



CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Approved by AICTE | Accredited by NAAC with 'A' Grade
All B. Tech Programs Accredited by NBA

DevOps Lab

B. Tech 3rd Year (CSE, DS)



CMR Institute of Technology

Kandlakoya, Medhchal, Hyderabad 501401

Syllabus

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	Ansible Setup and SSH keys.
5	Build WAR file in DevOps.
6	Create a docker image for any application using Docker file and push it to Docker hub.
7	Improvise the docker image quality using DevOps.
8	Build a selenium grid in DevOps.
9	Build and deploy a grid for Chrome and Firefox based testing.
10	Deploy a tested image on the server.
11	Perform automation using Jenkins.

References

1. DevOps Lab Manual, Department of CSE (DS), 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.

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.

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.

Experiment-2

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

Aim: Setups Eclipse for Devops

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

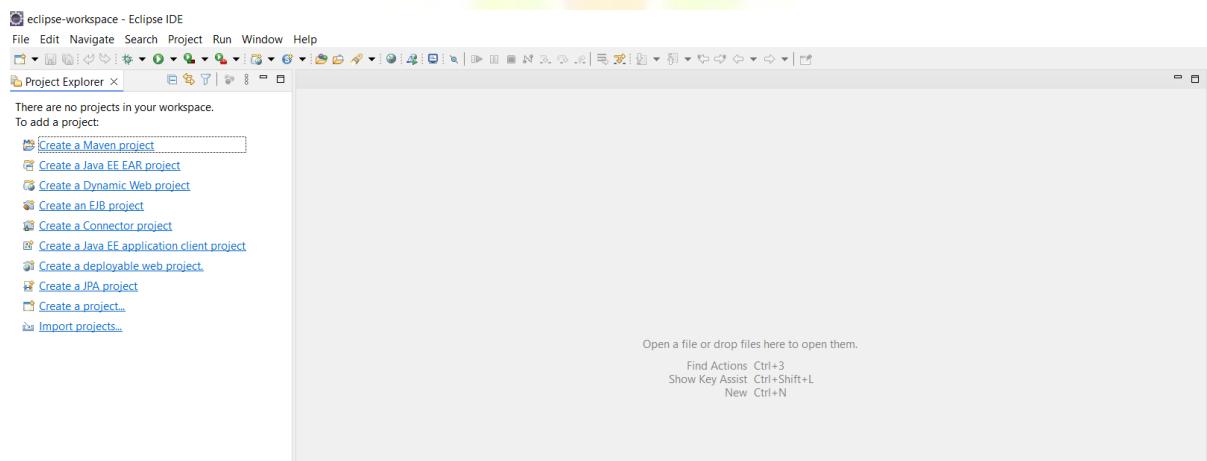
Step-2: Download eclipse zip file and extract the contents the all eclipse file

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

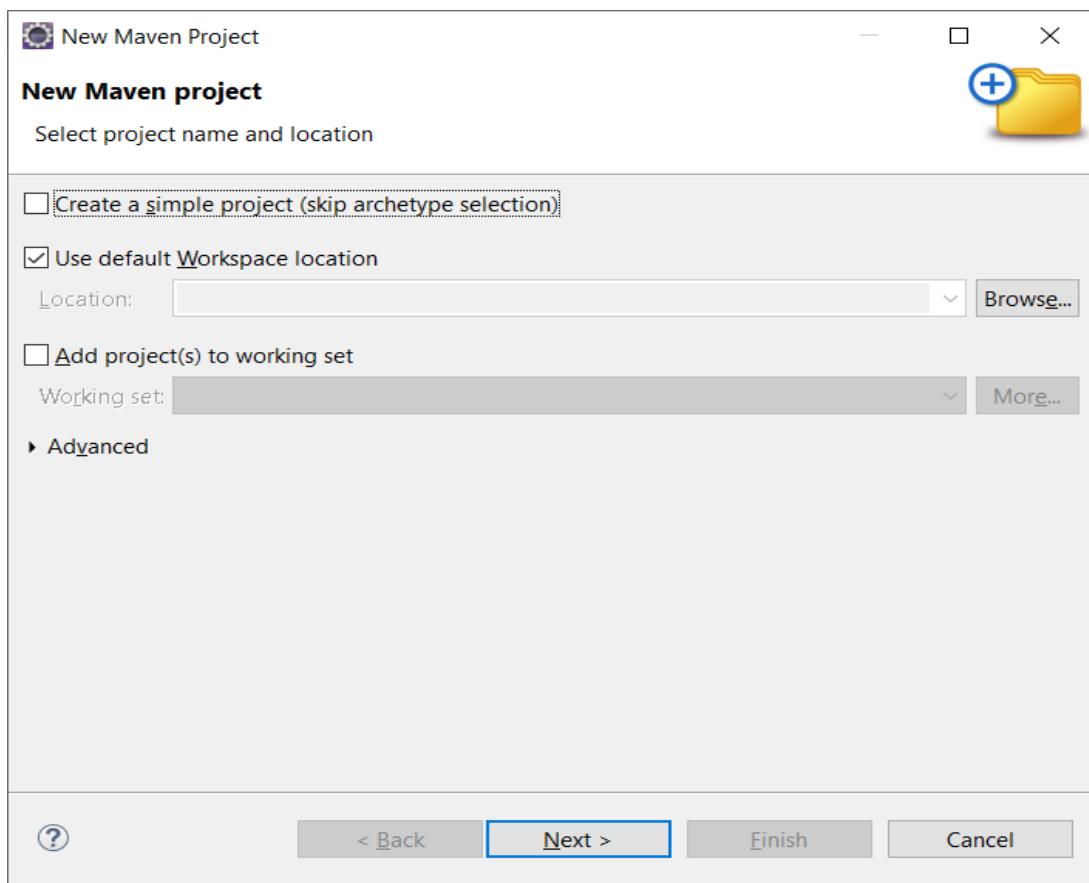
OR

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

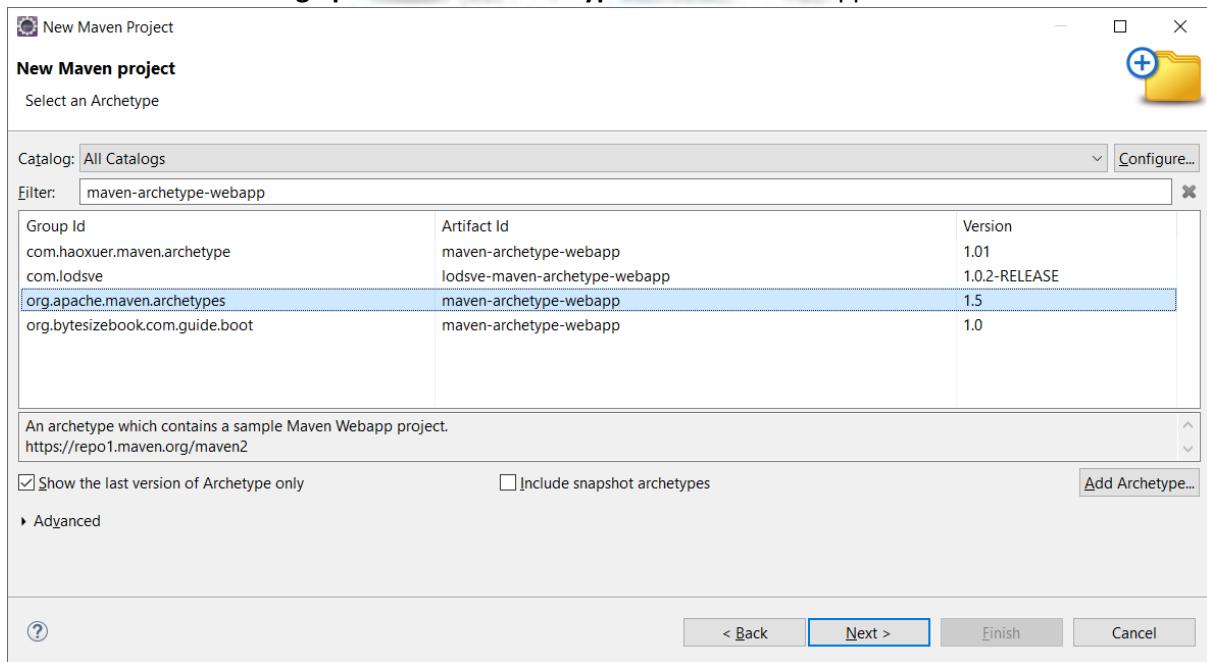
a.



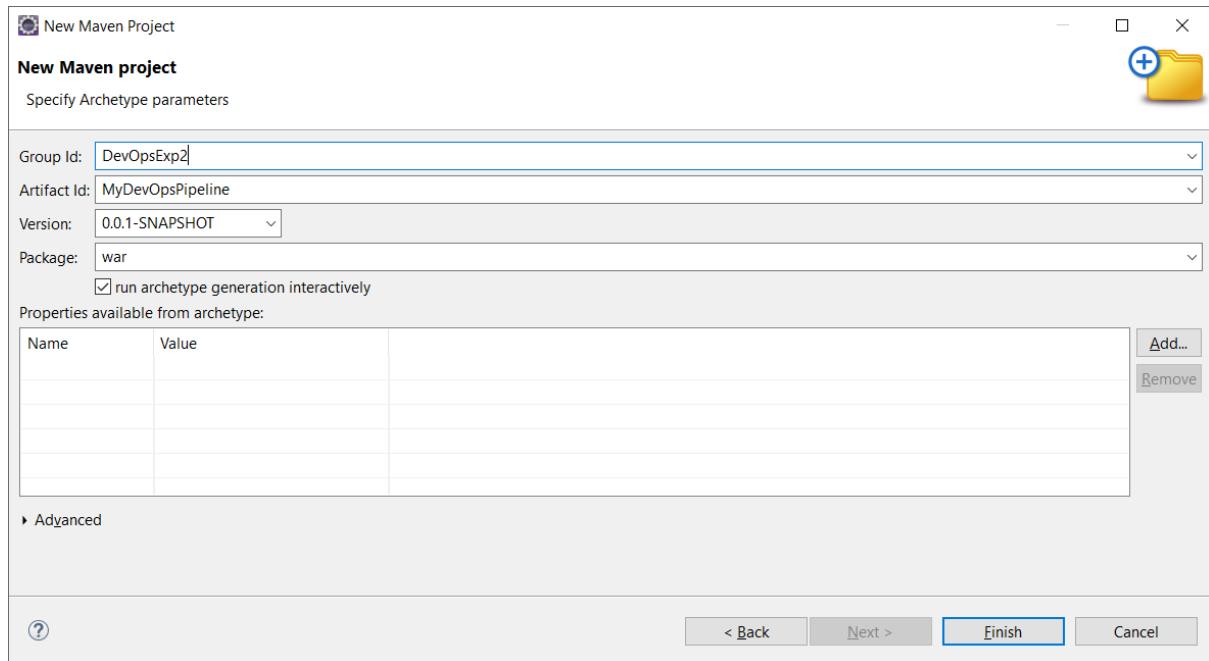
b.



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 a **artifact id:** you can type anything like your roll number



- d. Click Finish

[Create a Connector project](#)
[Create a Java EE application client project](#)
[Create a deployable web project](#)
[Create a JPA project](#)
[Create a project...](#)
[Import projects...](#)

```

Open a file or drop files here to open them.
Find Actions Ctrl+3
Show Key Assist Ctrl+Shift+L
New Ctrl+N

Problems Servers Terminal Data Source Explorer Properties Console X
C:\eclipse\plugins\org.eclipse.justj.openjdkhotspot\jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (26-Dec-2024, 7:07:36 pm) [pid: 5840]
Progress (1): 16/16 MB
Progress (1): 16 MB

Downloaded from central: https://repo.maven.apache.org/maven2/archetype-catalog.xml (16 MB at 3.3 MB/s)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-webapp:1.5] found in catalog remote
[INFO] Using property: groupId = DevOpsExp2
[INFO] Using property: artifactId = MyDevOpsPipeline
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = war
Confirm properties configuration:
groupId: DevOpsExp2
artifactId: MyDevOpsPipeline
version: 0.0.1-SNAPSHOT
package: war
Y: Y

```

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

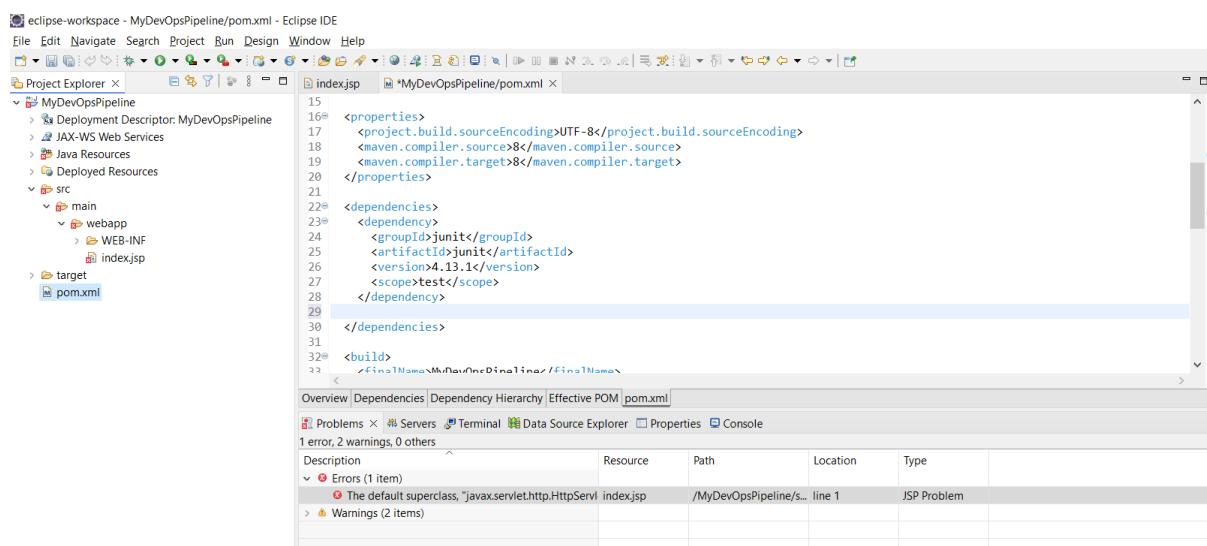
```

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

Problems Servers Terminal Data Source Explorer Properties Console X
<terminated> C:\eclipse\plugins\org.eclipse.jdt.core\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: packagingInPathFormat, 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] -----

```

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 screenshot shows the Apache Tomcat official website. The main navigation bar includes links for Home, Documentation, and Downloads. The 'Downloads' section is expanded, showing options for Tomcat 11, 10, 9, and the 'Tomcat Migration Tool for Jakarta EE'. A red arrow points to the 'Tomcat Migration Tool for Jakarta EE 1.0.9 Released' section. This section contains a summary of the release, a changelog, and a link to the full changelog. The Apache Software Foundation logo is visible in the top right corner.

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 \(pgp, sha512\)](#)



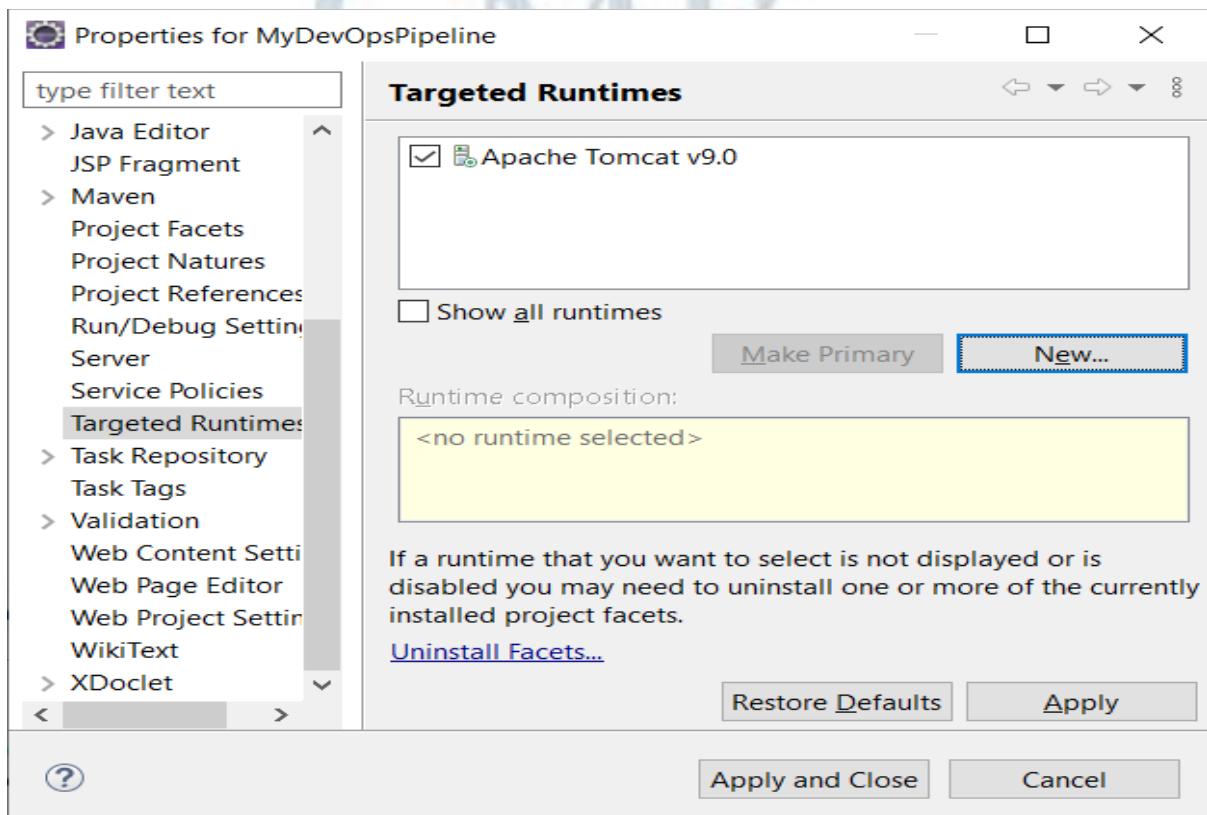
Source Code Distributions

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

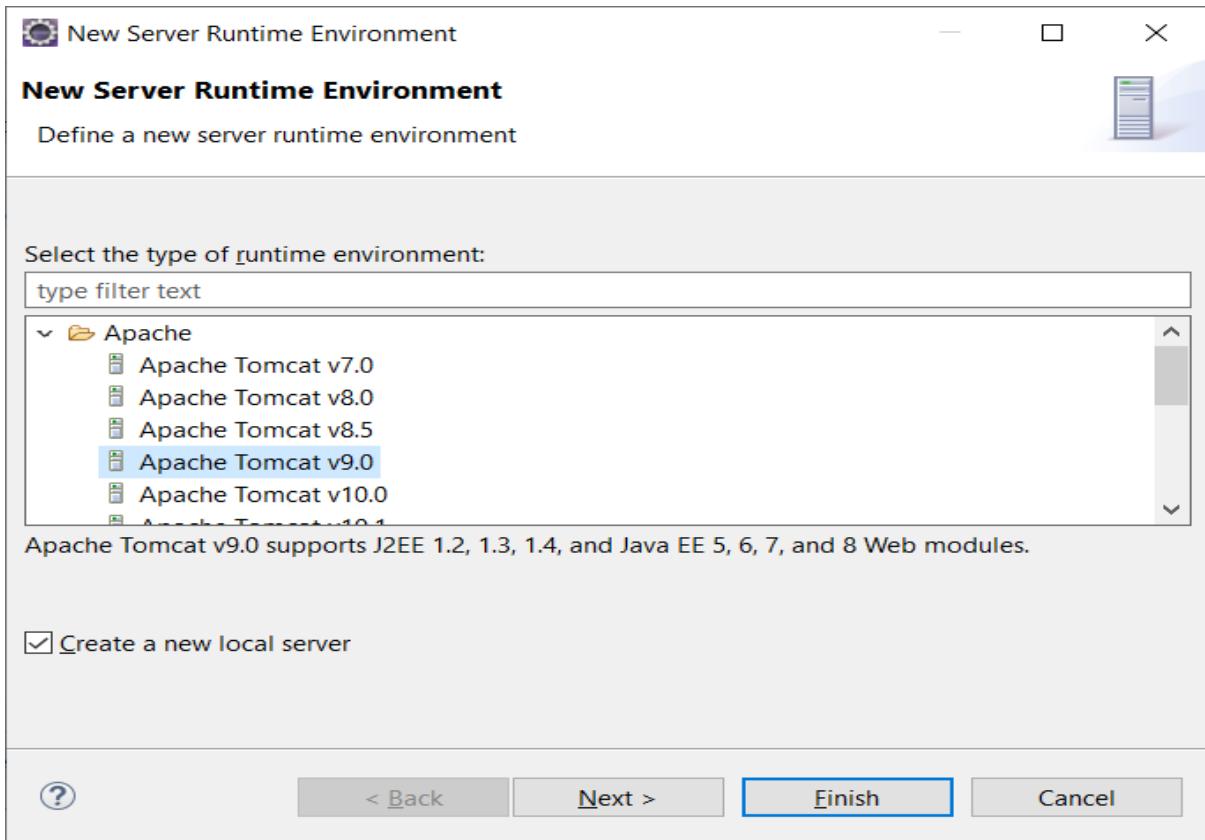
Step-7: After Download the Apache tomcat, Extract the .zip file and paste your apache-tomcat-9.0.98 folder in your folder

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

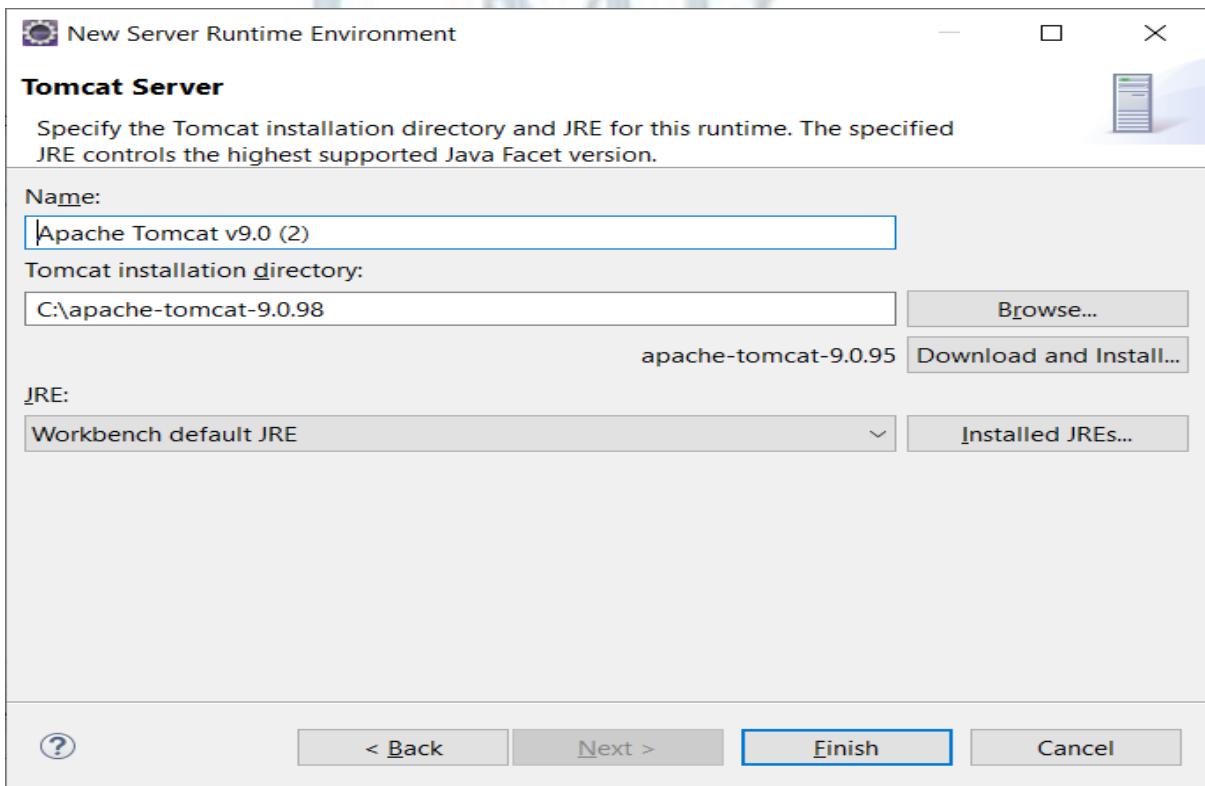
Step-9: Click on **new or follow the given image**



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

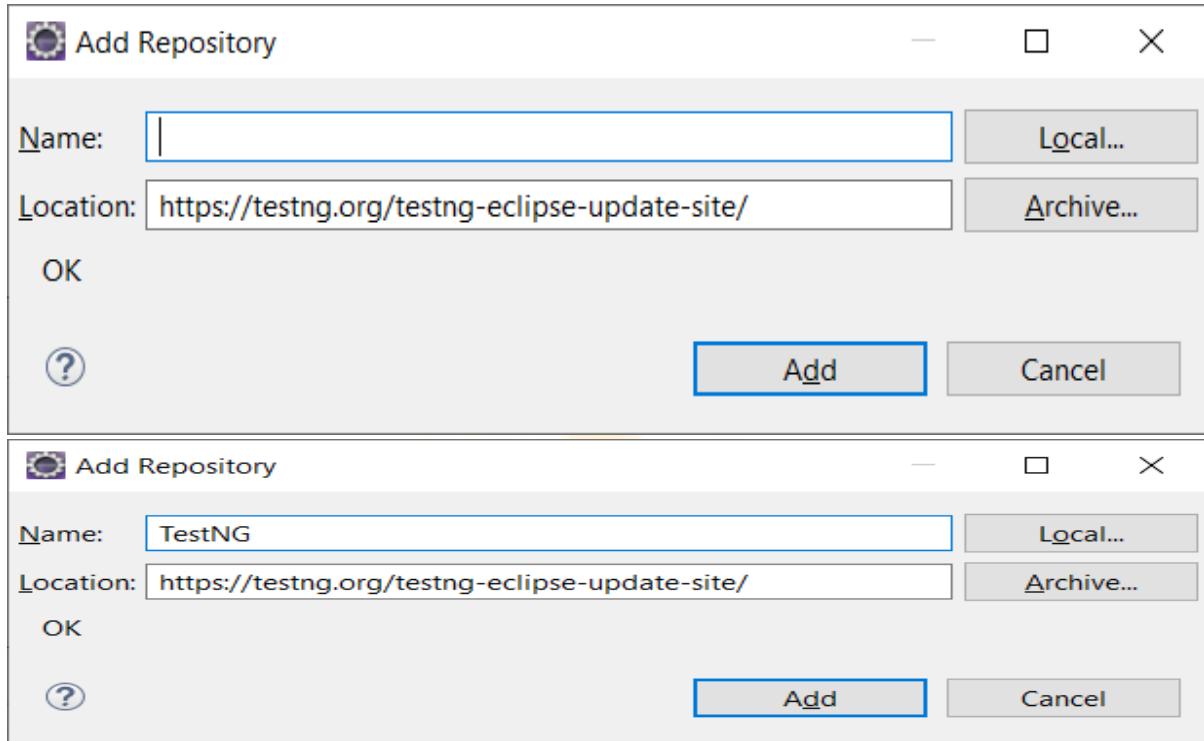


Step-12: Now Click on **help** Menu -> click on **Install new Software**.

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

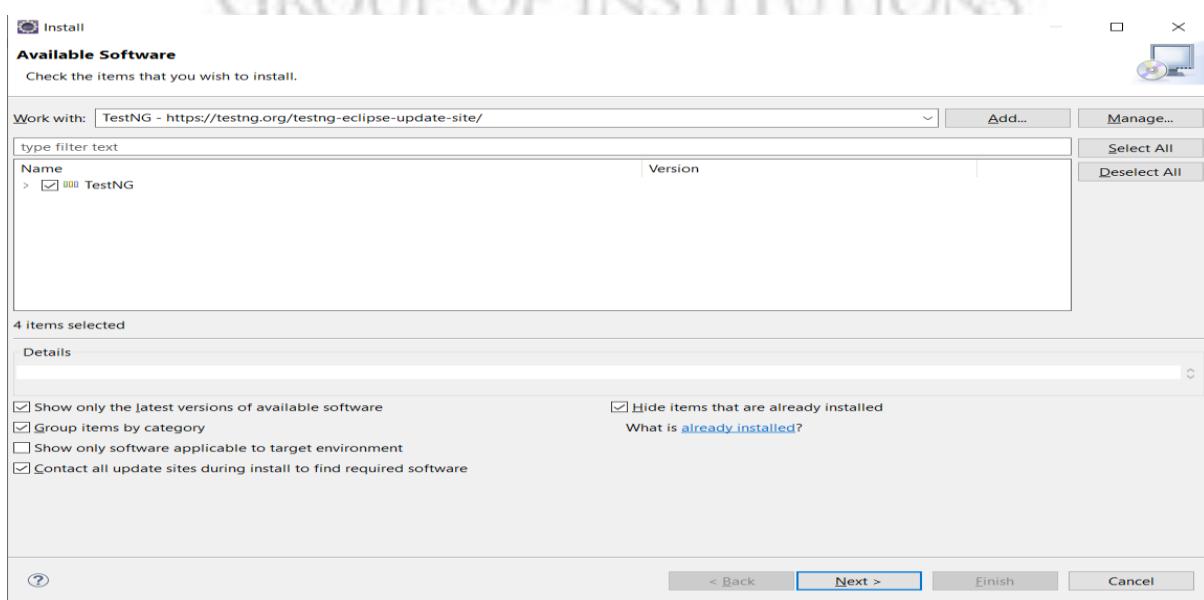
In the place of Name type: **TestNG**

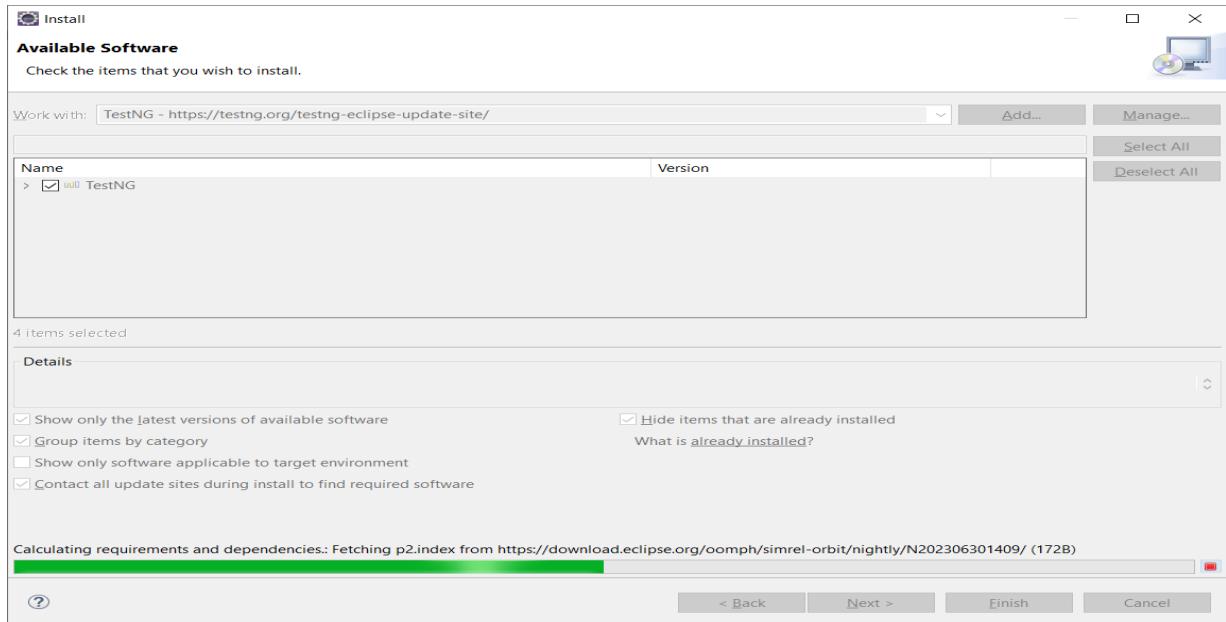
In the place of Location type: <https://testng.org/testng-eclipse-update-site/>



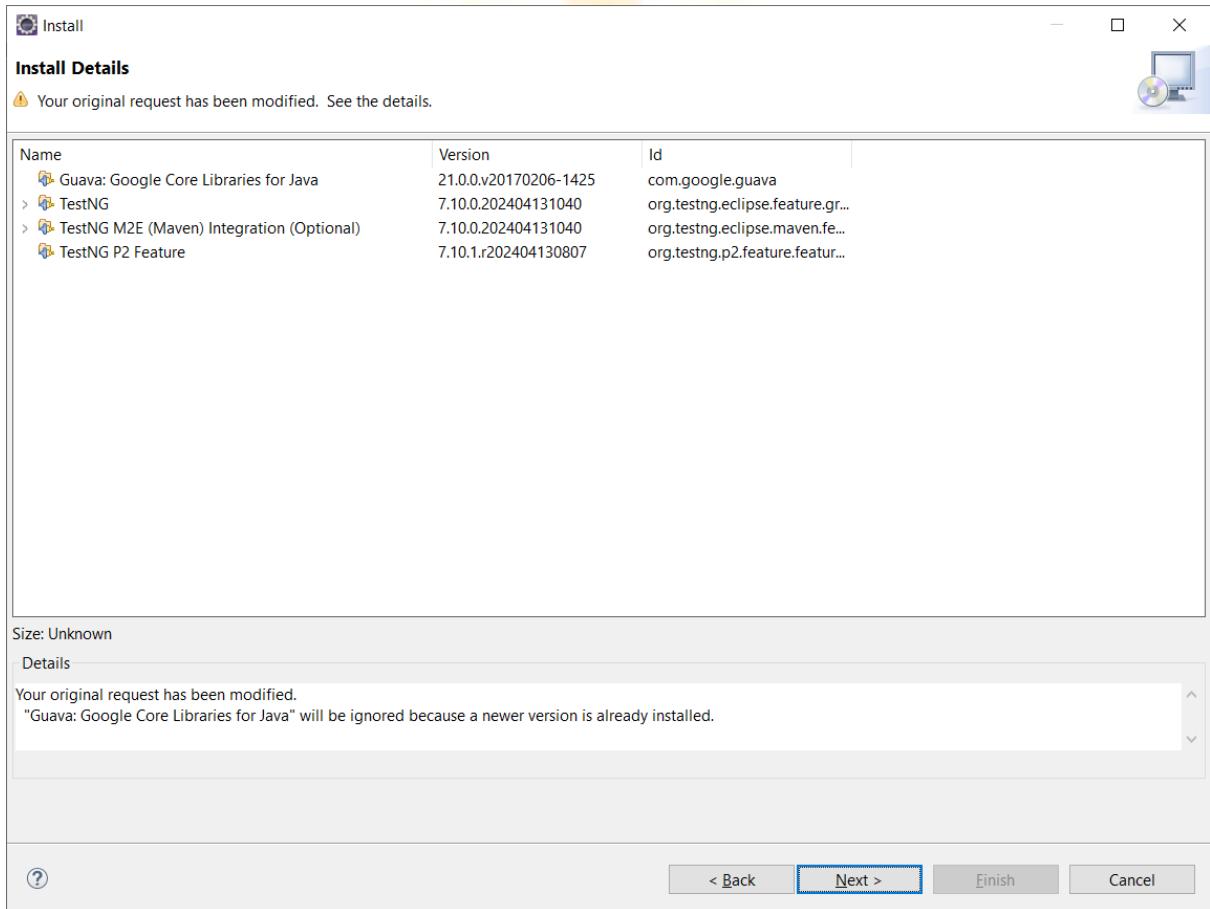
Step-14: Click on **Add** -> It will load a **testNG Dependencies** -> Select **TestNg** like given Image and then click **Next**.

It will take 10 minute to update TestNG in our Project

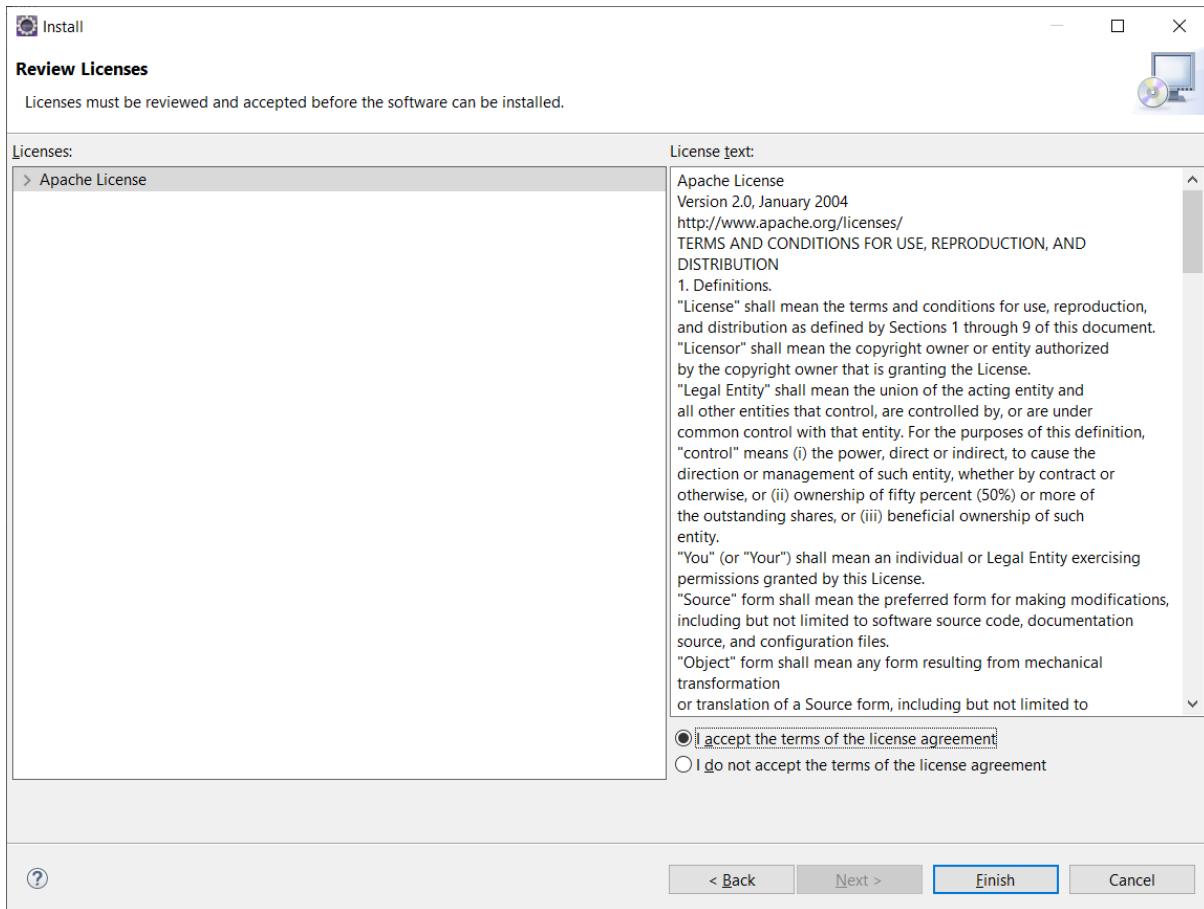




Step-15: After downloading the all dependencies it will show some file select all and click on next.



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**.

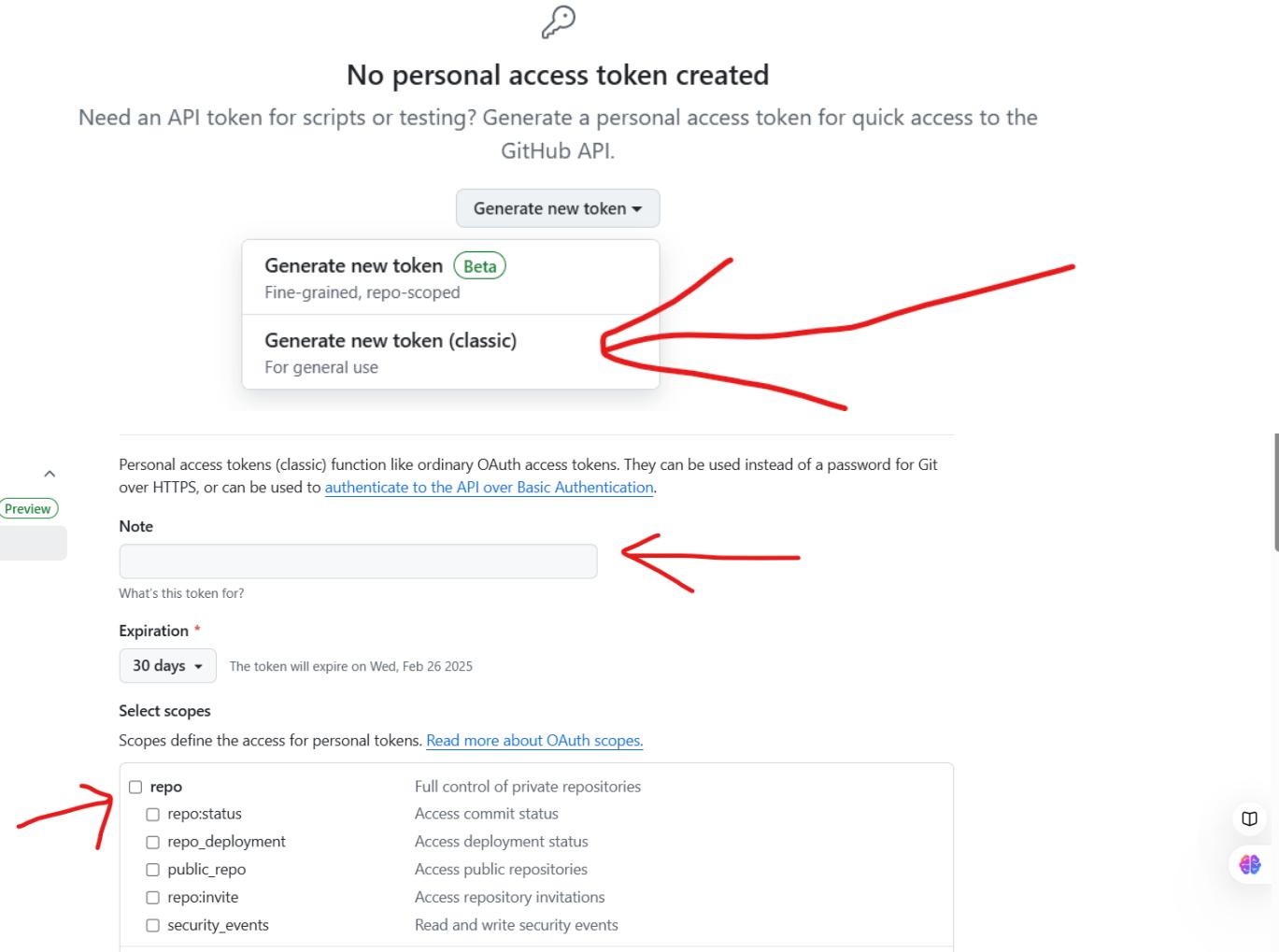
Step-19: Create a New **Repository** and Copy your **Repository** and paste in notepad

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

The screenshot shows the GitHub dashboard. On the left, there's a sidebar with 'Top repositories' and a search bar. The main area displays 'Trending repositories' with items like 'Jiayi-Pan/TinyZero' and 'deepseek-ai/DeepSeek-V3'. On the right, there's a 'GitHub Copilot' section with a banner and an 'Open Copilot' button. Below it is a 'Latest changes' feed. The top right corner features a user profile icon with a red arrow pointing to it.

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)

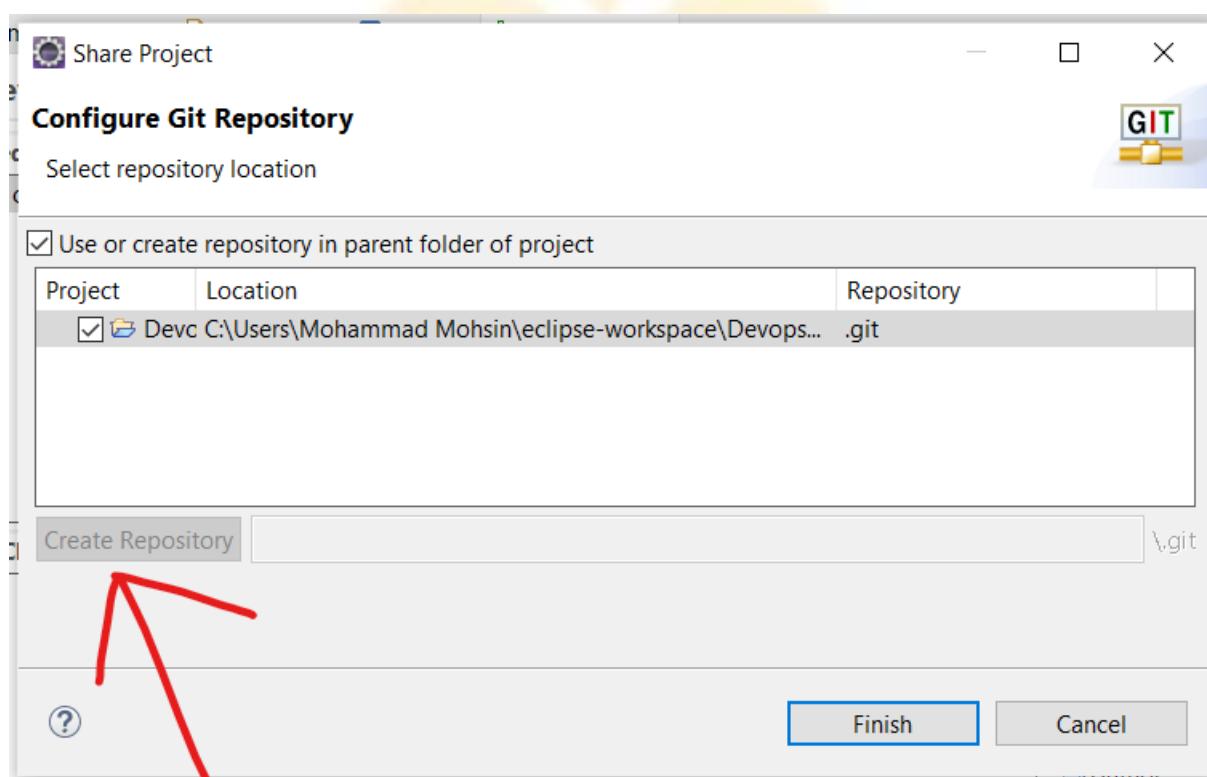
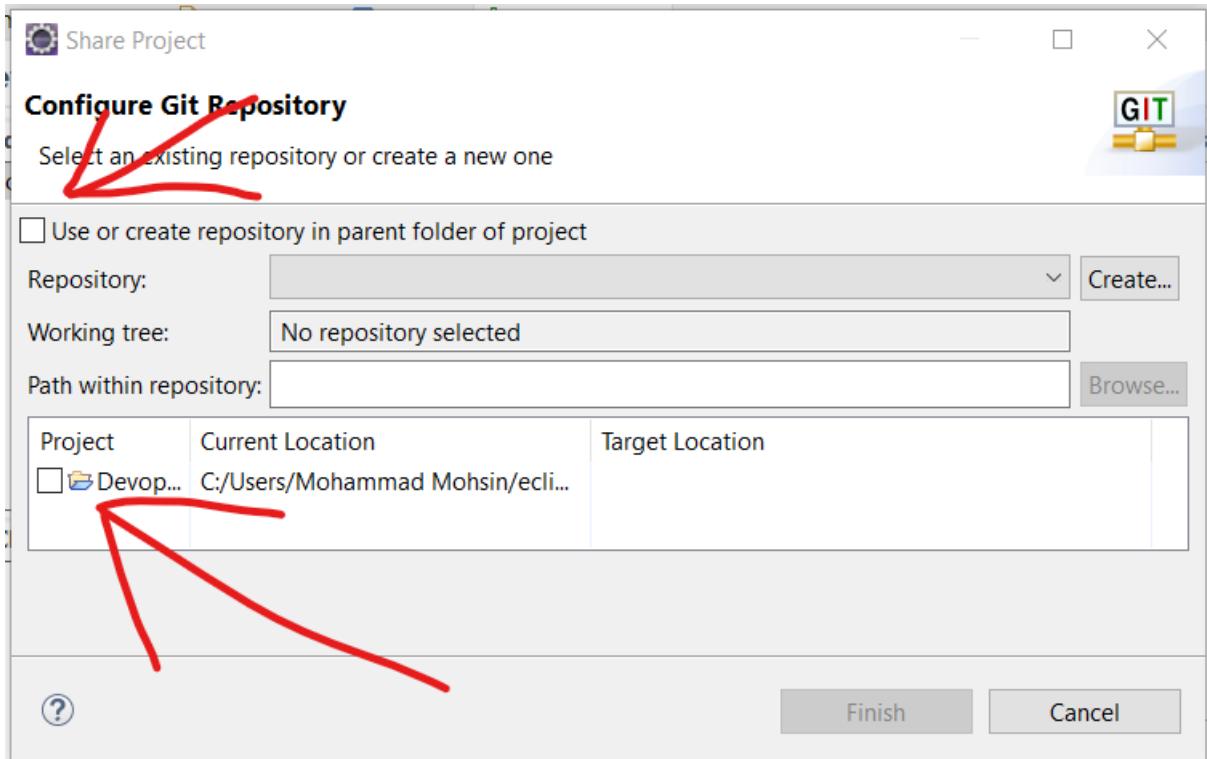
The screenshot shows the GitHub Dashboard. At the top right, there is a user profile icon with a red arrow pointing to it. On the left, there's a sidebar with links like 'Dashboard', 'Top repositories', 'Ask Copilot', 'Home', 'Trending repositories', 'Jiayi-Pan/TinyZero', 'deepseek-ai/DeepSeek-V3', 'Packages', 'Copilot', 'Pages', 'Saved replies', 'Security', 'Code security', 'Integrations', 'Applications', 'Scheduled reminders', 'Archives', 'Security log', 'Sponsorship log', and 'Developer settings'. A red arrow points from the bottom left towards the 'Developer settings' link. The main content area shows trending repositories and a GitHub Copilot promotional banner. Below the trending repos, there are sections for 'Social accounts' (with four empty fields) and 'Company' (with a field for linking a company GitHub organization). A 'Location' field is also present. A red arrow points to the 'Display current local time' checkbox under the location section. At the bottom, there's a 'GitHub Apps' section with 'Personal access tokens' selected, and a 'No GitHub Apps' section with a 'New GitHub App' button.



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 GitHub Setup, select the option **Use or create repository in parent folder of project** -> Select your Project and Click on **Create Repository** and click on **Finish**.

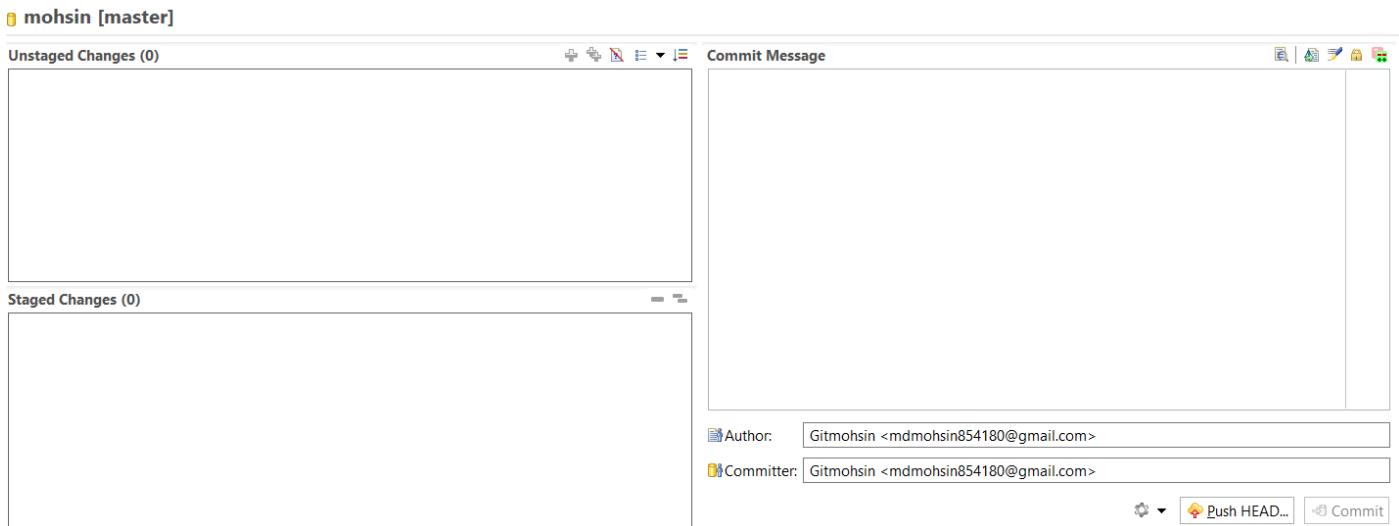


Step-25: After that again **Right click on your Project** and select the **Team -> click on Commit -> and stage your all file ->** and Write a comment (i.e. First Commit) and click on **Commit and push ->** after that it show an **error dialog ->** click **OK ->** now again click on **Push Head Button**

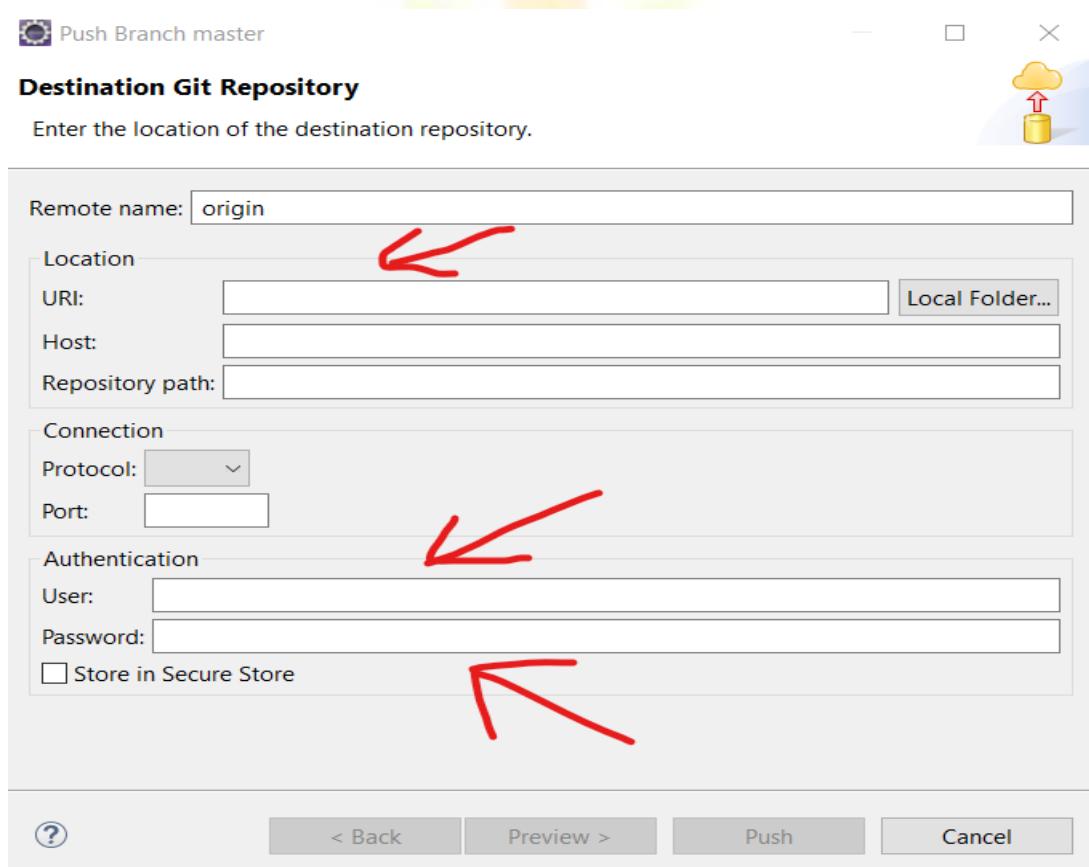
The screenshot shows the Eclipse IDE interface with the Git integration. In the top window, the 'Unstaged Changes' section contains 17 items, including '.classpath', '.gitignore', 'index.jsp', and various Maven configuration files. The 'Staged Changes' section is currently empty. On the right, the 'Commit Message' dialog is open with the message 'Unborn branch: this commit will create the branch 'master''. Below the message, the 'Author' and 'Committer' fields are set to 'Gitmohsin <mdmohsin854180@gmail.com>'. At the bottom of the dialog are buttons for 'Commit and Push...', 'Commit', and other options.

In this screenshot, the 'Unstaged Changes' section is now empty. The 'Staged Changes' section remains empty. The 'Commit Message' dialog shows the message 'first committ'. The author and committer information is identical to the previous screenshot. The bottom buttons are the same.

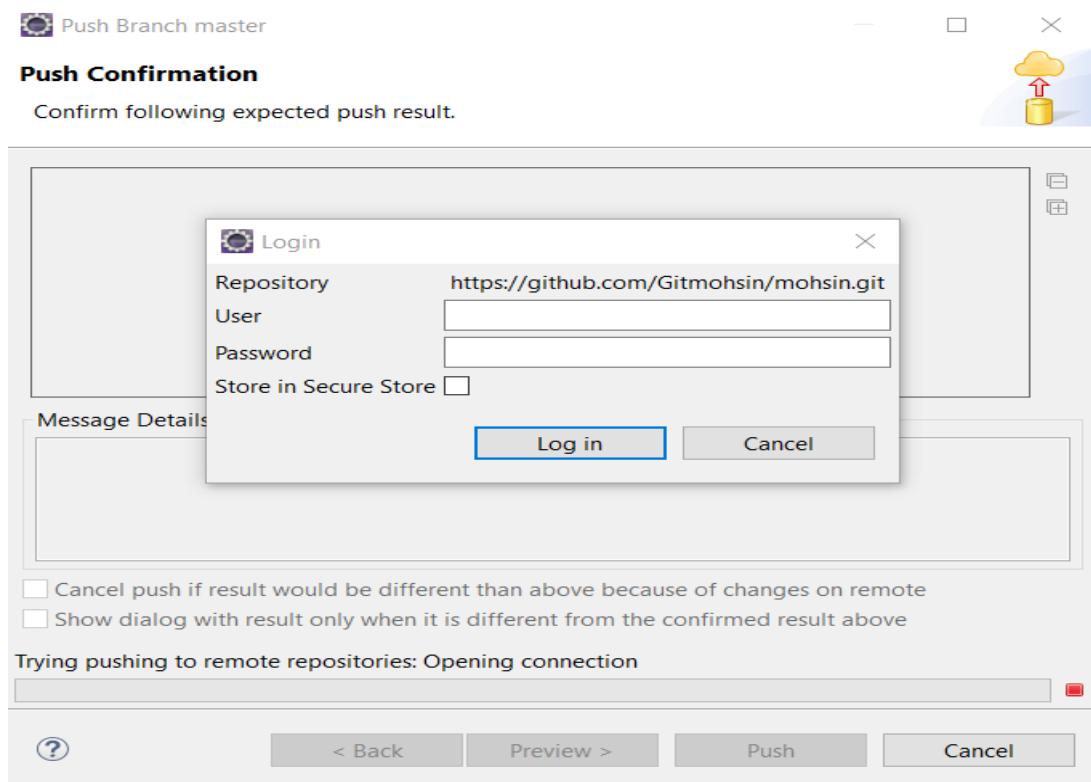
This is a separate 'Problem Occurred' dialog box. It features a red circular icon with a white 'X'. The text inside the box reads: "'Push to mohsin refs/heads/master - origin' has encountered a problem. Can't connect to any repository:". At the bottom are two buttons: 'OK' and 'Details >>'.



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 click on **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 **unstage 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.

Experiment-3

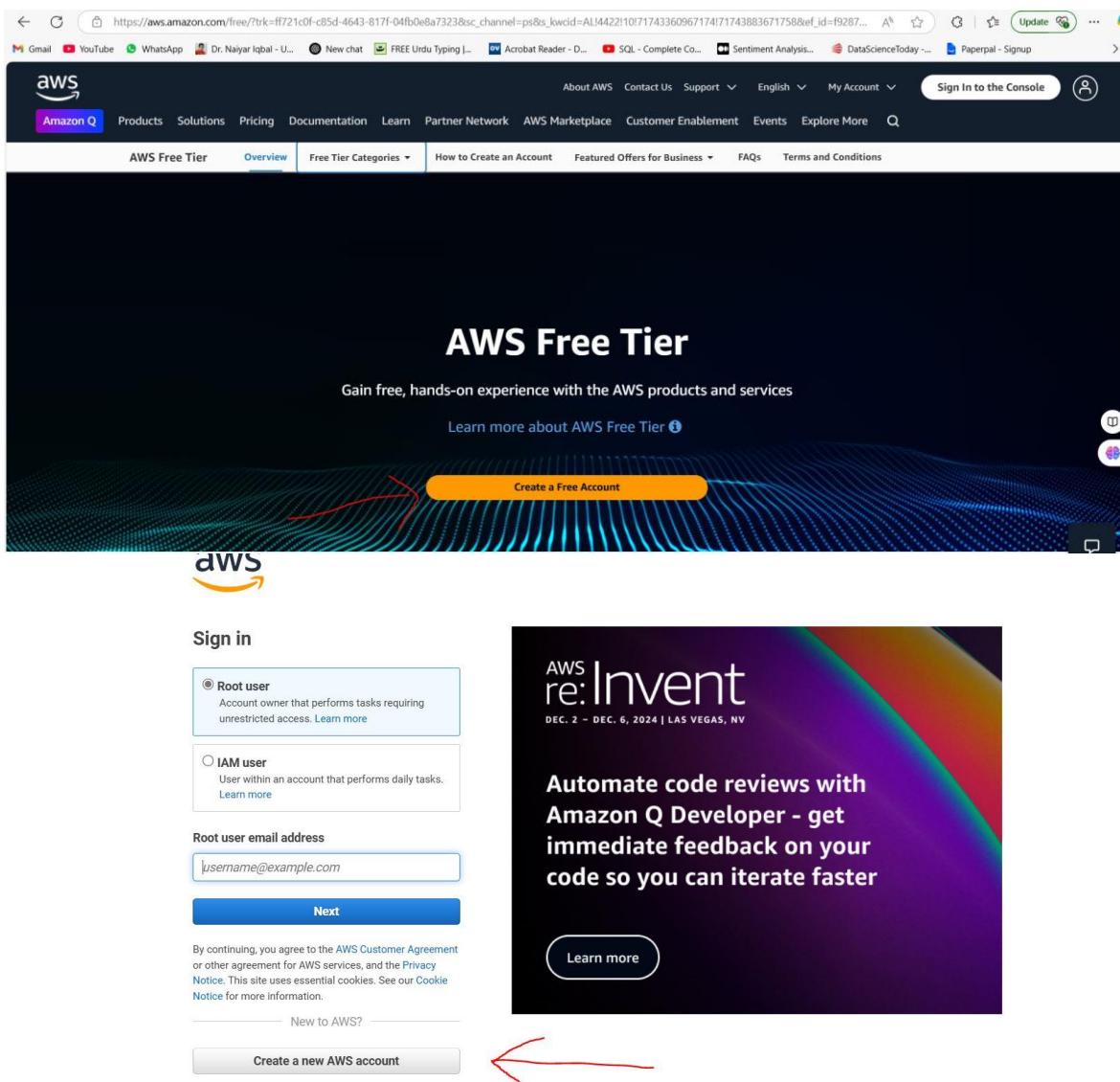
Note: Write the given steps in your lab manual. The provided image is just for your better understanding.

Aim: Jenkins Setup on AWS.

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)





Sign up for AWS

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

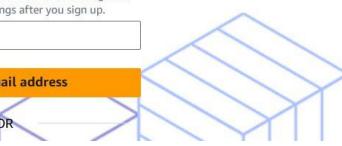


Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR



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](#)

The screenshot shows the AWS Console Home page. On the left, under 'Recently visited', the 'EC2' link is highlighted with a red arrow pointing towards it. On the right, there's a 'Launch instance' button in a box labeled 'Launch instance'. Other sections include 'Applications' (0) and 'View all services'.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main area has sections for 'Resources' (listing 0 instances, 0 auto scaling groups, etc.) and 'Launch instance' (with a 'Launch instance' button). Another 'Launch instance' button is located in the 'Service health' section. Other sections include 'Account attributes' (with a 'Default VPC' link), 'Explore AWS' (with a '10 Things You Can Do Today to Reduce AWS Costs' link), and 'Zones'.

Step-4: After Clicking on Launch Instance It will ask Instance Name and Other things -> Write Instance Name as Jenkins -> Select Application and IOS 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.

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

Jenkins

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

Recents

Quick Start

Amazon Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Linux



Debian



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

▼ 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](#)

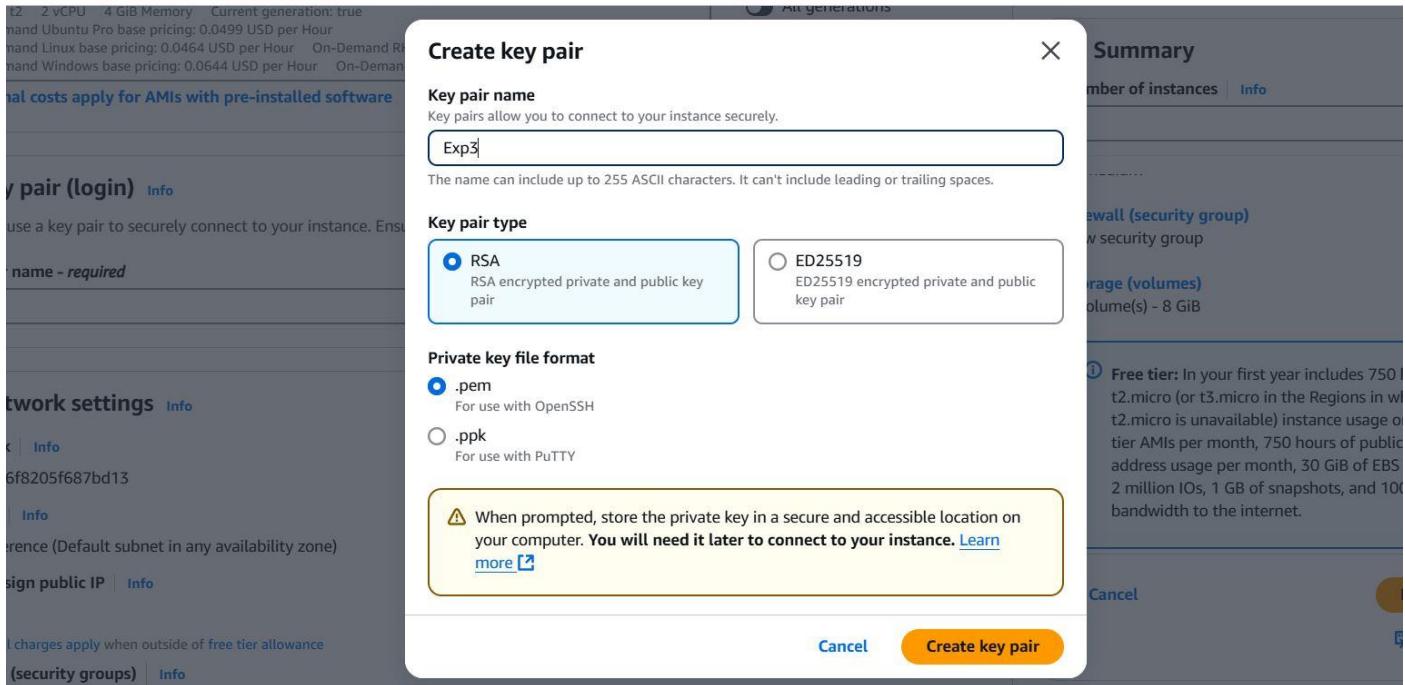
▼ Key pair (login) Info

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.

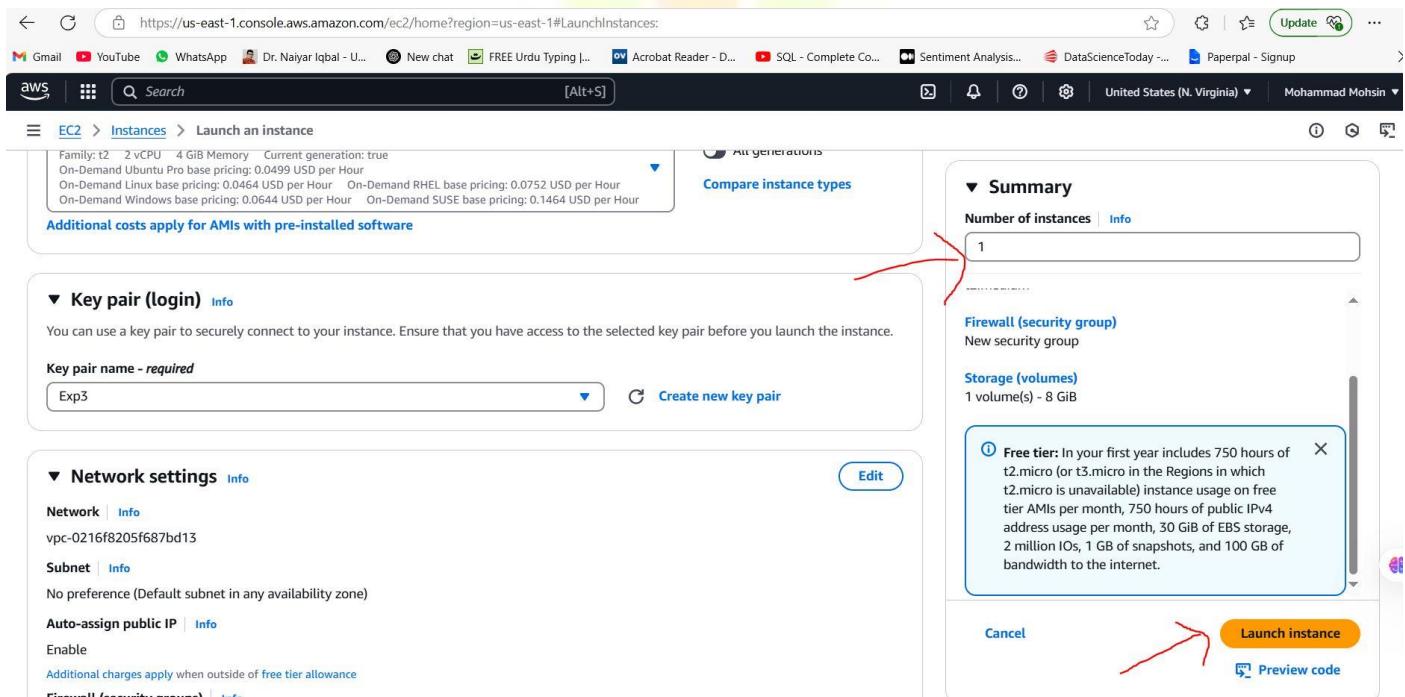
Key pair name - required

Select

Create new key pair



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 Resources page. On the left sidebar, under the 'Instances' section, there is a red arrow pointing to the 'Instances (running)' button, which is currently highlighted. The main content area displays a table of resources:

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

Below the table, there are sections for 'Launch instance' (with 'Launch instance' and 'Migrate a server' buttons) and 'Service health' (with 'AWS Health Dashboard' and status information for the US East (N. Virginia) Region).

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' button, which is currently highlighted. The main content area shows a table of instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Jenkins	i-09fa49fc85615f9d2	Running	t2.medium	Initializing	View alarms +	us-east-1b	ec2-34-123-45-67

Below the table, there is a 'Select an instance' section.

The screenshot shows the AWS EC2 Instances page, similar to the previous one, but with a red arrow pointing to the 'Security' tab in the navigation bar at the bottom of the instance details panel. The main content area shows the same instance details as before.

Step-7: After Clicking on Security -> 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

Instance state = running | Clear filters | All states

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)

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

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

Inbound rules

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

Add rule

Inbound rules [Info](#)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-075fd1ed42717eadd	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 X
-	HTTP	TCP	80	Anyw...	<input type="text"/> 0.0.0.0/0 X
-	Custom TCP	TCP	8080	Anyw...	<input type="text"/> 0.0.0.0/0 X

[Add rule](#)

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

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

Step-7: Now Search [Download MobaXterm](#) in your Browser-> Click on [Download](#) or First Link -> Click on [Home Edition](#) or Free Edition -> Click on [MobaXterm Portable Edition](#) -> It Will take few Minute and Download the File.

M Gmail YouTube WhatsApp Dr. Naiyer Iqbal - U... New chat FREE Urdu Typing |... Acrobat Reader - D... SQL - Complete Co... Sentiment Analysis... DataScienceToday -... Paperpal - Signup

 [MobaXterm](#) Home Demo Features [Download](#) Plugins Help Contact [Customer area](#) [Buy](#)

Home Edition

Free

- 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
- Max. 360 seconds for Tftp, Nfs and Cron

[Download now](#)

Professional Edition

\$69 / 49€ per user*

* Excluding tax. Volume discounts [available](#)

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
- Lifetime right to use

[Subscribe online / Get a quote](#)

MobaXterm Home Edition

Download MobaXterm Home Edition (current version):

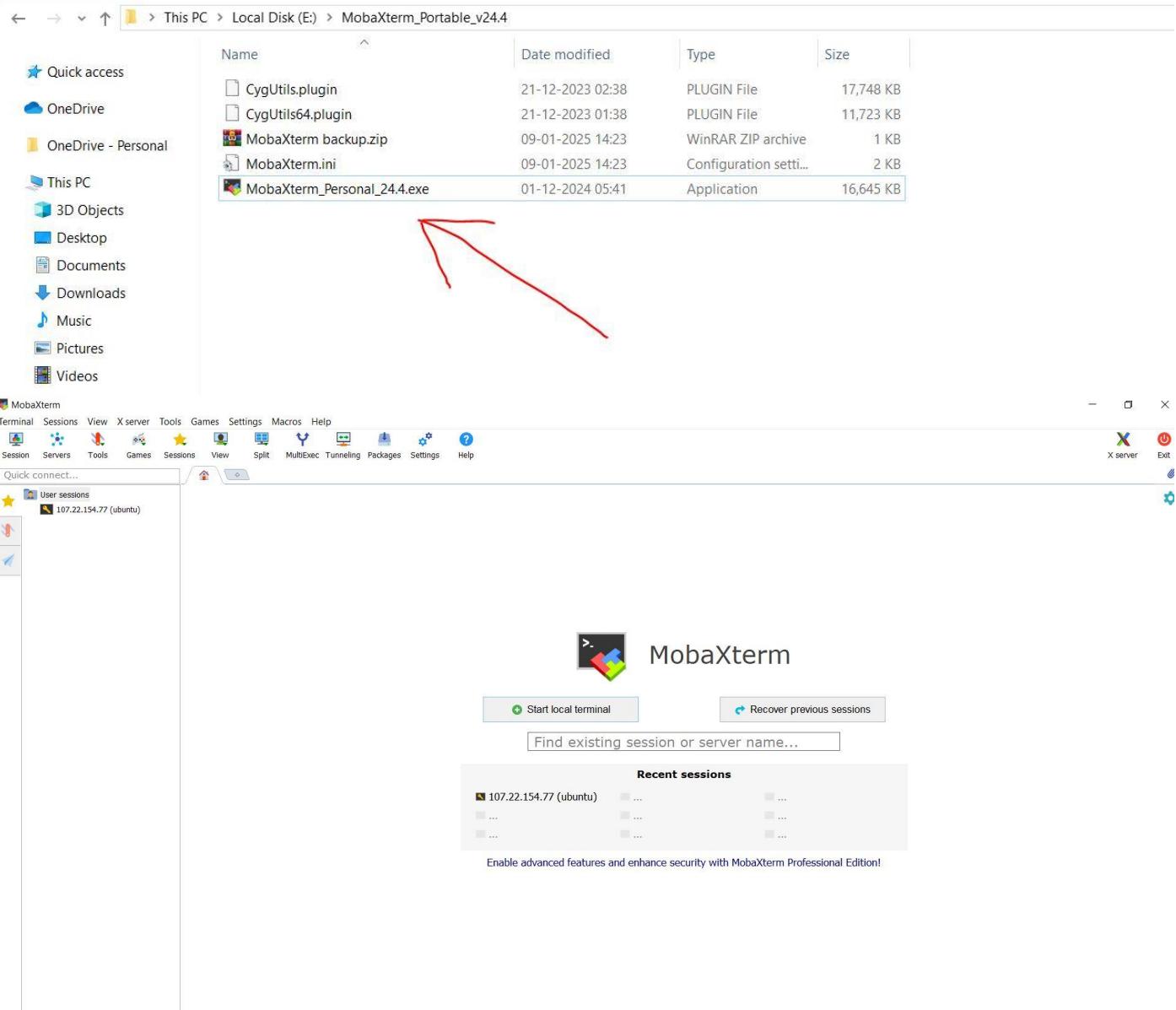

[MobaXterm Home Edition v25.0
\(Portable edition\)](#)


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

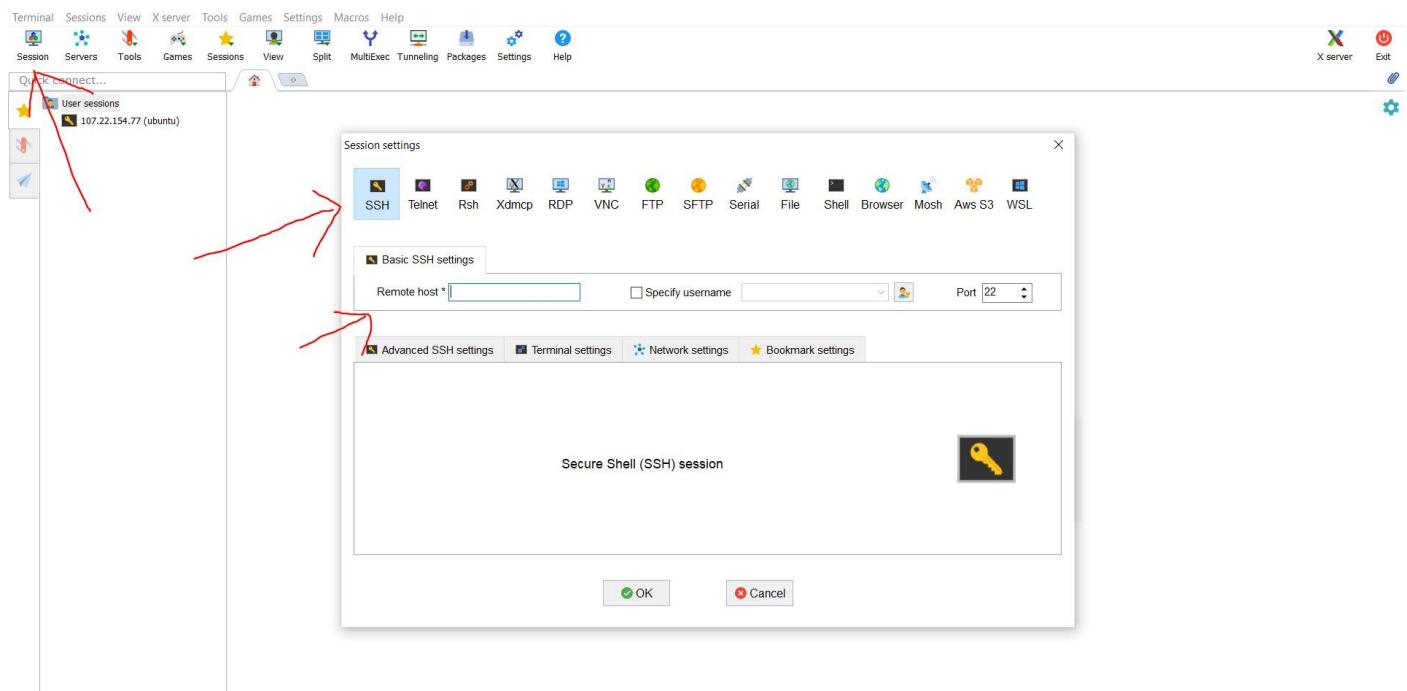
Download previous stable version: [MobaXterm Portable v24.4](#) [MobaXterm Installer v24.4](#)

Step-8: After Downloading the [MobaXterm.Zip](#) file right click and Extract all this file and Paste in your Devops Folder or Anywhere.

Step-9: Now Open [MobaXterm Extracted Folder](#) and double click on [.exe file](#) of application type file -> it will open [MobaXterm Software](#)

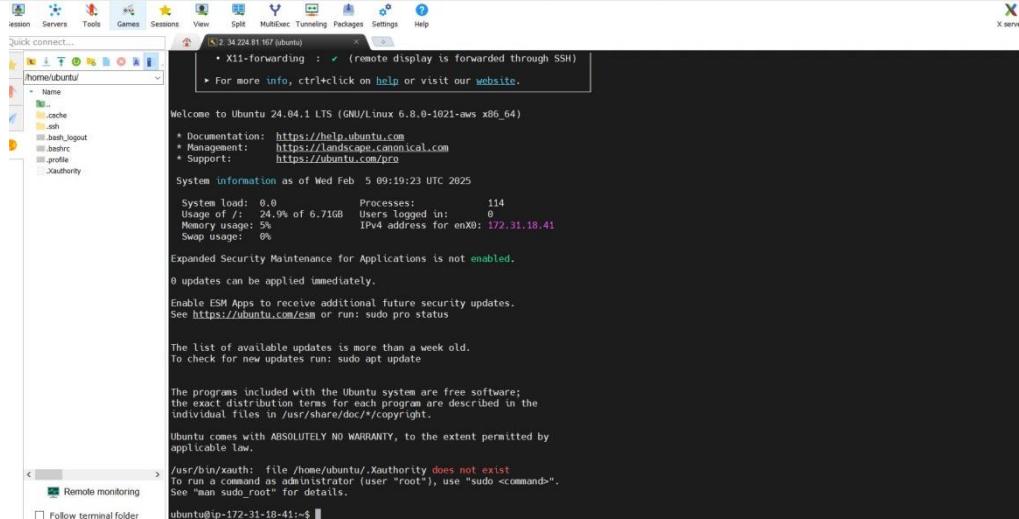
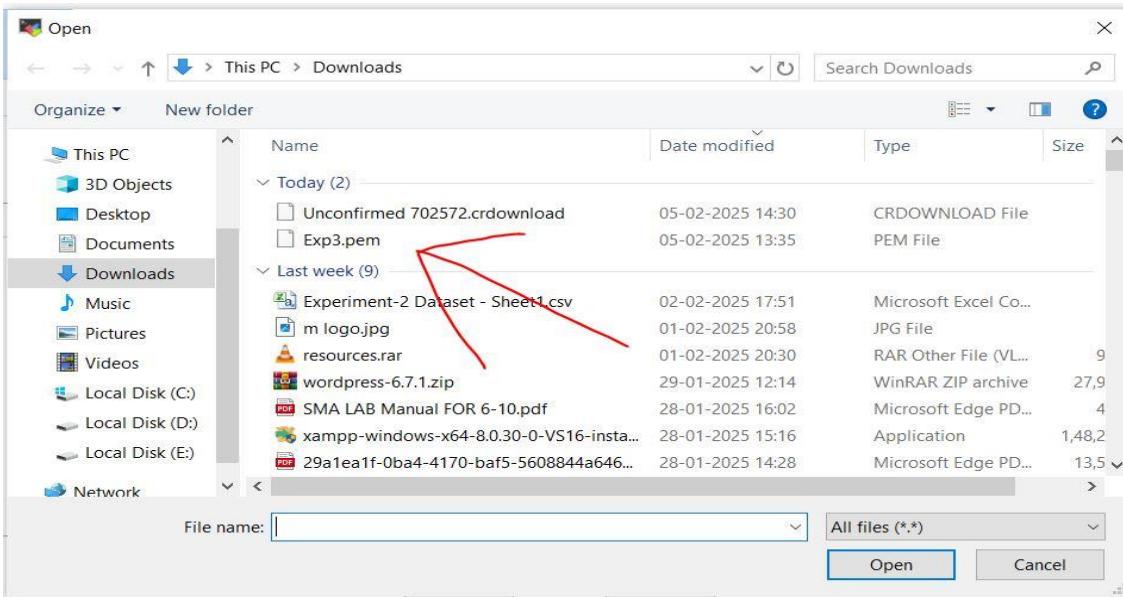
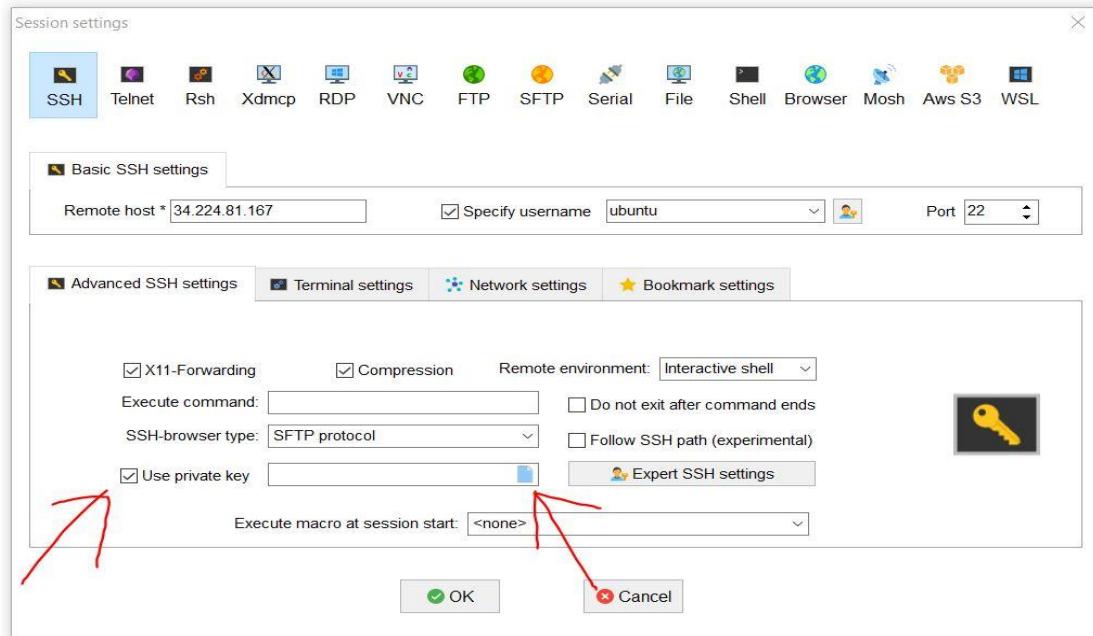


Step-10: Now Click on Session (Left upper Corner) -> Click on SSH -> it will ask remote host and specify username.



Step-11: In the Place of Remote Host Paste the **Public IP of your Instance** (Go to Instance Details and Check your Public IPv4) -> Click on **Specify Username** and Write **Ubuntu** -> Click on **Advance SSH Setting** -> Select **Use Private Key** -> Browse the **.pem file** That You already Downloaded during Instance Creation -> After that click on **OK** -> It will open a **Linux Command Prompt**

The screenshot shows the AWS CloudWatch Instances console. At the top, there's a search bar and filters for 'Name' and 'Instance ID'. Below, a table lists one instance: 'Jenkins' (Instance ID: i-09fa49fc85615f9d2). The instance is 'Running' (Status check: 2/2 checks passed) and has an 't2.medium' instance type. It's located in 'us-east-1b' availability zone and has a 'Public IP' of 34.224.81.167. A red arrow points from the instance ID 'i-09fa49fc85615f9d2 (Jenkins)' to the 'Details' tab at the bottom. The 'Details' tab is active, showing the instance summary. The summary includes the instance ID, public and private IPv4 addresses, and the public IPv4 DNS name (ec2-34-224-81-167.compute-1.amazonaws.com). The instance state is listed as 'Running'.



Step-12: 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 the Microsoft Bing search results page. The search query 'jenkins Documents' is entered in the search bar. The top result is 'Jenkins User Documentation' from <https://www.jenkins.io/doc>. A red arrow points to this link. Below it, there are other links: 'Installing Jenkins', 'Blue Ocean', and a 'Jenkins - Wikipedia' entry.

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

Blue Ocean

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...

See more on Wikipedia

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](#) (develc 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, testi 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 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.

[Index](#)

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](#)

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](#).

Chapter Sub-Sections
Docker
Kubernetes
Linux
macOS
Windows

Table of Contents
Prerequisites
Debian/Ubuntu
Long Term Support release
Weekly release
Installation of Java

Step-13: 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:

```
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
```

Step-14: Clear your Screen -> Now Update your Project Using Given Code.

Code:

```
sudo apt-get update
```

Step-15: clear your screen -> Now Check Java version Using Command: **java --version**

Step-16: 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:

```
sudo apt install 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 libxaw7 libxcb-dri2-0 libxcb-dr
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libcomposite1 libcursor1 libxdamage1 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxtst6t64 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-17: 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-18: Now [Enable the Jenkins Server](#) Using Given Command

Command: `sudo systemctl enable jenkins`

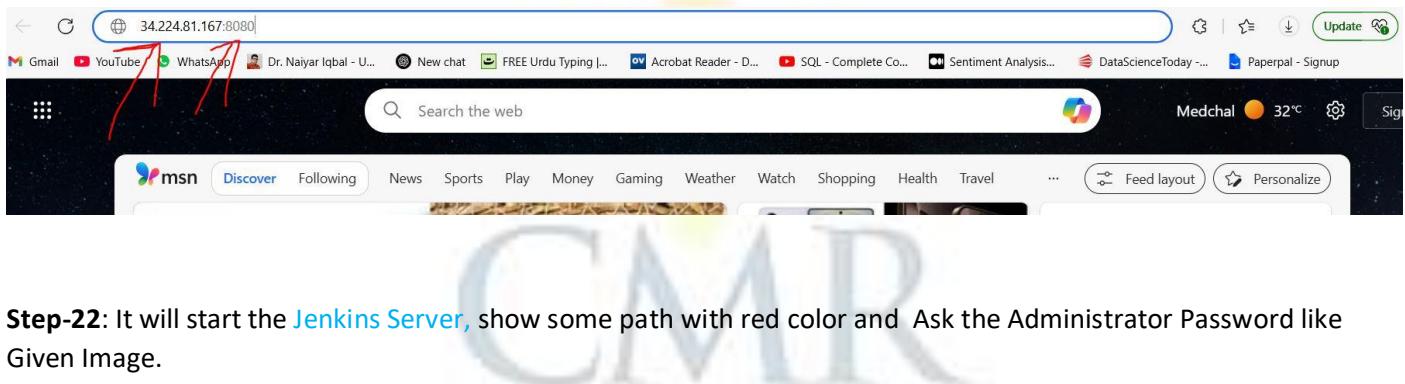
Step-19: After [Enable the Jenkins Server](#) Start [Jenkins Server](#) Using Given Command

Command: `sudo systemctl start jenkins`

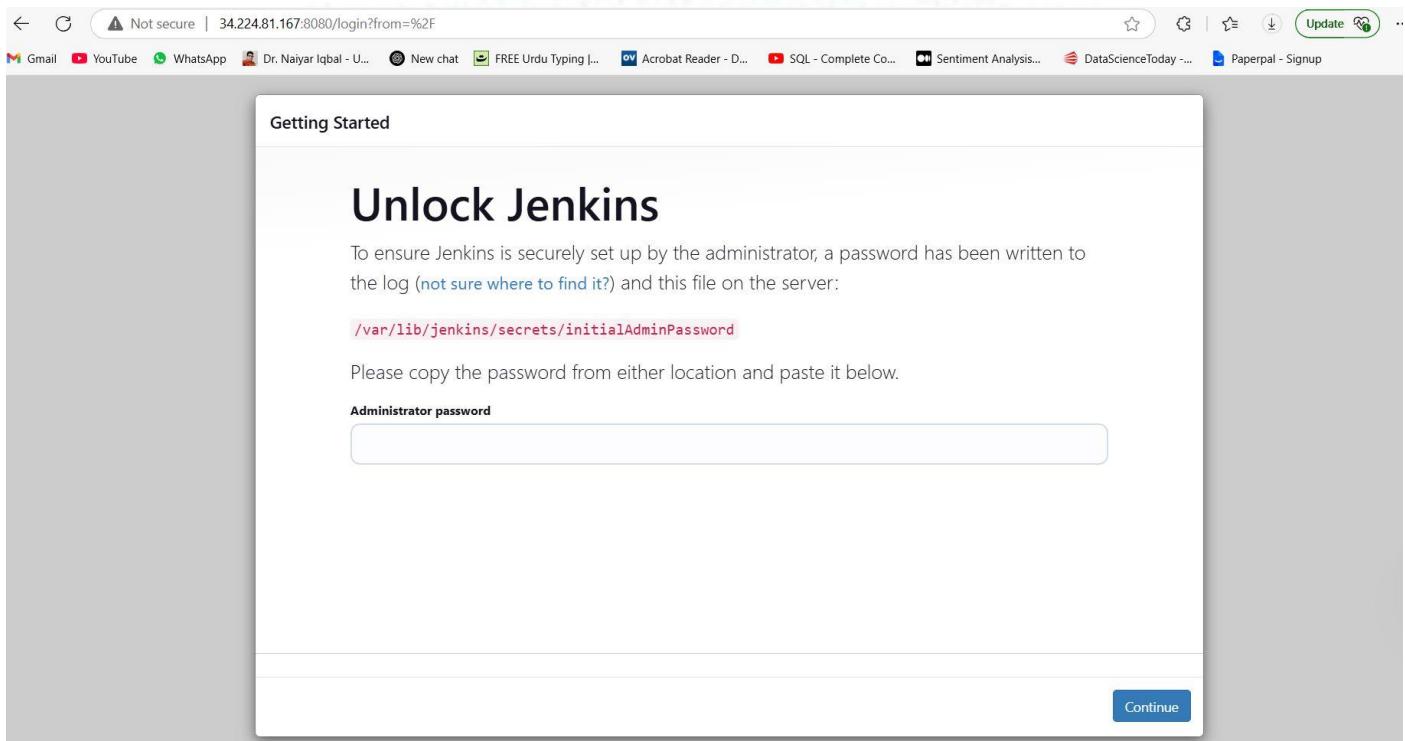
Step-20: After [Starting the Jenkins Server](#) Check [Jenkins Server](#) Status using Given Command. It will show the Status Active running ans show the status of memory, CPU etc.

Command: `sudo systemctl status jenkins`

Step-21: 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



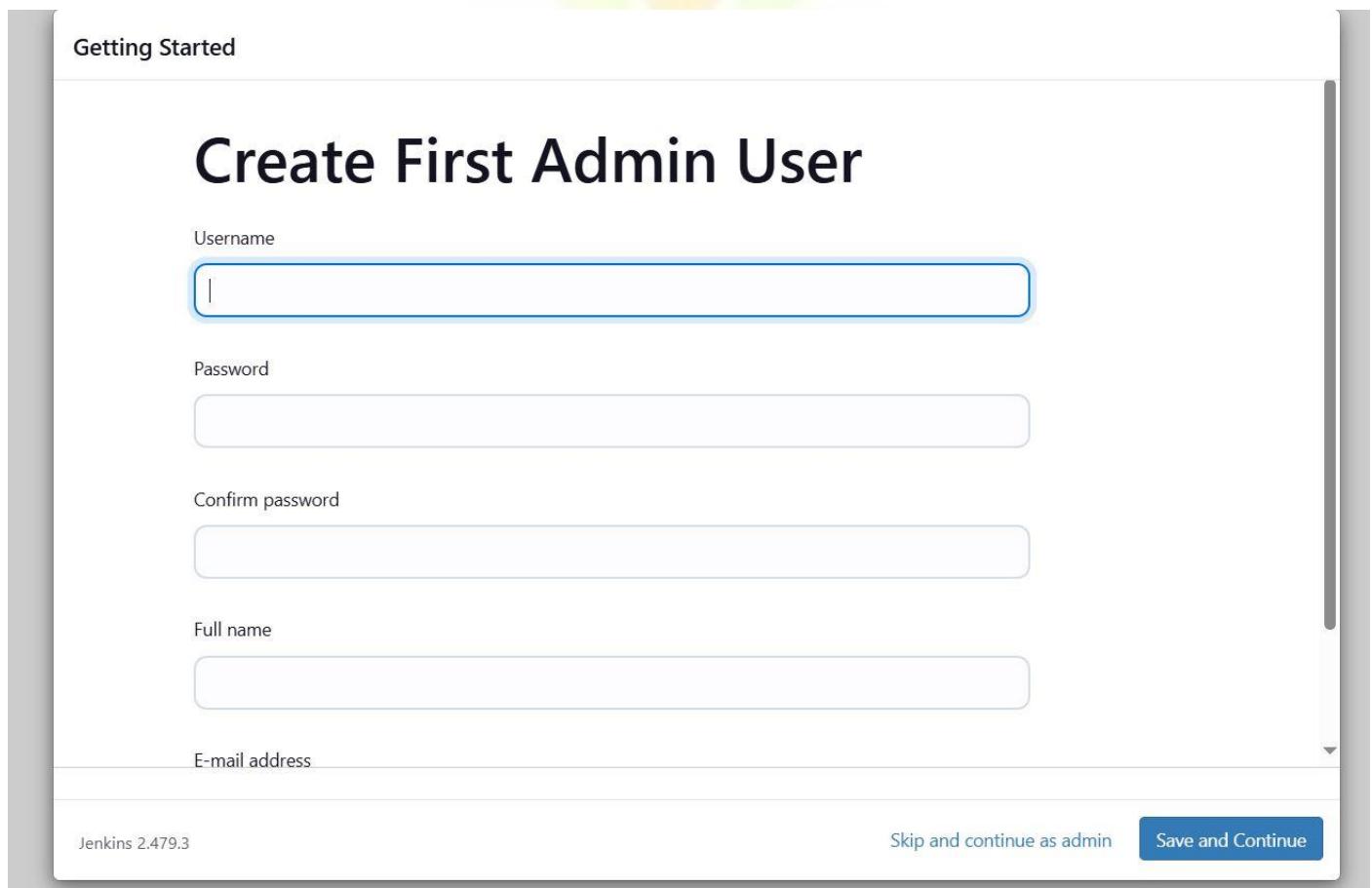
Step-22: It will start the [Jenkins Server](#), show some path with red color and Ask the Administrator Password like Given Image.



Step-23: 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  
  
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-24: 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-24: 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.

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

OUTPUT

Your Output is the Jenkins Dashboard

The screenshot shows the Jenkins dashboard interface. At the top, there is a navigation bar with various icons and links. Below the navigation bar, the main header reads "Welcome to Jenkins!". A sub-header below it says "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." There are several buttons and sections: "Create a job" (with a plus sign), "Set up a distributed build" (with three sub-options: "Set up an agent", "Configure a cloud", and "Learn more about distributed builds"), and "Build Queue" (which is currently empty). On the right side, there is a user profile for "Mohammad Mohsin" and a "log out" link. The overall layout is clean and modern, designed for easy navigation and setup.