# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

**(AN AUTONOMOUS INSTITUTION)**

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


## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## ((Artificial Intelligence and Machine Learning)

# LAB RECORD

# SOFTWARE ENGINEERING LAB

**B.Tech. III YEAR I SEM (RKR21)**

**ACADEMIC YEAR 2024-25**

**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

**(AN AUTONOMOUS INSTITUTE)**

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

# Certificate

This is to certify that following is a Bonafide Record of the workbook task done by

_____bearing Roll No_____ of _____

Branch of _____ year B. Tech Course in the_____

Subject during the Academic year _____ & _____ under our supervision.

Number of week tasks completed: _____

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner

# KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

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

## Daily Laboratory Assessment Sheet

Name of the Lab: Name of the Student:

Class: HT. No:

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

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

# INDEX

| Exp No: | List of Experiments | PAGE NO |
|---|---|---|
| 8 | A. AWS ACADEMY LEARNING ACCOUNT CREATION<br>B. PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE<br>C. MAVEN WEB PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE | |

**Vision of the Institution:**

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

**Mission of the Institution:**

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

## Department of Computer Science & Engineering (AI & ML)

**Vision of the Department**:

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

**Mission of the Department**:

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

**PROGRAM OUTCOMES (POs)**

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

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

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

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

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

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

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

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

**PO9: Individual and Team Work**: Function effectively as an individual, and as a member or

leader in diverse teams, and in multidisciplinary settings.

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

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

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

## PROGRAM SPECIFIC OUTCOMES (PSOs)

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

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

## PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

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

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

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

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

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

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 3 | 1.5 |

**Pre-requisites/ Co-requisites:**

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

**Course Objectives: The course will help to**

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

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

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

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

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

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Study and usage of any Design phase CASE tool
4. Creating the project and committing using Git and GitHub
5. Creating Maven Java and Maven Web project using Eclipse and Push them to GitHub.
6. Building the CI/CD pipeline using Jenkins for the project in the previous experiment.
7. Local Deployment of project using Docker, Kubernetes and Monitoring using Nagios tool.
8. Cloud Deployment of a project in the AWS Cloud using EC2 instance.

**Sample Projects:**

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

**TEXT BOOKS:**


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


**REFERENCE BOOKS:**


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

## Course Outcomes and CO-PO-PSO Mapping

### Course Outcomes:

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

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

## CO-PO-PSO MAPPING:

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

## Software and Hardware Requirements

### 1. Software Tools

**a. Star UML**
**b. Java 11/17/21 Version**
**c. Apache Tomcat 9**
**d. Eclipse IDE**
**e. Visual Studio**
**f. Jenkins**
**g. Git Bash for Windows**
**h. Windows Docker Desktop**
**i. AWS Cloud account (Basic Plan)**

### 3. Hardware Requirements

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

# Experiment-1: Installation of StarUML, Git Bash and GitHub Account Creation

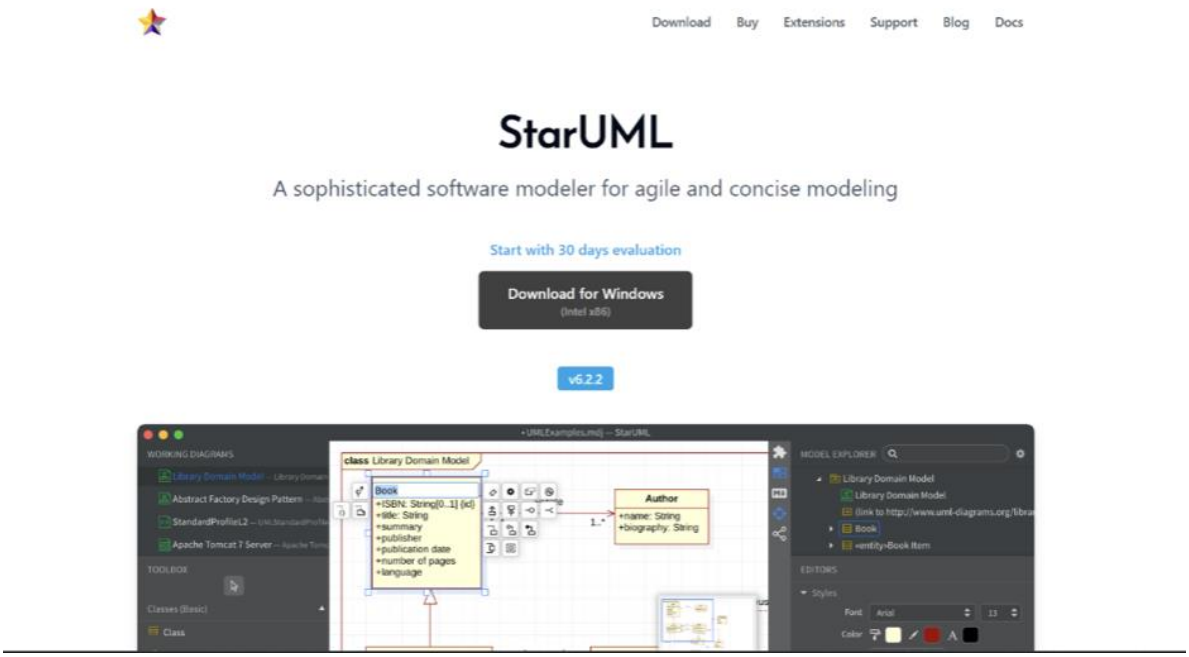
**Aim**: To install StarUML, Git Bash, and create a GitHub account.

## Introduction:

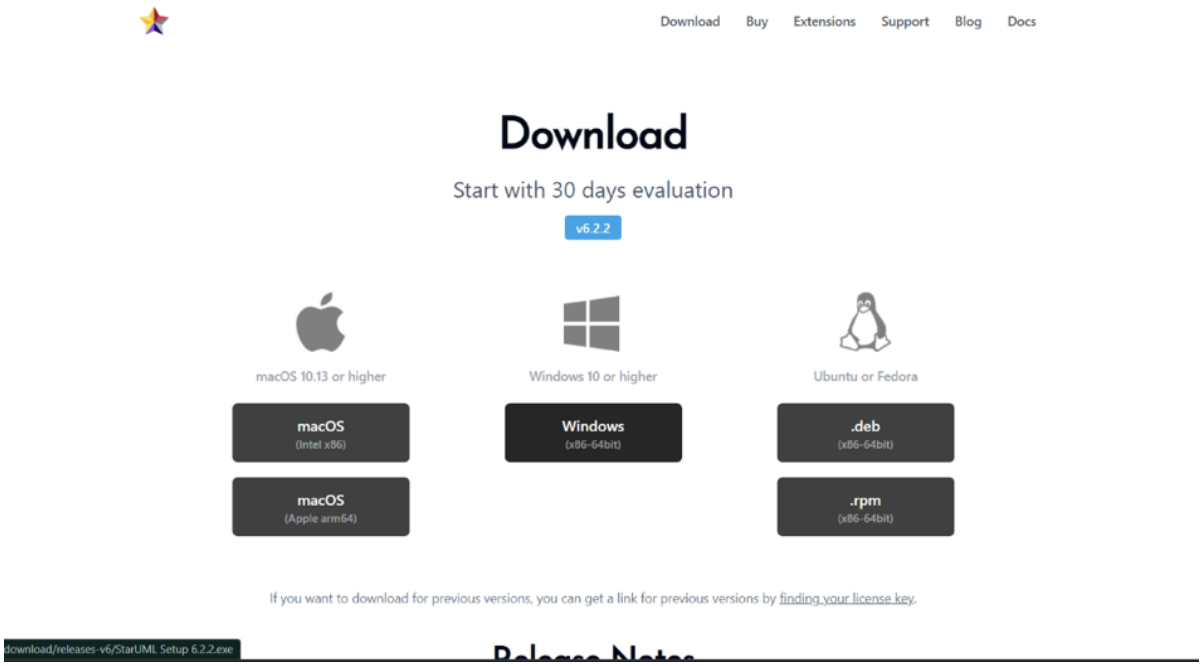
1. **StarUML:** StarUML is a sophisticated software modeling tool that supports UML (Unified Modeling Language) for designing and describing software architecture and systems.
2. **UML Diagrams:** UML diagrams are graphical representations used to visualize, specify, construct, and document the structure and behavior of a software system.
3. **Git Bash and GitHub:**
    - Git: Git is a popular version control system created by Linus Torvalds in 2005, designed for managing multiple versions of a code base across multiple developers or teams. Functions of Git include:
        - Managing projects with repositories
        - Cloning projects for local work
        - Controlling and tracking changes with staging and committing
        - Branching and merging for different parts and versions of a project
        - Pulling the latest project version locally
        - Pushing local updates to the main project
    - GitHub: An online platform for hosting and collaborating on Git repositories.

## Procedure:

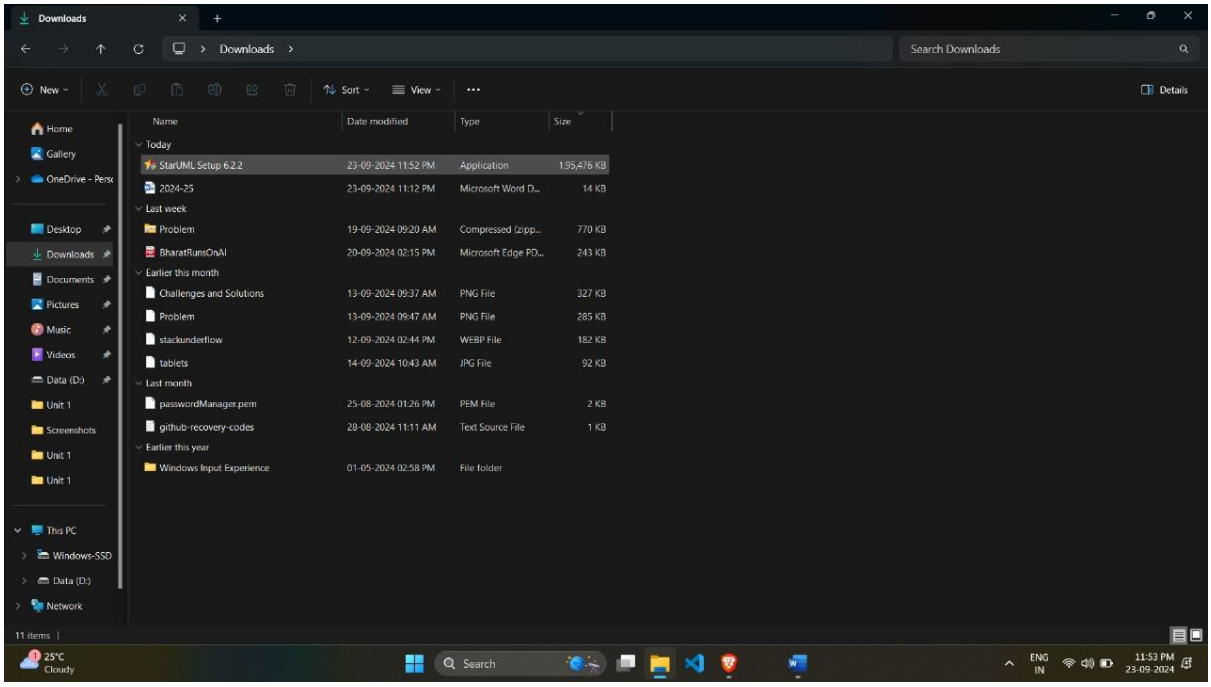
1. Download and Install StarUML from staruml.io website.



2. Click on download for windows button and select based on your system architecture.



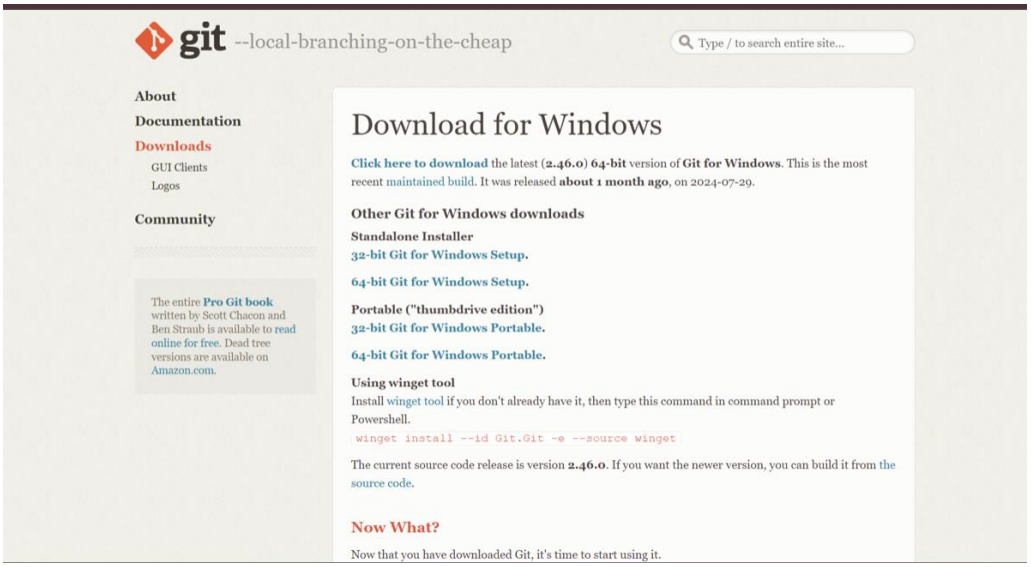

3. Install the StarUML software which will be present in the downloads section .



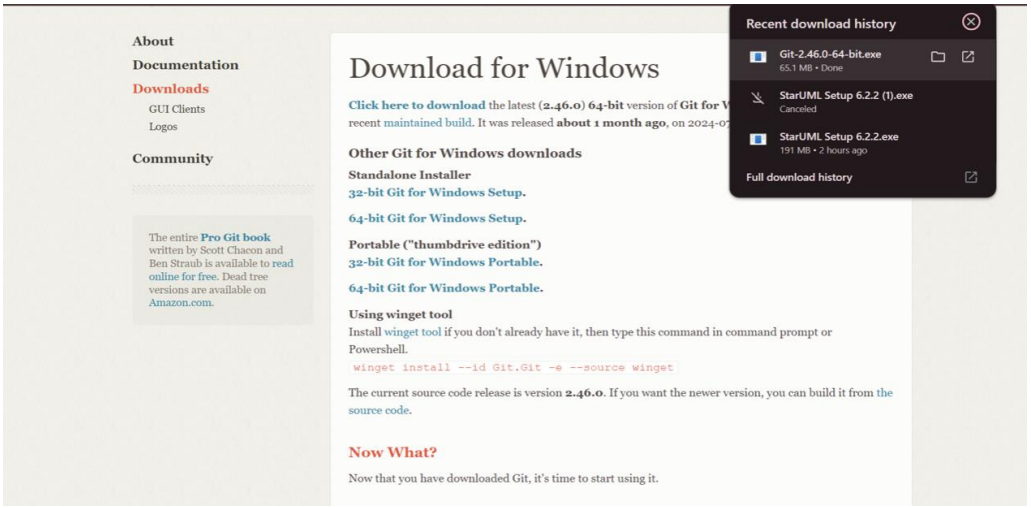
4. Explore available features


# Installation of Git

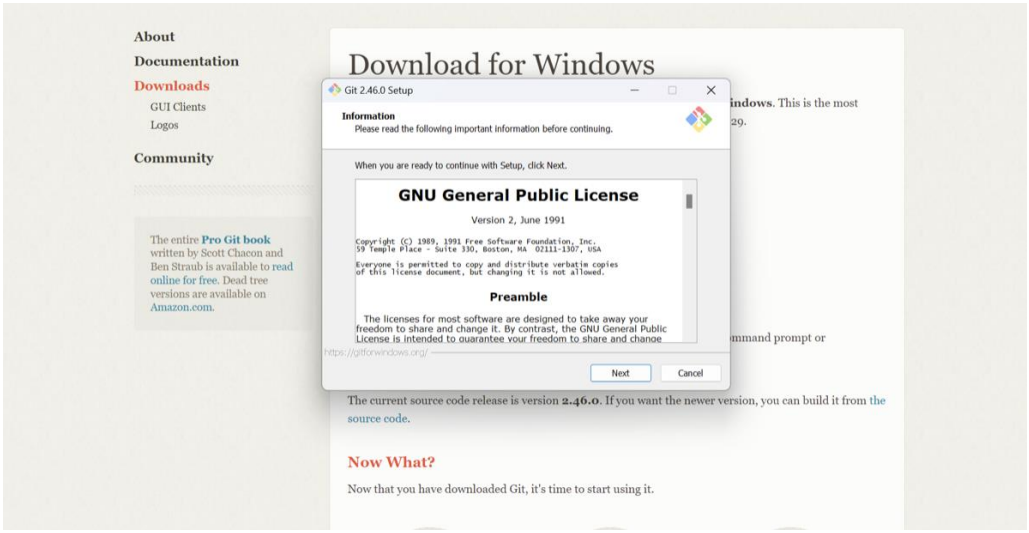
1. Go to the Download section of https://git-scm.com/



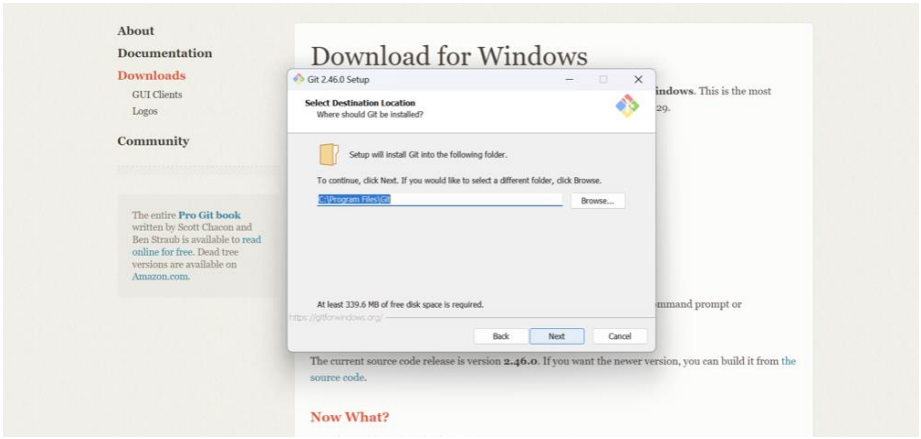

2.Select the latest version of Git software for the required OS and download it.



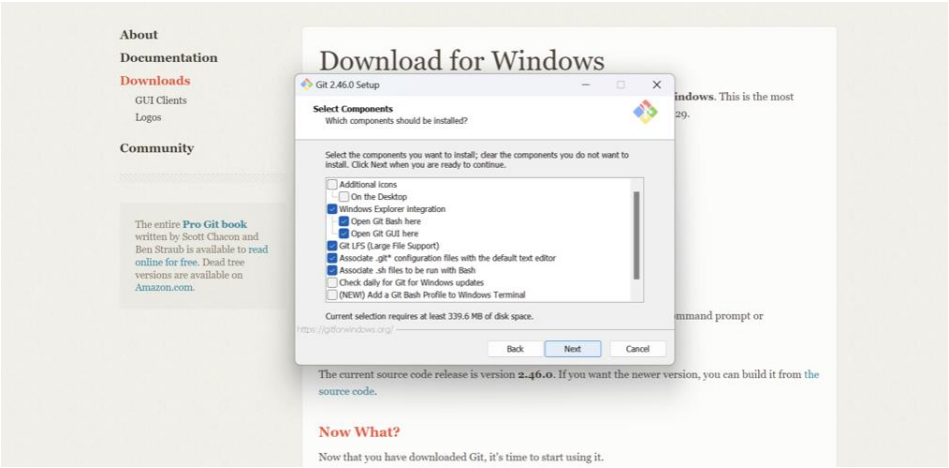

3.Install the downloaded Git software. Click next



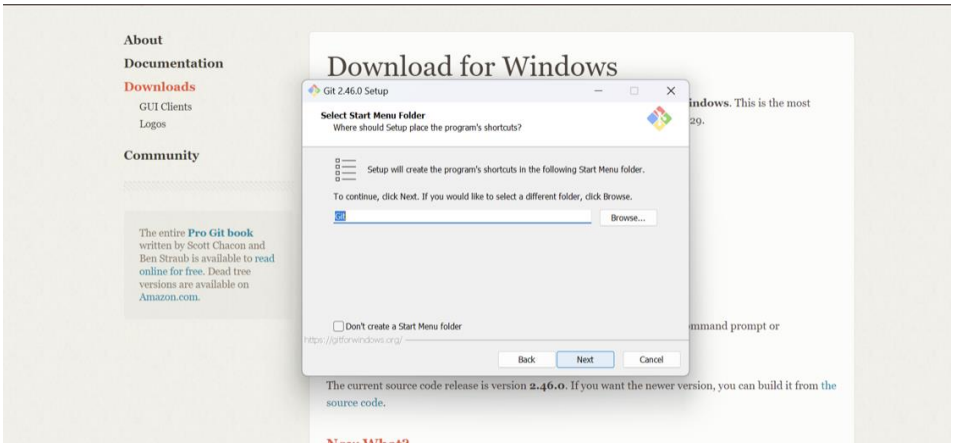

4.Select the destination location for the installation.



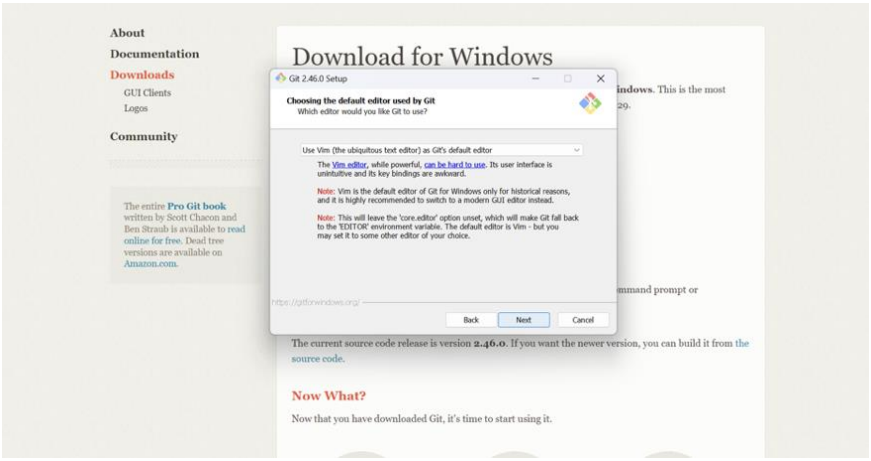

5.Select the required components and click next.



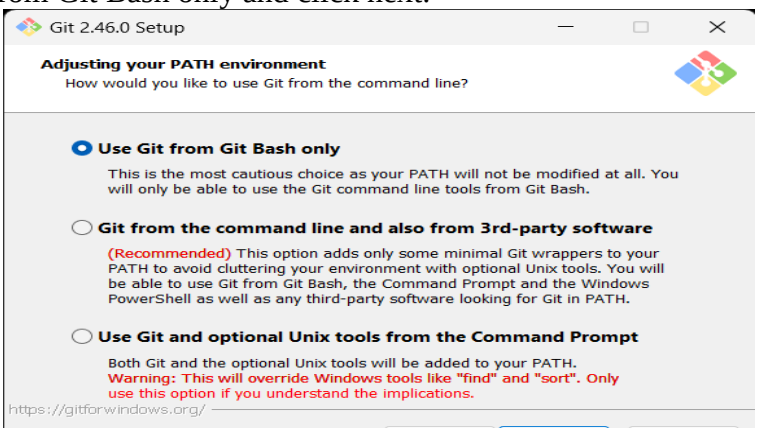
6.Select the start menu folder and click next.



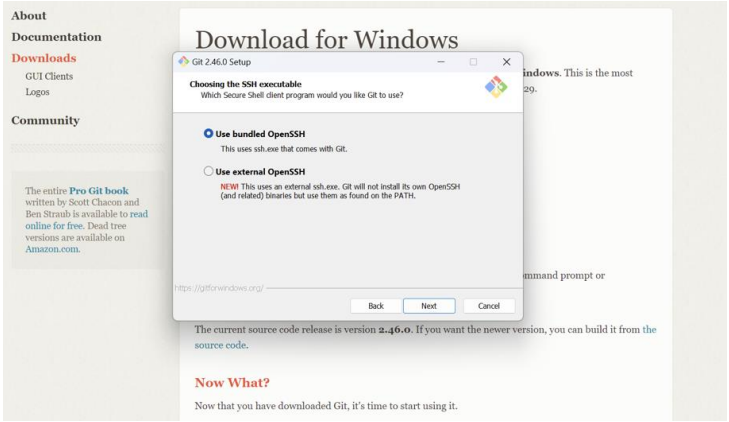

7.Choose the default editor and click next.



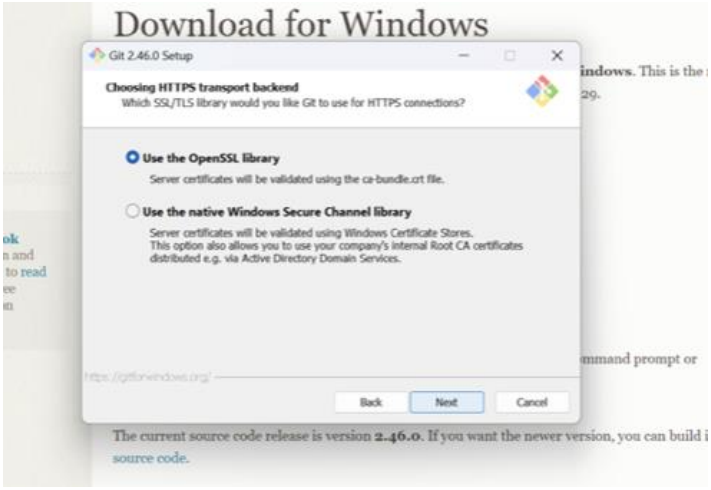

8.Select use Git from Git Bash only and click next.



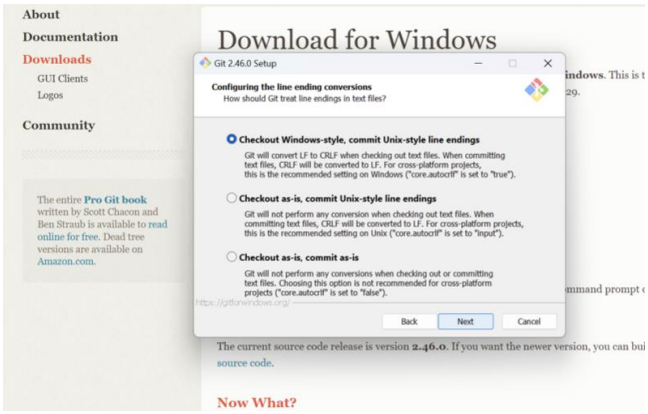
9.Choose the SSH executable and click next.




9.Choose the HTTPS transport library as OpenSSL library. Click next.

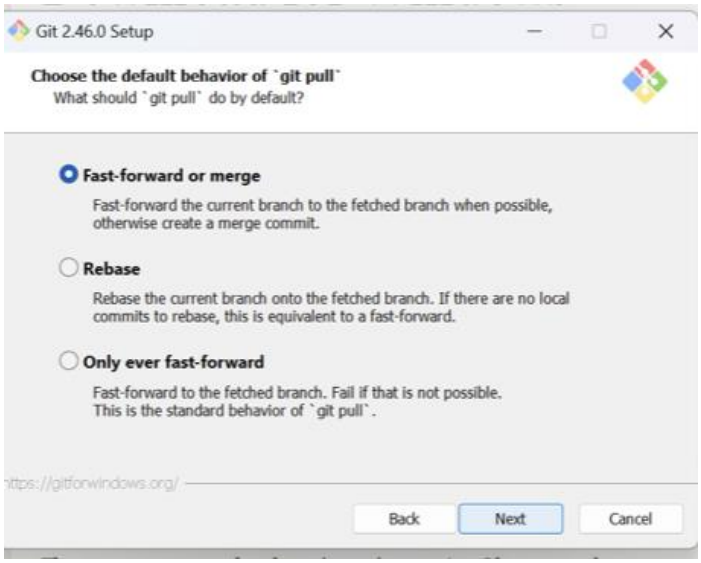



10.Configure the line ending conversions and Click next to proceed.

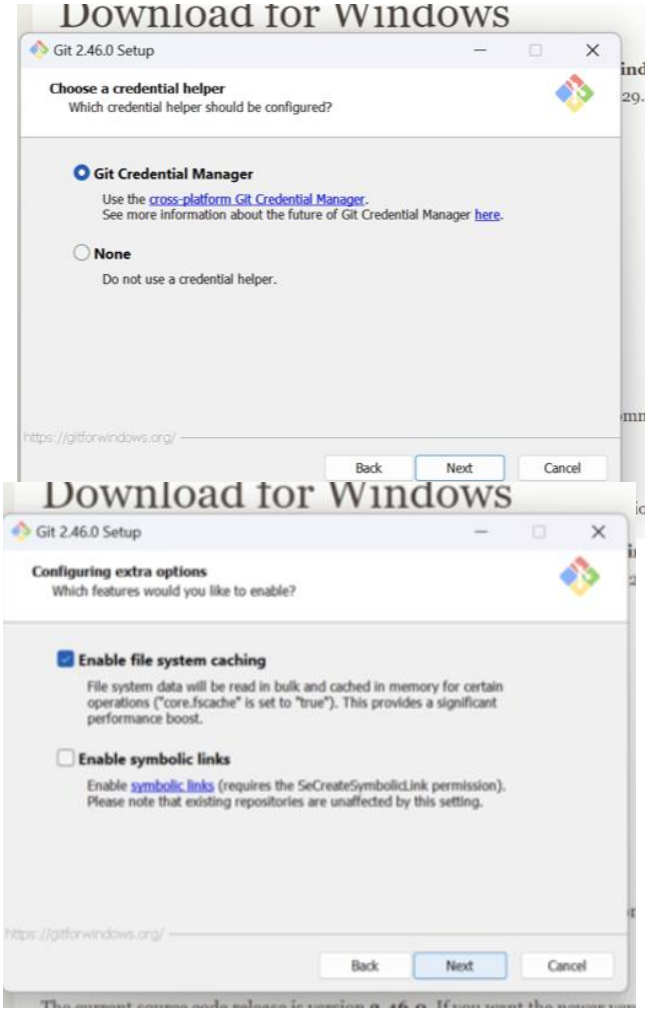


11.Select the terminal emulator to use with Git Bash and Click next.


12.Choose the default actions of git pull. Click next.

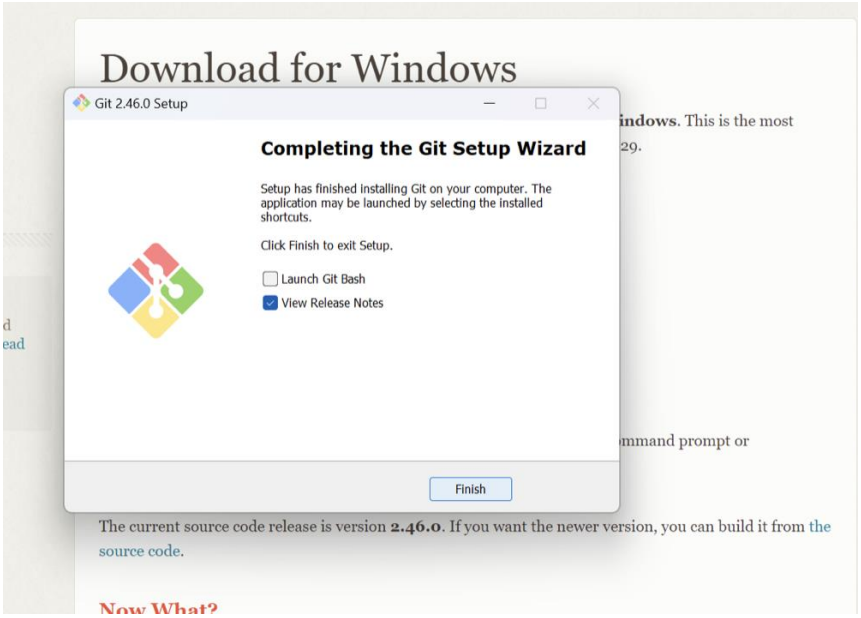



13.Select the Git credential manager. Click next.

14.Enable  file system caching and click next.

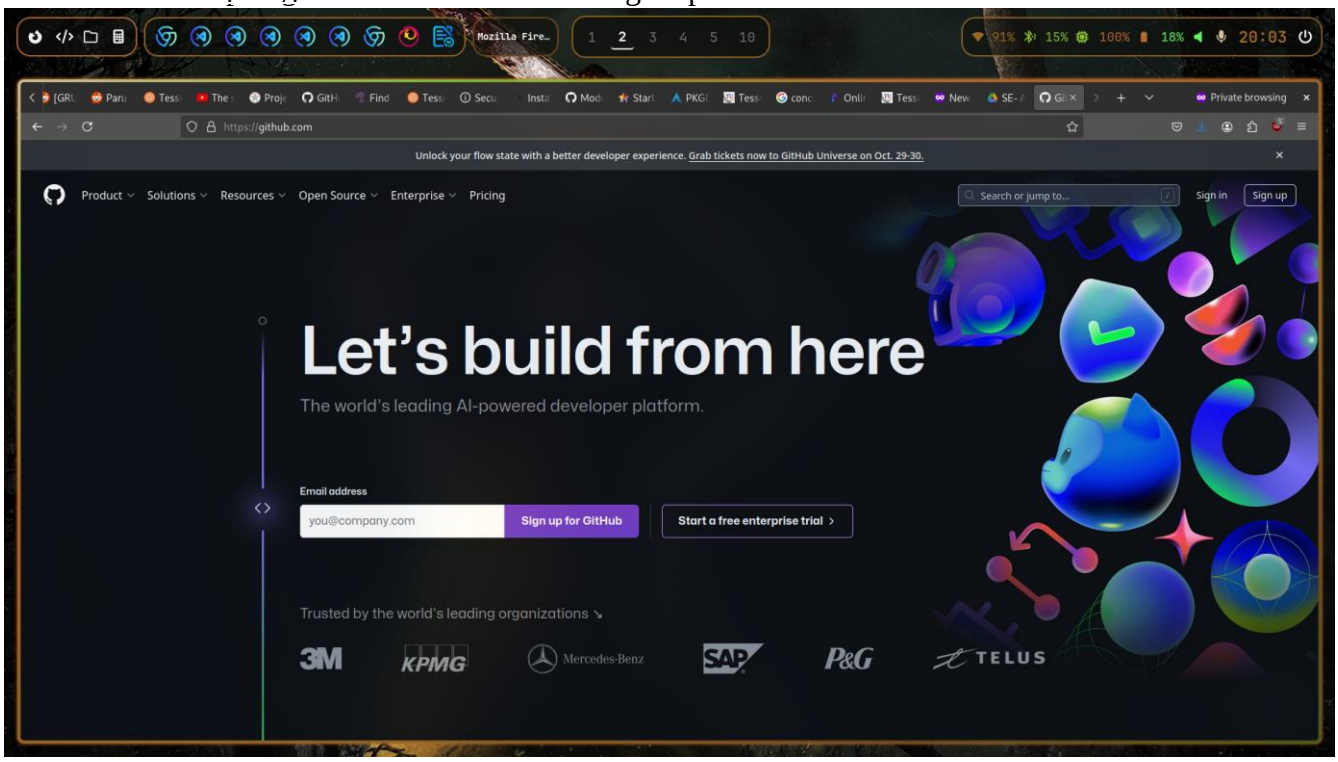


15.Click on install. And Click on finish to complete the installation process.

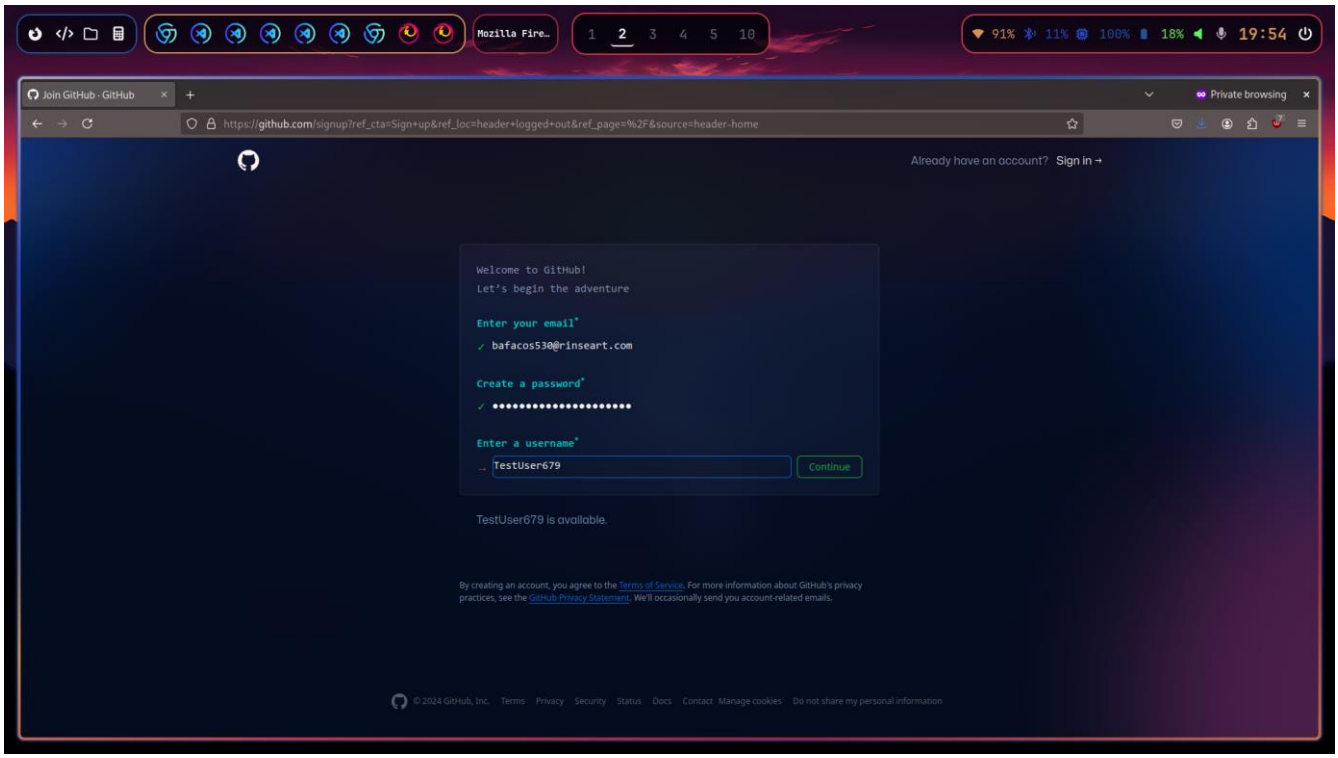



# Creating GitHub account

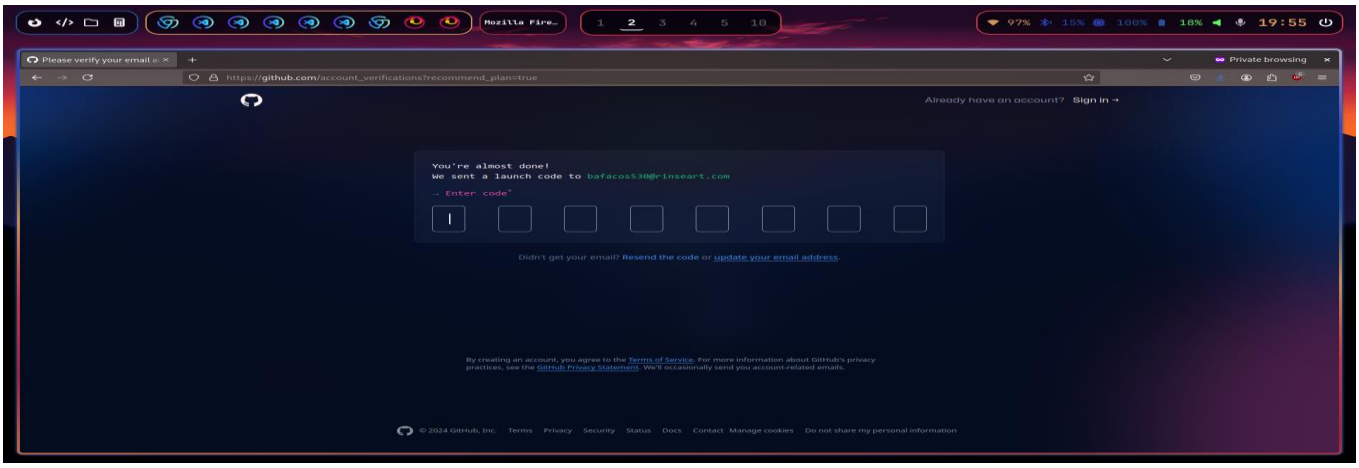
1. Go to https://github.com and click on Sign Up



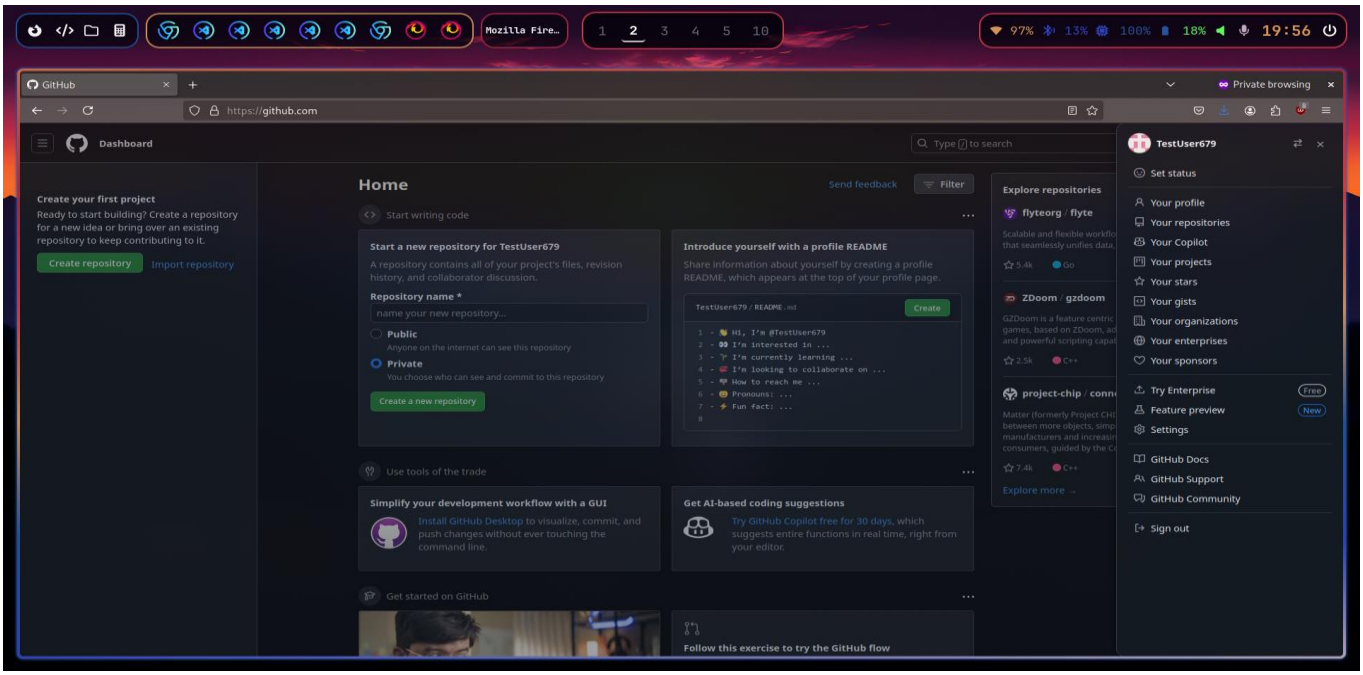

2. Enter your Email address and create a password




3. Click continue, then enter the verification code sent to your email




4. You are now logged in to GitHub




Result: Installed Star UML and explored it's functionality, installed Git SCM and created a GitHub account.

# Experiment-2: Preparation of Software Requirements Specification Document

Aim: To study and understand the components and importance of a Software Requirements Specification (SRS).

Introduction:

1. SRS: A Software Requirements Specification is a document that comprehensively describes the functional and non-functional requirements of a software system. It serves as a formal agreement between stakeholders and developers, outlining what the software must achieve.
2. Purpose of SRS:
    - Ensures a clear understanding of project goals and requirements.
    - Acts as a reference throughout the development lifecycle.
    - Facilitates communication among stakeholders, developers, and testers.
3. Key Components of SRS:
    - Functional Requirements: Define specific behaviors or functions of the software.
    - Non-Functional Requirements: Include performance, security, and usability considerations.
    - System Constraints: Highlight limitations or dependencies of the project.

# Software Requirements Specification For E-Ticketing

Version 3.0 approved


**Prepared by:**

1. Aditya - 22BD1A6602
2. Harsha - 22BD1A660J
3. Armaan - 22BD1A6611
4. Anand - 22BD1A6615
5. Varshith - 22BD1A6618

**Keshav Memorial Institute of Technology**


**11-10-2024**.

# Table of Contents

# Revision History

| Name | Date | Reason for changes | Version |
|---|---|---|---|
| Week-1 | 11-10-24 | SRS Template | 1.0 |
| Week-2 | 17-10-24 | SRS ( DIAGRAMS ) | 2.0 |
| Week-3 | 25-10-2024 | SRS Final document | 3.0 |

# 1. Introduction

The Software Requirement Specification is designed to document and describe the agreement between the customer and developer regarding the specification of the software product requested.

This documentation is done to provide a clear idea of customer requirements. This document can be used as reference in further development of the software system.

## 1.1 Purpose:

This is the Software Requirements Specification (SRS) for our **E-TICKETING** platform. The project focuses on simplifying ticket booking for events, travel, and other activities through an online system. It will allow users to purchase tickets conveniently without the need to visit physical counters. We will also provide event organizers and service providers with the ability to upload and manage their events or transport schedules for sale. The platform aims to streamline the entire process between the ticket buyer and the provider, making the transaction quick and easy.

## 1.2 Document Convention:

- **Heading**:
  - Font-Size:16
  - Font-Style: Bold
  - Font: Times New Roman

- **Subheading**:
  - Font-Size:14
  - Font-Style: Bold
  - Font: Times New Roman

- **Content**:
  - Font-Size:12
  - Font: Times New Roman.

## 1.3 Intended Audience and Reading Suggestions:

This document serves as a prototype for the **E-TICKETING** platform. It is designed for various audiences with different purposes in mind. Developers can use this document to design and implement the project; managers can use it for planning, managing costs, and tracking timelines. Event organizers and service providers can use it to understand the system's features, while users can assess if the platform meets their needs and suggest any improvements. Testers can refer to this document for testing functionalities. The functional and non-functional requirements outlined here are also beneficial for developers during the development process.

## 1.4 Product Scope:

Our project is similar to existing e-ticketing systems but introduces several new features and improvements. The platform will be available 24/7, with periodic maintenance scheduled once a month. The primary goal is to enhance user convenience by allowing them to book tickets from anywhere, saving significant time compared to traditional ticketing methods. Service providers won't need to maintain physical stock and can manage bookings digitally. Our system will provide users with ticket recommendations based on their preferences and needs. Users will also have the ability to rate event organizers or service providers based on the quality of the service, enabling better customer experience and feedback management. User data will be secured against breaches and unauthorized access. Regular surveys will be conducted to gather user feedback and continuously improve the platform. A robust database will be maintained for both users and service providers.

User details will be kept safe and secure from data theft. We conduct surveys on how we can make our site better and what changes can be made. We also maintain a database for different users.

## 1.5 References:

References were taken from various ticketing systems and platforms, including industry standards and best practices from ticketing websites like Ticketmaster, Eventbrite, IRCTC, BookMyShow, and other similar platforms.

# 2. Overall Description

## 2.1 Product Perspective

Our project is an evolution of existing **e-ticketing** systems. It will include key functionalities found in ticketing platforms, such as allowing event organizers or service providers to set up digital listings, customers to browse through available events or services, and administrators to manage event categories, accept or reject service providers, and maintain platform policies. The system will be **mobile-friendly** with highly customizable themes. Additionally, it incorporates several new advancements such as an **advanced search engine** for easy ticket search, **high data security** to protect user information from breaches, and a **user-friendly interface** that ensures a smooth experience for users of all types.

## 2.2 Product functions

### 2.2.1 Administrator

- Administrators will have the ability to insert, modify, and delete event listings or service offerings.
- Can accept or reject event organizers or service providers based on platform policies and compliance with payment methods.
- Can add, edit, and arrange events or services into categories for easier navigation.
- Will be able to notify users about special discounts, promotional offers, and relevant updates.
- Administrators can record canceled or refunded tickets based on user actions.

### 2.2.2 Customers/Users

- Users will receive up-to-date information about new events or services, as well as ongoing discounts or special offers.
- They will be able to manage and modify their account details, including payment information and preferences.
- Users can search for tickets by event, category, location, or any relevant search criteria to find exactly what they’re looking for.

- They will have the ability to cancel or modify their bookings according to the platform's terms and conditions.
- Users can suggest events or services to be added to the platform, and provide reviews or feedback on the event or service providers.

## 2.3 Operating Environment

- This e-ticketing platform will function seamlessly across all major web browsers, including but not limited to Firefox, Chrome, Safari, and Edge. For a model reference, it is compatible with modern versions of these browsers with JavaScript and HTML5 capabilities.
- The platform will be accessible on Windows (7, 8, 10, 11), macOS, and Linux operating systems.
- Minimum system requirements include a Pentium 4 processor or higher.
- The platform requires at least 1GB of RAM or higher for optimal performance.

## 2.4 User Characteristics

The users of this e-ticketing platform include customers, event organizers, and administrators who maintain the system.

- **Customers** are individuals booking tickets for events, transport, or other services. They are assumed to have a basic understanding of computers and internet browsing.
- **Event organizers** or service providers are responsible for uploading event details and managing ticket sales on the platform. They should have the necessary skills to manage their listings and handle bookings.
- **Administrators** are responsible for maintaining the overall system, ensuring smooth operations, and resolving any issues that arise. They should have a deeper understanding of the internal modules of the platform and be capable of troubleshooting problems as needed.

## 2.5 Design and Implementation Constraints

- **Stable Internet Connection**: Users must have a device with a stable internet connection and internet browsing capabilities to access the e-ticketing platform.

- **Database Access**: The information about users, ticket availability, and event or transport listings will be stored in a database, which the platform can access in real-time.
- **Availability**: The platform will be available **24/7** to ensure that users can book tickets at any time.
- **Platform Compatibility**: The software is designed to be cross-platform and can run on any operating system, including mobile devices.
- **User Authentication**: Users must log in with the correct username and password to access their accounts and perform actions such as booking or managing tickets.

## 2.6 User Documentation

The user documentation for the e-ticketing system will provide comprehensive guides for various users, ensuring they can effectively navigate and utilize the platform. The documentation will include:

- **User Manuals for Customers:** Detailed step-by-step guides explaining how to create an account, search for available tickets, filter based on destination or price, book tickets, make payments, and manage bookings (e.g., cancellations or refunds).
- **Administrator Manuals**: Guides for system administrators on how to manage user accounts, view and edit ticket listings, update destinations, oversee payment systems, and generate detailed reports on user activity and transactions.
- **Training Guides for Support Staff**: These will help customer support teams assist users with ticket booking issues, payment queries, and troubleshooting common problems.
- **Online Help and FAQs**: A well-organized help section available on the platform itself, providing quick access to common issues, FAQs, and troubleshooting steps for users to solve problems without additional assistance.

  Additionally, video tutorials and interactive demos will be developed for key features of the system, such as booking tickets and using advanced search filters.

## 2.7 Assumptions and Dependencies

The website requires following third party products.

- **MongoDB**: For storing user data, ticket availability, and booking details in a scalable NoSQL database.

- **Express.js**: To handle the server-side application logic and API requests.
- **React.js**: For building an interactive and dynamic user interface that offers a seamless user experience.
- **Node.js**: To power the backend server, manage API routes, and handle requests efficiently.

# 3. External Interface Requirements

## 3.1 User Interfaces

The e-ticketing system will feature a user-friendly, responsive interface designed for seamless navigation. Users will have access to an intuitive dashboard, allowing them to search for tickets by location, date, price, and category. The interface will support filters for narrowing down ticket options and real-time updates on ticket availability. The ticket booking process will be clear, with options to review and confirm bookings, view payment history, and manage user profiles.

## 3.2 Hardware Interfaces

The system will be compatible with multiple devices, including Windows, Mac, and Linux systems, as well as mobile platforms (iOS and Android). It will require a minimum hardware specification of a 1.7 GHz processor and 2 GB of RAM to ensure smooth operation. It will also support standard peripherals like monitors, keyboards, and touch-screen devices.

## 3.3 Software Interfaces

The system will be developed using **JavaScript**, **Node.js**, and **React** for frontend, with **MySQL** for database management. It will also integrate with third-party payment systems such as **PayPal**, **Stripe**, and **Google Pay**. The backend will be deployed on cloud platforms like **AWS** or **Heroku** and will support RESTful APIs for seamless communication between client and server.

## 3.4 Communication Interfaces

The system will use **HTTP/HTTPS** for secure data communication, ensuring encryption through SSL certificates. Web services will allow users to interact with the system, including receiving email or SMS notifications for ticket confirmations or cancellations. The platform will also support integration with external ticketing services through APIs.

# 4. System Features

## 4.1 User Registration and Login

### 4.1.1 Description and Priority

This feature allows users to create accounts and log in using email and password or via social login (Google, Facebook). High priority as it is essential for user interaction with the system.

### 4.1.2 Stimulus/Response Sequences

- **Stimulus**: A user submits registration information.
- **Response**: The system creates a new user account and logs them in, displaying a confirmation message.

### 4.1.3 Functional Requirements

- **REQ-1**: The system must validate unique emails during registration.
- **REQ-2**: Users should be able to log in using social media credentials.
- **REQ-3**: Users must receive a confirmation email after registration.

## 4.2 Ticket Booking and Management

### 4.2.1 Description and Priority

This feature allows users to search for tickets, view availability, and book tickets for events or travel. High priority, as this is the core function of the system.

### 4.2.2 Stimulus/Response Sequences

- **Stimulus**: A user selects ticket preferences (e.g., location, date).
- **Response**: The system displays available tickets, updates based on filters, and confirms the booking once payment is processed.

### 4.2.3 Functional Requirements

- **REQ-4**: The system must display real-time availability for tickets.
- **REQ-5**: Users should be able to add multiple tickets to a cart for bulk purchase.
- **REQ-6**: The system must provide a booking confirmation after payment.

## 4.3 Payment Processing

### 4.3.1 Description and Priority

This feature handles payment processing through various gateways, ensuring secure transactions. High priority for ensuring successful ticket purchases.

### 4.3.2 Stimulus/Response Sequences

- **Stimulus**: A user proceeds to payment after selecting tickets.
- **Response**: The system processes the payment, confirms it, and updates the user's account with the transaction.

### 4.3.3 Functional Requirements

- **REQ-7**: The system must support multiple payment methods (credit card, PayPal, Google Pay).
- **REQ-8**: Payment information must be encrypted for security.
- **REQ-9**: The system should notify the user of payment success or failure.

## 4.4 Admin Management

### 4.4.1 Description and Priority

Admins will manage events, users, payments, and ticket availability. This feature is critical to maintaining system integrity and ensuring correct ticketing information.

### 4.4.2 Stimulus/Response Sequences

- **Stimulus**: An admin logs in and updates an event's ticket availability.
- **Response**: The system reflects the changes in real-time for all users.

### 4.4.3 Functional Requirements

- **REQ-10**: Admins must be able to add, edit, or delete events and ticket listings.
- **REQ-11**: The system must generate reports on ticket sales and user activity.
- **REQ-12**: Admins should be able to view and manage user accounts.

## 4.5 Notifications and Alerts

### 4.5.1 Description and Priority

This feature sends real-time notifications about booking status, cancellations, or payment issues. High priority for keeping users informed.

### 4.5.2 Stimulus/Response Sequences

- **Stimulus**: A user books a ticket or an event is cancelled.
- **Response**: The system sends a notification via email or SMS.

### 4.5.3 Functional Requirements

- **REQ-13**: The system must send booking confirmations to users.
- **REQ-14:** The system should notify users of cancellations or changes in ticket availability.
- **REQ-15**: The system must ensure notifications are sent in real time.

# 5. Nonfunctional Requirements

## 5.1 Performance Requirements

The e-ticketing system must be optimized for high performance, capable of handling large volumes of ticket searches, transactions, and user queries efficiently. Data retrieval for ticket availability, payment history, or booking details must occur within 3 seconds under normal operating conditions. The system should support up to 5000 concurrent users without performance degradation, and search results for available tickets should be generated in less than 5 seconds. Additionally, real-time ticket availability updates should reflect changes (e.g., sold-out events) within 2 seconds.

## 5.2 Safety Requirements

Safety is critical to prevent loss of transaction data or incorrect bookings. The system must include a reliable backup mechanism, with automated backups scheduled hourly. A rollback feature must be available to allow administrators to undo erroneous actions, such as mistaken cancellations or payments. The system should ensure consistency in ticket inventory to avoid overselling or booking errors during high-demand periods.

## 5.3 Security Requirements

The e-ticketing system must prioritize security by implementing encryption for both data at rest and data in transit. Role-based access control will be enforced to limit access to sensitive data (e.g., payment information) to authorized users only. Payment transactions will be secured via SSL and tokenization to prevent unauthorized access or fraud. An audit trail will record all access and modifications to user accounts, tickets, and payment details for accountability.

## 5.4 Software Quality Attributes

- **Reliability**: The system must maintain 99.9% uptime, ensuring users can access ticketing services without interruptions.
- **Scalability**: The system must be designed to scale horizontally, accommodating increased traffic during peak booking periods.

- **Maintainability**: The system must have a modular architecture, allowing for easy updates, bug fixes, and the integration of new features.
- **Usability**: The interface must be intuitive, with minimal training required for both users and administrators.
- **Compatibility**: The system will integrate smoothly with third-party payment processors and ticket distribution services, using industry-standard protocols (e.g., REST APIs).

## 5.5 Business Rules

The system must be flexible enough to handle various ticketing models (prepaid, postpaid) and discount schemes (bulk tickets, early bird discounts). It must comply with local accounting standards, providing accurate reports for tax purposes and ensuring ticket availability and pricing are properly updated in real time. Refund and cancellation policies must be automated, issuing refunds based on predefined rules.

# 6. Other Requirements

## 6.1 Regulatory Compliance

The system must comply with data privacy regulations, such as the **General Data Protection Regulation (GDPR)** and **California Consumer Privacy Act (CCPA)**, ensuring the secure handling of user data. It must also adhere to **Payment Card Industry Data Security Standard (PCI DSS)** for processing payment information securely.

## 6.2 Data Backup and Recovery

Automated daily backups must be performed to prevent data loss. A **disaster recovery** plan must ensure that the system can be restored with minimal downtime in case of failure, including data replication across multiple locations to guarantee ticketing services remain available even during emergencies.

## 6.3 User Accessibility

The system must comply with **Web Content Accessibility Guidelines (WCAG) 2.1** to ensure accessibility for users with disabilities. This includes support for screen readers, keyboard navigation, and alternative text for images, making the system usable for all.

## 6.4 Localization and Internationalization

The system will support **multiple languages and currencies**, enabling a global audience to use the platform. Localization features will include the adaptation of date formats, currency symbols, and local ticketing standards.

## 6.5 Scalability

The system must be designed with scalability in mind, ensuring that it can efficiently handle increased traffic during peak times, such as high-demand ticket releases. This includes scalable cloud infrastructure to manage server loads dynamically.

### 6.6 Session Management

The system must implement **secure session management**, including automatic session expiration after a period of inactivity and **multi-factor authentication (MFA)** for additional security. The system will ensure secure logout protocols, especially on shared or public devices.

## Appendix A: Glossary

- **REST API**: A set of rules allowing systems to communicate and exchange data securely over the web.
- **SSL Encryption**: Secure Sockets Layer, a standard security protocol for establishing encrypted links between web servers and browsers.
- **Tokenization**: The process of substituting sensitive data with unique identification symbols that retain essential information without compromising security.

## Appendix B:

Analysis Models (Included as appropriate, such as use case diagrams, data flow diagrams, etc.)

## Use case template

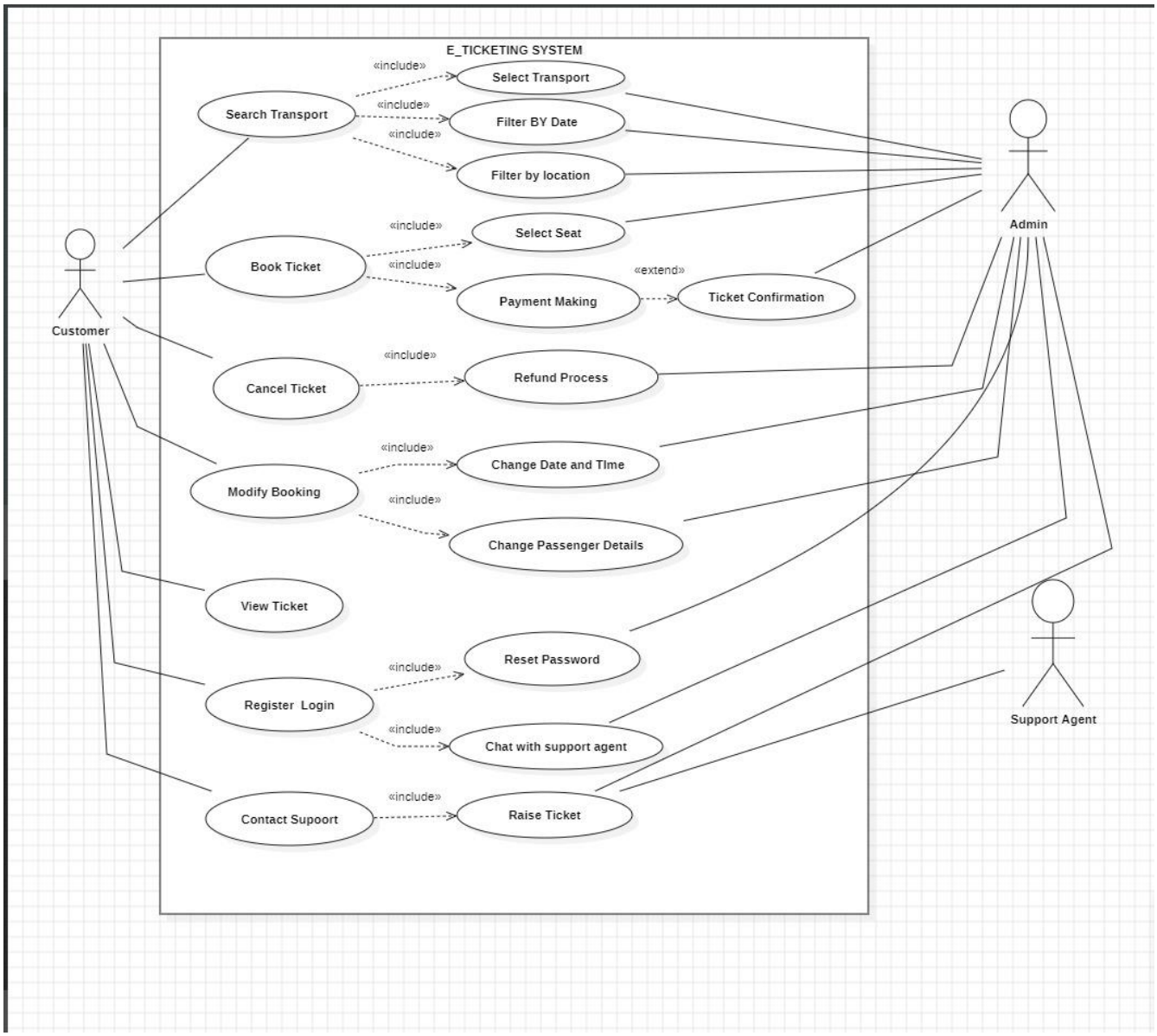
| | | | |
|---|---|---|---|
| **Use Case ID:** | 0123456789 | | |
| **Use Case Name:** | E-Ticketing | | |
| **End Objective:** | Automate and facilitate the whole process of shopping | | |
| **Created by:** | 1. Aditya<br>2. Harsha<br>3. Armaan<br>4. Anand<br>5. Varshith | **On(date):** | October 16, 2024 |
| **User/Actor:** | Customer, Seller and Administrator | | |
| **Trigger:** | Customer buying ticket by logging in the site | | |
| **Basic/Normal Flows** | | | |
| **User Actions** | **System Actions** | | |

| | |
|---|---|
| The user tries to log in but doesn't have an account. | The page requests the user to register an account on the registration page before proceeding with the login. |
| The user tries to log in by entering details. | If the details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details. |
| The user tries to book a ticket. | If the ticket is not available, the system displays a "Not available" message for that specific booking. |
| The user completes ticket booking and proceeds to payment. | The system confirms the availability of the ticket and redirects the user to the payment page. |

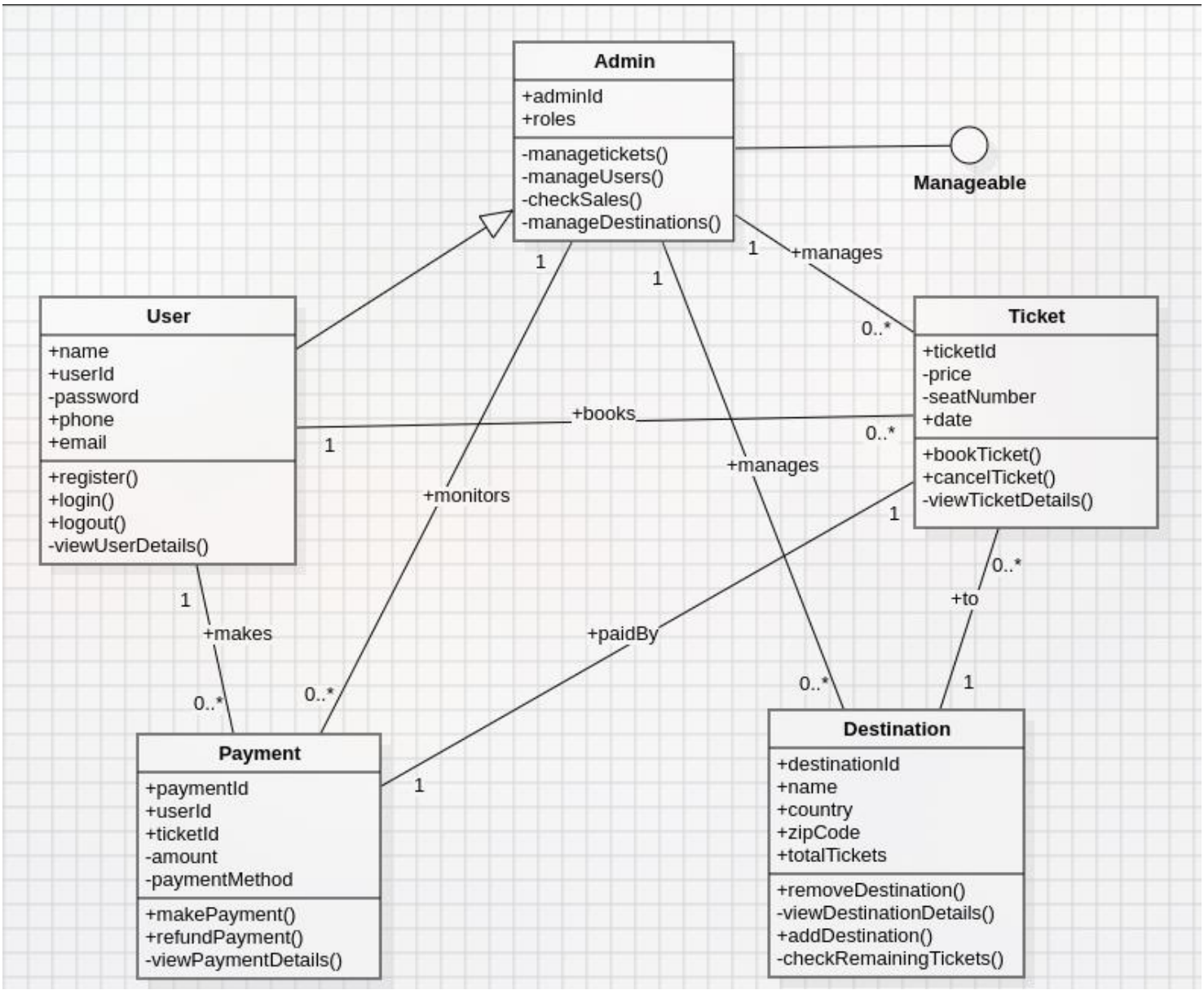
**Exception Flows**

| **User Actions** | **System Actions** |
|---|---|
| The user tries to log in but doesn't have an account. | The system prompts the user to register an account on the registration page before proceeding with the login. |
| The user tries to log in by entering details. | If the login details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details. |
| The user tries to book a ticket. | If the ticket is not available, the system displays a "Not available" message for that specific booking. |

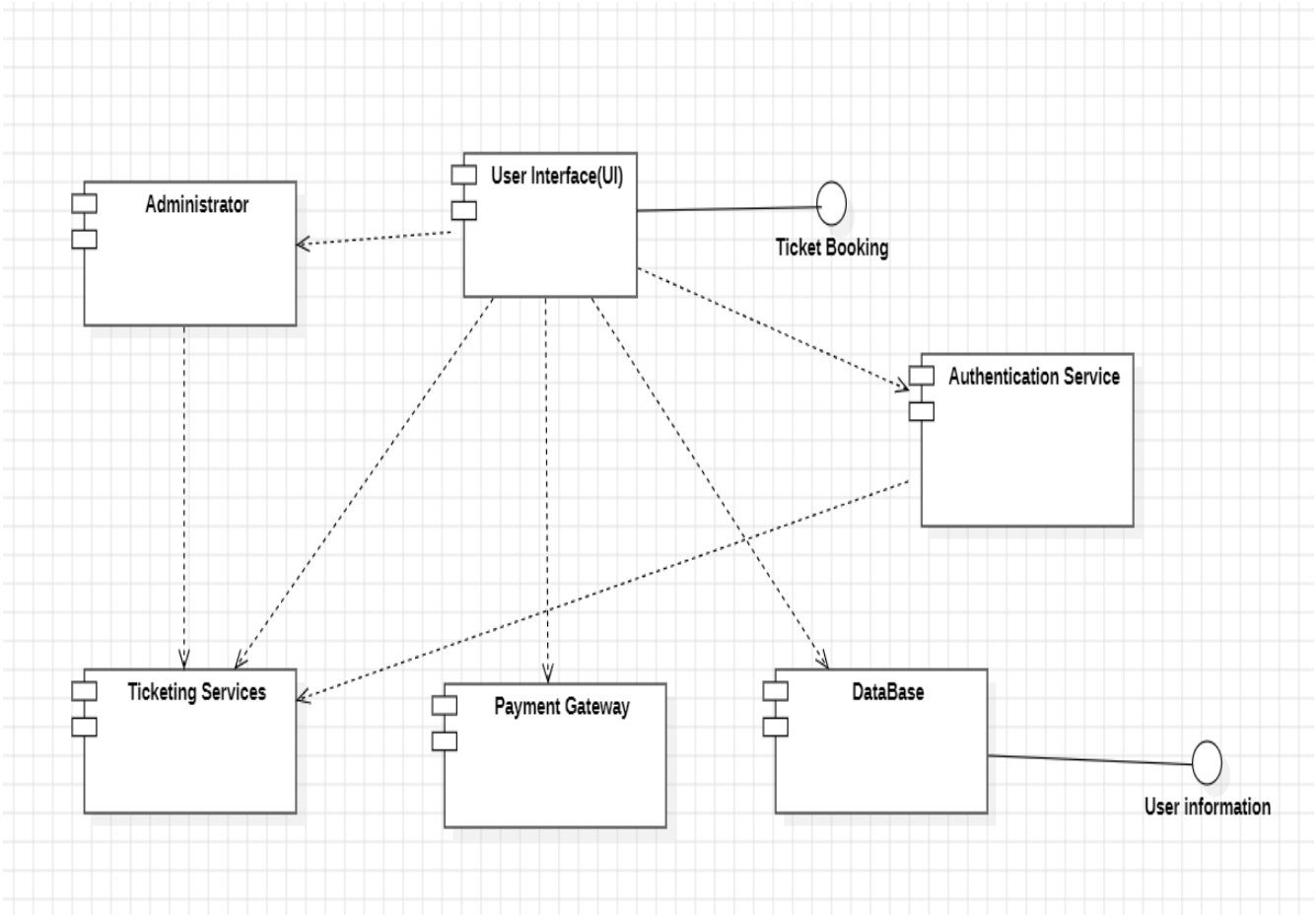
# Use-Case Diagram



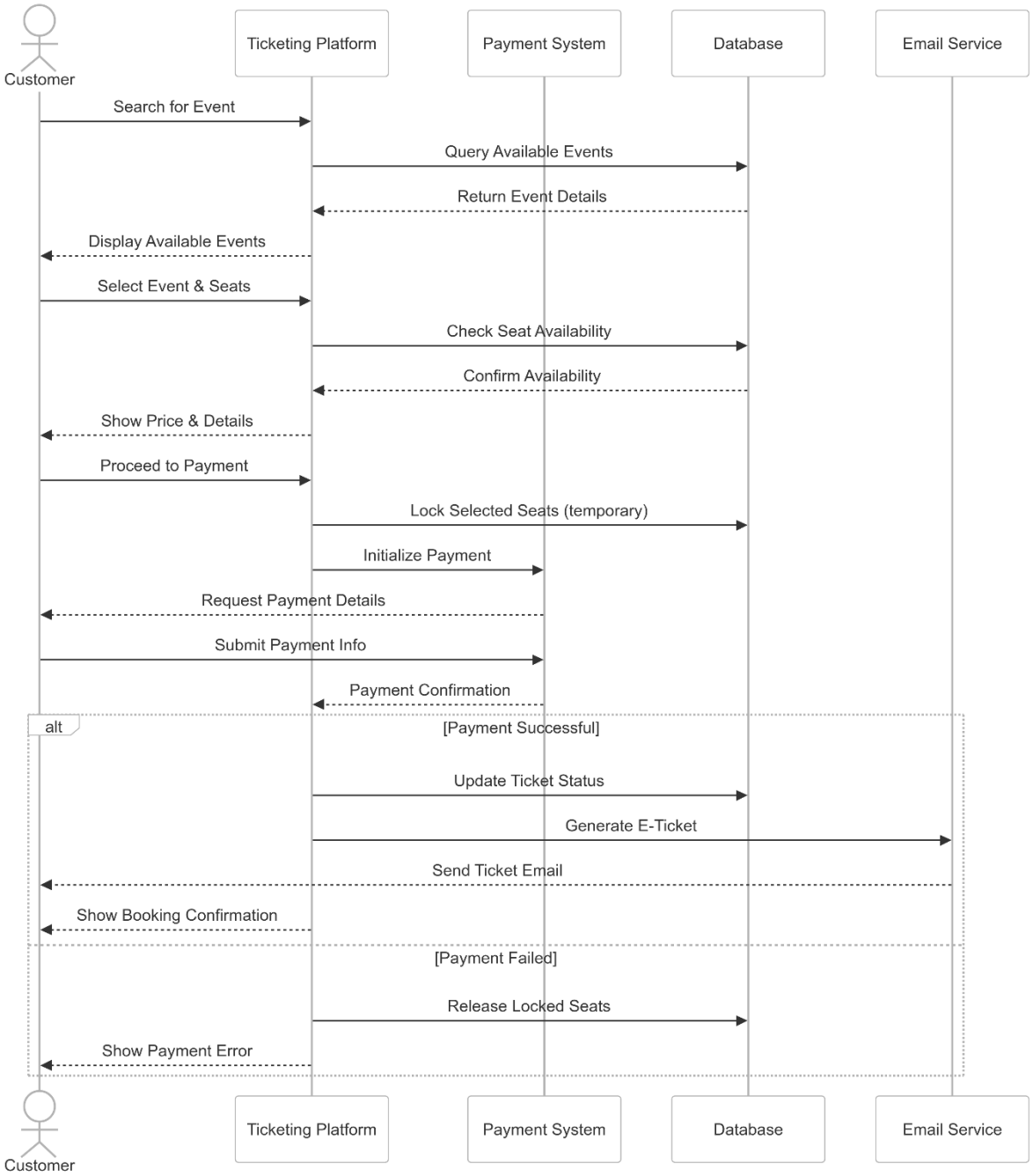
## Class Diagram



# Component Diagram



# Sequence Diagram



# Appendix C: To Be Determined (TBD) List

This appendix lists items that need further clarification or finalization during the project's development. These items will be addressed as the project evolves, and final decisions will be made during subsequent stages.

1. **Third-Party Payment Systems**
    - The exact payment gateways (e.g., PayPal, Stripe, Google Pay) to be integrated for processing payments securely and efficiently are still under evaluation.
2. **Finalized User Roles and Permissions**
    - User roles and permission levels, especially for the **Admin** class, need further definition. This will include detailed permissions (e.g., adding events, managing users, generating reports) and access control for sensitive actions.
3. **Ticket Types and Categories**
    - Finalization of the specific ticket types (e.g., VIP, standard, student discounts) and categories (e.g., transportation, event types) the system will support.

Result: The Software Requirements Specification (SRS) document was made, understood and its key components identified, ensuring clarity in project requirements.

# Experiment-3: Installation of Maven, Eclipse and Tomcat

**Aim**: To install Maven, Eclipse, and Tomcat.

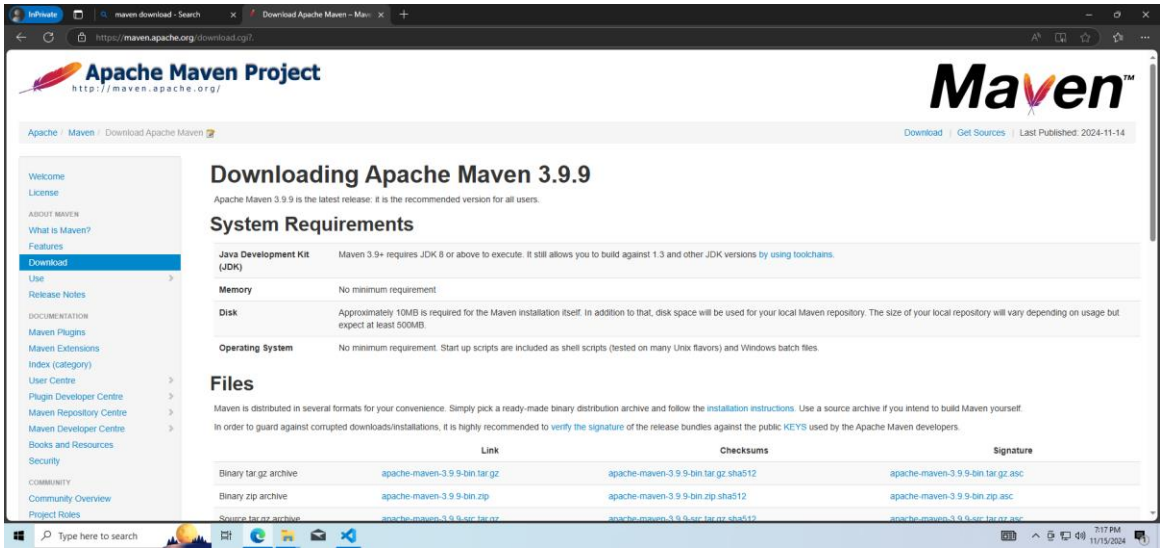
**Introduction**:

1. **Maven**: Maven is a build automation tool primarily used for Java projects. It helps in managing project dependencies, compiling code, packaging applications, and deploying them. It simplifies the build process by providing a consistent environment for building, testing, and deploying Java applications.
2. **Eclipse**: Eclipse is a popular Integrated Development Environment (IDE) used mainly for Java development. It provides tools for writing, debugging, and managing Java code. It also supports plugins for other languages and frameworks, making it versatile for a wide range of software development tasks.
3. **Tomcat**: Tomcat is an open-source web server and servlet container developed by Apache. It is used for running Java-based web applications. Tomcat executes Java servlets and renders JavaServer Pages (JSP), enabling dynamic content in web applications.

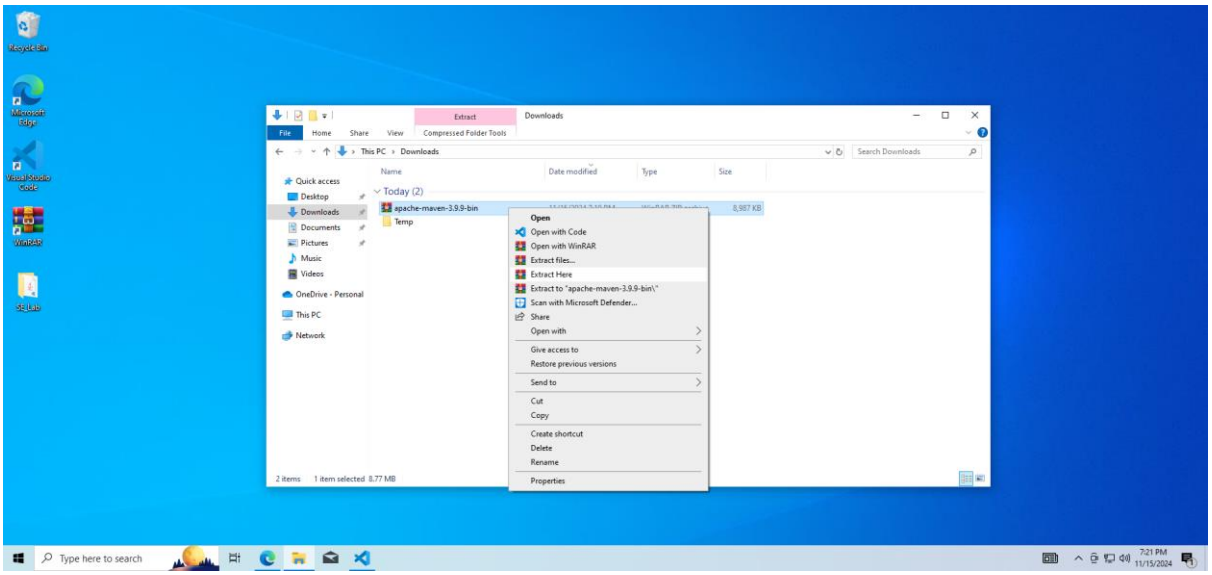
**Procedure:**

# 1.Maven Installation

Step 1: Go to https://maven.apache.org/download.cgi, and install the Binary zip archive for Maven.

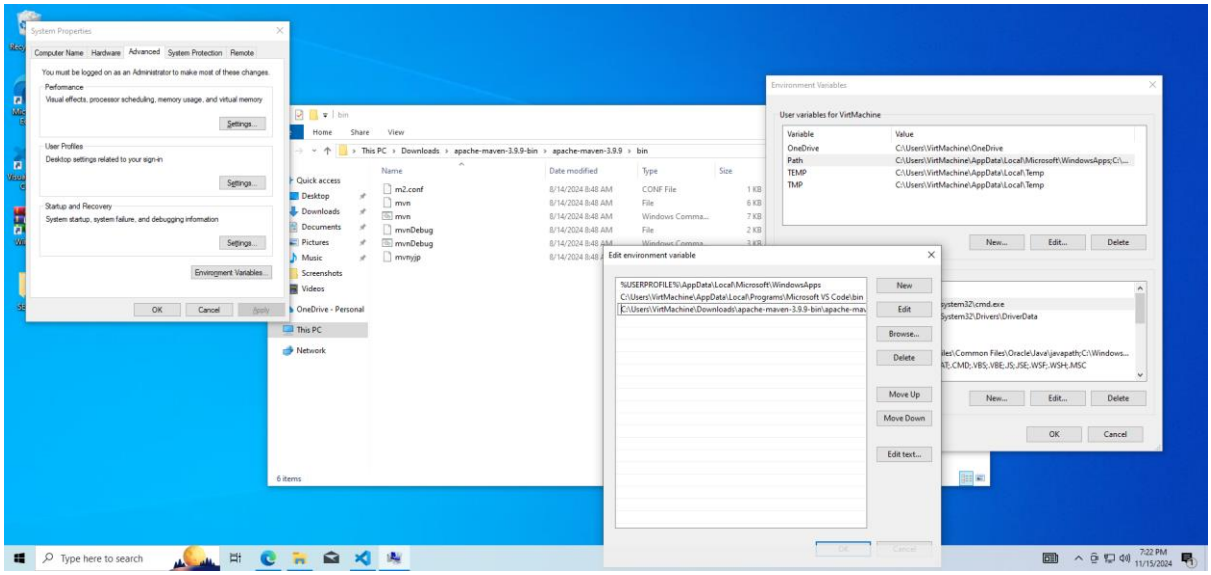



Step 2: Unzip the compressed folder present in the Downloads folder.

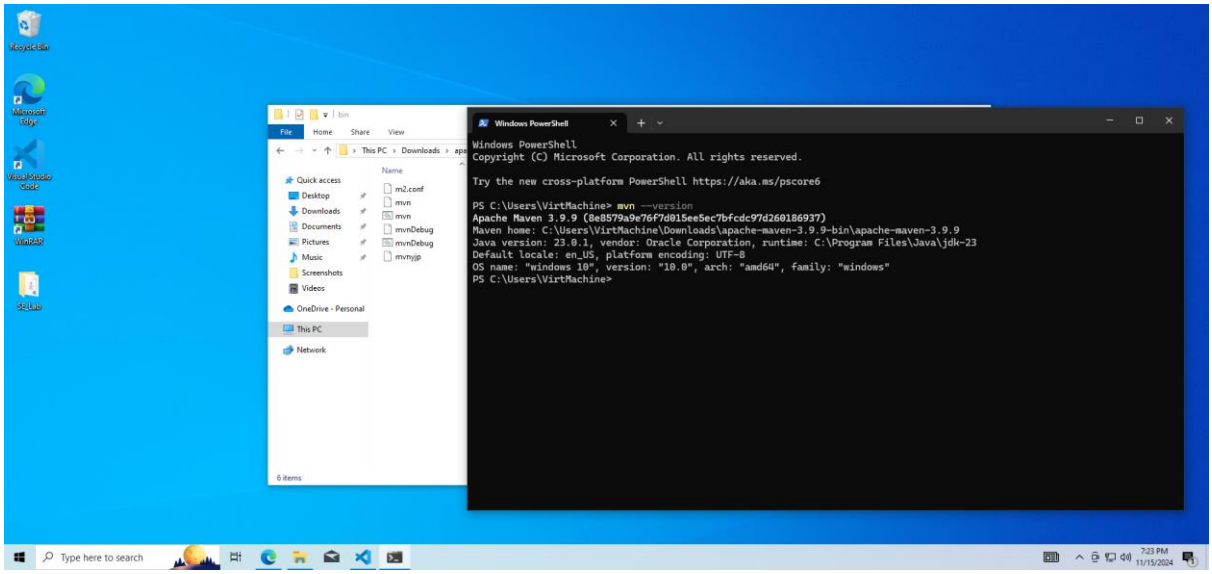


Step 3: Select the location for the extracted folder.

Step 4: Add classpath for the bin folder inside the Maven folder in the environment variables.

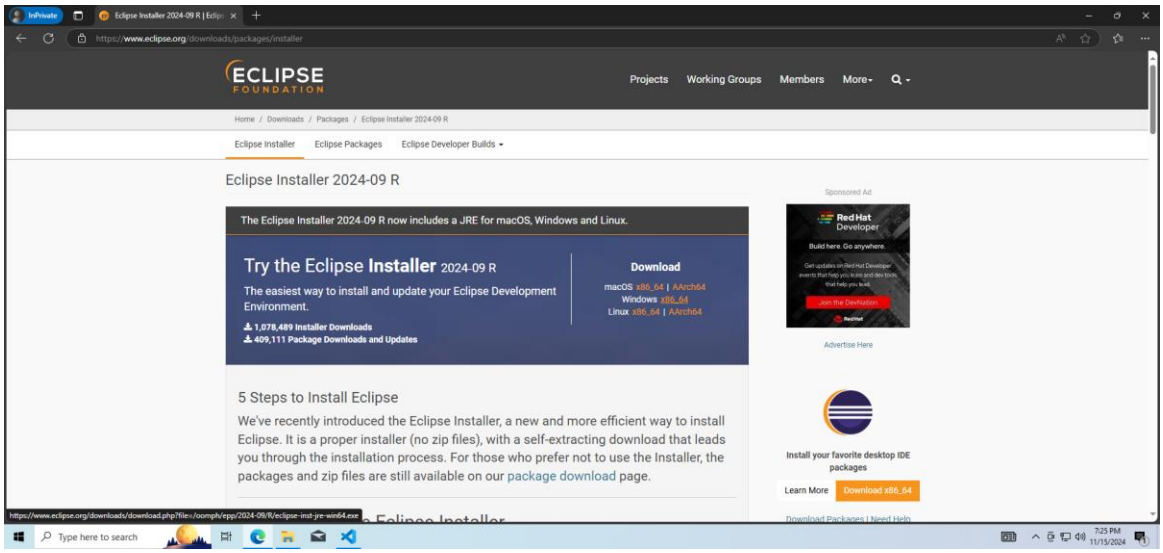



Step 5: Check maven installation

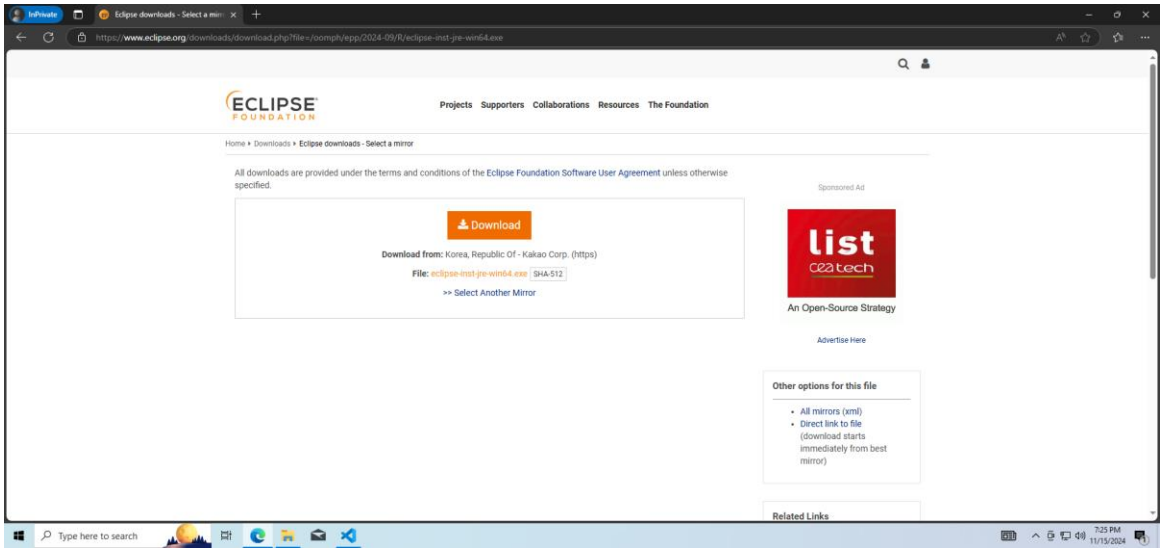



# 2. Eclipse Installation

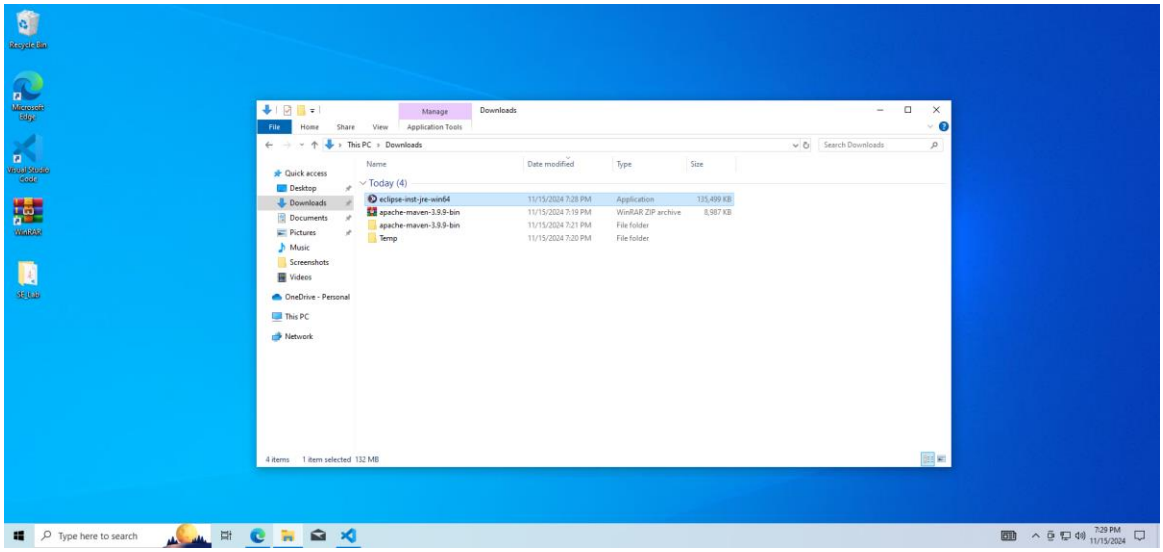
Step 1: Go to https://eclipse.org/downloads, and click on Download x86_64 to begin installation.



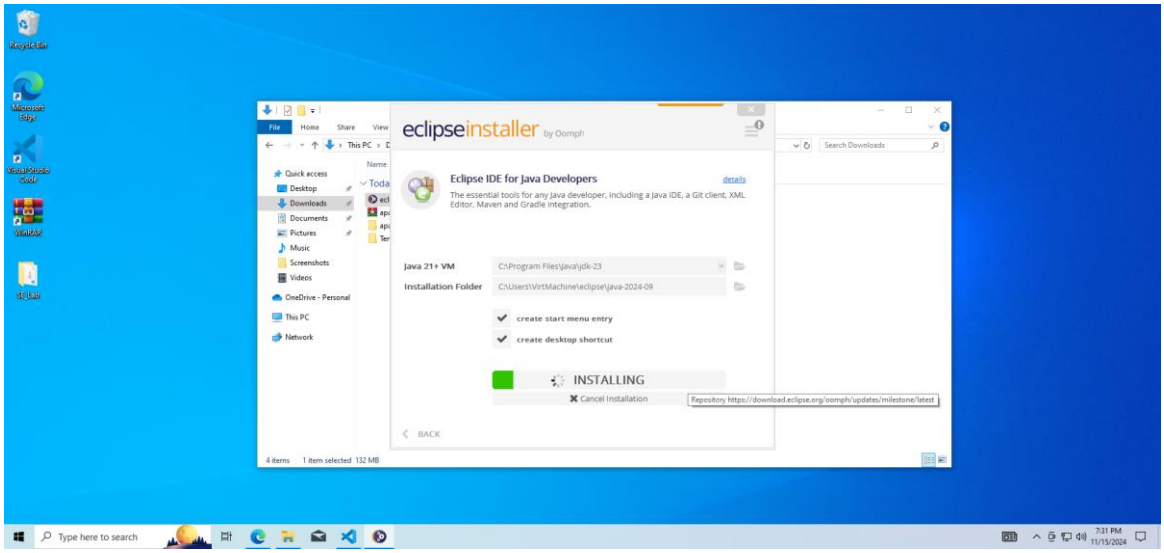

Step 2: Click download again on the next page.



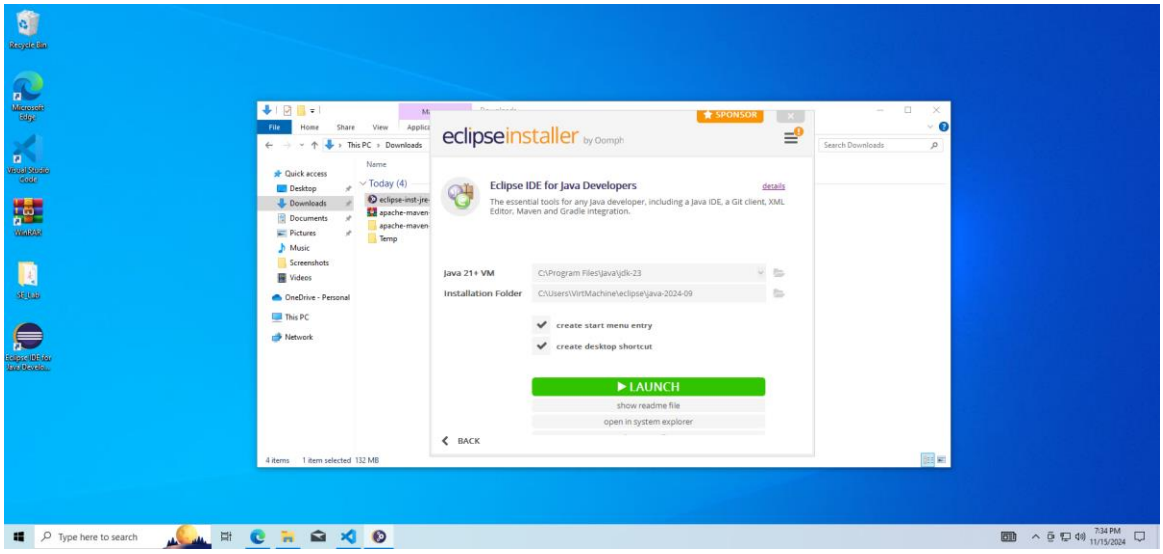

Step 3: Run the installer from the Downloads folder and click on “Eclipse IDE for Java Developers”.



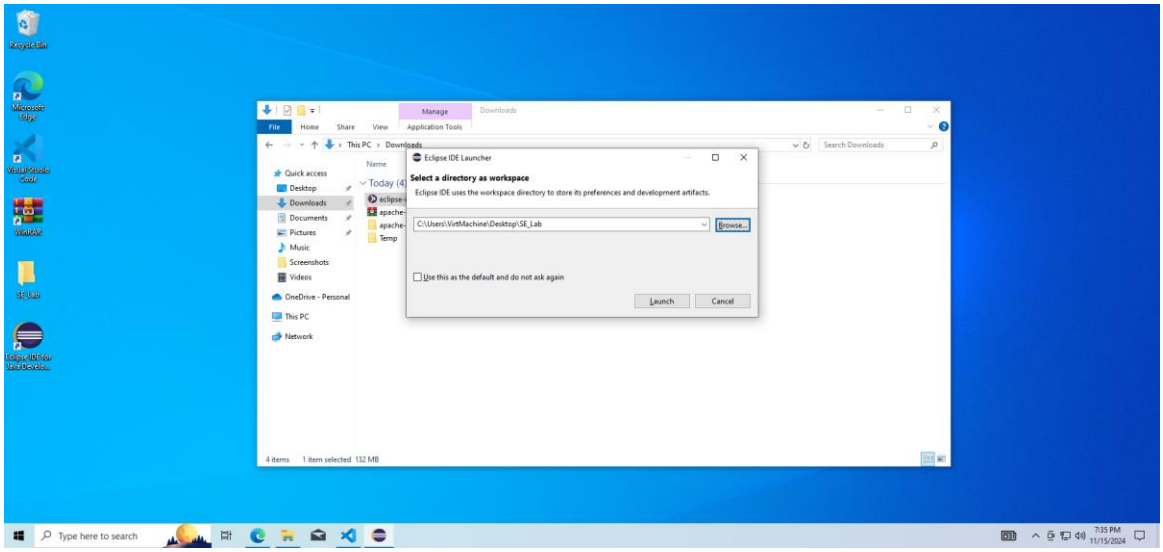

Step 4: Select the installation folder in the next page, and click on Install.



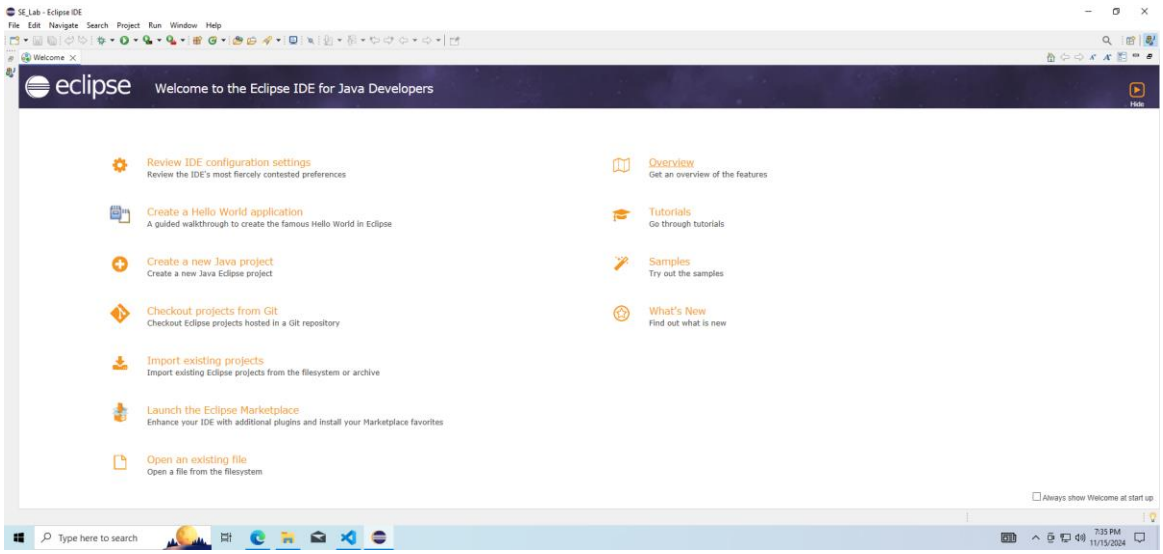

Step 5: Click on “Launch” after the installation is done.




Step 6: Select a directory as workspace for Eclipse and click on “Launch”.

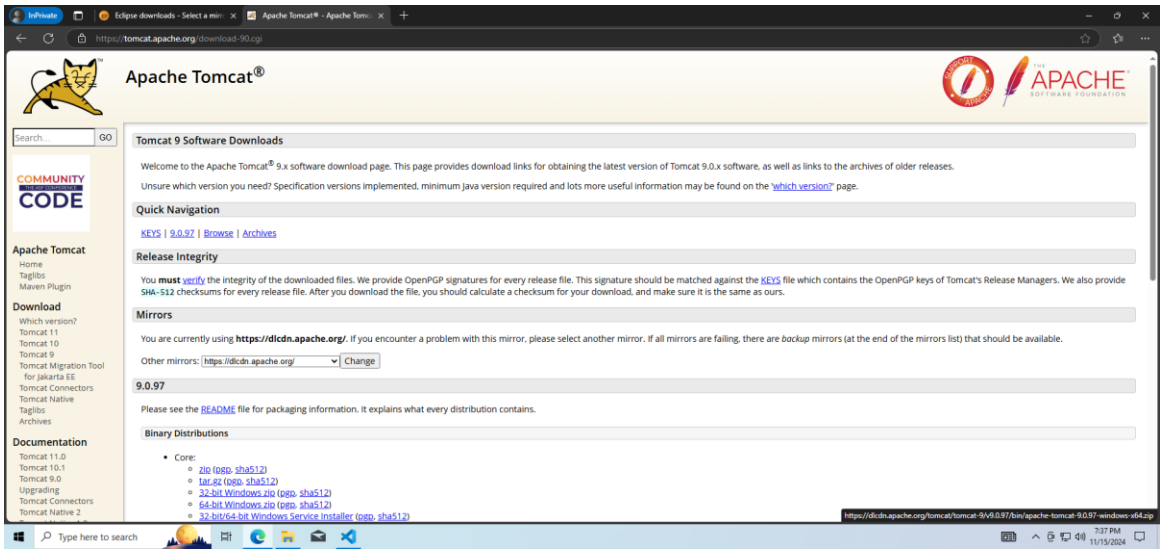



Step 7: You will see the welcome page in the Eclipse IDE

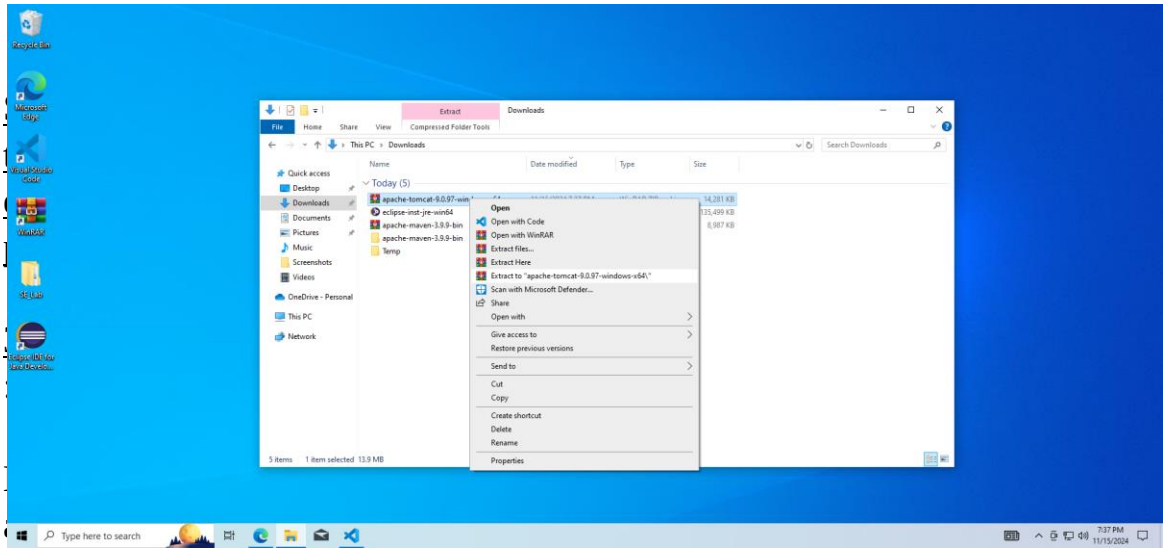


# 3. Tomcat Installation

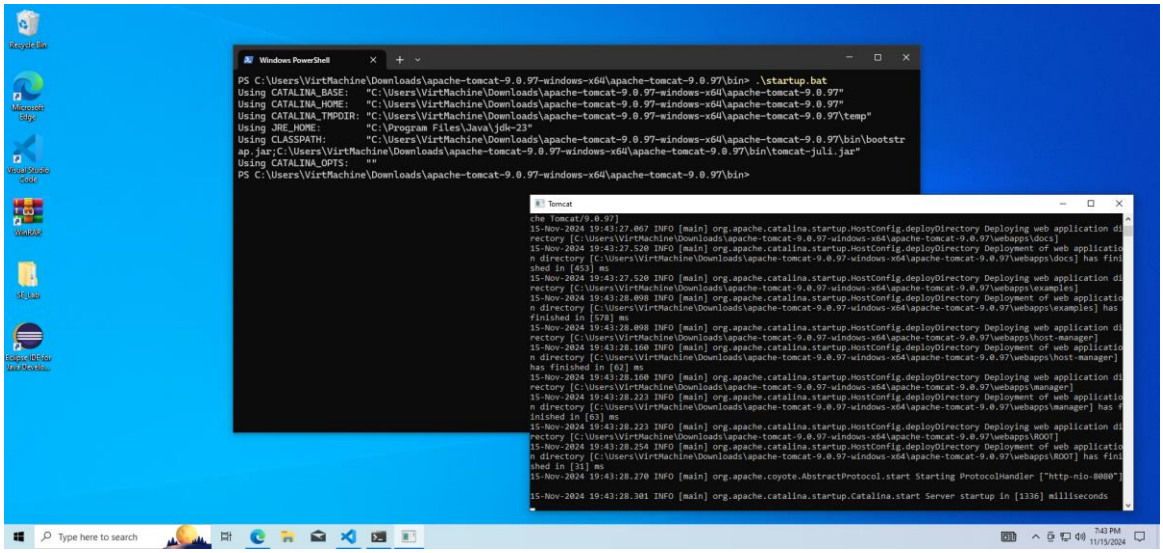
Step 1: Go to https://tomcat.apache.ord/download-90.cgi, and download the “64-bit Windows zip” folder for Tomcat.




Step 2: Unzip the apache-tomcat-9.0.97-windows-x64 folder by right clicking it and click on extract all and selecting the destination folder.

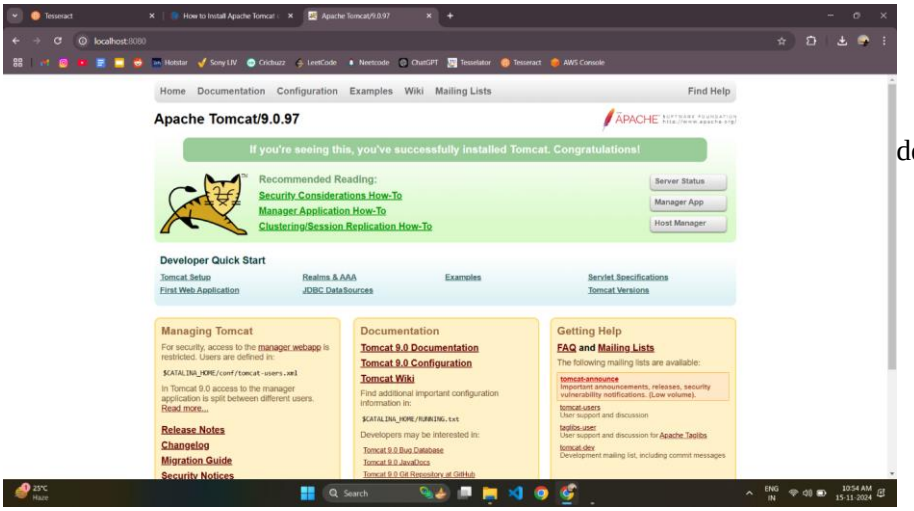


vigate to the bin folder inside the extracted tomcat folder and run startup as shown.




Step 4: The following window appears showing logs for tomcat. The tomcat server is running now!

Step 5: Navigate to http://localhost:8080. If you see the following, your installation was successful.

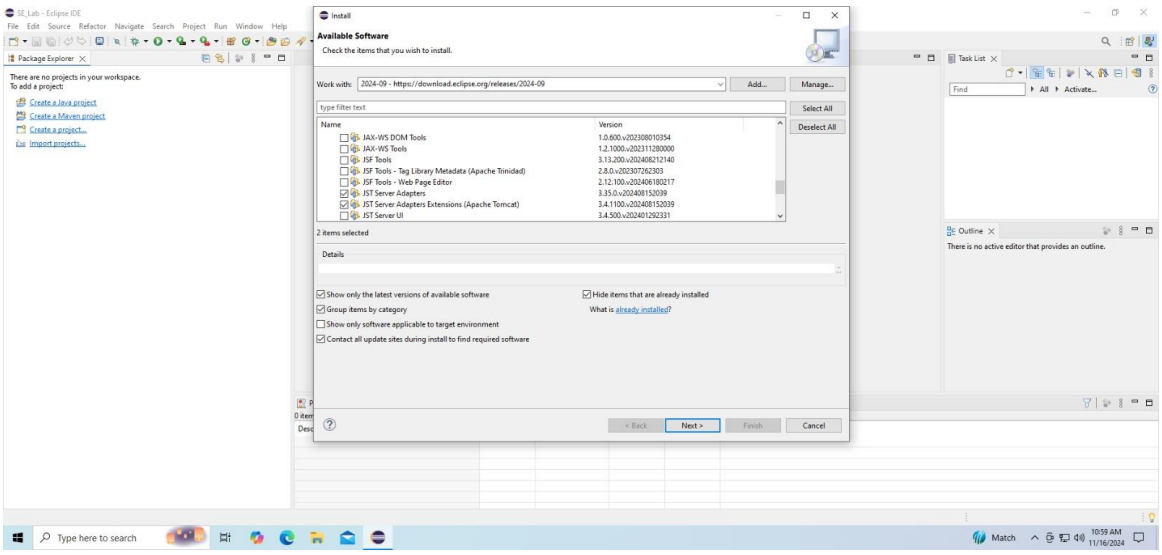



ddg

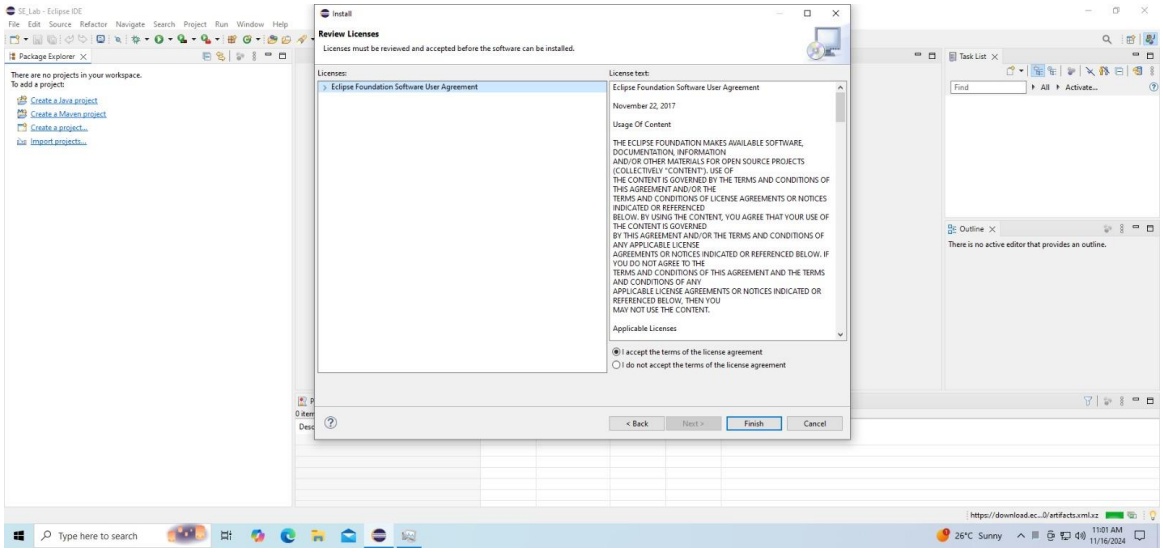
Step 6: Click on CTRL+C in the tomcat window, to stop the tomcat server.

# 4. Maven and Tomcat configurtion

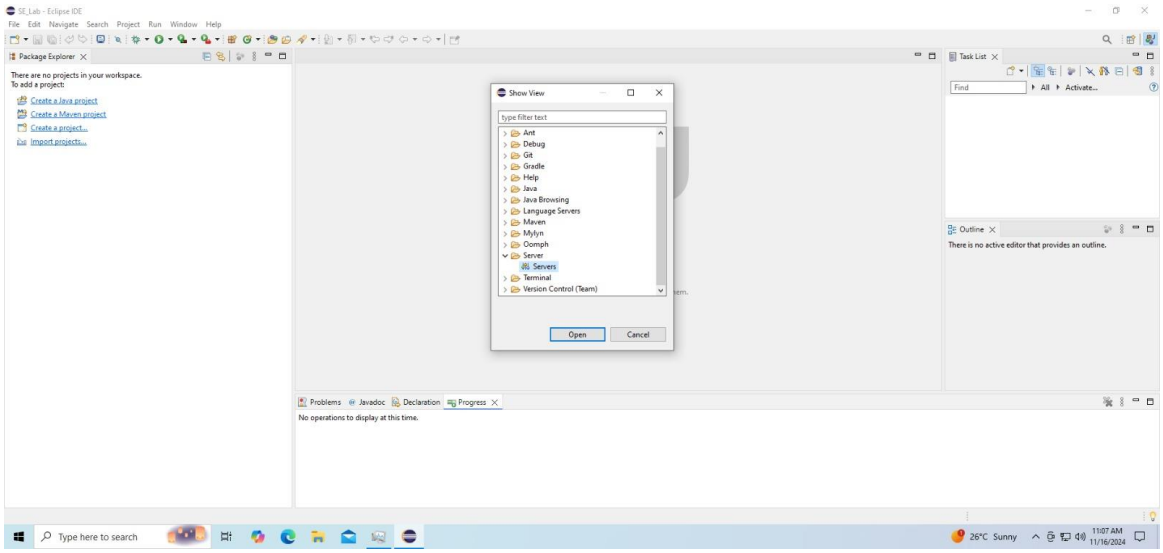
Step 1: Install Java Serverlet and Tomcat Server updates.



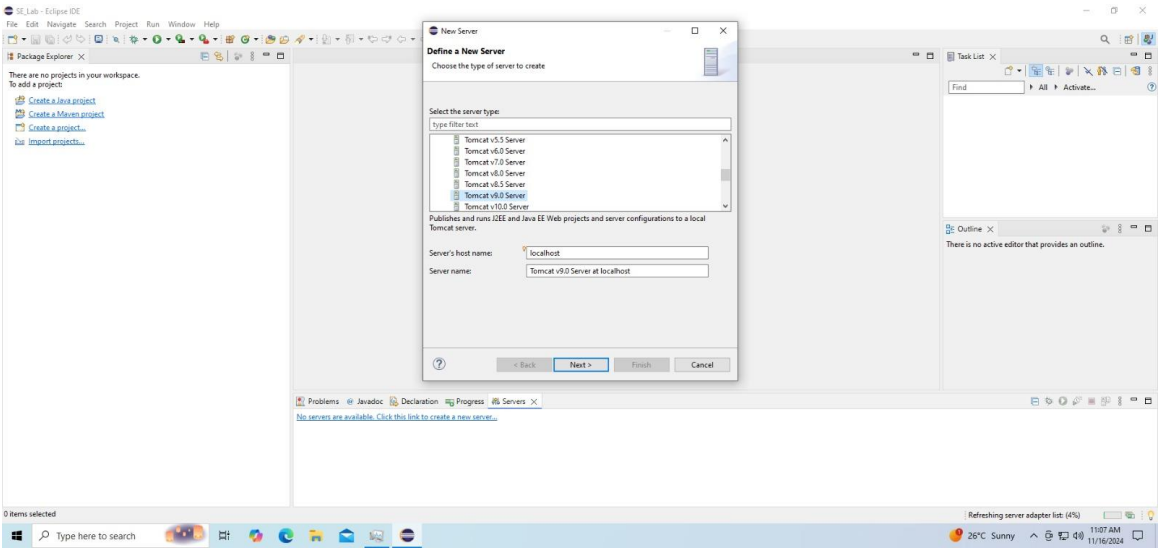

Step 2: Click on finish and restart Eclipse.

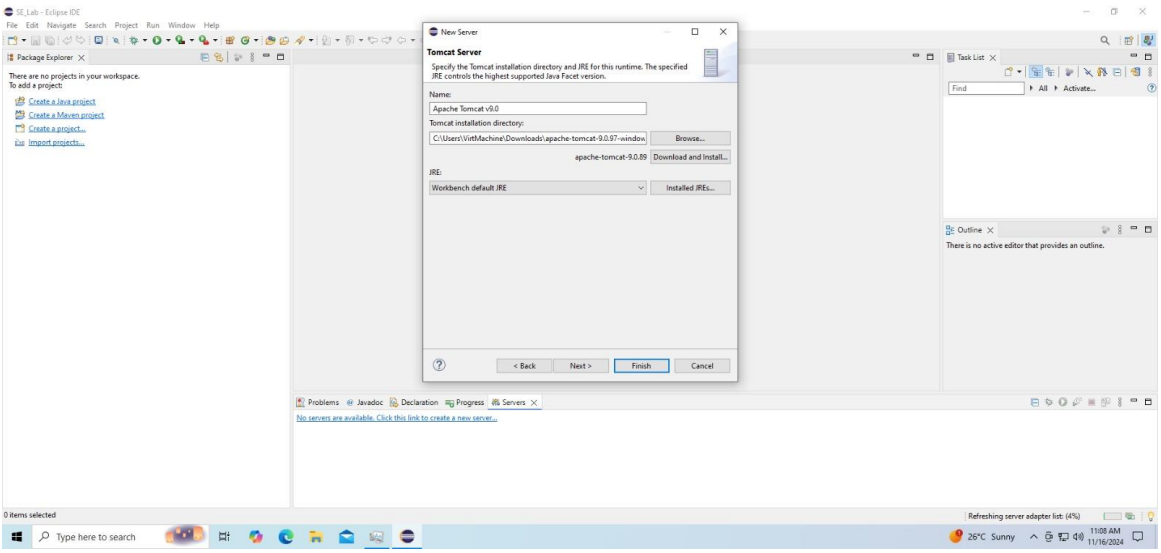



Step 3: Click on Servers in the Other tab in the Window tab.

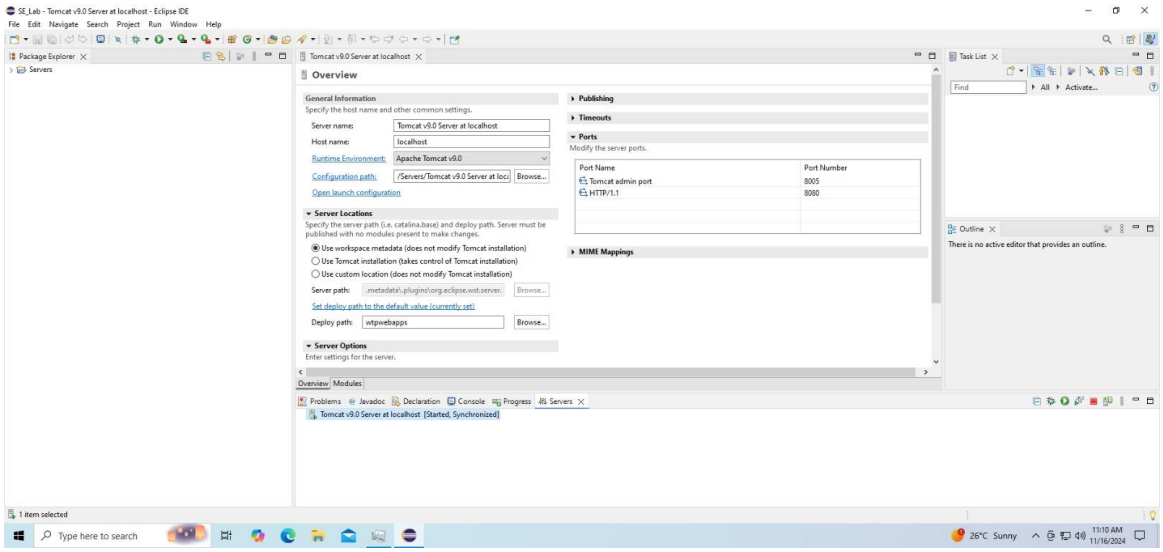



Step 5: Select your Tomcat version

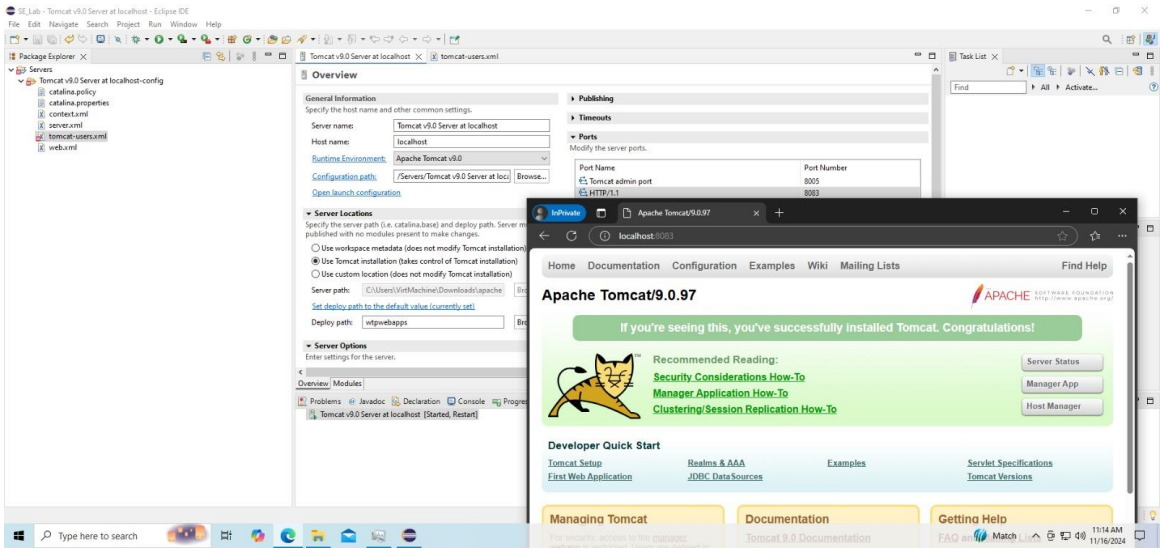


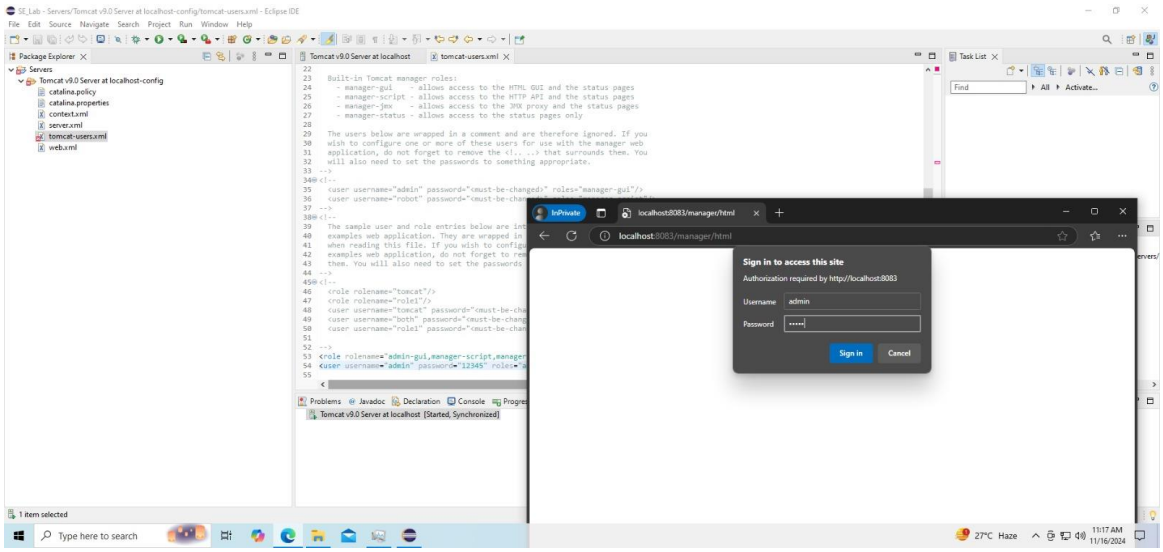

Step 6: Click on start server for Tomcat



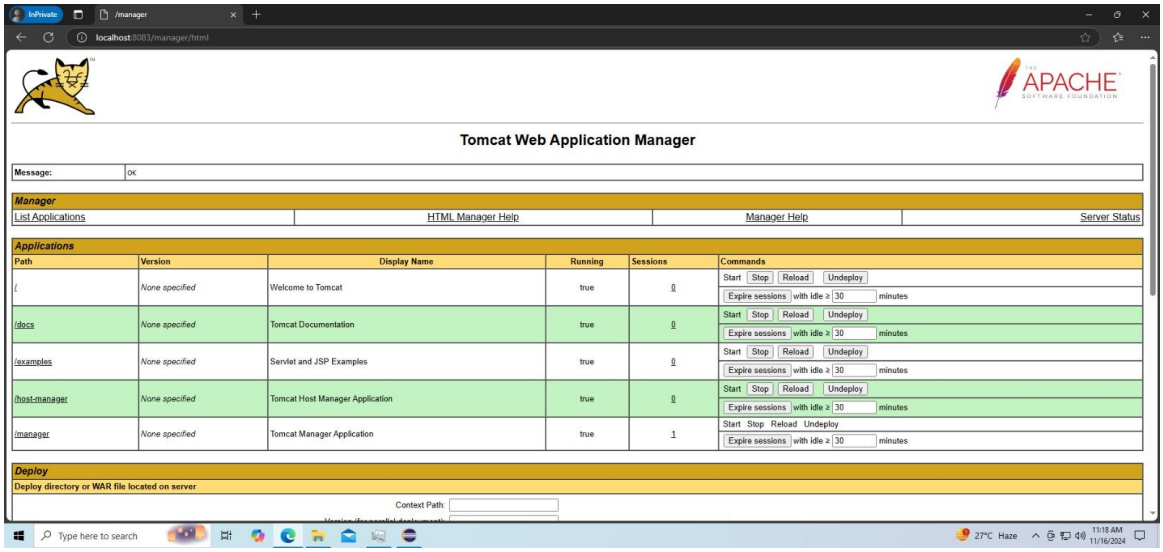

Step 7: The Tomcat Server is started on port 8083



Step 8: Modify tomcat-users.xml and add new user




Step 8: In the web browser, select manage apps, then login with your credentials.



Result: Eclipse, Maven and Tomcat were installed and Tomcat was successfully configured with Eclipse.

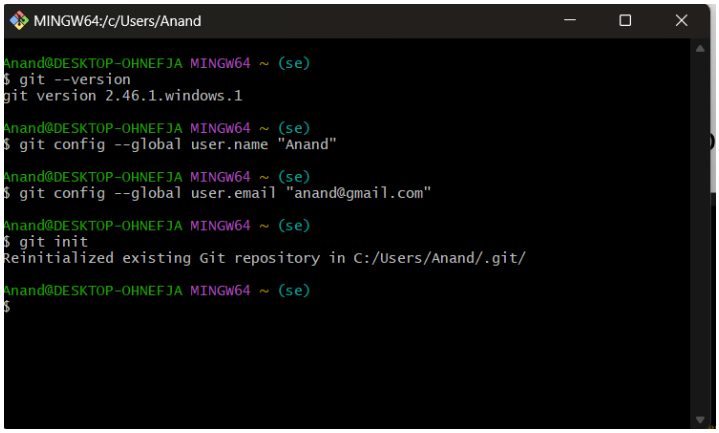
# Experiment-4: Creating the project, and commiting it using Git and GitHub

Aim: To explore Git Commands, create a repository on GitHub and committing a project using Git and GitHub.

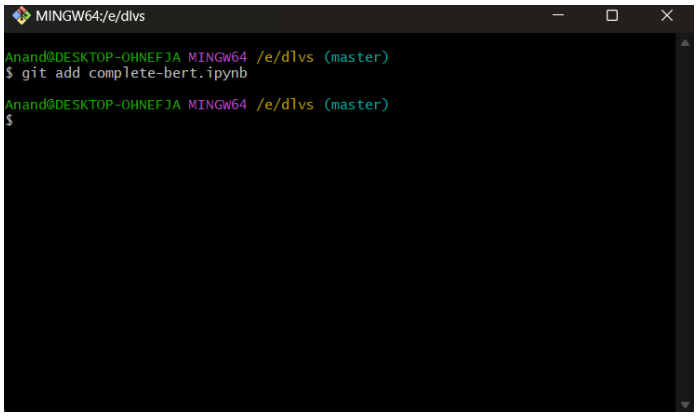
Introduction:

1. Git: A distributed version control system that facilitates efficient management of code repositories, tracking changes, and collaboration across teams.
2. GitHub: An online platform for hosting Git repositories, enabling developers to collaborate on projects, track changes, and manage contributions seamlessly.
3. Workflow Overview:
    - Initialize a local repository using Git.
    - Create or modify files in the project.
    - Stage and commit changes locally.
    - Connect the local repository to GitHub using a remote URL.
    - Push the project to GitHub for sharing and collaboration.

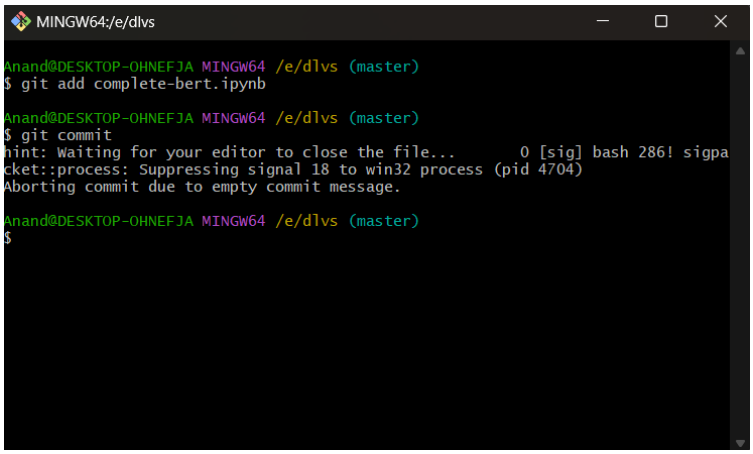
1. **git init**



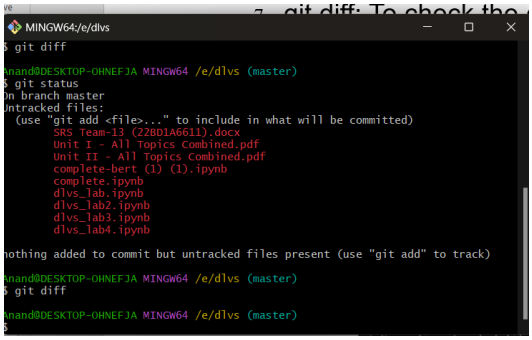

## 2. **git add** and **git status**



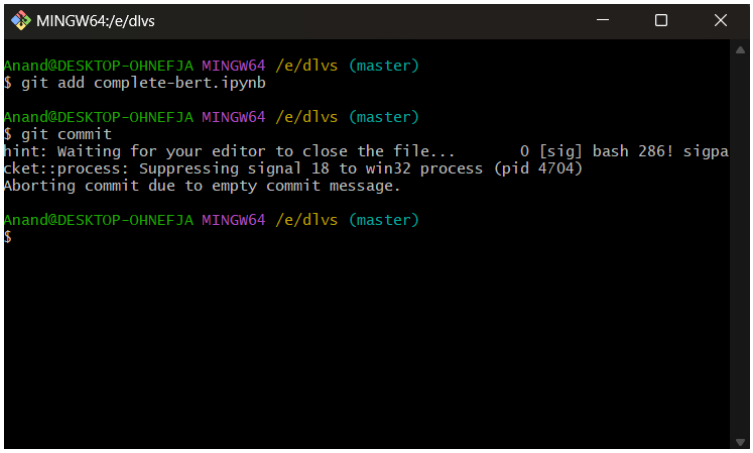

## 3. **git commit**



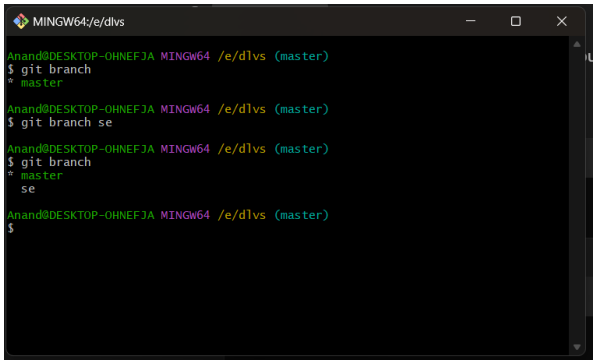

## 4. **git diff**



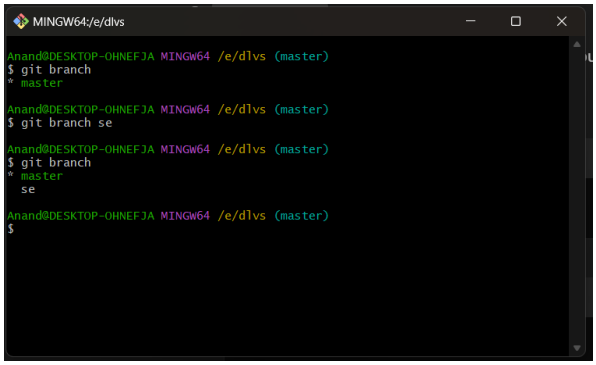

## 5. **git diff** and **git commit**



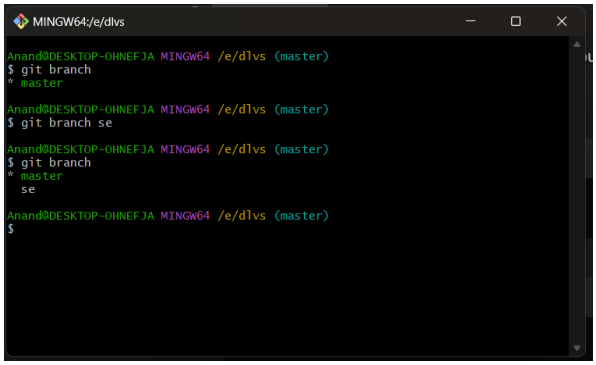

## 6. **git branch -a**



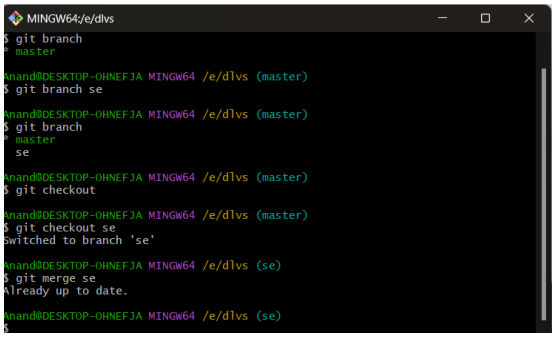

## 7. **git branch**



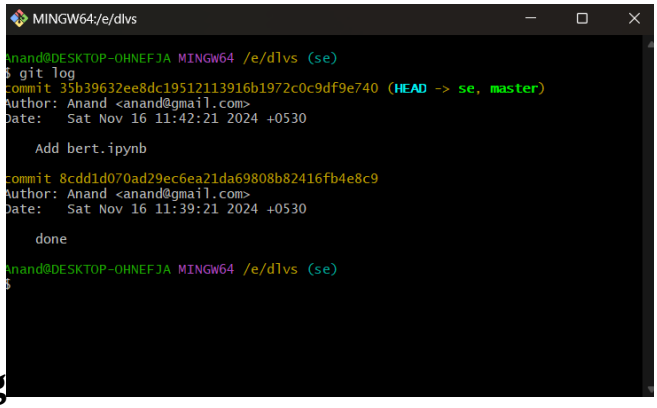

## 8. **git branch -d**



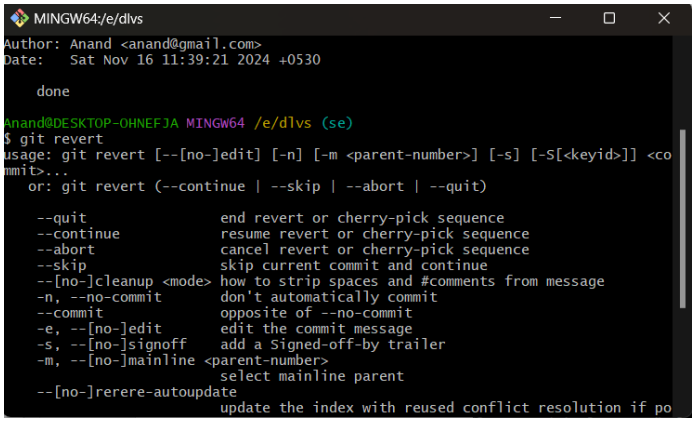

## 11. **git merge**



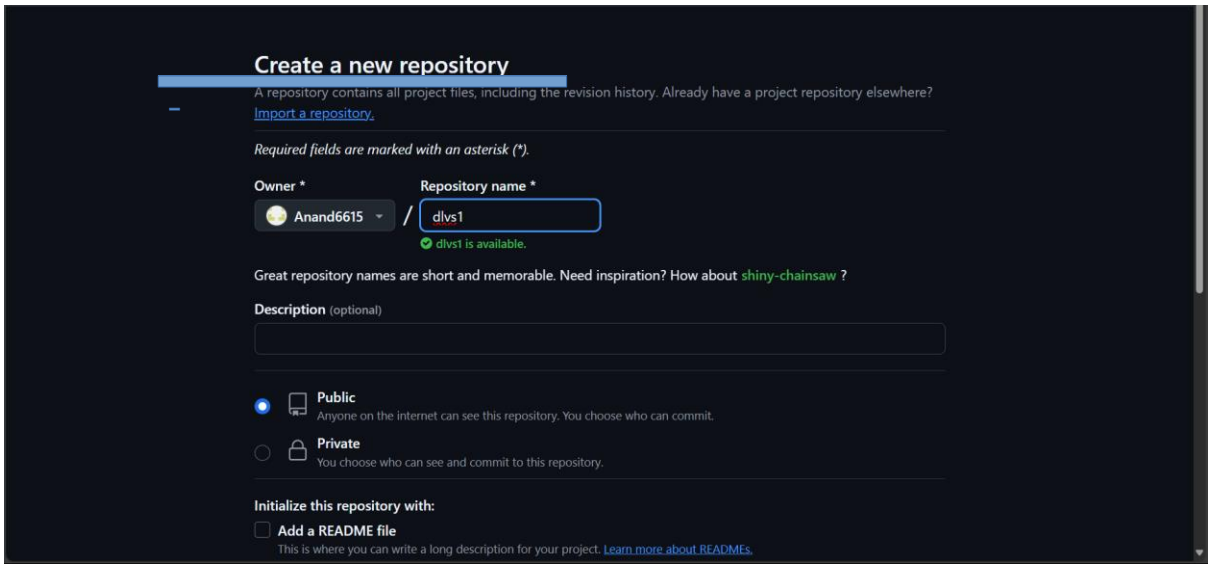
12. **git log**



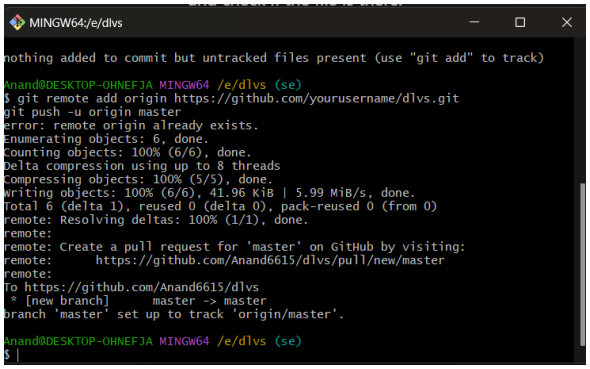

13. **git revert**



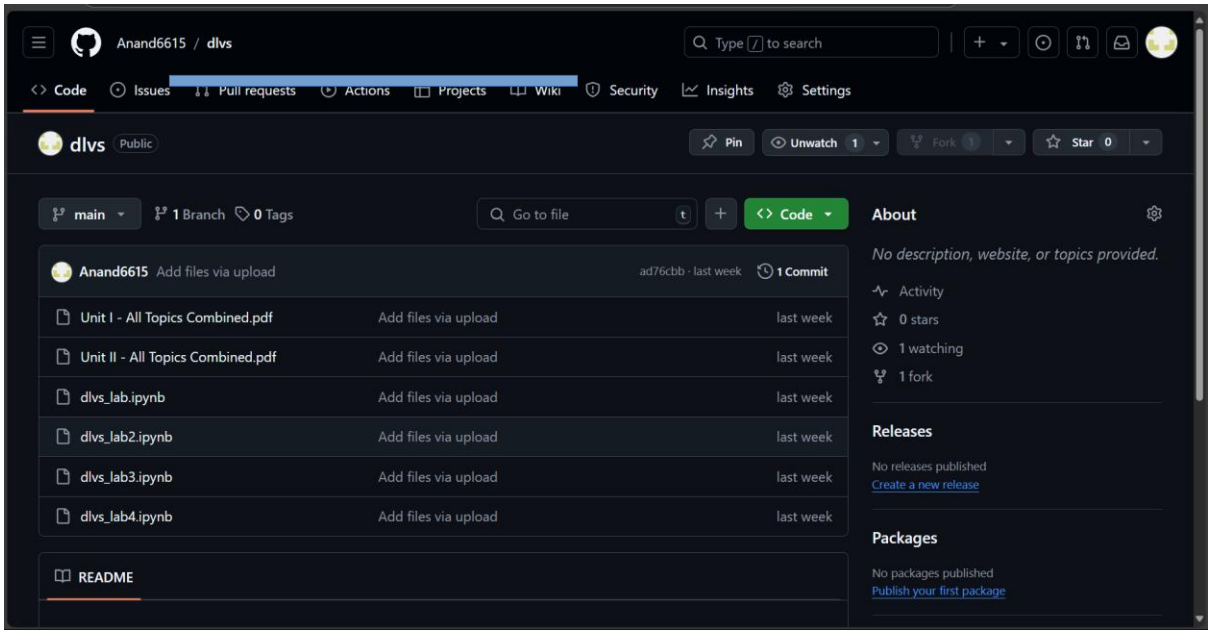

14. **Create new GitHub repository**



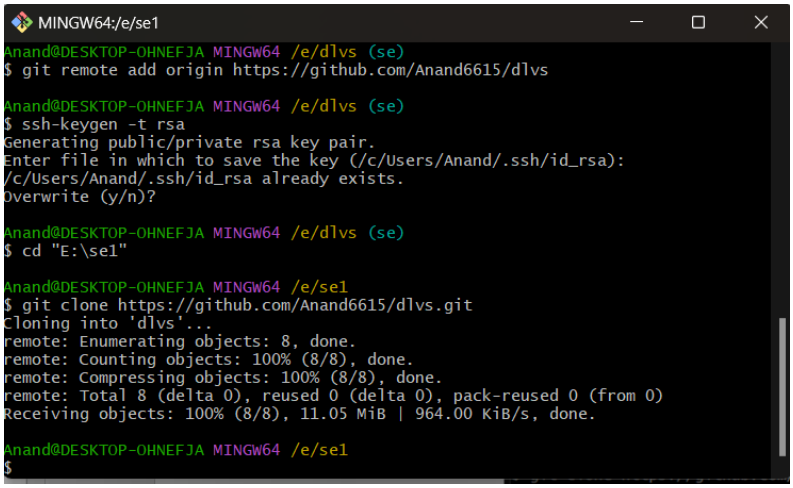
15. **git push**




16. **After git push in GitHub**

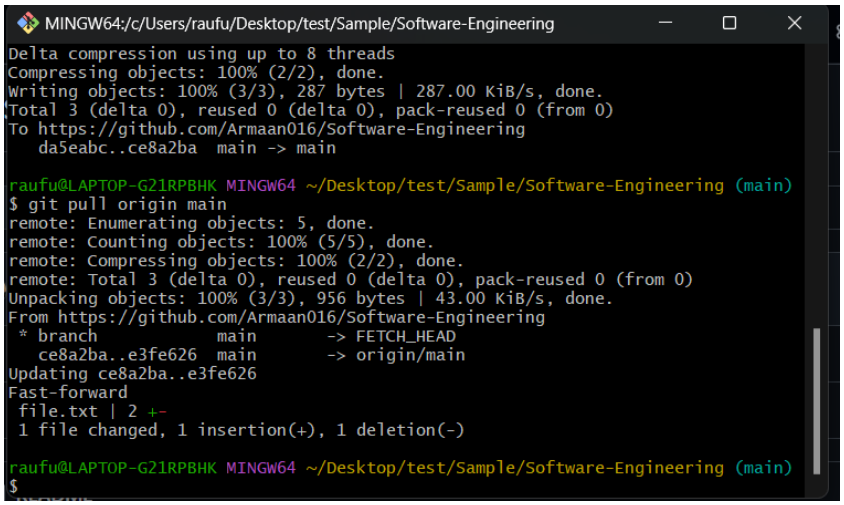



17. **git clone**

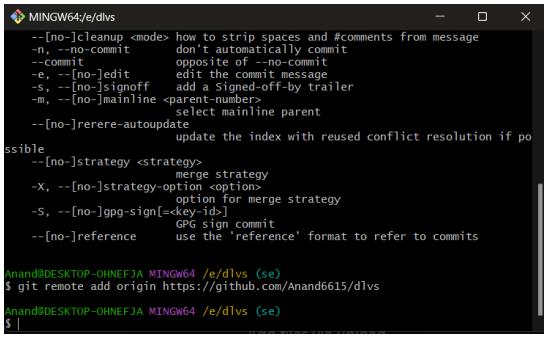



18. After **git clone**

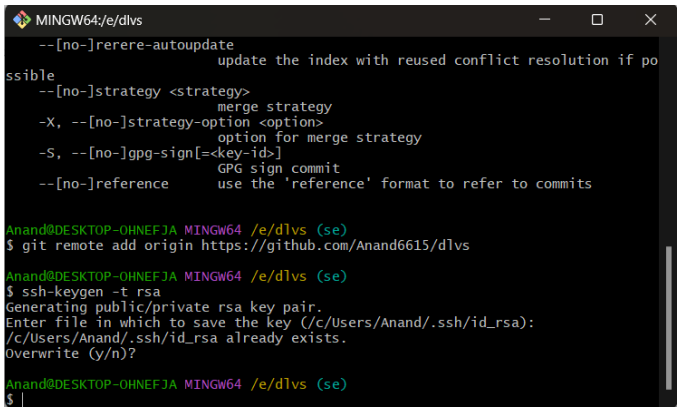
19. **git pull**

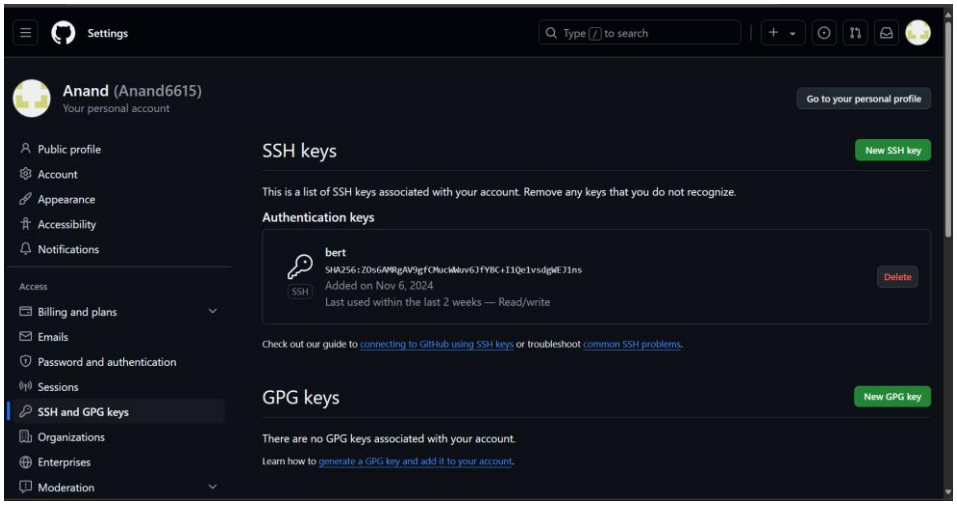



20. **git remote -v**

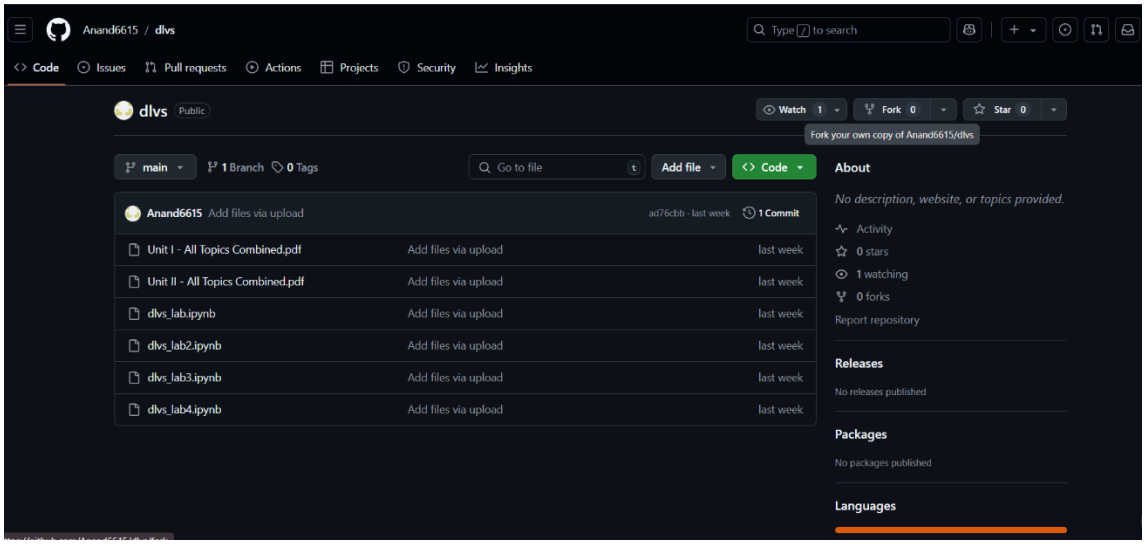



# 21. Adding SSH Keys

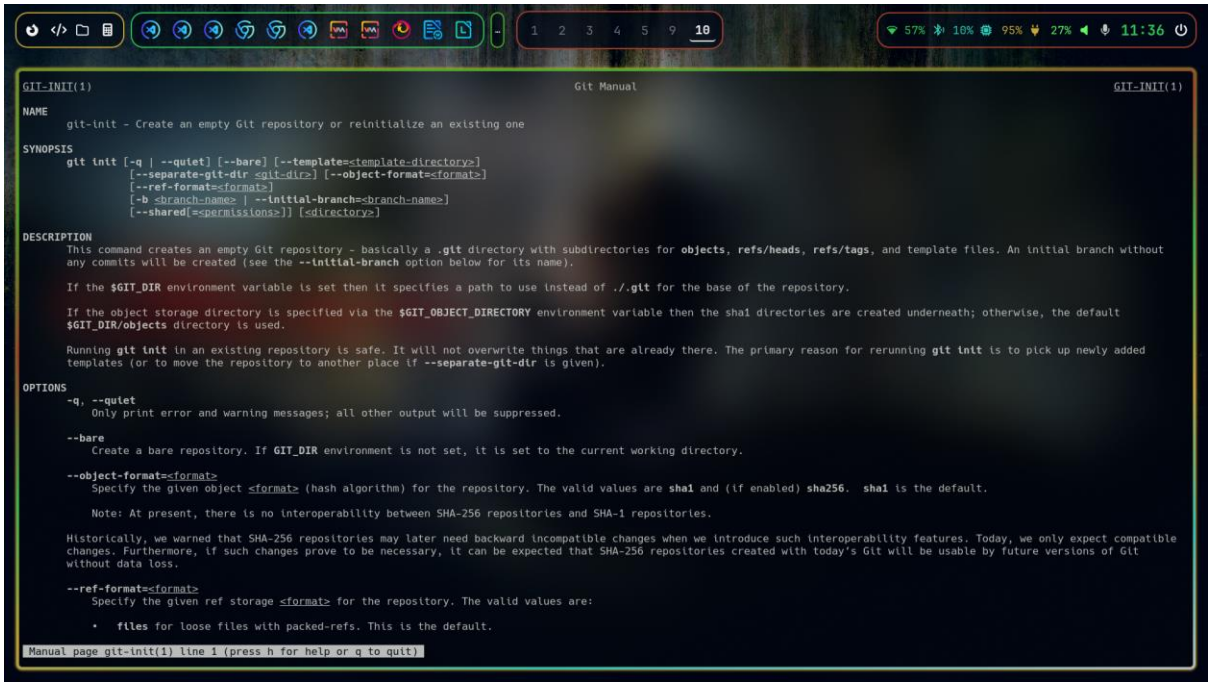



# 22. Git Fork


Anand6615 / dlvs
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
dlvs
Public
Watch 1
Fork 0
Star 0
Fork your own copy of Anand6615/dlvs
main
1 Branch
0 Tags
Go to file
t
Add file
Code
About
No description, website, or topics provided.
Anand6615 Add files via upload
ad76cbb · last week
1 Commit
Unit I - All Topics Combined.pdf
Add files via upload
last week
Unit II - All Topics Combined.pdf
Add files via upload
last week
dlvs_lab.ipynb
Add files via upload
last week
dlvs_lab2.ipynb
Add files via upload
last week
dlvs_lab3.ipynb
Add files via upload
last week
dlvs_lab4.ipynb
Add files via upload
last week
Activity
0 stars
1 watching
0 forks
Report repository
Releases
No releases published
Packages
No packages published
Languages

## 23. **git help**




Result: Git commands were explored, a project was successfully created using Git, and the repository was pushed to GitHub for collaborative development.

## Experiment-5: Create Maven Java and Web Projects and Push them to GitHub

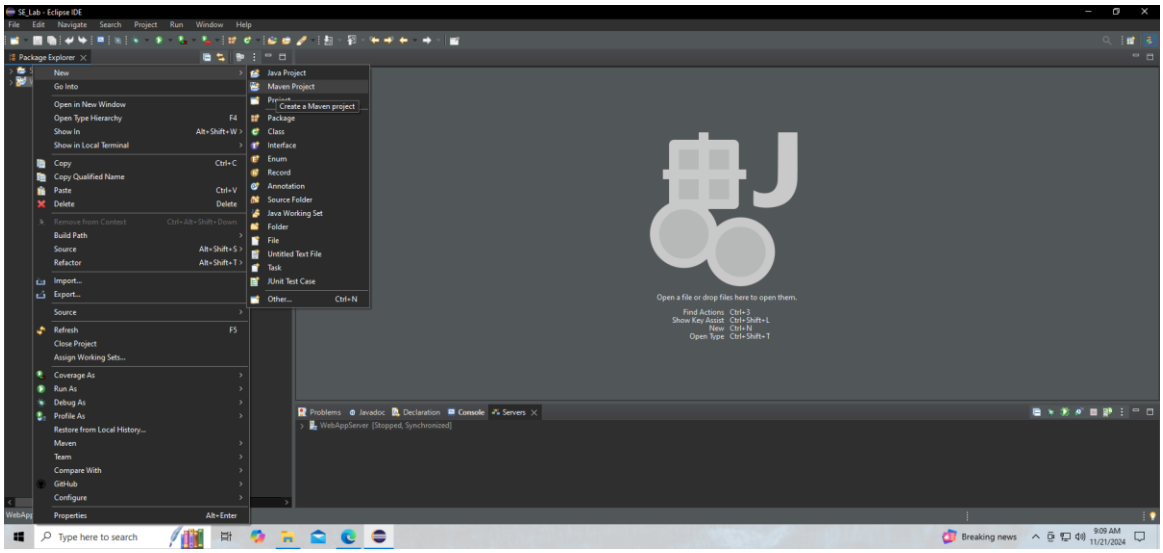
Aim: To install Maven, Tomcat, and Eclipse, configure Tomcat with Eclipse, create a Java and web project using Maven in Eclipse, and push the project to GitHub.

Introduction:

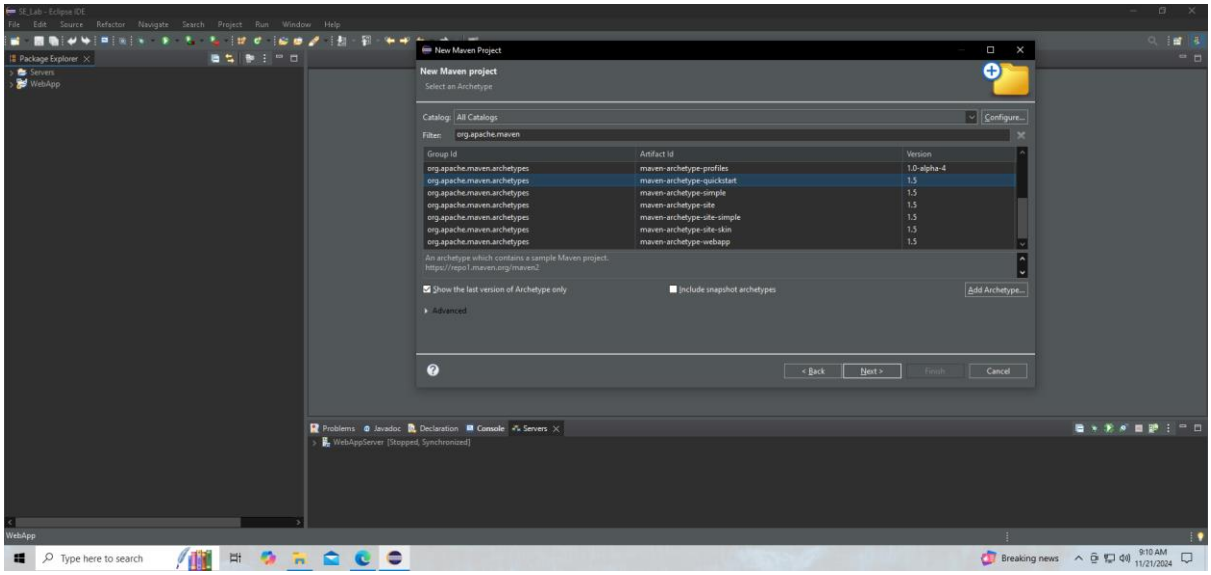
1. Maven: A powerful project management tool that simplifies the build process by automating dependency management, project structure setup, and build configurations. It follows a declarative approach through a pom.xml file.
2. Tomcat: An open-source web server and servlet container that supports the deployment of Java-based web applications. Configuring Tomcat with Eclipse enables seamless integration for development and testing.
3. Eclipse: A widely used Integrated Development Environment (IDE) for Java and other programming languages. It provides robust tools for creating, managing, and debugging applications.
4. Workflow Overview:
    - Install Maven, Tomcat, and Eclipse on the system.
    - Configure Tomcat with Eclipse for efficient web application development.
    - Use Maven to create a Java and web project in Eclipse, benefiting from dependency and build management.
    - Push the completed project to GitHub for version control and collaboration.

# Creating Maven Java Project and pushing into GitHub

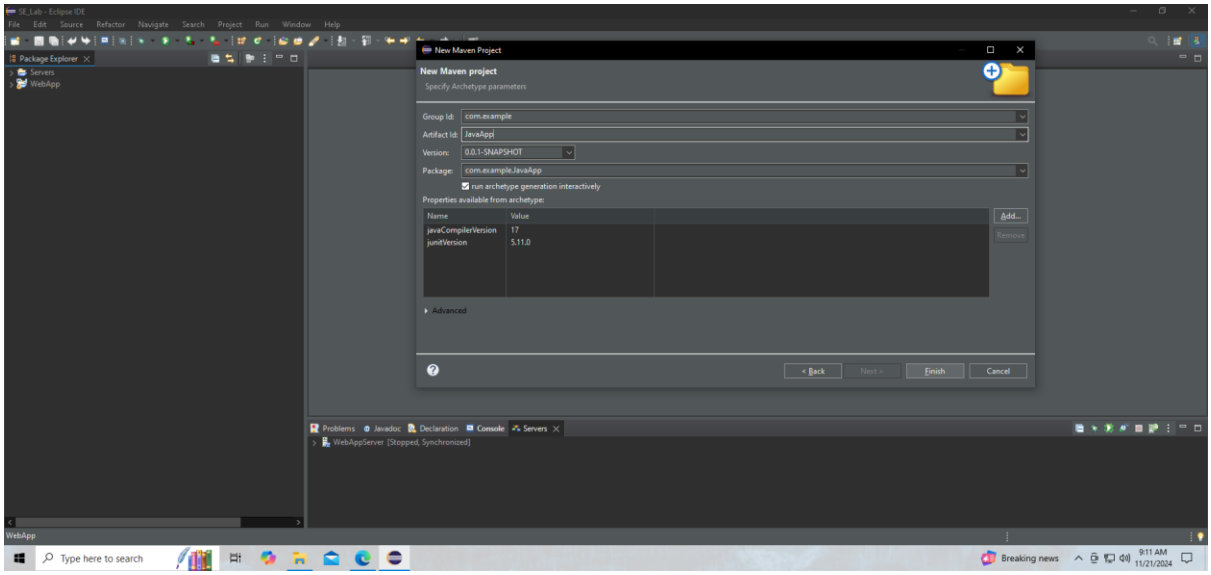
1. Creating new **Maven Project**



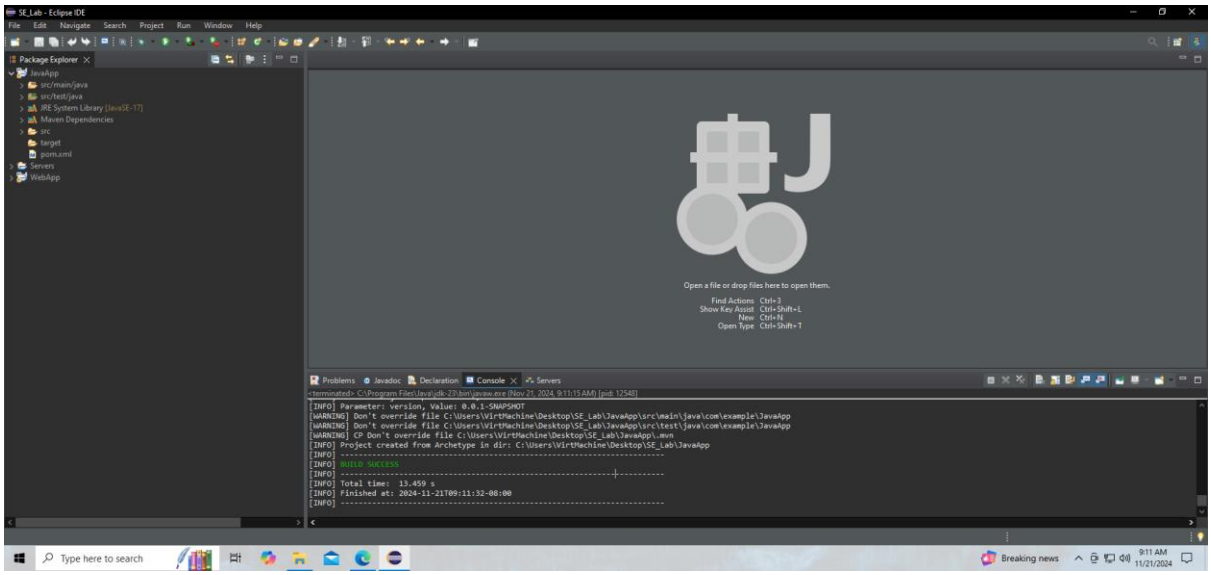

2. Selecting **maven-archetype-quickstart**



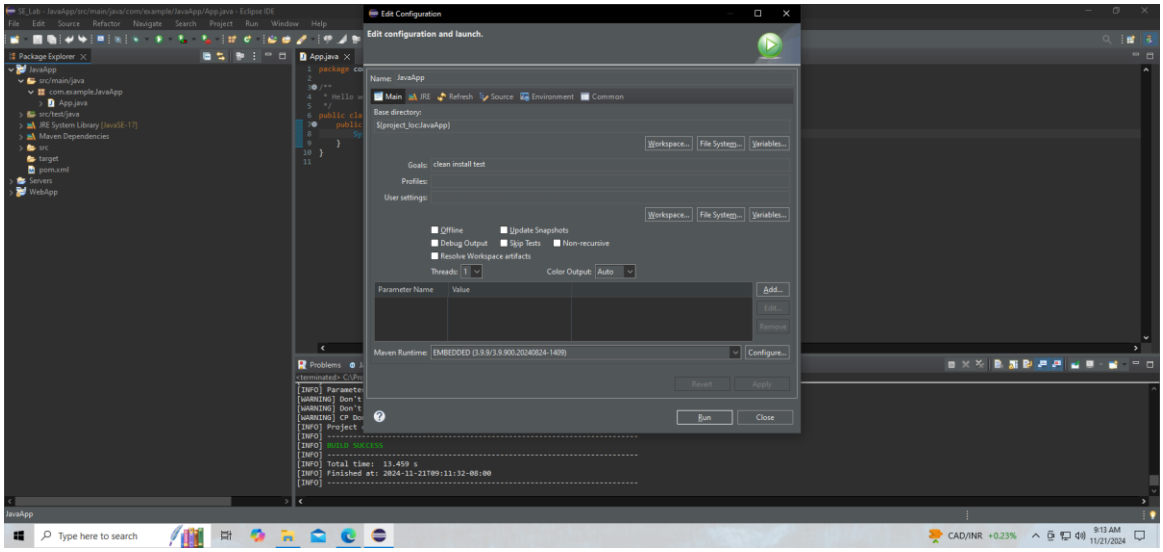

3. Creating new **Group Id** and **Artifact Id**



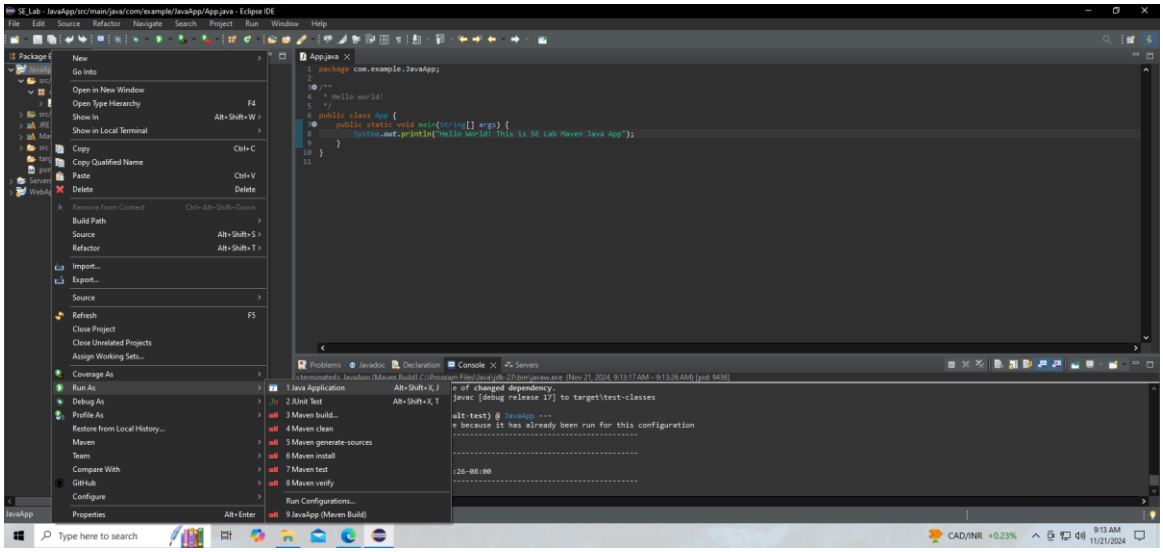

4. Project created



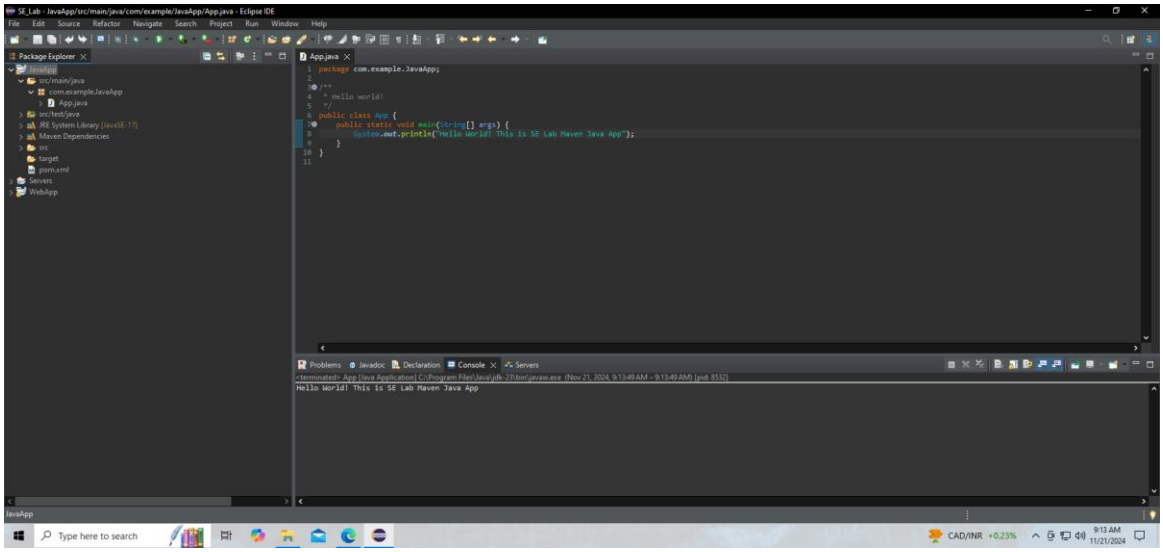

5. Creating **Maven Build** configuration



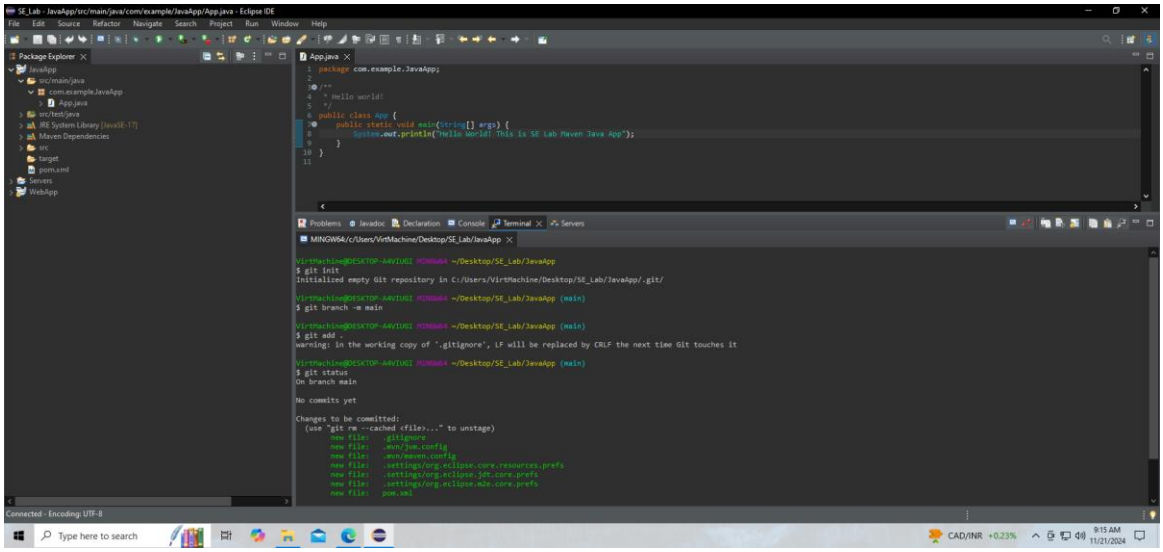

6. Run as **Java Application**

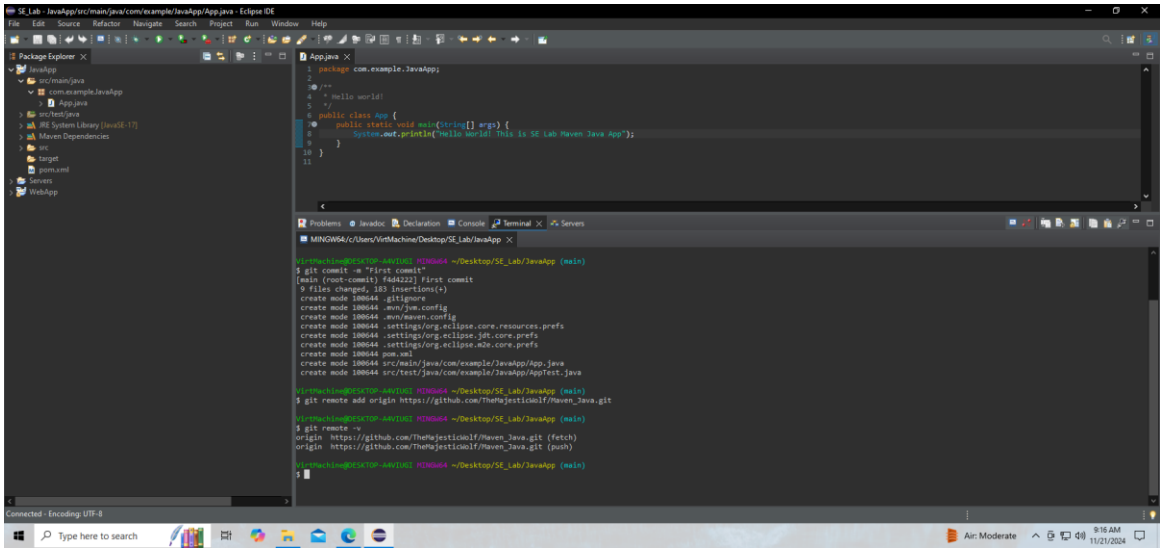



7. Output after running program

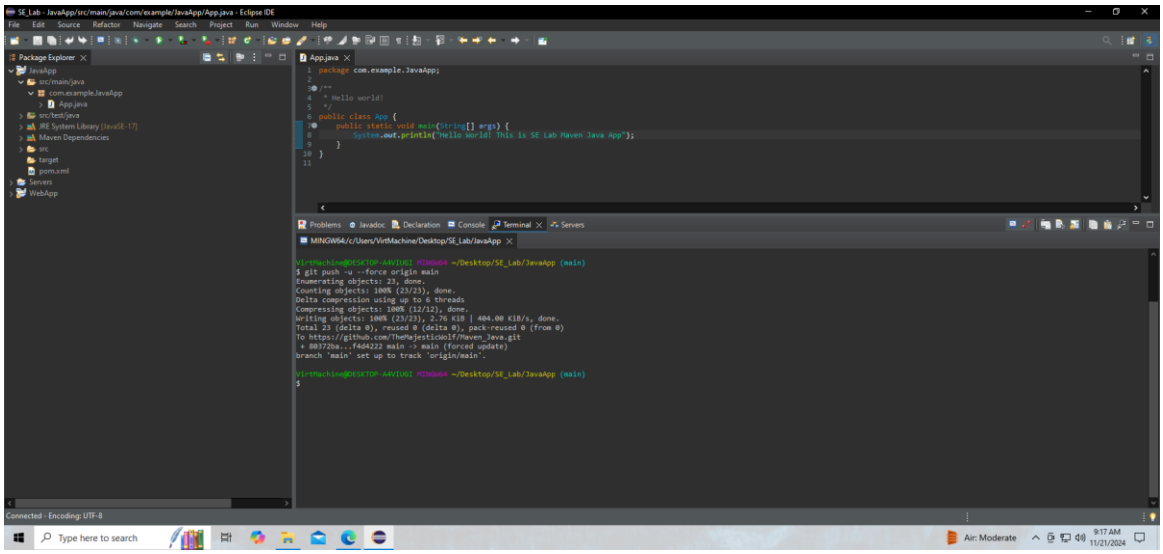



## 9. Pushing to **GitHub**

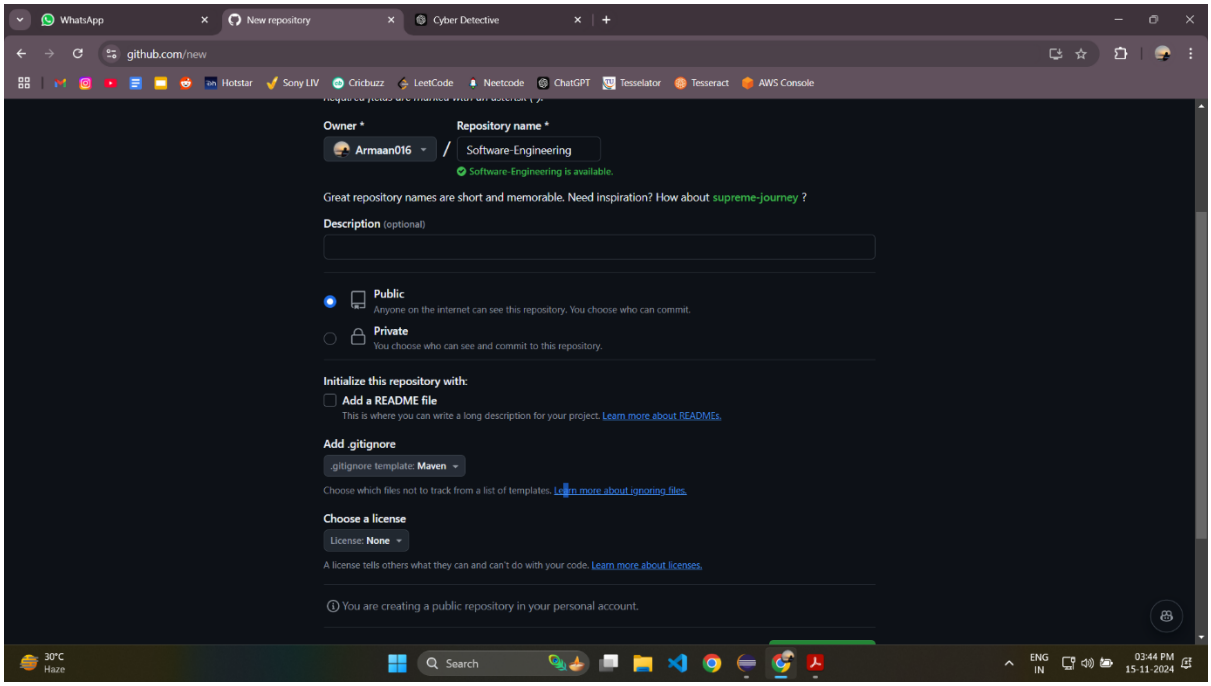




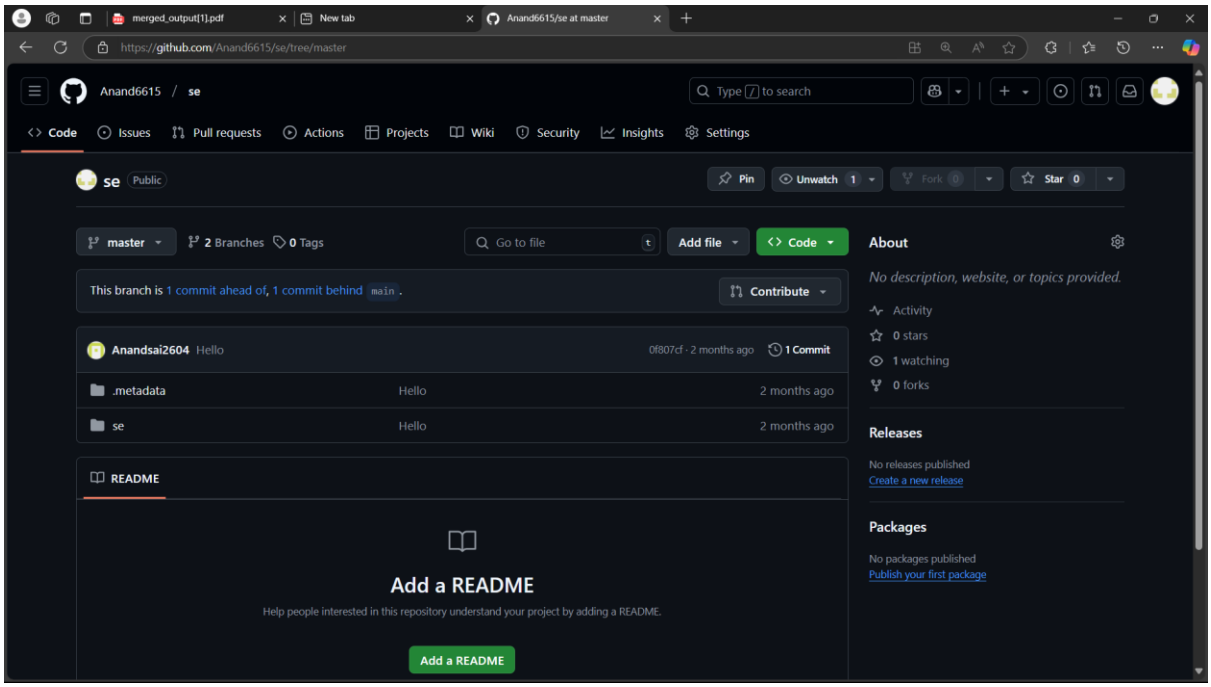

11. Successfully pushed into **GitHub**



# 12. Creating new GitHub repository

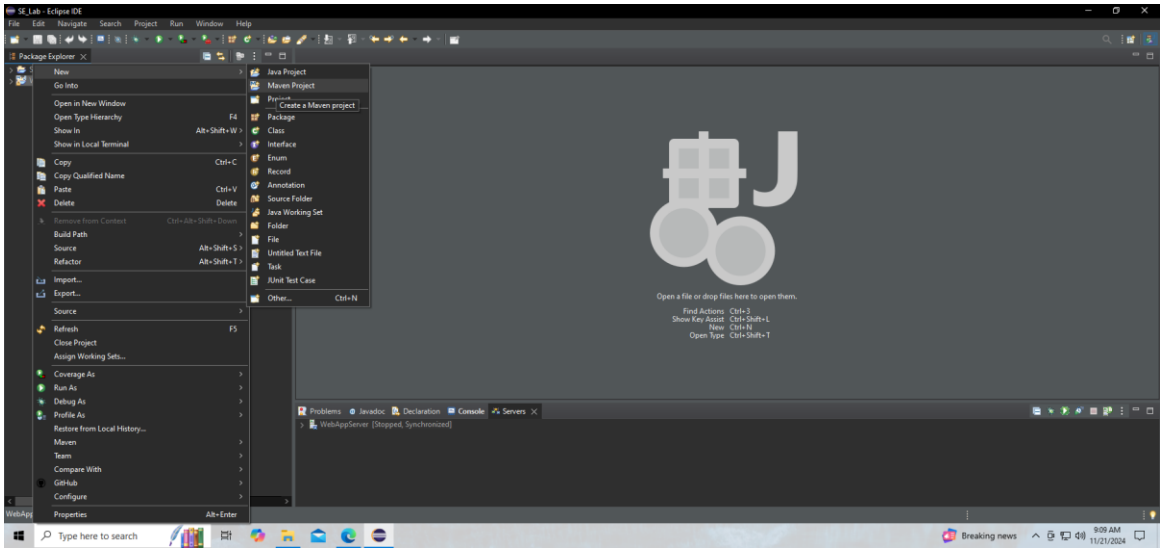



# 13. After pushing

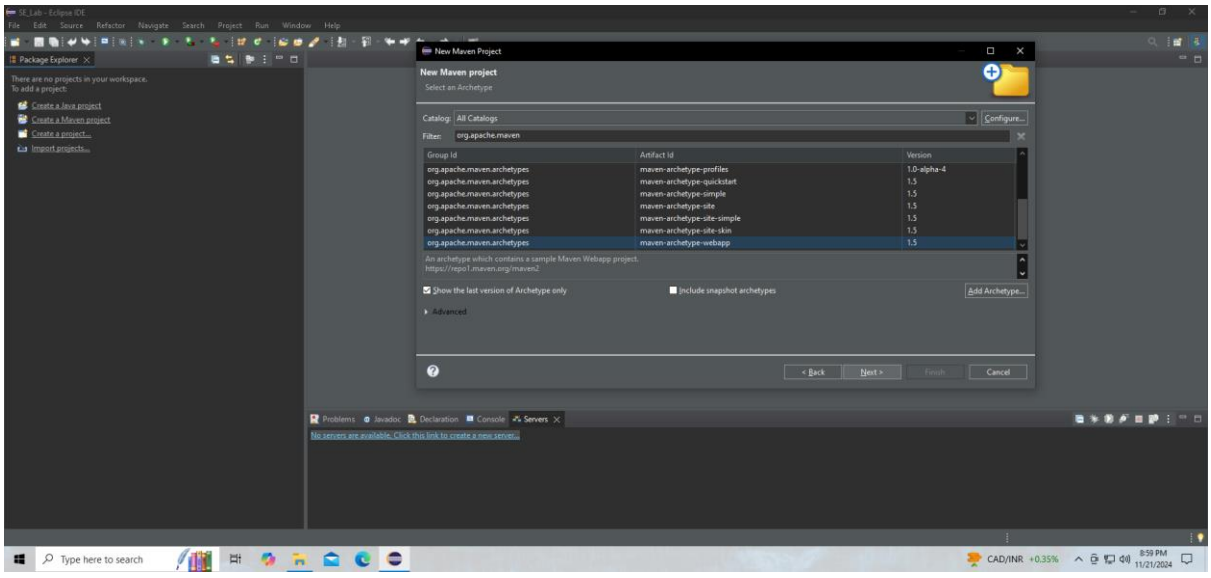


# Creating Maven Web App and pushing into GitHub

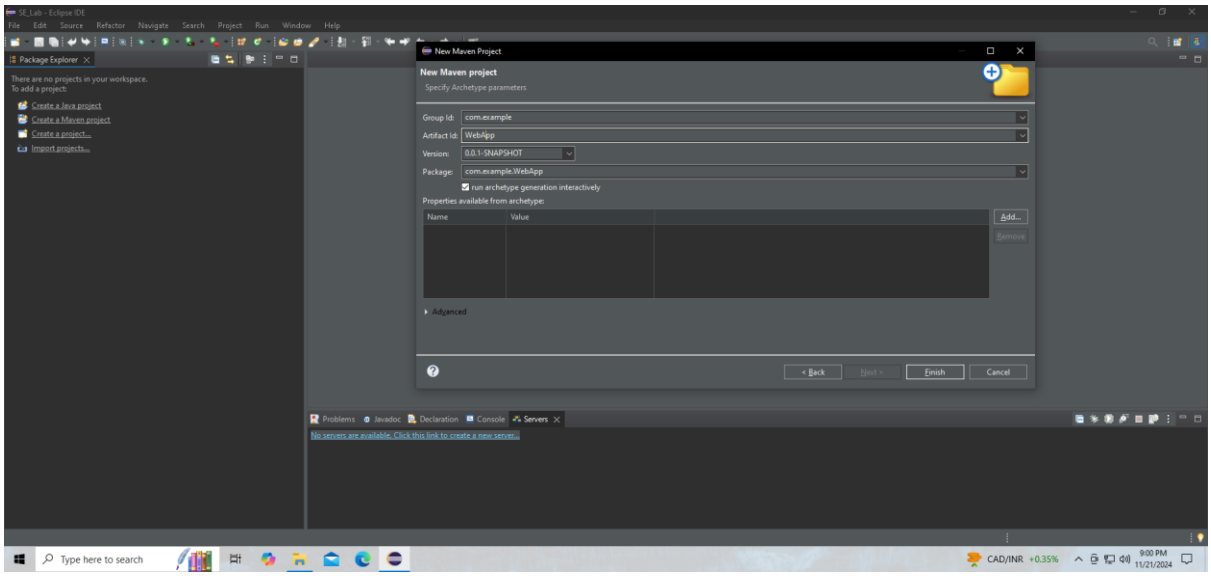
1. Creating new **Maven Project**



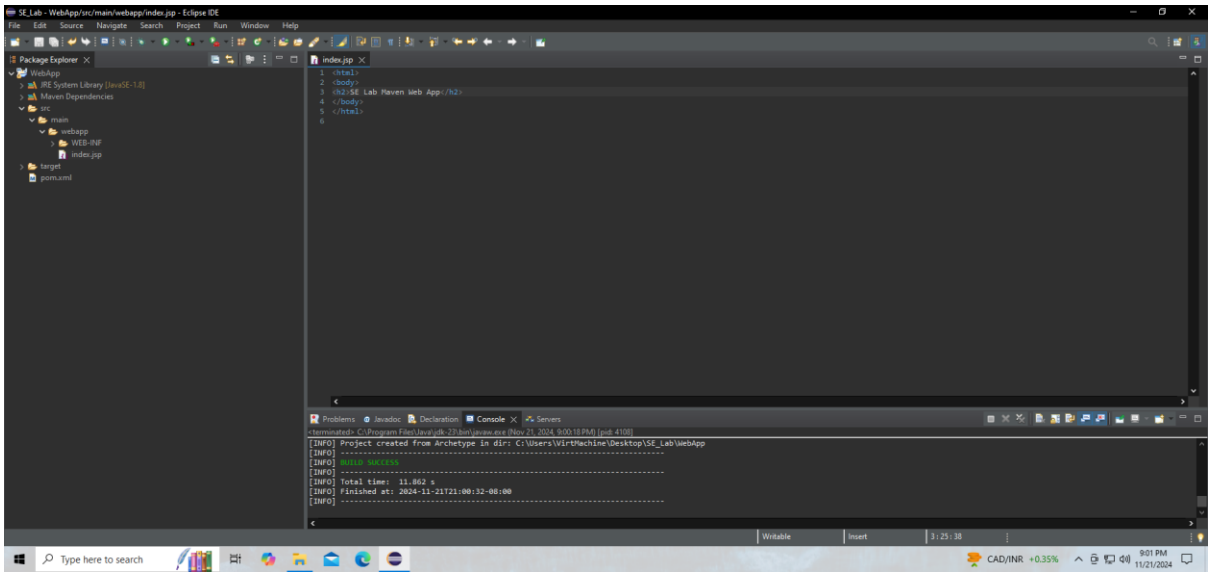

2. Selecting **maven-archetype-webapp**



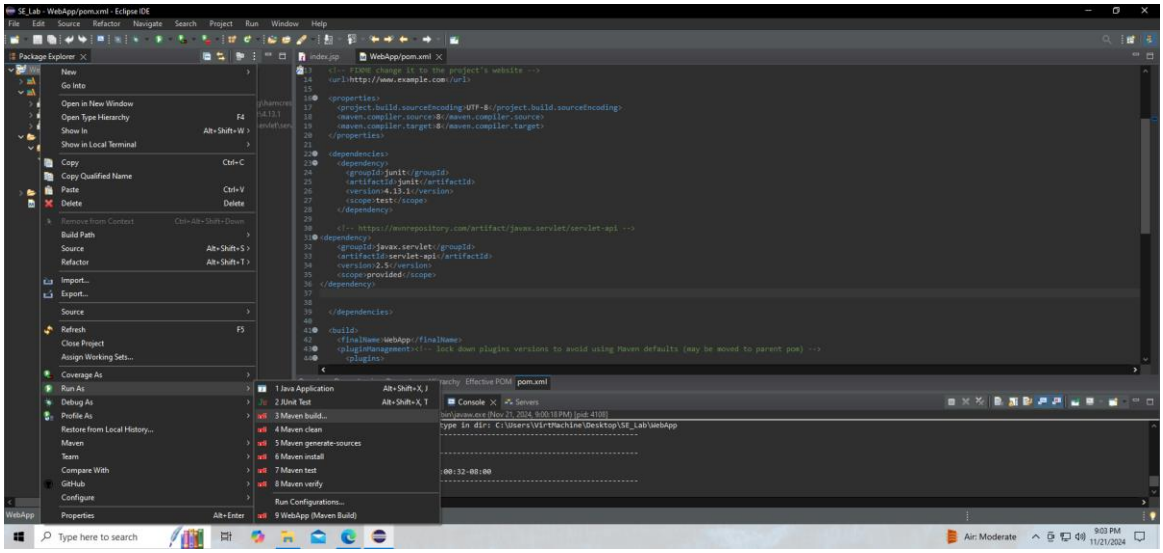
3. Creating new **Group Id** and **Artifact Id**



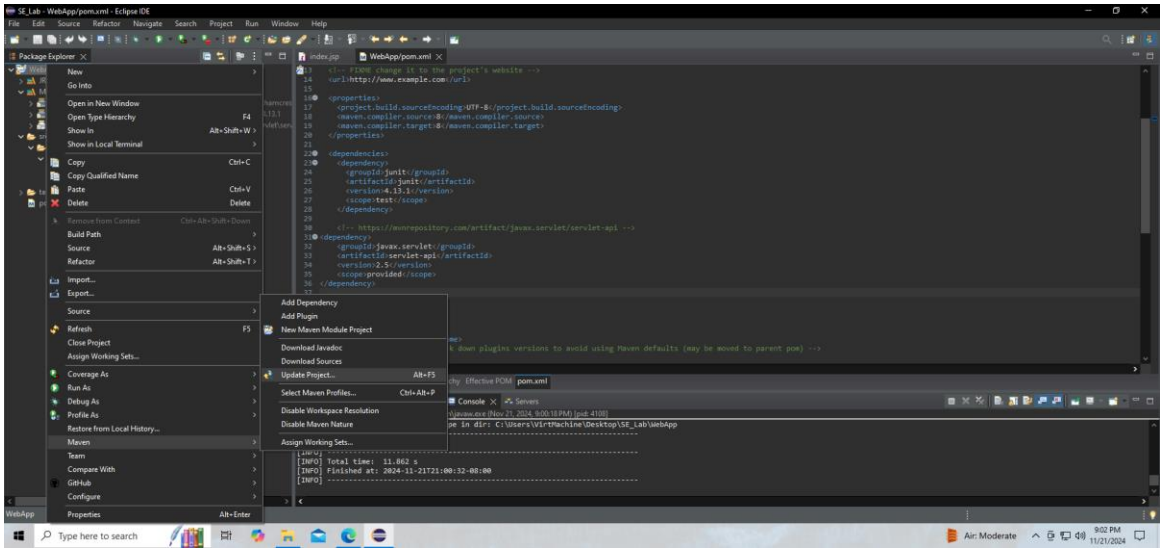

4. Project created



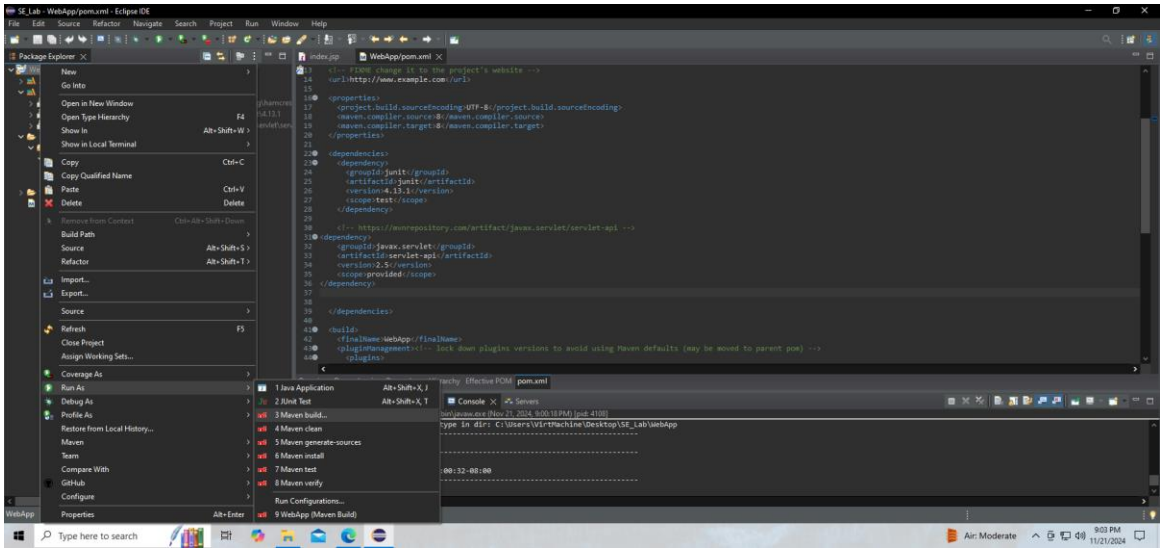
5. Add **servlet-api** dependency to **pom.xml**



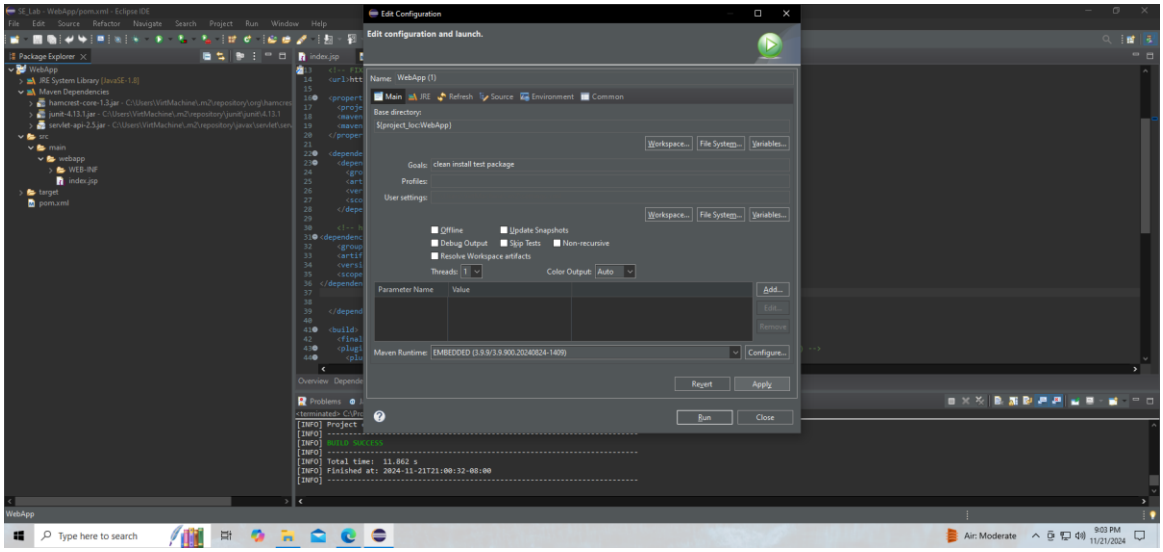
6. Click on **Update Project**



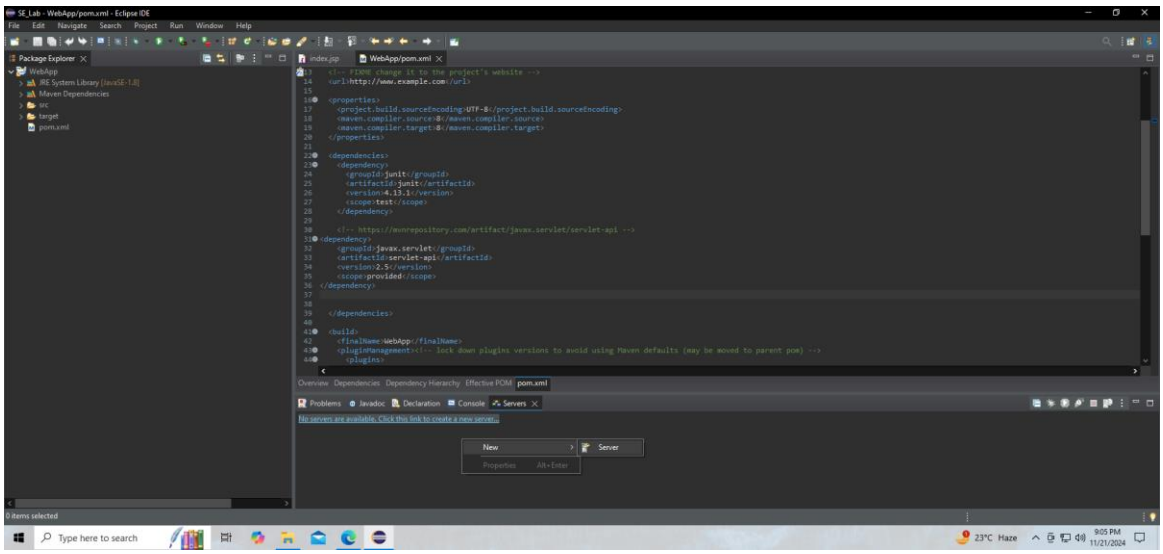
7. Click on **Maven build**



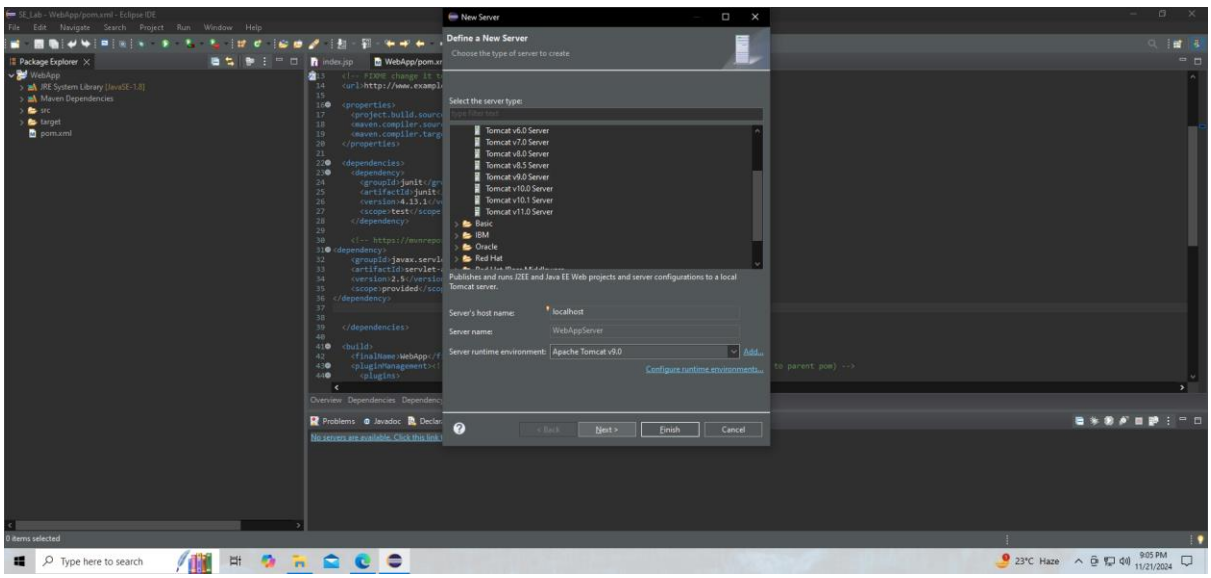
9. Add **clean install test package** and then run it



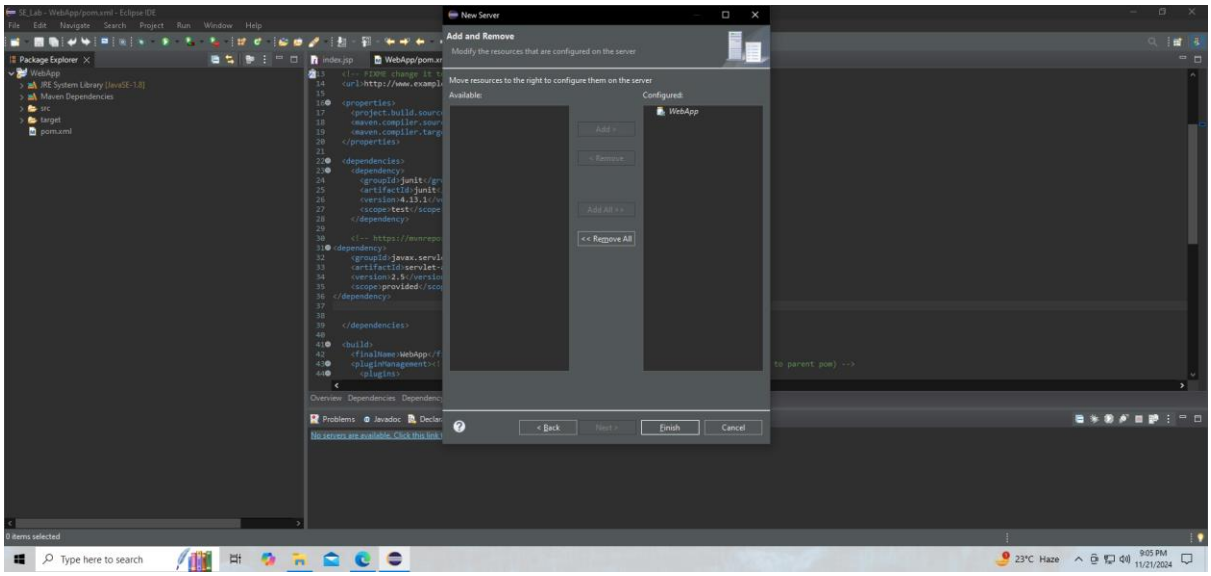
11. Click on **Servers** tab and create new server



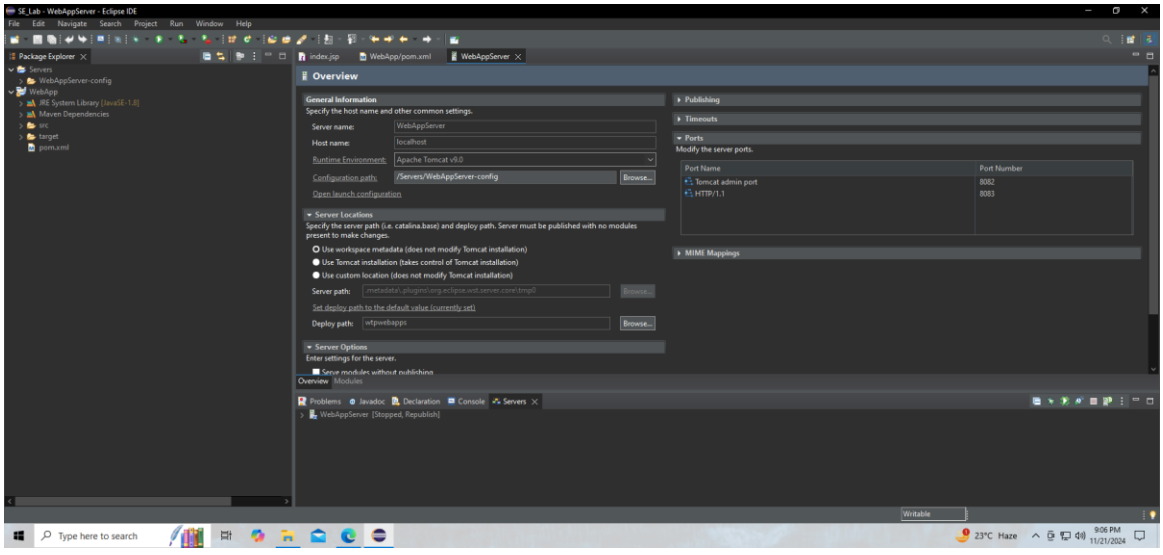
12. Select **Tomcat v9.0 Server** and click **Next**



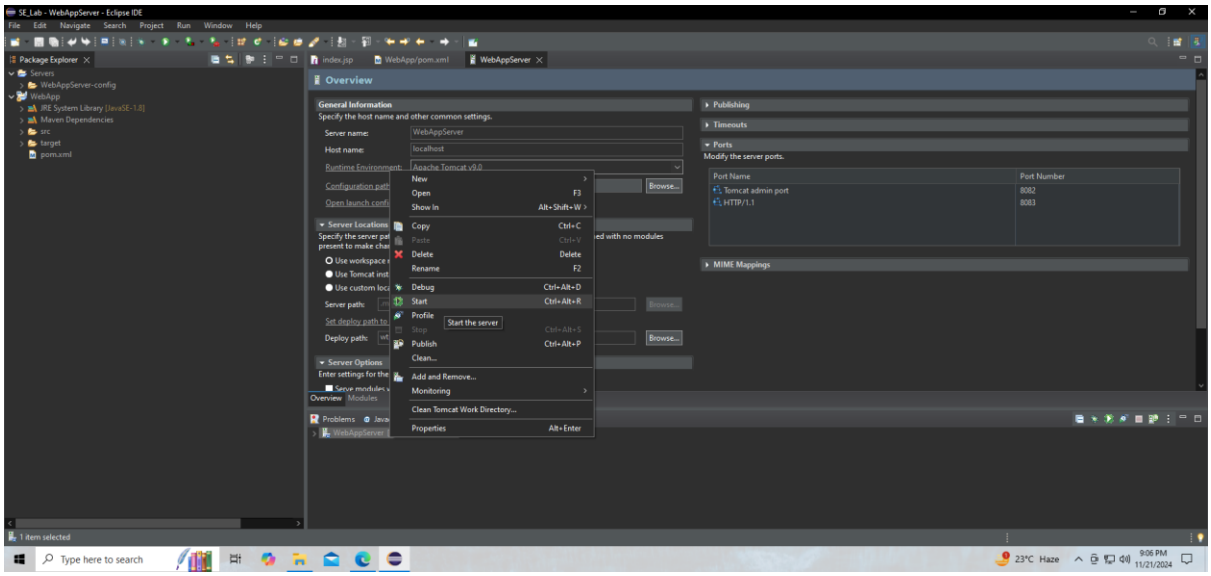
13. Click on your project and click Add, then Finish



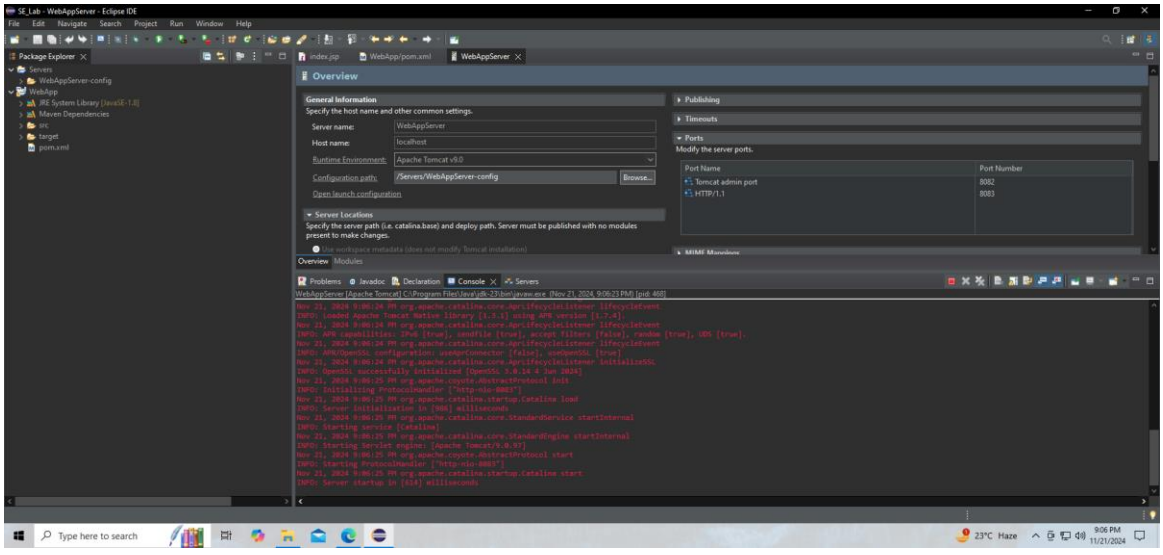

13. Tomcat Server Added



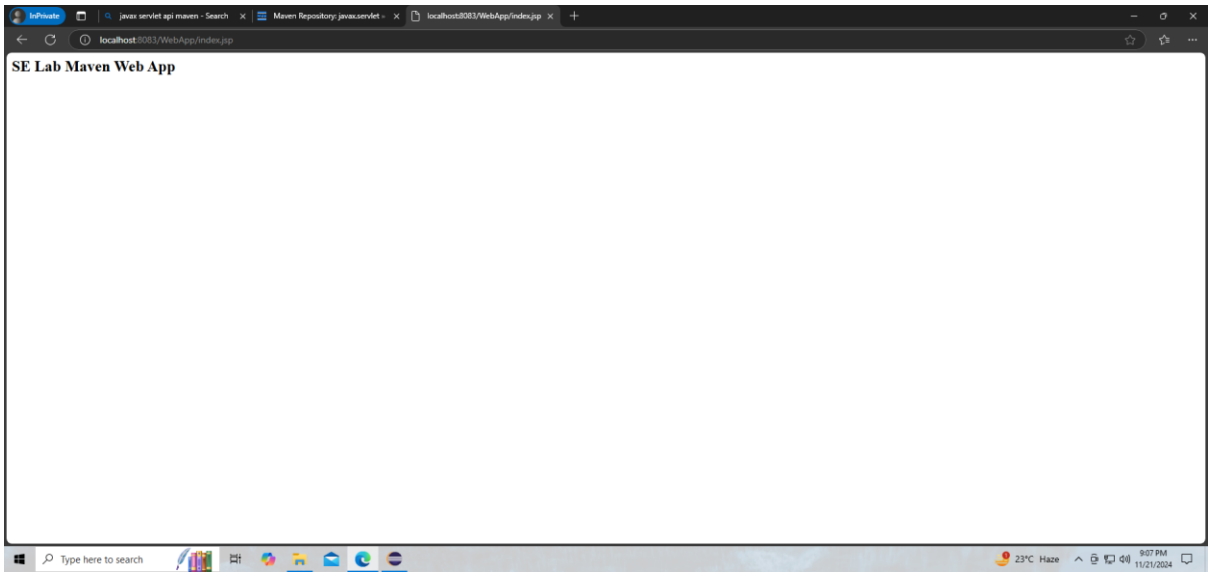
13. Click on **Start**



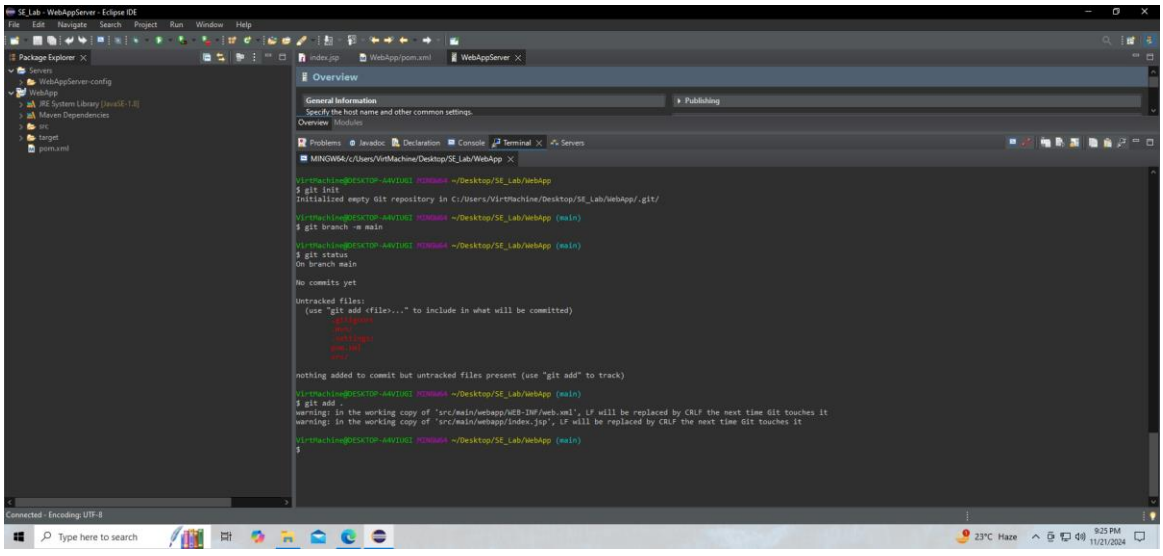

13. Tomcat Server started

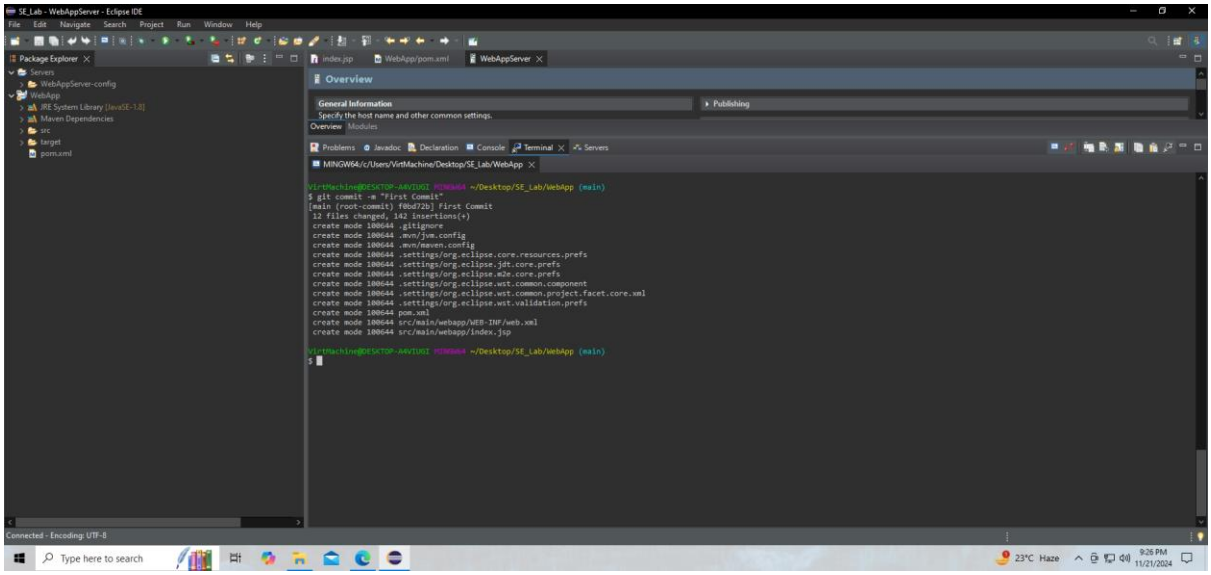


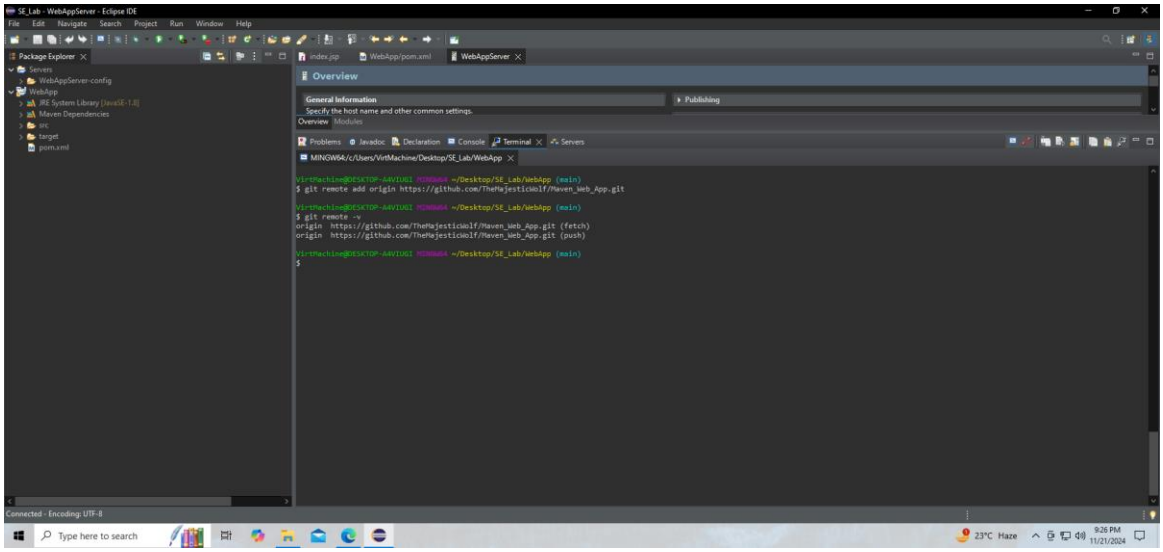
13. Visit **localhost:8083/WebApp/index.jsp**

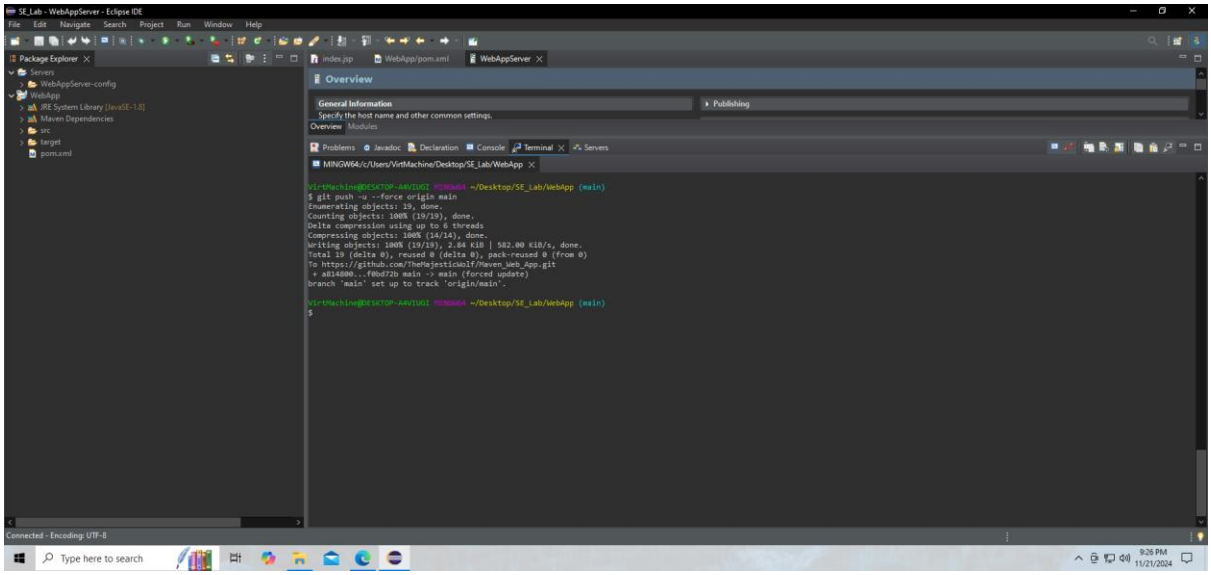


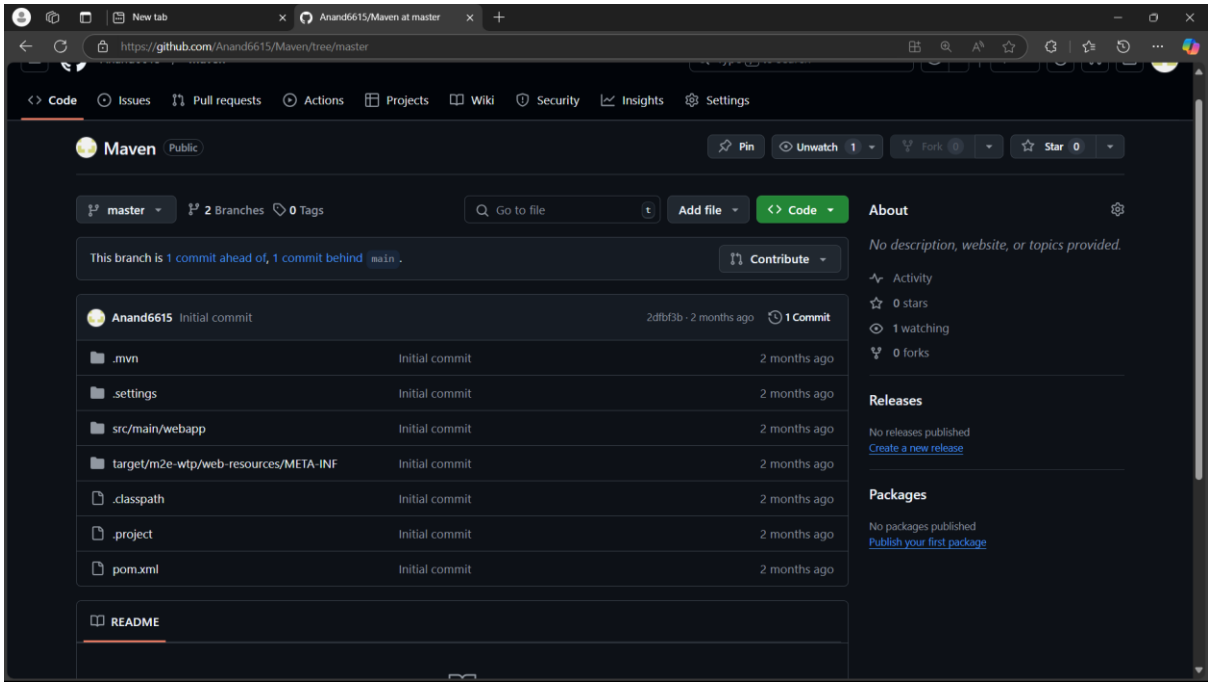

13. Pushing into **GitHub**




SE_Lab - WebAppServer - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Servers
WebAppServer-config
WebApp
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
pom.xml
index.jsp
WebApp/pom.xml
WebAppServer
Overview
General Information
Specify the host name and other common settings.
Publishing
Overview Modules
Problems Javadoc Declaration Console Terminal Servers
MINGW64:/c/Users/VirtMachine/Desktop/SE_Lab/WebApp
VirtMachine@DESKTOP-A4V1UGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$ git commit -m "First Commit"
[main (root-commit) f0bd72b] First Commit
12 files changed, 142 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/jvm.config
create mode 100644 .mvn/maven.config
create mode 100644 .settings/org.eclipse.core.resources.prefs
create mode 100644 .settings/org.eclipse.jdt.core.prefs
create mode 100644 .settings/org.eclipse.m2e.core.prefs
create mode 100644 .settings/org.eclipse.wst.common.component
create mode 100644 .settings/org.eclipse.wst.common.project.facet.core.xml
create mode 100644 .settings/org.eclipse.wst.validation.prefs
create mode 100644 pom.xml
create mode 100644 src/main/webapp/WEB-INF/web.xml
create mode 100644 src/main/webapp/index.jsp
VirtMachine@DESKTOP-A4V1UGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$
Connected - Encoding: UTF-8
Type here to search
23°C Haze
9:26 PM
11/21/2024


SE_Lab - WebAppServer - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Servers
WebAppServer-config
WebApp
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
pom.xml
index.jsp
WebApp/pom.xml
WebAppServer
Overview
General Information
Specify the host name and other common settings.
Publishing
Overview Modules
Problems Javadoc Declaration Console Terminal Servers
MINGW64:/c/Users/VirtMachine/Desktop/SE_Lab/WebApp
VirtMachine@DESKTOP-A4VIUGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$ git remote add origin https://github.com/TheHajesticWolf/Maven_Web_App.git
VirtMachine@DESKTOP-A4VIUGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$ git remote -v
origin https://github.com/TheHajesticWolf/Maven_Web_App.git (fetch)
origin https://github.com/TheHajesticWolf/Maven_Web_App.git (push)
VirtMachine@DESKTOP-A4VIUGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$
Connected - Encoding: UTF-8
Type here to search
23°C Haze
9:26 PM
11/21/2024


SE_Lab - WebAppServer - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Servers
WebAppServer-config
WebApp
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
pom.xml
index.jsp
WebApp/pom.xml
WebAppServer
Overview
General Information
Specify the host name and other common settings.
Publishing
Overview Modules
Problems Javadoc Declaration Console Terminal Servers
MINGW64:/c/Users/VirtMachine/Desktop/SE_Lab/WebApp
VirtMachine@DESKTOP-A4VIUGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$ git push -u --force origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 6 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (19/19), 2.84 KiB | 582.00 KiB/s, done.
Total 19 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/TheHajesticWolf/Maven_Web_App.git
+ a814800...f0bd72b main -> main (forced update)
branch 'main' set up to track 'origin/main'.
VirtMachine@DESKTOP-A4VIUGI MINGW64 ~/Desktop/SE_Lab/WebApp (main)
$
Connected - Encoding: UTF-8
Type here to search
9:26 PM
11/21/2024




Result: Maven Java and web projects were created and pushed to GitHub.

# Experiment-6: Building a CI/CD pipeline using Jenkins for projects created in Experiment – 5

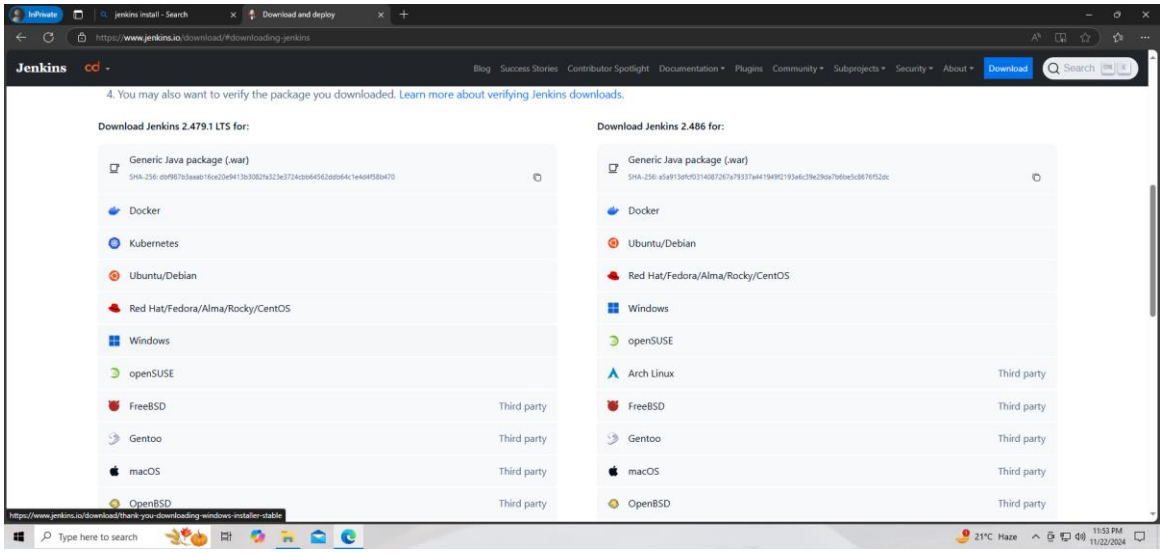
Aim: To install Jenkins and use it for Continuous Integration and Continuous Deployment (CI/CD) of the Maven Java and web projects created in the previous experiment.

Introduction:

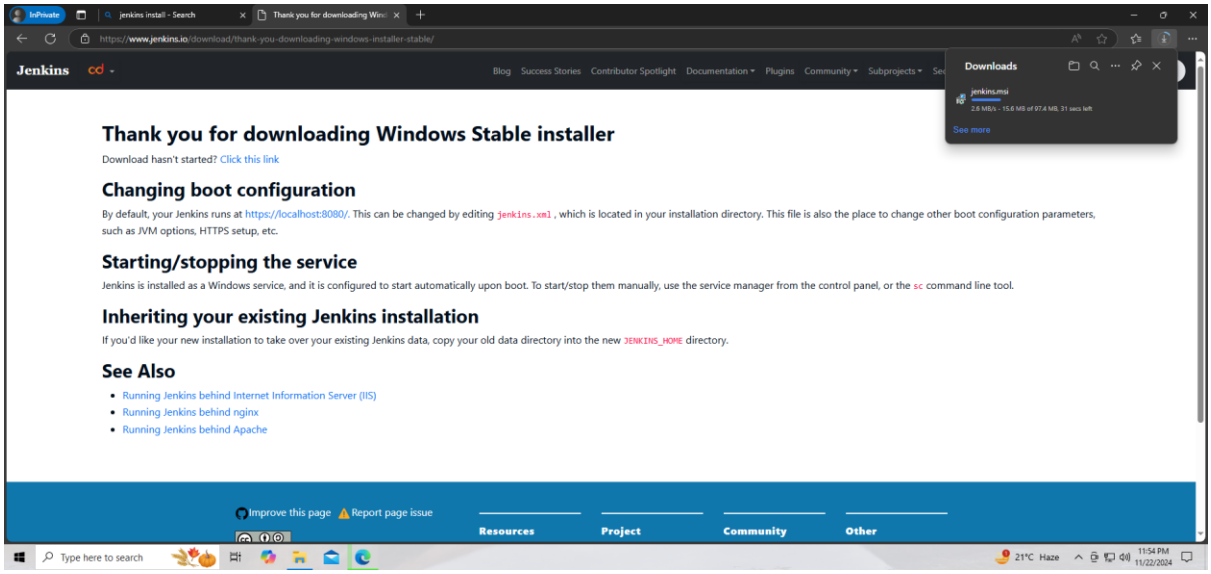
1. Jenkins: Jenkins is an open-source automation server widely used for implementing Continuous Integration (CI) and Continuous Deployment (CD). It automates building, testing, and deploying applications, thereby enhancing software delivery efficiency and reliability.
2. Continuous Integration (CI):
    - Automatically integrates code changes from multiple contributors.
    - Triggers builds and tests upon code updates to ensure stability and detect issues early.
3. Continuous Deployment (CD):
    - Extends CI by automating the release of stable builds to production environments.
    - Reduces manual intervention and accelerates deployment cycles.

# Installation of Jenkins

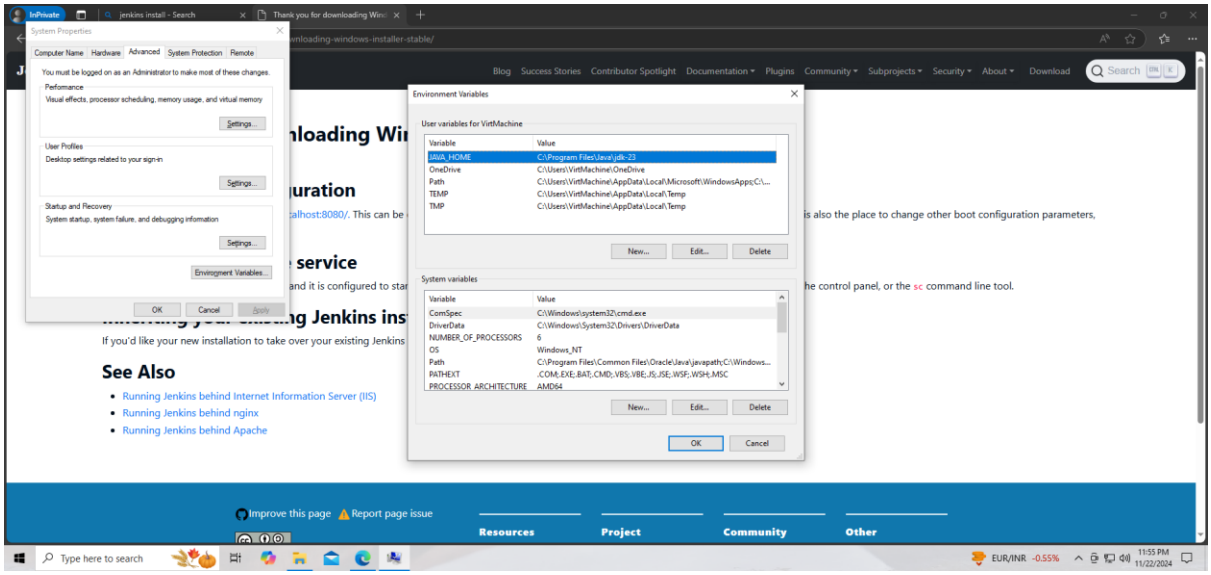
1. Download the Jenkins installer from the official website



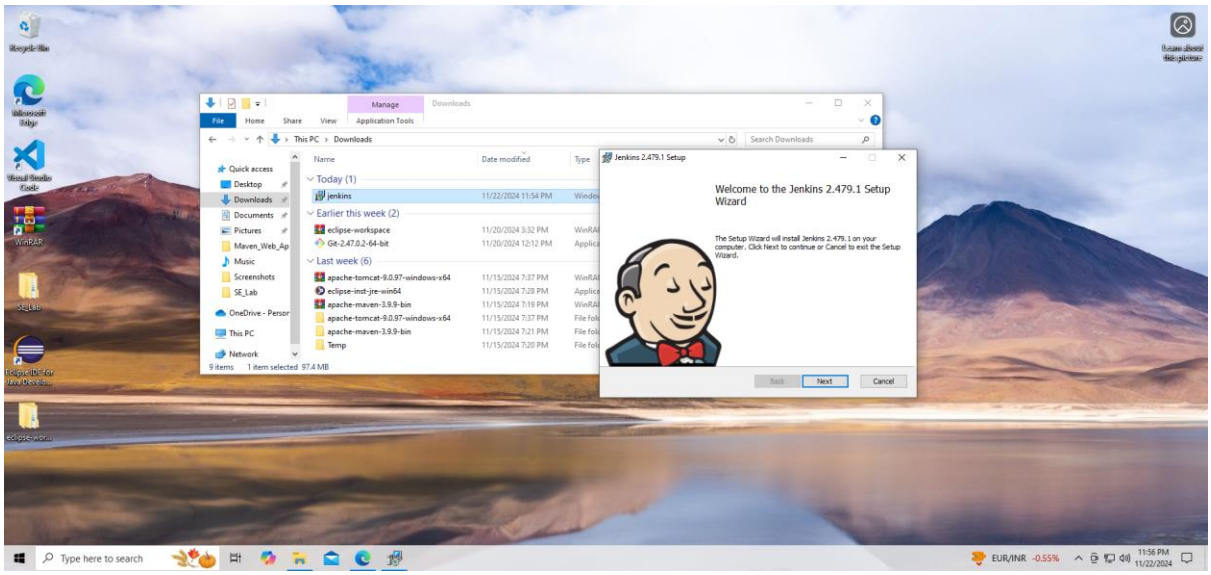

2. Wait for the download to complete



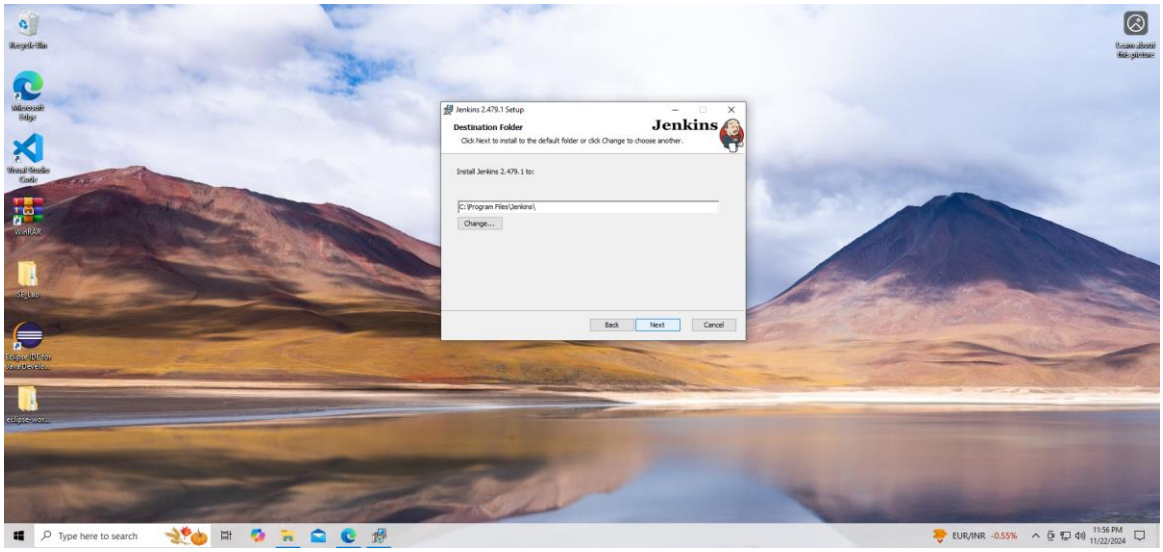

3. Add JAVA_HOME environment variable if not added

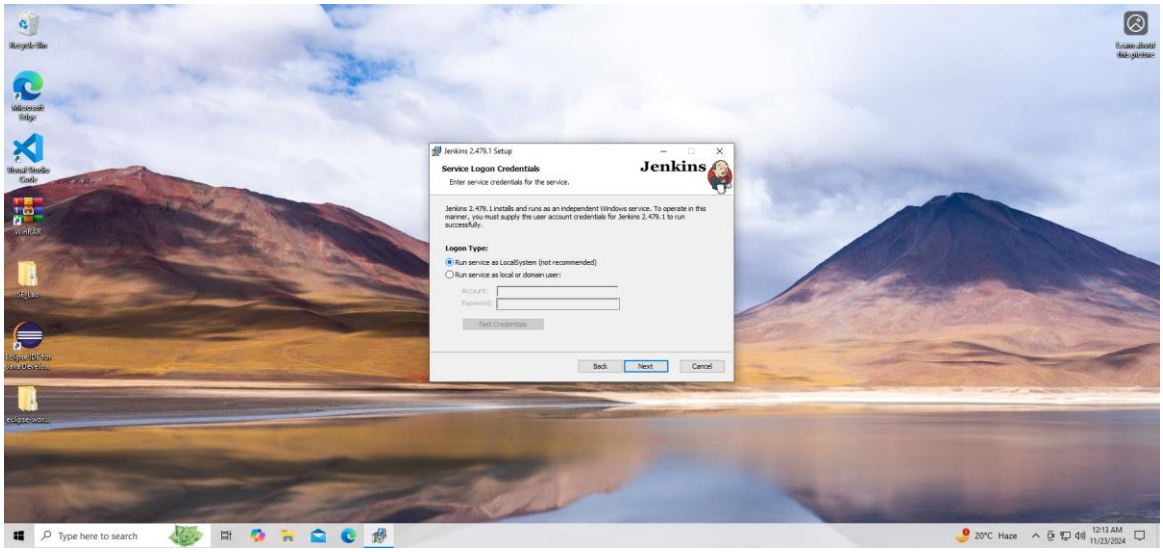


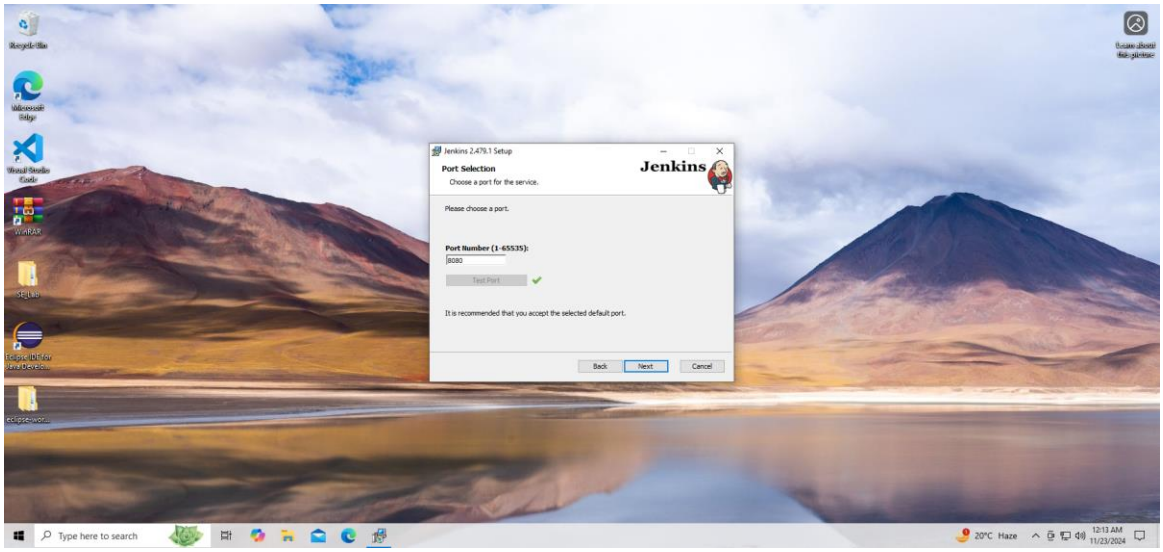

## 4. Run the Jenkins installer

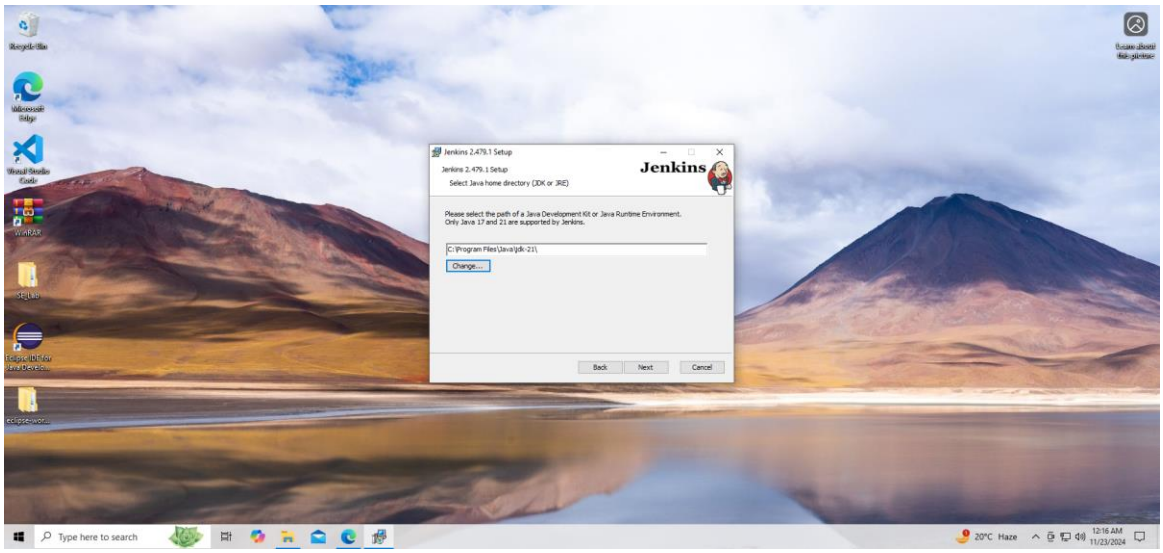


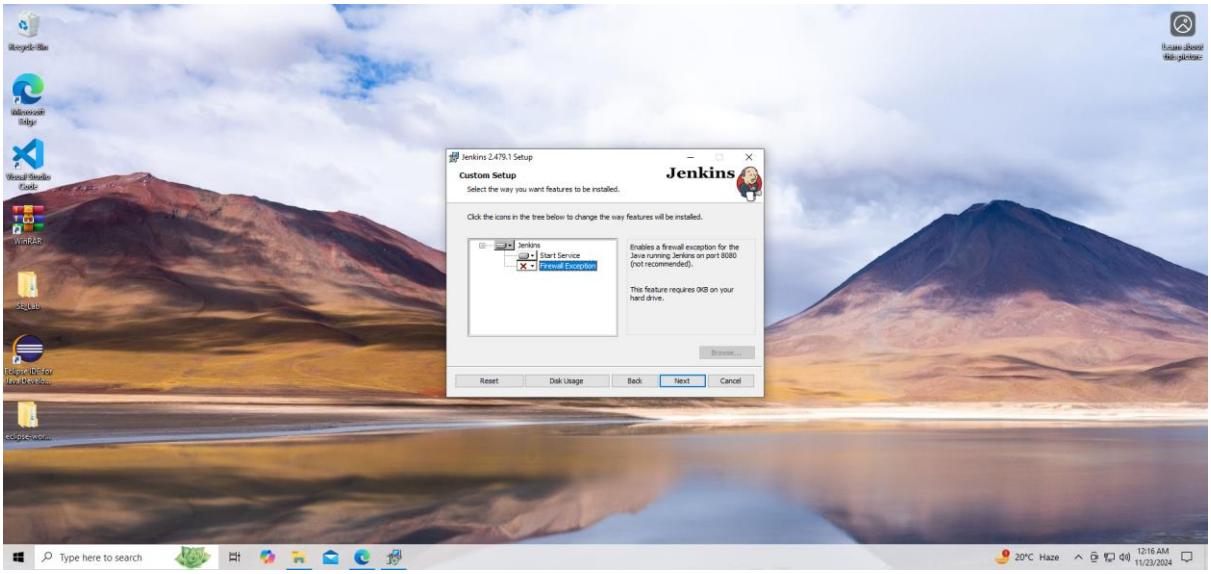

## 5. Follow on screen instructions

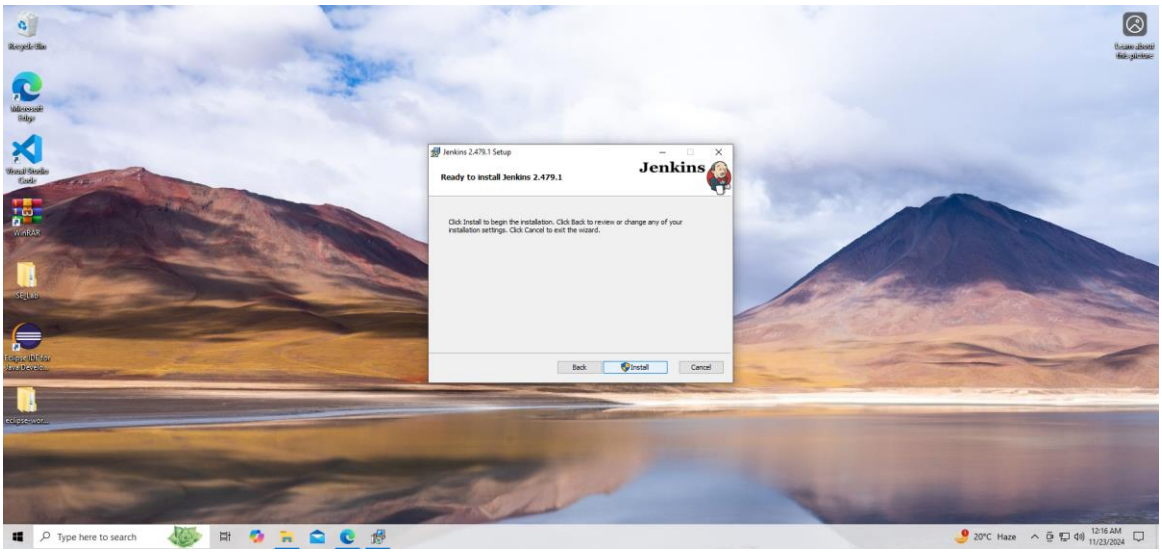

Jenkins 2.479.1 Setup
Destination Folder
Click Next to install to the default folder or click Change to choose another.
Jenkins
Install Jenkins 2.479.1 to:
C:\Program Files\Jenkins\
Change...
Back
Next
Cancel
Type here to search
EUR/INR -0.55%
11:56 PM
11/22/2024

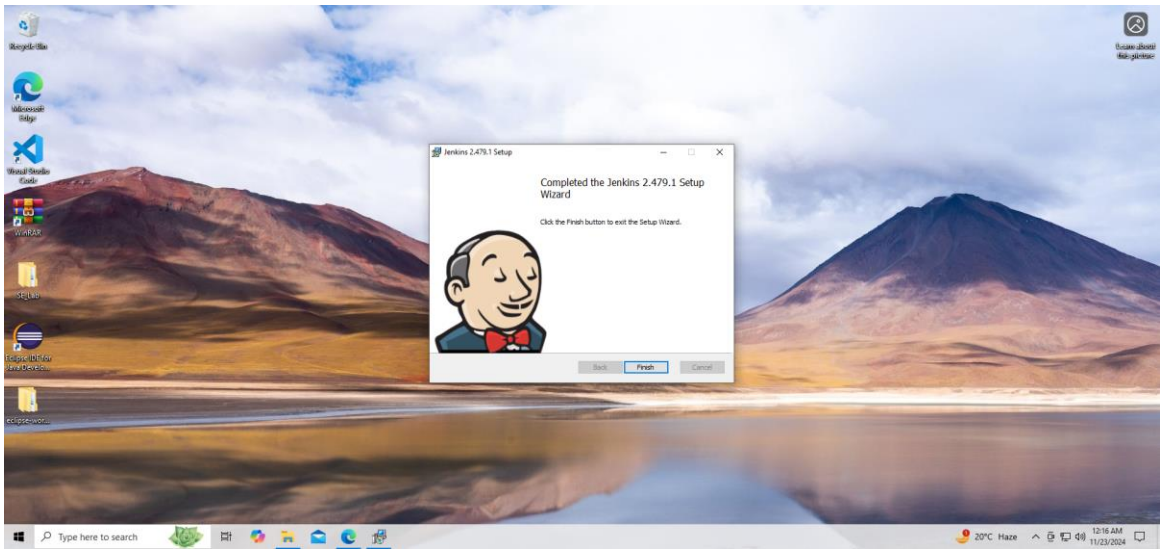

Jenkins 2.479.1 Setup
Service Logon Credentials
Enter service credentials for the service.
Jenkins
Jenkins 2.479.1 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.479.1 to run successfully.
Logon Type:
Run service as LocalSystem (not recommended)
Run service as local or domain user:
Account:
Password:
Test Credentials
Back
Next
Cancel
Type here to search
20°C Haze
12:13 AM
11/23/2024


Jenkins 2.479.1 Setup
Port Selection
Choose a port for the service.
Jenkins
Please choose a port.
Port Number (1-65535):
8080
Test Port
It is recommended that you accept the selected default port.
Back
Next
Cancel
Type here to search
20°C Haze
12:13 AM
11/23/2024

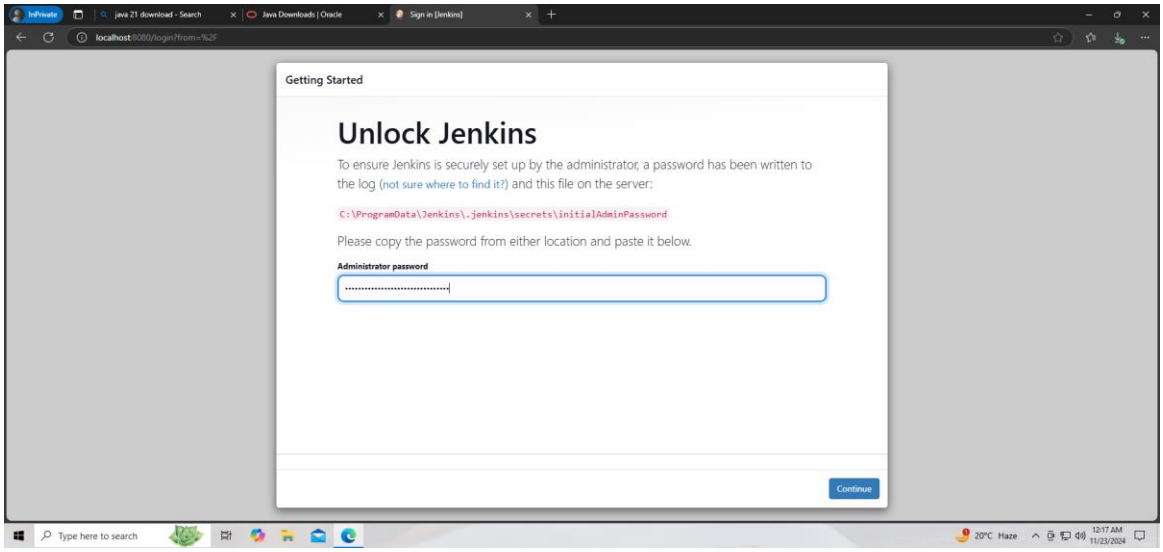

Jenkins 2.479.1 Setup
Jenkins 2.479.1 Setup
Select Java home directory (JDK or JRE)
Jenkins
Please select the path of a Java Development Kit or Java Runtime Environment.
Only Java 17 and 21 are supported by Jenkins.
C:\Program Files\Java\jdk-21\
Change...
Back
Next
Cancel
Type here to search
20°C Haze
12:16 AM
11/23/2024

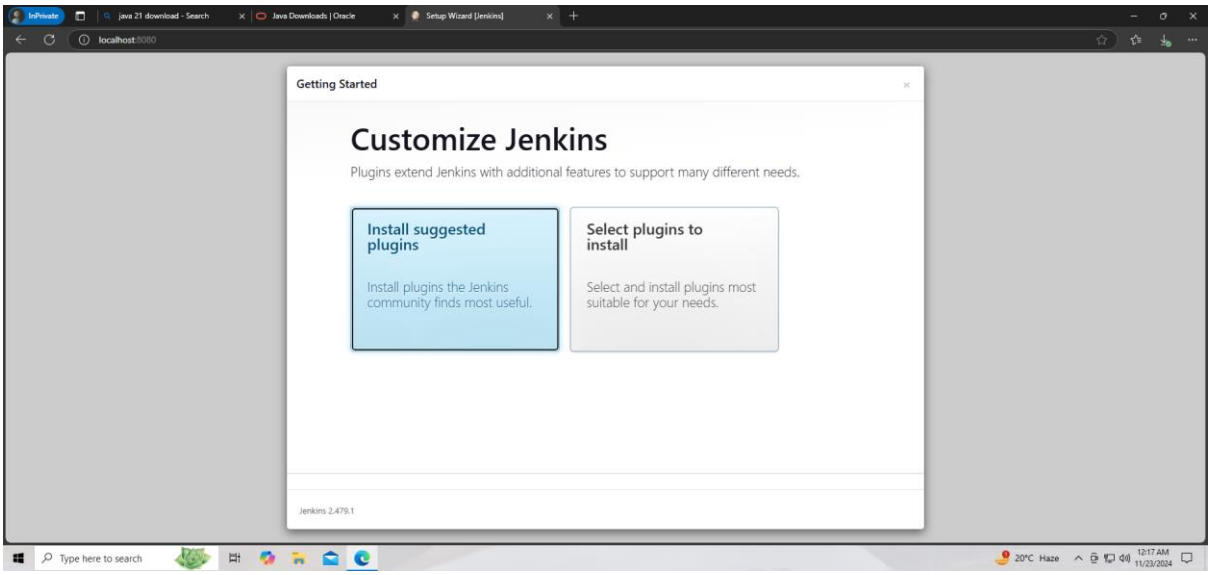

Jenkins 2.479.1 Setup
Custom Setup
Select the way you want features to be installed.
Jenkins
Click the icons in the tree below to change the way features will be installed.
Jenkins
Start Service
Firewall Exception
Enables a firewall exception for the Java running Jenkins on port 8080 (not recommended).
This feature requires 0KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel
Type here to search
20°C Haze
12:16 AM
11/23/2024

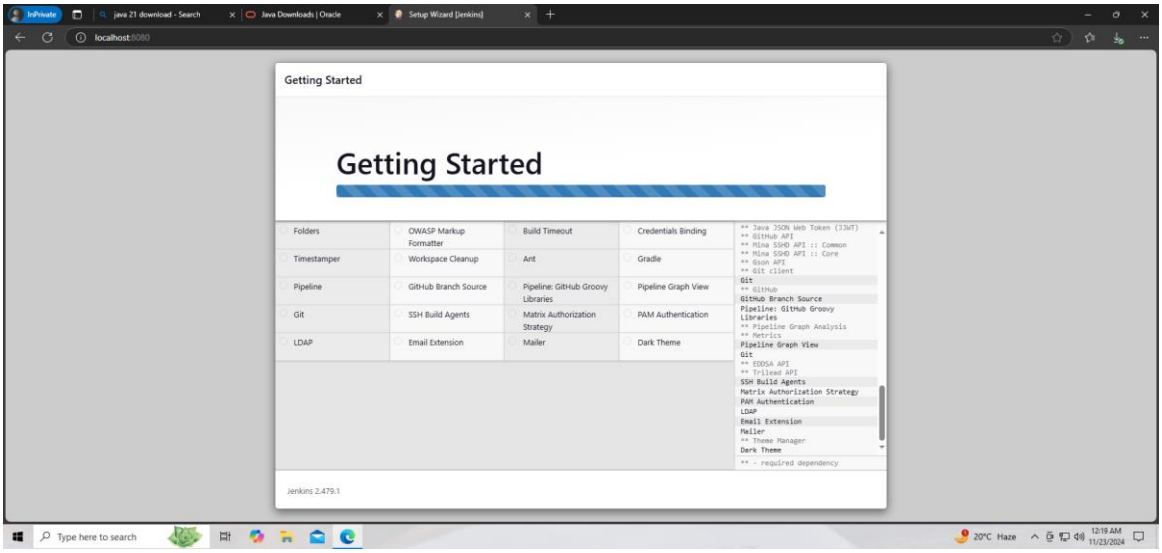

Jenkins 2.479.1 Setup
Ready to install Jenkins 2.479.1
Jenkins
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel
Type here to search
20°C Haze
12:16 AM
11/23/2024

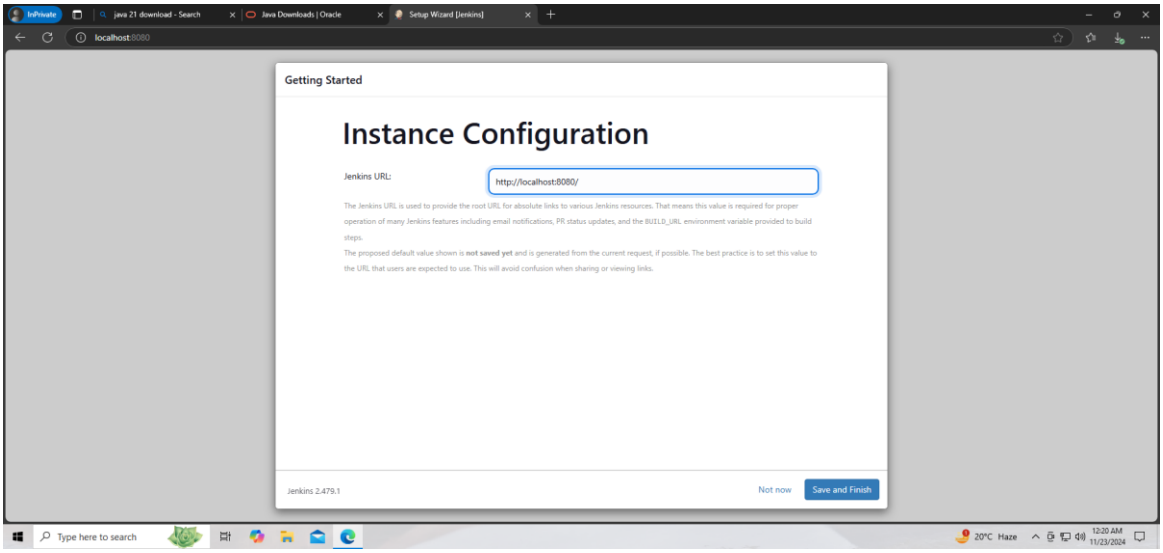


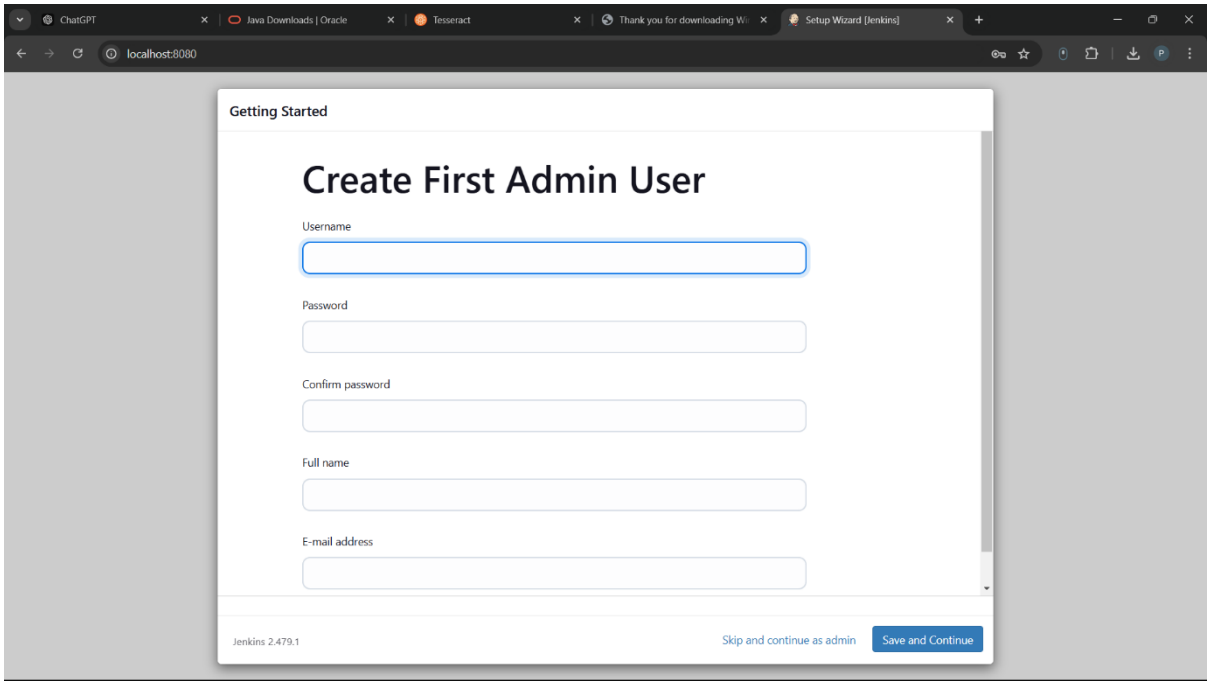

6. After installation, open localhost:8080 in browser

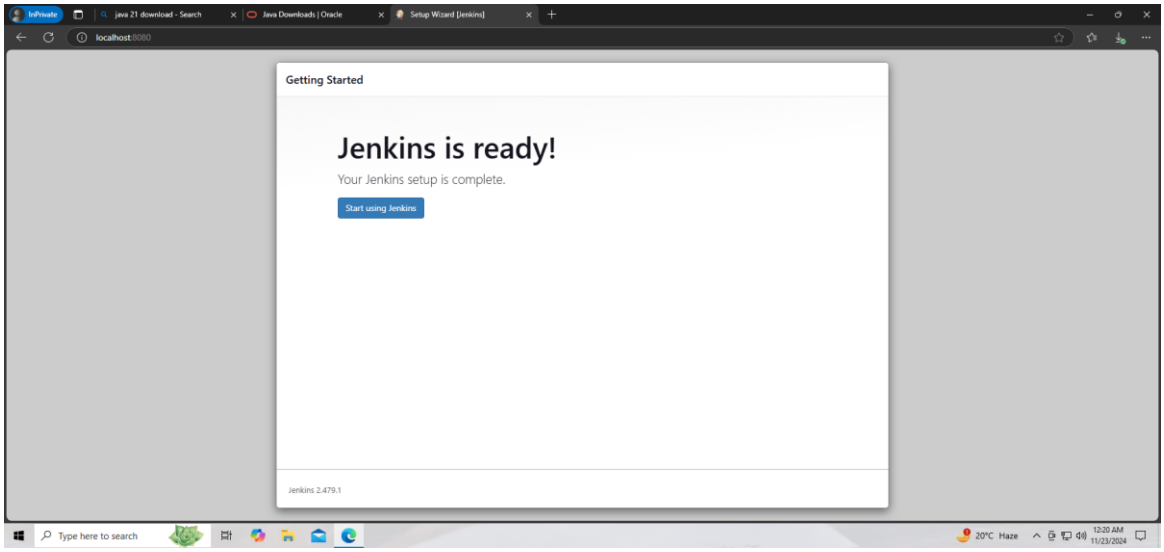


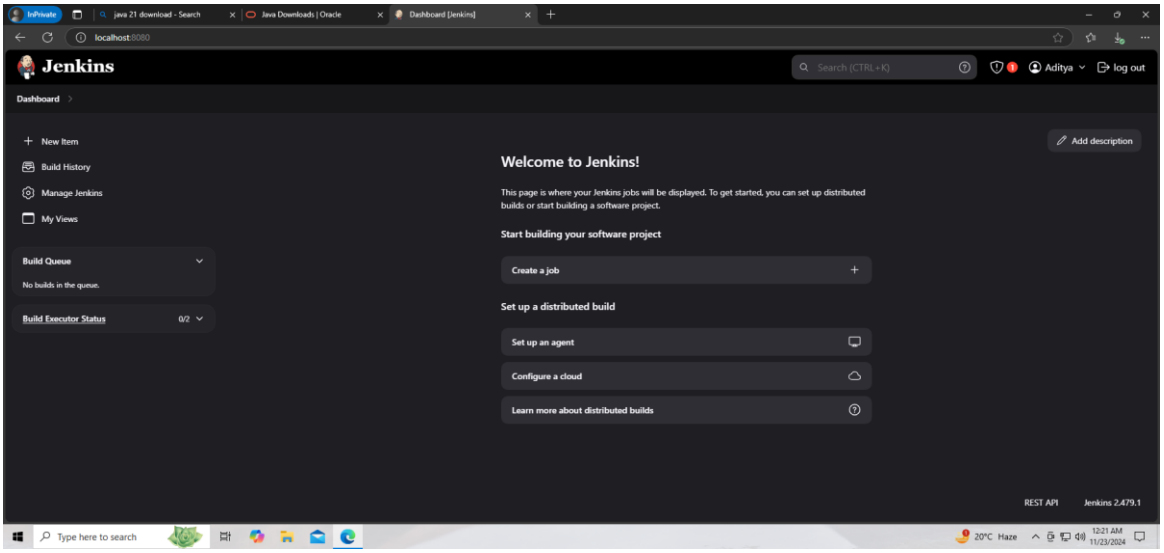

Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.479.1


Getting Started
Getting Started
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
Jenkins 2.479.1

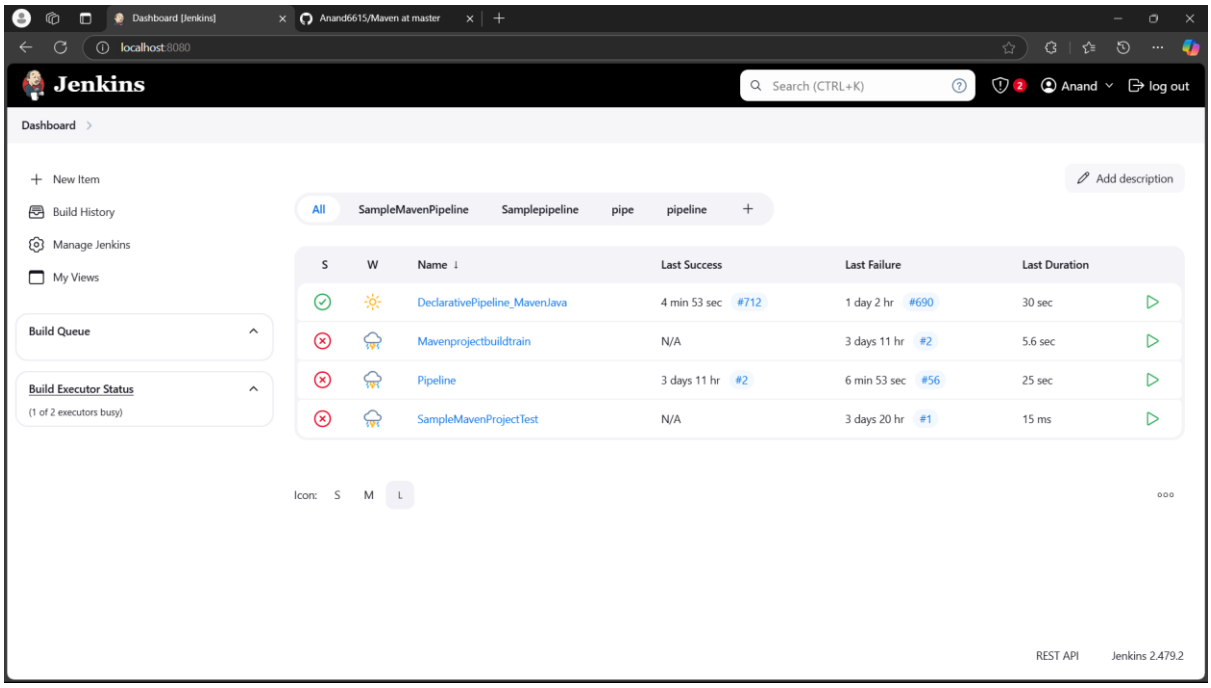

Getting Started
Instance Configuration
Jenkins URL:
http://localhost:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.
The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
Jenkins 2.479.1
Not now
Save and Finish

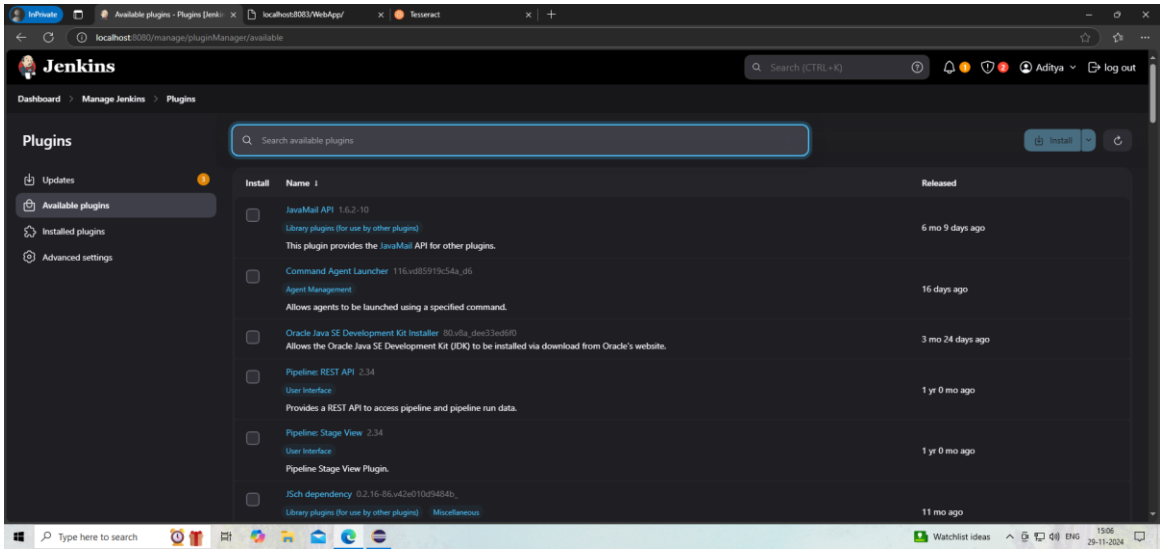

Getting Started
Create First Admin User
Username
Password
Confirm password
Full name
E-mail address
Jenkins 2.479.1
Skip and continue as admin
Save and Continue

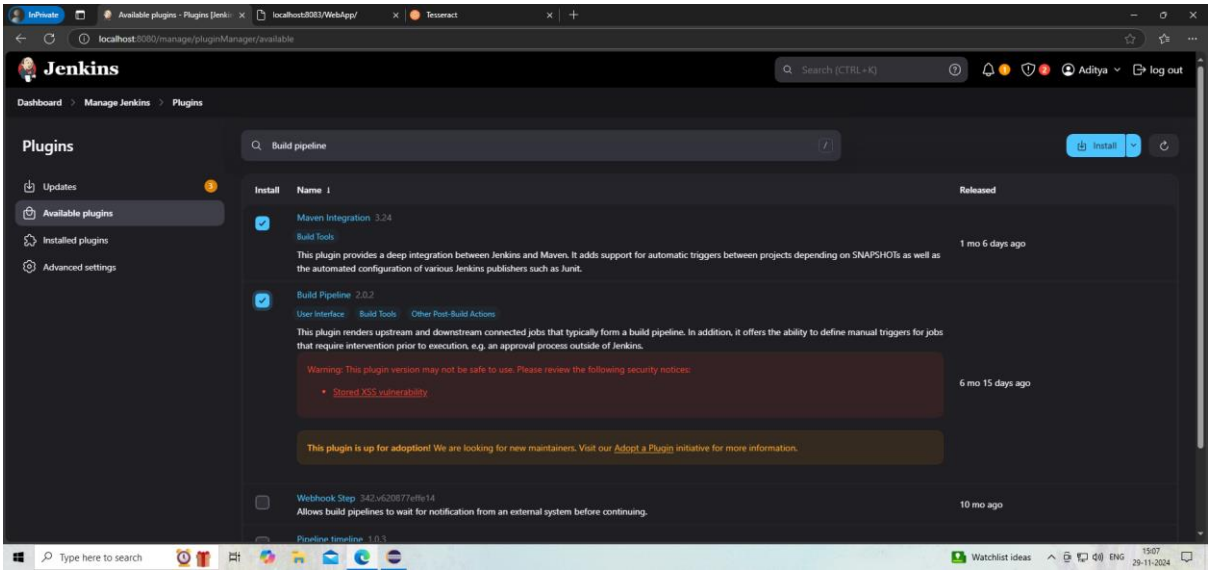

Getting Started
Jenkins is ready!
Your Jenkins setup is complete.
Start using Jenkins
Jenkins 2.479.1

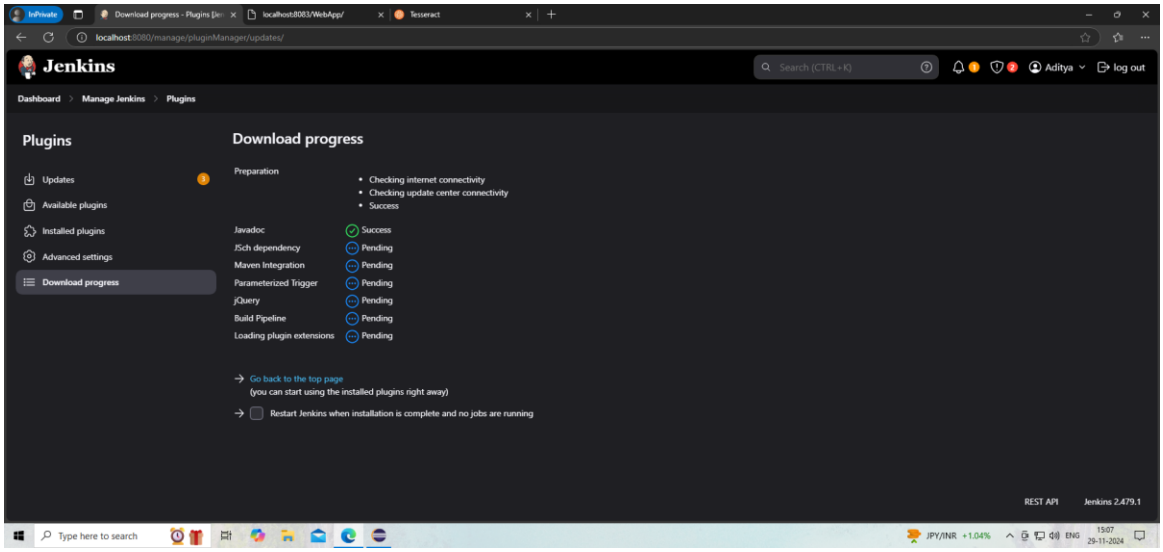

Jenkins
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building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

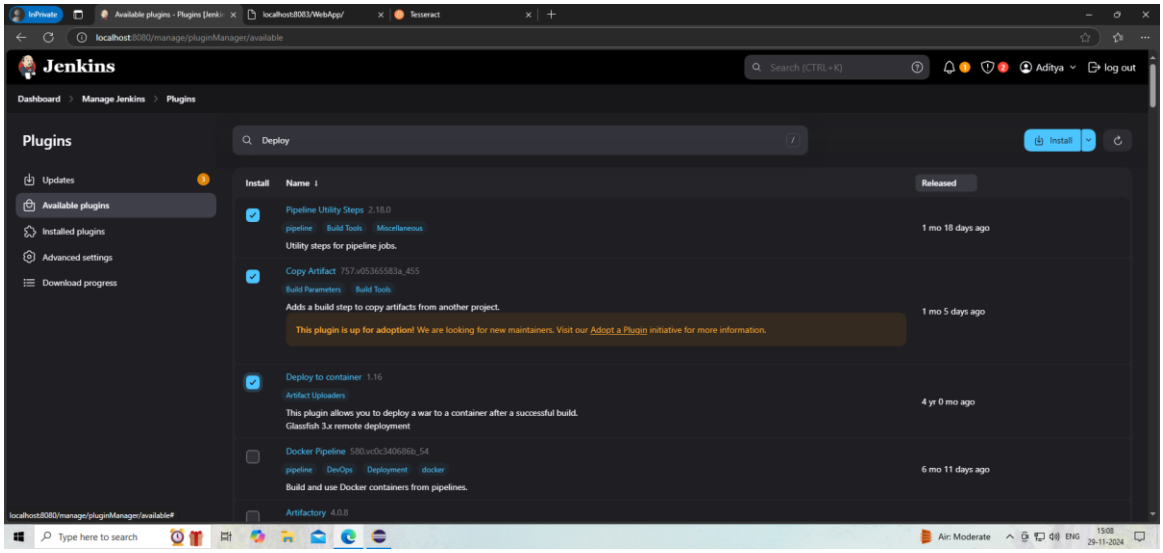
# Creating pipeline for Jenkins Web app

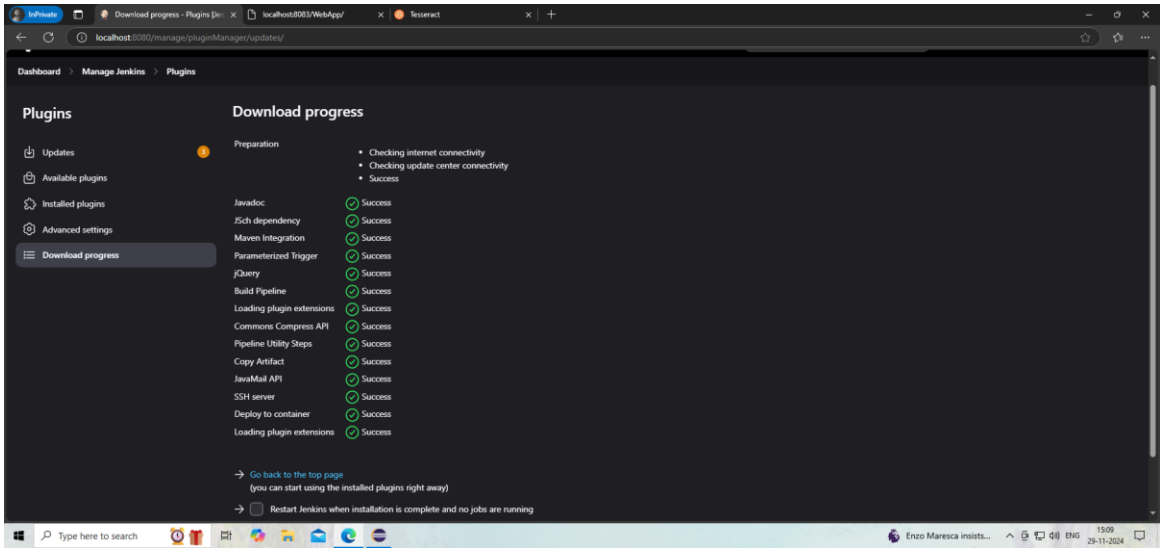


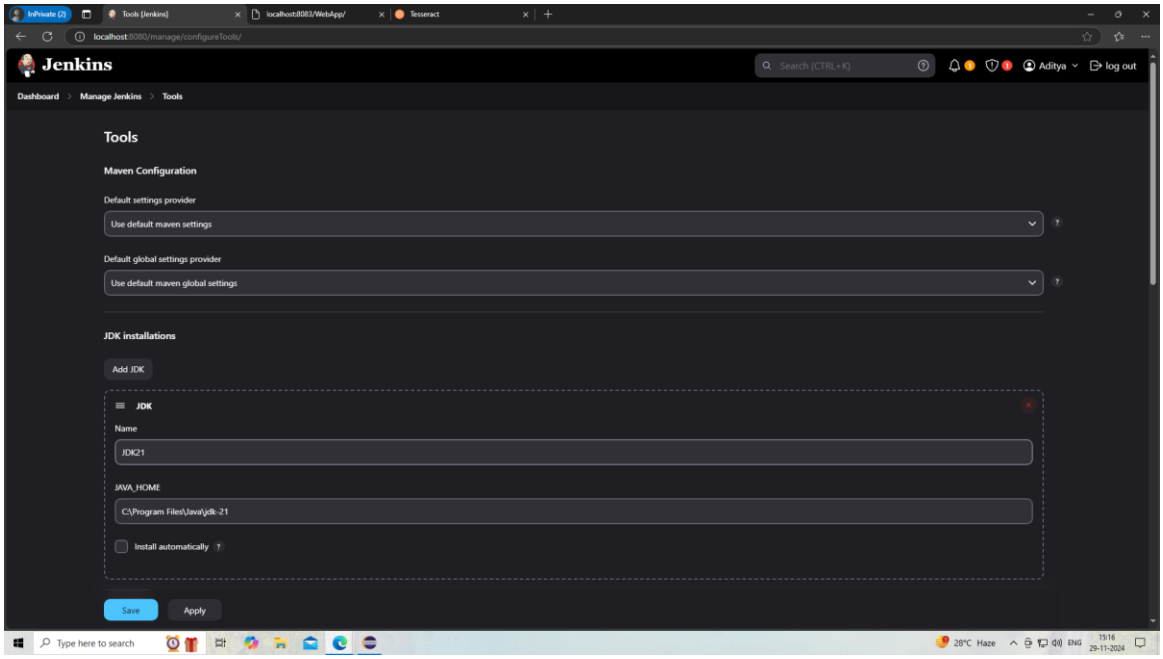



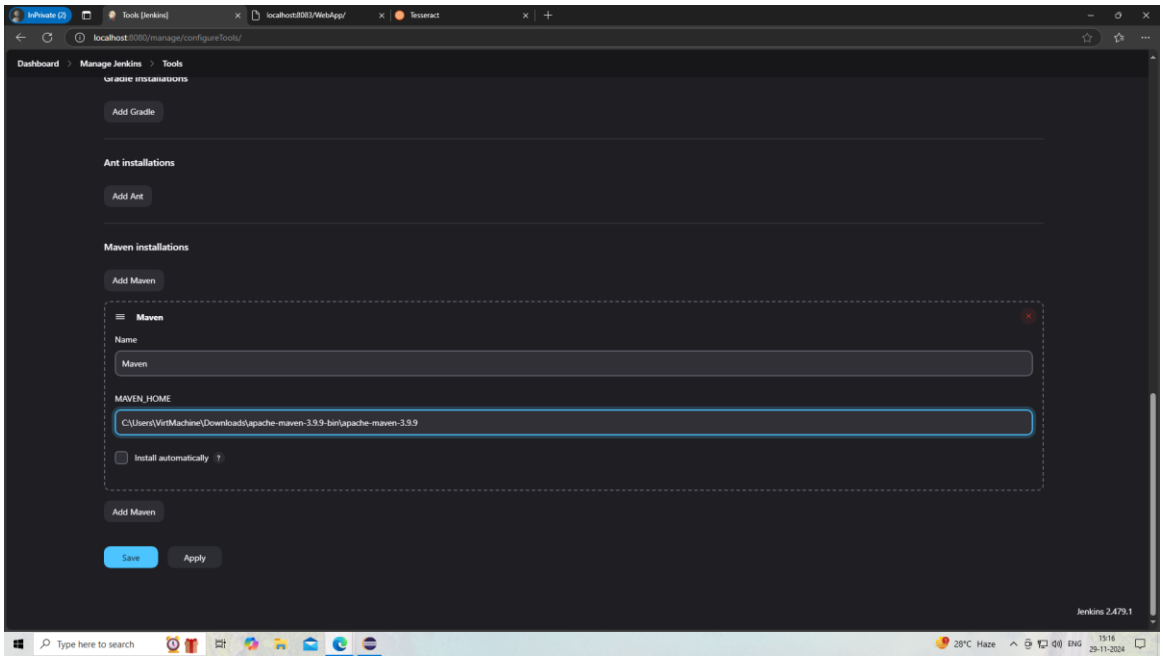

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
Manage Jenkins
Plugins
Plugins
Build pipeline
Install
Updates
Available plugins
Installed plugins
Advanced settings
Install
Name
Released
Maven Integration 3.24
Build Tools
This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.
1 mo 6 days ago
Build Pipeline 2.0.2
User Interface
Build Tools
Other Post-Build Actions
This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.
Warning: This plugin version may not be safe to use. Please review the following security notices:
Stored XSS vulnerability
6 mo 15 days ago
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
Webhook Step 342.v620877effe14
Allows build pipelines to wait for notification from an external system before continuing.
10 mo ago
Pipeline timeline 1.0.3

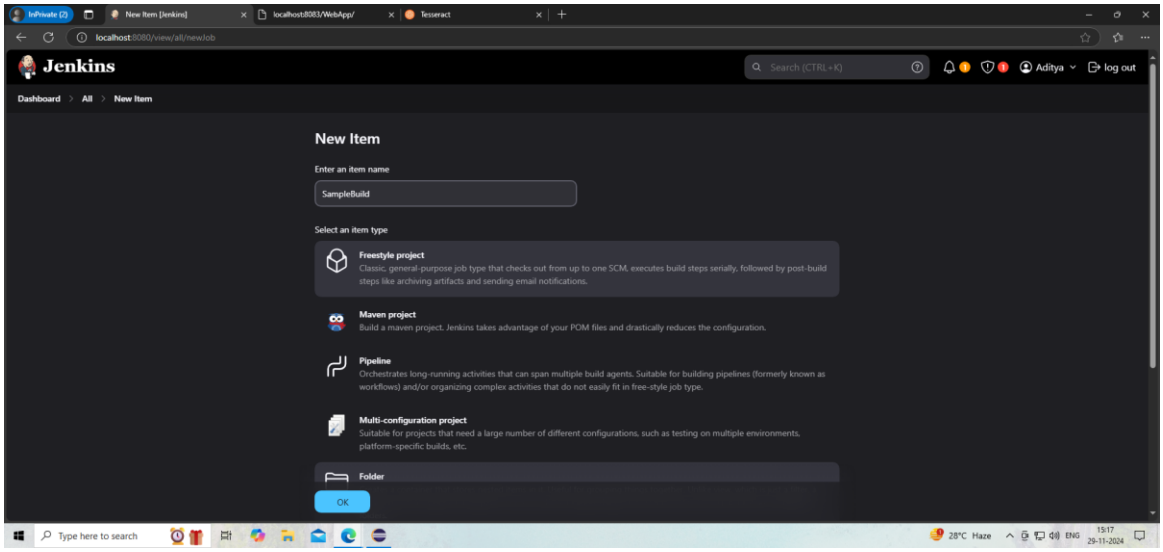

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
Manage Jenkins
Plugins
Plugins
Download progress
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
Javadoc
Success
JSch dependency
Pending
Maven Integration
Pending
Parameterized Trigger
Pending
jQuery
Pending
Build Pipeline
Pending
Loading plugin extensions
Pending
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
REST API
Jenkins 2.479.1

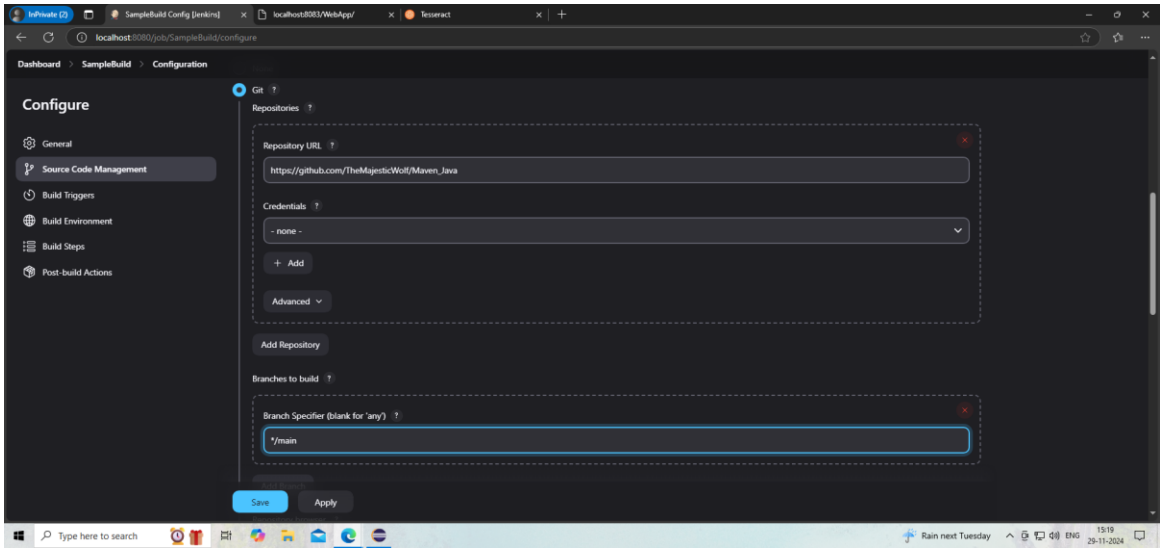

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
Manage Jenkins
Plugins
Plugins
Deploy
Install
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Install
Name
Released
Pipeline Utility Steps 2.18.0
pipeline Build Tools Miscellaneous
Utility steps for pipeline jobs.
1 mo 18 days ago
Copy Artifact 757.v05365583a_455
Build Parameters Build Tools
Adds a build step to copy artifacts from another project.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
1 mo 5 days ago
Deploy to container 1.16
Artifact Uploaders
This plugin allows you to deploy a war to a container after a successful build.
Glassfish 3.x remote deployment
4 yr 0 mo ago
Docker Pipeline 580.vc0c340686b_54
pipeline DevOps Deployment docker
Build and use Docker containers from pipelines.
6 mo 11 days ago
Artifactory 4.0.8
localhost:8080/manage/pluginManager/available#
Type here to search
Air: Moderate
ENG
15:08
29-11-2024

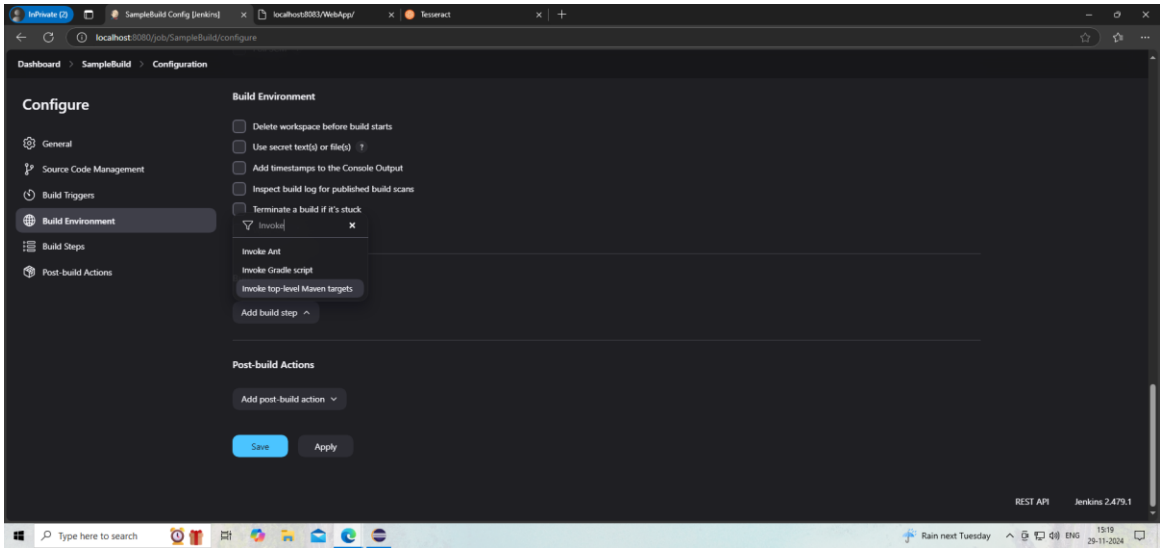

Dashboard
Manage Jenkins
Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
Javadoc Success
JSch dependency Success
Maven Integration Success
Parameterized Trigger Success
jQuery Success
Build Pipeline Success
Loading plugin extensions Success
Commons Compress API Success
Pipeline Utility Steps Success
Copy Artifact Success
JavaMail API Success
SSH server Success
Deploy to container 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
Type here to search
Enzo Maresca insists...
ENG
15:09
29-11-2024

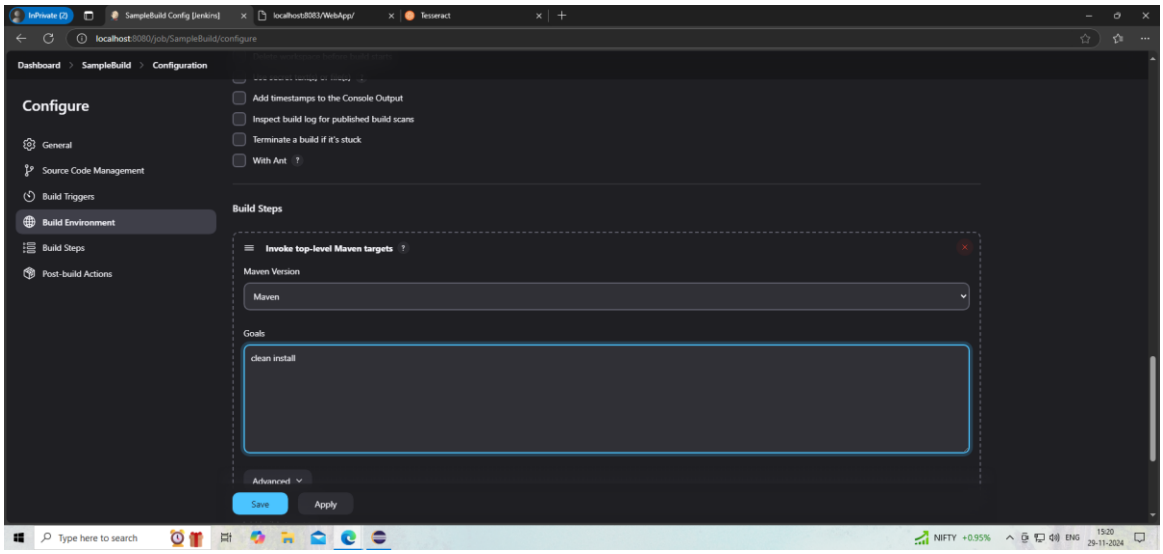

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard > Manage Jenkins > Tools
Tools
Maven Configuration
Default settings provider
Use default maven settings
Default global settings provider
Use default maven global settings
JDK installations
Add JDK
JDK
Name
JDK21
JAVA_HOME
C:\Program Files\Java\jdk-21
Install automatically
Save
Apply

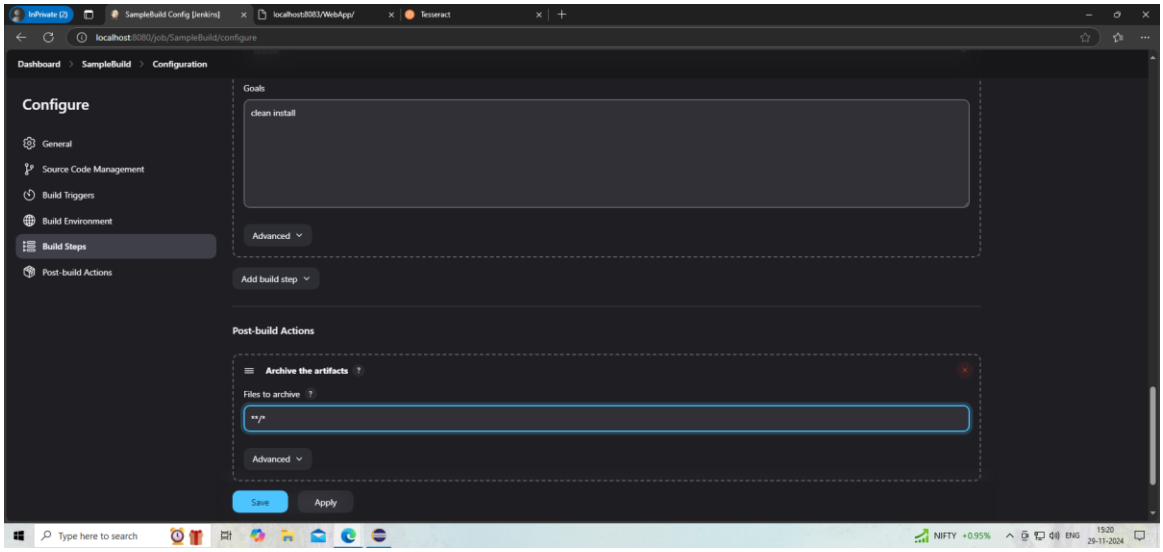

Dashboard > Manage Jenkins > Tools
Add Gradle
Ant installations
Add Ant
Maven installations
Add Maven
Maven
Name
Maven
MAVEN_HOME
C:\Users\VirtMachine\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9
Install automatically
Add Maven
Save
Apply
Jenkins 2.479.1

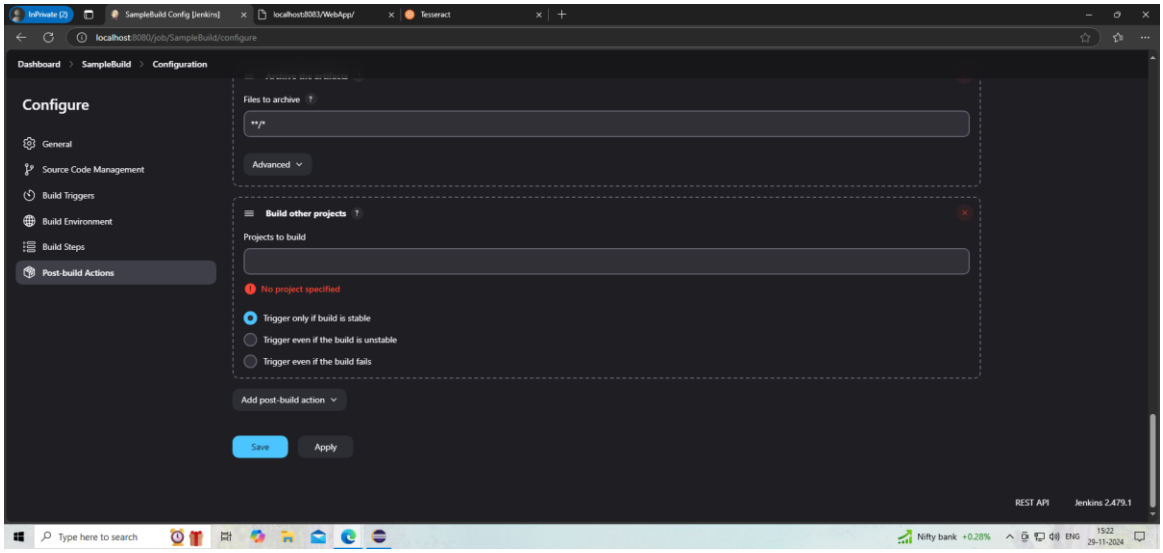

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
All
New Item
New Item
Enter an item name
SampleBuild
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK

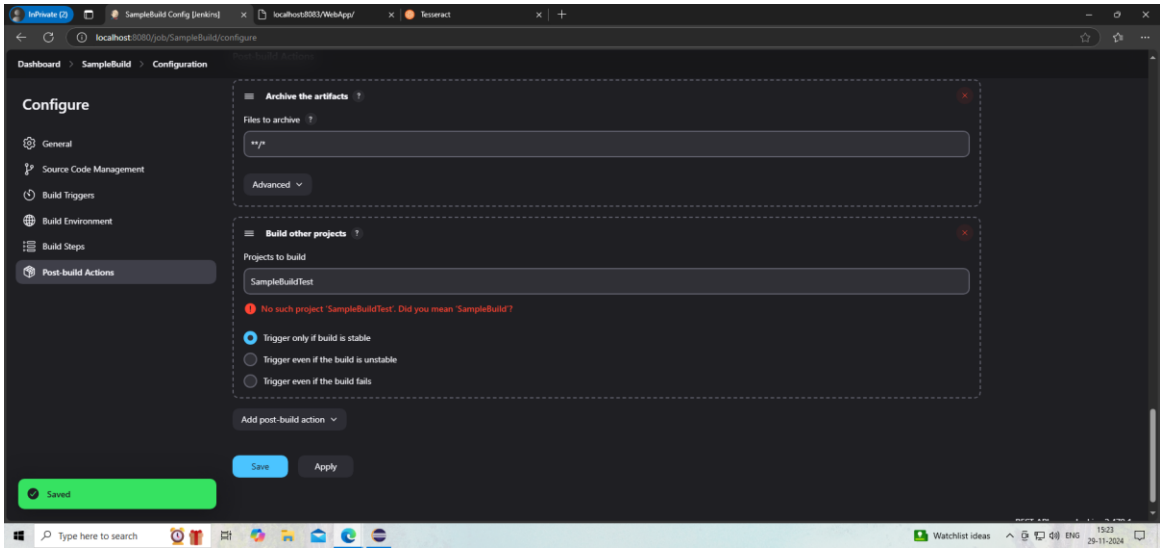

Dashboard
SampleBuild
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Git
Repositories
Repository URL
https://github.com/TheMajesticWolf/Maven_Java
Credentials
- none -
Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Save
Apply

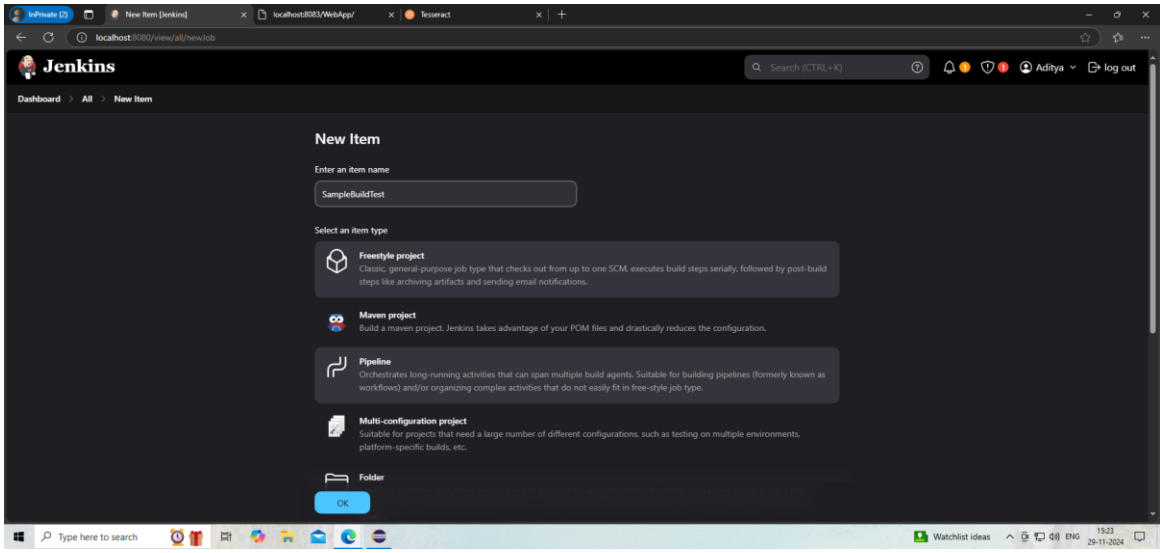

Dashboard > SampleBuild > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Environment
Delete workspace before build starts
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
Invoke
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Add build step
Post-build Actions
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

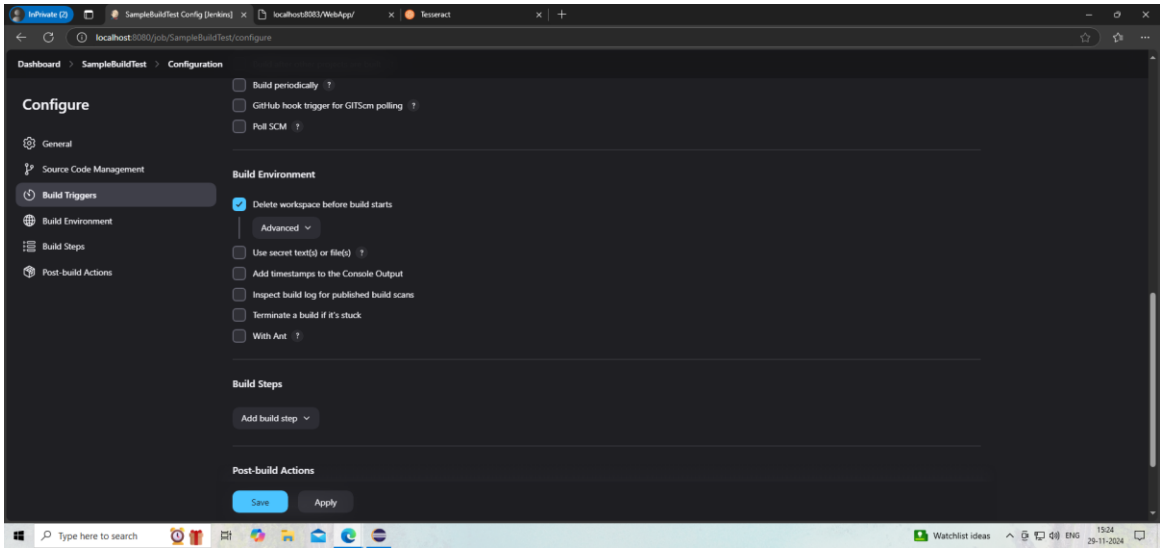

Dashboard > SampleBuild > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Invoke top-level Maven targets
Maven Version
Maven
Goals
clean install
Advanced
Save
Apply

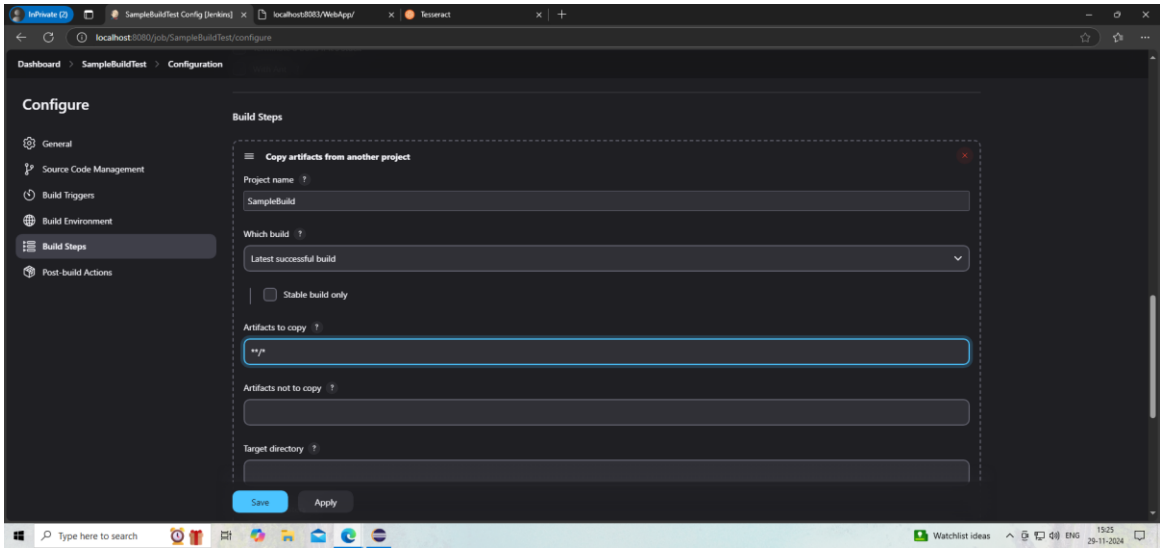

Dashboard > SampleBuild > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Goals
clean install
Advanced
Add build step
Post-build Actions
Archive the artifacts
Files to archive
**/*
Advanced
Save
Apply

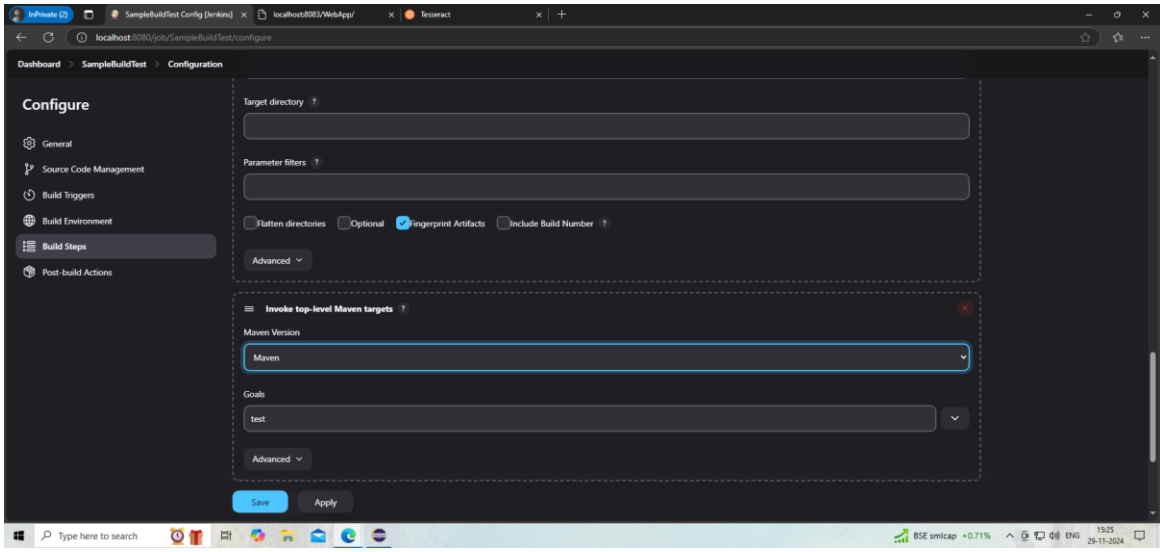

Dashboard > SampleBuild > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Files to archive
**/*
Advanced
Build other projects
Projects to build
No project specified
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

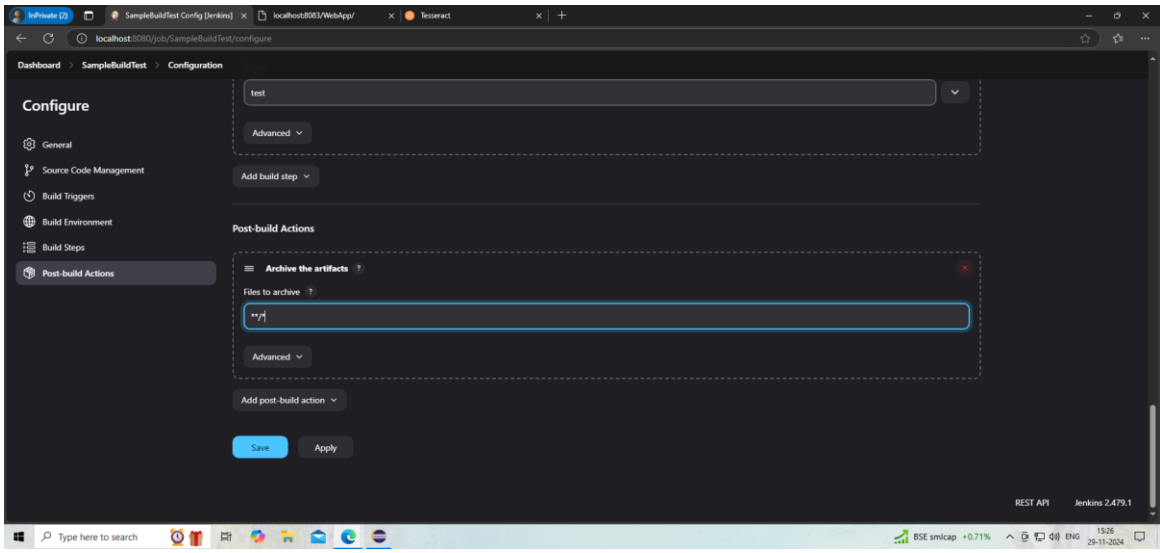

Dashboard > SampleBuild > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Archive the artifacts
Files to archive
**/*
Advanced
Build other projects
Projects to build
SampleBuildTest
No such project 'SampleBuildTest'. Did you mean 'SampleBuild'?
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply
Saved

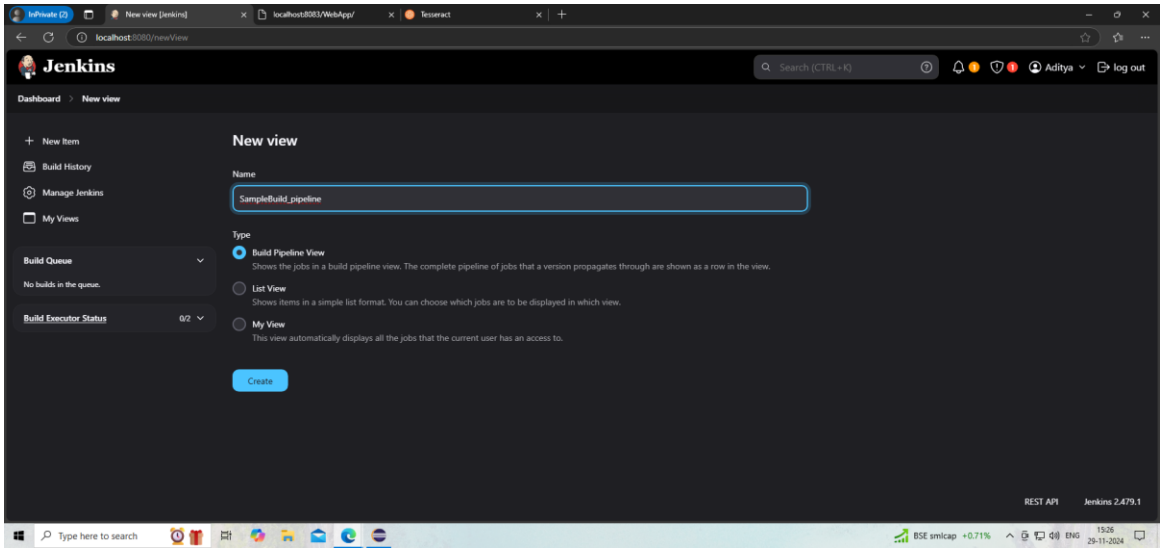

Jenkins
Dashboard > All > New Item
New Item
Enter an item name
SampleBuildTest
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK

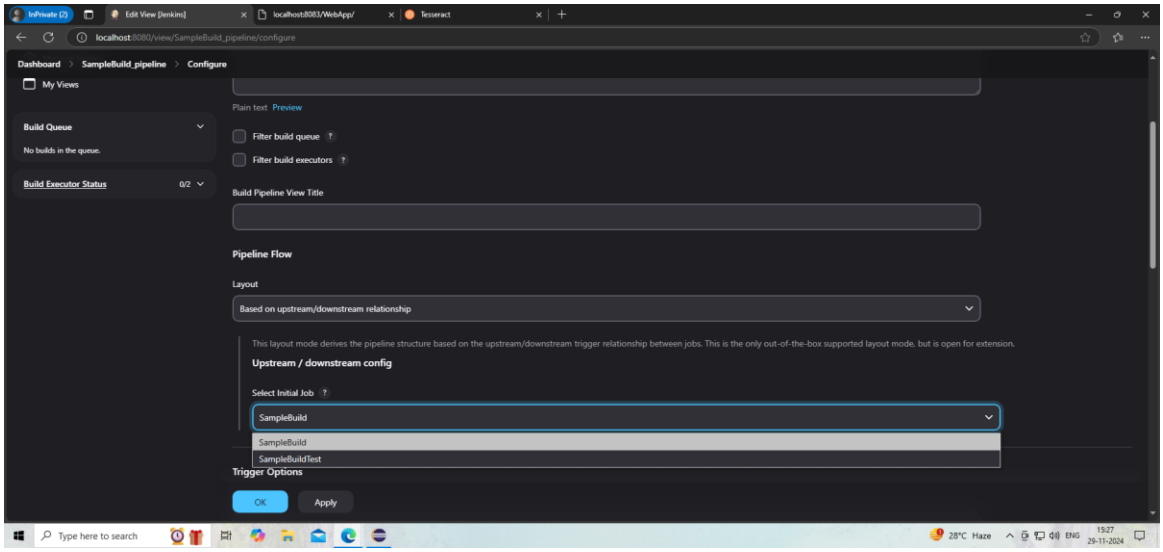

Dashboard > SampleBuildTest > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Build Environment
Delete workspace before build starts
Advanced
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Add build step
Post-build Actions
Save
Apply

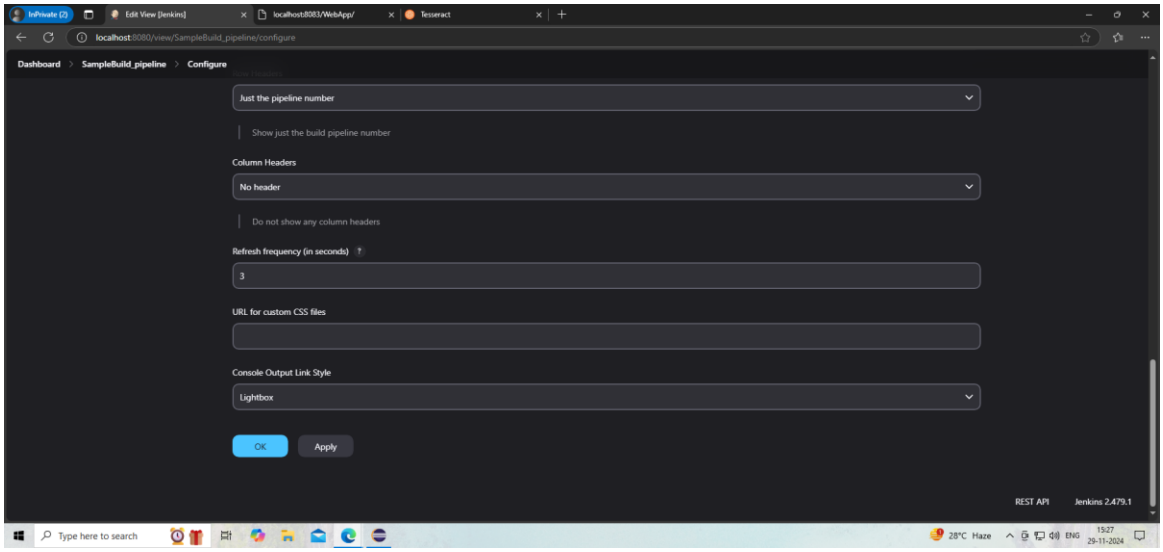

Dashboard > SampleBuildTest > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Steps
Copy artifacts from another project
Project name
SampleBuild
Which build
Latest successful build
Stable build only
Artifacts to copy
**/*
Artifacts not to copy
Target directory
Save
Apply

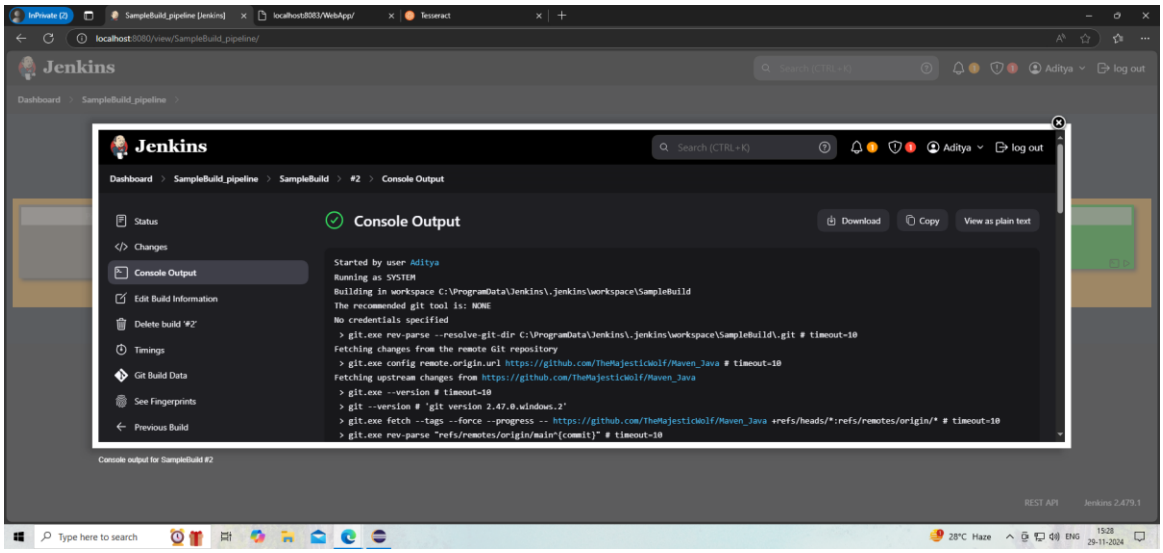

Dashboard
SampleBuildTest
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Target directory
Parameter filters
Flatten directories
Optional
Fingerprint Artifacts
Include Build Number
Advanced
Invoke top-level Maven targets
Maven Version
Maven
Goals
test
Advanced
Save
Apply

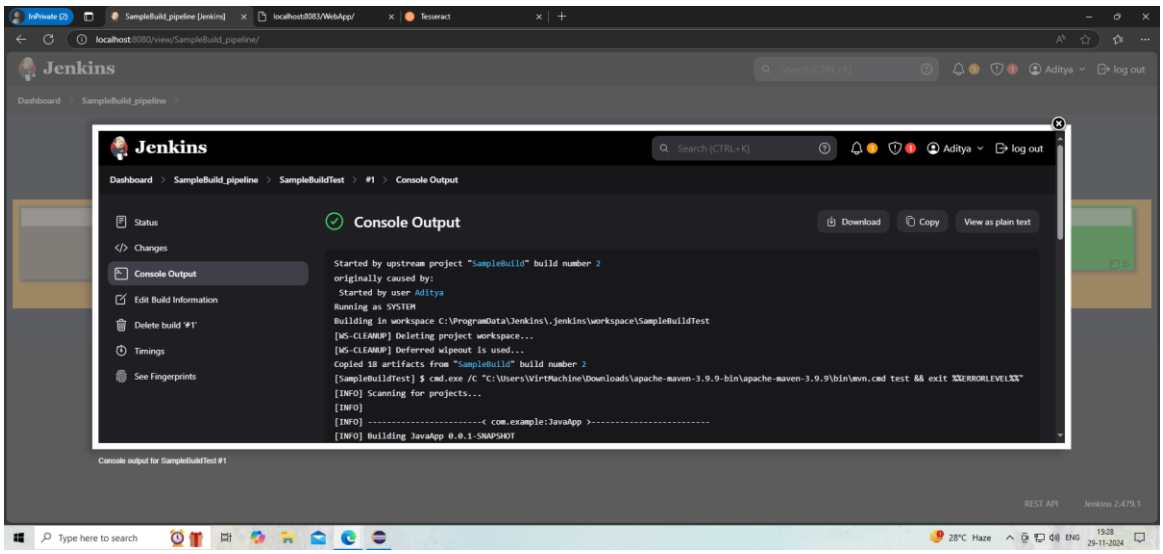

Dashboard
SampleBuildTest
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
test
Advanced
Add build step
Post-build Actions
Archive the artifacts
Files to archive
**/*
Advanced
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

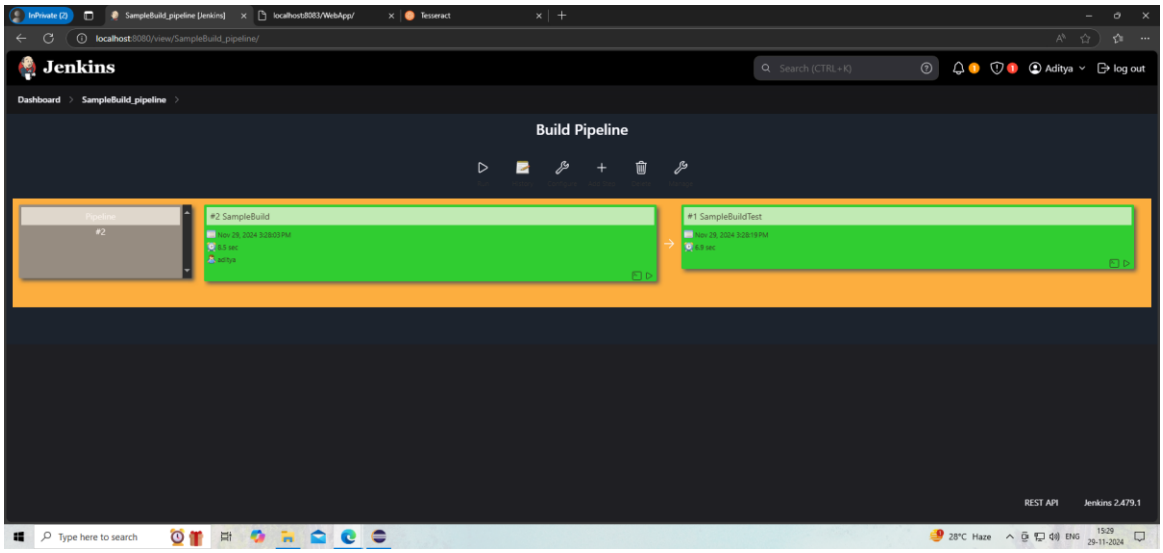

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
New view
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
New view
Name
SampleBuild_pipeline
Type
Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.
List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.
My View
This view automatically displays all the jobs that the current user has an access to.
Create
REST API
Jenkins 2.479.1
Type here to search
BSE smlcap +0.71%
ENG
15:26
29-11-2024


Dashboard
SampleBuild_pipeline
Configure
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
Plain text
Preview
Filter build queue
Filter build executors
Build Pipeline View Title
Pipeline Flow
Layout
Based on upstream/downstream relationship
This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.
Upstream / downstream config
Select Initial Job
SampleBuild
SampleBuild
SampleBuildTest
Trigger Options
OK
Apply
Type here to search
28°C Haze
ENG
15:27
29-11-2024

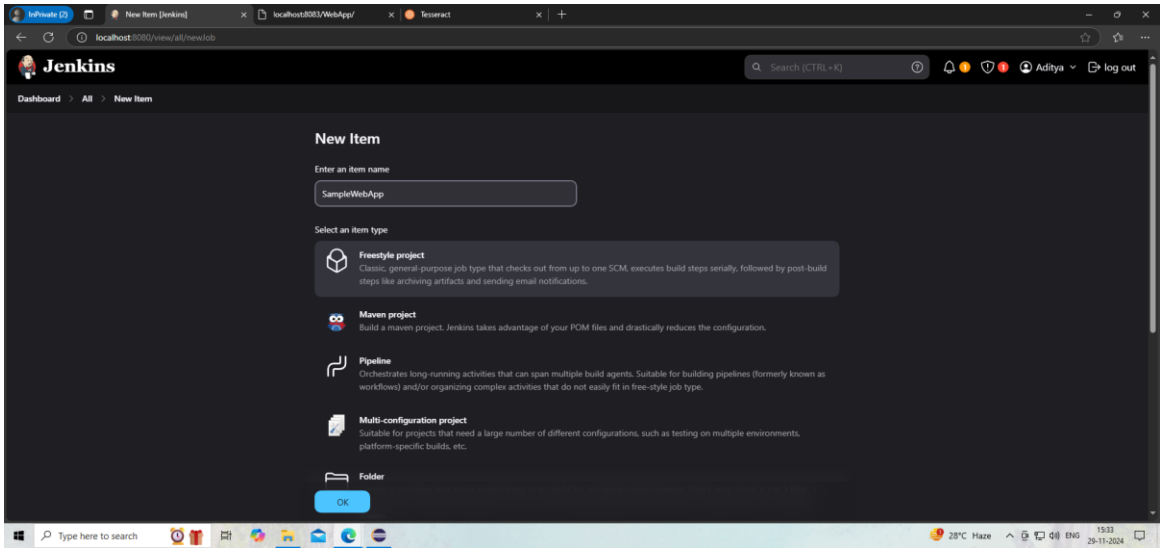

Dashboard
SampleBuild_pipeline
Configure
Just the pipeline number
Show just the build pipeline number
Column Headers
No header
Do not show any column headers
Refresh frequency (in seconds)
3
URL for custom CSS files
Console Output Link Style
Lightbox
OK
Apply
REST API
Jenkins 2.479.1

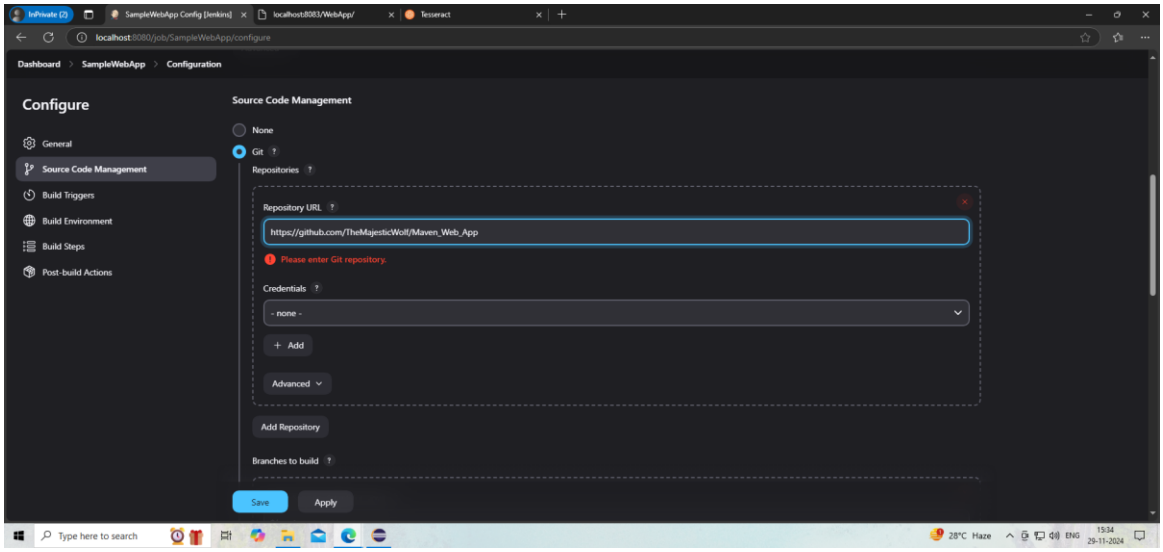

Jenkins
Dashboard
SampleBuild_pipeline
SampleBuild
#2
Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#2'
Timings
Git Build Data
See Fingerprints
Previous Build
Download
Copy
View as plain text
Started by user Aditya
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SampleBuild
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SampleBuild\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/TheMajesticWolf/Maven_Java # timeout=10
Fetching upstream changes from https://github.com/TheMajesticWolf/Maven_Java
> git.exe --version # timeout=10
> git --version # 'git version 2.47.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/TheMajesticWolf/Maven_Java +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Console output for SampleBuild #2

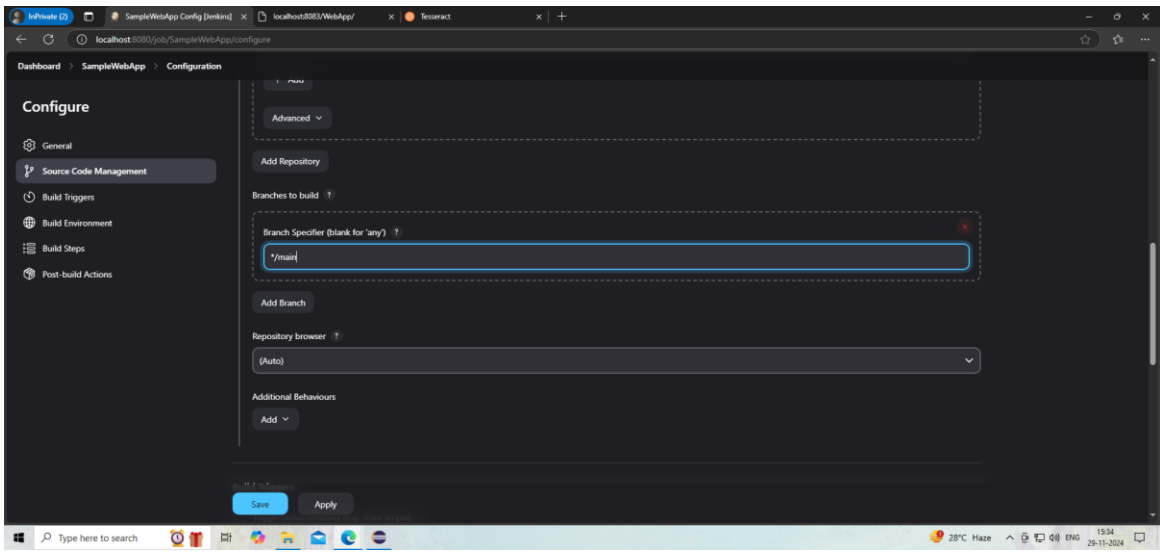

Jenkins
Dashboard
SampleBuild_pipeline
SampleBuildTest
#1
Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
See Fingerprints
Download
Copy
View as plain text
Started by upstream project "SampleBuild" build number 2
originally caused by:
Started by user Aditya
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SampleBuildTest
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
Copied 18 artifacts from "SampleBuild" build number 2
[SampleBuildTest] $ cmd.exe /C "C:\Users\VirtMachine\Downloads\apache-maven-3.9.9-bin\apache-maven-3.9.9\bin\mvn.cmd test && exit %%ERRORLEVEL%%"
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.example:JavaApp >-------------------------
[INFO] Building JavaApp 0.0.1-SNAPSHOT
Console output for SampleBuildTest #1
REST API
Jenkins 2.479.1

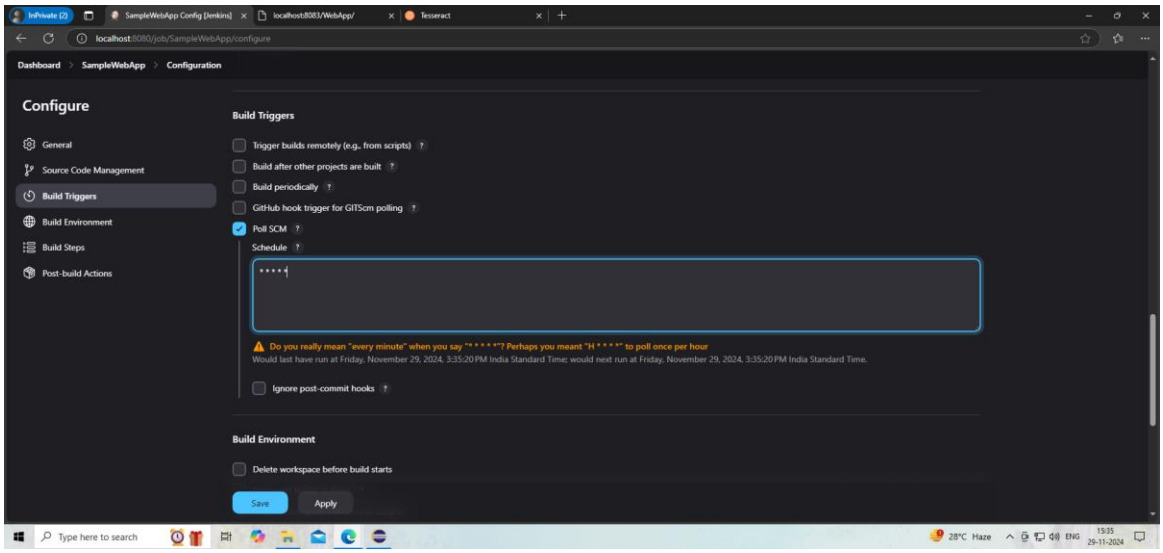

Jenkins
Dashboard
SampleBuild_pipeline
Build Pipeline
#2
#2 SampleBuild
#1 SampleBuildTest
REST API
Jenkins 2.479.1

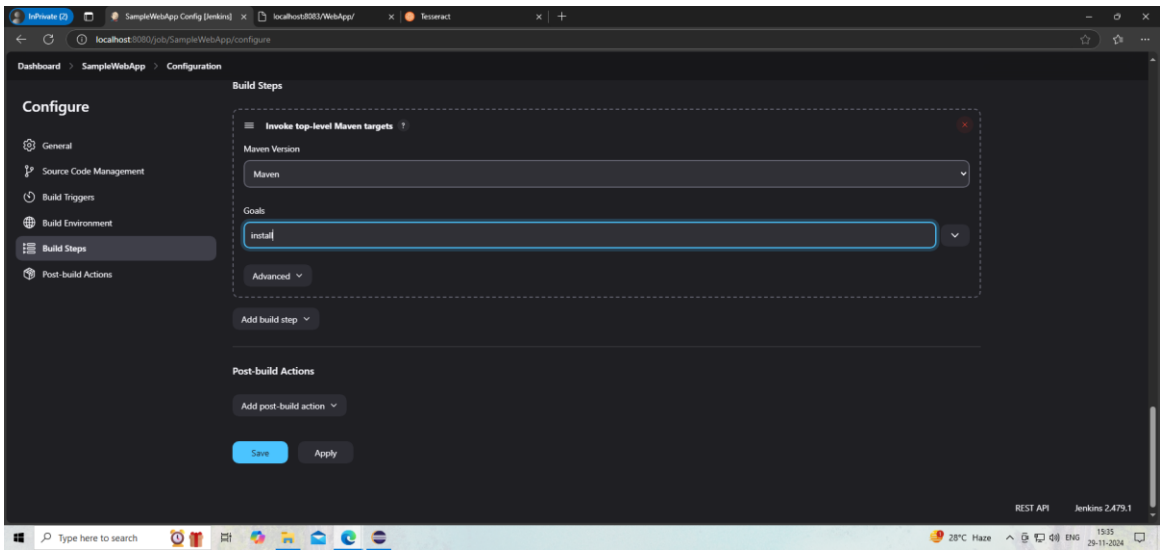
# Creating pipeline for Jenkins Web app

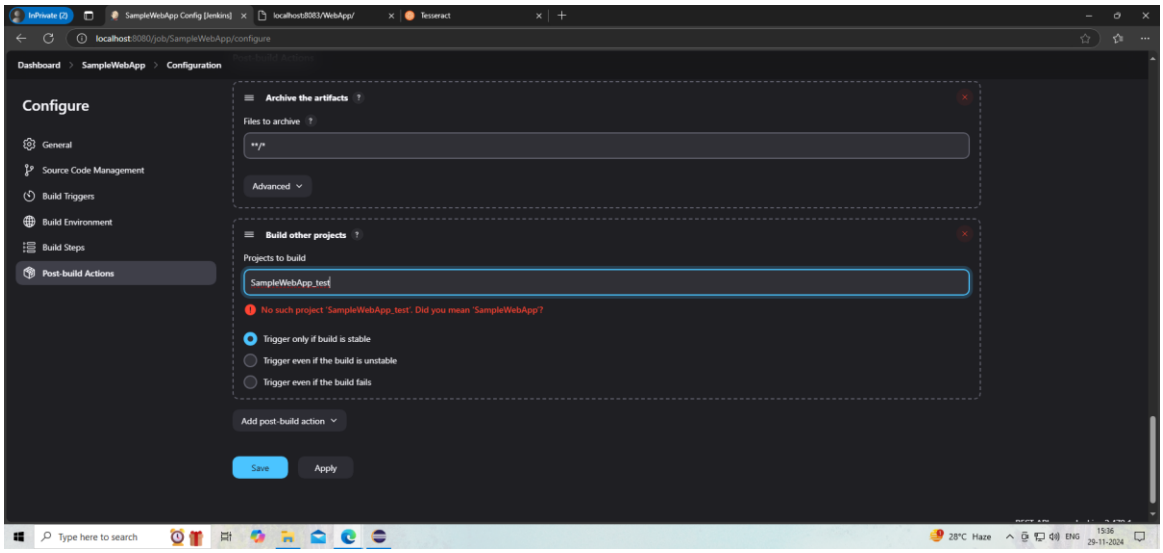


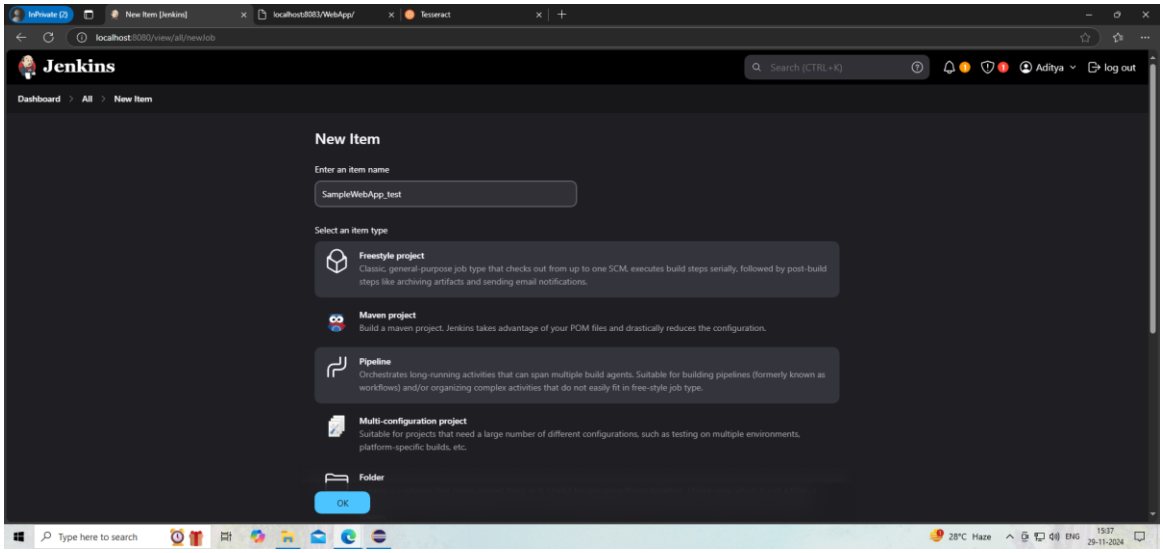



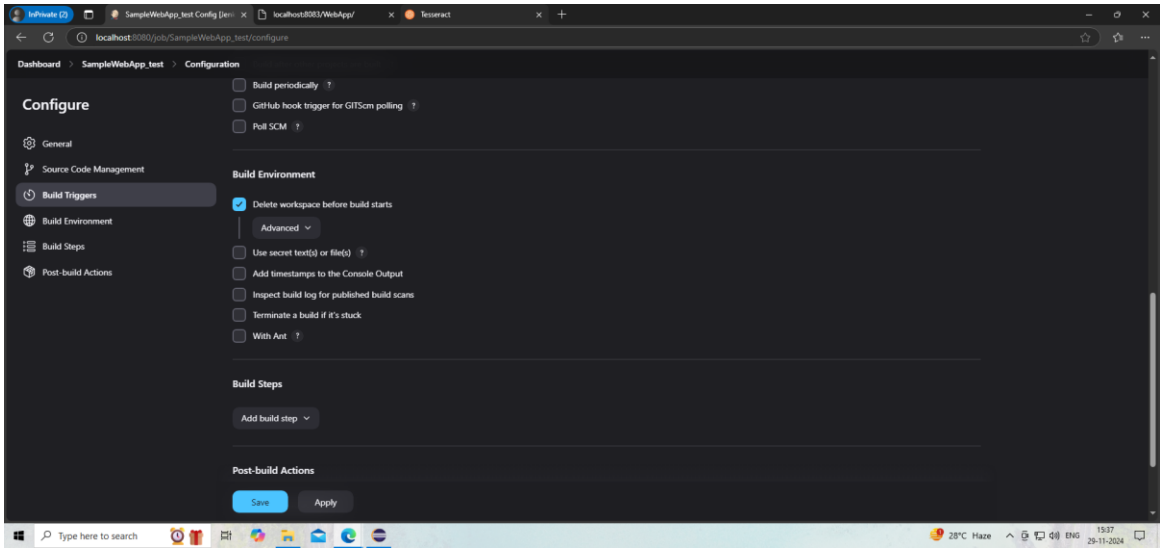

localhost:8080/job/SampleWebApp/configure
Dashboard > SampleWebApp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add
Save
Apply

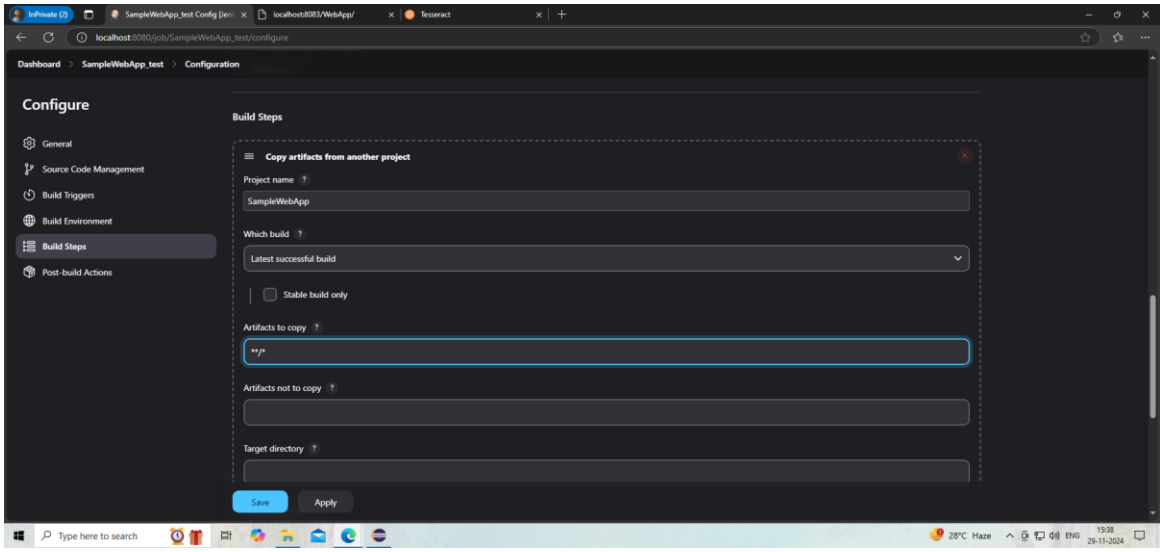

localhost:8080/job/SampleWebApp/configure
Dashboard > SampleWebApp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Schedule
* * * * *
Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Friday, November 29, 2024, 3:35:20 PM India Standard Time; would next run at Friday, November 29, 2024, 3:35:20 PM India Standard Time.
Ignore post-commit hooks
Build Environment
Delete workspace before build starts
Save
Apply

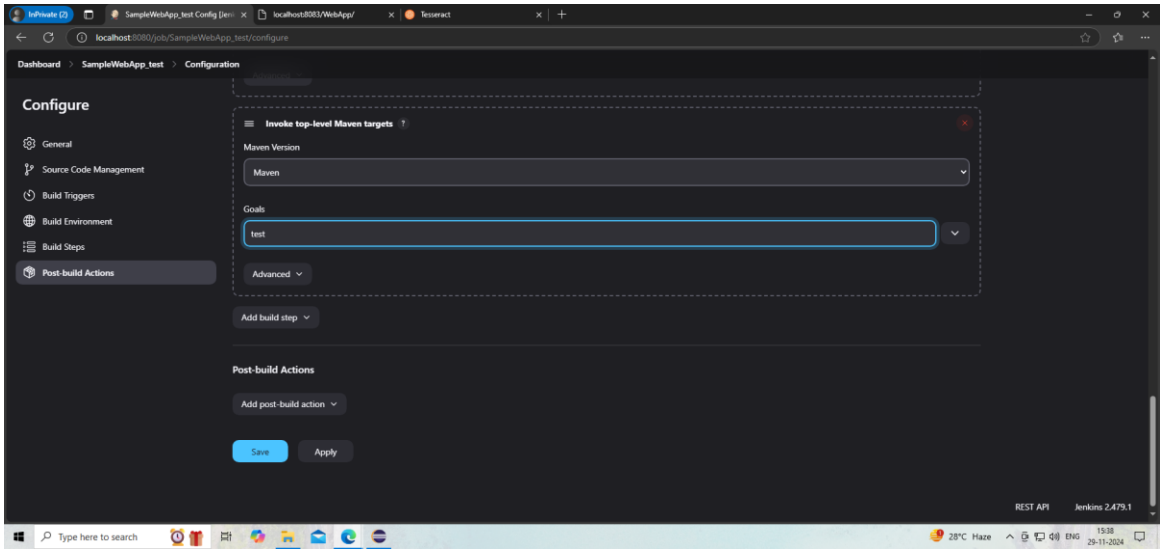

localhost:8080/job/SampleWebApp/configure
Dashboard > SampleWebApp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Steps
Invoke top-level Maven targets
Maven Version
Maven
Goals
install
Advanced
Add build step
Post-build Actions
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

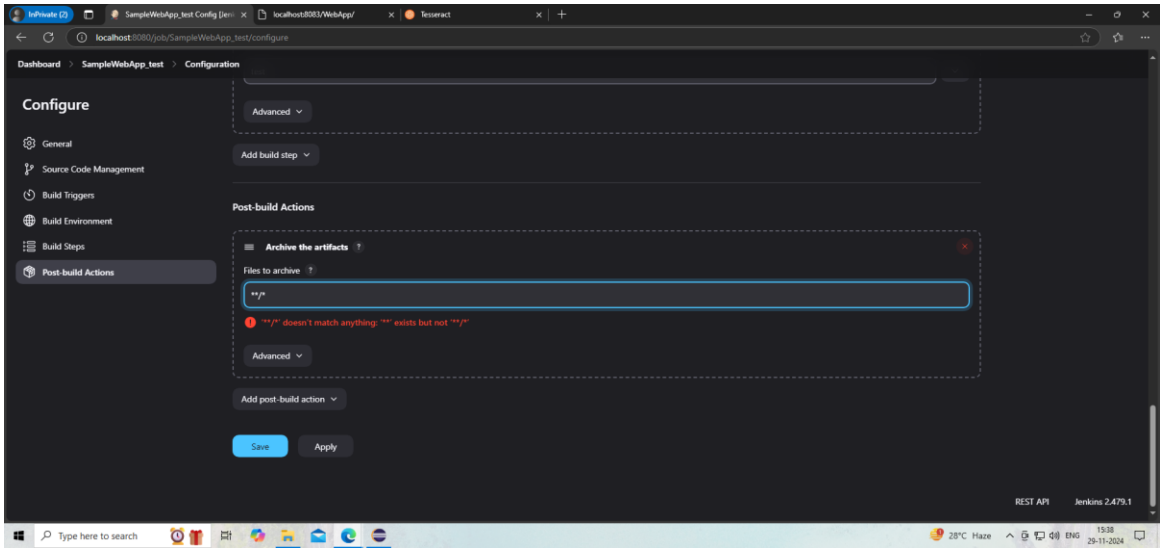

localhost:8080/job/SampleWebApp/configure
Dashboard > SampleWebApp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Archive the artifacts
Files to archive
**/*
Advanced
Build other projects
Projects to build
SampleWebApp_test
No such project 'SampleWebApp_test'. Did you mean 'SampleWebApp'?
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply

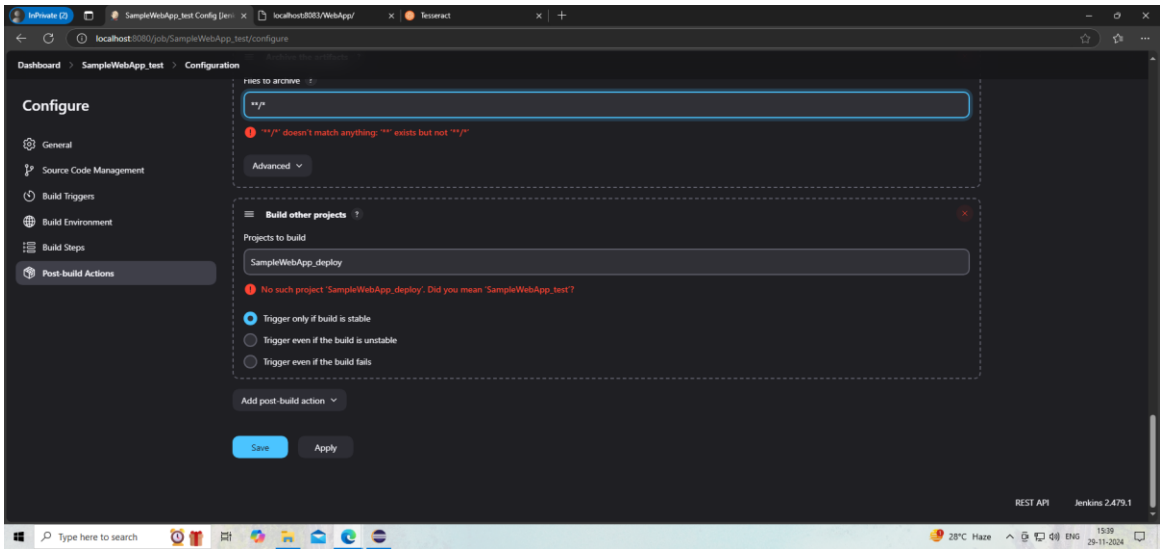

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
All
New Item
New Item
Enter an item name
SampleWebApp_test
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK

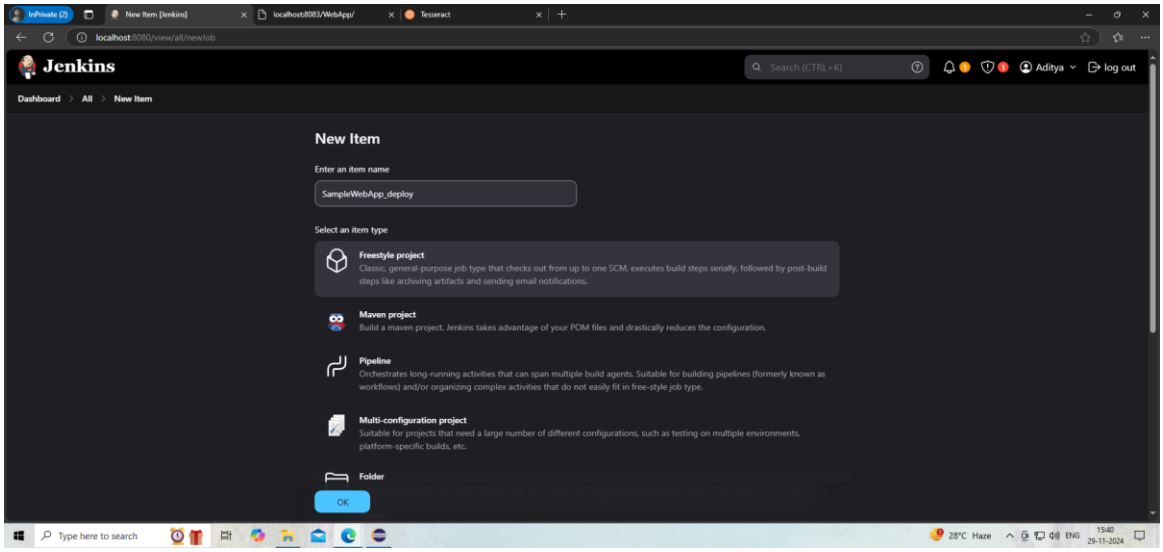

Dashboard
SampleWebApp_test
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build periodically
GitHub hook trigger for GITScm polling
Poll SCM
Build Environment
Delete workspace before build starts
Advanced
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Add build step
Post-build Actions
Save
Apply

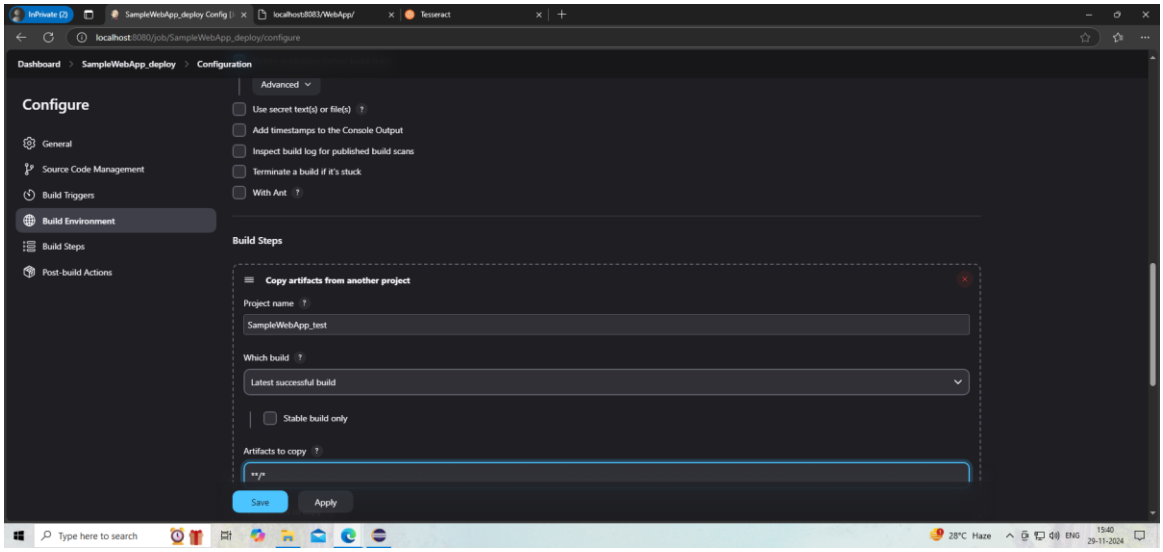

Dashboard
SampleWebApp_test
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Steps
Copy artifacts from another project
Project name
SampleWebApp
Which build
Latest successful build
Stable build only
Artifacts to copy
**/*
Artifacts not to copy
Target directory
Save
Apply

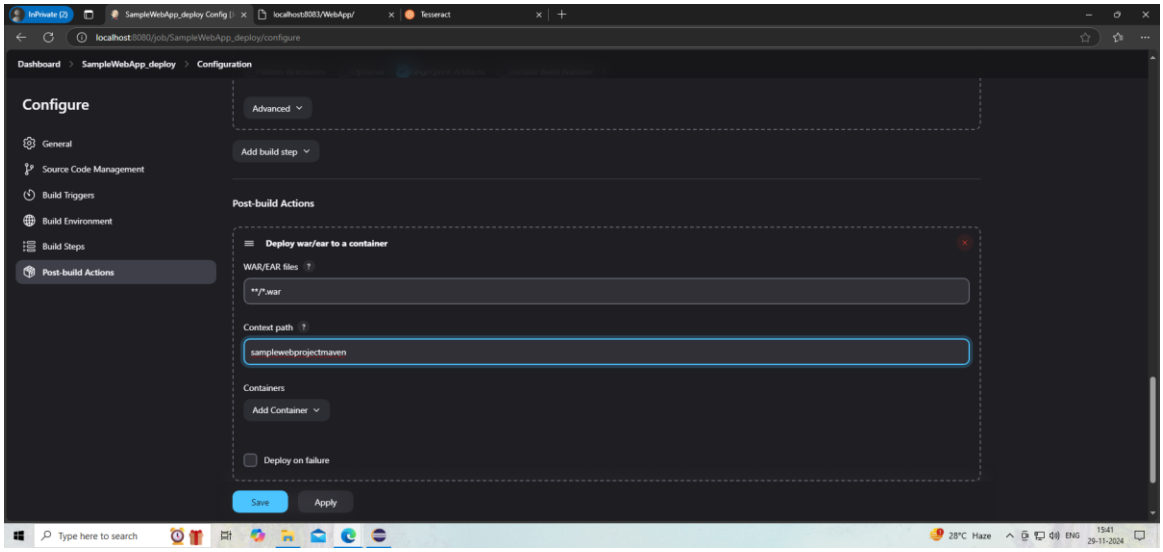

Dashboard
SampleWebApp_test
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Invoke top-level Maven targets
Maven Version
Maven
Goals
test
Advanced
Add build step
Post-build Actions
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

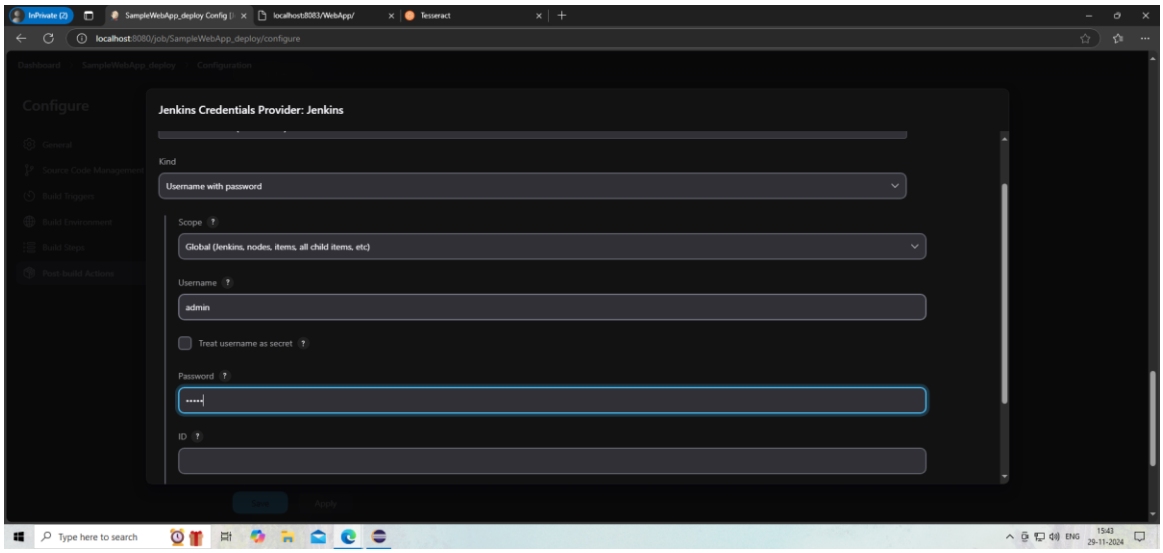

Dashboard > SampleWebApp_test > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Advanced
Add build step
Post-build Actions
Archive the artifacts
Files to archive
**/*
'**/*' doesn't match anything: '**' exists but not '**/*'
Advanced
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

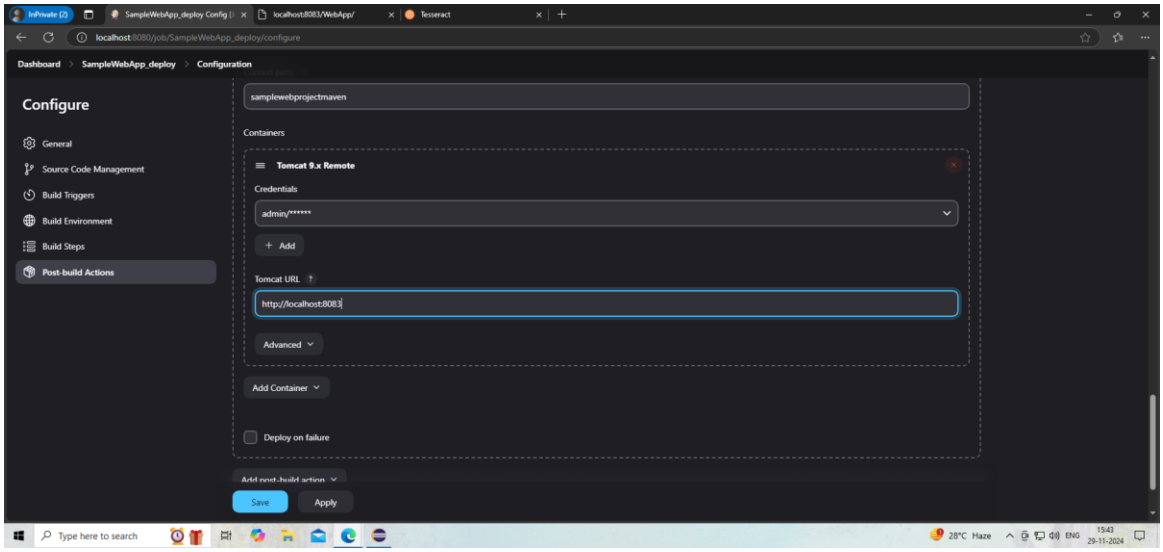

Dashboard > SampleWebApp_test > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
**/*
'**/*' doesn't match anything: '**' exists but not '**/*'
Advanced
Build other projects
Projects to build
SampleWebApp_deploy
No such project 'SampleWebApp_deploy'. Did you mean 'SampleWebApp_test'?
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Add post-build action
Save
Apply
REST API
Jenkins 2.479.1

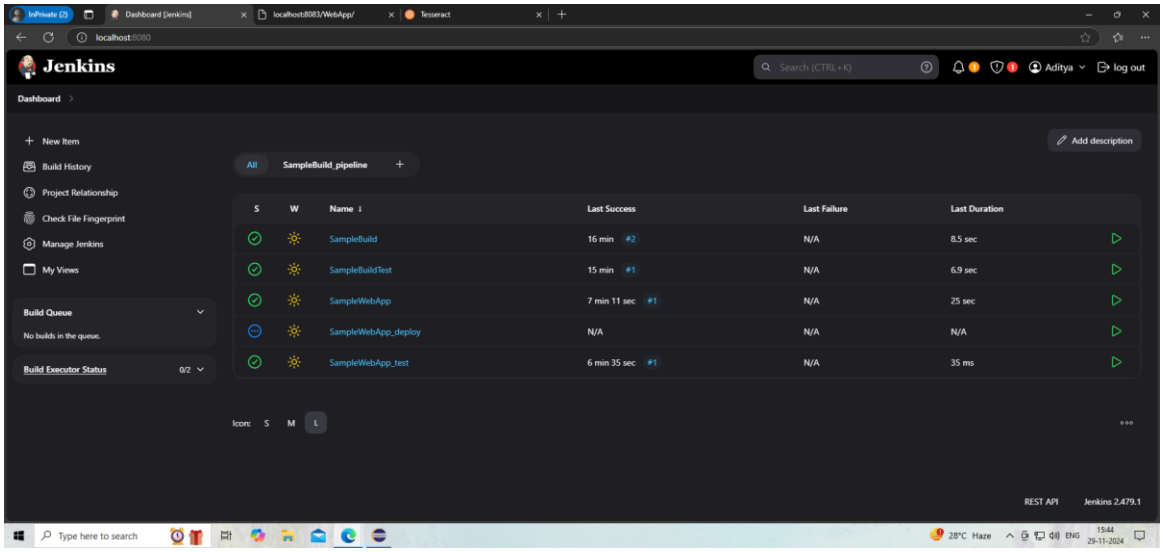

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
All
New Item
New Item
Enter an item name
SampleWebApp_deploy
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK

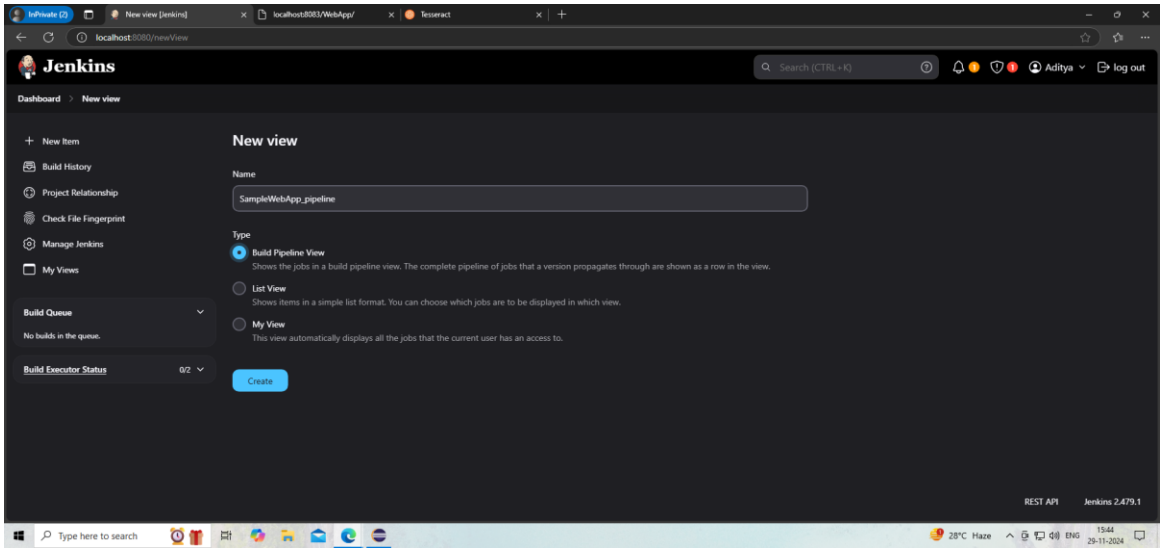

Dashboard
SampleWebApp_deploy
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Advanced
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Copy artifacts from another project
Project name
SampleWebApp_test
Which build
Latest successful build
Stable build only
Artifacts to copy
**/*
Save
Apply

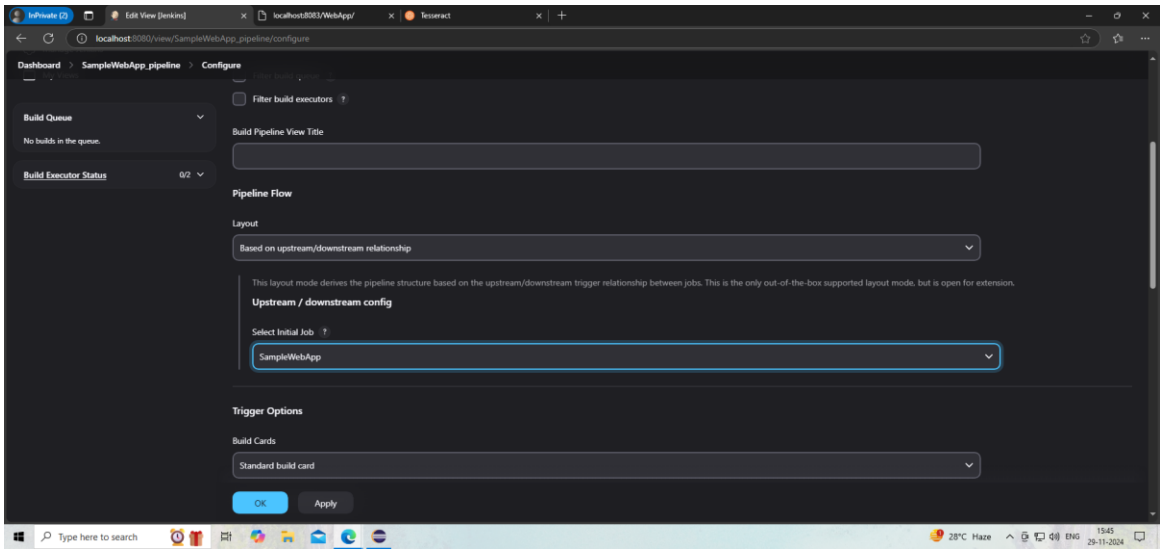

Dashboard > SampleWebApp_deploy > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Advanced
Add build step
Post-build Actions
Deploy war/ear to a container
WAR/EAR files
**/*.war
Context path
samplewebprojectmaven
Containers
Add Container
Deploy on failure
Save
Apply

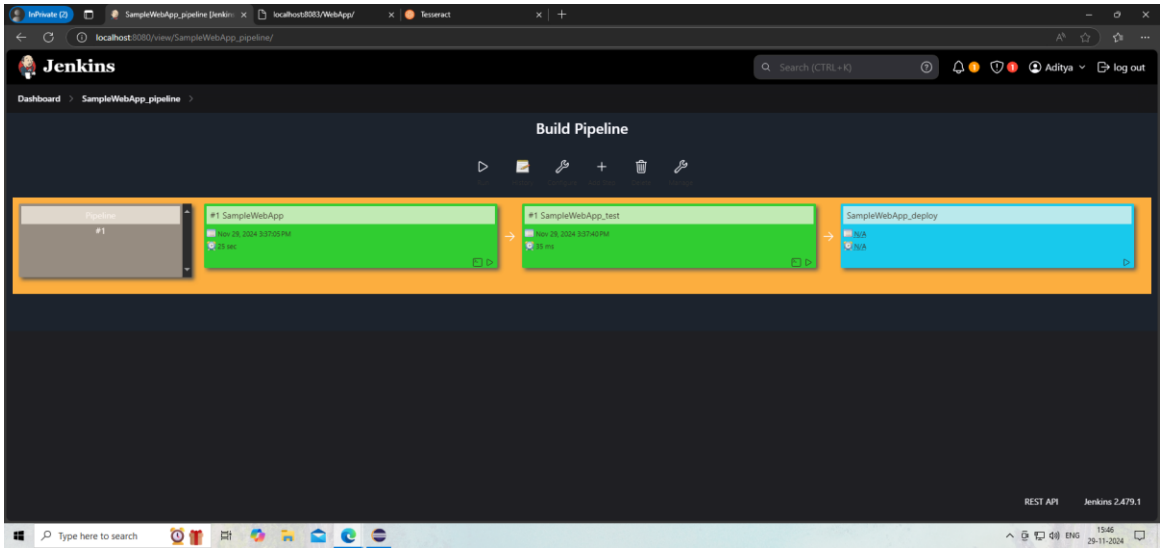

Jenkins Credentials Provider: Jenkins
Kind
Username with password
Scope
Global (Jenkins, nodes, items, all child items, etc)
Username
admin
Treat username as secret
Password
ID

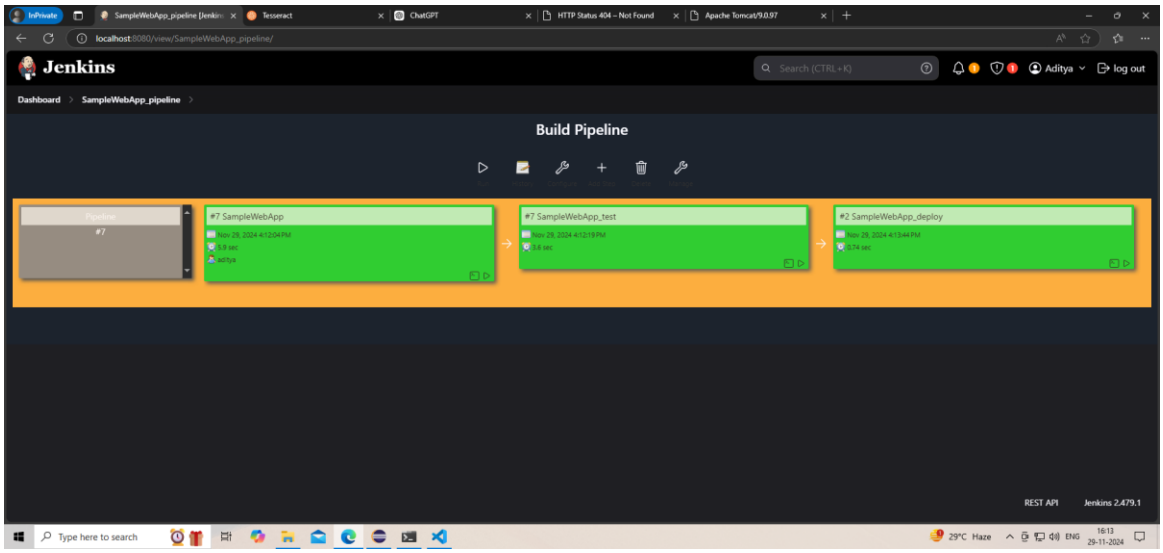

Dashboard > SampleWebApp_deploy > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
samplewebprojectmaven
Containers
Tomcat 9.x Remote
Credentials
admin/******
+ Add
Tomcat URL
http://localhost:8083
Advanced
Add Container
Deploy on failure
Save
Apply

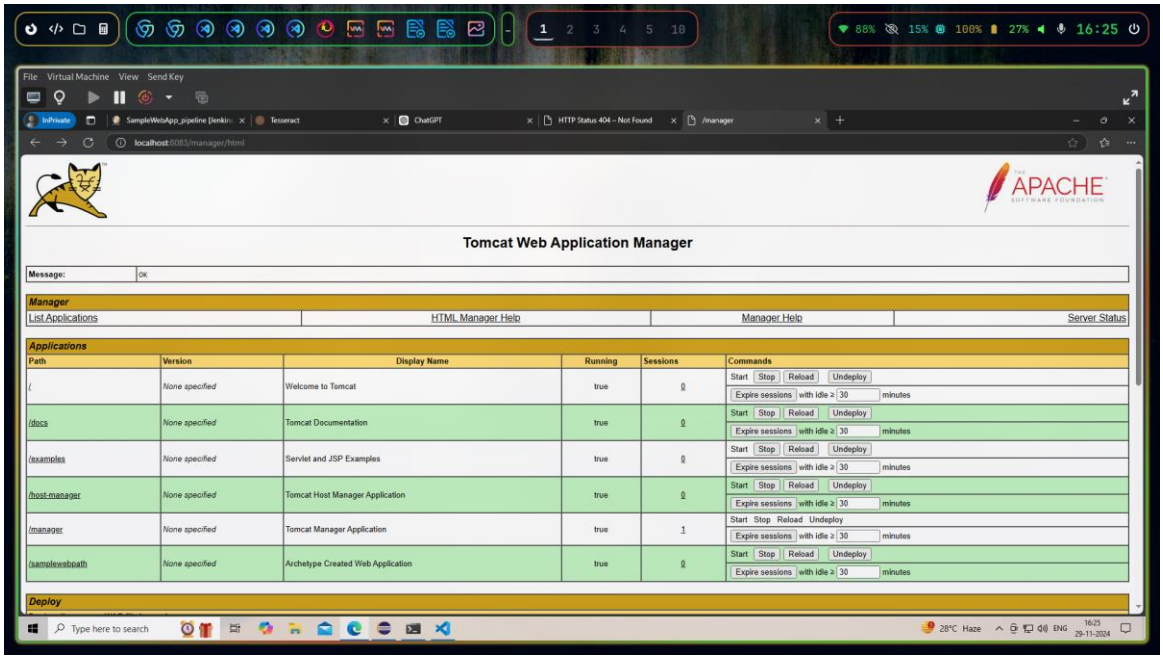

Jenkins
Dashboard
New Item
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
Add description
All
SampleBuild_pipeline
S
W
Name
Last Success
Last Failure
Last Duration
SampleBuild
16 min #2
N/A
8.5 sec
SampleBuildTest
15 min #1
N/A
6.9 sec
SampleWebApp
7 min 11 sec #1
N/A
25 sec
SampleWebApp_deploy
N/A
N/A
N/A
SampleWebApp_test
6 min 35 sec #1
N/A
35 ms
REST API
Jenkins 2.479.1

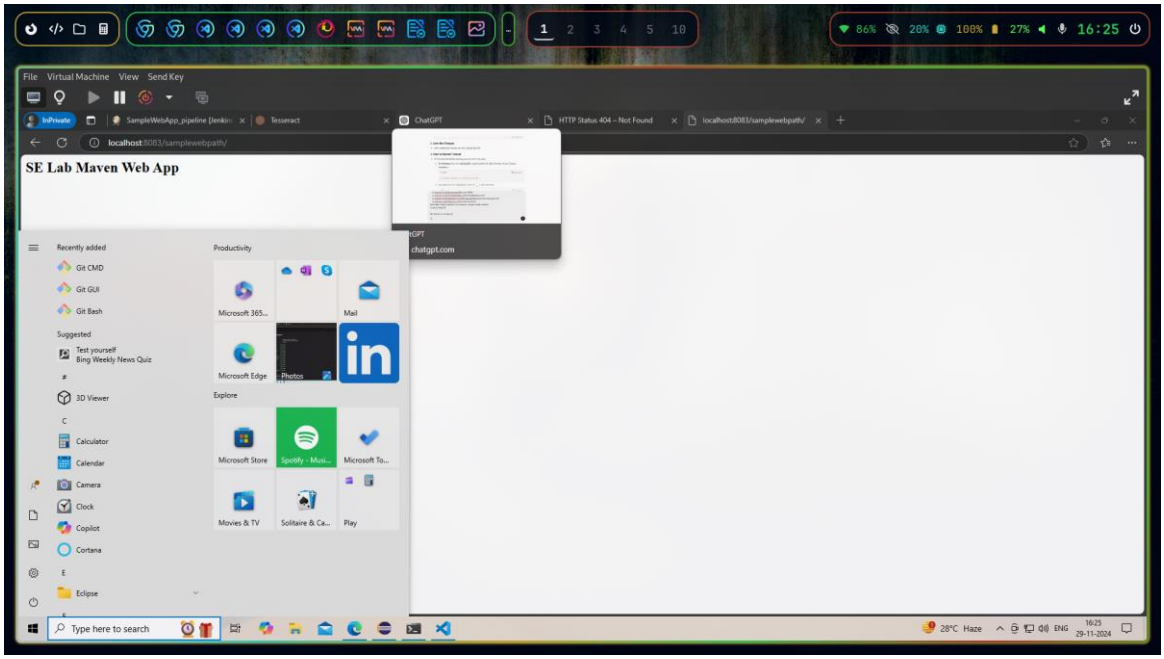

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
New view
New Item
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
0/2
New view
Name
SampleWebApp_pipeline
Type
Build Pipeline View
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.
List View
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.
My View
This view automatically displays all the jobs that the current user has an access to.
Create
REST API
Jenkins 2.479.1

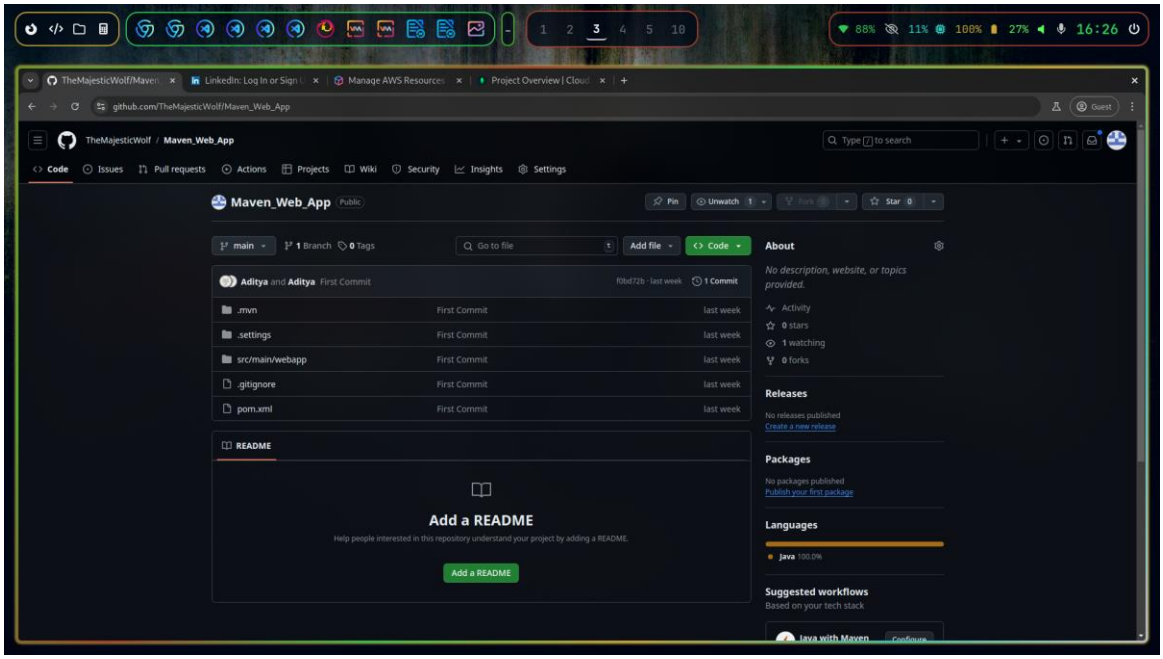

Dashboard
SampleWebApp_pipeline
Configure
Build Queue
No builds in the queue.
Build Executor Status
0/2
Filter build executors
Build Pipeline View Title
Pipeline Flow
Layout
Based on upstream/downstream relationship
This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.
Upstream / downstream config
Select Initial Job
SampleWebApp
Trigger Options
Build Cards
Standard build card
OK
Apply

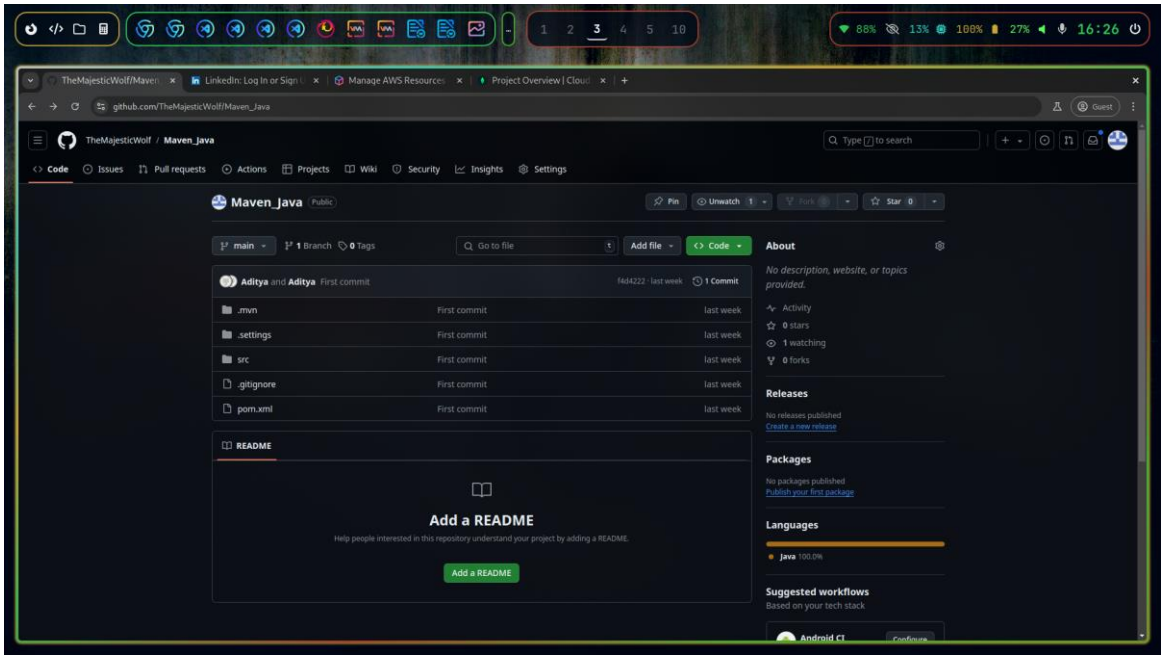

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
SampleWebApp_pipeline
Build Pipeline
Pipeline
#1
#1 SampleWebApp
#1 SampleWebApp_test
SampleWebApp_deploy
N/A
N/A
REST API
Jenkins 2.479.1
Type here to search
ENG
15:46
29-11-2024


Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
SampleWebApp_pipeline
Build Pipeline
Pipeline
#7
#7 SampleWebApp
#7 SampleWebApp_test
#2 SampleWebApp_deploy
REST API
Jenkins 2.479.1
Type here to search
29°C Haze
ENG
16:13
29-11-2024

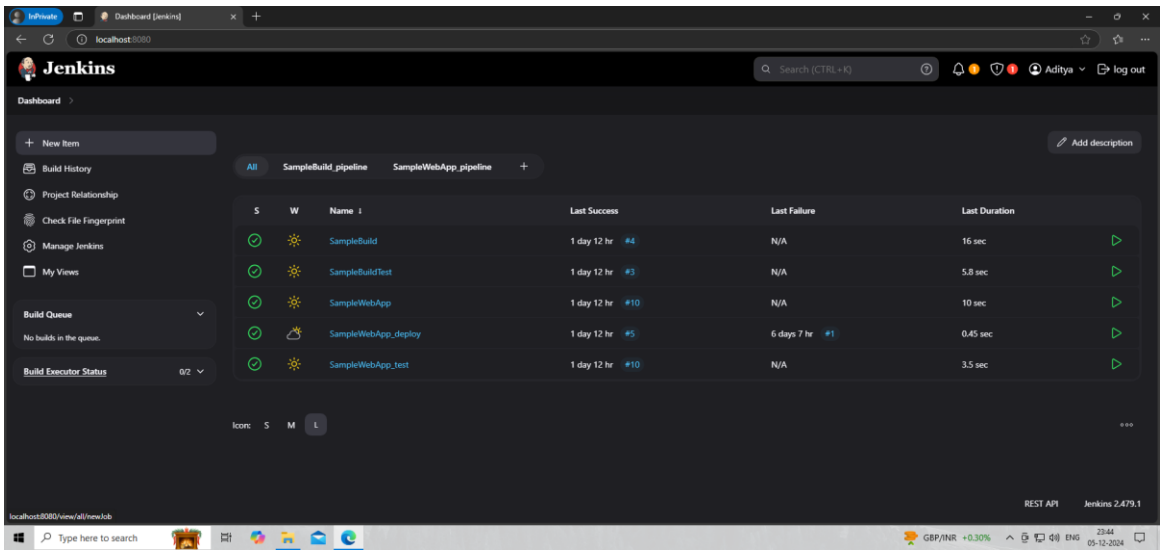

Tomcat Web Application Manager
Message: OK
Manager
List Applications | HTML Manager Help | Manager Help | Server Status
Applications
Path | Version | Display Name | Running | Sessions | Commands
/ | None specified | Welcome to Tomcat | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/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
/samplewebpath | None specified | Archetype Created Web Application | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
Deploy

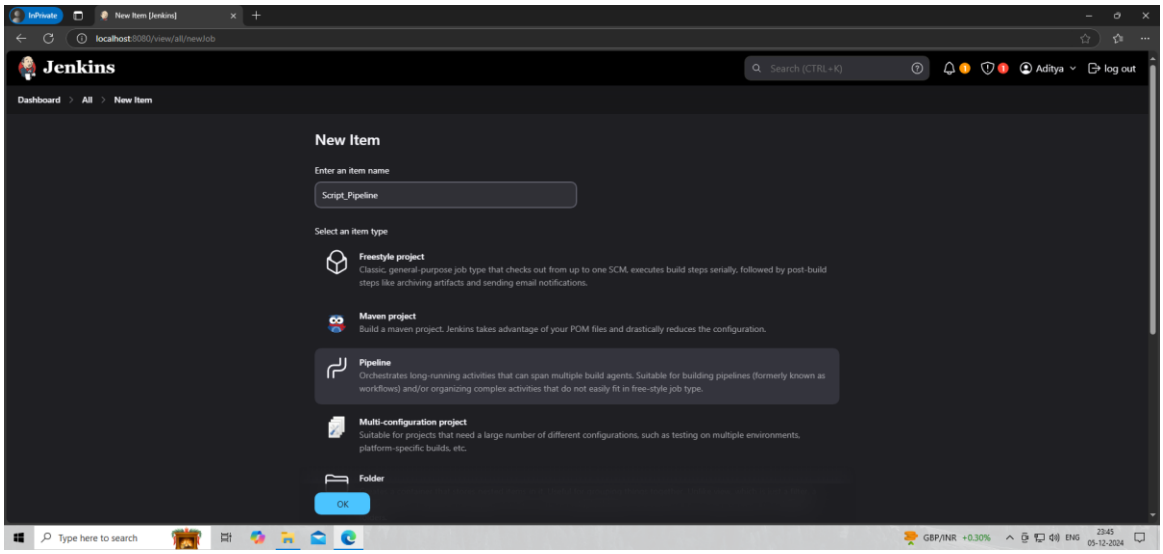

SE Lab Maven Web App

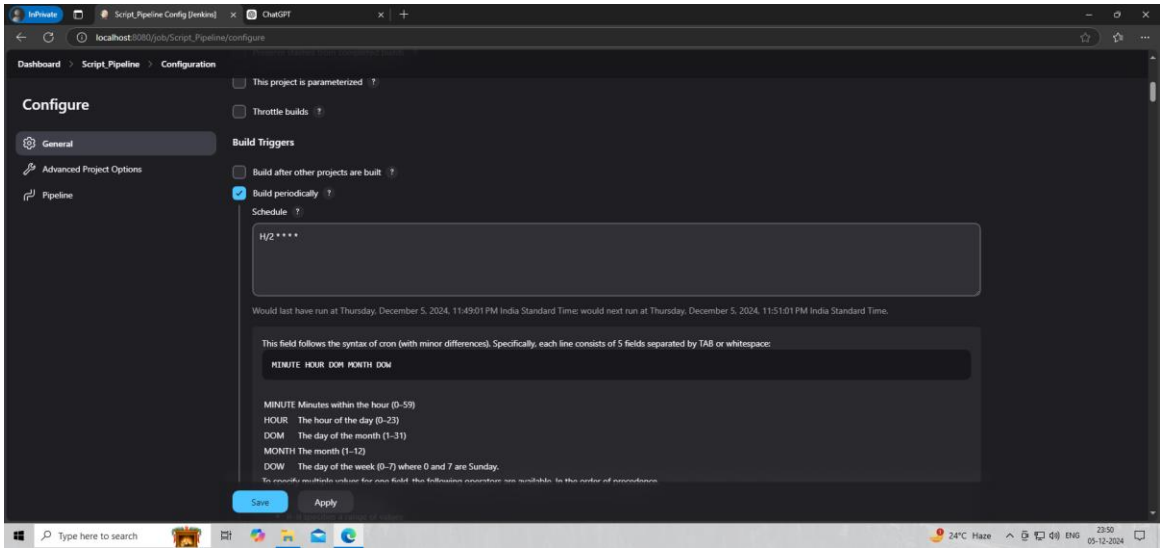


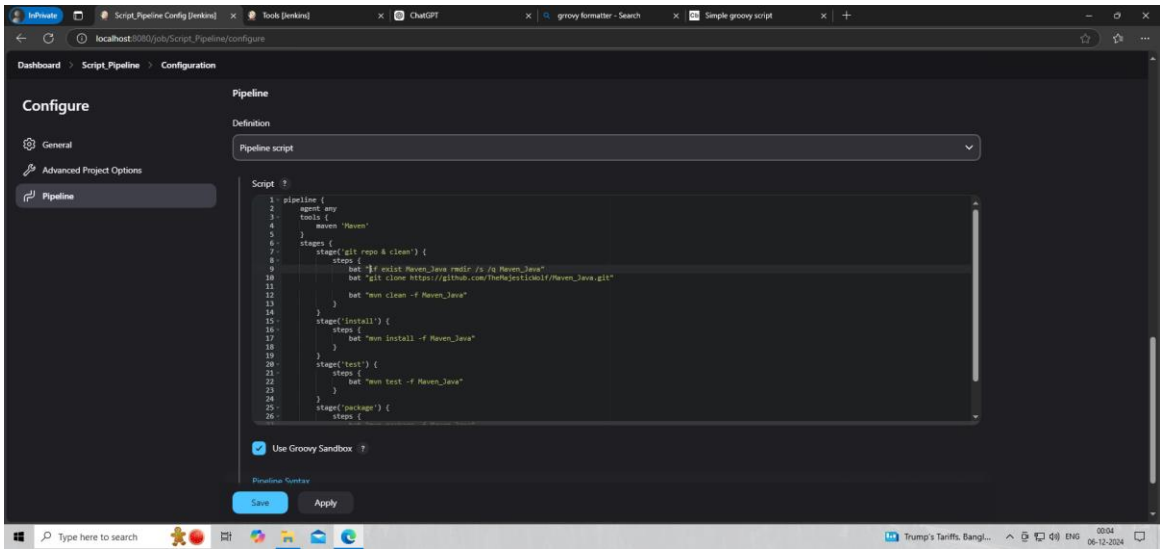



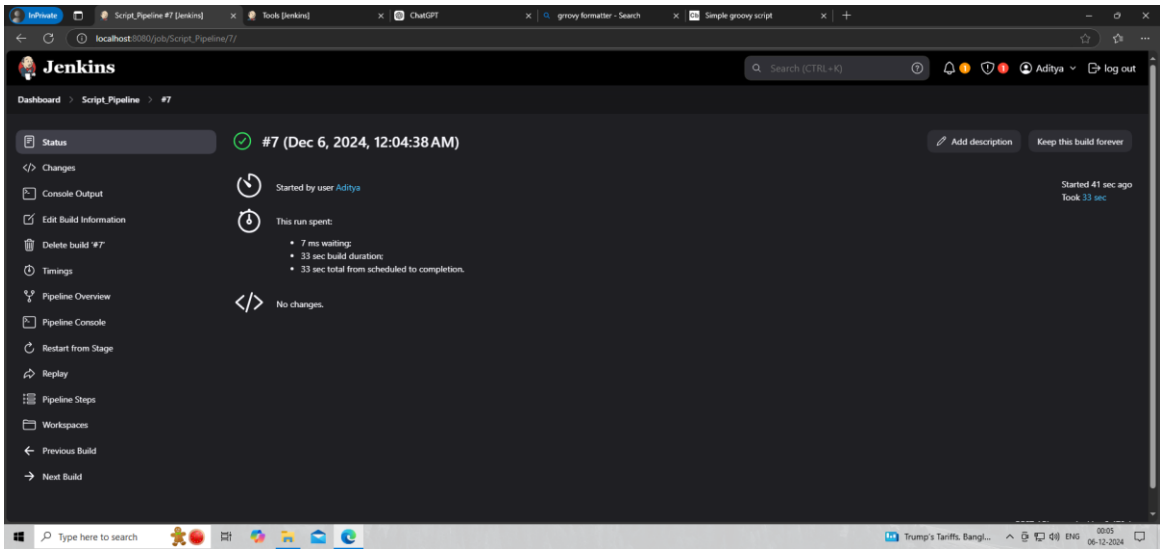

# Jenkins Scripted Pipeline with Poll SCM

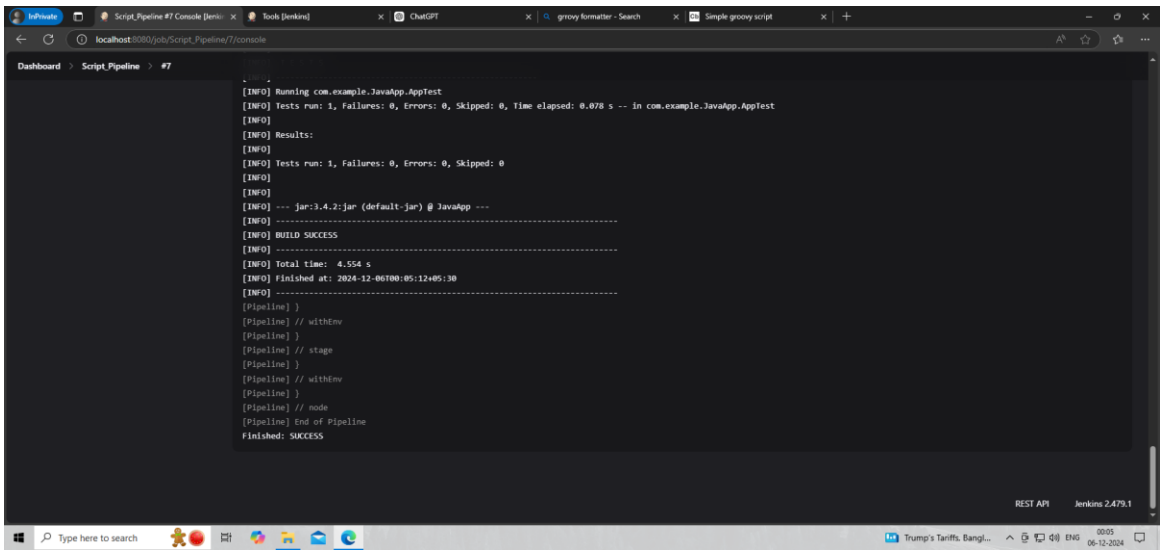

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
New Item
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status 0/2
Add description
All
SampleBuild_pipeline
SampleWebApp_pipeline
S W Name ↓ Last Success Last Failure Last Duration
SampleBuild 1 day 12 hr #4 N/A 16 sec
SampleBuildTest 1 day 12 hr #3 N/A 5.8 sec
SampleWebApp 1 day 12 hr #10 N/A 10 sec
SampleWebApp_deploy 1 day 12 hr #5 6 days 7 hr #1 0.45 sec
SampleWebApp_test 1 day 12 hr #10 N/A 3.5 sec
Icon: S M L
REST API
Jenkins 2.479.1

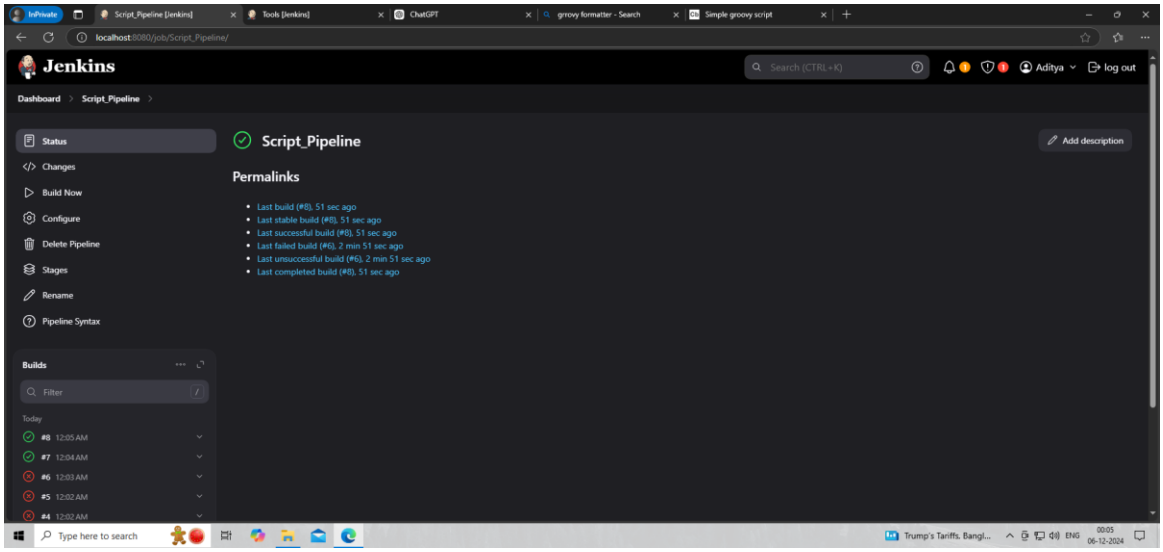

Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard > All > New Item
New Item
Enter an item name
Script_Pipeline
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK


Script_Pipeline Config [Jenkins]
ChatGPT
localhost:8080/job/Script_Pipeline/configure
Dashboard > Script_Pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
This project is parameterized
Throttle builds
Build Triggers
Build after other projects are built
Build periodically
Schedule
H/2 * * * *
Would last have run at Thursday, December 5, 2024, 11:49:01 PM India Standard Time; would next run at Thursday, December 5, 2024, 11:51:01 PM India Standard Time.
This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:
MINUTE HOUR DOM MONTH DOW
MINUTE Minutes within the hour (0–59)
HOUR The hour of the day (0–23)
DOM The day of the month (1–31)
MONTH The month (1–12)
DOW The day of the week (0–7) where 0 and 7 are Sunday.
Save
Apply
Type here to search
24°C Haze
ENG
23:50
05-12-2024


Script_Pipeline Config [Jenkins]
Tools [Jenkins]
ChatGPT
groovy formatter - Search
Simple groovy script
localhost:8080/job/Script_Pipeline/configure
Dashboard > Script_Pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Pipeline
Definition
Pipeline script
Script
pipeline {
agent any
tools {
maven 'Maven'
}
stages {
stage('git repo & clean') {
steps {
bat "if exist Maven_Java rmdir /s /q Maven_Java"
bat "git clone https://github.com/TheMajesticWolf/Maven_Java.git"
bat "mvn clean -f Maven_Java"
}
}
stage('install') {
steps {
bat "mvn install -f Maven_Java"
}
}
stage('test') {
steps {
bat "mvn test -f Maven_Java"
}
}
stage('package') {
steps {
Use Groovy Sandbox
Pipeline Syntax
Save
Apply
Type here to search
Trump's Tariffs, Bangl...
ENG
00:04
06-12-2024


Jenkins
Search (CTRL+K)
Aditya
log out
Dashboard
Script_Pipeline
#7
Status
Changes
Console Output
Edit Build Information
Delete build '#7'
Timings
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Next Build
#7 (Dec 6, 2024, 12:04:38 AM)
Add description
Keep this build forever
Started by user Aditya
Started 41 sec ago
Took 33 sec
This run spent:
7 ms waiting;
33 sec build duration;
33 sec total from scheduled to completion.
No changes.


Dashboard
Script_Pipeline
#7
[INFO] Running com.example.JavaApp.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.078 s -- in com.example.JavaApp.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.2:jar (default-jar) @ JavaApp ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.554 s
[INFO] Finished at: 2024-12-06T00:05:12+05:30
[INFO] ------------------------------------------------------------------------
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
REST API
Jenkins 2.479.1




Result: Jenkins was installed and successfully configured for CI/CD, automating the build and deployment processes for the Maven projects.

# Experiment-7: Local Deployment of Project using Docker, Kubernetes and Monitoring using Nagios tool

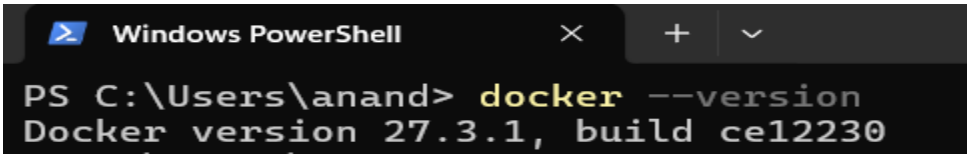
Aim: To explore Docker CLI commands, create and push Docker images and files to DockerHub, run and manage containers using Docker Compose, deploy and scale applications with Minikube, and implement monitoring systems using Nagios.

Introduction:

1. Docker: A platform for containerization that allows developers to package applications and their dependencies into lightweight containers.
    - Docker CLI Commands: Used for managing containers, images, networks, and volumes.
    - DockerHub: A cloud-based repository for sharing Docker images.
2. Docker Compose: A tool for defining and managing multi-container applications. It uses a docker-compose.yml file to configure and run multiple containers together.
3. Minikube: A lightweight Kubernetes implementation for deploying and scaling applications locally, simulating a Kubernetes cluster.
4. Nagios: An open-source monitoring system used to track the performance and health of applications, services, and infrastructure.

# 7A - Docker Commands Part-1

1. docker –version && docker




2. docker images

```
PS C:\Users\anand> docker images
REPOSITORY                    TAG       IMAGE ID       CREATED         SIZE
gcr.io/k8s-minikube/kicbase   v0.0.45   aeed0e1d4642   3 months ago    1.28GB
hello-world                   latest    d2c94e258dcb   19 months ago   13.3kB
```

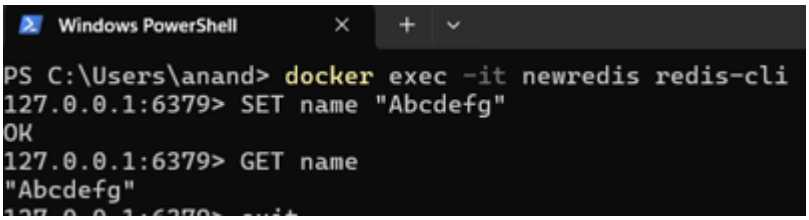
3. docker image rm hello-world

```
PS C:\Users\anand> docker run hello-world

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

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

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

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

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4. docker pull redis

```
PS C:\Users\anand> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
bc0965b23a04: Pull complete
9501a6ec095f: Pull complete
98e7597530ef: Pull complete
75dffa679c9b: Pull complete
3912a88e73c8: Pull complete
141f00d6fee8: Pull complete
4f4fb700ef54: Pull complete
3242f9d5b464: Pull complete
Digest: sha256:ea96c435dc17b011f54c6a883c3c45e7726242b075de61c6fe40a10ae6ae0f83
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\anand>
```

5. docker exec -it my-redis redis-cli



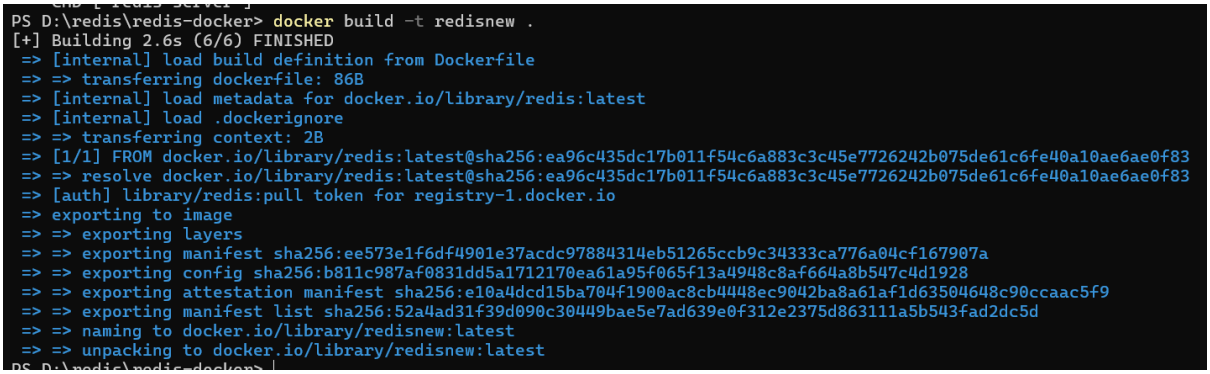

6. docker stop my-redis && docker start my-redis

7. docker rm my-redis && docker image rm redis:latest

```
PS D:\dockerpro\redis> docker rm newredis
newredis
PS D:\dockerpro\redis>
```

# 7A - Docker Commands Part-2

1. docker build -t redisnew .




2. docker run –name myredisnew redisnew && docker ps

```
PS D:\redis\redis-docker> docker run --name myredisnew -d redisnew
038ca4fed49fd6b70ab2a67f3487fa81606e584ebb95e1b117acf97594cd69c4
PS D:\redis\redis-docker>
```

3. docker stop myredisnew

```
PS D:\redis\redis-docker> docker stop myredisnew
myredisnew
PS D:\redis\redis-docker>
```

4. docker login

```
PS D:\redis\redis-docker> docker login
Authenticating with existing credentials...
Login Succeeded
PS D:\redis\redis-docker>
```

5. docker ps -a

```
PS D:\redis\redis-docker> docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED             STATUS                      PORTS     NAMES
038ca4fed49f   redisnew           "docker-entrypoint.s…"   About a minute ago   Exited (0) 44 seconds ago             myredisnew
b320f5d01862   anand-app          "docker-entrypoint.s…"   36 minutes ago       Exited (0) 36 minutes ago             serene_montalcini
50b775d3a3cf   anand-app:latest   "docker-entrypoint.s…"   37 minutes ago       Exited (0) 37 minutes ago             romantic_kepler
4da780b7b41d   hello-world        "/hello"                 3 hours ago          Exited (0) 3 hours ago                funny_panini
da58924546dd   hello-world        "/hello"                 3 hours ago          Exited (0) 3 hours ago                naughty_bhaskara
19f826e3b21f   hello-world        "/hello"                 3 hours ago          Exited (0) 3 hours ago                blissful_curran
PS D:\redis\redis-docker>
```

6. docker commit

```
PS D:\redis\redis-docker> docker commit myredisnew anand/redisnew
sha256:506b72d9a5db8754cf36792c148fe79faff5213ddb8401905c16321e0be0c87
PS D:\redis\redis-docker>
```

```
PS D:\redis\redis-docker> docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
anand/redisnew      latest    506b72d9a5db   17 seconds ago   173MB
anand/anand-app     latest    8306550e6d38   41 minutes ago   1.35GB
anand/myimages      latest    8306550e6d38   41 minutes ago   1.35GB
anand-app           latest    8306550e6d38   41 minutes ago   1.35GB
redisnew            latest    52a4ad31f39d   2 months ago     173MB
redis               latest    ea96c435dc17   2 months ago     173MB
hello-world         latest    305243c73457   19 months ago    24.4kB
```

6. docker images

7. docker push

```
PS D:\redis\redis-docker> docker push anand/redisnew
Using default tag: latest
The push refers to repository [docker.io/anand/redisnew]
bc0965b23a04: Waiting
75dffa679c9b: Waiting
141f00d6fee8: Waiting
4f4fb700ef54: Waiting
9501a6ec095f: Waiting
8242f9d5b464: Waiting
98e7597530ef: Waiting
8912a88e73c8: Waiting
```

8. docker rm myredisnew

```
PS D:\redis\redis-docker> docker rm myredisnew
myredisnew
PS D:\redis\redis-docker>
```

9. docker image rm anand/redis1

```
PS D:\redis\redis-docker> docker rm myredisnew
myredisnew
PS D:\redis\redis-docker>
```

9. docker ps -a
10. docker pull anand/redis1
11. docker run --name myredis -d anand/redis1:latest
12. Running inside the container
13. docker ps -a
14. Stopping container, deleting it and logging out

# 7B - Modify and Push image

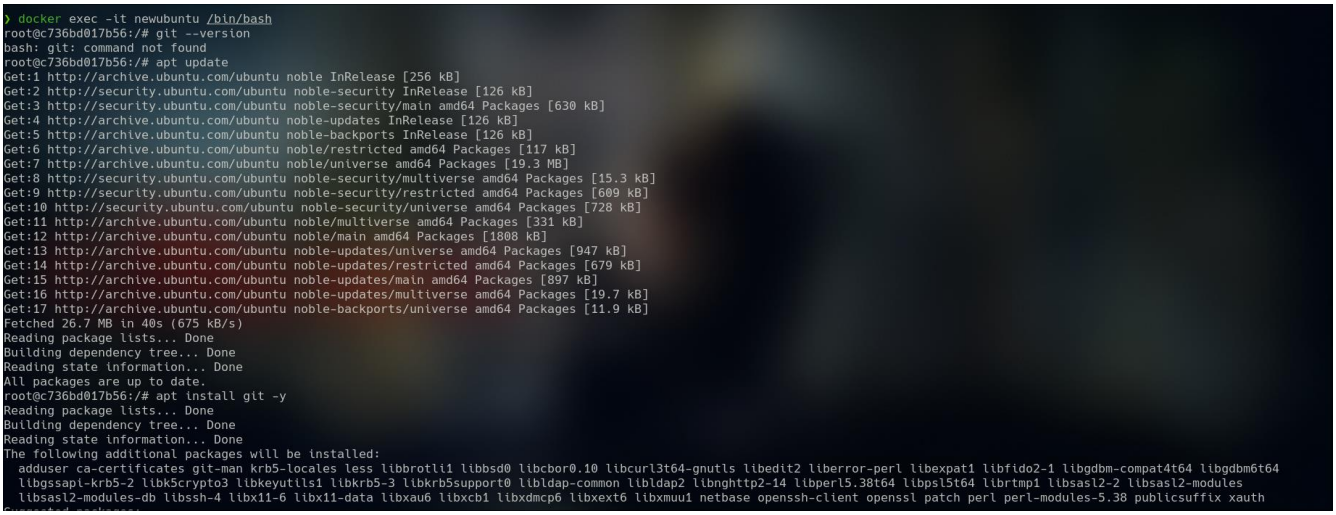
1. docker pull ubuntu

```
❯ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
de44b265507a: Pull complete
Digest: sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redis
✔ 42s 15:06:56
```

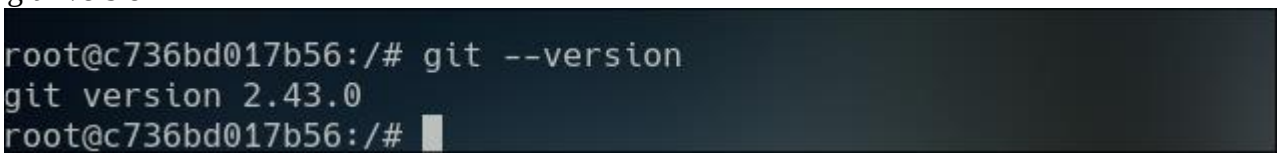
2. docker run -it –name newubuntu -d ubuntu:latest && docker ps

```
❯ docker run -it --name newubuntu -d ubuntu:latest
c736bd017b56a045705a70008c046b483c8d4eaae86bbe233c4a29f8f5ebd801
❯ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS          PORTS                                       NAMES
c736bd017b56   ubuntu:latest         "/bin/bash"              14 seconds ago   Up 13 seconds                                               newubuntu
bdd6dcca08fa   cyber-detective-flask "python flask_server…"   10 days ago      Up 18 hours     0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   cyber-detective-flask-1
f65f34b81661   cyber-detective-backend "docker-entrypoint.s…" 13 days ago      Up 18 hours     0.0.0.0:6969->6969/tcp, :::6969->6969/tcp   cyber-detective-backend-1
98425af9d1a0   cyber-detective-frontend "docker-entrypoint.s…" 13 days ago     Up 18 hours     0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   cyber-detective-frontend-1

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/Redis
✔ 15:08:30
```

3. apt update && apt install git -y

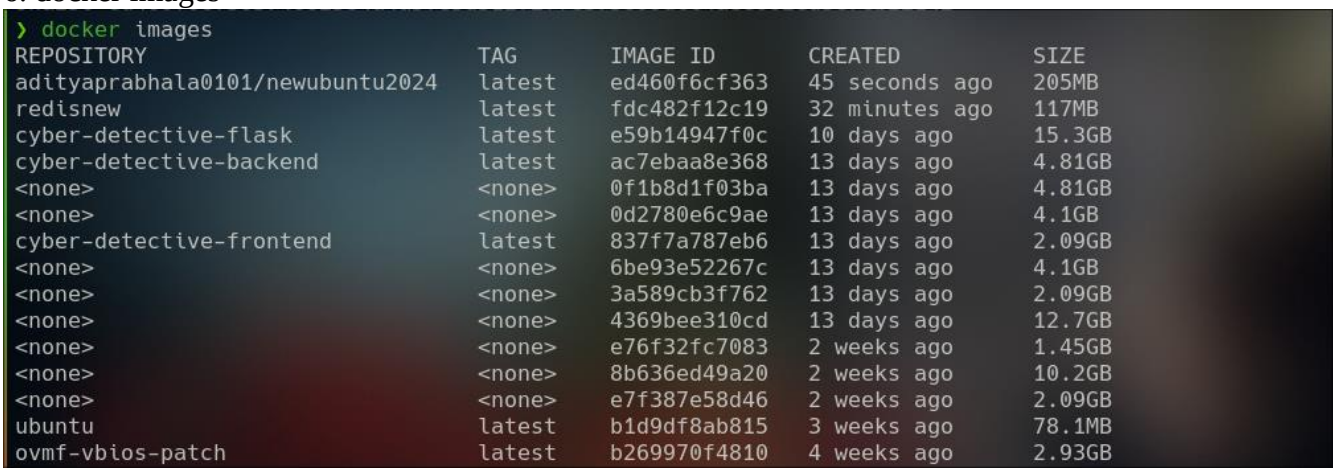



4. git –version

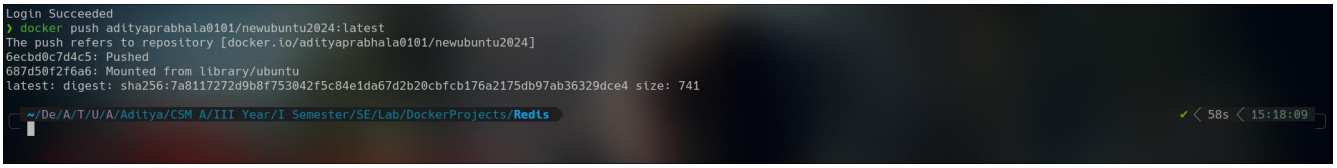



5. docker commit c736bd017b56 adityaprabhala0101/mynewubuntu

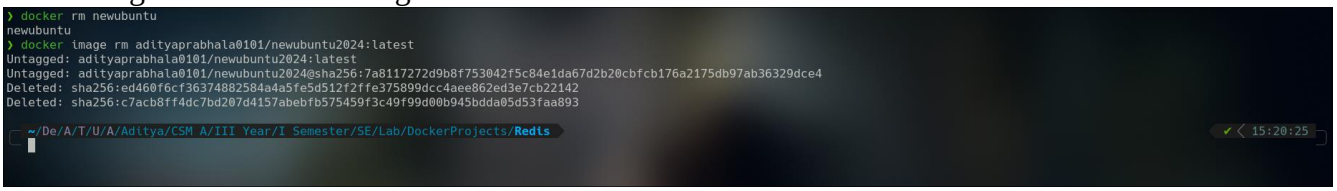

6. docker images



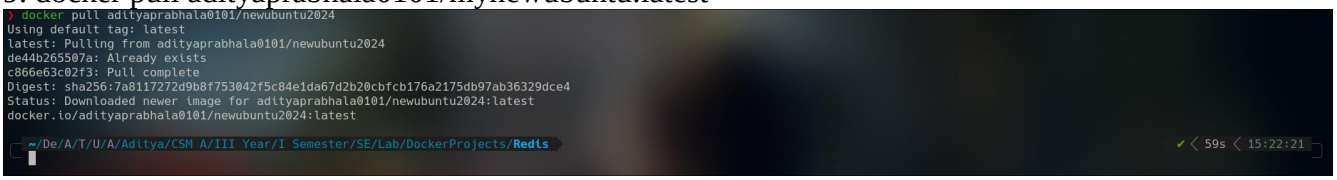

7. docker login && docker push adityaprabhala0101/mynewubuntu:latest



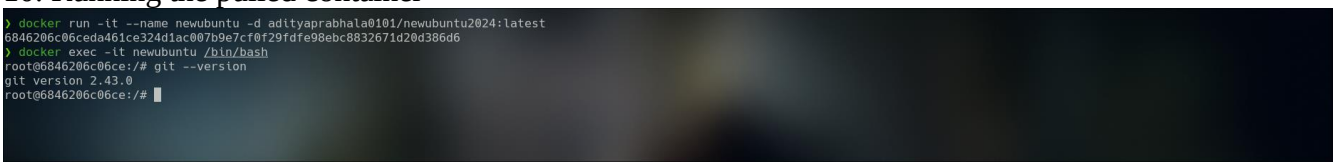

## 8. Deleting containers and images



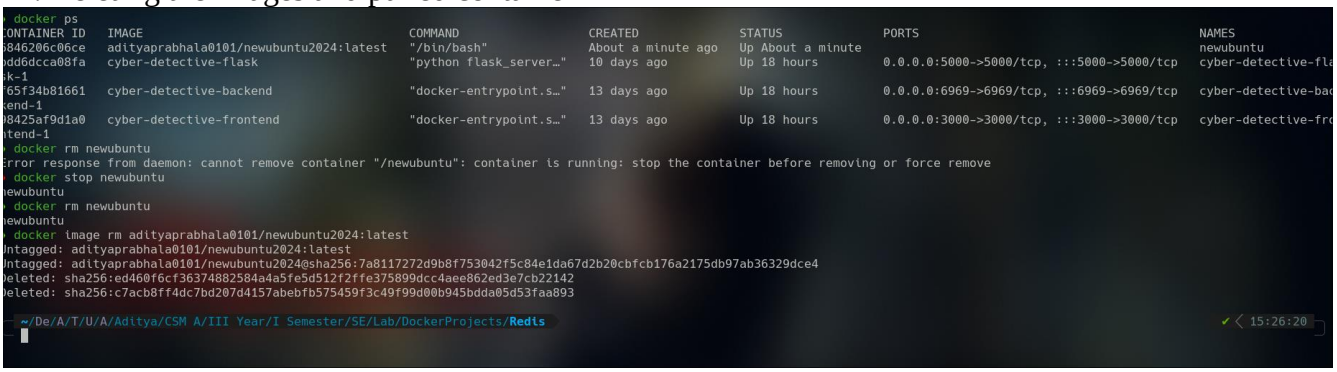

## 9. docker pull adityaprabhala0101/mynewubuntu:latest




## 10. Running the pulled container

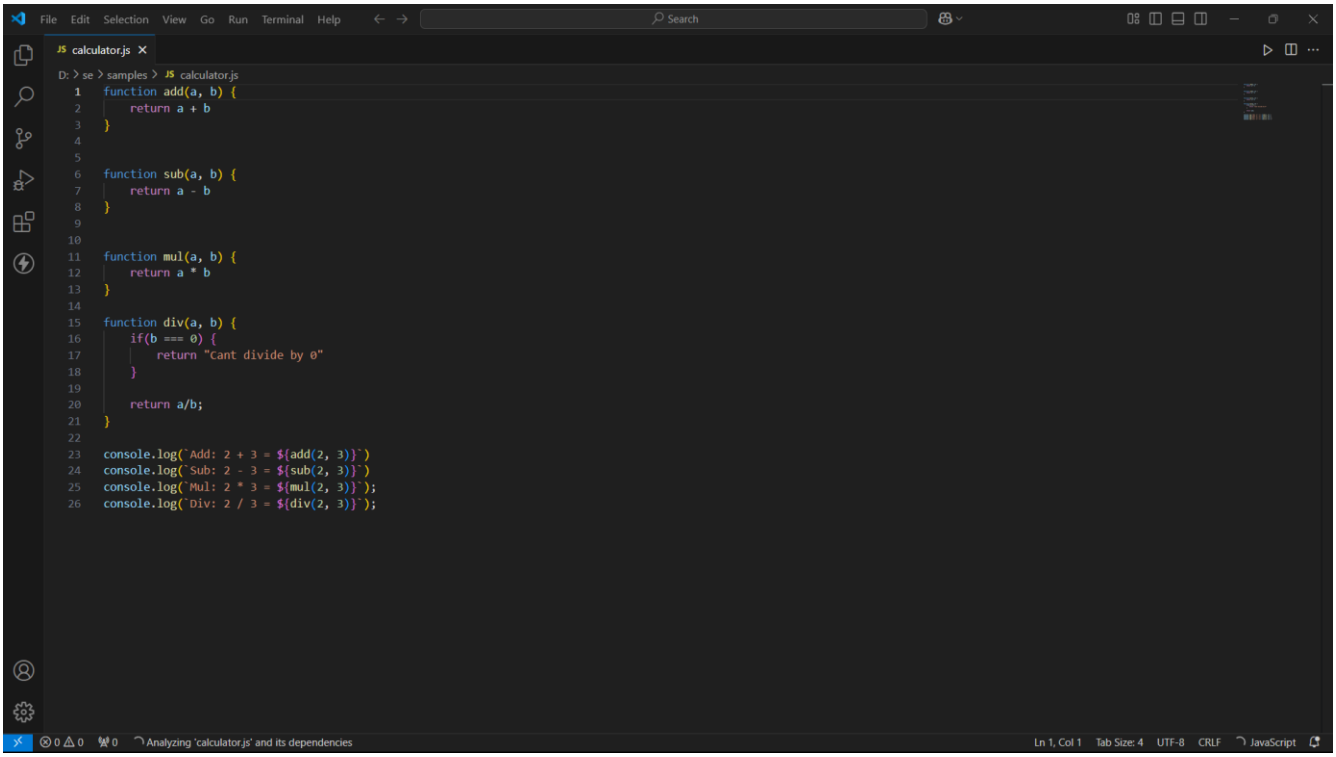



## 11. Deleting the images and pulled container



# 7C - Create and Push Dockerfile image

1. Create calculator.js



2. Create Dockerfile


3. Building docker image

```
❯ docker build -t simple-calc .
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 1/4 : FROM node:22
22: Pulling from library/node
fdf894e782a2: Pull complete
5bd71677db44: Pull complete
551df7f94f9c: Downloading [==============================>                    ]  40.31MB/64.39MB
ce82e98d553d: Downloading [======>                                            ]  25.84MB/211.3MB
551df7f94f9c: Pull complete
ce82e98d553d: Pull complete
9207a94a3132: Pull complete
788306d1e45b: Pull complete
a6916b58c81c: Pull complete
e650db35f240: Pull complete
Digest: sha256:35a5dd72bcac4bce43266408b58a02be6ff0b6098ffa6f5435aeea980a8951d7
Status: Downloaded newer image for node:22
 ---> 5dc60a49c105
Step 2/4 : WORKDIR /app
 ---> Running in 4fd2a0a672df
 ---> Removed intermediate container 4fd2a0a672df
 ---> 06fd5e7b58cd
Step 3/4 : COPY calculator.js .
 ---> a851f86efee2
Step 4/4 : CMD ["node", "calculator.js"]
 ---> Running in 6995cc9eb96f
 ---> Removed intermediate container 6995cc9eb96f
 ---> 76dd9c8fe4f3
Successfully built 76dd9c8fe4f3
Successfully tagged simple-calc:latest
```

4. docker run simple-calc

```
❯ docker run simple-calc:latest
Add: 2 + 3 = 5
Sub: 2 - 3 = -1
Mul: 2 * 3 = 6
Div: 2 / 3 = 0.6666666666666666

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs
✔ 〈 15:47:23
```

5. docker tag

```
Login Succeeded

❯ docker tag simple-calc:latest adityaprabhala0101/simple-calc

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs
✔ 〈 15:48:43
```

6. Pushing to docker hub

```
❯ docker push adityaprabhala0101/simple-calc:latest
The push refers to repository [docker.io/adityaprabhala0101/simple-calc]
614c2b6afbc1: Pushed
41526607e3a7: Pushed
e0dba9f2a2ef: Mounted from library/node
b37454b23b8f: Mounted from library/node
fb6c888e3ccc: Mounted from library/node
952f5ee3867b: Mounted from library/node
0aeeeb7c293d: Mounted from library/node
c81d4fdb67fc: Mounted from library/node
0e82d78b3ea1: Mounted from library/node
301c1bb42cc0: Mounted from library/node
latest: digest: sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d size: 2417

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs
✔ 〈 27s 〈 15:49:41
```

7. Deleting container and images

```
❯ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS                     PORTS                                       NAMES
956c8552a2b0   simple-calc:latest                   "docker-entrypoint.s…"   2 minutes ago   Exited (0) 2 minutes ago                                               nervous_kepler
7b1bd8eb2fef   d2c94e258dcb                         "/hello"                 6 hours ago     Exited (0) 6 hours ago                                                 modest_ritchie
3a3fc21e89b0   d2c94e258dcb                         "/hello"                 6 days ago      Exited (0) 6 days ago                                                  frosty_mayer
f6dada14956b   d2c94e258dcb                         "/hello"                 6 days ago      Exited (0) 6 days ago                                                  festive_margulis
40effa25bd57   cyber-detective-flask:latest         "/bin/bash"              10 days ago     Exited (130) 10 days ago                                               romantic_lamarr
c3ca95b4a5e6   cyber-detective-flask:latest         "/bin/bash"              10 days ago     Exited (130) 10 days ago                                               sad_edison
00146e90875d   cyber-detective-flask:latest         "/bin/bash"              10 days ago     Created                                                                wizardly_goldwasser
bdd6dcca08fa   cyber-detective-flask                "python flask_server…"   10 days ago     Up 19 hours                0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   cyber-detective-flask-1
fafd287f1036   cyber-detective-flask:latest         "/bin/bash"              10 days ago     Exited (130) 10 days ago                                               musing_hodgkin
4fe86df7c7a9   4369bee310cd                         "/bin/bash"              10 days ago     Exited (130) 10 days ago                                               modest_albattani
24d760a7ba12   cyber-detective-backend:latest       "docker-entrypoint.s…"   12 days ago     Exited (130) 12 days ago                                               tender_ardinghelli
f65f34b81661   cyber-detective-backend              "docker-entrypoint.s…"   13 days ago     Up 19 hours                0.0.0.0:6969->6969/tcp, :::6969->6969/tcp   cyber-detective-backend-1
98425af9d1a0   cyber-detective-frontend             "docker-entrypoint.s…"   13 days ago     Up 19 hours                0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   cyber-detective-frontend-1
518016ecef83   4369bee310cd                         "/bin/bash"              13 days ago     Exited (130) 13 days ago                                               festive_dhawan
a0c13f3125d8   8b636ed49a20                         "/bin/bash"              13 days ago     Exited (130) 13 days ago                                               thirsty_ishizaka
164ef81186a8   e76f32fc7083                         "docker-entrypoint.s…"   13 days ago     Exited (130) 13 days ago                                               goofy_mendeleev
6834539a01d5   e76f32fc7083                         "docker-entrypoint.s…"   13 days ago     Exited (130) 13 days ago                                               inspiring_pascal
d67e53c52dd4   e76f32fc7083                         "docker-entrypoint.s…"   2 weeks ago     Exited (130) 2 weeks ago                                               beautiful_herschel
0ba8b37feff8   8b636ed49a20                         "/bin/bash"              2 weeks ago     Exited (130) 2 weeks ago                                               nifty_meitner
84ae0dfd807a   8b636ed49a20                         "-c /bin/bash"           2 weeks ago     Created                    5000/tcp                                    heuristic_elgamal
b5bcae073ca6   ovmf-vbios-patch                     "/bin/bash"              4 weeks ago     Exited (129) 3 weeks ago                                               sleepy_faraday
58150ac376d5   c46c12b9e2be                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               suspicious_sinoussi
5e07734db381   c46c12b9e2be                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               busy_carver
5cdda040590a   c46c12b9e2be                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               lucid_gould
ad30cd33c535   c46c12b9e2be                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               hungry_archimedes
0a6175ae0274   c46c12b9e2be                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               gallant_perlman
4db082f3032b   beec17136783                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               objective_archimedes
5e6f3164b611   a21e51a49310                         "/bin/bash"              4 weeks ago     Exited (130) 4 weeks ago                                               quirky_montalcini
8ff35a8c051e   ollama/ollama                        "/bin/ollama serve"      5 weeks ago     Exited (0) 3 weeks ago                                                 ollama-gpu
159511a6b078   ollama/ollama                        "/bin/ollama serve"      5 weeks ago     Exited (0) 3 weeks ago                                                 ollama-cpu
❯ docker rm 956c8552a2b0 && docker image rm adityaprabhala0101/simple-calc:latest
956c8552a2b0
Untagged: adityaprabhala0101/simple-calc:latest
Untagged: adityaprabhala0101/simple-calc@sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs                    ✔ 〈 NORMAL 〈 15:51:01
```

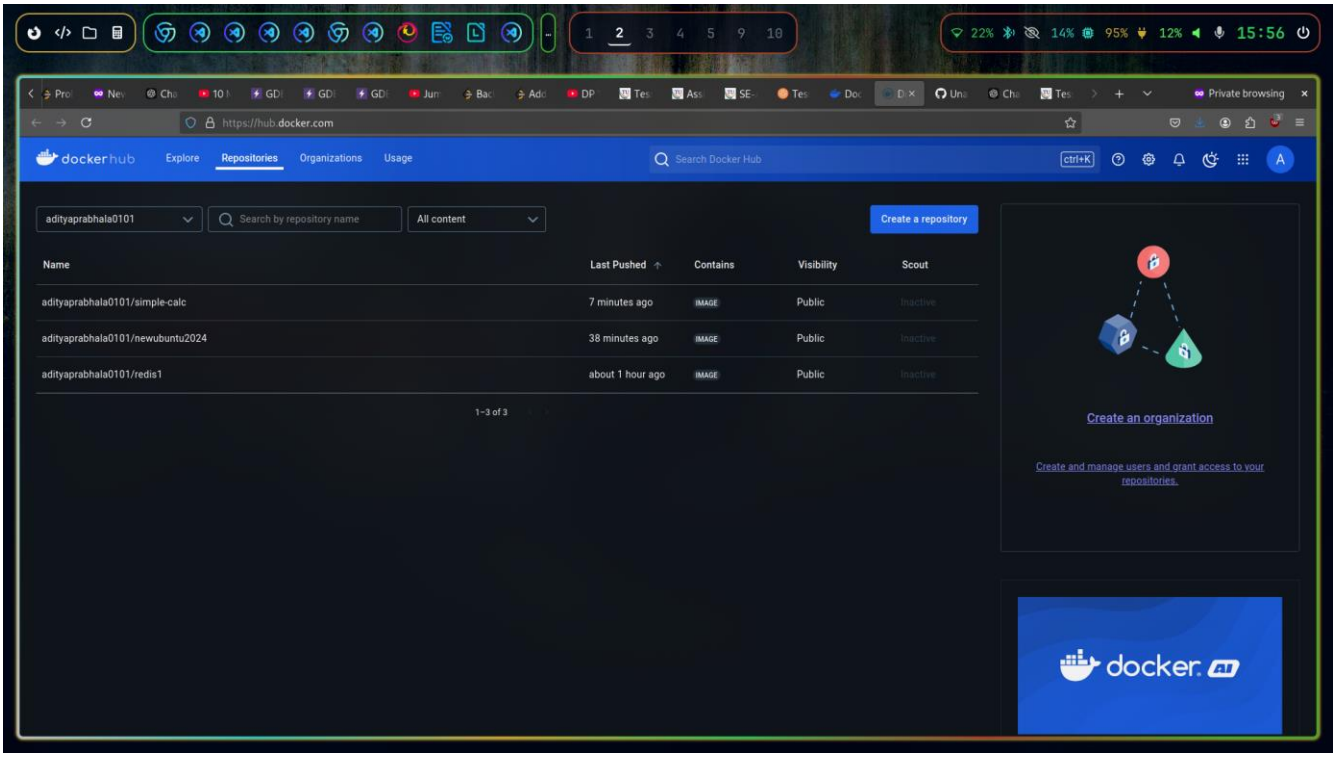
8. docker pull adityaprabhala0101/simple-calc && docker run simple-calc

```
❯ docker pull adityaprabhala0101/simple-calc
Using default tag: latest
latest: Pulling from adityaprabhala0101/simple-calc
Digest: sha256:949adad465edb676117097ced57d333f58a6c151af1231910d31df2eb03c990d
Status: Downloaded newer image for adityaprabhala0101/simple-calc:latest
docker.io/adityaprabhala0101/simple-calc:latest
❯ docker run simple-calc:latest
Add: 2 + 3 = 5
Sub: 2 - 3 = -1
Mul: 2 * 3 = 6
Div: 2 / 3 = 0.6666666666666666

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs                    ✔ 〈 15:52:31
```

9. Deleting the containers and removing images

```
❯ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS                      PORTS                                       NAMES
05a5b95f51a4   simple-calc:latest                   "docker-entrypoint.s…"   46 seconds ago   Exited (0) 45 seconds ago                                               sleepy_euclid
7b1bd8eb2fef   d2c94e258dcb                         "/hello"                 6 hours ago      Exited (0) 6 hours ago                                                  modest_ritchie
3a3fc21e89b0   d2c94e258dcb                         "/hello"                 6 days ago       Exited (0) 6 days ago                                                   frosty_mayer
f6dada14956b   d2c94e258dcb                         "/hello"                 6 days ago       Exited (0) 6 days ago                                                   festive_margulis
40effa25bd57   cyber-detective-flask:latest         "/bin/bash"              10 days ago      Exited (130) 10 days ago                                                romantic_lamarr
c3ca95b4a5e6   cyber-detective-flask:latest         "/bin/bash"              10 days ago      Exited (130) 10 days ago                                                sad_edison
00146e90875d   cyber-detective-flask:latest         "/bin/bash"              10 days ago      Created                                                                 wizardly_goldwasser
bdd6dcca08fa   cyber-detective-flask                "python flask_server…"   10 days ago      Up 19 hours                 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   cyber-detective-flask-1
fafd287f1036   cyber-detective-flask:latest         "/bin/bash"              10 days ago      Exited (130) 10 days ago                                                musing_hodgkin
4fe86df7c7a9   4369bee310cd                         "/bin/bash"              10 days ago      Exited (130) 10 days ago                                                modest_albattani
24d760a7ba12   cyber-detective-backend:latest       "docker-entrypoint.s…"   12 days ago      Exited (130) 12 days ago                                                tender_ardinghelli
f65f34b81661   cyber-detective-backend              "docker-entrypoint.s…"   13 days ago      Up 19 hours                 0.0.0.0:6969->6969/tcp, :::6969->6969/tcp   cyber-detective-backend-1
98425af9d1a0   cyber-detective-frontend             "docker-entrypoint.s…"   13 days ago      Up 19 hours                 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   cyber-detective-frontend-1
518016ecef83   4369bee310cd                         "/bin/bash"              13 days ago      Exited (130) 13 days ago                                                festive_dhawan
a0c13f3125d8   8b636ed49a20                         "/bin/bash"              13 days ago      Exited (130) 13 days ago                                                thirsty_ishizaka
164ef81186a8   e76f32fc7083                         "docker-entrypoint.s…"   13 days ago      Exited (130) 13 days ago                                                goofy_mendeleev
6834539a01d5   e76f32fc7083                         "docker-entrypoint.s…"   13 days ago      Exited (130) 13 days ago                                                inspiring_pascal
d67e53c52dd4   e76f32fc7083                         "docker-entrypoint.s…"   2 weeks ago      Exited (130) 2 weeks ago                                                beautiful_herschel
0ba8b37feff8   8b636ed49a20                         "/bin/bash"              2 weeks ago      Exited (130) 2 weeks ago                                                nifty_meitner
84ae0dfd807a   8b636ed49a20                         "-c /bin/bash"           2 weeks ago      Created                     5000/tcp                                    heuristic_elgamal
b5bcae073ca6   ovmf-vbios-patch                     "/bin/bash"              4 weeks ago      Exited (129) 3 weeks ago                                                sleepy_faraday
58150ac376d5   c46c12b9e2be                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                suspicious_sinoussi
5e07734db381   c46c12b9e2be                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                busy_carver
5cdda040590a   c46c12b9e2be                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                lucid_gould
ad30cd33c535   c46c12b9e2be                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                hungry_archimedes
0a6175ae0274   c46c12b9e2be                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                gallant_perlman
4db082f3032b   beec17136783                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                objective_archimedes
5e6f3164b611   a21e51a49310                         "/bin/bash"              4 weeks ago      Exited (130) 4 weeks ago                                                quirky_montalcini
8ff35a8c051e   ollama/ollama                        "/bin/ollama serve"      5 weeks ago      Exited (0) 3 weeks ago                                                  ollama-gpu
159511a6b078   ollama/ollama                        "/bin/ollama serve"      5 weeks ago      Exited (0) 3 weeks ago                                                  ollama-cpu
❯ docker rm 05a5b95f51a4 && docker image rm simple-calc:latest && docker logout
05a5b95f51a4
Untagged: simple-calc:latest
Removing login credentials for https://index.docker.io/v1/

~/De/A/T/U/A/Aditya/CSM A/III Year/I Semester/SE/Lab/DockerProjects/samplejs                    ✔ 〈 15:53:40
```

## 10. DockerHub

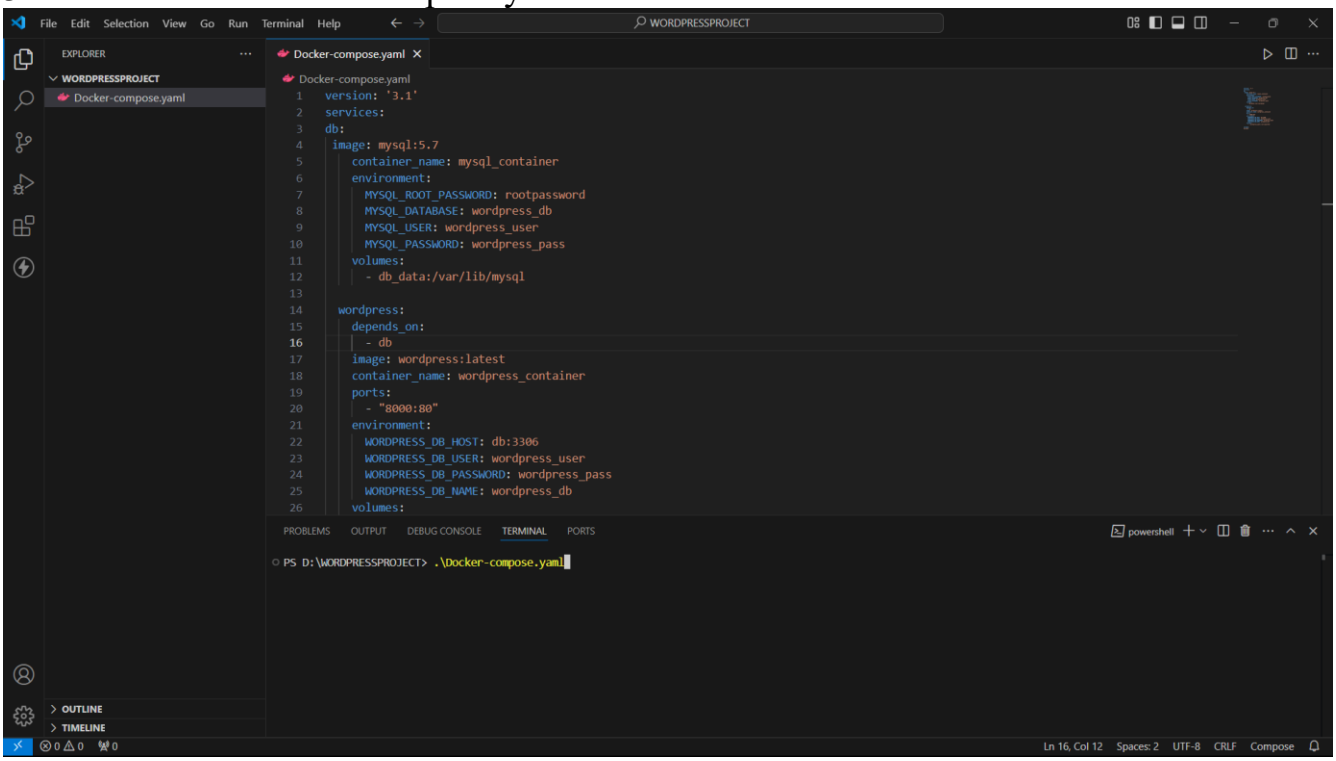


# 7D – Multiple containers using Docker Compose

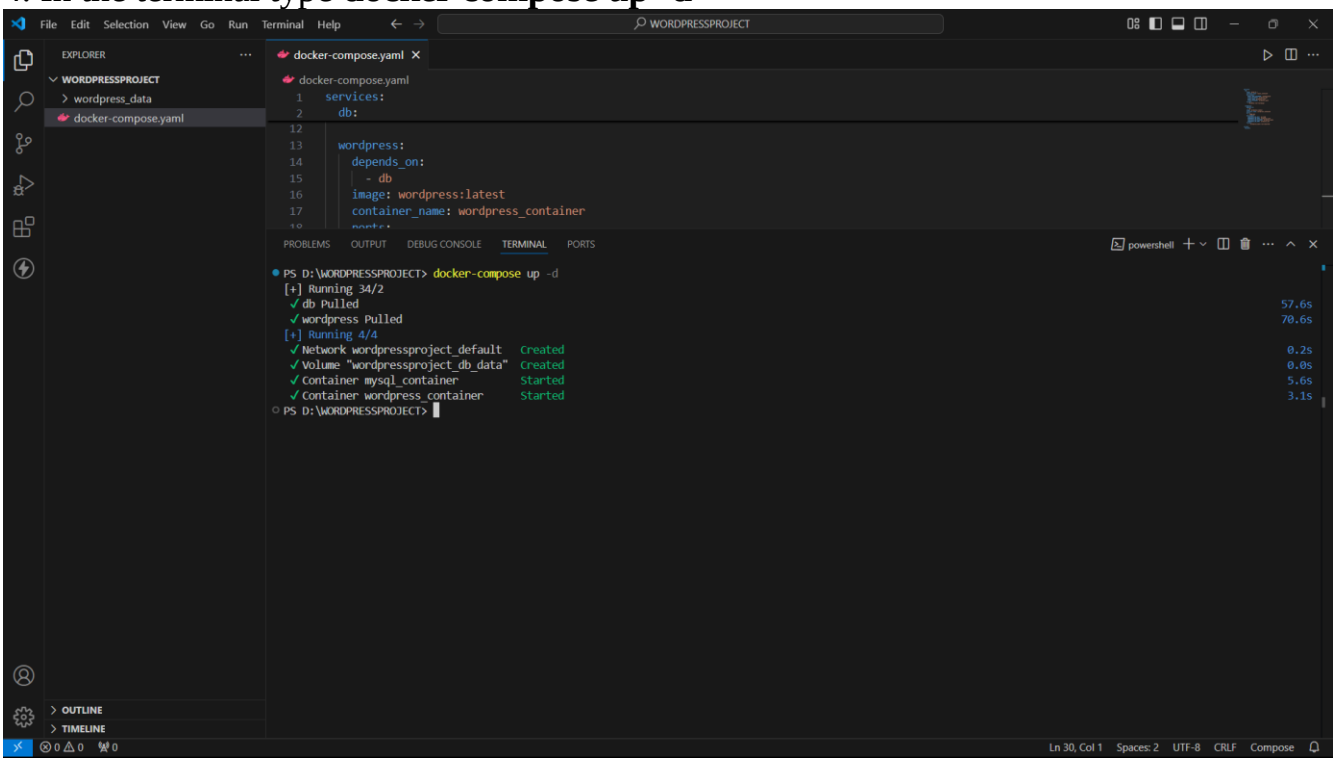
1. Create a new directory

2. Open the directory in a text editor

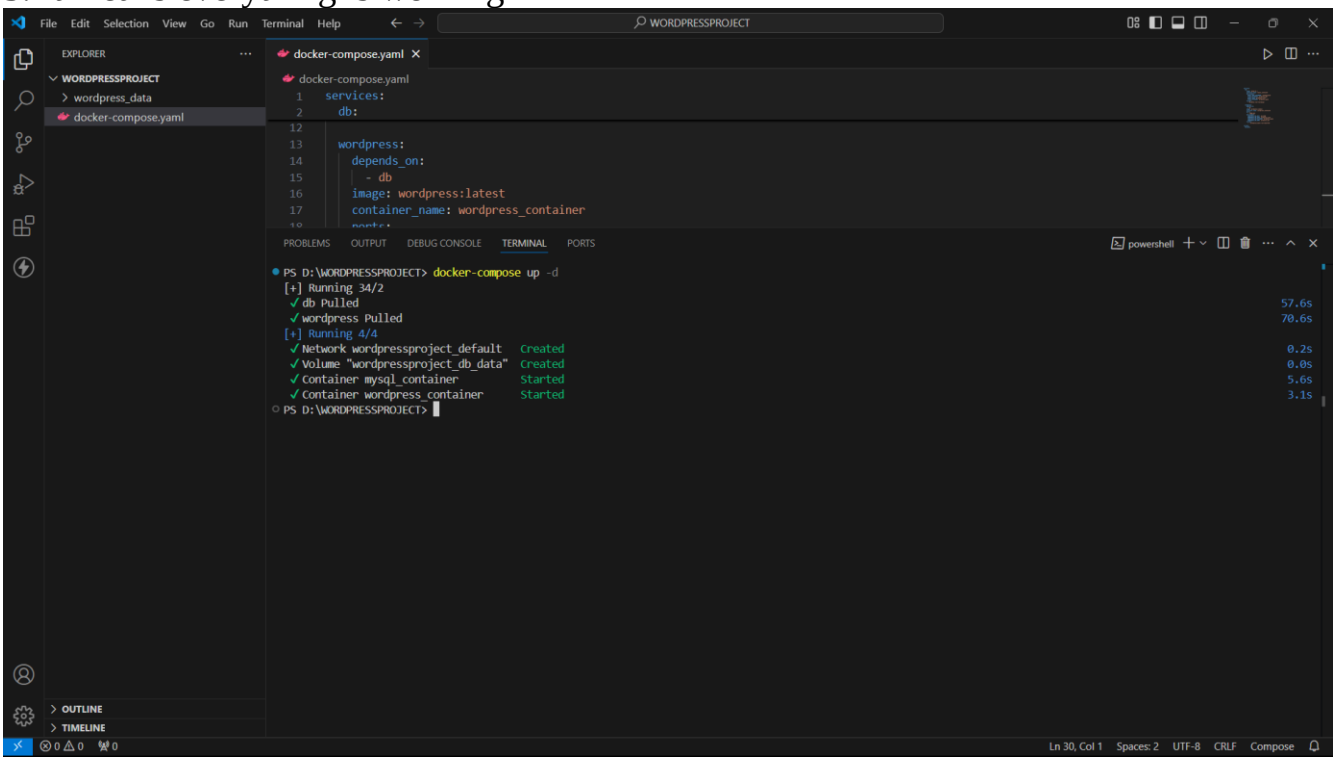
3. Create a file “docker-compose.yaml”



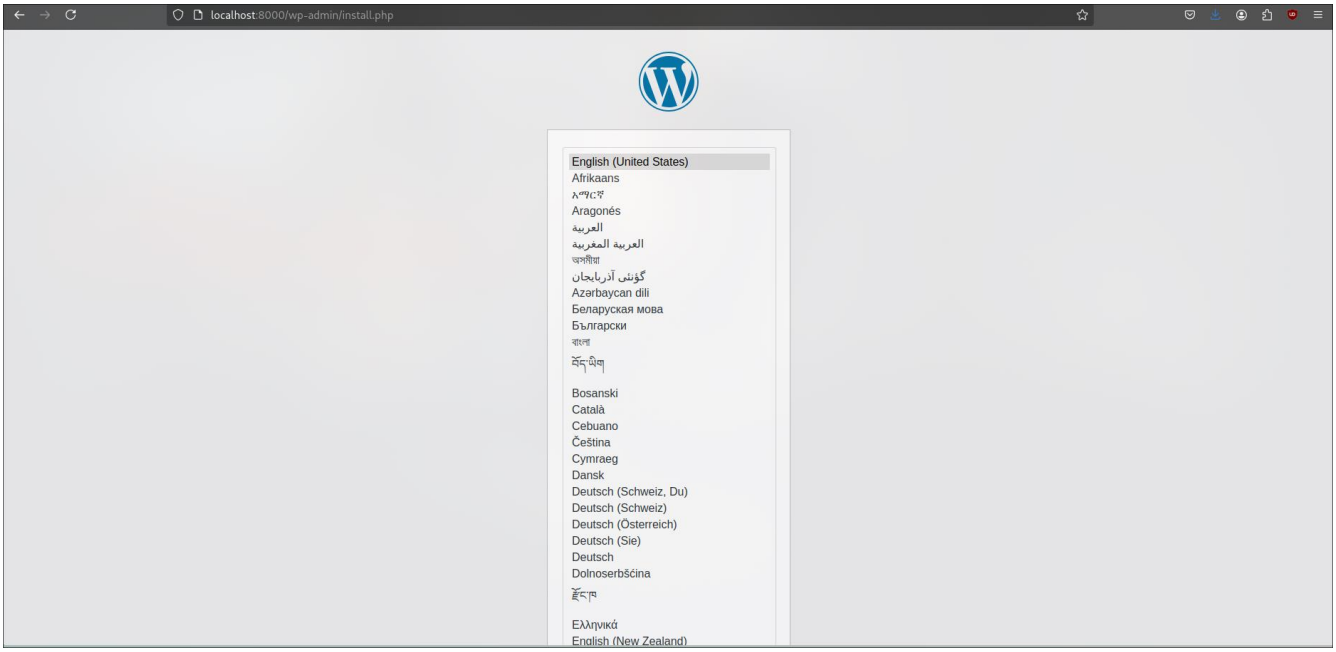
4. In the terminal type **docker compose up -d**



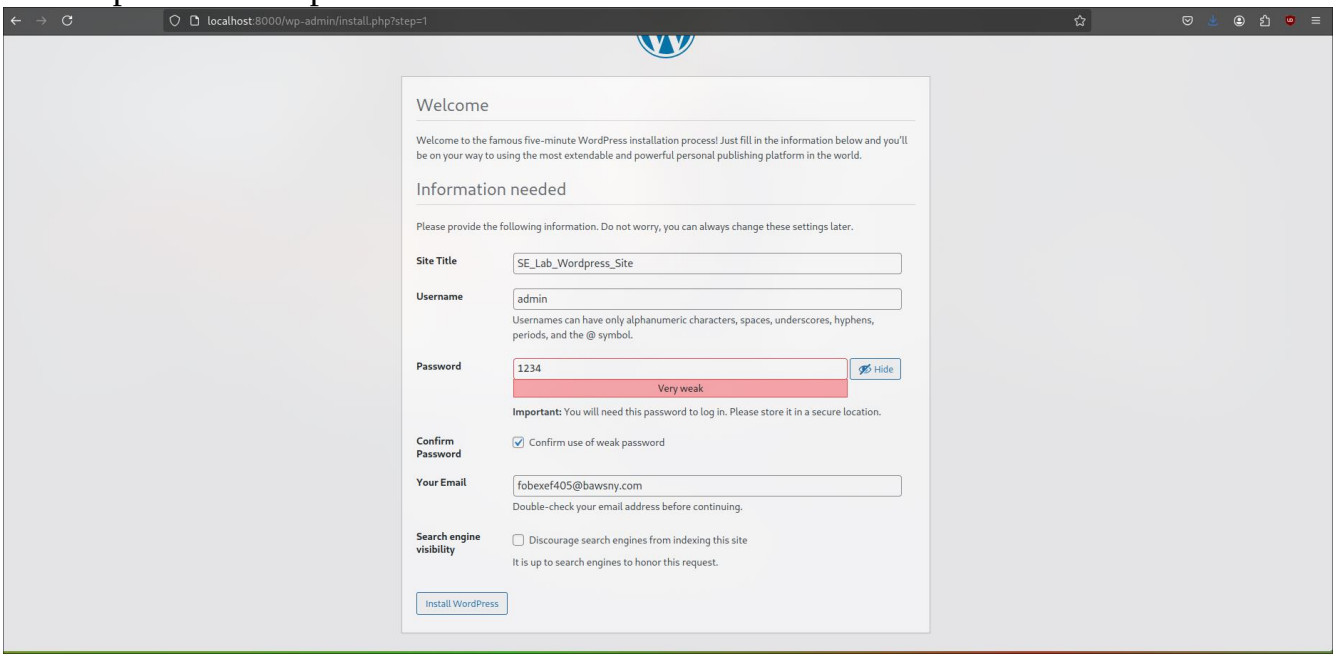

5. It means everything is working



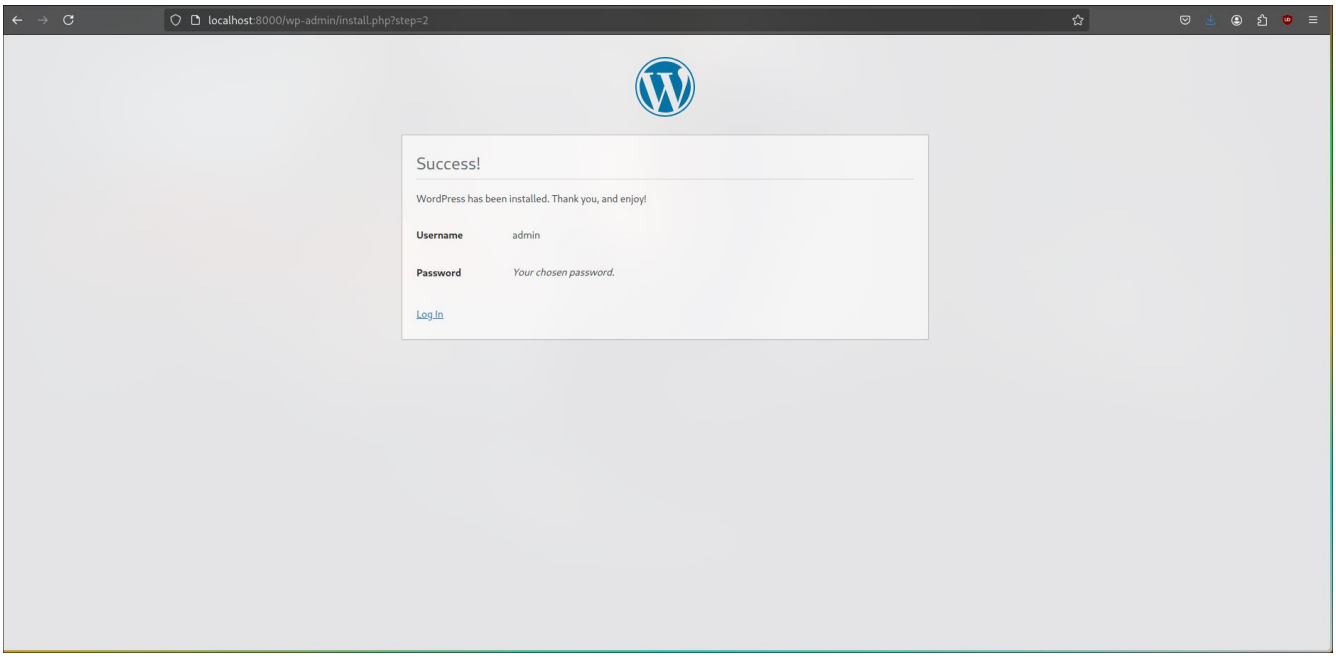
6. Go to localhost:8000



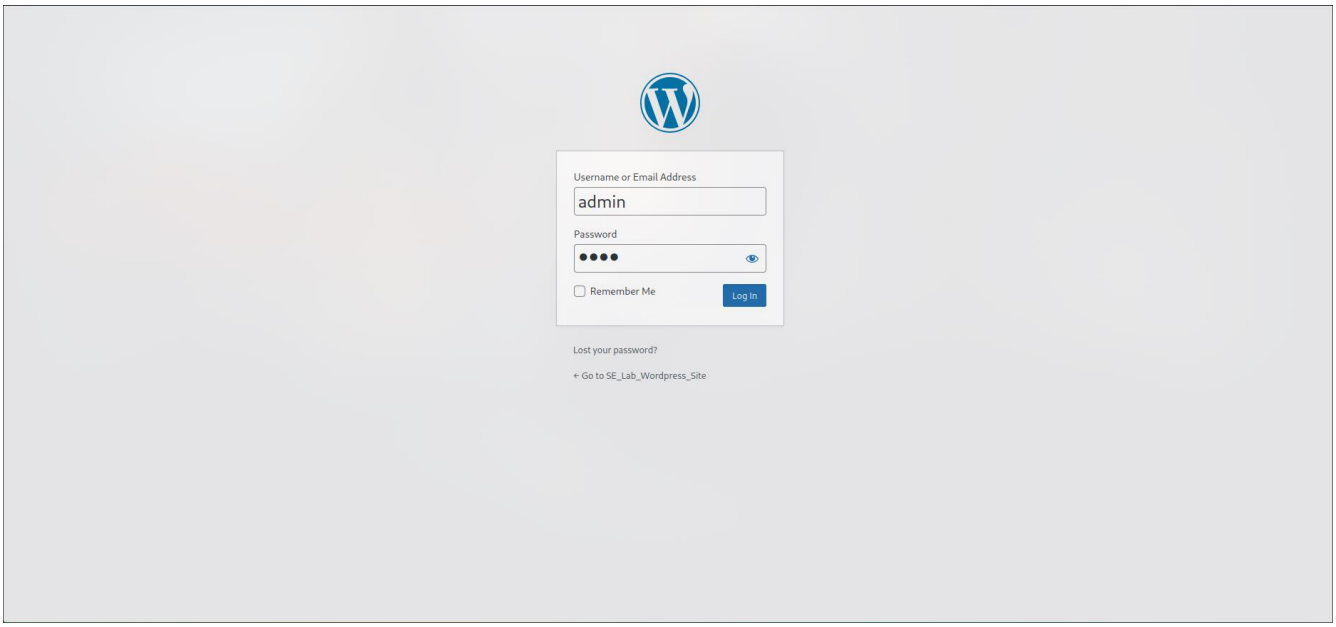

7. Complete the setup



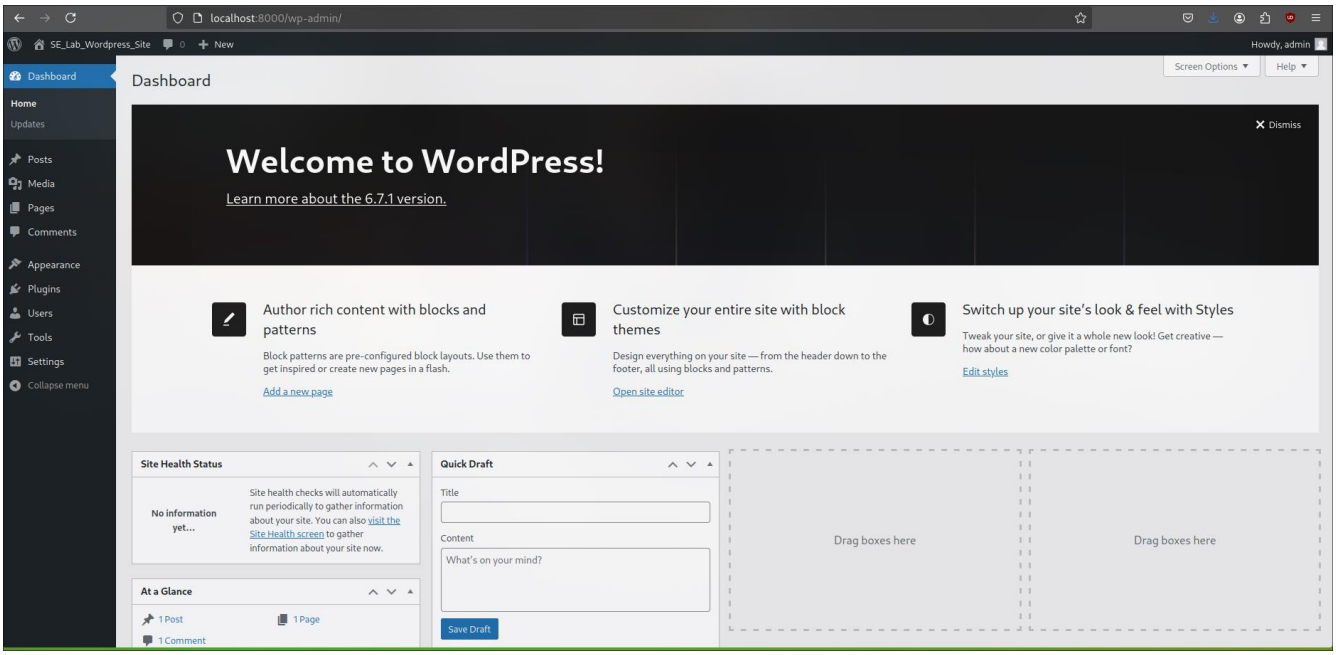
8. Setup complete



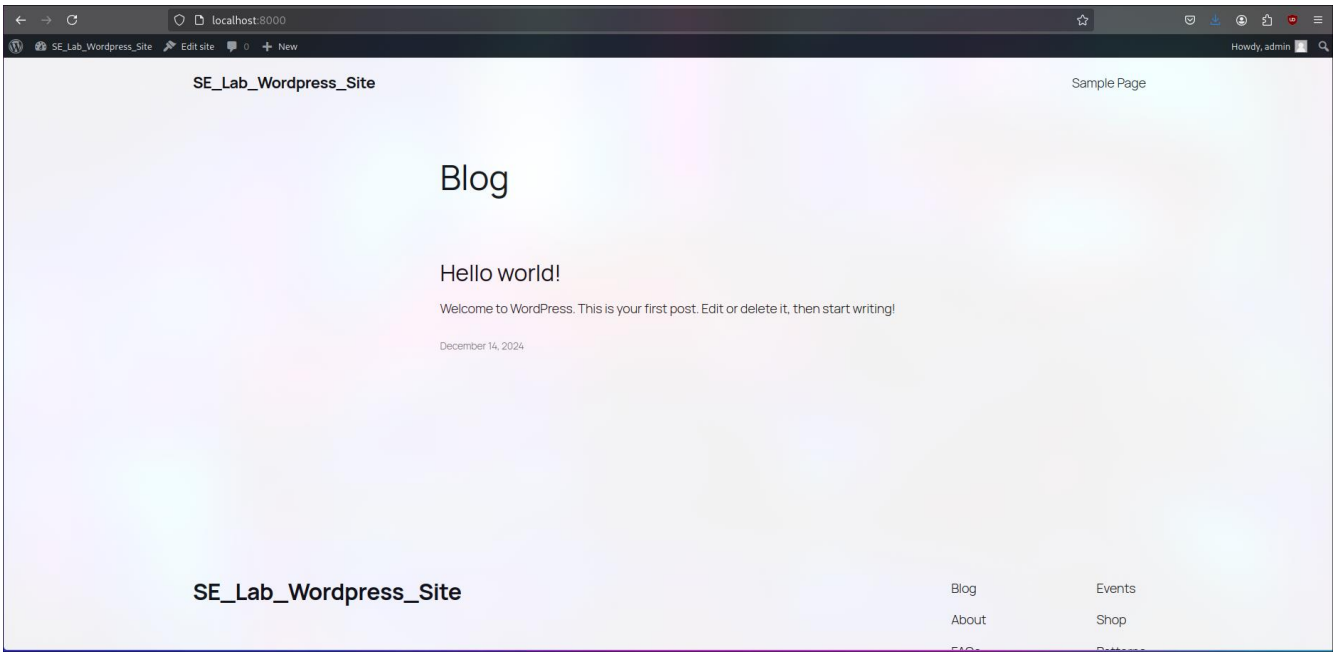

9. Login to wordpress



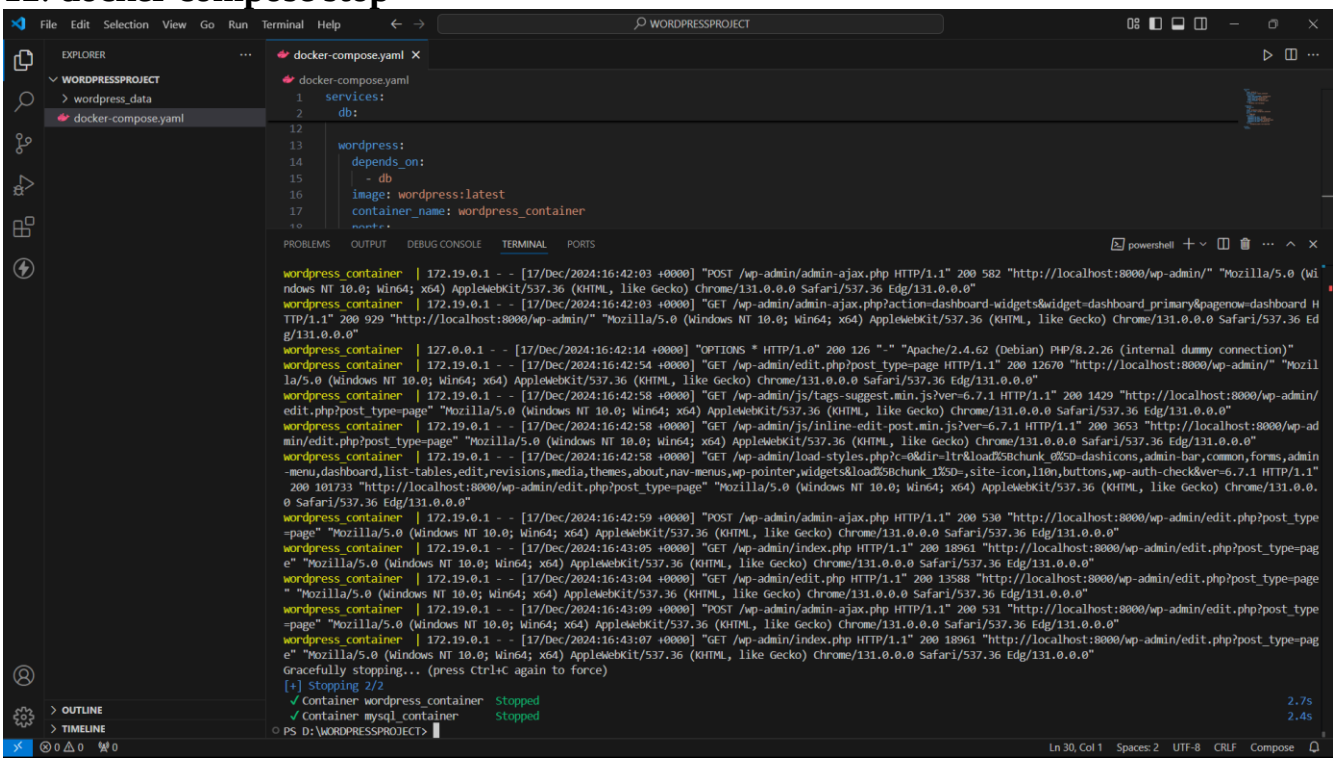
10. After login



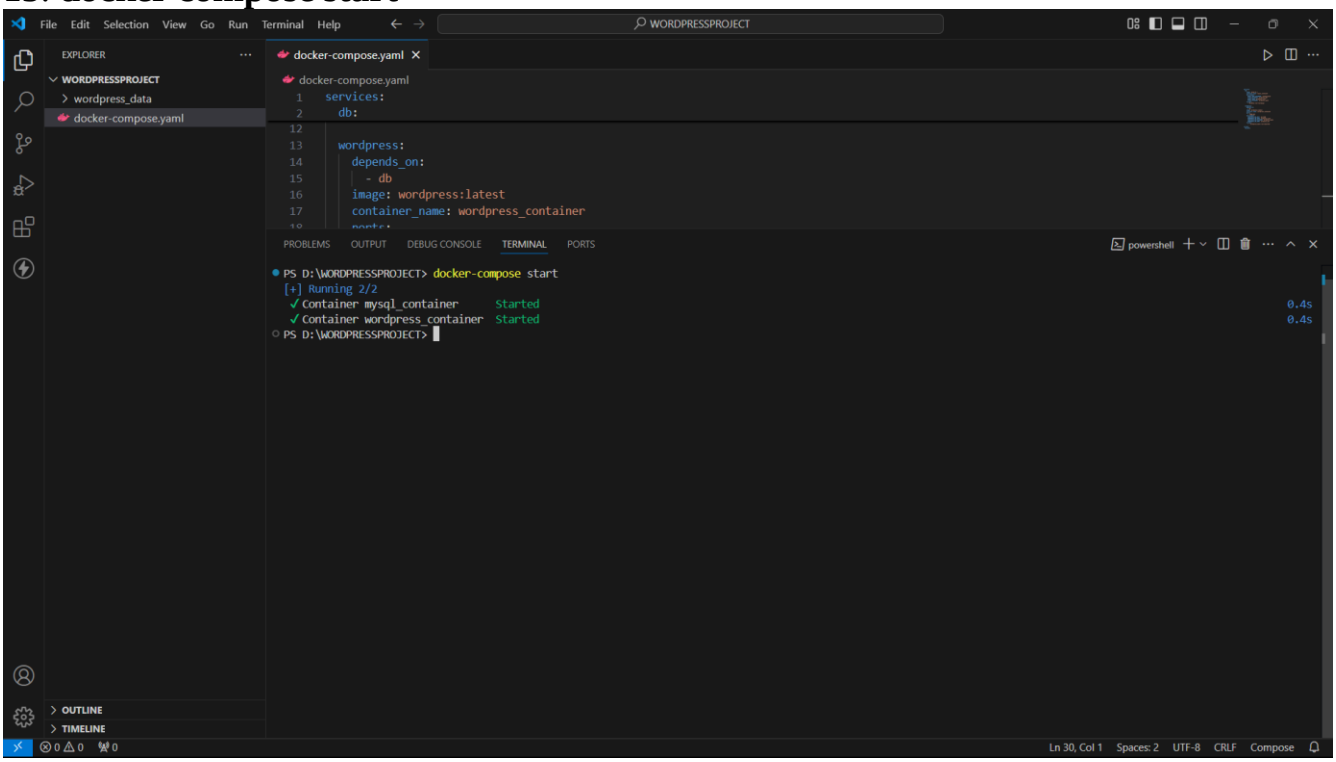

11. Page in Browser



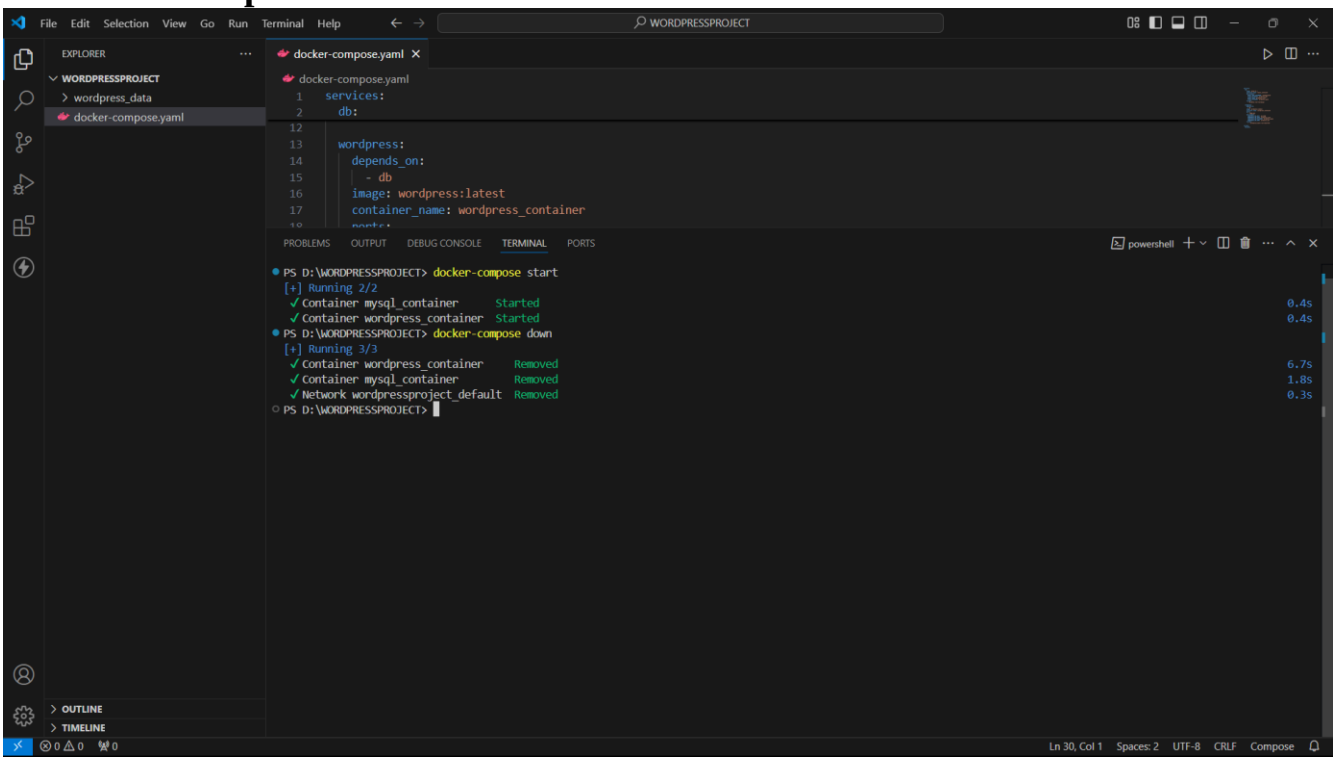
## 12. **docker compose stop**




## 13. **docker compose start**

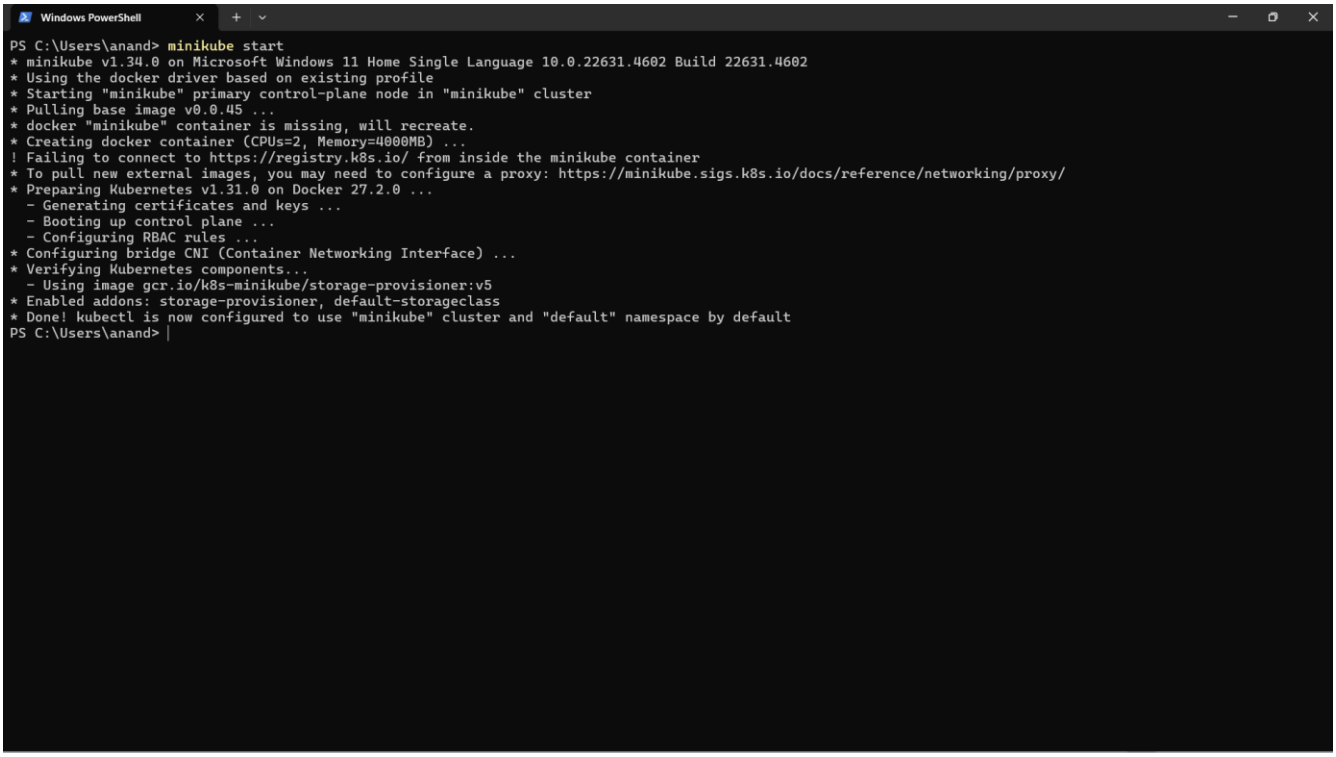


14. **docker compose down**

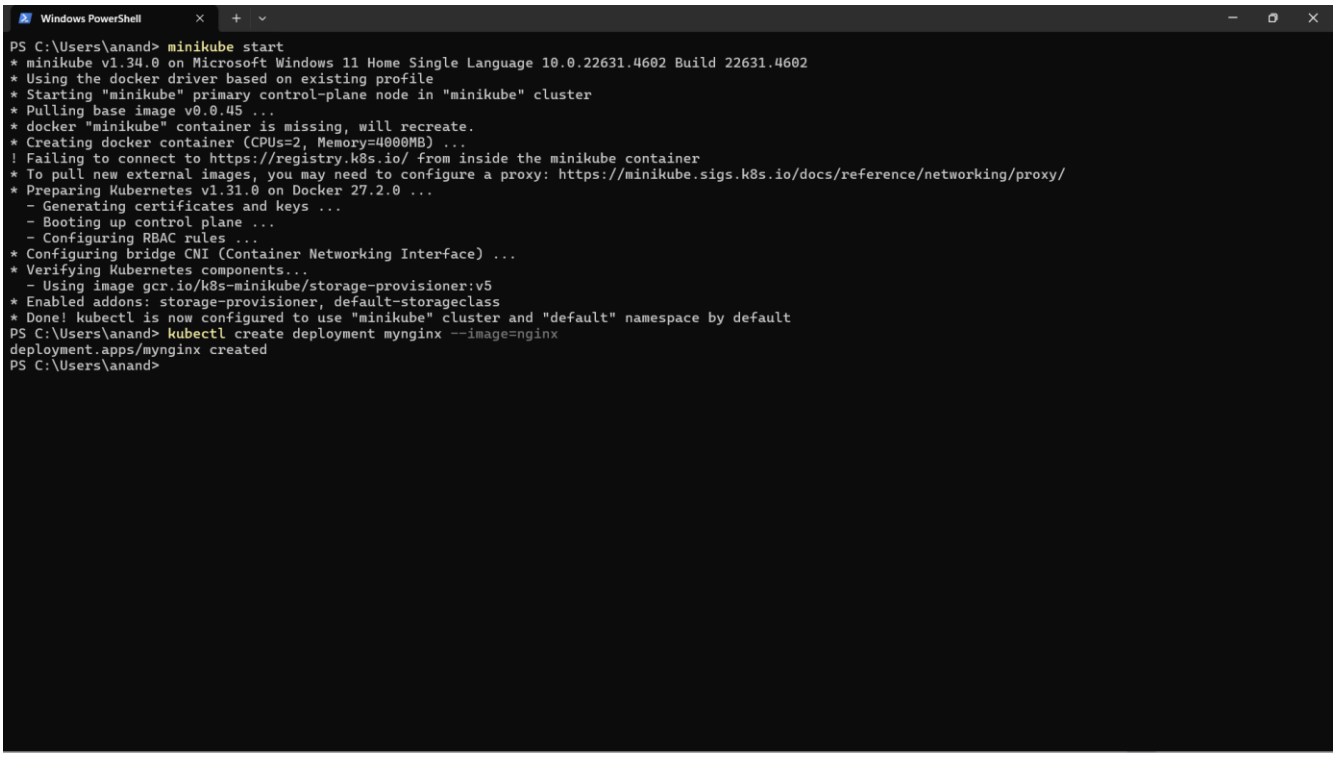



# 7E – Deploying and Scaling using Minikube

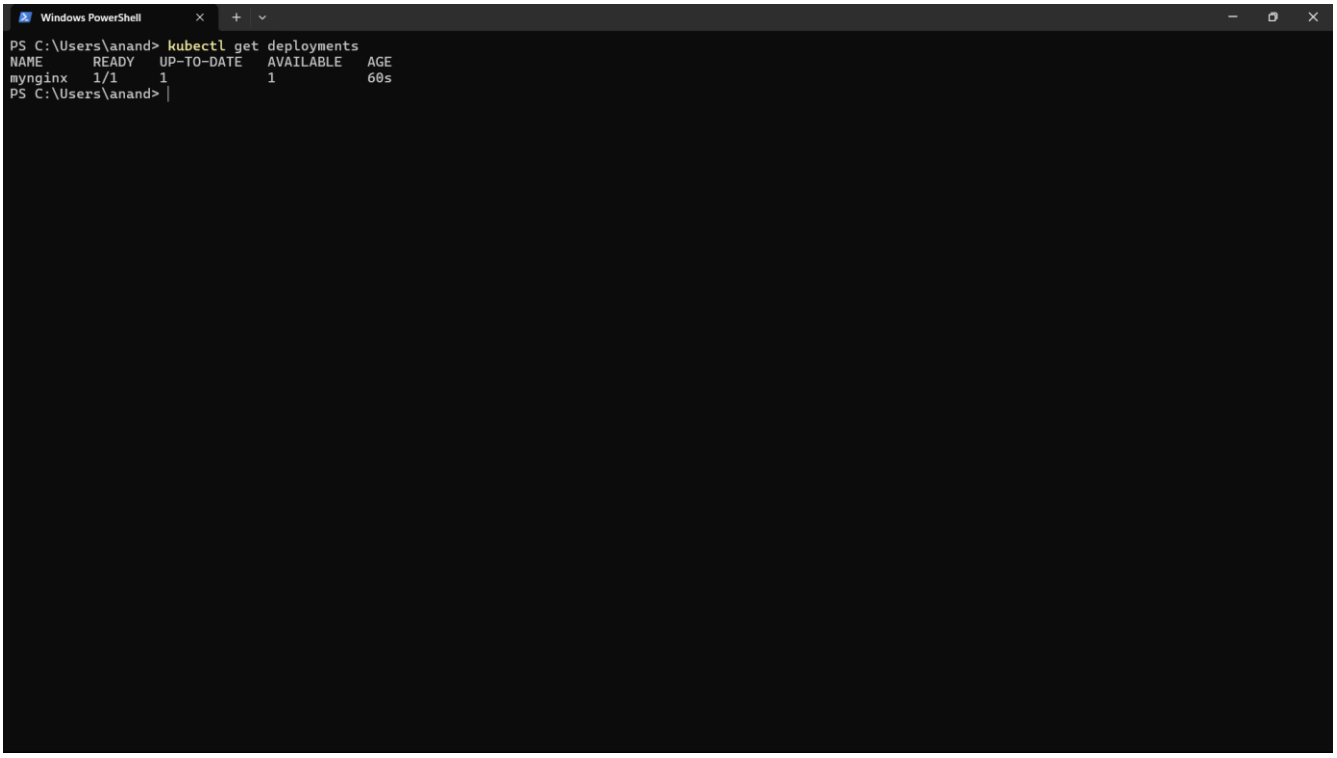
1. **minikube start**



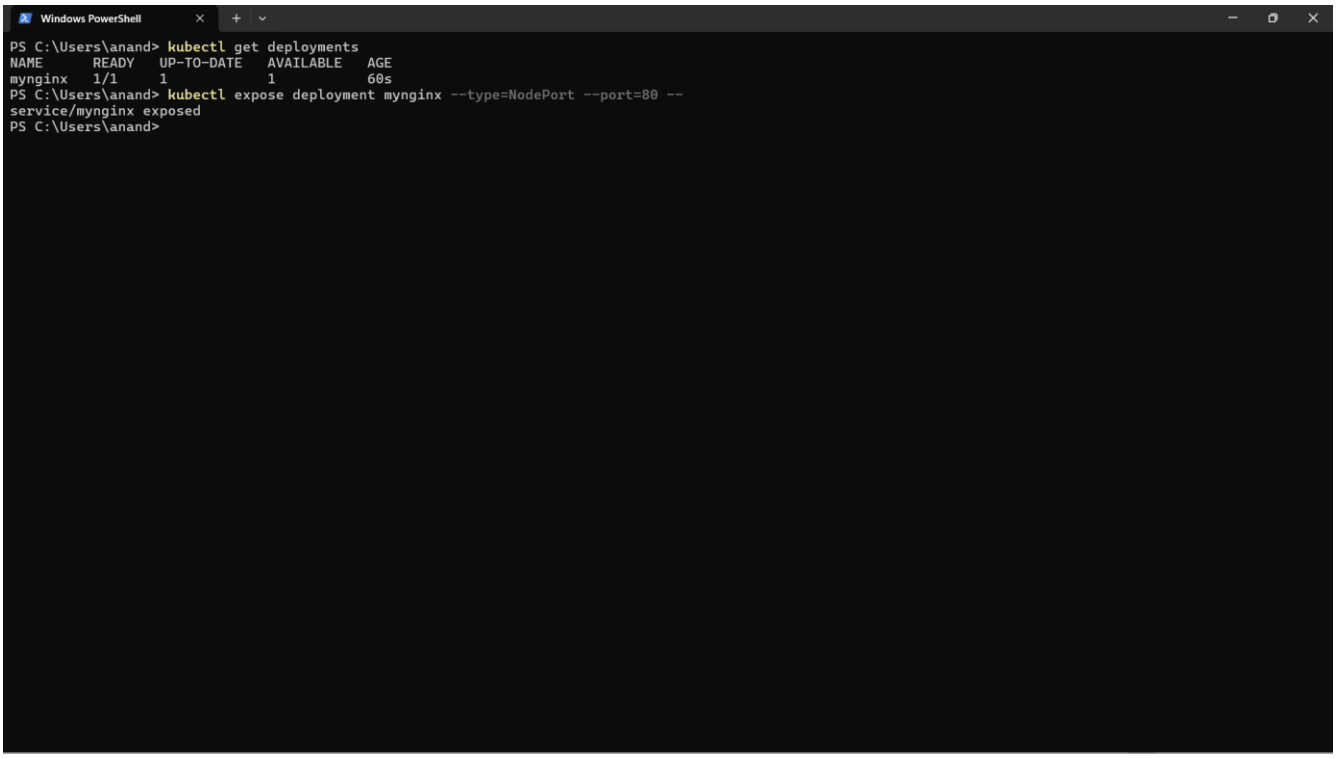

2. **minikube kubectl -- create deployment mynginx --image=nginx**



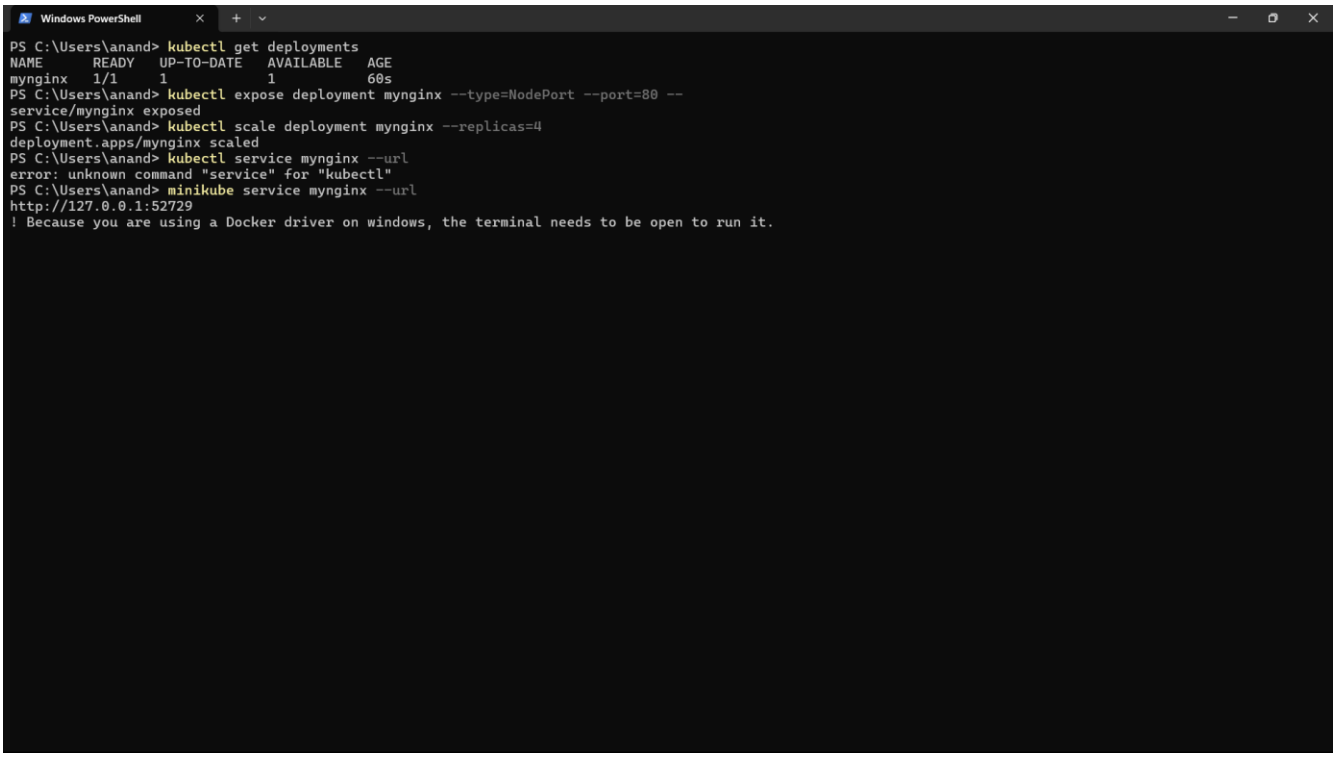

3. **minikube kubectl –- get deployments**




4. **kubectl expose deployment mynginx --type=NodePort --port=80 --target-port=80**

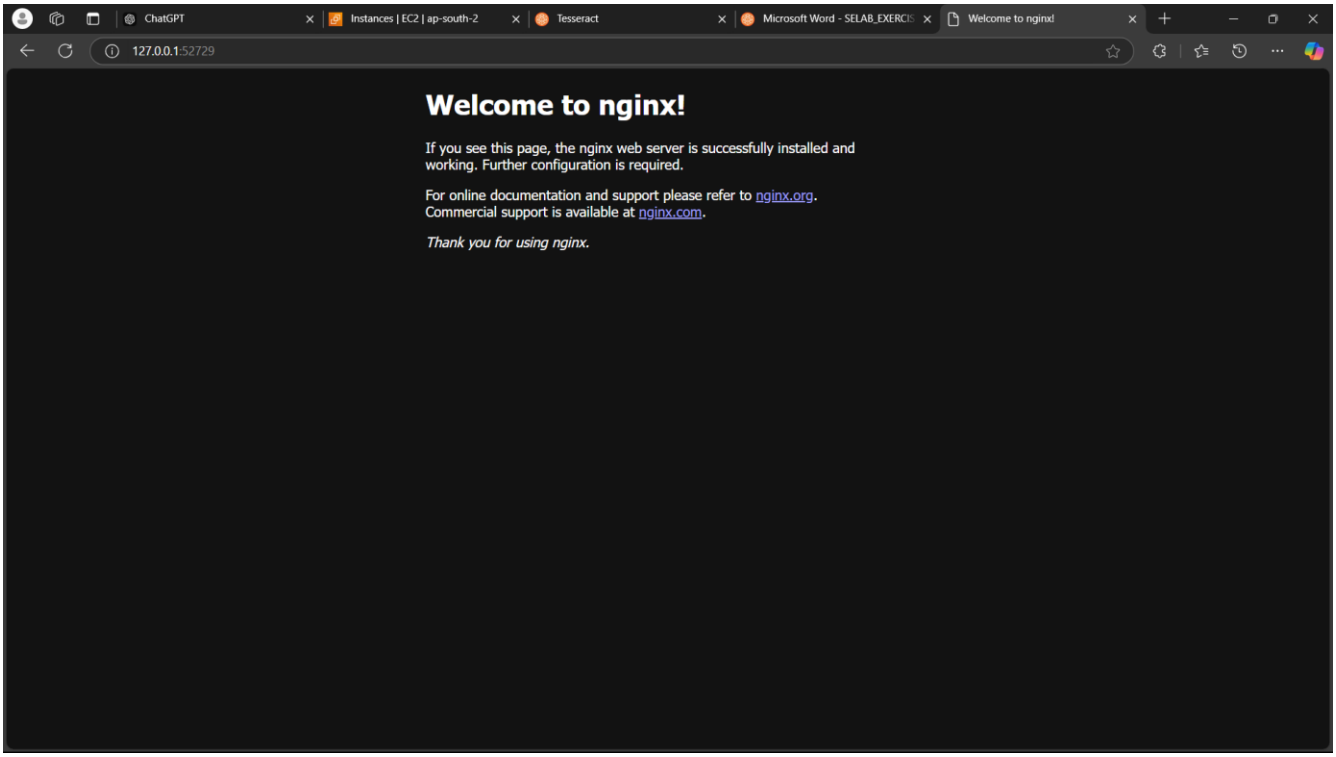


5. **minukube kubectl -- scale deployment mynginx --replicas=4**

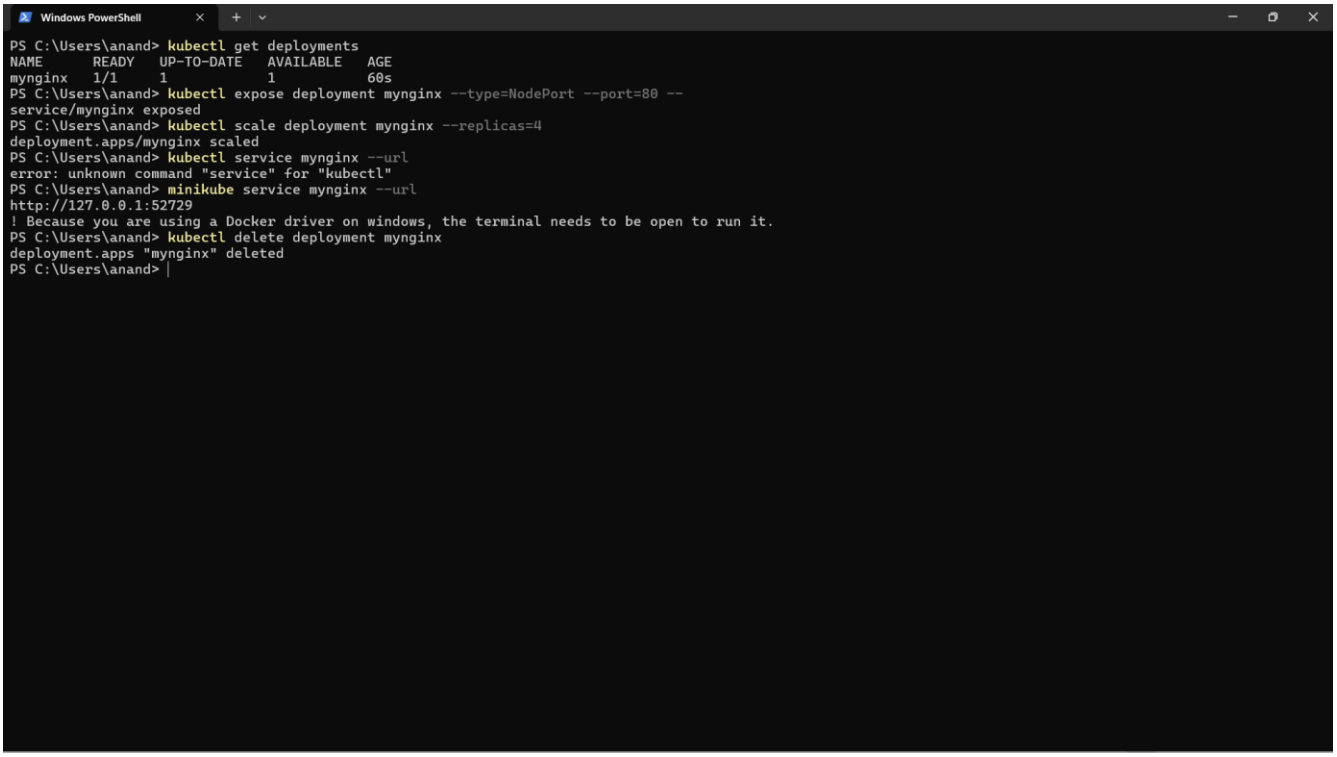



6. **minikube kubectl -- port-forward svc/mynginx 8081:80**

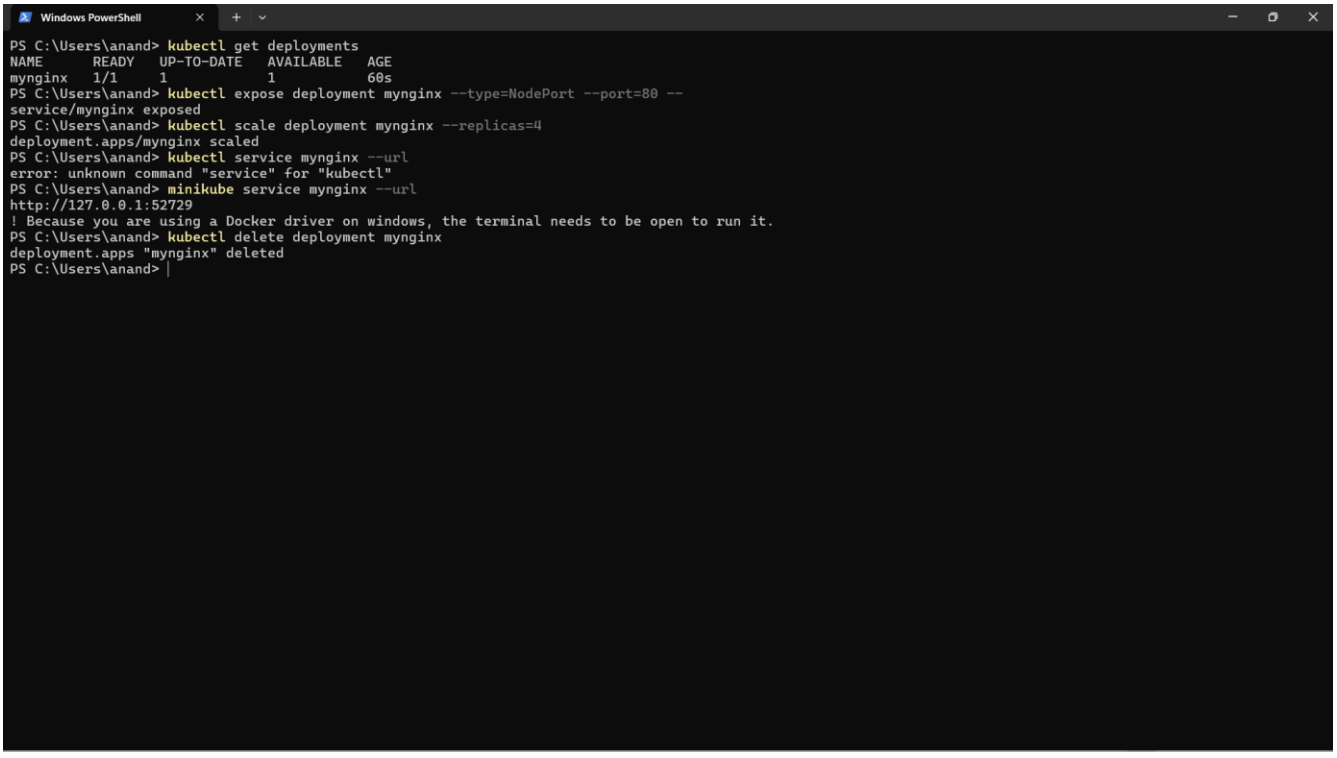

7. Go to localhost:8081 in browser




8. **minikube kubectl -- delete deployment mynginx**

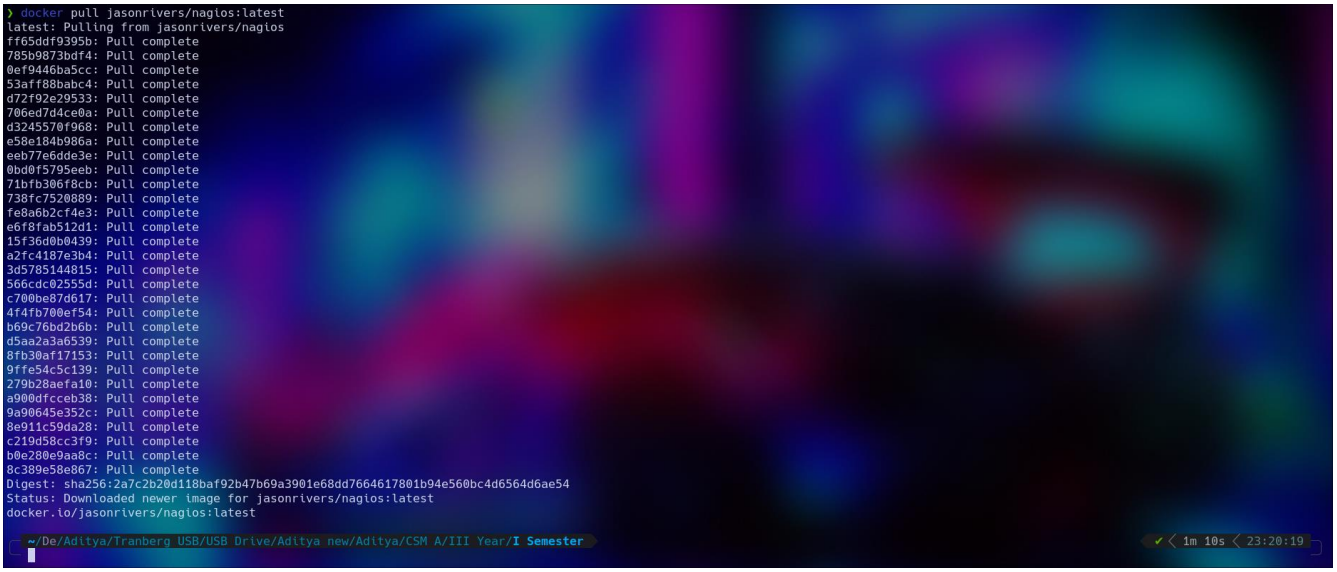



## 9. **minikube kubectl -- delete service mynginx**

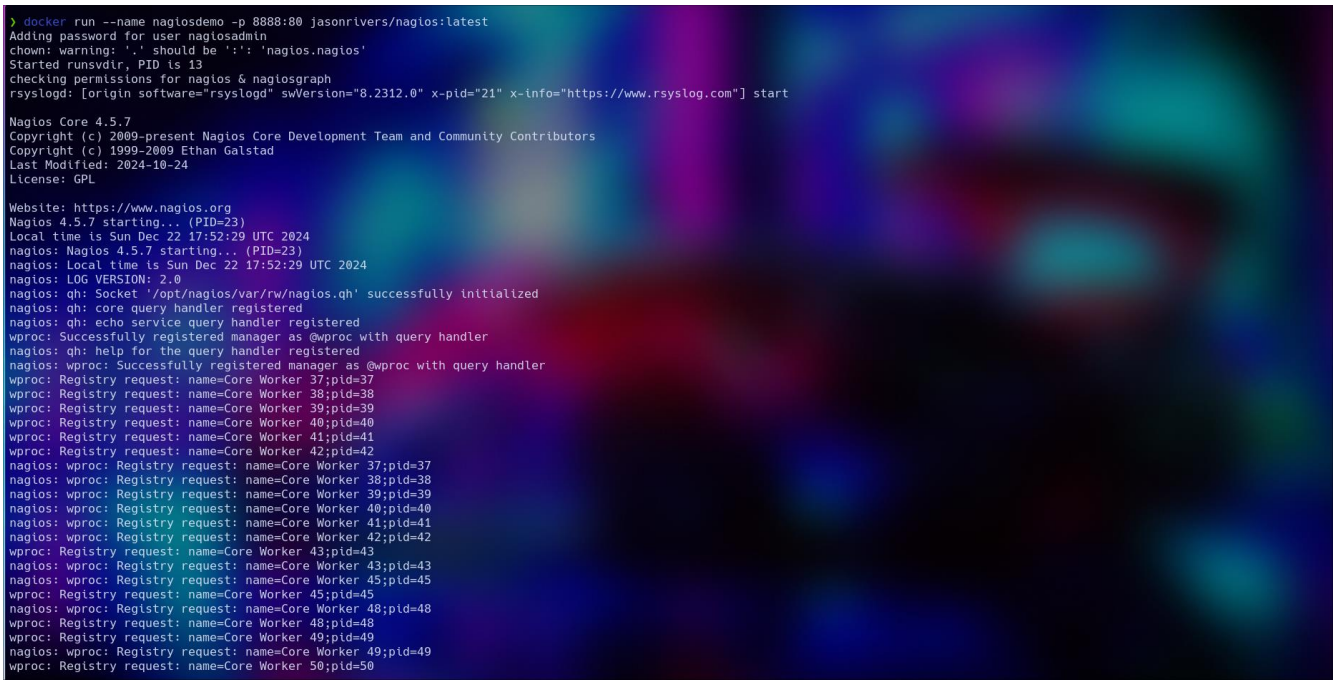


# 7F – Continuous Monitoring using Nagios

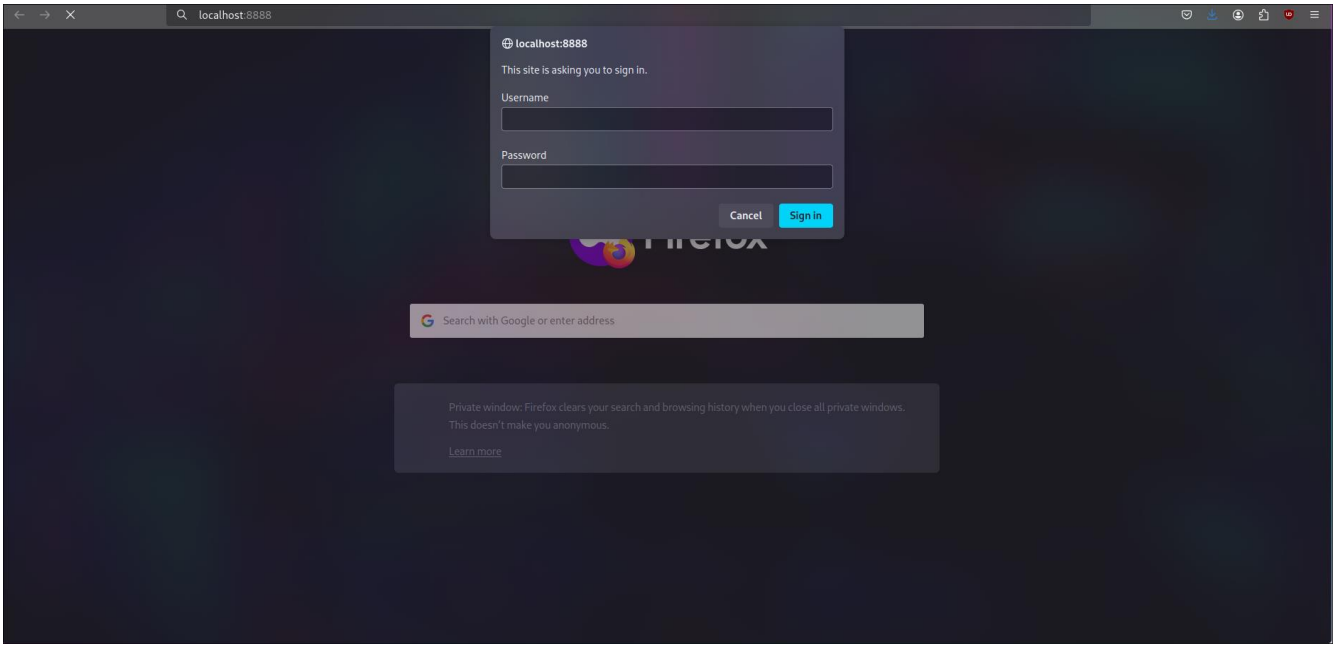
1. docker pull jasonrivers/nagios:latest



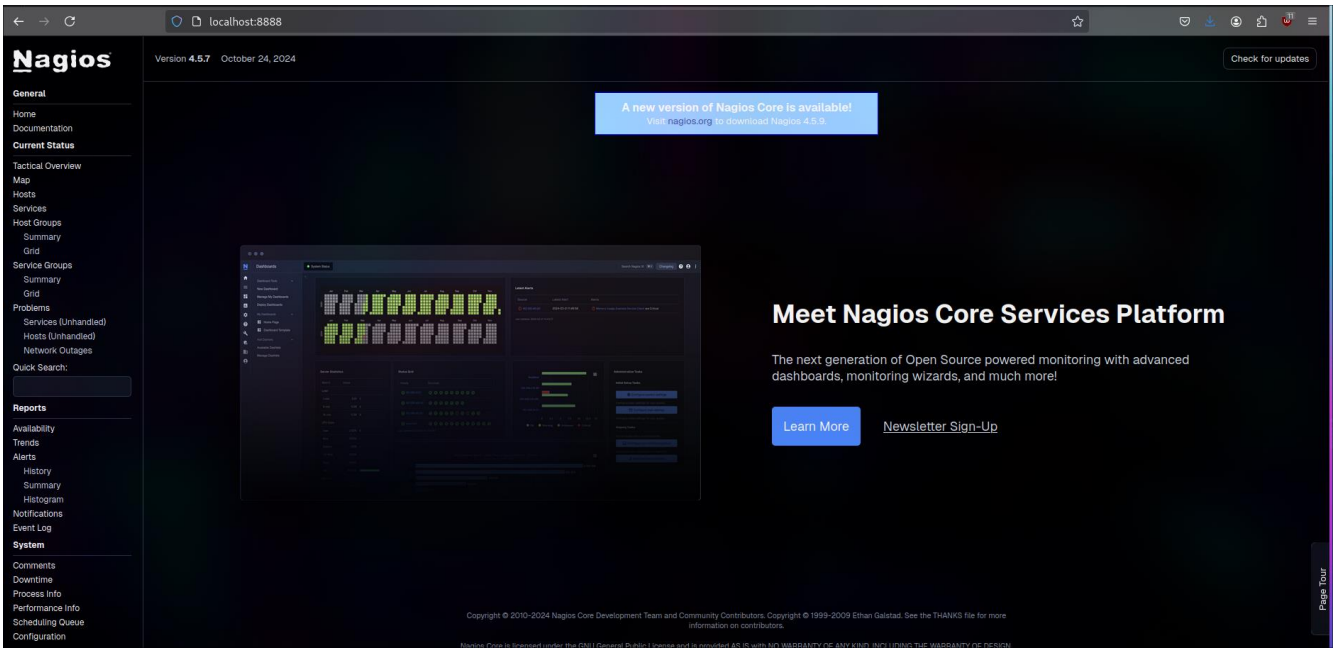

2. docker run --name nagiosdemo -p 8888:80 jasonrivers/nagios:latest



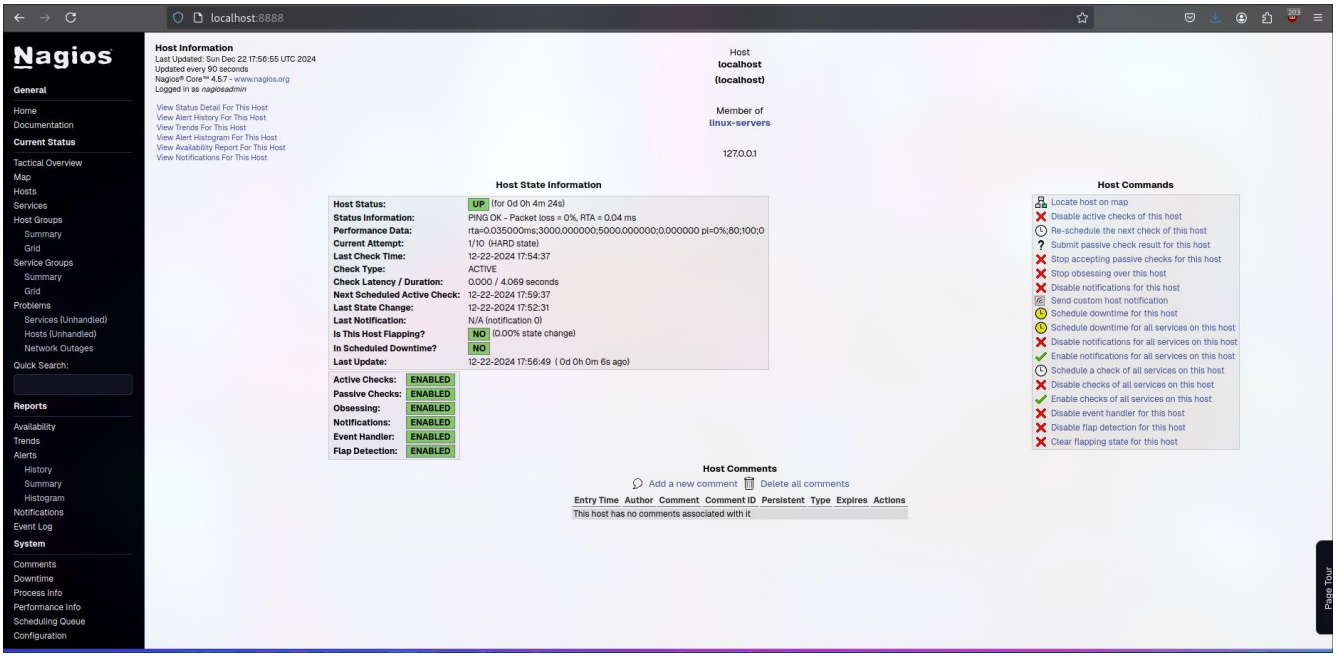

3. Go to localhost:8888




4. Nagios dashboard




5. Monitoring the host




## 6. Stopping the container


## 7. Deleting the container


## Result: Local Deployment of Project using Docker, Kubernetes and Monitoring using Nagios tool were demonstrated

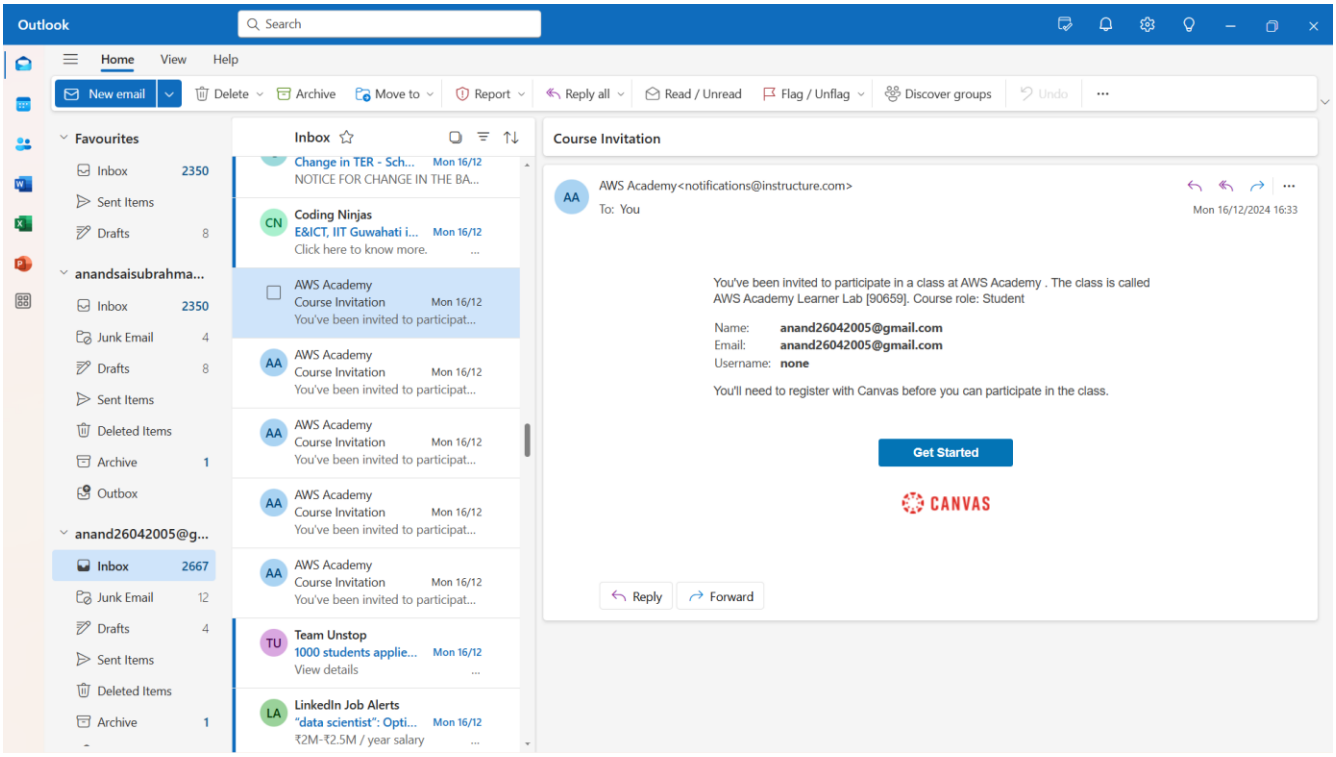
# Experiment-8: Deployment of a project into AWS Cloud using EC2 Instance

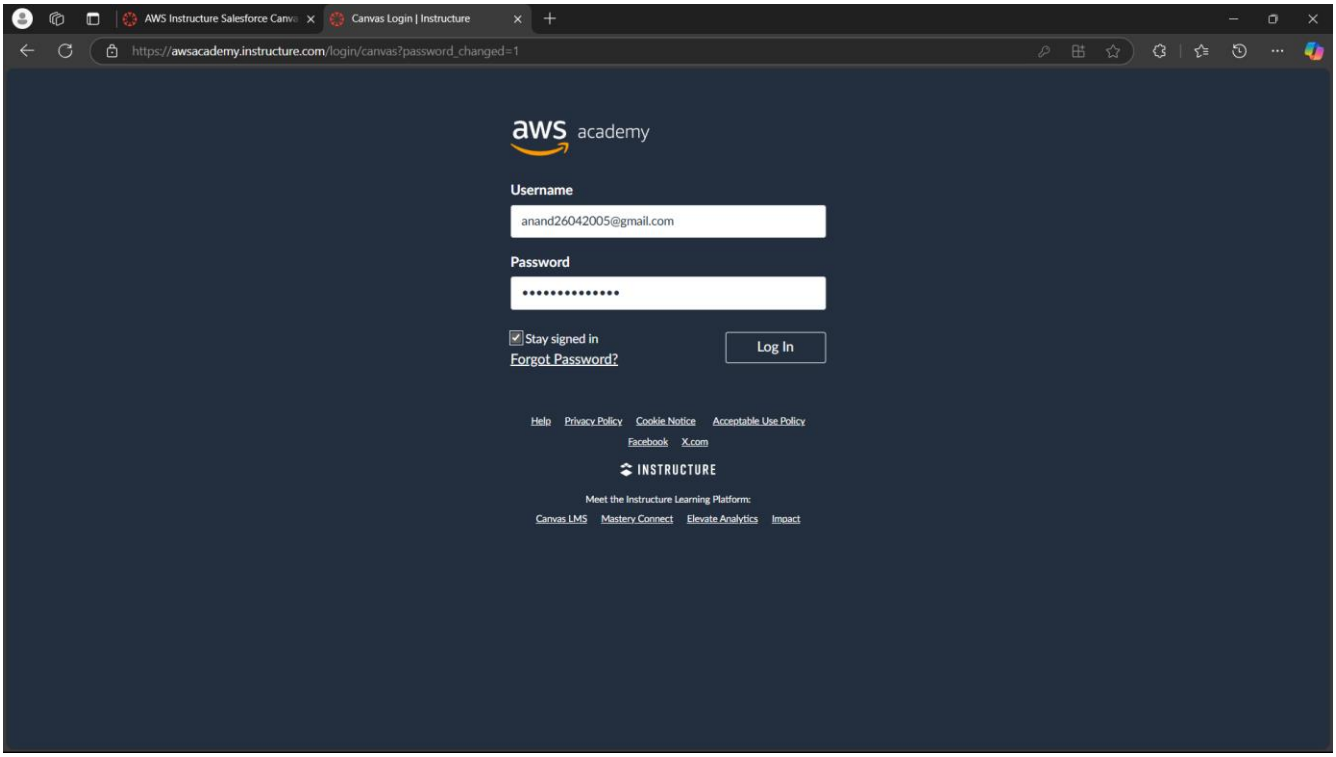
Aim: To create an AWS Learning Account, deploy a sample HTML file on AWS EC2, and deploy the Maven web application created in a previous experiment onto AWS EC2.

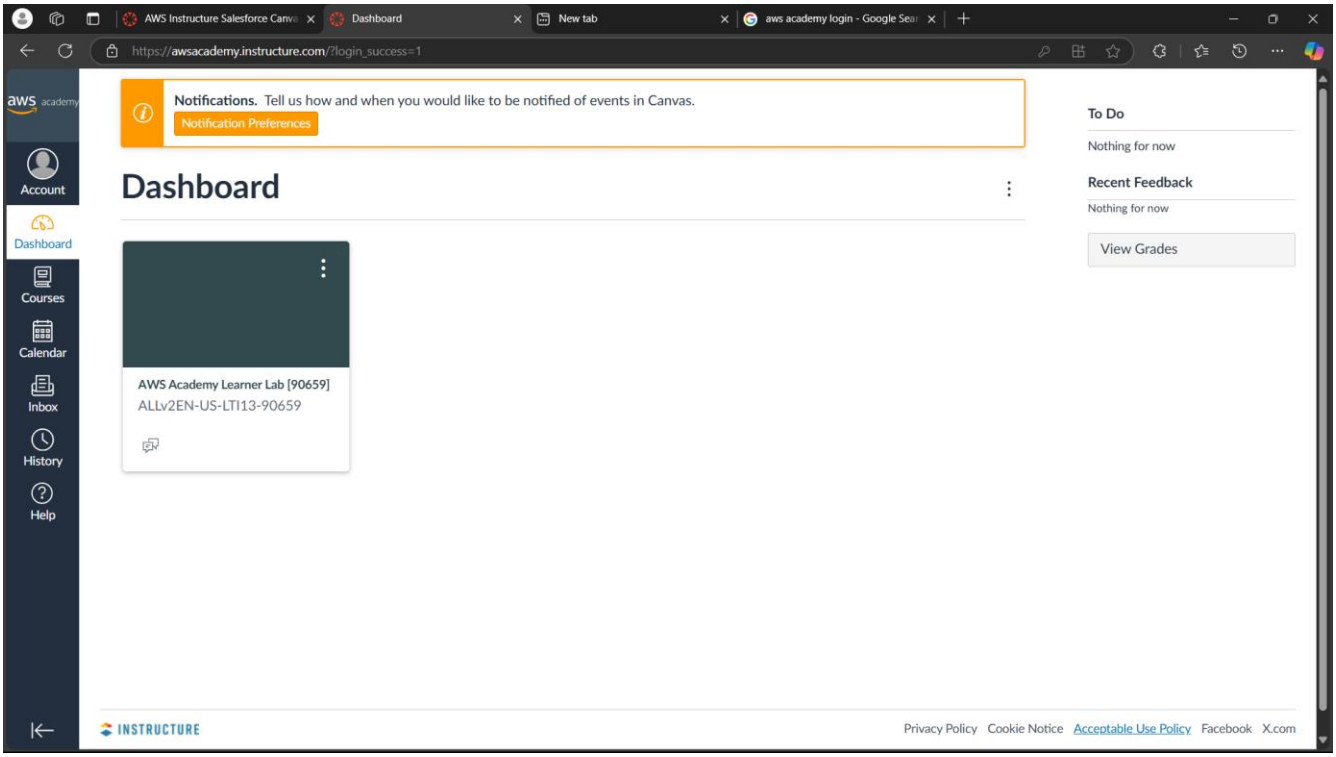
Introduction:

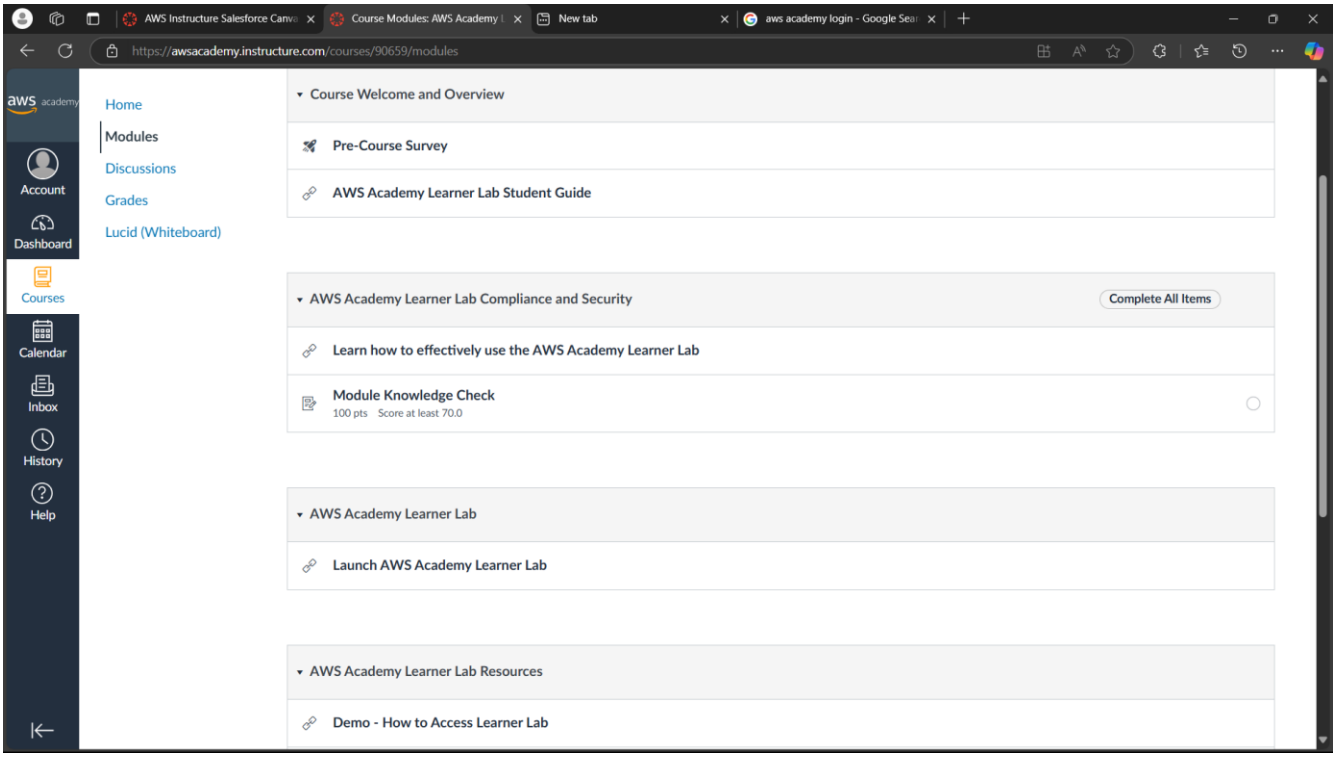
1. AWS (Amazon Web Services): A leading cloud computing platform offering scalable and reliable infrastructure for deploying applications and services.
    - AWS Learning Account: Provides access to AWS services for educational purposes, enabling hands-on experience with cloud computing.
2. AWS EC2 (Elastic Compute Cloud): A service that provides virtual servers (instances) for running applications in the cloud.
    - Instances can host web servers, databases, or any software, offering scalability and cost-efficiency.

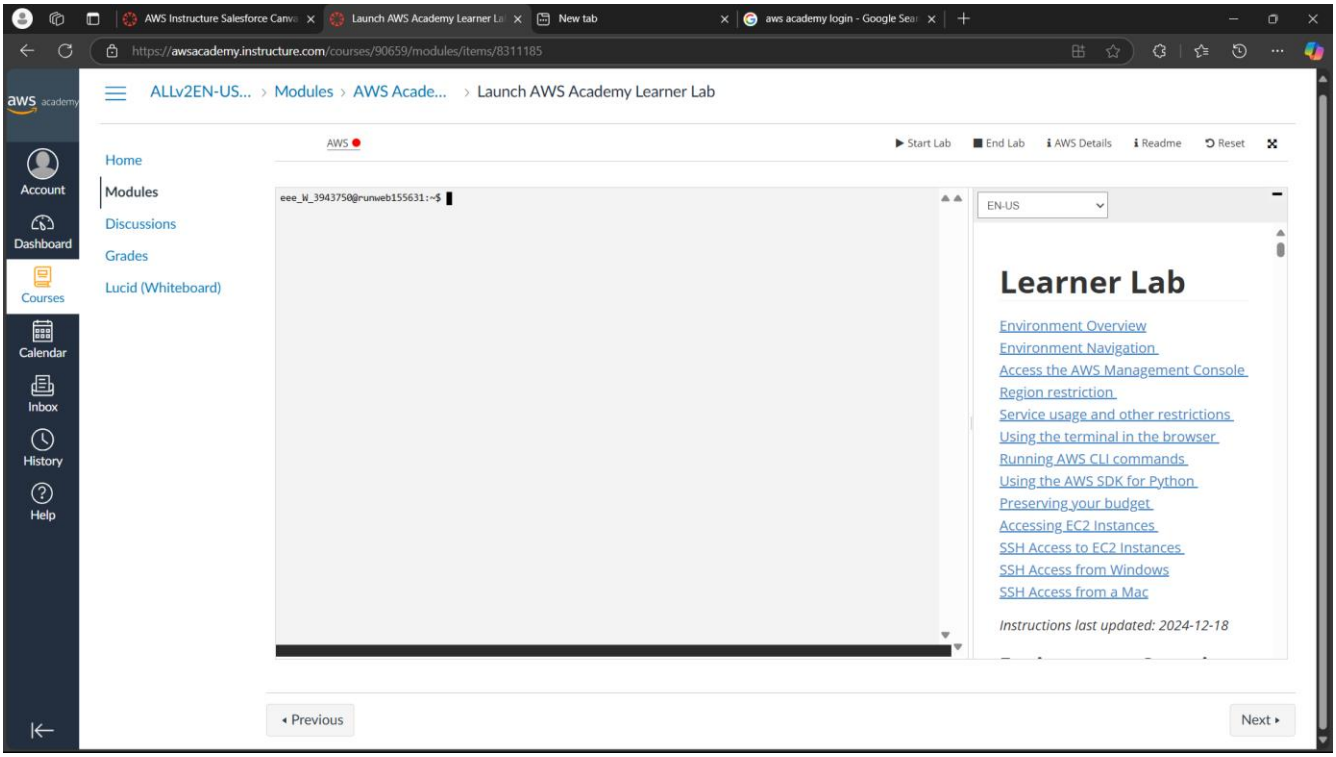
# AWS – Account Creation

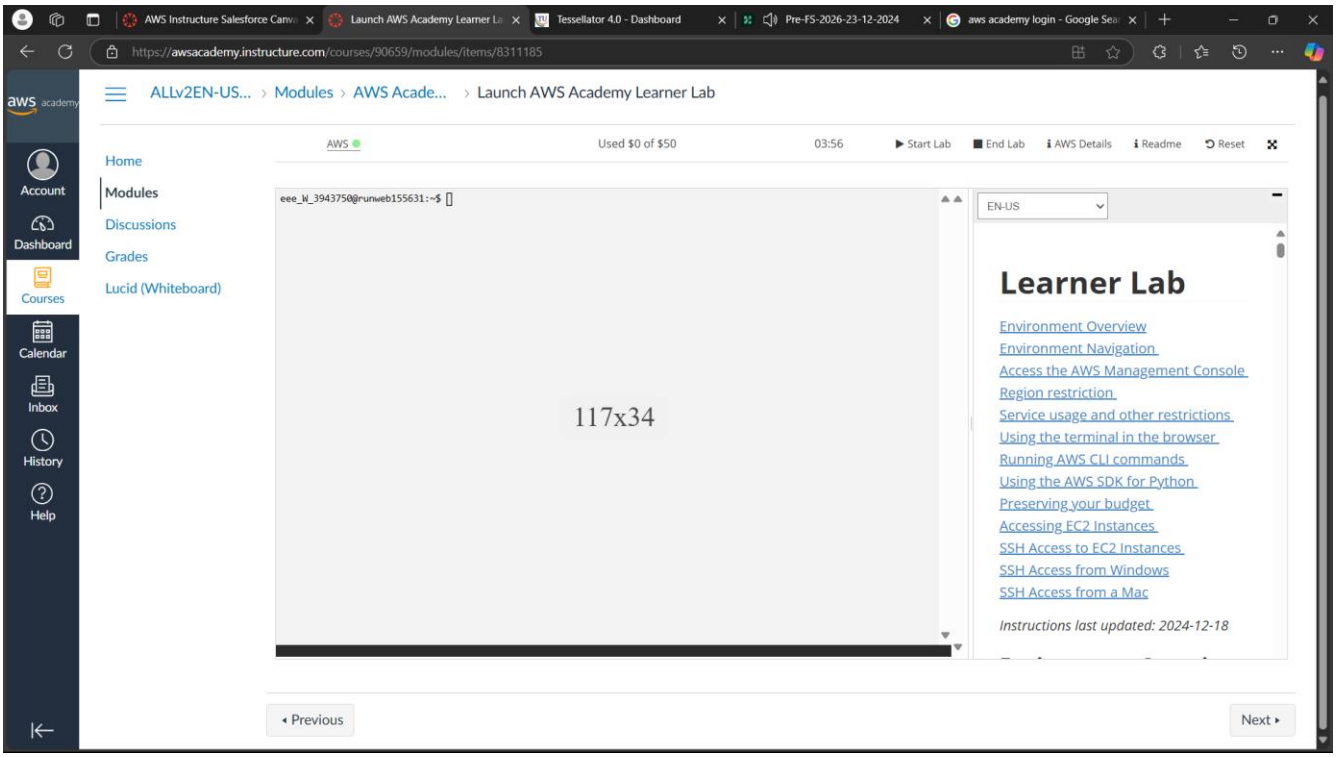

Outlook
Search
Home
View
Help
New email
Delete
Archive
Move to
Report
Reply all
Read / Unread
Flag / Unflag
Discover groups
Undo
Favourites
Inbox 2350
Sent Items
Drafts 8
anandsaisubrahma...
Inbox 2350
Junk Email 4
Drafts 8
Sent Items
Deleted Items
Archive 1
Outbox
anand26042005@g...
Inbox 2667
Junk Email 12
Drafts 4
Sent Items
Deleted Items
Archive 1
Inbox
Change in TER - Sch... Mon 16/12
NOTICE FOR CHANGE IN THE BA...
Coding Ninjas
E&ICT, IIT Guwahati i... Mon 16/12
Click here to know more.
AWS Academy
Course Invitation Mon 16/12
You've been invited to participat...
AWS Academy
Course Invitation Mon 16/12
You've been invited to participat...
AWS Academy
Course Invitation Mon 16/12
You've been invited to participat...
AWS Academy
Course Invitation Mon 16/12
You've been invited to participat...
AWS Academy
Course Invitation Mon 16/12
You've been invited to participat...
Team Unstop
1000 students applie... Mon 16/12
View details
LinkedIn Job Alerts
"data scientist": Opti... Mon 16/12
₹2M-₹2.5M / year salary
Course Invitation
AWS Academy<notifications@instructure.com>
To: You
Mon 16/12/2024 16:33
You've been invited to participate in a class at AWS Academy . The class is called AWS Academy Learner Lab [90659]. Course role: Student
Name: anand26042005@gmail.com
Email: anand26042005@gmail.com
Username: none
You'll need to register with Canvas before you can participate in the class.
Get Started
CANVAS
Reply
Forward

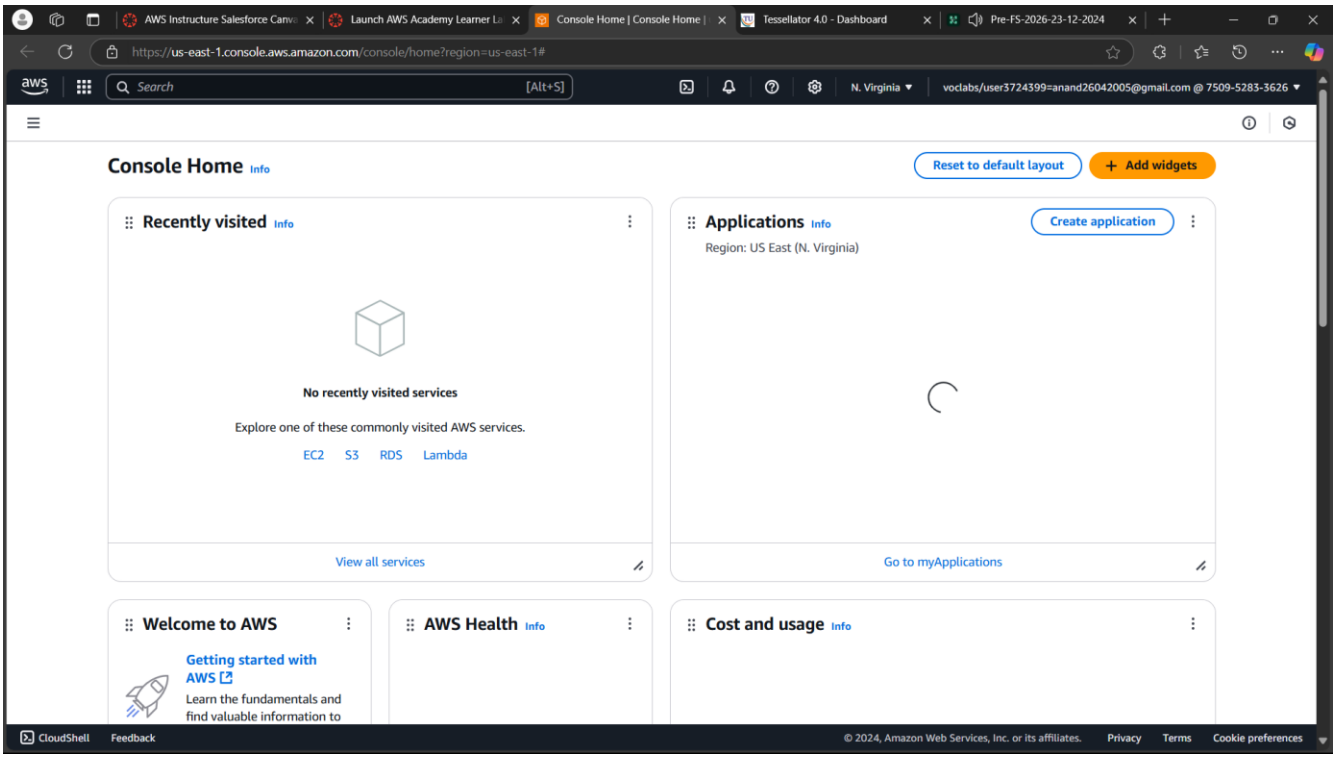

AWS Instructure Salesforce Canva
Canvas Login | Instructure
https://awsacademy.instructure.com/login/canvas?password_changed=1
aws academy
Username
anand26042005@gmail.com
Password
Stay signed in
Forgot Password?
Log In
Help
Privacy Policy
Cookie Notice
Acceptable Use Policy
Facebook
X.com
INSTRUCTURE
Meet the Instructure Learning Platform:
Canvas LMS
Mastery Connect
Elevate Analytics
Impact

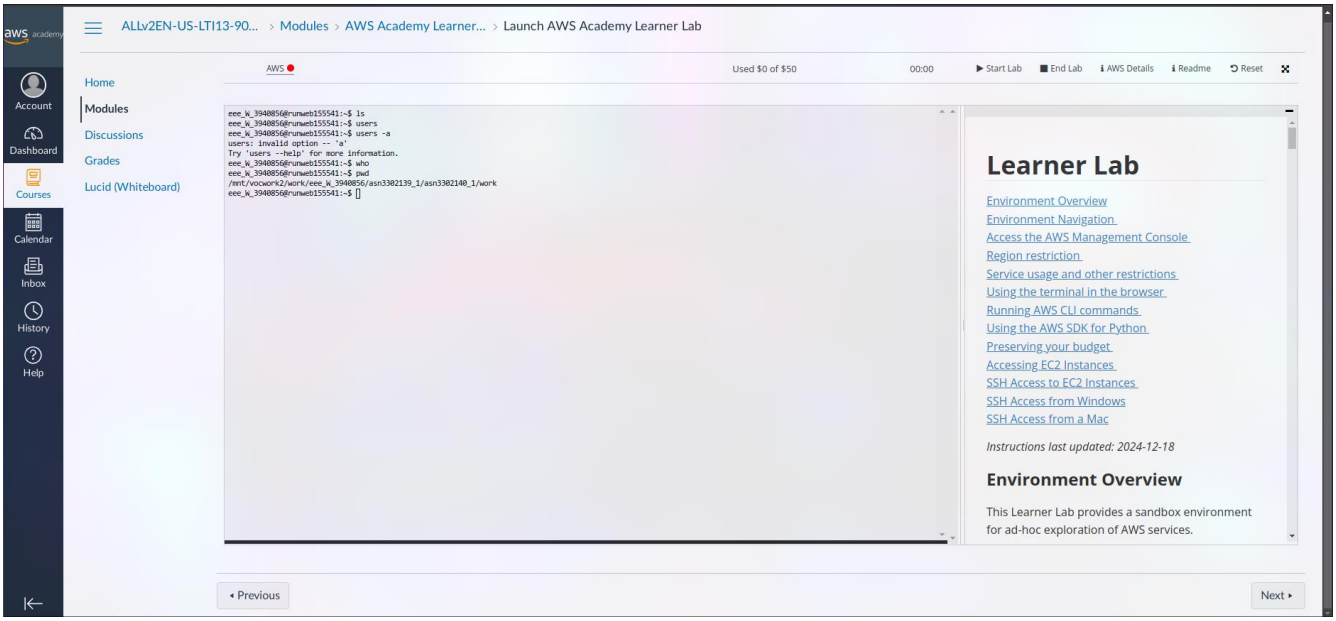

AWS Instructure Salesforce Canva
Dashboard
New tab
aws academy login - Google Sear
https://awsacademy.instructure.com/?login_success=1
aws academy
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Notifications. Tell us how and when you would like to be notified of events in Canvas.
Notification Preferences
Dashboard
AWS Academy Learner Lab [90659]
ALLv2EN-US-LTI13-90659
To Do
Nothing for now
Recent Feedback
Nothing for now
View Grades
INSTRUCTURE
Privacy Policy
Cookie Notice
Acceptable Use Policy
Facebook
X.com


AWS Instructure Salesforce Canva
Course Modules: AWS Academy L
New tab
aws academy login - Google Sear
https://awsacademy.instructure.com/courses/90659/modules
aws academy
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
Course Welcome and Overview
Pre-Course Survey
AWS Academy Learner Lab Student Guide
AWS Academy Learner Lab Compliance and Security
Complete All Items
Learn how to effectively use the AWS Academy Learner Lab
Module Knowledge Check
100 pts Score at least 70.0
AWS Academy Learner Lab
Launch AWS Academy Learner Lab
AWS Academy Learner Lab Resources
Demo - How to Access Learner Lab


AWS Instructure Salesforce Canva
Launch AWS Academy Learner La
New tab
aws academy login - Google Sear
https://awsacademy.instructure.com/courses/90659/modules/items/8311185
aws academy
ALLv2EN-US... › Modules › AWS Acade... › Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
AWS
Start Lab
End Lab
AWS Details
Readme
Reset
eee_w_3943750@runweb155631:~$
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
SSH Access from a Mac
Instructions last updated: 2024-12-18
Previous
Next

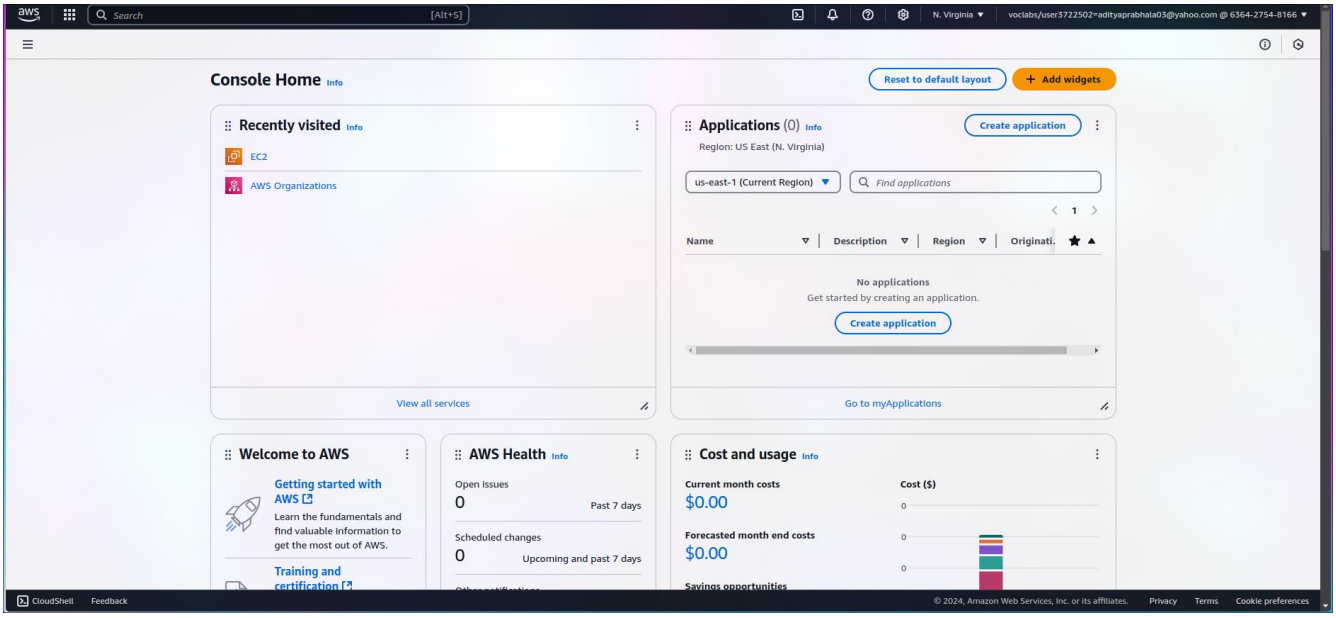

AWS Instructure Salesforce Canva
Launch AWS Academy Learner La
Tesselator 4.0 - Dashboard
Pre-FS-2026-23-12-2024
aws academy login - Google Sear
https://awsacademy.instructure.com/courses/90659/modules/items/8311185
aws academy
ALLv2EN-US... › Modules › AWS Acade... › Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
AWS
Used $0 of $50
03:56
Start Lab
End Lab
AWS Details
Readme
Reset
eee_w_3943750@runweb155631:~$
117x34
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
SSH Access from a Mac
Instructions last updated: 2024-12-18
Previous
Next


AWS Instructure Salesforce Canva
Launch AWS Academy Learner La
Console Home | Console Home
Tessellator 4.0 - Dashboard
Pre-FS-2026-23-12-2024
https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1#
Search
[Alt+S]
N. Virginia
voclabs/user3724399=anand26042005@gmail.com @ 7509-5283-3626
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
No recently visited services
Explore one of these commonly visited AWS services.
EC2 S3 RDS Lambda
View all services
Applications Info
Create application
Region: US East (N. Virginia)
Go to myApplications
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to
AWS Health Info
Cost and usage Info
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

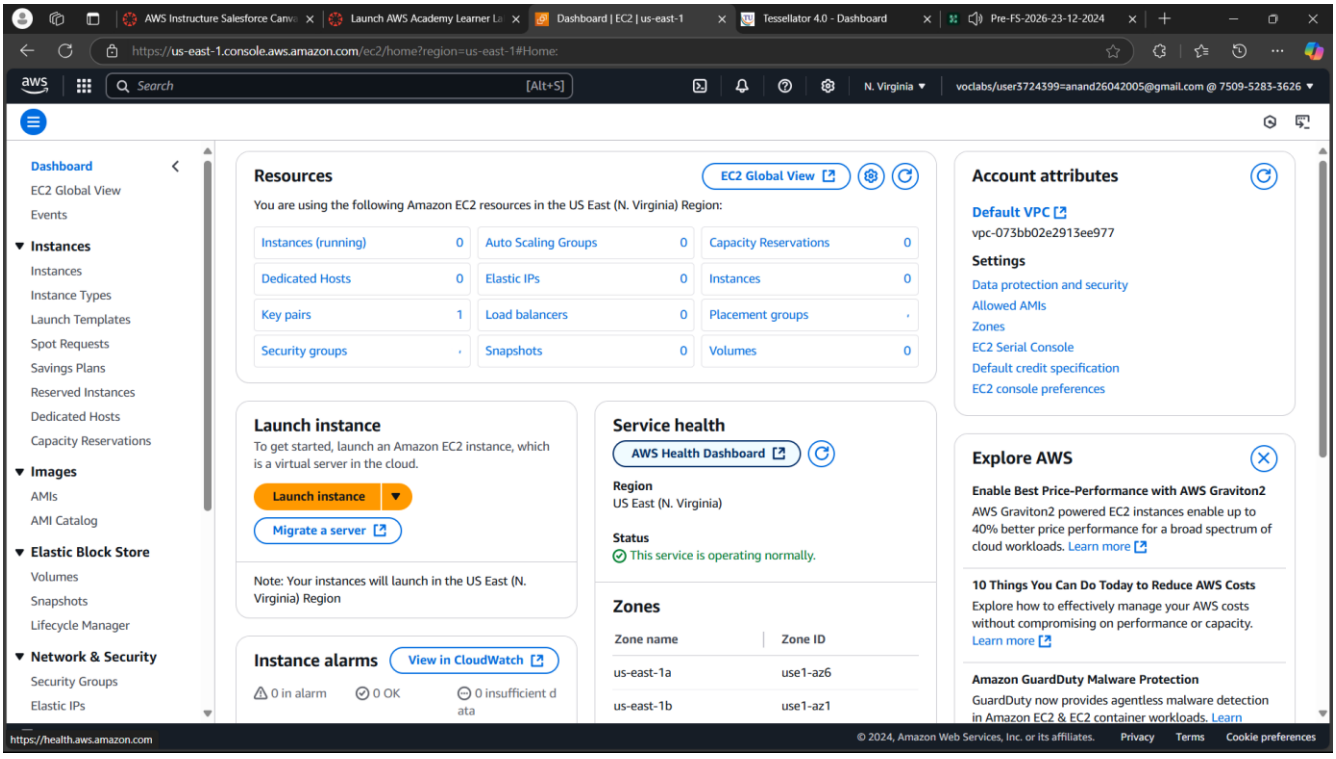

ALLv2EN-US-LTI13-90...
Modules
AWS Academy Learner...
Launch AWS Academy Learner Lab
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Modules
Discussions
Grades
Lucid (Whiteboard)
AWS
Used $0 of $50
00:00
Start Lab
End Lab
AWS Details
Readme
Reset
eee_W_3940856@runweb155541:~$ ls
eee_W_3940856@runweb155541:~$ users
eee_W_3940856@runweb155541:~$ users -a
users: invalid option -- 'a'
Try 'users --help' for more information.
eee_W_3940856@runweb155541:~$ who
eee_W_3940856@runweb155541:~$ pwd
/mnt/vocwork2/work/eee_W_3940856/asn3302139_1/asn3302140_1/work
eee_W_3940856@runweb155541:~$
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
SSH Access from a Mac
Instructions last updated: 2024-12-18
Environment Overview
This Learner Lab provides a sandbox environment for ad-hoc exploration of AWS services.
Previous
Next

# AWS – Deploying index.html

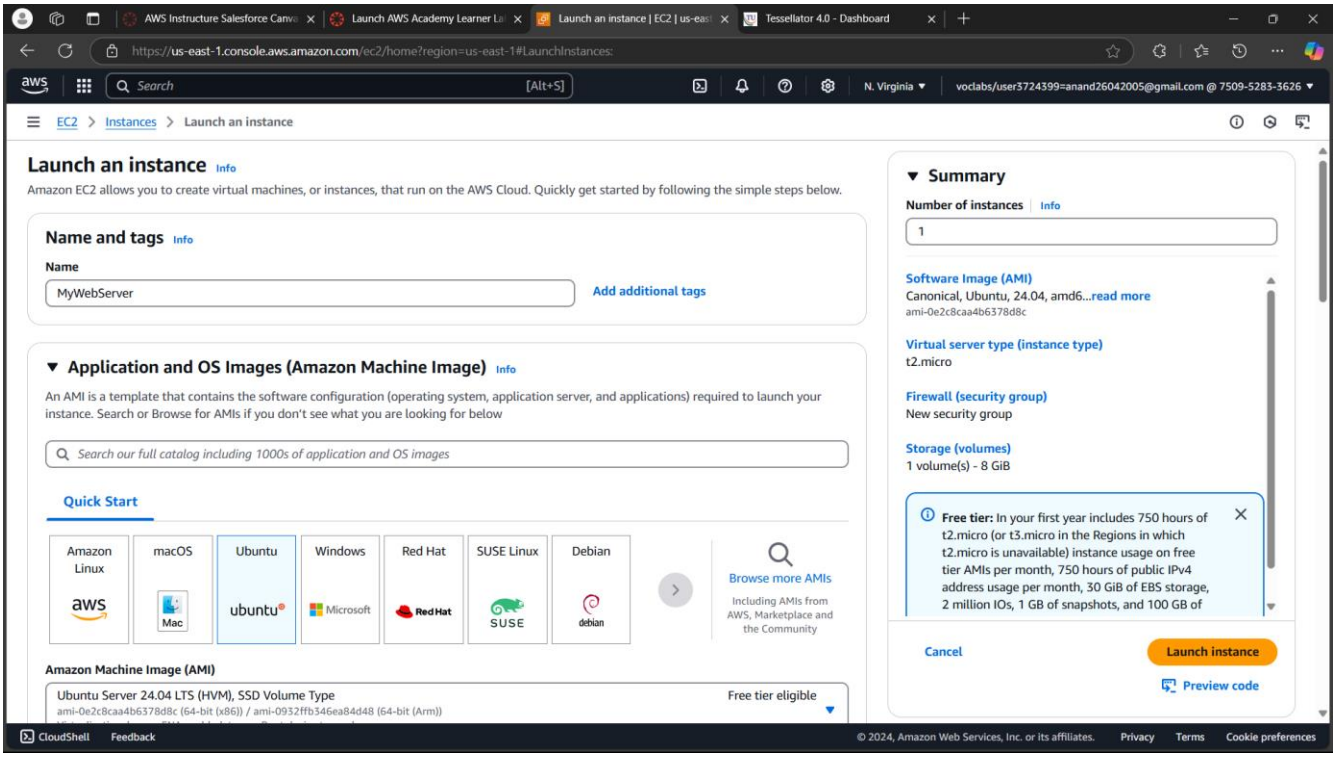
1. Login to AWS dashboard



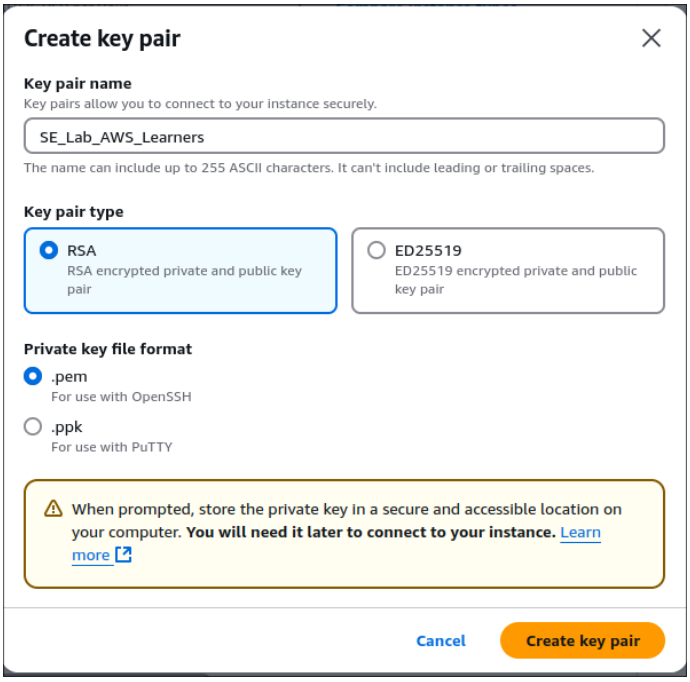

2. Click on launch instance



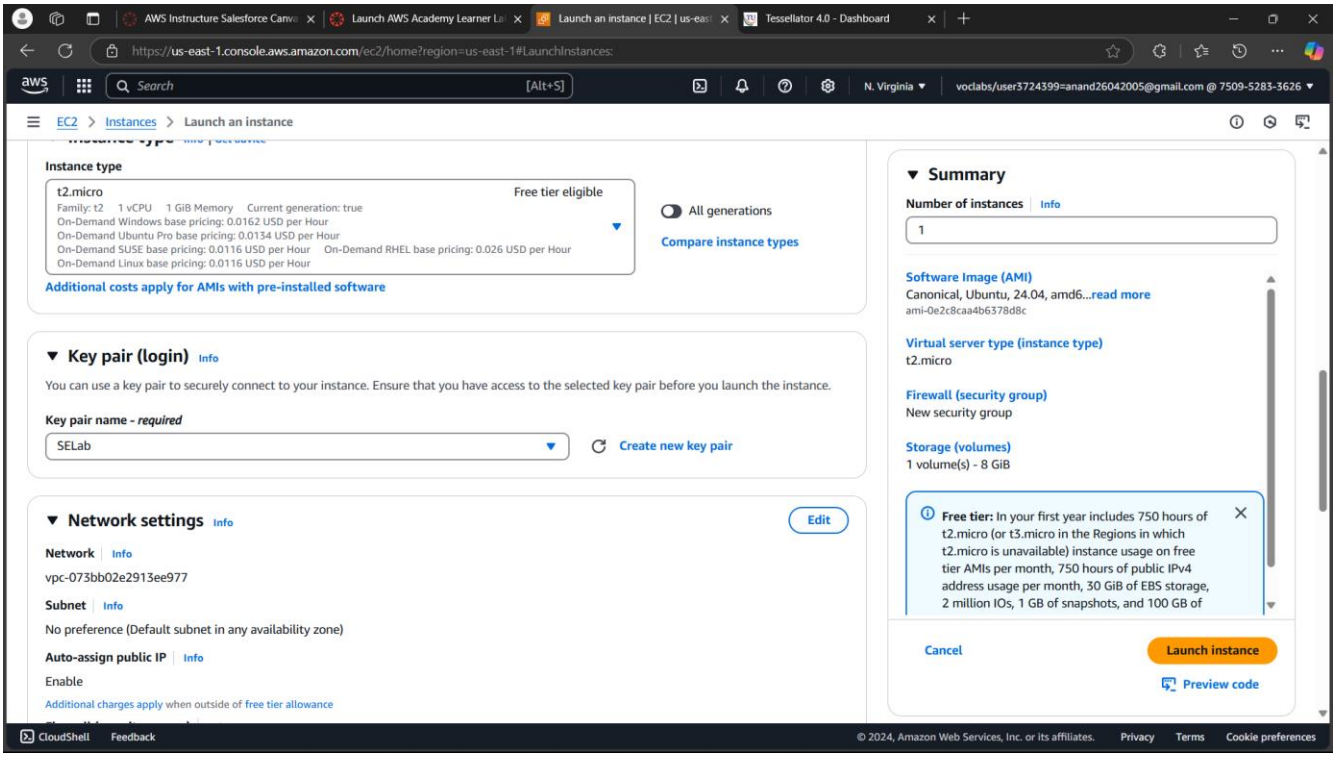

3. Select your choices



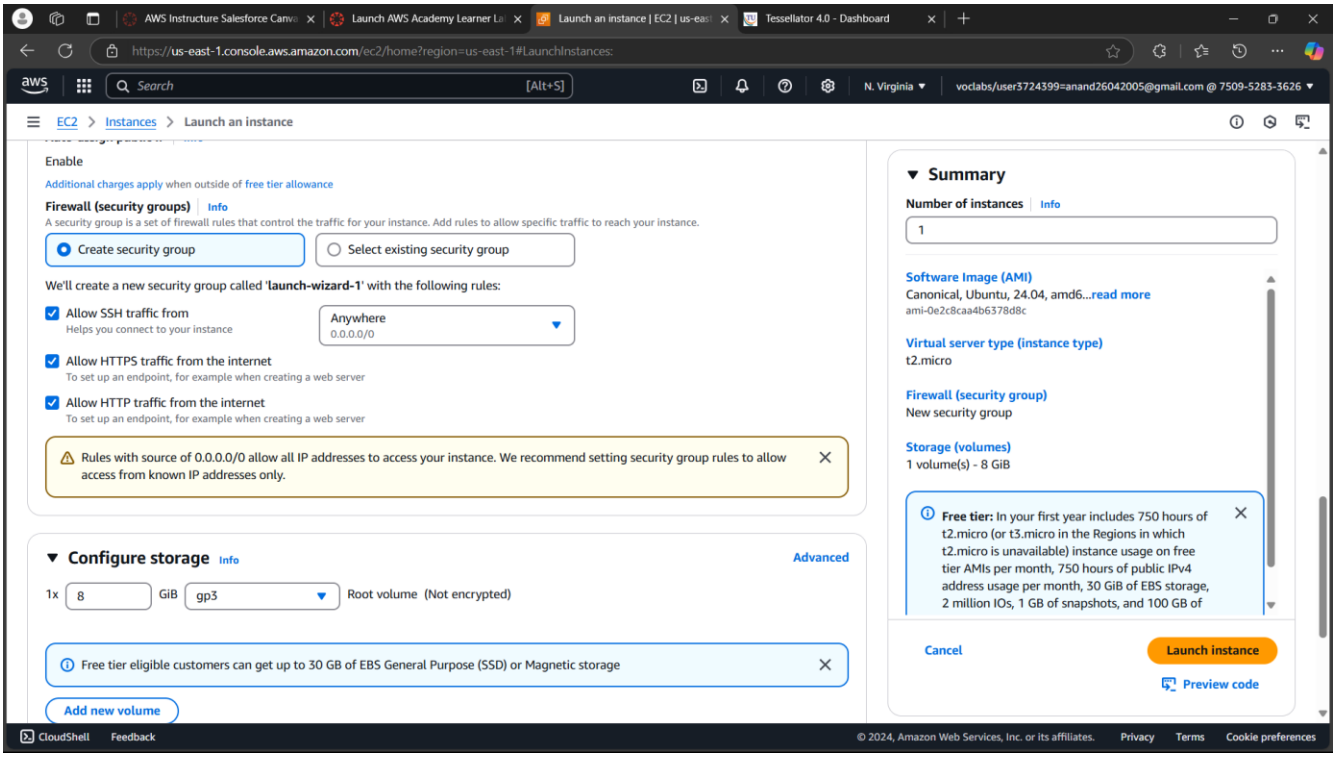

4. Create an SSH key pair



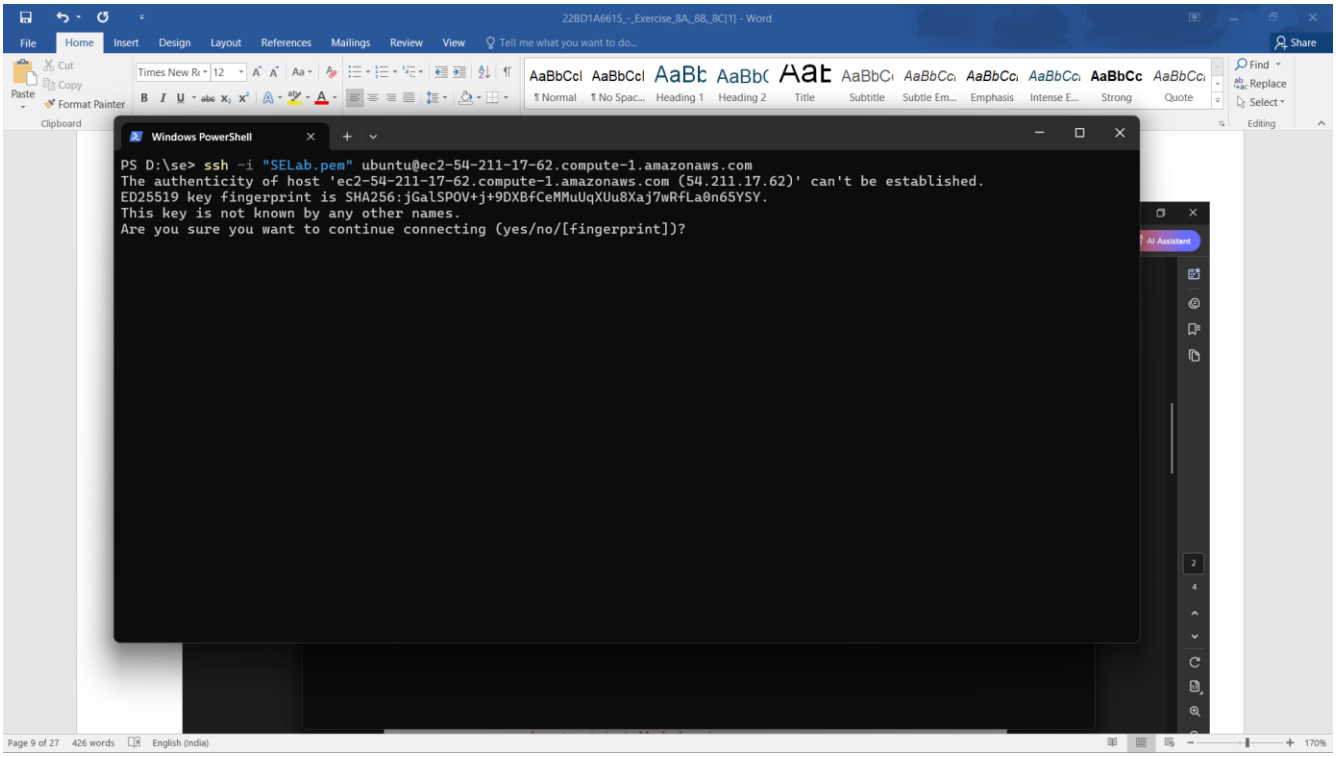

5. Allow traffic from everywhere



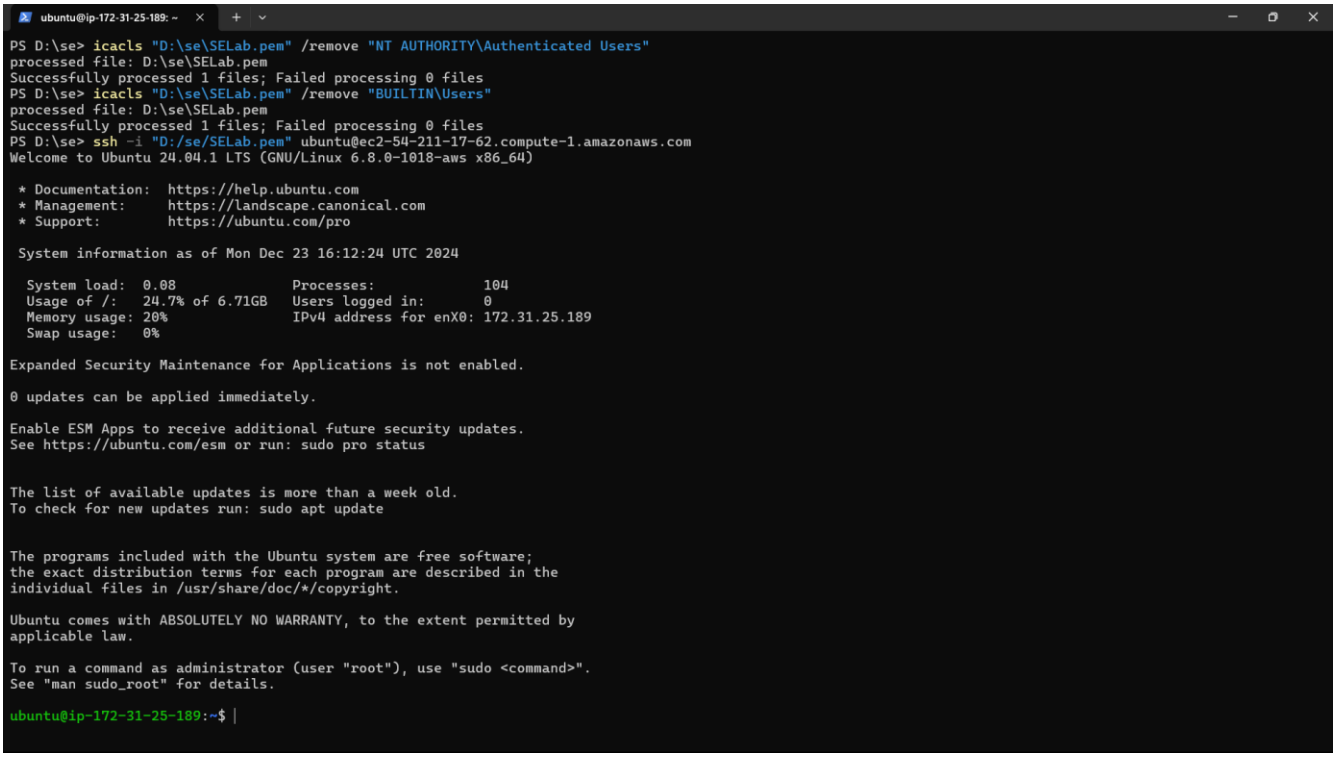

6. Use 8GB default storage



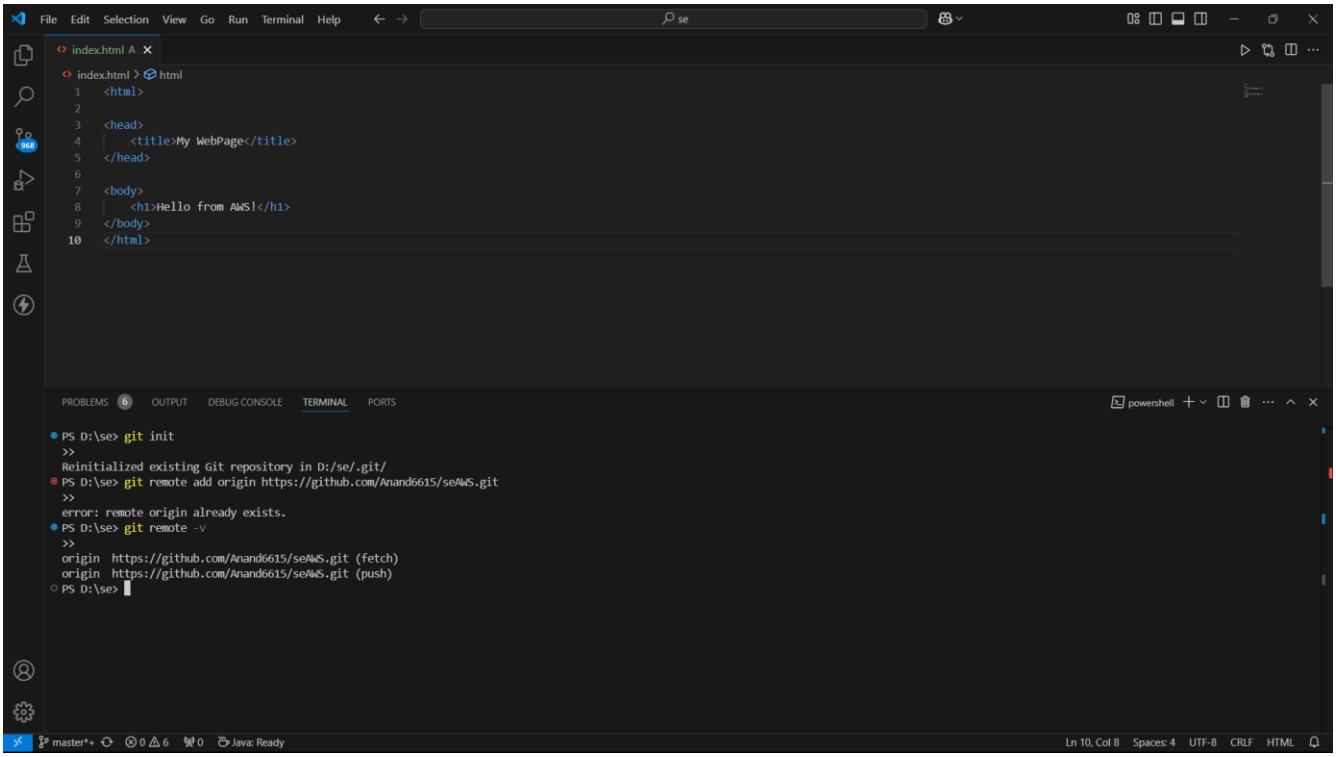

7. SSH into the instance



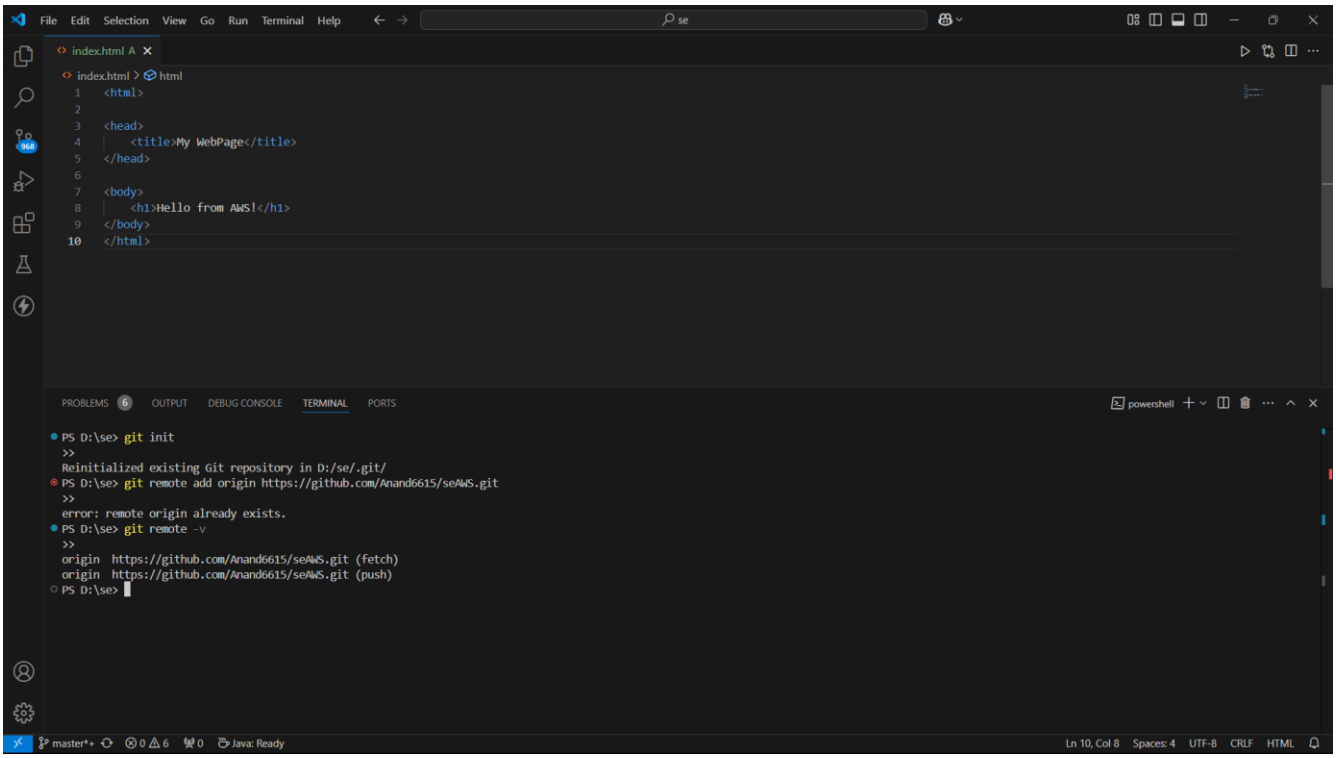

8. Install tools like Docker, git, vim, nano



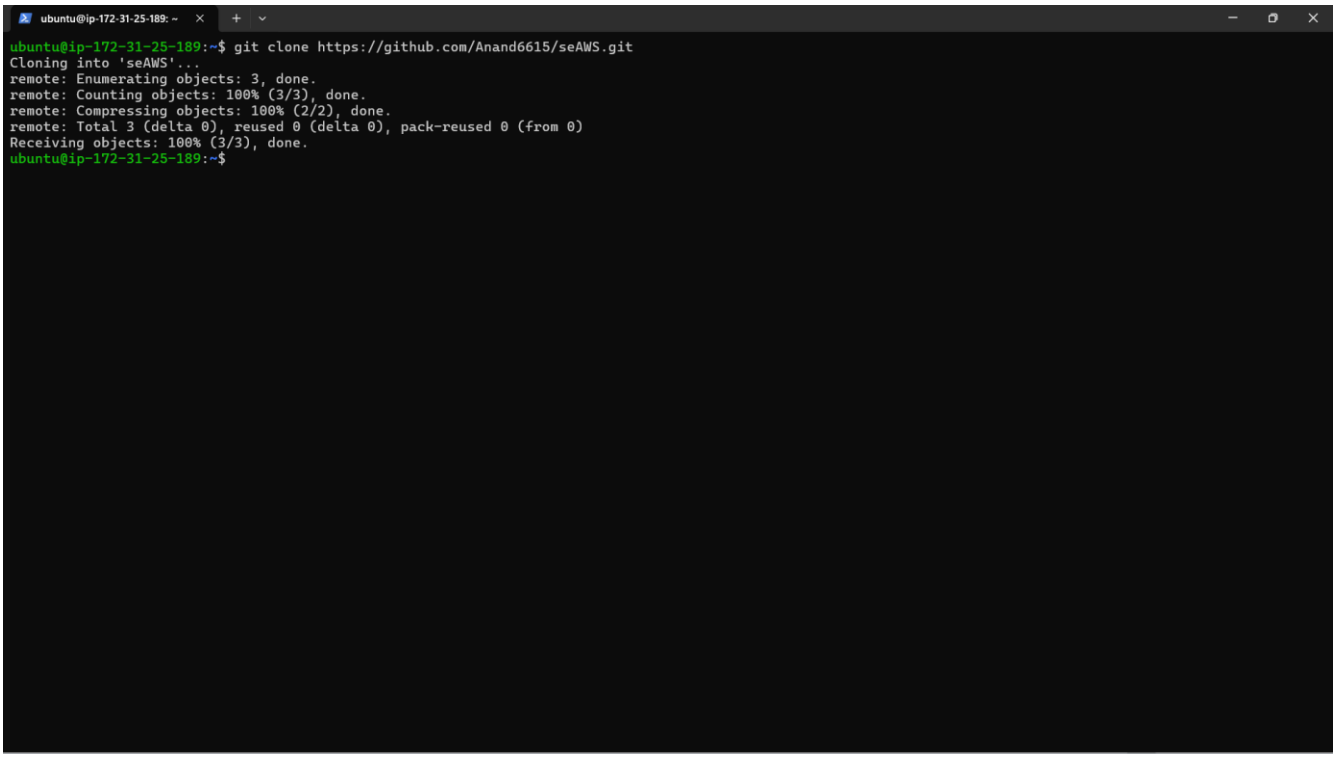

9. In your local system, create a folder and inside it create an index.html



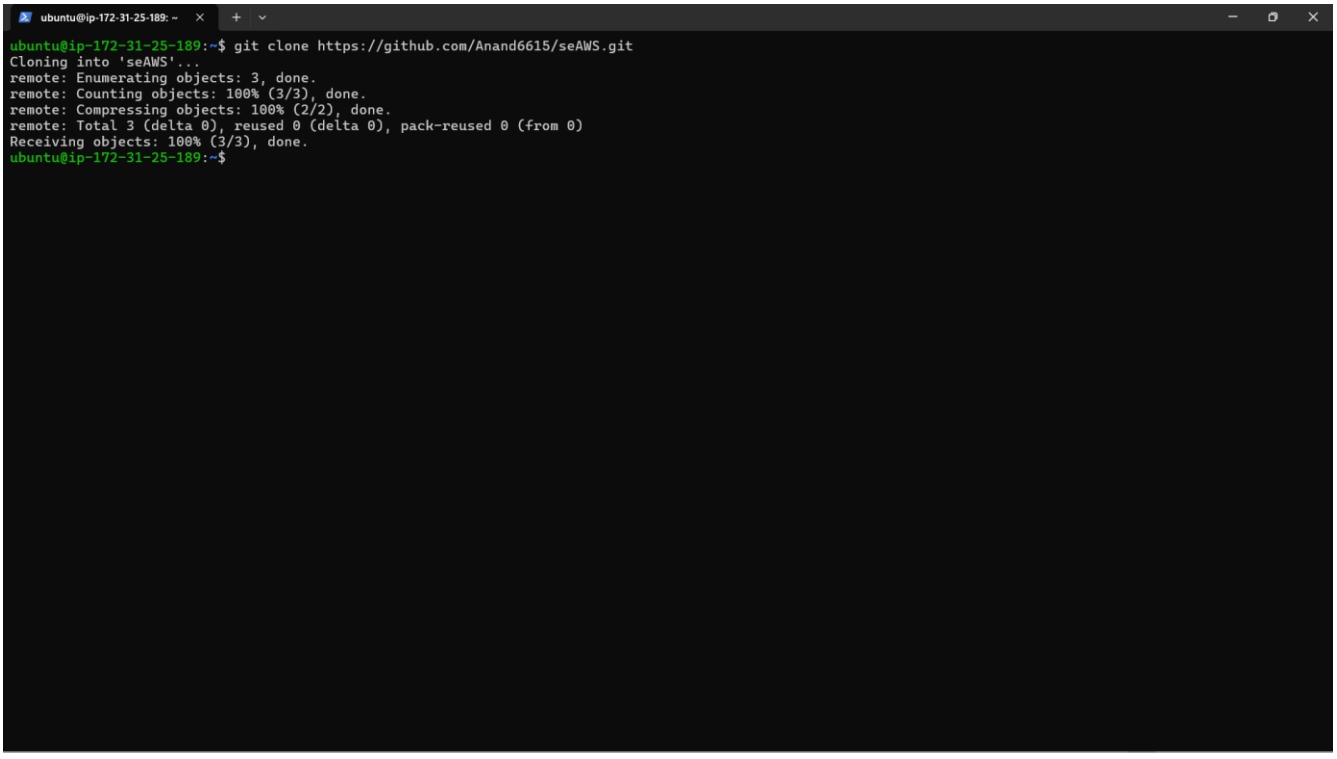

10. Initialise a git repo and commit your changes



11. Push into github

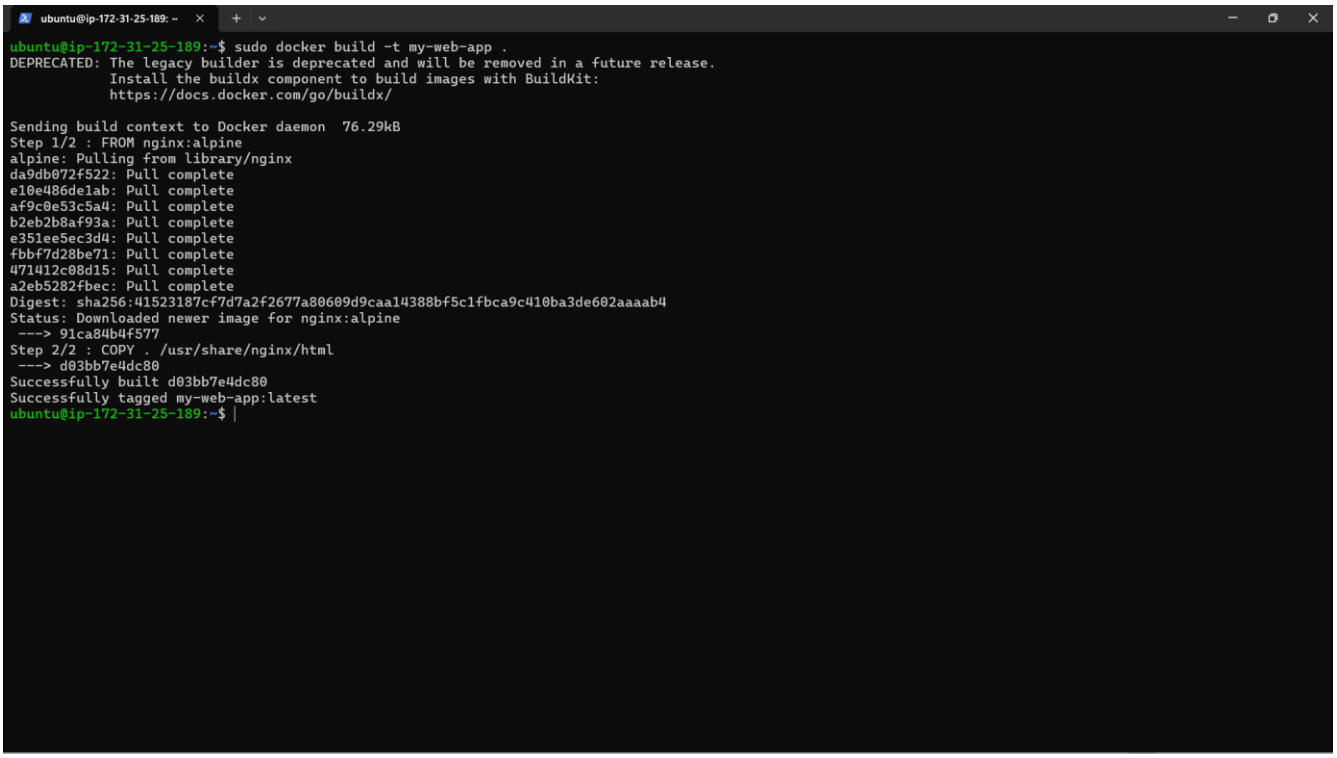



12. In the EC2 instance, clone your repo

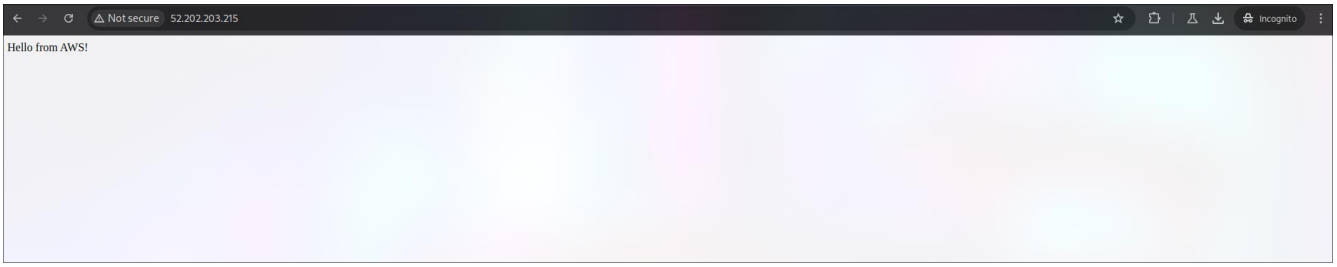


13. Create a dockerfile and add these lines

14. Create docker image and run the container



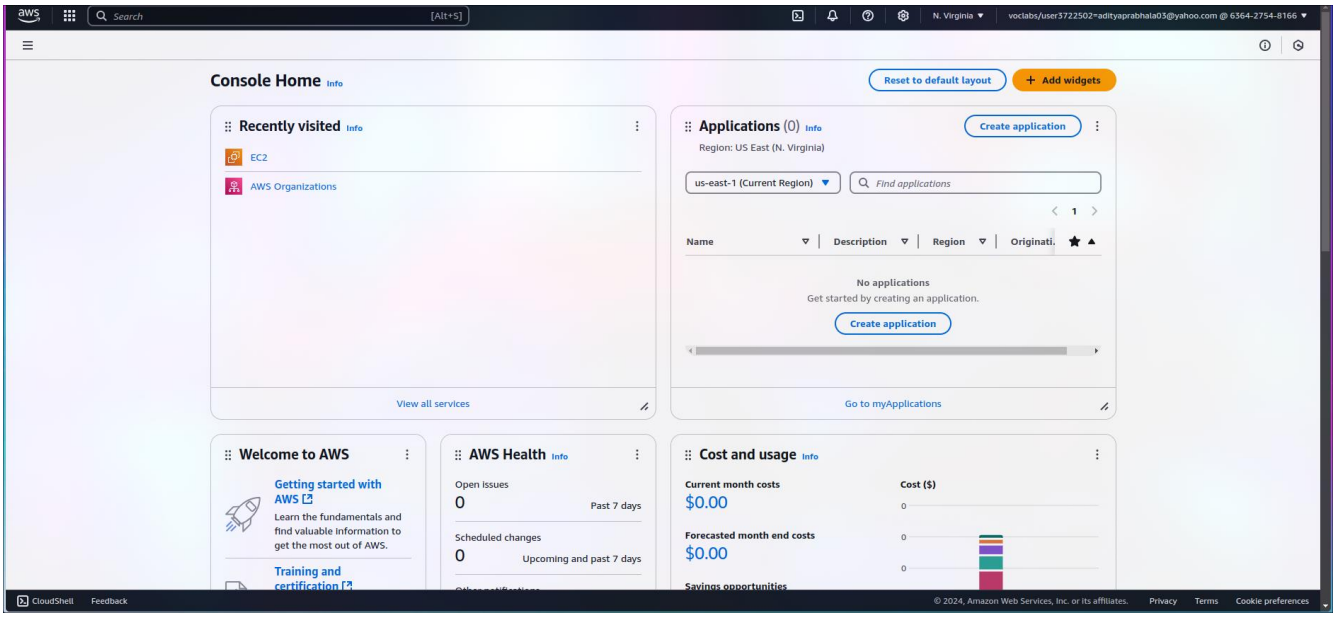

15. In your EC2 dashboard, copy your public IP address and open in browser



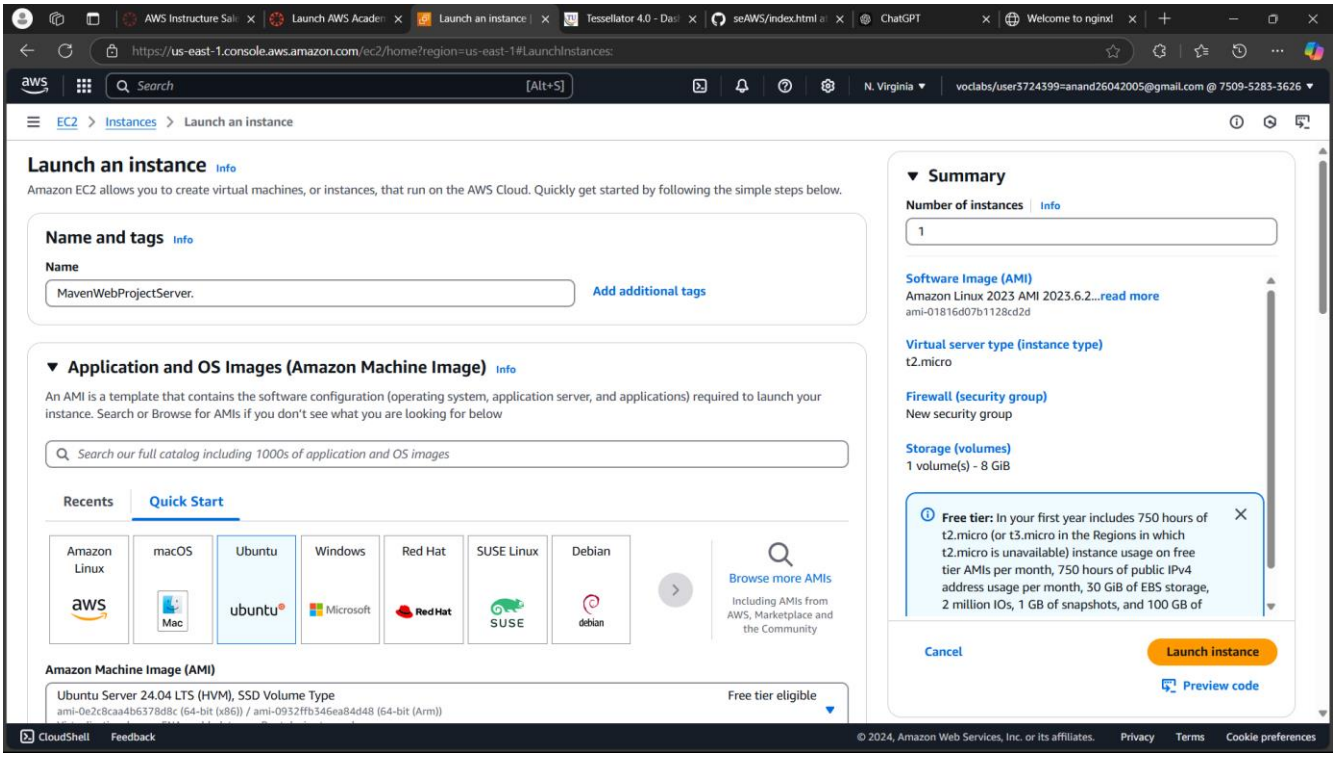

16. Finally stop your container


# AWS – Deploying Maven Web Project

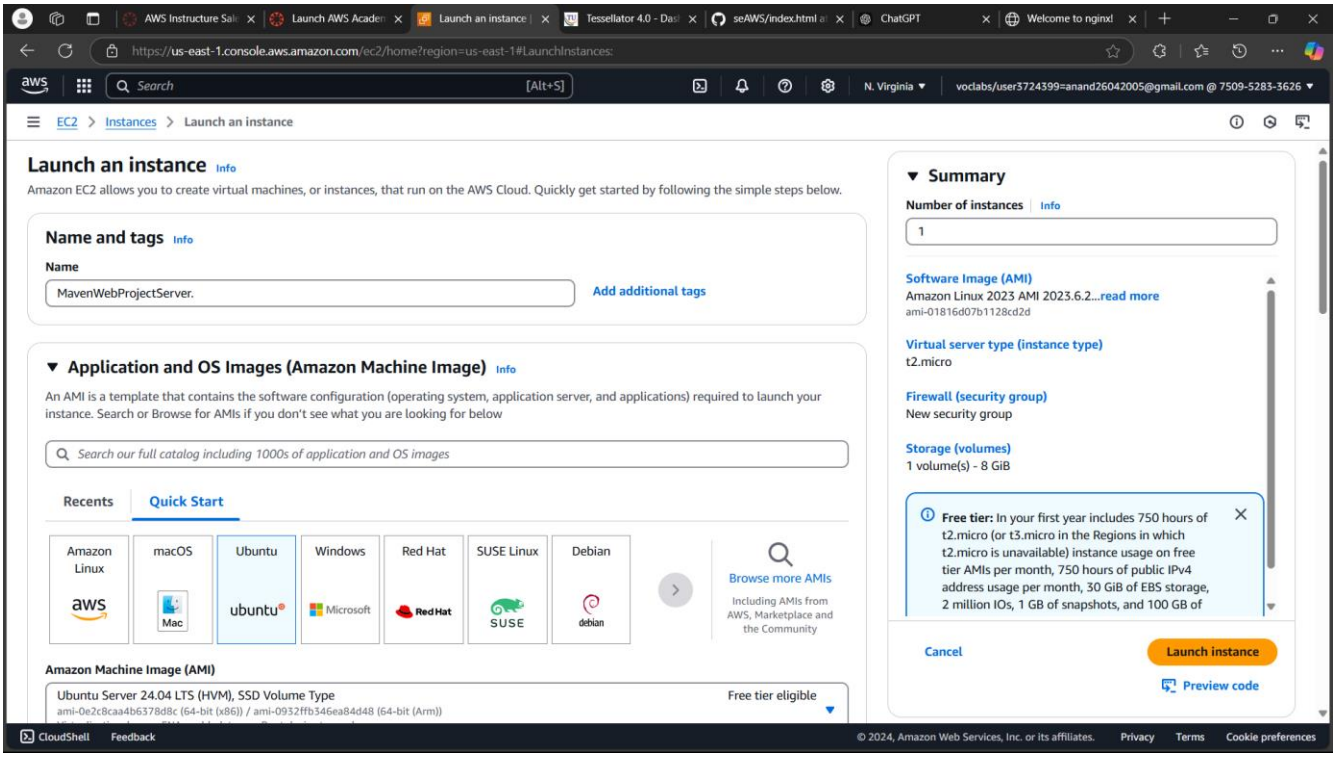
1. Login to AWS dashboard



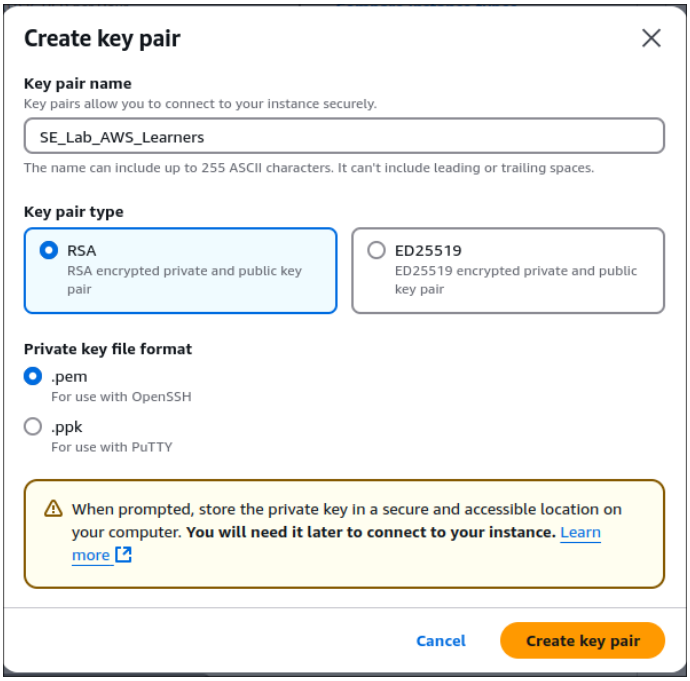
2. Click on launch instance



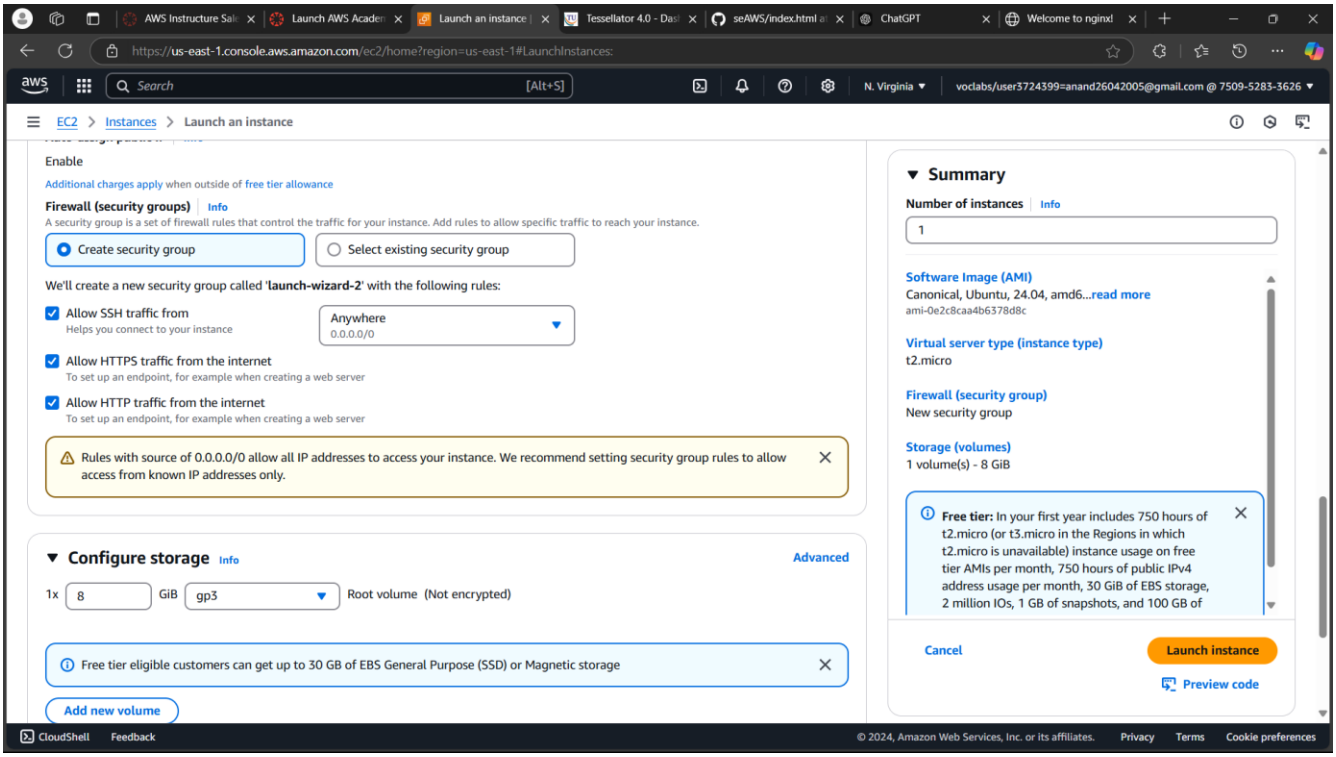

3. Select your choices



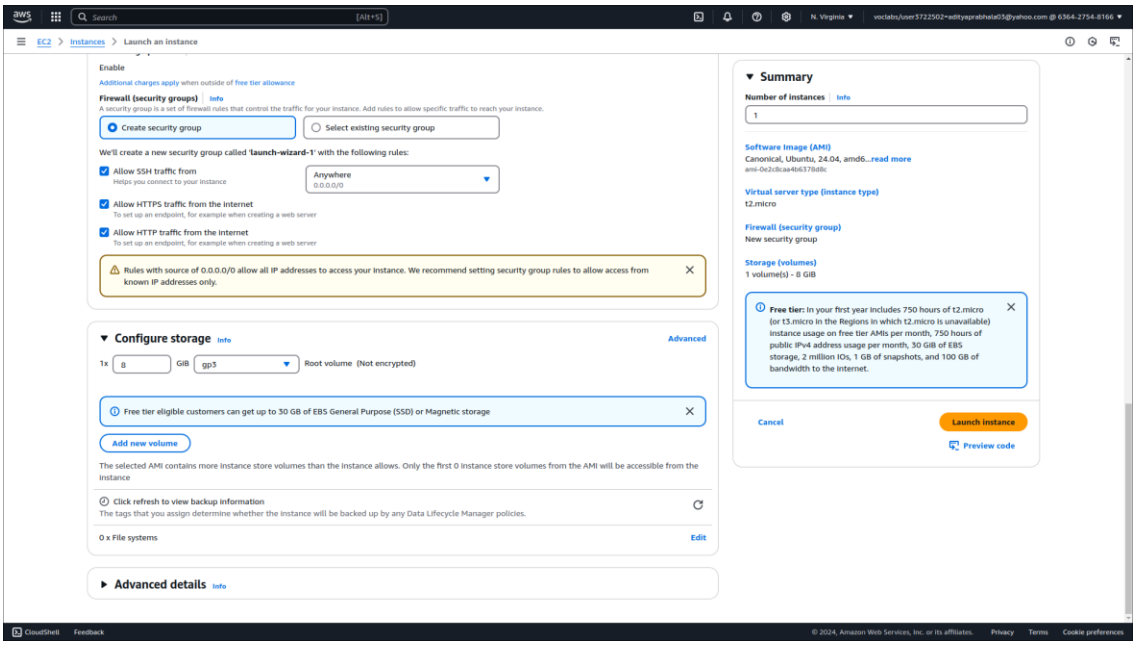
4. Create an SSH key pair



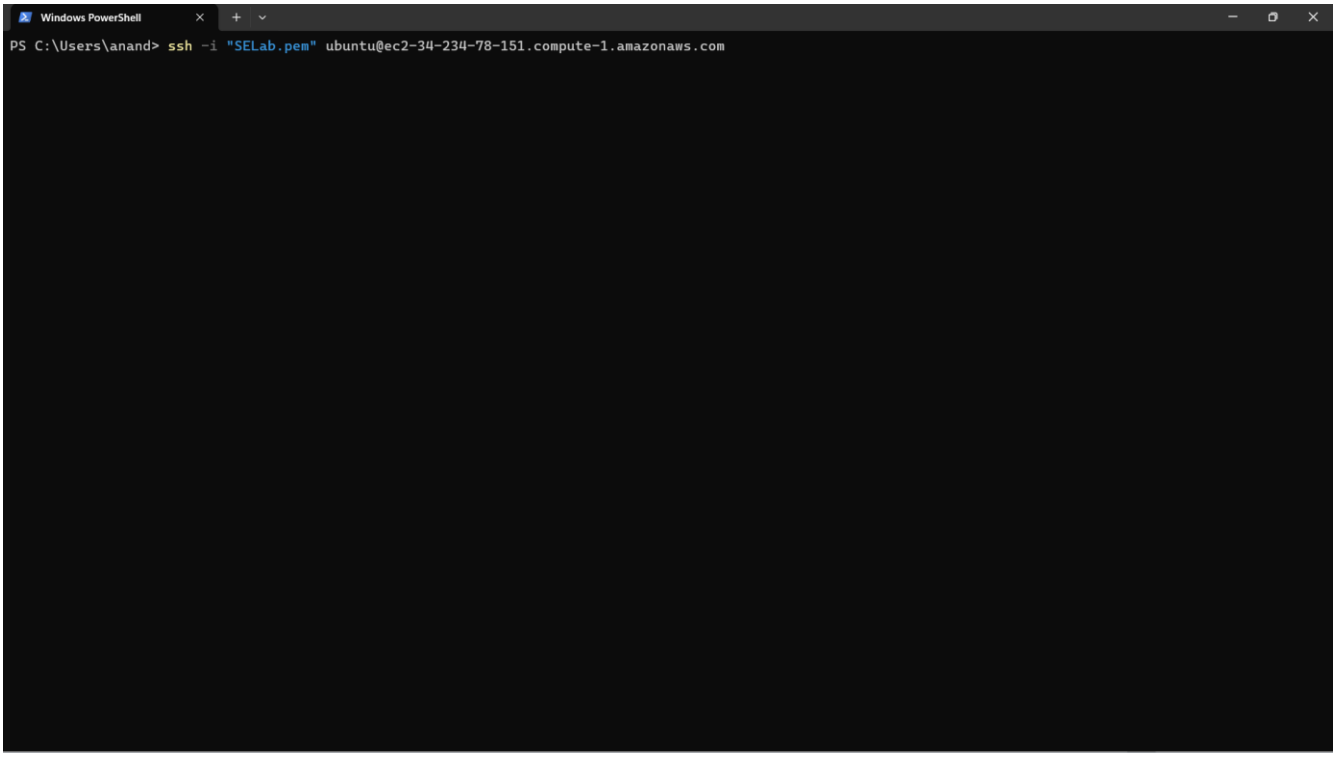

5. Allow traffic from everywhere

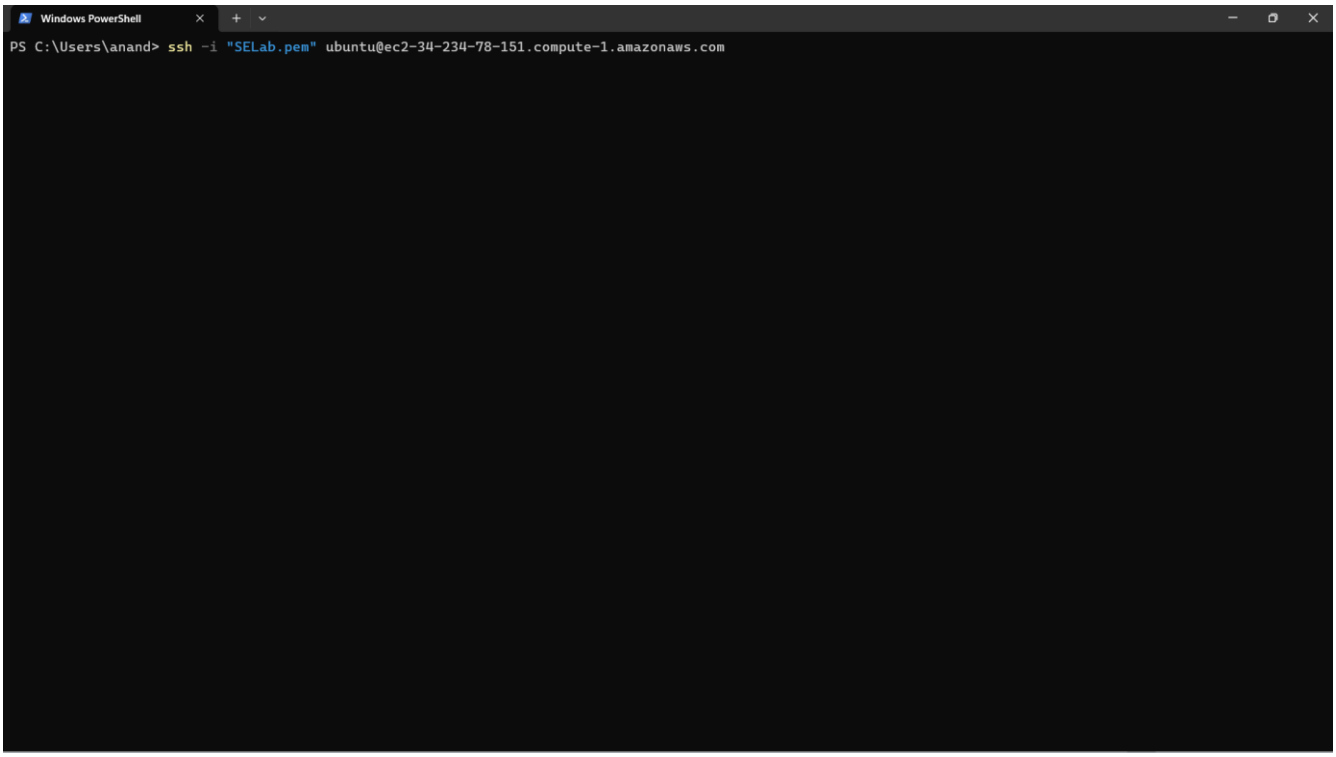



6. Use 8GB default storage

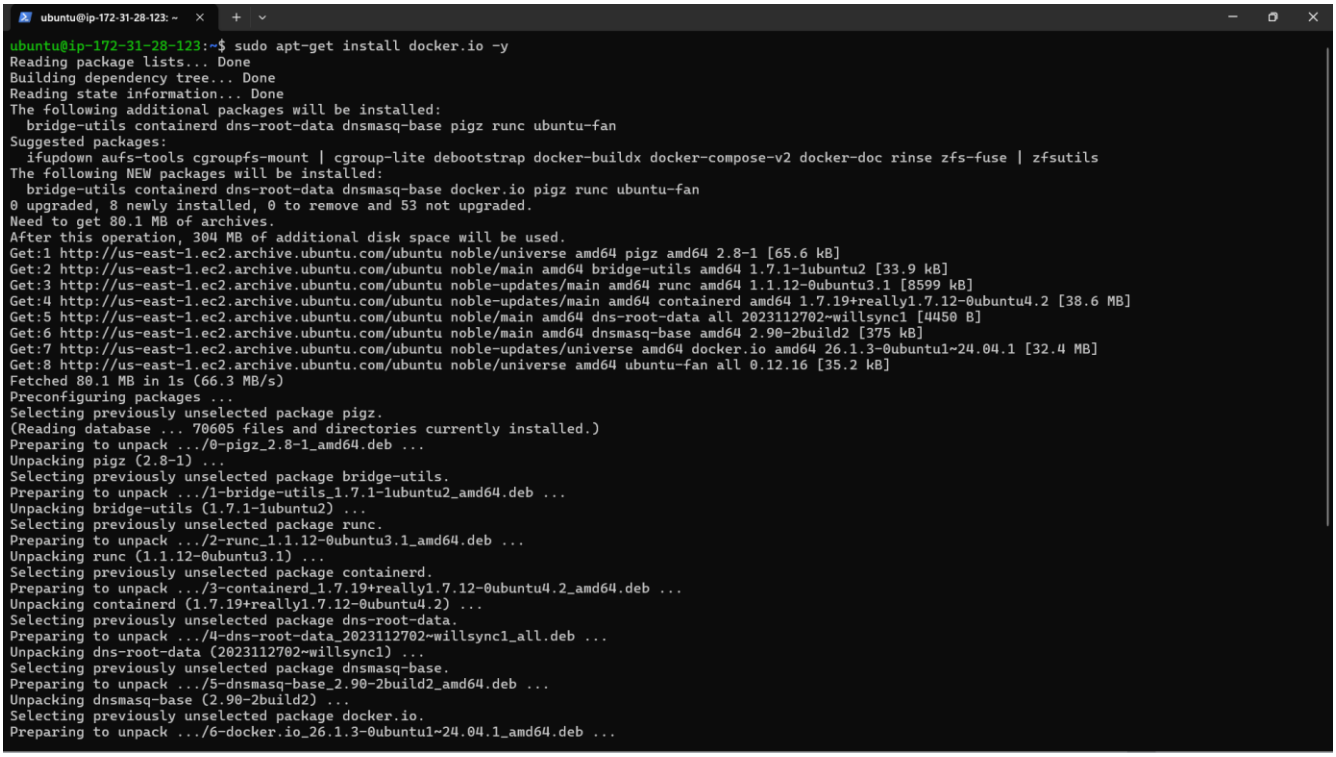



7. SSH into the instance

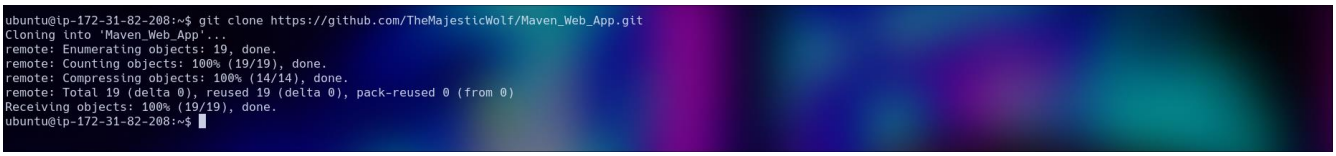




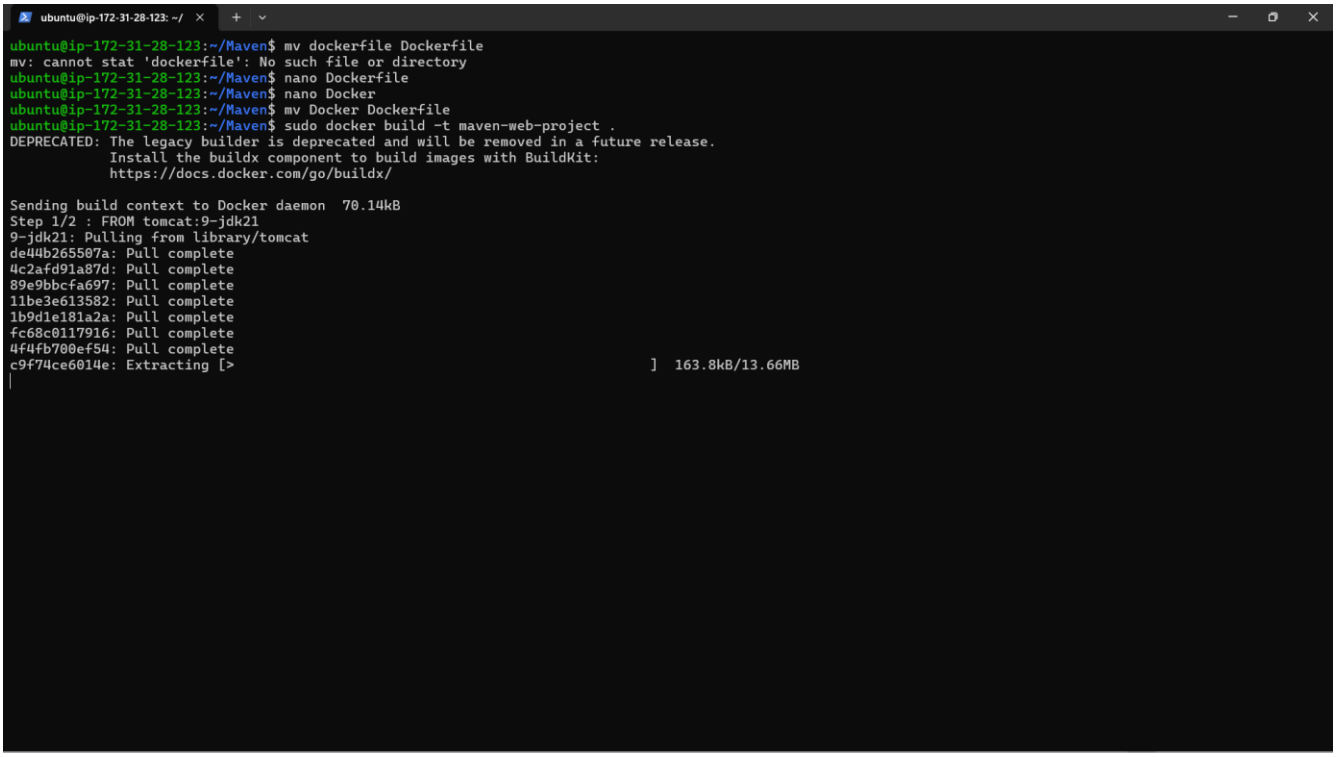

8. Install tools like Docker, git, vim, nano



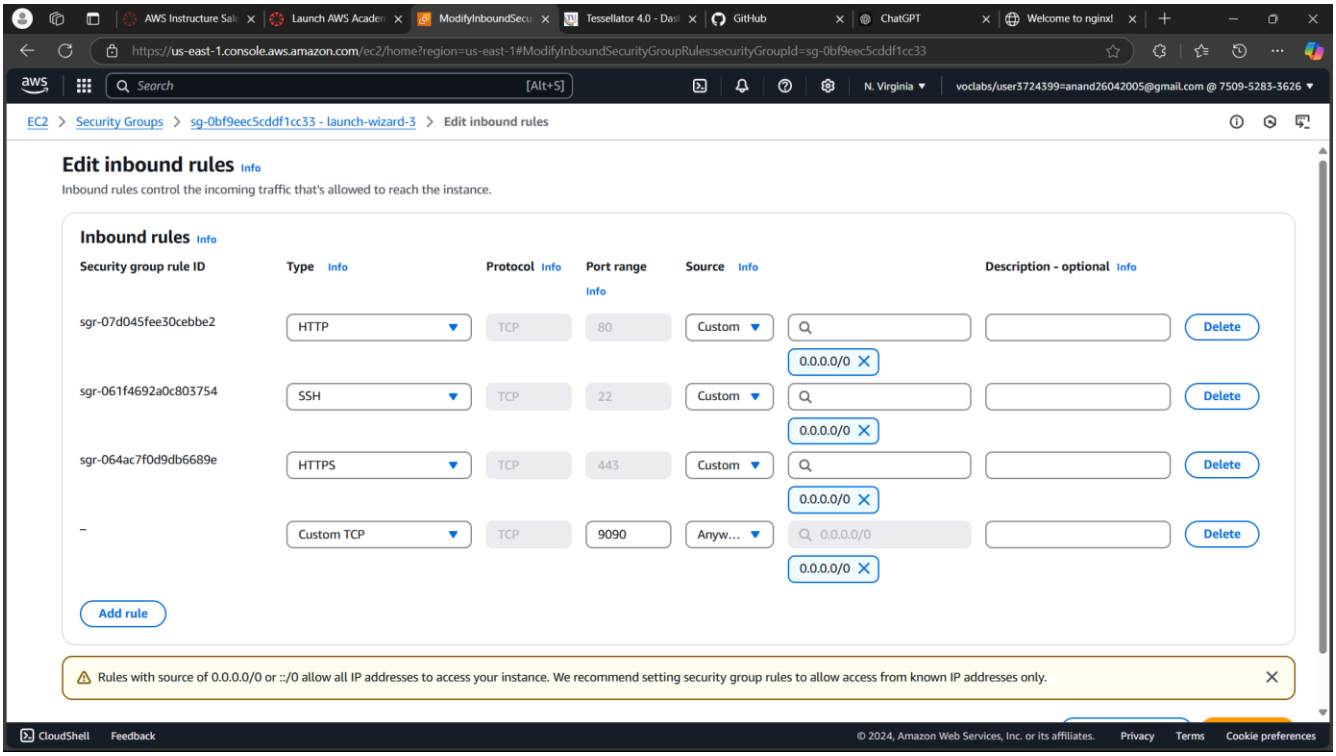

9. Clone Maven Web App




10. Create docker image and then run the container

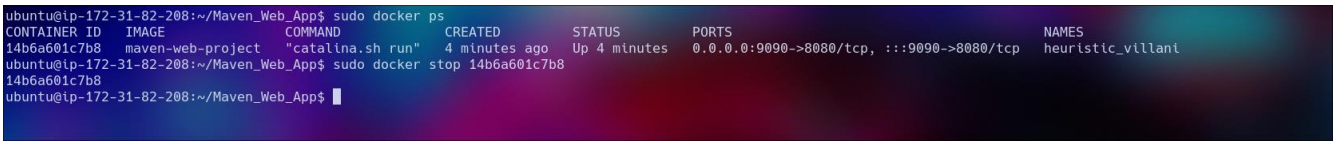



11. Add inbound rule to allow traffic



12. In browser, go to http://<public-ip>:9090/<project-name>


13. Stop your container




Result: An AWS Learning Account was created, a sample HTML file was deployed on AWS EC2, and the Maven web application was successfully hosted.