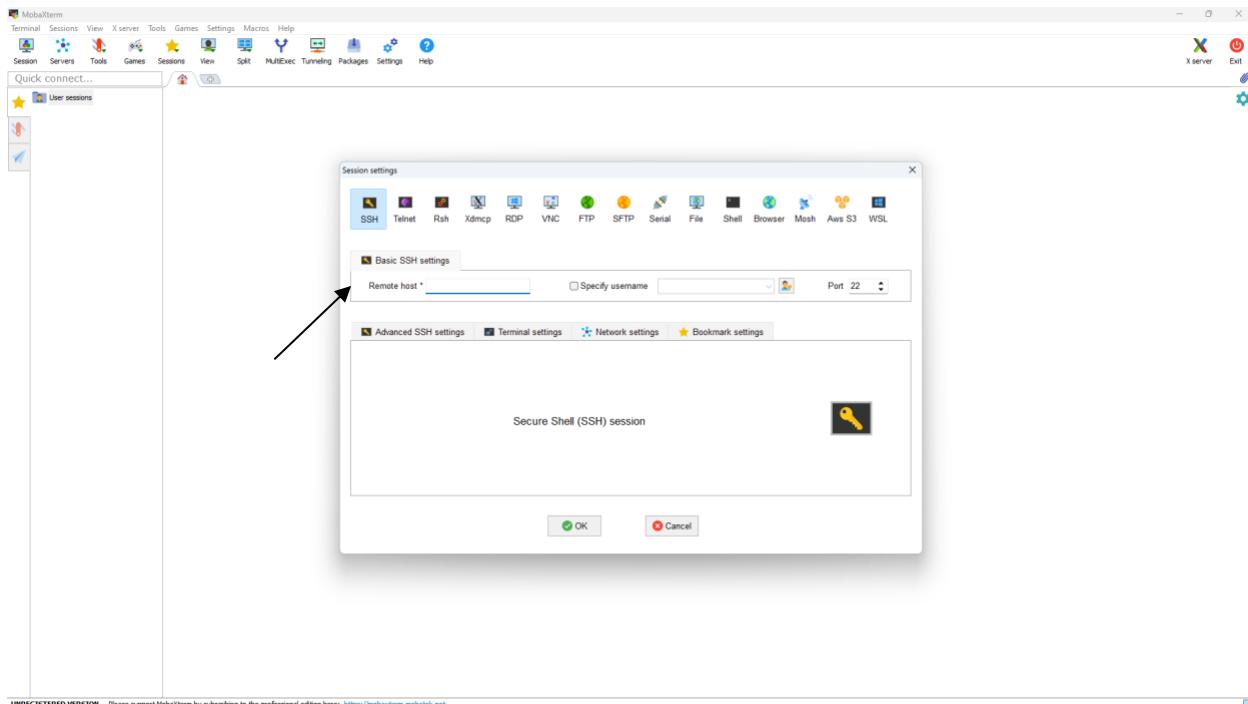
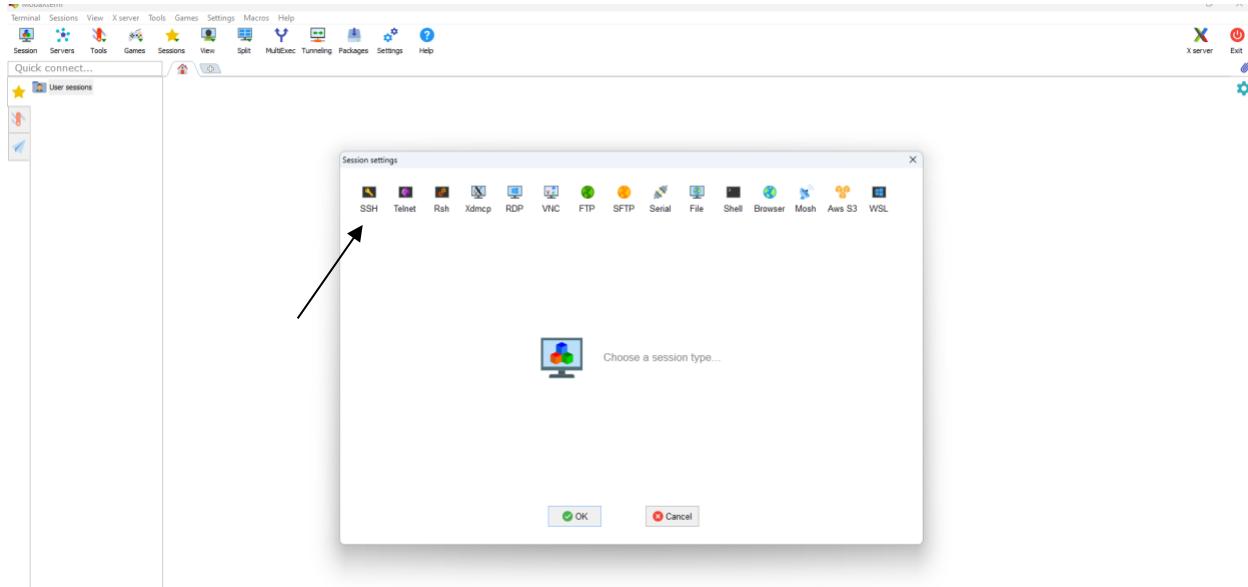
You can download the third party plugins and components sources [here](#)

If you use MobaXterm inside your company, you should consider subscribing to [MobaXterm Professional Edition](#): your subscription will give you access to professional support and to the "Customizer" software. This customizer will allow you to generate personalized versions of MobaXterm including your own logo, your default settings and your welcome message.
Please [contact us](#) for more information.

Name	Date modified	Type	Size
MobaXterm_Installer_v25.0	Date rr		
p5.pem	Date rr	Type: PEM File	Size: 1.
Docker-Zero-to-Hero-main	Date rr		
MobaXterm_Installer_v25.0	Date rr		
Earlier this week			
exp4-main (1)	Date rr		
Downloads > MobaXterm_Installer_v25.0			
Name	Date modified	Type	Size
Yesterday			
MobaXterm_installer.dat	2/27/2025 11:16 AM	DAT File	29,471 KB
MobaXterm_installer_25.0	2/27/2025 11:16 AM	Windows Installer ...	13,580 KB







UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Remote host we want public key and .pem file also so login to AWS Account

The image shows two screenshots of the AWS interface. The top screenshot is the 'Sign in' page, which includes a 'Try the new sign in UI' banner, a 'Root user' selection, a 'Root user email address' field containing 'username@example.com', and a 'Create a new AWS account' button. A black arrow points from the 'username@example.com' field towards the 'Launch instances' button on the EC2 Instances page. The bottom screenshot is the 'Instances' page under the EC2 service, showing the 'Instances Info' section with a search bar, filters, and a 'Launch instances' button. The sidebar on the left lists various AWS services like EC2, S3, Lambda, etc.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, a red arrow points from the 'Name' input field (containing 'Docker') to the 'Create key pair' modal. The modal is titled 'Create key pair' and contains fields for 'Key pair name' (set to 'p5'), 'Key pair type' (set to 'RSA'), and 'Private key file format' (set to '.pem'). A warning message at the bottom of the modal states: '⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)'.

Launch an instance info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name Add additional tags

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recent **Quick Start**

Amazon Linux macos Ubuntu Windows Red Hat SUSE Linux Debian

Ubuntu 

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-04b4f1a9cf54c11d0 (64-bit (x86)) / ami-0a7a4e87939439934 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

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

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64... [read more](#)

ami-04b4f1a9cf54c11d0

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Launch instance [Preview code](#)

Additional costs apply for AMIs with pre-installed software

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

Key pair name - required
p5

Network settings Info
Edit

Network | Info
vpc-0a06788c976618b9d

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

Auto-assign public IP | Info
Enable

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

Create security group Select existing security group

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

- Allow SSH traffic from Anywhere (0.0.0.0/0)
- Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

Storage (volumes)
1 volume(s) - 8 GiB

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6... [read more](#)
ami-0d4f4fa9a9cf54e11d0

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch instance

Preview code

Here you can Directly connect Ubuntu also but better use mobaxterm

EC2

Dashboard
EC2 Global View
Events

Instances

Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

Images

AMIs
AMI Catalog

Elastic Block Store

Volumes
Snapshots
Lifecycle Manager

Network & Security

Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces

Load Balancing

Load Balancers

Instances (1/1) Info

Last updated less than a minute ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Docker	i-09d3e11cc114228c5	Running	t2.micro	Initializing		us-east-1c	ec2-54-167-3-167.com...	54.167.3.167	-

i-09d3e11cc114228c5 (Docker)

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

Instance summary

Instance ID: i-09d3e11cc114228c5
IPv6 address:
Hostname type: IP name: ip-172-31-90-47.ec2.internal
Answer private resource DNS name: IPv4 (A)

Public IPv4 address: 54.167.3.167 | [open address](#)
Instance state: Running
Private IP DNS name (IPv4 only): ip-172-31-90-47.ec2.internal
Instance type: t2.micro

Private IPv4 addresses: 172.31.90.47
Public IPv4 DNS: ec2-54-167-3-167.compute-1.amazonaws.com | [open address](#)
Elastic IP addresses: -

Connect to instance

Connect to your instance i-09d3e11cc114228c5 (Docker) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

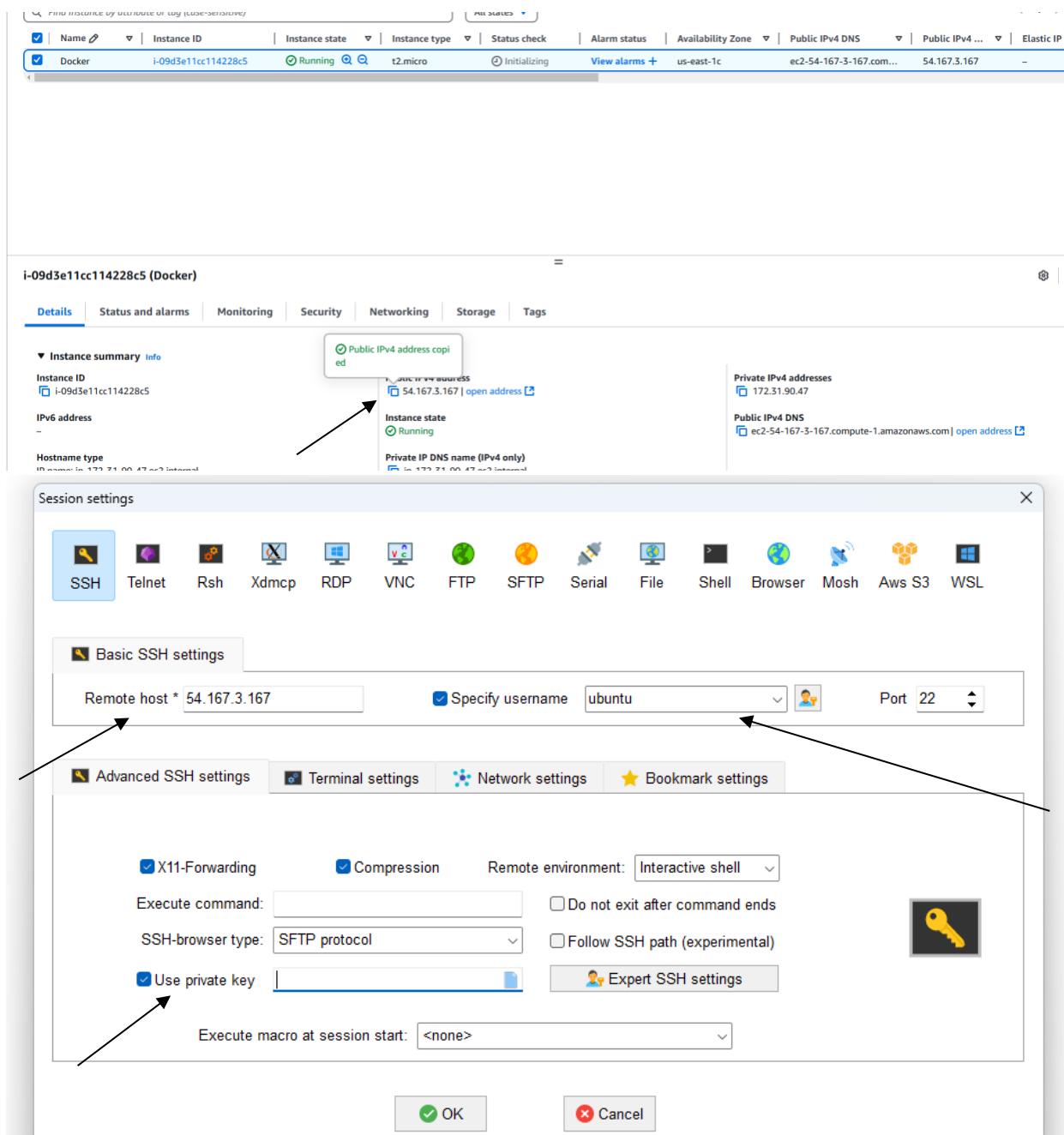
Public IPv4 address
54.167.3.167

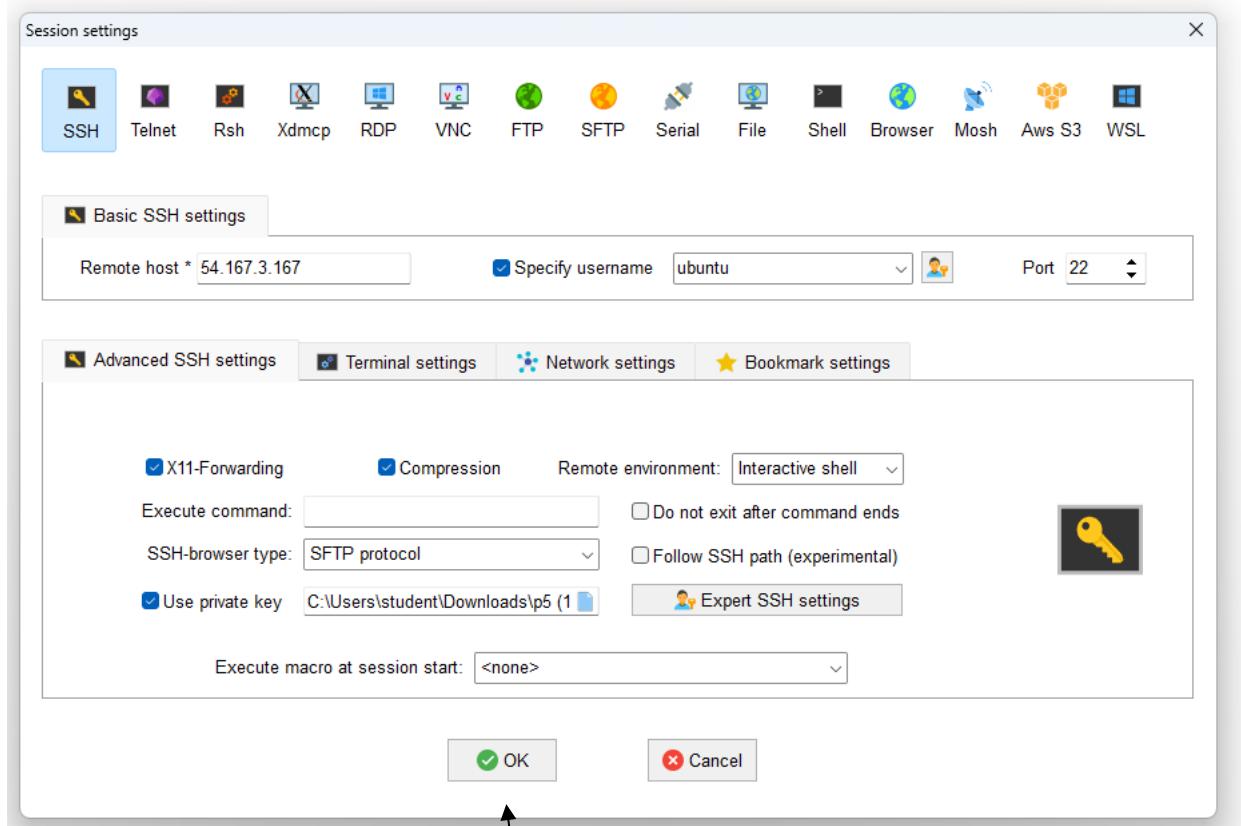
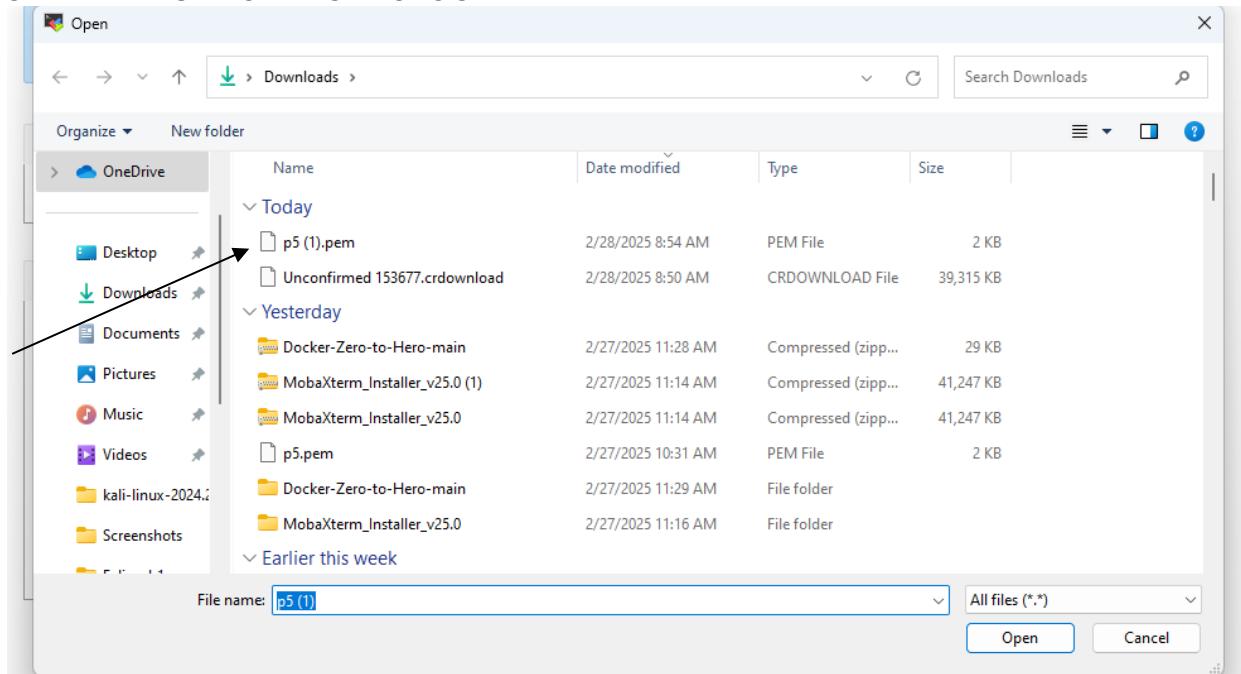
IPv6 address

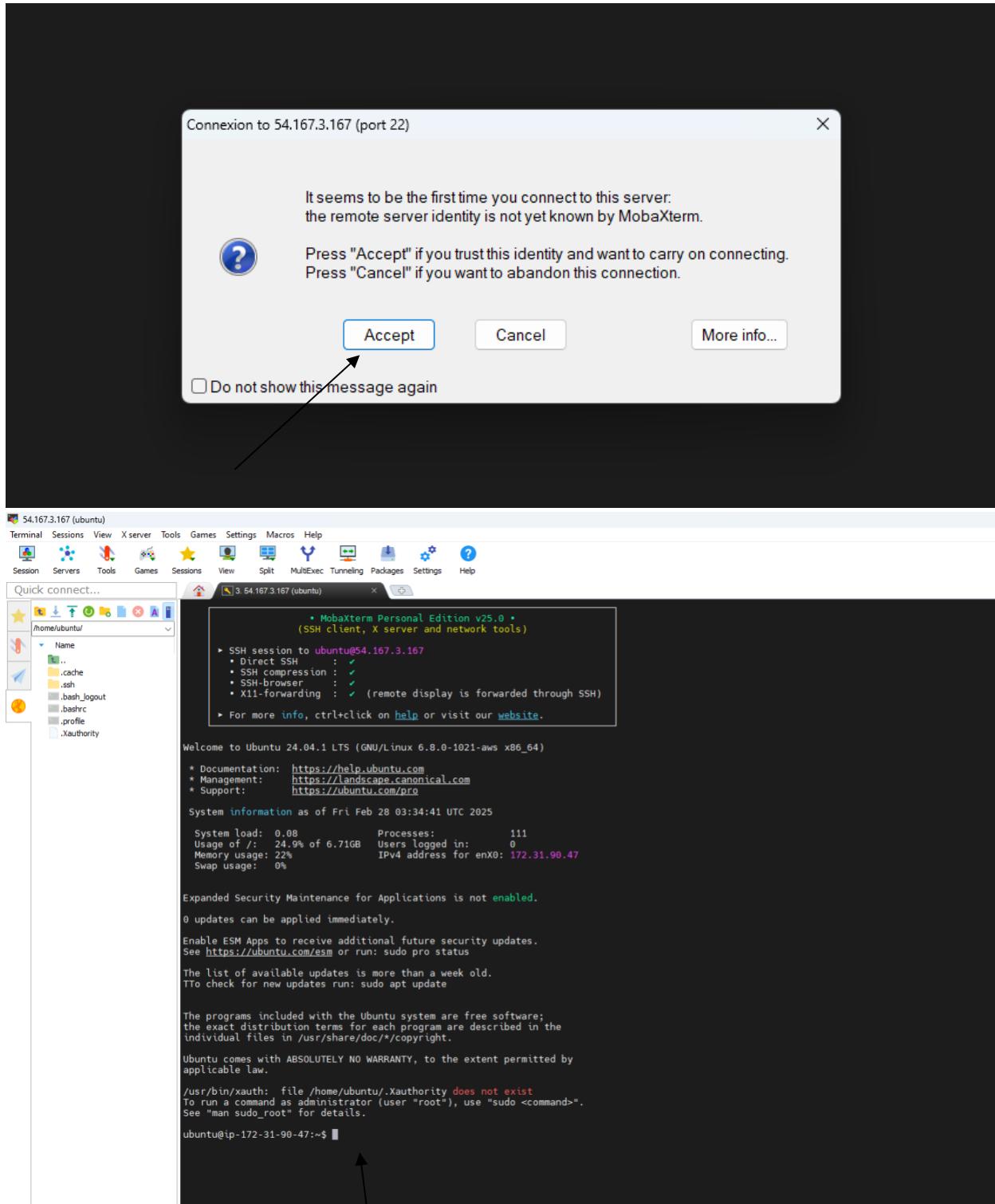
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Connect

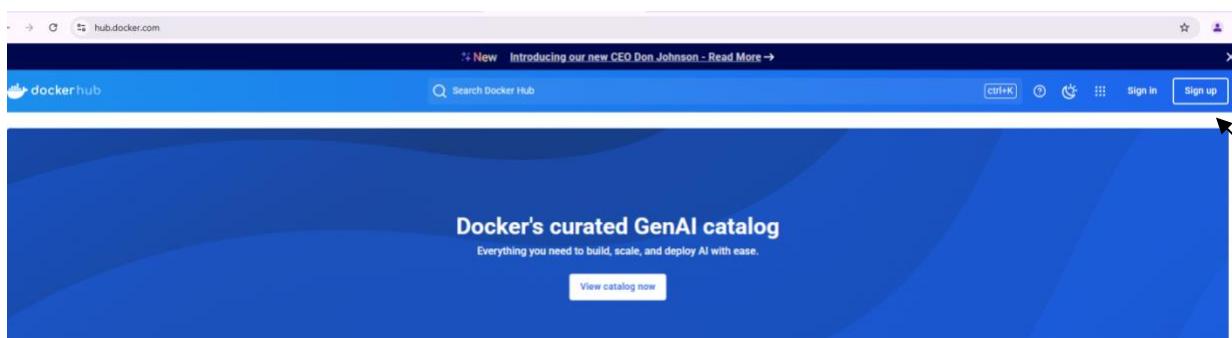






Create Docker Hub Account and create repository in Docker Hub

A screenshot of the Docker Hub homepage. At the top, there's a search bar with "docker hub" typed in, and standard browser controls (X, microphone, refresh). Below the search bar is a navigation menu with links for All, Shopping, Images, Videos, Short videos, News, Forums, and More. A vertical sidebar on the left lists categories: Docker Hub (https://hub.docker.com), Docker image, Signup, Search for Docker Images, Gen AI, and Sign in. Each category has a brief description and a "More" link. An arrow points from the "Docker Hub" link in the sidebar to the "Docker Hub Container Image Library | App Containerization" section. Another arrow points from the "Sign up" button in the sidebar to the "Sign up" button in the main content area.



The image shows the Docker account creation page. At the top is the Docker logo. Below it is the heading "Create your account". There are three input fields: "Email", "Username", and "Password" (with a visibility toggle). Below the password field is a checkbox labeled "Send me occasional product updates and announcements." An arrow points from the left towards this checkbox. To the right of the checkbox is a "Sign up" button. Below the sign-up button is the text "OR". Under "OR" are two social login options: "Continue with Google" and "Continue with GitHub". At the bottom is a link "Already have an account? Sign in".

Email

We suggest signing up with your work email address.

Username

Password

Send me occasional product updates and announcements.

Sign up

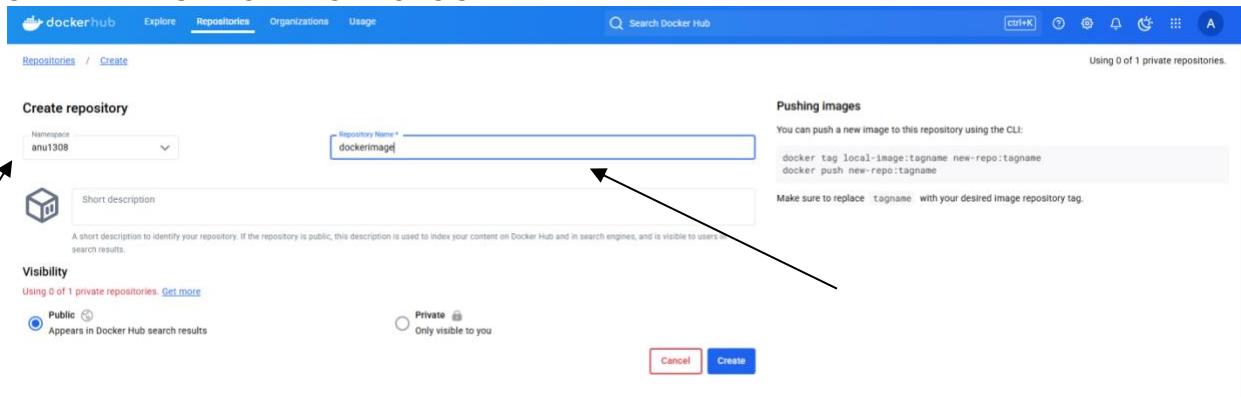
OR

Continue with Google

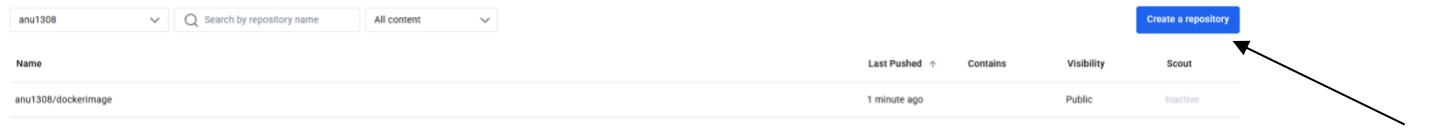
Continue with GitHub

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

The image shows two screenshots of the Docker interface. The top screenshot is the 'Sign in' page, which features the Docker logo at the top, followed by the title 'Sign in'. Below this, a message reads: 'Using Docker for work? We recommend signing in with your work email address.' A blue input field labeled 'Username or email address*' is shown, followed by a large blue 'Continue' button. Below the button is the text 'OR'. Three alternative sign-in options are listed: 'Continue with Google', 'Continue with GitHub', and 'Continue with SSO'. A black arrow points from the left towards the 'Continue' button. The bottom screenshot is a screenshot of the Docker Hub dashboard under the 'Repositories' tab. It shows a list of repositories, with one entry for 'anu1308/devopspipeline_demo' marked as 'DELETING'. At the top right of the dashboard, there is a blue 'Create a repository' button. A black arrow points from the right towards this button.



Docker Hub interface showing the 'Create repository' form. The 'Repository Name' field contains 'dockrimage'. The 'Visibility' section shows 'Public' selected. A 'Create' button is at the bottom right.



Docker Hub search results for 'anu1308'. The table shows one repository:

Name	Last Pushed	Contains	Visibility	Scout
anu1308/dockerimage	1 minute ago		Public	Inactive



Docker Hub search results for 'anu1308'. The table shows one repository:

Name	Last Pushed	Contains	Visibility	Scout
anu1308/dockerimage	1 minute ago		Public	Inactive

Experiment 11

Aim:-Setup Grafana for Devops.

ACCESS GRAFANA :

```
kubectl get secret prometheus-grafana -n monitoring -o jsonpath=".data.admin-user" | base64 --decode ; echo
```

If you run the above command u can see the username for grafana (**admin**)

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get secret prometheus-grafana -n monitoring -o jsonpath=".data.admin-user" | base64 --decode ; echo
admin
```

```
kubectl get secret prometheus-grafana -n monitoring -o jsonpath=".data.admin-password" | base64 --decode ; echo
```

If you run the abvoe command u can see the password for grafana (**prom-operator**)

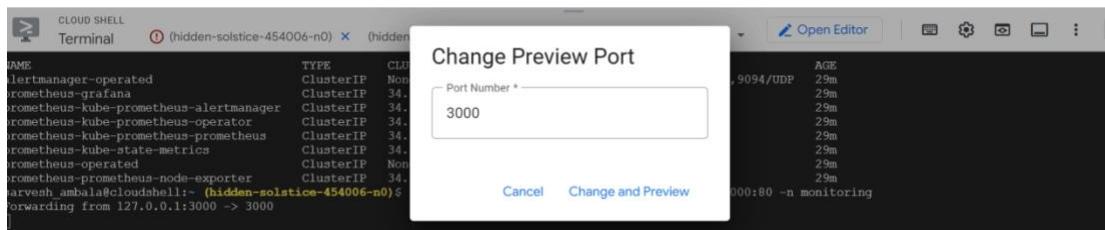
```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get secret prometheus-grafana -n monitoring -o jsonpath=".data.admin-password" | base64 --decode ; echo
prom-operator
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

PORt FORWARDING

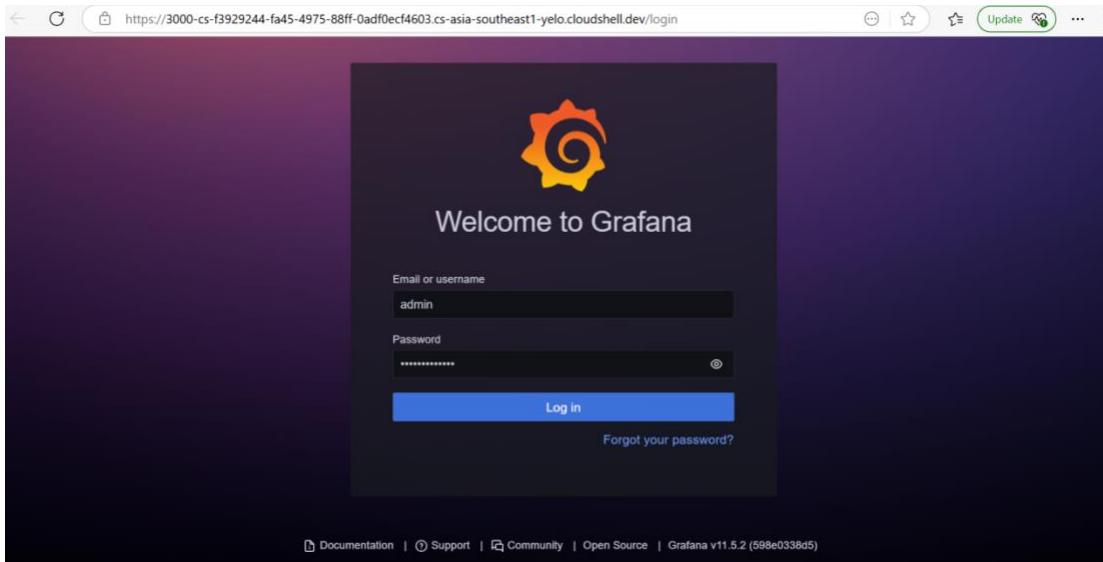
```
kubectl port-forward svc/prometheus-grafana 3000:80 -n monitoring
```

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl port-forward svc/prometheus-grafana 3000:80 -n monitoring
Forwarding from 127.0.0.1:3000 -> 3000
□
```

Click on the web preview give the port no 3000 and click on change and preview u can see the grafana



Output:-



You can login with admin and prom-operator

A screenshot of the Grafana dashboard titled 'Welcome to Grafana'. The top navigation bar includes links for Documentation, Tutorials, Community, and Public Slack. Below the title, there's a 'Basic' section with a description of the steps to set up Grafana. To the right, there are three cards: 'DATA SOURCE AND DASHBOARDS' (TUTORIAL), 'Add your first data source' (COMPLETE), and 'Create your first dashboard' (COMPLETE). Each card has a 'Remove this panel' link in the top right corner.

Experiment 12

Aim:-Setup Prometheus for Devops.

PROMETHEUS SETUP:

helm repo add prometheus <https://prometheus-community.github.io/helm-charts>

```
sarvesh ambala@cloudshell:~ (hidden-solstice-454006-n0)$ helm repo add prometheus https://prometheus-community.github.io/helm-charts
"prometheus" has been added to your repositories
```

helm repo update

```
Prometheus has been added to your repositories
sarvesh ambala@cloudshell:~ (hidden-solstice-454006-n0)$ helm repo update
Hang tight while we grab the latest from your chart repositories...
  ..Successfully got an update from the "grafana" chart repository
  ..Successfully got an update from the "prometheus1" chart repository
  ..Successfully got an update from the "prometheus" chart repository
  ..Successfully got an update from the "prometheus-community" chart repository
  ..Successfully got an update from the "pc" chart repository
```

helm install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring --create-namespace

This will install prometheus,alermanager and grafana

```
sarvesh ambala@cloudshell:~ (hidden-solstice-454006-n0)$ helm install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring --create-namespace
```

Check the prometheus pods and services

kubectl get pods -n monitoring

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get pods -n monitoring
NAME                                         READY   STATUS    RESTARTS   AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0   2/2     Running   0          3m12s
prometheus-grafana-75bb7d6986-rzq4q                   3/3     Running   0          3m19s
prometheus-kube-prometheus-operator-65c669f8f9-qcjwc   1/1     Running   0          3m19s
prometheus-kube-state-metrics-645c667b6-6bowl         1/1     Running   0          3m19s
prometheus-prometheus-kube-prometheus-prometheus-0     2/2     Running   0          3m11s
prometheus-prometheus-node-exporter-f87x4              1/1     Running   0          3m20s
prometheus-prometheus-node-exporter-gz8rs              1/1     Running   0          3m20s
prometheus-prometheus-node-exporter-vnb86              1/1     Running   0          3m20s
```

kubectl get svc -n monitoring

```
karthik_kumar@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl get svc -n monitoring
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP    PORT(S)          AGE
alertmanager-operated  ClusterIP  None         <none>        9093/TCP,9094/TCP,9094/UDP  4m27s
prometheus-grafana   ClusterIP  34.118.235.147  <none>        80/TCP          4m36s
prometheus-kube-prometheus-alertmanager  ClusterIP  34.118.225.122  <none>        9093/TCP,8080/TCP  4m36s
prometheus-kube-prometheus-operator     ClusterIP  34.118.232.165  <none>        443/TCP         4m36s
prometheus-kube-prometheus-prometheus ClusterIP  34.118.232.239  <none>        9090/TCP,8080/TCP  4m36s
prometheus-kube-state-metrics       ClusterIP  34.118.228.217  <none>        8080/TCP         4m36s
prometheus-operated            ClusterIP  None         <none>        9090/TCP         4m26s
prometheus-prometheus-node-exporter ClusterIP  34.118.232.54   <none>        9100/TCP         4m36s
karthik_kumar@cloudshell:~ (hidden-solstice-454006-n0)$ 
```

Access prometheus and port forwarding

```
kubectl port-forward svc/prometheus-kube-prometheus-prometheus 9090:9090 -n monitoring
```

```
sarvesh_ambala@cloudshell:~ (hidden-solstice-454006-n0)$ kubectl port-forward svc/prometheus-kube-prometheus-prometheus 9090:9090 -n monitoring
Forwarding from 127.0.0.1:9090 -> 9090
```

Click on the webpreview



Change port no to 9090

Click on change and preview

solstice-454006-n0) \$
prometheus-kube-prometheus
solstice-454006-n0) \$
90

Status Name ↑ Location Tier ? Number of nodes Total vCPUs Total memory

my-cluster us-central1-a Standard 3 6 12 GB

-454006-n0) X (hidden

Change Preview Port

Port Number *

9090

Cancel Change and Preview

Now u can able to see prometheus in the browser

Output:-

The screenshot shows the Prometheus web interface. At the top, there is a navigation bar with links for 'Prometheus', 'Query' (which is selected), 'Alerts', and 'Status'. On the right side of the navigation bar are icons for 'Update', 'Logs', and other system status indicators. A prominent red banner at the top states 'Server time is out of sync' with a note about a 2m 3.055s drift. Below the banner is a search bar labeled 'Enter expression (press Shift+Enter for newlines)' with an 'Execute' button. Underneath the search bar are three tabs: 'Table' (selected), 'Graph', and 'Explain'. A small panel below these tabs shows the current evaluation time. The main content area displays the message 'No data queried yet'. At the bottom left, there is a button for '+ Add query'.